



2023 SSPD

Edinburgh, UK

2023 SENSOR SIGNAL PROCESSING FOR DEFENCE CONFERENCE (SSPD)

12th and 13th September 2023

Welcome

Programme

Keynote

Invited Speakers

Technical Committee



ISBN: 979-8-3503-3732-7 (Xplore)
979-8-3503-3731-0 (USB)

Part Number: CFP23SPD-ART (Xplore)
CFP23SPD-USB (USB)

2023 Sensor Signal Processing for Defence Conference (SSPD)

Copyright © 2023 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved.

Copyright and Reprint Permissions

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

For other copying, reprint or republication permission, write to IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854.
All rights reserved.

ISBN: 979-8-3503-3732-7 (Xplore)
 979-8-3503-3731-0 (USB)

Part Number: CFP23SPD-ART (Xplore)
 CFP23SPD-USB (USB)

Printed copies of this publication are available from:

Curran Associates, Inc
57 Morehouse Lane
Red Hook, NY 12571 USA
Phone: (845) 758-0400
Fax: (845) 758-2633
E-mail: curran@proceedings.com

IEEE eXpress
Conference
Publishing

Produced by **IEEE eXpress Conference Publishing**

For information on producing a conference proceedings and receiving an estimate, contact
conferencepublishing@ieee.org
<http://www.ieee.org/conferencepublishing>

Table of Contents

SSPD 2023 Welcome Message	v
SSPD 2023 Programme	vi
SSPD 2023 Keynote Speakers	ix
SSPD 2023 Invited Speakers.....	x
SSPD 2023 Conference Committee	xiii
Session 1: Navigation and Tracking	
1.1 Adaptive Kernel Kalman Filter for Magnetic Anomaly Detection-Based Metallic Target Tracking.....	1
<i>Mengwei Sun, Richard Hodgskin-Brown, Mike E. Davies, Ian K. Proudler, and James R. Hopgood</i>	
1.2 Implementation of Adaptive Kernel Kalman Filter in Stone Soup	6
<i>James S. Wright, James R. Hopgood, Mike E. Davies, Ian K. Proudler, and Mengwei Sun</i>	
Session 2: Poster Presentations	
2.1 A Lower Complexity Deep Learning Method for Drones Detection.....	11
<i>Mohamad Kassab, Amal El Fallah Seghrouchni, Frederic Barbaresco, and Raed Abu Zitar</i>	
2.2 Kalman Filter-Based Suspicious Object Tracking for Border Security and Surveillance Systems using Fixed Automotive Radar.....	16
<i>Ji-il Park, SeungHyeon Jo, Hyung-Tae Seo, Keun Ha Choi, Jihyuk Park, and Kyung-Soo Kim</i>	
2.3 Joint Learning with Shared Latent Space for Self-Supervised Monaural Speech Enhancement.....	21
<i>Yi Li, Yang Sun, Wenwu Wang, and Syed Mohsen Naqvi</i>	
2.4 Underwater Passive Target Classification Based on β Variational Autoencoder and MFCC.....	26
<i>Adarsh Sunilkumar, K. Shamju Joseph, and K. Manoj Kumar</i>	
2.5 Association Based Feedback Aided Underwater Passive Target Tracking.....	31
<i>Adarsh Sunilkumar, K. Shamju Joseph, and K. Manoj Kumar</i>	
2.6 Computational Enhancement of Accumulated CA-CFAR Process in Side Scan Sonar Data.....	36
<i>Ansila V. M., Bibin Basheer, and Sooraj K. Ambat</i>	
2.7 Multi-Target Tracking using a Swarm of UAVs by Q-Learning Algorithm	41
<i>Seyed Ahmad Soleymani, Shidrokh Goudarzi, Xingchi Liu, Lyudmila Mihaylova, Wenwu Wang, and Pei Xiao</i>	
2.8 Generalised Sequential Matrix Diagonalisation for the SVD of Polynomial Matrices	46
<i>Faizan A. Khattak, Ian K. Proudler, John G. McWhirter, and Stephan Weiss</i>	

Session 3: Multi-Sensor Multi-Target Tracking Detection

3.1 A Novel Adaptive Architecture: Joint Multi-Targets Detection and Clutter Classification	51
<i>Linjie Yan, Carmine Clemente, Sudan Han, Chengpeng Hao, Danilo Orlando, and Giuseppe Ricci</i>	

3.2 Consensus-Based Distributed Variational Multi-Object Tracker in Multi-Sensor Network	56
<i>Qing Li, Runze Gan, and Simon Godsill</i>	

3.3 Joint Sensor Scheduling and Target Tracking with Efficient Bayesian Optimisation	61
<i>Xingchi Liu, Chenyi Lyu, Seyed Ahmad Soleymani, Wenwu Wang, and Lyudmila Mihaylova</i>	

Session 4: Images and Video

4.1 Simulation of Anisoplanatic Turbulence for Images and Videos	66
<i>D. Vint, G. Di Caterina, P. Kirkland, R. A. Lamb, and D. Humphreys</i>	

4.2 Investigation of an End-to-End Neural Architecture for Image-Based Source Term Estimation.....	71
<i>A. Abdulaziz, Y. Altmann, S. McLaughlin, and M. E. Davies</i>	

Session 6: Sonar, Radar and Maritime

6.1 Random Sampling for Robust Detection of Data modulated LFM Waveforms.....	76
<i>Kaiyu Zhang, Fraser K. Coutts, and John Thompson</i>	

6.2 Generalized Polynomial Power Method	81
<i>Faizan A. Khattak, Ian K. Proudler, and Stephan Weiss</i>	

6.3 Joint Optimization of Sonar Waveform Selection and Sonobuoy Placement.....	86
<i>Christopher M. Taylor, Simon Maskell, Alexey Narykov, and Jason F. Ralph</i>	

6.4 Development of the Line Scatterer Model for Bistatic Wind Turbine Clutter.....	91
<i>Juhani Nissila, Pasi Pertilä, Minna Väilä, and Juha Jylhä</i>	

6.5 DB-Drift: Concept Drift Aware Density-Based Anomaly Detection for Maritime Trajectories.....	96
<i>Amelia Henriksen</i>	

Author Index	101
---------------------------	------------

SSPD Conference 2023 - Welcome

Dear Colleagues,

We warmly welcome you to this year's SSPD Conference. This event is the 12th conference of the Sensor Signal Processing for Defence series and provides a chance to present, listen to and discuss the latest scientific findings in signal processing for defence.

A welcome also extends to our keynote and invited speakers as well as our panel speakers from Defence, Industry and Academia and the presenters of scientific papers presenting their novel research through live oral presentations. We look forward to some interesting debate and discussion throughout the conference.

We would like to take this opportunity to thank the speakers, reviewers, session chairs and the technical committee for their contribution to this event.

We hope you enjoy our conference.

Mike Davies

Steve McLaughlin

Jordi Barr

Gary Heald

Chairs of the Organising Committee

Technical sponsorship is provided by IEEE Signal Processing Society. Proceedings will be submitted to the Xplore Digital Library. The conference is organised by the University Defence Research Collaboration (UDRC) in Signal Processing, sponsored by the Defence Science and Technology Laboratory (Dstl) and the Engineering and Physical Sciences Research Council (EPSRC).



Programme - SSPD2023

Tuesday 12th September 2023

8:30 to 9:00 Refreshments

Session 1 -Navigation and Tracking- Mike Davies, University of Edinburgh

9:00 Introduction and Welcome to Day 1/Session 1 – Mike Davies, University of Edinburgh.

9:10 – 10:10 Keynote Speaker: Instabilities in Navigation - Balancing on the Head of a Pin, Jason Ralph, University of Liverpool.

10:10 – 10:35 Adaptive Kernel Kalman Filter for Magnetic Anomaly Detection-based Metallic Target Tracking, Mengwei Sun¹, Ian Proudler², Mike E Davies¹, James R Hopgood¹, ¹University of Edinburgh, ²University of Strathclyde.

10:35 – 11:00 Refreshments

11:00 – 11:25 Implementation of Adaptive Kernel Kalman Filter in Stone Soup, James Wright¹, James R Hopgood², Mike E Davies², Ian Proudler³, Mengwei Sun², ¹Dstl, ²University of Edinburgh, ³University of Strathclyde.

Session 2 - Panel Discussion and Posters – Chair – Jordi Barr - Dstl

11:25 Introduction and Welcome to Session 2 – Jordi Barr, Dstl

11:25 – 12:25 Panel Discussion: The Future for Defence Signal Processing.

12:25 - 14:30 Poster Presentations and Lunch

- **P1.** A Lower Complexity Deep Learning Method for Drones Detection, Amal El-Fallah-Seghrouchni¹, Frederic Barbaresco², Mohamad Kassab³, Raed Abu Zitar⁴, ¹University of Pierre and Marie Curie, ²Thales Air Systems, ³Mohamad Bin Zayed University of Artificial Intelligence, ⁴Sorbonne University-Abu Dhabi.
- **P2.** Kalman Filter-Based Suspicious Object Tracking for Border Security and Surveillance System using Fixed Automotive Radar, Ji-il Park¹, SeungHyeon Jo², Hyung-Tae Seo³, Keun Ha Choi⁴, Jihyuk Park⁵, Kyung-Soo Kim⁴, ¹Ministry of National Defense, ²DXC Luxoft, ³Kyonggi University, ⁴KAIST, ⁵Automotive Engineering & Yeungnam University.
- **P3.** Joint Learning with Shared Latent Space for Self-Supervised Monaural Speech Enhancement, Yi Li¹, Yang Sun², Wenwu Wang³, Syed Mohsen Naqvi⁴, ¹Lancaster University, ²University of Oxford, ³University of Surrey, ⁴Newcastle University.
- **P4.** Underwater Passive Target Classification based on β Variational Autoencoder and MFCC, Adarsh Sunilkumar¹, Shamju Joseph K¹, Manoj Kumar K¹, ¹Naval Physical Oceanographic Laboratory.

- **P5.** Association based Feedback Aided Underwater Passive Target Tracking, Adarsh Sunilkumar¹, Shamju Joseph K¹, Manoj Kumar K¹, ¹Naval Physical Oceanographic Laboratory.
- **P6.** Computational Enhancement of Accumulated CA-CFAR Process in Side Scan Sonar Data, Ansila Veliyathparambil Muhamedali¹, Bibin Basheer¹, Sooraj K. Ambat¹, ¹Defence Research and Development Organisation.
- **P7.** Multi-Target Tracking Using a Swarm of UAVs by Q-learning Algorithm, Seyed Ahmad Soleymani¹, Shidrokh Goudarzi², Xingchi Liu³, Lyudmila Mihaylova³, Wenwu Wang¹, Pei Xiao¹, ¹University of Surrey, ²University of West London, ³University of Sheffield.
- **P8.** Generalised Sequential Matrix Diagonalisation for the SVD of Polynomial Matrices, Faizan Khattak¹, Ian Proudler¹, John G McWhirter², Stephan Weiss¹, ¹University of Strathclyde, ²Cardiff University.

14:30 Introduction and Welcome to Session 3 – James Hopgood, University of Edinburgh

14:30 – 15:00 Invited Speaker: Adiabatic computing for low power image sensing, ¹Alexandrou Serb, ¹University of Edinburgh.

Session 3 – Multi-sensor Multi-target Tracking Detection – Chair – James Hopgood – University of Edinburgh

15:00 – 15:25 A Novel Adaptive Architecture: Joint Multi-targets Detection and Clutter Classification, Linjie Yan¹, Carmine Clemente², Sudan Han³, Chengpeng Hao¹, Danilo Orlando⁴, Giuseppe Ricci⁵, ¹Institute of Acoustics, Chinese Academy of Sciences, ²University of Strathclyde, ³National Innovation Institute of Defense Technology, ⁴Universita' degli Studi Niccolo' Cusano, ⁵University of Salento.

15:25 – 15:45 Refreshments

15:45 – 16:10 Consensus-based Distributed Variational Multi-object Tracker in Multi-Sensor Network, Qing Li¹, Runze Gan¹, Simon Godsill¹, ¹University of Cambridge.

16:10 – 16:35 Joint Sensor Scheduling and Target Tracking with Efficient Bayesian Optimisation, Xingchi Liu¹, Chenyi Lyu¹, Seyed Ahmad Soleymani², Wenwu Wang², Lyudmila Mihaylova¹, ¹University of Sheffield, ²University of Surrey.

16:35 Closing remarks

19:30 Conference Reception Drinks – Royal College of Physicians

20:00 Conference Dinner

Wednesday 13th September 2023**8:30 to 9:00 Refreshments**

9:00 Introduction and Welcome to Day 2/Session 4 – Images and Video – Steve McLaughlin – Heriot-Watt University

9:05 – 10:05 Keynote Speaker: TBC

10:05 – 10:35 Invited Speakers: Machine Learning for Defence Signal Processing and Communications, Kin Leung and Thanos Gkelias, Imperial College London.

10:35 – 11:05 Refreshments**Session 4 Images and Video – Chair – Steve McLaughlin, Heriot-Watt University**

11:05 – 11:30 Simulation of Anisoplanatic Turbulence for Images and Videos, David Vint¹, Gaetano Di Caterina¹, Robert Lamb², David Humphreys², Paul Kirkland¹, ¹University of Strathclyde, ²Leonardo.

11:30 – 11:55 Investigation of an end-to-end neural architecture for image-based source term estimation, Abdullah Abdulaziz¹, Mike E Davies², Yoann Altmann¹, Steve McLaughlin¹, ¹Heriot-Watt University, ²University of Edinburgh.

Session 5 – Military Panel Discussion – Chair – Jordi Barr - Dstl

11:55 Introduction and Welcome to Session 5 – Cdr Nick Jones, Dstl

11:55 – 12:55 Panel Discussion: Military Panel: Signal processing – the user experience.

12:55 – 13:55 Lunch**Session 6 – Sonar, Radar and Maritime – Chair – Gary Heald, Dstl**

13:55 Introduction and Welcome to Session 6 – Gary Heald, Dstl

13:55 – 14:20 Random Sampling for Robust Detection of Data modulated LFM Waveforms, Kaiyu Zhang¹, Fraser K Coutts¹, John Thompson¹, ¹University of Edinburgh.

14:20 – 14:45 Generalised Polynomial Power Method, Faizan Khattak¹, Ian Proudler¹, Stephan Weiss¹, ¹University of Strathclyde.

14:45 – 15:10 Refreshments

15:10 – 15:35 Joint Optimization of Sonar Waveform Selection and Sonobuoy Placement, Christopher M Taylor¹, Jason F. Ralph¹, Simon Maskell¹, Alexey Narykov¹, ¹University of Liverpool.

15:35 – 16:00 Development of the Line Scatterer Model for Bistatic Wind Turbine Clutter, Juhani Nissilä¹, Pasi Pertilä¹, Minna Väilä¹, Juha Jylhä¹, ¹Patria Aviation Oy.

16:00 – 16:25 DB-Drift: Concept drift aware density-based anomaly detection for maritime trajectories -y, Amelia Henriksen¹, ¹Sandia National Laboratories.

16:25 Closing remarks

Keynote Speakers

Prof. Jason Ralph, University of Liverpool

Jason Ralph received a DPhil degree from the University of Sussex (UK) in 1993, and a BSc in Physics with Mathematics from the University of Southampton (UK) in 1989. Between 1992 and 1996 he was employed as a Research Fellow at the University of Sussex (UK), the Scuola Normale Superiore (Pisa, Italy), and the University of Auckland (New Zealand). He worked for the UK Defence Evaluation and Research Agency between 1997 and 1999, before taking up a Lecturer position in the Department of Electrical Engineering and Electronics at the University of Liverpool (UK). He was promoted to Senior Lecturer in 2004, a Reader (Associate Professor) in 2007, and a Chair in 2013. He was Head of the Department of Electrical Engineering and Electronics from 2012 to 2015. His research covers quantum technologies and quantum control, infrared



imaging and image processing, guidance and navigation, and target tracking algorithms. Professor Ralph is a Chartered Engineer (CEng), a Fellow of the Royal Aeronautical Society (UK), a Fellow of the Institute of Physics (UK), a member of the Royal Institute of Navigation (UK) and the American Institute of Aeronautics and Astronautics.

Abstract: Instabilities in Navigation - Balancing on the head of a pin

The problem of navigation with dead reckoning is one of instability. Many of the usual methods that we learn from school and college are problematic as a result. We can augment dead reckoning using alternatives that provide direct measurements of position, but none are perfect. This talk outlines the main problems associated with dead reckoning navigation and the use of inertial sensors to measure the dynamics of a vehicle/platform. We discuss current sensor technologies and the possible use of quantum technology to augment classical inertial sensing, and we try to highlight the difficulties in using signal processing in unstable systems of equations.

Invited Speakers

Dr Athanasios Gkelias and Prof Kin K. Leung, Imperial College London

Dr Athanasios Gkelias, Imperial College London

Athanasios (Thanos) Gkelias received his MEng in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Greece, and his MSc and PhD degrees from King's College London. Currently, he is a Research Fellow and project manager at Imperial. In the past, he served as the project manager of the University Defence Research Centre (UDRC) in Signal Processing at Imperial College, sponsored by the U.K. Dstl. He has been involved and made significant research contributions to several and diverse ICT projects funded by the EU, EPSRC, U.K. Dstl and U.S. Army. He has published more than 50 peer-reviewed journals, conference papers and book chapters. He was the co-recipient of the Best Student Paper Award in PIMRC 2012 and the IEEE Communications Society Best Survey Paper Award in 2022.



Prof Kin Leung, Imperial College London

Kin K. Leung received his Ph.D. degree from University of California, Los Angeles. He worked at AT&T Bell Labs in New Jersey from 1986 to 2004. Since 2004, he has been the Tanaka Chair Professor at Imperial College. He is also the Head of Communications and Signal Processing Group in the Electrical and Electronic Engineering Department. His research focuses on optimization and machine learning for large-scale communications, computer and sensor networks. He also works on multi-antenna and cross-layer designs for wireless networks.

He is a Fellow of the Royal Academy of Engineering, IEEE Fellow, IET Fellow, and member of Academia Europaea. He received the Distinguished Member of Technical Staff Award at AT&T Bell Labs and the Royal Society Wolfson Research Merits Award. He co-received the IEEE Communications Society (ComSoc) Leonard G. Abraham Prize (2021), the IEEE ComSoc Best Survey Paper Award (2022), the U.S.–UK Science and Technology Stocktake Award (2021), the Lanchester Prize Honorable Mention Award (1997), and several best conference paper awards. He is an IEEE ComSoc Distinguished Lecturer (2022-23). He was a member (2009-11) and the chairman (2012-15) of the IEEE Fellow Evaluation Committee for the ComSoc. He has served as editor for 10 IEEE and ACM journals and chaired the Steering Committee for the IEEE Transactions on Mobile Computing. Currently, he is an editor for the ACM Computing Survey and International Journal of Sensor Networks. Website: www.commsp.ee.ic.ac.uk/~kkleung/



Abstract: Machine Learning for Defence Signal Processing and Communications

Machine learning (ML) has been successfully applied to a very wide range of defence signal processing and communications problems. A few key challenges deserve further attention. First, there often is a lack of sufficient signals/data to train the ML algorithms in use. Second, the huge volume of signals/data are often collected by sensors at geographically distributed locations. Third, after proper training, the trained models may operate in an environment different from that where the training signals/data is collected. This talk will present exemplary techniques to address these challenges and briefly discuss open issues for future work.

First, we consider a classification problem of electromagnetic (EM) signals to illustrate a technique to overcome lack of training data. Specifically, a system using Generative Adversarial Network (GAN) will be presented that can detect and classify EM signals as friendly or hostile, even when there is no prior data of the hostile signals. The proposed approach is validated by use of a simulated waveform dataset. Second, to support defence applications, federated learning can be used to learn the model parameters from signals/data collected at distributed nodes, without data sharing with any other node, and adapts according to the limited availability of resources. Using real datasets, the experimentation results show that the proposed approach performs near to the optimum with various ML models. Third, by using a network for defence analytic processing, we highlight the potential advantage of transfer learning for speeding up reinforcement learning when the operating environment has significant changes.

Dr Alex Serb, University of Edinburgh

Dr. Serb is a reader in Unconventional AI Hardware Technologies at the University of Edinburgh. He is a MIET and SMIEEE. He has led a grant portfolio of ~£2M worth of projects, including a DSTL contract for studying the feasibility of adiabatic capacitive neural networks and the UKRI New Investigator Award "ANAGRAM". He has further been a co-investigator in UKRI MINDS CDT. He has supervised 20+ PhD students and 6× postdocs. His research interests span across circuit and system design and AI, with particular focus on emerging technologies such as memimpedance elements. He has published 50+ journal articles and 50+ conference papers, and filed 5× patents. Furthermore, he is co-founder and CTO of ArC instruments Ltd., a company that manufactures instrumentation for memimpedance device characterisation and testing (<https://www.arc-instruments.co.uk/>).

**Abstract: Adiabatic computing for low power image sensing**

The world has an insatiable appetite for data, which leads to vast demand for data processing. Nowhere do the constraints of this model -especially power constraints- become more evident, than in the domain of image processing. We at the univ. of Edinburgh are engineering a hardware accelerator aimed at making neural network-based image processing more power efficient. Crucially, the accelerator uses “adiabatic techniques” -which will be explained in the talk- in order to reduce the power consumed by the neural network it embodies to below the CV^2 limit typically associated with digital circuits, and which forms a hard limit for non-adiabatic systems. This is significant, as 90%+ power dissipation savings can be made according to data gathered so far, allowing designs implemented in e.g. 180nm CMOS technology to be theoretically competitive with designs in 65nm CMOS (which also has strategic implications given the current state of semiconductor markets). In this talk we will cover the basic experiments and results carried out so far in this nascent field and discuss the perspectives for the future development of the technology.

SSPD2023 Conference Committee

General Chairs

Mike Davies - University of Edinburgh

Stephen McLaughlin - Heriot-Watt University

Jordi Barr - Dstl

Gary Heald - Dstl

Publicity and Local Arrangements Chair

Janet Forbes - University of Edinburgh

Technical Programme Committee

Abderrahim Halimi - Heriot-Watt University

Alasdair Hunter - Dstl

Andreas Ahrens - Hochschule Wismar

Andrew Wallace - Heriot-Watt University

Athanasios Gkelias - Imperial College London

Augusto Aubry - Universita degli studi di Napoli

Bernard Mulgrew - University of Edinburgh

Brian Barber - Dstl

Bruno Clerckx - Imperial College London

Carmine Clemente - University of Strathclyde

Christoph Wasserzier - Fraunhofer Institute for High Frequency Physics and Radar Techniques FHR

Christos Ilioudis - University of Strathclyde

Cristian Rusu - University of Edinburgh

Dave Harvey - Thales

David Blacknell - Dstl

David Cormack - Leonardo

David Garren - Naval Postgraduate School

David Greig - Leonardo

Domenico Gaglione - Centre for Maritime Research and Experimentation (CMRE)

Duncan Williams - Dstl

Emma Goldstein - Dstl

Geert Leus - Delft University of Technology

Henry Gouk - University of Edinburgh

Ian Proudler - University of Strathclyde

Ivo Bukovsky - University of South Bohemia, Ceske Budejovice

James Hopgood - University of Edinburgh

Jason Ralph - University of Liverpool

Joao Mota - Heriot-Watt University

John Thompson - University of Edinburgh

Julian Deeks - Dstl

Ken McEwan - Dstl

Krishnaprasad Nambur Ramamohan - Delft University of Technology

Lyudmila Mihaylova - University of Sheffield

Mahesh Banavar - Clarkson University

Maria Greco - University of Pisa

Mark Hadley - Kaon Limited

Mathini Sellathurai - Heriot-Watt University

Mehrdad Yaghoobi - University of Edinburgh

Murat Uney - University of Liverpool

Nikolaos Dionelis - University of Edinburgh

Oliver Sims - Leonardo

Paul Thomas - Dstl

Richard Jackson - Dstl

Sami Aldalahmeh - Al-Zaytoonah University of Jordan

Sen Wang - Heriot-Watt University

Simon Godsill - University of Cambridge

Simon Maskell - University of Liverpool

Stephan Weiss - University of Strathclyde

Stephen Ablett - Dstl

Suresh Jacob - Dstl

Vladimir Stankovic - University of Strathclyde

Wenwu Wang - University of Surrey

Wolfgang Koch - Fraunhofer FKIE

Yoann Altmann - Heriot-Watt University

Adaptive Kernel Kalman Filter for Magnetic Anomaly Detection-based Metallic Target Tracking

Mengwei Sun, Richard Hodgskin-Brown, Mike E. Davies, Ian K. Proudler, James R. Hopgood

Abstract—This paper proposes the use of the adaptive kernel Kalman filter (AKKF) to track metallic targets using magnetic anomaly detection (MAD). The proposed AKKF-based approach enables accurate tracking of moving metallic targets using magnetometer sensors, even in the presence of dynamic and unknown magnetic moments. The experimental results demonstrate that the proposed method exhibits favourable tracking and estimation performance with reduced computational complexity compared with the bootstrap particle filter (PF). For example, in magnetic moment strength estimation, the relative root mean square error (RRMSE) of the proposed algorithm using 50 particles can approach 2.5% with a computation time of 0.18 seconds, whereas the RRMSE of the PF using 2000 particles is 4.5% with a computation time of 1.4 seconds. This study highlights the potential of AKKF in MAD for metallic target tracking using magnetometer sensors.

Index Terms—Adaptive kernel Kalman filter, magnetic anomaly detection, metallic target tracking

I. INTRODUCTION

Detecting and tracking targets are critical in automated surveillance and security systems that aim to keep up with evolving safety and security risks. In recent years, magnetic anomaly detection (MAD) has been widely studied for various applications in military and civilian contexts [1], such as airborne maritime surveillance [2], shipwrecks [3], access control [4], and tracking of moving metallic vehicles [5], [6]. The magnetic field is an intrinsic characteristic of many objects. The ability to detect and track magnetic fields provides a non-invasive and contactless method for monitoring and analysing these objects. Tracking techniques based on MAD typically utilise magnetic sensors, such as magnetometers [5], [6], to detect and measure the magnetic field generated by the objects. The position and orientation of the target can then be estimated based on the measured magnetic field [2], [5], [6]. Unlike other tracking technologies, such as optical

M. W. Sun, M. E. Davies and J. R. Hopgood are with the Institute of Digital Communications, University of Edinburgh, Edinburgh, EH9 3FG, U.K. E-mail: (msun; mike.davies; james.hopgood)@ed.ac.uk.

I. Proudler is with the Centre for Signal & Image Processing (CeSIP), Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, U.K. E-mail: ian.proudler@strath.ac.uk.

Richard Hodgskin-Brown is with the Electromagnetic Sensing Group, Department of Electronics & Electrical Engineering, University of Manchester, Manchester, M13 9PL, UK. E-mail: richard.hodgskin-brown@postgrad.manchester.ac.uk

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing; EPSRC Impact Acceleration Accounts (IAA) Project number PV022.

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

or ultrasonic methods, magnetic tracking is emerging as an occlusion-free tracking scheme for estimating the position and orientation of the target [7].

The tracking problem can be formulated under the Bayesian framework by understanding the relationship between the magnetic moment of the target and its kinematic parameters. In [5], [6], magnetometer sensor models for tracking metallic point targets and extended targets are proposed and validated. The suitability of magnetometer sensors for tracking is analysed regarding local observability and the Cramér–Rao lower bound (CRLB). The extended Kalman filter (EKF) and the weighted least squares algorithm, by minimising the cost function, are used for estimating the kinematic parameters and magnetic moment, respectively. However, the time-varying magnetic dipole moment, which arises due to the moving vehicle’s heading, is ignored and set to be constant. In [2], the authors investigate the use of various nonlinear filters for kinematic and magnetic dipole tracking applications and compare their performances. The nonlinear filters that are compared include the EKF, unscented Kalman filter (UKF), generic particle filter (GPF), auxiliary particle filter (APF), a combination of EKF and GPF and a combination of UKF and GPF. Sithiravel et al. [2] also include the derivation of the posterior CRLB to quantify the possible best estimation accuracy for MAD.

The proposed sensor model in [2], [5], [6] results in a sequential Bayesian estimation problem that is both highly nonlinear and high-dimensional. Choosing a Bayesian filter involves balancing between accuracy and computation complexity. While the EKF is computationally efficient, its accuracy may suffer when the system’s nonlinearity is high. In contrast, the UKF and particle filter (PF) can provide better accuracy for highly nonlinear problems. However, the computational cost of the UKF can increase for high-dimensional systems, while the PF can suffer from the curse of dimensionality. Recently, the adaptive kernel Kalman filter (AKKF) has been proposed [8]–[11], which demonstrates significant improvement in estimation performance compared to other nonlinear Kalman filters (KFs) while reducing computation complexity and avoiding resampling, as is often required with most PFs in tracking systems. This paper investigates the potential of using the AKKF within MAD-based vehicle tracking with the following contributions:

- Exploring a new application for the AKKF. While previous work focused on utilising the AKKF for object tracking problems, this paper uses the AKKF for joint tracking and magnetic parameters estimation, which are high-dimensional and high nonlinear problems.

- The simulations evaluate the performance of the AKKF in tracking and estimating magnetic parameters. The results demonstrate improved computation efficiency in vehicle tracking and magnetic parameters estimation. For example, compared with the PF, the relative root mean square error (RRMSE) of the magnetic moment strength estimation achieved by the AKKF can be improved from 7% to 2% when using 100 particles.

The paper is structured as follows: Section II describes the system model, Section III presents the AKKF-based algorithm, Section IV provides the simulation results, and Section V draws the conclusions.

II. SYSTEM MODEL

The system for MAD-based vehicle tracking is shown in Fig. 1, where two magnetometer sensors are positioned close to a straight road with vector coordinates denoted as \mathbf{s}_1 and \mathbf{s}_2 , respectively. The vehicle is moving to pass the stationary magnetometer sensors. The vehicle is approximated as a point magnetic dipole. The dynamic state-space model (DSSM) comprises a motion model that describes the target's position and velocity over time and a measurement model that relates the target's magnetic field to the measurements obtained by the sensors. The target evolves as

$$\mathbf{x}_n = F\mathbf{x}_{n-1} + \mathbf{u}_n = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{n-1} + \mathbf{u}_n. \quad (1)$$

Here, ΔT is the sampling interval and is set as $\Delta T = 1$, \mathbf{u}_n is the process noise vector, the time index n is defined as $n = 1, \dots, N$, where N represents the number of time steps. The hidden states are $\mathbf{x}_n = [\xi_n, \dot{\xi}_n, \eta_n, \dot{\eta}_n, \zeta_n]^T$, where (ξ_n, η_n, ζ_n) represents the target dipole position in X-axis, Y-axis and Z-axis, and $(\dot{\xi}_n, \dot{\eta}_n)$ represent the corresponding velocity in X-axis and Y-axis. We only consider the vehicle's motion in the X-Y 2D plane and ignore the velocity in the Z-axis, as the vehicle is constrained to move on a flat surface and cannot move up or down.

The measurement at the k -th magnetometer sensor is based on a nonlinear model that can be described as follows [6]

$$\begin{aligned} \mathbf{y}_{n,k} &= h_k(\mathbf{x}_n, \mathbf{m}_n) + \mathbf{e}_{n,k} \\ &= \mathbf{B}_0 + \frac{\mu_0}{4\pi} \frac{3(\mathbf{r}_{n,k} \cdot \mathbf{m}_n)\mathbf{r}_{n,k} - \|\mathbf{r}_{n,k}\|^2 \mathbf{m}_n}{\|\mathbf{r}_{n,k}\|^5} + \mathbf{e}_{n,k}. \end{aligned} \quad (2)$$

Here, the constant \mathbf{B}_0 is the Earth's magnetic field, $\mathbf{r}_{n,k} = [\xi_n, \eta_n, \zeta_n]^T - \mathbf{s}_k$ is the target position relative to the k -th sensor at time n , and \cdot denotes the dot product. The magnetic dipole moment of the target is \mathbf{m}_n , and the additive white Gaussian noise (AWGN) associated with the measurement is $\mathbf{e}_{n,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. The magnetic field of the metallic objects, as shown in Fig. 1, is induced partly due to the ferromagnetic content (hard iron) and partly due to the deflection of the

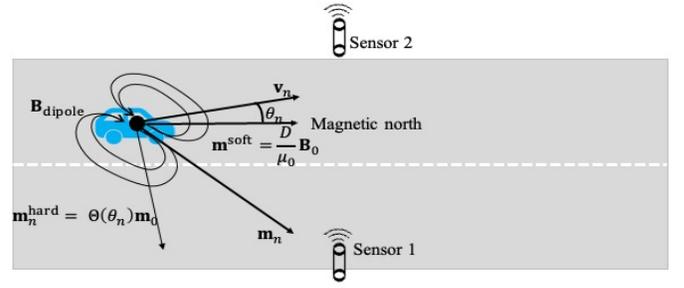


Fig. 1: System setup: Two magnetometer sensors, the vehicle is moving to pass them.

Earth's magnetic field (soft iron). Hence, the magnetic moment of the metallic objects is modelled as [6]:

$$\mathbf{m}_n = \mathbf{m}_n^{\text{hard}} + \mathbf{m}_n^{\text{soft}} = \Theta(\theta_n)\mathbf{m}_0 + \frac{D}{\mu_0}\mathbf{B}_0, \quad (3)$$

The rotation matrix $\Theta(\theta_n)$ is used to model the effect of the heading on the magnetic field refers to the magnetic north, and it can be expressed as [6]:

$$\Theta(\theta_n) = \begin{bmatrix} \cos \theta_n & -\sin \theta_n & 0 \\ \sin \theta_n & \cos \theta_n & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

The magnetic dipole moment of the target, denoted by \mathbf{m}_0 , is assumed to be independent of the external magnetic field. The scalar constant D accounts for the magnetic field induced by the target's ferromagnetic content and deflection of the Earth's magnetic field. The permeability of the vacuum, represented by μ_0 , is a fundamental physical constant that describes the magnetic properties of free space.

III. AKKF-BASED TRACKING AND ESTIMATION ALGORITHMS

The purpose of metallic target tracking is to precisely track the target's movement and simultaneously estimate its magnetic moment. This is accomplished through the utilisation of a posterior probability density function (pdf), which explains the joint distribution of the target's hidden states $\mathbf{X}_n = [\mathbf{x}_n^T, \mathbf{m}_n^T, \mathbf{m}_0, D]^T$, considering the observations $\mathbf{y}_{1:n,1:2}$ at two sensors which are located at $\mathbf{s}_{1:2}$. The joint posterior pdf is decomposed in Equation (5). In this section, we will discuss how to use the PF and the AKKF to sequentially approximate the joint posterior pdf.

A. PF-based algorithm

The PF approximates the joint posterior pdf by using a weighted set of particles. Each particle represents a possible value of the joint state variables \mathbf{X}_n at each time step $n = 1, 2, \dots, N$. The joint posterior distribution in (5) can be estimated as follows:

$$\begin{aligned} p(\mathbf{X}_n | \mathbf{y}_{1:n,1:2}) \\ \approx \frac{1}{M} \sum_{i=1}^M w_n^{[i]} \delta(\mathbf{x}_n - \mathbf{x}_n^{[i]}, \mathbf{m}_n - \mathbf{m}_n^{[i]}, \mathbf{m}_0 - \mathbf{m}_{0,n}^{[i]}, D - D_n^{[i]}). \end{aligned} \quad (6)$$

$$p(\mathbf{X}_n | \mathbf{y}_{1:n,1:2}) = p(\mathbf{x}_n, \mathbf{m}_n, \mathbf{m}_0, D | \mathbf{y}_{1:n,1:2}) = p(\mathbf{y}_{1:n,1:2} | \mathbf{x}_n, \mathbf{m}_n, \mathbf{m}_0, D) \times \frac{\iiint p(\mathbf{x}_n | \mathbf{x}_{n-1}) p(\mathbf{m}_n | \mathbf{x}_n, \mathbf{m}_{n-1}, \mathbf{m}_0, D) p(\mathbf{m}_0, D) p(\mathbf{x}_{n-1}, \mathbf{m}_{n-1}, \mathbf{m}_0, D | \mathbf{y}_{1:n-1,1:2}) d\mathbf{x}_{n-1} d\mathbf{m}_{n-1} d\mathbf{m}_0 dD}{p(\mathbf{y}_{n,1:2} | \mathbf{y}_{1:n-1,1:2})} \quad (5)$$

Here, $w_n^{(i)}$ represents the weight of the i -th particle at time step n , δ denotes the Dirac delta function, and M is the number of particles. At each time step n , the weight $w_n^{(i)}$ is updated based on the likelihood of the observation $\mathbf{y}_{1:n,1:2}$ given the particle's state variables $\{\mathbf{x}_n^{(i)}, \mathbf{m}_n^{(i)}, \mathbf{m}_0^{(i)}, D^{(i)}\}$, i.e., $w_n^{(i)} = w_{n-1}^{(i)} p(\mathbf{y}_{1:n,1:2} | \mathbf{x}_n^{(i)}, \mathbf{m}_n^{(i)}, \mathbf{m}_0^{(i)}, D^{(i)})$. The state variables of each particle are updated using the transition probabilities as (7), where $\theta_n^{(i)} = \arctan 2(\dot{\eta}_n^{(i)}, \dot{\xi}_n^{(i)})$, and $\mathbf{u}_n^{(i)}$ represents a process noise sample drawn from the process noise distribution.

$$\mathbf{x}_n^{(i)} = F\mathbf{x}_{n-1}^{(i)} + \mathbf{u}_n^{(i)} \quad (7a)$$

$$\mathbf{m}_n^{(i)} = \Theta(\theta_n^{(i)})\mathbf{m}_{0,n}^{(i)} + \frac{D_n^{(i)}}{\mu_0}\mathbf{B}_0 \quad (7b)$$

$$\mathbf{m}_{0,n}^{(i)} = \mathbf{m}_{0,n-1}^{(i)} \quad (7c)$$

$$D_n^{(i)} = D_{n-1}^{(i)}. \quad (7d)$$

After updating the particles and their weights, the particles are resampled to obtain a new set of particles for the next time step. The resampling process involves randomly selecting particles from the current set with probability proportional to their weights, with replacement.

However, the computational cost of the PF grows exponentially with the number of state variables, making it impractical for high-dimensional problems. In high-dimensional problems, it is difficult to obtain a sufficient number of particles to represent the posterior pdf accurately, leading to particle degeneracy, where only a small subset of particles have non-zero weights, and the rest are effectively ignored. This can result in poor estimation accuracy and instability in the estimates. To address this issue, we investigate the use of the AKKF to solve high-dimensional problems with low computational costs and favourable accuracy.

B. AKKF-based algorithm

The proposed AKKF [8] enables us to obtain the empirical kernel mean embedding (KME) of the posterior pdf of the hidden state in (5). This is accomplished using a set of feature mappings of generated particles and their corresponding kernel weights. The particles are updated and propagated in the data space based on the parametric DSSMs, and the corresponding kernel weights are predicted and updated linearly. Common kernel functions used for KMEs include linear, quadratic, quartic, and Gaussian kernels. The quartic kernel can be used when the data is highly nonlinear and complex. Considering the system setup and the DSSM in equations (1) and (2), the nonlinearity of the measurement model is highly nonlinear. Therefore, we apply the quartic kernel to approximate the predictive and posterior pdfs in this paper. The quartic kernel

Algorithm 1 AKKF-based metallic target tracking algorithm

Require: DSSM: motion model and measurement model.

- 1: **Initialisation:** Set the initial particles in the data space $\tilde{\mathbf{x}}_0^{(i=1:M)} \sim P_{\text{init}}$, $\mathbf{w}_0 = 1/M [1, \dots, 1]^T$.
 - 2: **for** $n = 1 : N$ **do**
 - 3: **Prediction:**
 - In the data space, propagate proposal particles following (7),
 - \Rightarrow In the kernel feature space with basis Φ_n : $\mathbf{w}_n^- = \Gamma_n \mathbf{w}_{n-1}^+$, $S_n^- = \Gamma_n S_{n-1}^+ \Gamma_n^T + V_n$.
 - 4: **Update:**
 - In the data space: $\mathbf{y}_n^{(i)} = h(\mathbf{X}_n^{(i)}, \mathbf{e}_n^{(i)})$,
 - \Rightarrow In the kernel feature space with basis Φ_n : $\mathbf{w}_n^+ = \mathbf{w}_n^- + Q_n (G_{:,y_n} - G_{yy} \mathbf{w}_n^-)$, $S_n^+ = S_n^- - Q_n G_{yy} S_n^-$.
 - 5: **Proposal particles draw:**
 - In the data space: $\tilde{\mathbf{X}}_n^{(i=1:M)} \sim \mathcal{N}(\mathbb{E}(\mathbf{X}_n), \text{Cov}(\mathbf{X}_n))$,
 - \Rightarrow Get the kernel feature space with basis Ψ_n .
 - 6: **end for**
-

function $k(\mathbf{X}, \mathbf{Y})$ and its corresponding feature mapping $\phi_{\mathbf{X}}(\mathbf{X})$ are defined as:

$$k(\mathbf{X}, \mathbf{Y}) = (\mathbf{X}^T \mathbf{Y} + c)^4 \quad (8a)$$

$$\phi_{\mathbf{X}}(\mathbf{X}) = [a_1, \dots, a_j, \dots, a_d]^T, \quad (8b)$$

where $c \geq 0$ is a free parameter that trades off the influence of higher-order versus lower-order terms in the polynomial, and the element in the quartic kernel feature mapping is

$$a_j = \frac{\sqrt{4!}}{\sqrt{\varrho_1! \dots \varrho_K! \varrho_{K+1}!}} x_1^{\varrho_1} \dots x_K^{\varrho_K} \sqrt{c^{\varrho_{K+1}}}, \quad \varrho_1 + \dots + \varrho_{K+1} = 4.$$

Here, $\varrho_1, \dots, \varrho_{K+1}$ are non-negative integers representing the powers of the corresponding input dimensions. The dimension of $\phi_{\mathbf{X}}(\mathbf{X})$ is $d = (K+4)!/(4!K!)$, where K is the dimension of the hidden state $\mathbf{X} = [x_1, \dots, x_k, \dots, x_K]^T$.

The proposed AKKF-based algorithm is realised sequentially by embedding the pdf $p(\mathbf{X}_n | \mathbf{y}_{1:n,1:2})$ into an reproducing kernel Hilbert space (RKHS) as an empirical KME,

$$p(\mathbf{X}_n | \mathbf{y}_{1:n,1:2}) \rightarrow \hat{\mu}_{\mathbf{X}_n}^+ = \Phi_n \mathbf{w}_n^+, \quad (9)$$

where Φ_n represents the kernel feature mappings of particles and \mathbf{w}_n^+ is the updated kernel weight. The AKKF-based algorithm consists of three main steps, which we will further explain in the following subsections. The algorithm is summarised in Algorithm I. See [8] for details of the AKKF.

1) *Draw Proposal Particles at Time $n - 1$* : The posterior distribution pdf at time $n - 1$, i.e., $p(\mathbf{X}_{n-1} | \mathbf{y}_{1:n-1,1:2})$ is

empirically as approximated by an element $\hat{\mu}_{\mathbf{x}_{n-1}}^+$ in the RKHS based on the AKKF, resulting in $p(\mathbf{X}_{n-1} | \mathbf{y}_{1:n-1,1:2}) \rightarrow \hat{\mu}_{\mathbf{x}_{n-1}}^+ = \Phi_{n-1} \mathbf{w}_{n-1}^+$. Here, $\Phi_{n-1} = [\phi_{\mathbf{x}}(\mathbf{X}_{n-1}^{(1)}), \dots, \phi_{\mathbf{x}}(\mathbf{X}_{n-1}^{(M)})]$ represents the kernel feature mappings of the particles $\mathbf{X}_{n-1}^{(1:M)}$ using the quartic kernel function, and \mathbf{w}_{n-1}^+ is the weight vector with a positive definite weight covariance matrix denoted as S_{n-1}^+ . Then, $\mathbb{E}(\mathbf{X}_{n-1})$ and $\text{Cov}(\mathbf{X}_{n-1})$ from $\hat{\mu}_{\mathbf{x}_{n-1}}^+$ are extracted and passed to the data space following [8]. Next, proposal particles are generated according to the importance of distribution as $\tilde{\mathbf{X}}_{n-1}^{(i=1:M)} \sim \mathcal{N}(\mathbb{E}(\mathbf{X}_{n-1}), \text{Cov}(\mathbf{X}_{n-1}))$, and mapped to the RKHS as $\Psi_{n-1} = [\phi_{\mathbf{x}}(\tilde{\mathbf{X}}_{n-1}^{(1)}), \dots, \phi_{\mathbf{x}}(\tilde{\mathbf{X}}_{n-1}^{(M)})]$.

2) *Prediction from Time $n-1$ to Time n* : The empirical KME of the predictive probability at time n is approximated using a linear conditional operator in the RKHS:

$$\begin{aligned} p(\mathbf{X}_n | \mathbf{y}_{1:n-1,1:2}) &\mapsto \hat{\mu}_{\mathbf{x}_n}^- = \hat{C}_{\mathbf{X}_n | \tilde{\mathbf{X}}_{n-1}} \hat{\mu}_{\tilde{\mathbf{X}}_{n-1}}^+ \\ &= \Phi_n \underbrace{(K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} + \lambda_{\tilde{K}} I)^{-1} K_{\tilde{\mathbf{x}}\mathbf{x}}}_{\Gamma_n} \mathbf{w}_{n-1}^+ = \Phi_n \mathbf{w}_n^-. \end{aligned} \quad (10)$$

Here, $\Phi_n = [\phi_{\mathbf{x}}(\mathbf{X}_n^{(1)}), \dots, \phi_{\mathbf{x}}(\mathbf{X}_n^{(M)})]$ represent the feature mappings of the state particles at time n , which are obtained by propagating $\tilde{\mathbf{X}}_{n-1}^{(i=1:M)}$ through the process function following (7). The Gram matrices $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} = \Psi_{n-1}^T \Psi_{n-1}$ and $K_{\tilde{\mathbf{x}}\mathbf{x}} = \Psi_{n-1}^T \Phi_{n-1}$. And Γ_{n-1} represents the change of sample representation from Φ_{n-1} to Ψ_{n-1} . The regularisation parameter, $\lambda_{\tilde{K}}$, ensures the inverse is well-defined, and I is the identity operator matrix. Following the derivation in [8], the kernel weight covariance matrix, S_n^- , is calculated as $S_n^- = \Gamma_n S_{n-1}^+ \Gamma_n^T + V_n$, where V_n is the finite matrix representation of the transition residual matrix [8].

3) *Update at Time n* : The observation particles are updated based on the observation models in (2). The kernel mappings of observation particles in the kernel feature space are $\Upsilon_n = [\phi_{\mathbf{y}}(\mathbf{y}_{n,1:2}^{(1)}), \dots, \phi_{\mathbf{y}}(\mathbf{y}_{n,1:2}^{(M)})]$. Based on the derivations in [8], the KME vector, the weight vector, and the kernel weight covariance matrix are updated as shown in Equations (11a) to (11c), respectively.

$$\hat{\mu}_{\mathbf{x}_n}^+ = \hat{\mu}_{\mathbf{x}_n}^- + Q_n [\phi_{\mathbf{y}}(\mathbf{y}_n) - \hat{C}_{\mathbf{y}_n | \mathbf{x}_n} \hat{\mu}_{\mathbf{x}_n}^-] = \Phi_n \mathbf{w}_n^+, \quad (11a)$$

$$\mathbf{w}_n^+ = \mathbf{w}_n^- + Q_n (G_{\cdot:\mathbf{y}_n} - G_{\mathbf{y}\mathbf{y}} \mathbf{w}_n^-) \quad (11b)$$

$$S_n^+ = S_n^- - Q_n G_{\mathbf{y}\mathbf{y}} S_n^-. \quad (11c)$$

Here, Q_n is the kernel Kalman gain, $G_{\cdot:\mathbf{y}_n} = \Upsilon_n^T \phi_{\mathbf{y}}(\mathbf{y}_n)$, and the Gram matrix of the observation at time n is $G_{\mathbf{y}\mathbf{y}} = \Upsilon_n^T \Upsilon_n$ [8].

IV. SIMULATION RESULTS

The simulation parameters are set as follows: the initial state of the vehicle is set to $\mathbf{x}_1 = [-7.56, 3.75, 6.75, 0.4]^T$, and the hard iron dipole moment of the vehicle is $\mathbf{m}_0 = [-203, 124, 267]^T \text{Am}^2$ [6]. The soft iron scalar is $D = 1 \text{m}^3$ [6]. The sensors' axes are $\mathbf{s}_1 = [0, 0, 0.3]^T$ and $\mathbf{s}_2 = [0, 9, 0.7]^T$, and the measurement noise covariance matrices are [6]

$$R_1 = 10^{-15} \begin{bmatrix} 0.1303 & -0.0073 & -0.0114 \\ -0.0073 & 0.1112 & 0.0117 \\ -0.0114 & 0.0117 & 0.1558 \end{bmatrix}$$

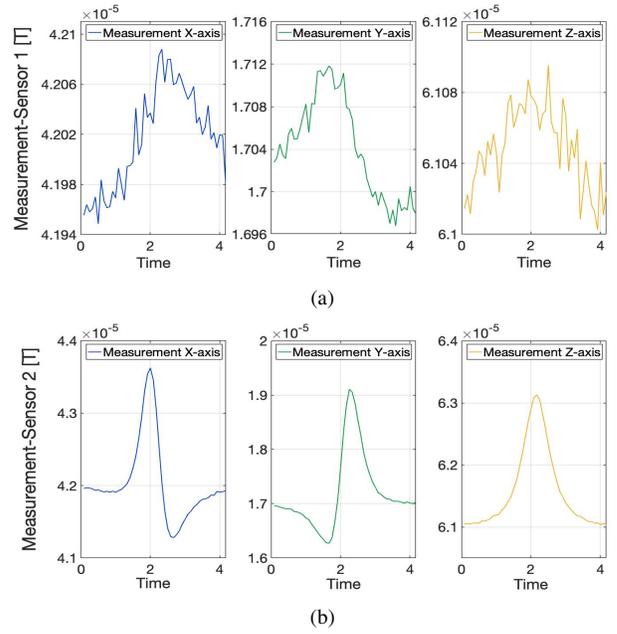


Fig. 2: Measured magnetic field strength in X/Y/Z axes at two sensors. (a) Sensor 1; (b) Sensor 2.

$$R_2 = 10^{-15} \begin{bmatrix} 0.1500 & 0.0205 & 0.0215 \\ 0.0205 & 0.1937 & 0.0310 \\ 0.0215 & 0.0310 & 0.1483 \end{bmatrix}.$$

Here, the unit of measurement is Tesla. The magnetic field strength measured in X-axis, Y-axis, and Z-axis at two sensors is shown in Fig. 2. The initial prior distribution of the hidden states for particles is drawn following the settings as $\xi_0^{(i=1:M)} \sim \mathcal{U}(-7.6, -7.4)$, $\eta_0^{(i=1:M)} \sim \mathcal{U}(6, 8)$, $\xi_0^{(i=1:M)} \sim \mathcal{N}(\xi_0, 10^{-2})$, $\eta_0^{(i=1:M)} \sim \mathcal{N}(\eta_0, 10^{-2})$, $z_0^{(i=1:M)} \sim \mathcal{N}(z_0, 10^{-1})$, $D_0^{(i=1:M)} \sim \mathcal{N}(D_0, 10^{-2})$, $\mathbf{m}_0^{(i=1:M)} \sim \mathcal{N}(\mathbf{m}_0, 10^3 I)$.

Fig. 3 displays a representative trajectory and the tracking performance obtained by the AKKF and the PF. Fig. 4 and Fig. 5 display the estimation performance of the hard iron dipole moment \mathbf{m}_0 and the soft iron scalar D , obtained from these two filters. The AKKF uses $M^{\text{AKKF}} = 100$ particles, while $M^{\text{PF}} = 2000$ particles are used for the PF. From Fig. 3 to Fig. 5, we can see that the AKKF with a smaller number of particles achieved favourable tracking and estimation performance compared to the PF with a large number of particles. We then compare the average root mean square error (RMSE) of the AKKF and the PF using the same number of particles, along with its standard deviation for tracking performance. RMSE is defined in (12). We obtain 100 Monte Carlo (MC) realisations with an increasing number of particles, specifically $M = [50, 100, 200]$, while the bootstrap PF with 2000 particles is considered as the benchmark performance, as shown in 6(a).

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^N (\xi_n - \hat{\xi}_n)^2 + (\eta_n - \hat{\eta}_n)^2}{N}}. \quad (12)$$

We also compared the RRMSE and its standard deviation

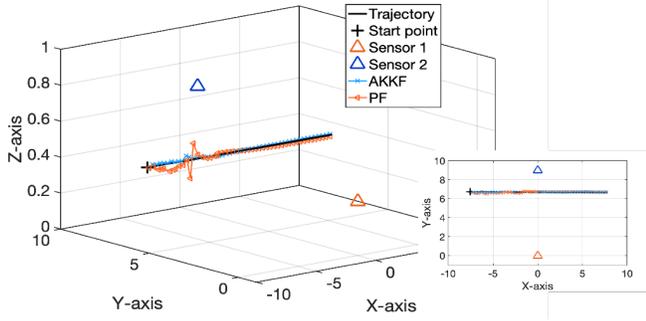


Fig. 3: Ground truth trajectory versus tracking performance achieved by the AKKF and the PF.

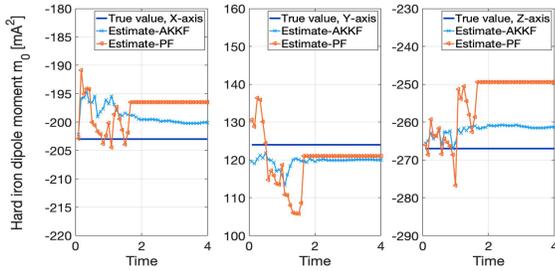


Fig. 4: True hard iron dipole moment \mathbf{m}_0 versus estimated values.

for the estimation performance of \mathbf{m}_0 and D , as well as the computation time, as shown in Figures 6(b) to 6(d), respectively.

Based on the simulation results, we draw the following conclusions: the proposed AKKF demonstrates significantly improved performance with the same number of particles compared to the PF, especially for trajectory tracking and magnetic moment strength estimation. For example, with 200 particles, the tracking accuracy can be improved by 0.13m, and magnetic moment strength estimation accuracy can be improved by 5%. Moreover, compared with the benchmark performance achieved by the PF with 2000 particles, the AKKF shows satisfactory tracking and estimation performance with significantly reduced computational complexity when dealing with high nonlinear and high-dimensional problems. This improved performance and reduced computational complexity are due to the ability of the AKKF to efficiently represent high-dimensional data using kernels, which can capture more information about the data in the rich feature space of the kernel. The feature mappings can then be used to perform computations more efficiently. In contrast, the PF works with the data directly and may struggle to handle high-dimensional data.

V. CONCLUSIONS

This paper explores a new application for the AKKF by utilising it for joint tracking and magnetic parameters estimation in high-dimensional and high nonlinear problems. The simulations presented demonstrate improved computational efficiency in vehicle tracking and magnetic parameter estimation.

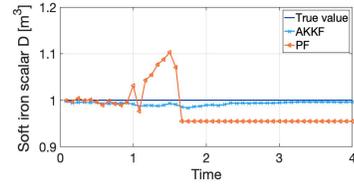


Fig. 5: True soft iron scalar D versus estimated values.

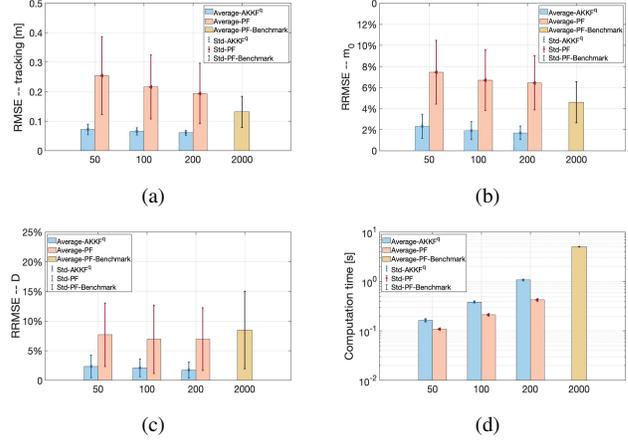


Fig. 6: Average and standard deviation of tracking RMSE. RMSE and computation performance (a) Tracking; (b) Hard iron dipole moment estimation; (c) Soft iron scalar estimation; (d) Computation time.

REFERENCES

- [1] J. Lenz and S. Edelstein, "Magnetic sensors and their applications," *IEEE Sensors Journal*, vol. 6, no. 3, pp. 631–649, 2006.
- [2] R. Sithiravel, B. Balaji, B. Nelson, M. K. McDonald, R. Tharmarasa, and T. Kirubarajan, "Airborne maritime surveillance using magnetic anomaly detection signature," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3476–3490, 2020.
- [3] P. Lin, N. Zhang, M. Chang, and L. Xu, "Research on the model and the location method of ship shaft-rate magnetic field based on rotating magnetic dipole," *IEEE Access*, vol. 8, pp. 162 999–163 005, 2020.
- [4] C. Ascì, W. Wang, and S. Sonkusale, "Security monitoring system using magnetically-activated rfid tags," in *2020 IEEE SENSORS*, 2020, pp. 1–4.
- [5] N. Wahlström, J. Callmer, and F. Gustafsson, "Magnetometers for tracking metallic targets," in *2010 13th International Conference on Information Fusion*, 2010, pp. 1–8.
- [6] N. Wahlström and F. Gustafsson, "Magnetometer modeling and validation for tracking metallic targets," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 545–556, 2014.
- [7] V. Biancalana, R. Cecchi, P. Chessa, M. Mandalà, G. Bevilacqua, Y. Dancheva, and A. Vigilante, "Validation of a fast and accurate magnetic tracker operating in the environmental field," *Instruments*, vol. 5, no. 1, p. 11, mar 2021.
- [8] M. Sun, M. E. Davies, I. K. Proudler, and J. R. Hoggood, "Adaptive kernel Kalman filter," *IEEE Trans. Signal Process.*, vol. 71, pp. 713–726, 2023.
- [9] —, "Adaptive kernel Kalman filter based belief propagation algorithm for maneuvering multi-target tracking," *IEEE Signal Process. Lett.*, vol. 29, pp. 1452–1456, 2022.
- [10] —, "Adaptive kernel Kalman filter multi-sensor fusion," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, 2021, pp. 1–8.
- [11] —, "Adaptive kernel Kalman filter," in *2021 Sensor Signal Processing for Defence Conference (SSPD)*, 2021, pp. 1–5.

Implementation of Adaptive Kernel Kalman Filter in Stone Soup

James S. Wright, James R. Hopgood, Mike E. Davies, Ian K. Proudler, Mengwei Sun

Abstract—The recently proposed adaptive kernel Kalman filter (AKKF) is an efficient method for highly nonlinear and high-dimensional tracking or estimation problems. Compared to other nonlinear Kalman filters (KFs), the AKKF has significantly improved performance, reducing computational complexity and avoiding resampling. It has been applied in various tracking scenarios, such as multi-sensor fusion and multi-target tracking. By using existing Stone Soup components, along with newly established kernel-based prediction and update modules, we demonstrate that the AKKF can work in the Stone Soup platform by being applied to a bearing-only tracking (BOT) problem. We hope that the AKKF will enable more applications for tracking and estimation problems, and the development of a whole class of derived algorithms in sensor fusion systems.

Index Terms—Adaptive kernel Kalman filter, Tracking, Stone Soup

I. INTRODUCTION

Target tracking in sensor networks is a fundamental problem that arises in a variety of applications, including surveillance, environmental monitoring, and military operations. The objective of target tracking is to estimate the location, velocity, and other relevant parameters of a target based on noisy and incomplete measurements obtained from one or more sensors. Nonlinear and non-Gaussian system models and measurement noise pose challenges that have been addressed using Bayesian filters, such as the extended Kalman filter (EKF) [1], unscented Kalman filter (UKF) [2], and particle filter (PF) [3]. The adaptive kernel Kalman filter (AKKF) has been proposed [4], [5] and demonstrated to achieve better performance, reduced computational complexity, and avoidance of resampling, especially in high nonlinear and high-dimensional problems. It has been applied to various tracking scenarios, such as multi-target

tracking [6], multi-sensor fusion [7], and magnetic anomaly detection-based metallic target tracking [8].

Stone Soup [9], [10] is a software project that provides a framework for developing and testing algorithms for the target tracking and state estimation community. The project prioritises flexibility over optimisation to aid in selecting components and algorithms for real-world problems. Stone Soup has a number of components used to both build algorithms and enable an environment for testing and assessment. For example, the Kalman filter (KF), EKF, UKF, and PF have been implemented in the Stone Soup framework [11].

In this work, our goal is to demonstrate the efficacy of the AKKF, by using the flexibility and metrics provided by Stone Soup while showing how augmentation of the framework can be undertaken. Specifically, we aim to realise adaptive updates of the empirical kernel mean embeddings (KMEs) for posterior probability density functions (pdfs) using the AKKF, which is executed in the state space, measurement space, and reproducing kernel Hilbert space (RKHS). In the state space, we generate the proposal state particles and propagate them through the motion model to get the state particles. In the measurement space, the measurement particles are achieved by propagating the state particles through the measurement model. We map all these particles into RKHSs as feature mappings and linearly predict and update the corresponding kernel weight mean vector and covariance matrix to approximate the empirical KMEs of the posterior pdfs in the RKHS.

Our contributions include the first attempt to implement the AKKF in Stone Soup, particularly in their simplest reference forms. We designed the key new components required in Stone Soup for the AKKF to run, including defining different component types, such as the `KernelParticleState`, `Kernel`, `AdaptiveKernelKalmanPredictor` and `AdaptiveKernelKalmanUpdater`. Stone Soup is designed for easy inheritance, which provides the design choice of the `Kernel` class to enable the use of different kernels and will permit the AKKF to be used for a wide variety of dynamic and measurement models, as well as future extensions for joint tracking and parameter estimation problems.

The paper is organised as follows: Section II describes the AKKF's general operation mathematically. Section III provides additional analysis and explanation of the AKKF implementation in Stone Soup. Section IV presents simulation results for a bearing-only tracking (BOT) problem, highlighting the improved performance of AKKF compared to the PF with few particles. Lastly, Section V outlines future extensions to our Stone Soup contribution.

M. W. Sun, M. E. Davies and J. R. Hopgood are with Institute of Digital Communications, University of Edinburgh, Edinburgh, EH9 3FG, U.K. E-mail: (msun; mike.davies; james.hopgood)@ed.ac.uk.

J. S. Wright is with Dstl, Dstl Porton Down, Bldg 5, Rm 101, Salisbury, Wiltshire, SP4 0JQ, U.K. E-mail: jwright2@dstl.gov.uk.

I. Proudler is with the Centre for Signal & Image Processing (CeSIP), Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, U.K. E-mail: ian.proudler@strath.ac.uk.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

The contents include material subject to ©Crown copyright (2023), Dstl. This material is licensed under the terms of the Open Government Licence except where otherwise stated. To view this licence, visit <http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3> or write to the Information Policy Team, The National Archives, Kew, London TW9 4DU, or email: psi@nationalarchives.gov.uk

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

II. ADAPTIVE KERNEL KALMAN FILTER (AKKF)

The AKKF was inspired by the KME and KF and originally formulated in 2021 to address shortcomings of existing Bayesian filters for tracking problems in nonlinear systems [4], [5]. In the AKKF, the posterior pdf is approximated with particles and weights, but not in the state space as in the PF. Specifically, the pdf is embedded into an RKHS as an empirical KME,

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \rightarrow \hat{\rho}_{\mathbf{x}_k}^+ = \Phi_k \mathbf{w}_k^+, \quad (1)$$

where \mathbf{x}_k represents the hidden state at the k -th time slot, and \mathbf{y}_k is the corresponding observation. The feature mappings of particles $\mathbf{x}_k^{\{1:M_A\}}$ are represented as Φ_k , i.e., $\Phi_k = [\phi_{\mathbf{x}}(\mathbf{x}_k^{(1)}), \dots, \phi_{\mathbf{x}}(\mathbf{x}_k^{(M_A)})]$, where M_A is the number of particles, and the weight vector \mathbf{w}_k^+ includes M_A non-uniform weights. The KME $\hat{\rho}_{\mathbf{x}_k}^+$ in (1) is an element in the RKHS that captures the feature value of the distribution.

A. How to choose different kernels in the AKKF

The feature mapping $\phi_{\mathbf{x}}(\mathbf{x})$ of different kernels can be chosen depending on the specific applications, such as linear kernels, polynomial kernels, including quadratic and quartic kernels, and Gaussian kernels. The following summarises kernel functions with the assumption that $\mathbf{x} = [x_1, \dots, x_n]^T$ and $\mathbf{x}' = [x'_1, \dots, x'_n]^T$.

- Linear kernel function

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'. \quad (2)$$

The linear kernel can capture the first-order moments of a distribution, such as the mean and variance. This kernel is often used in linear regression. It can be effective when the data is well-modelled by a linear relationship.

- Quadratic kernel function

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = (\alpha \langle \mathbf{x}, \mathbf{x}' \rangle + c)^2. \quad (3)$$

Here, $c \geq 0$ is a free parameter that trades off the influence of higher-order versus lower-order terms in the polynomial. The parameter α represents the slope that scales the input vectors' dot product. These parameters allow the kernel to emphasise or de-emphasise the importance of different input features. The quadratic kernel can capture the second-order moments of a distribution, such as the covariance and correlations between pairs of variables. The quadratic kernel is appropriate when the data is nonlinear but relatively simple.

- Quartic kernel function

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = (\alpha \langle \mathbf{x}, \mathbf{x}' \rangle + c)^4. \quad (4)$$

The quartic kernel can capture higher-order moments beyond the mean and covariance, such as skewness and kurtosis. The quartic kernel can be used when the data is highly nonlinear and complex.

- Gaussian kernel function

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right). \quad (5)$$

Here, σ is a parameter that determines the width of the Gaussian kernel. The Gaussian kernel can capture the mean and covariance of the data, as well as the smoothness and correlation structure. The Gaussian kernel is appropriate when the data is highly nonlinear and complex, and the relationship between the variables is not well-defined.

B. How to implement the AKKF

The AKKF includes three modules, as shown in Fig. 1: a predictor that utilises both prior and proposal information, at time $k - 1$, to update the prior state particles and predict the kernel weight mean and covariance at time k , an updater employs the predicted values to update the kernel weight and covariance, and an updater generates the proposal state particles.

1) *Predictor takes prior and proposal*: The predictor is executed in the state space and kernel space, i.e., RKHS. Suppose that the prior and proposal state particles at time $k - 1$ are represented as $\mathbf{x}_{k-1}^{\{i=1:M_A\}}$ and $\tilde{\mathbf{x}}_{k-1}^{\{i=1:M_A\}}$, respectively. Their feature mappings in RKHSs are given by:

$$\begin{aligned} \Phi_{k-1} &= [\phi_{\mathbf{x}}(\mathbf{x}_{k-1}^{(1)}), \dots, \phi_{\mathbf{x}}(\mathbf{x}_{k-1}^{(M_A)})] \\ \Psi_{k-1} &= [\phi_{\mathbf{x}}(\tilde{\mathbf{x}}_{k-1}^{(1)}), \dots, \phi_{\mathbf{x}}(\tilde{\mathbf{x}}_{k-1}^{(M_A)})]. \end{aligned}$$

■ At time k , the prior state particles in the state space are generated by passing the proposal particles at time $k - 1$, i.e., $\tilde{\mathbf{x}}_{k-1}^{\{i=1:M_A\}}$, through the motion model as

$$\mathbf{x}_k^{(i)} = \mathbf{f}(\tilde{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_k^{(i)}), \quad (6)$$

where $i = 1 \dots M_A$, $\mathbf{u}_k^{(i)}$ represents process noise samples which are drawn from the process noise distributions.

■ In RKHS, $\mathbf{x}_k^{\{i=1:M_A\}}$ are mapped to the RKHS as

$$\Phi_k = [\phi_{\mathbf{x}}(\mathbf{x}_k^{(1)}), \dots, \phi_{\mathbf{x}}(\mathbf{x}_k^{(M_A)})].$$

Based on [4], the transition matrix Γ_k , which represents the change of sample representation, is calculated as

$$\Gamma_k = (K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} + \lambda_{\tilde{R}} I)^{-1} K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}, \quad (7)$$

where the Gram matrices are $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} = \Psi_{k-1}^T \Psi_{k-1}$ and $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} = \Psi_{k-1}^T \Phi_{k-1}$. The regularisation parameter $\lambda_{\tilde{R}}$ is used to stabilise the inverse of $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$, and I is the identity operator matrix. In practice, $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ can become ill-conditioned and challenging to invert, leading to numerical instability and poor performance. To address this, the regularisation parameter $\lambda_{\tilde{R}}$ is added to the diagonal of $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$, which makes it better conditioned and easier to invert. The value of this regularisation parameter is usually chosen by cross-validation or other optimisation methods. Then, the predictive kernel weight vector, denoted as \mathbf{w}_k^- , and covariance matrix, denoted as S_k^- , are calculated as

$$\mathbf{w}_k^- = \Gamma_k \mathbf{w}_{k-1}^+, \quad (8)$$

$$S_k^- = \Gamma_k S_{k-1}^+ \Gamma_k^T + V_k. \quad (9)$$

Here, \mathbf{w}_{k-1}^+ and S_k^- are the posterior kernel weight mean vector and covariance matrix at time $k - 1$, respectively, and V_k

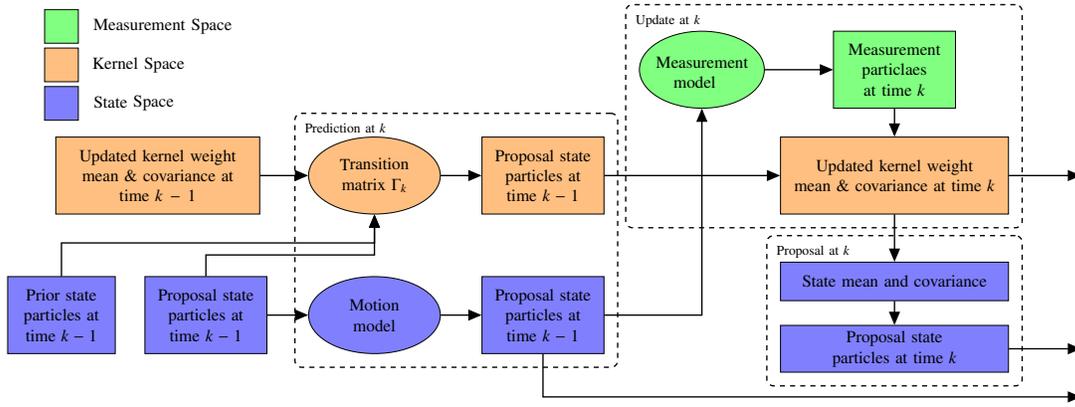


Fig. 1: Flow diagram of the AKKF.

represents the finite matrix representation of the transition residual matrix [4].

2) *Updater uses prediction*: The updater is executed in the measurement space and RKHS.

■ In the measurement space, the measurement particles are generated according to the measurement model as

$$\mathbf{y}_k^{[i]} = \mathbf{h}(\mathbf{x}_k^{[i]}, \mathbf{v}_k^{[i]}) \quad (10)$$

where $\mathbf{v}_k^{[i]}$ represent measurement noise samples which are drawn from the measurement noise distribution.

■ In RKHS, $\mathbf{y}_k^{\{i=1:M_A\}}$ are mapped in the RKHS as

$$\Upsilon_k = [\phi_{\mathbf{y}}(\mathbf{y}_k^{(1)}), \dots, \phi_{\mathbf{y}}(\mathbf{y}_k^{(M_A)})].$$

The posterior kernel weight vector and covariance matrix are updated as

$$\mathbf{w}_k^+ = \mathbf{w}_k^- + Q_k (\mathbf{g}_{\mathbf{y}\mathbf{y}_k} - G_{\mathbf{y}\mathbf{y}} \mathbf{w}_k^-) \quad (11)$$

$$S_k^+ = S_k^- - Q_k G_{\mathbf{y}\mathbf{y}} S_k^- \quad (12)$$

$$Q_k = S_k^- (G_{\mathbf{y}\mathbf{y}} S_k^- + \kappa I)^{-1}. \quad (13)$$

Here, $G_{\mathbf{y}\mathbf{y}} = \Upsilon_k^T \Upsilon_k$, $\mathbf{g}_{\mathbf{y}\mathbf{y}_k} = \Upsilon_k^T \phi_{\mathbf{y}}(\mathbf{y}_k)$ is the kernel vector of the measurement at time k , and κ is a regularisation parameter to ensure the inverse is well-defined. The kernel Kalman gain operator denoted as Q_k is derived by minimising the trace of the posterior covariance operator [4]. Then, the empirical KME of $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ is calculated as (1).

3) *Proposal generated in updater*: The proposal is executed in the state space.

■ The AKKF replaces $\mathbf{x}_k^{\{i=1:M_A\}}$ by new weighted proposal particles $\tilde{\mathbf{x}}_k^{\{i=1:M_A\}}$ to approximate the KME that can be exactly propagated through the non-linearity. The proposal particles are generated according to the importance distribution as

$$\tilde{\mathbf{x}}_k^{\{i=1:M_A\}} \sim \mathcal{N}(\mathbb{E}(X_k), \text{Cov}(X_k)). \quad (14)$$

In Stone Soup implementation, the state vector's mean and covariance for proposal are approximated using

$$\mathbb{E}(X_k) = X_k \mathbf{w}_k^+ \quad (15)$$

$$\text{Cov}(X_k) = X_k S_k^+ X_k^T, \quad (16)$$

where $X_k = \mathbf{x}_k^{\{i=1:M_A\}}$. We draw proposal particles from a Gaussian distribution for convenience, but other distributions with similar statistics could also be used. These particles are used to capture the key probability mass of the posterior pdf. It is not equivalent to approximating the posterior pdf with a Gaussian, but rather an adaptive change of basis within the feature space through a simple linear mapping.

III. IMPLEMENTATION IN STONE SOUP

The main goal behind the development of the Stone Soup framework is the ease of collaboration, consistent metrics and open standards. This enables fast prototyping and user-friendliness for researchers. To achieve this widespread adoption, it is crucial that the components are modular and the interfaces are consistent. Through this standardisation, users can utilise algorithms and components without requiring a full understanding of the algorithms. Stone Soup follows an object-oriented modular approach with *AbstractClass* forming the base class with a *DerivedClass* inheriting its properties and methods from the abstract class. This inheritance is important to preserve the fundamental methods required of a class. For example, all *Predictor* classes must have a `predict()` method and all *Updater* classes must have an `update()` method.

The following subsections highlight the key new components required in the Stone Soup framework for AKKF to run. More details of the implementation and a tutorial can be found in [12].

A. The KernelParticleState Types

The *KernelParticleState* inherits the functionality of the *ParticleState* and adds the `kernel_covar` property as defined in (9) and (12).

B. The Kernel Types

The *Kernel* class provides a transformation from state space into the RKHS represented in (8) by $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ and $K_{\tilde{\mathbf{x}}\mathbf{x}}$ or a transformation from measurement space into the KME space represented in (11) by $G_{\mathbf{y}\mathbf{y}}$ and $\mathbf{g}_{\mathbf{y}\mathbf{y}_k}$. The kernel can be represented as either a polynomial or a Gaussian kernel. The polynomial kernels, *QuadraticKernel* and *QuarticKernel* have the following properties:

- c : is the parameter that trades off the influence of higher-order versus lower-order terms in the polynomial. c in (3) and (4),
- ialpha : is the inverse of α and is the slope parameter that controls the influence of the dot product on the kernel value. $1/\alpha$ in (3) and (4).

The Gaussian kernel (`GaussianKernel`) has the following property

- **variance**: The variance parameter of the Gaussian kernel. σ^2 in (5).

C. Predictor Types

As discussed previously, every `Predictor` class inherits from the base class `Predictor`. All `Predictors` accept a prior *State*, require a `predict()` method and return a *StatePrediction*. Since the AKKF is a derivative of the Kalman filter the base class to inherit from is the `KalmanPredictor`. This allows the framework to distinguish the different class components.

The `AdaptiveKernelKalmanPredictor` is a subclass of `KalmanPredictor` and inherits the methods and properties of the `KalmanPredictor`. The `AdaptiveKernelKalmanPredictor` includes the following new properties;

- **lambda_predictor**: $\lambda_{\tilde{\mathbf{x}}}$ in (7), is a regularisation parameter to stabilise the inverse of the Gram matrix $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$. According to the simulation results presented in [4], the tracking performance of the AKKF is relatively insensitive to the values of $\lambda_{\tilde{\mathbf{x}}}$ when it falls within the range of $[10^{-4}, 10^{-2}]$ [4].
- **kernel**: The `Kernel` class which is chosen to be used to map the state space into the kernel space as described in section III-B.

D. Updater Types

In a similar way to the `KalmanPredictor` class represented the inheritance for all Kalman predictor subclasses, the `KalmanUpdater` provides the same for updaters. All `Updaters` accept a prediction-measurement pair, require an `update()` method and return a *StateUpdate*. The `AdaptiveKernelKalmanUpdater` is a subclass of `KalmanUpdater` and inherits the methods and properties of the `KalmanUpdater`. The `AdaptiveKernelKalmanUpdater` includes the following new properties;

- **lambda_updater**: κ in (13) is a regularisation parameter to ensure the inverse of $G_{\mathbf{y}\mathbf{y}}S_k^-$ is well-defined. The tracking performance of the AKKF is relatively insensitive to κ when $\kappa \sim [10^{-4}, 10^{-2}]$ [4].
- **kernel**: The `Kernel` class which is chosen to be used to map the measurement space into the kernel space.

E. Implementation

Based on the above descriptions, Algorithm 1 summarises the implementation of the AKKF in Stone Soup.

Algorithm 1 Adaptive kernel Kalman filter

- 1: **Initialisation**: Initial particles $\mathbf{x}_0^{(i=1:M_A)}$, Φ_0 , $\mathbf{w}_0 = 1/M_A [1, \dots, 1]_{M_A \times 1}^T$, $\Psi_0 = \Phi_0$. Kernel type and related parameters.
 - 2: **for** $k = 1 : K$ **do**
 - 3: The predictor
 - Input: i.e., $\{\mathbf{x}_{k-1}^{(i=1:M_A)}, \tilde{\mathbf{x}}_{k-1}^{(i=1:M_A)}, \mathbf{w}_{k-1}^+, S_{k-1}^+\}$.
 - Process: $\mathbf{x}_k^{(i)} = \mathbf{f}(\tilde{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_k^{(i)})$, $\mathbf{w}_k^- = \Gamma_k \mathbf{w}_{k-1}^+$, $S_k^- = \Gamma_k S_{k-1}^+ \Gamma_k^T + V_k$.
 - Output: $\{\mathbf{x}_k^{(i=1:M_A)}, \mathbf{w}_k^-, S_k^-\}$.
 - 4: The updater
 - Input: $\{\mathbf{x}_k^{(i=1:M_A)}, \mathbf{w}_k^-, S_k^-\}$.
 - Process: $\mathbf{y}_k^{(i)} = \mathbf{h}(\mathbf{x}_k^{(i)}, \mathbf{v}_k^{(i)})$, $\mathbf{w}_k^+ = \mathbf{w}_k^- + Q_k (\mathbf{g}_{\mathbf{y}\mathbf{y}k} - G_{\mathbf{y}\mathbf{y}} \mathbf{w}_k^-)$, $S_k^+ = S_k^- - Q_k G_{\mathbf{y}\mathbf{y}} S_k^-$.
 - Output: $\{\mathbf{x}_k^{(i=1:M_A)}, \mathbf{w}_k^+, S_k^+\}$.
 - 5: The proposal generator
 - Input: $\{\mathbf{x}_k^{(i=1:M_A)}, \mathbf{w}_k^+, S_k^+\}$.
 - Process: $\tilde{\mathbf{x}}_k^{(i=1:M_A)} \sim \mathcal{N}(\mathbb{E}(X_k), \text{Cov}(X_k))$.
 - Output: $\{\mathbf{x}_k^{(i=1:M_A)}, \tilde{\mathbf{x}}_k^{(i=1:M_A)}, \mathbf{w}_k^+, S_k^+\}$.
 - 6: **end for**
-

IV. DEMONSTRATION

In this section, we report the tracking performance of different filters in the Stone Soup platform. The corresponding dynamic state-space model (DSSM) is described as

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix} \mathbf{u}_k, \quad (17)$$

$$y_k = \tan^{-1} \left(\frac{\eta_k - \eta_s}{\xi_k - \xi_s} \right) + v_k. \quad (18)$$

Here, ΔT represents the sampling period, k represents the time index and $k = 1, \dots, K$. The hidden states are $\mathbf{x}_k = [\xi_k, \dot{\xi}_k, \eta_k, \dot{\eta}_k]^T$, where (ξ_k, η_k) and $(\dot{\xi}_k, \dot{\eta}_k)$ represent the target position and the corresponding velocity in X-axis and Y-axis, y_k is the corresponding observation. The sensor is located at $[\eta_s = 0, \xi_s = 0]$. The process noise \mathbf{u}_k follows Gaussian distribution $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 I)$ and $\sigma_u = 0.01$. Following [14], the prior distribution for the initial state is specified as $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{u}_0, P_0)$ with $\mathbf{u}_0 = [-0.5, 0.001, 0.7, -0.05]^T$ and,

$$P_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.1 & 1 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

Fig. 2 and Fig. 3 displays two representative trajectories and the tracking performance obtained by three filters: the AKKF uses a quartic kernel with 100 particles, the PF with 100 particles, and benchmark performance achieved by the PF with

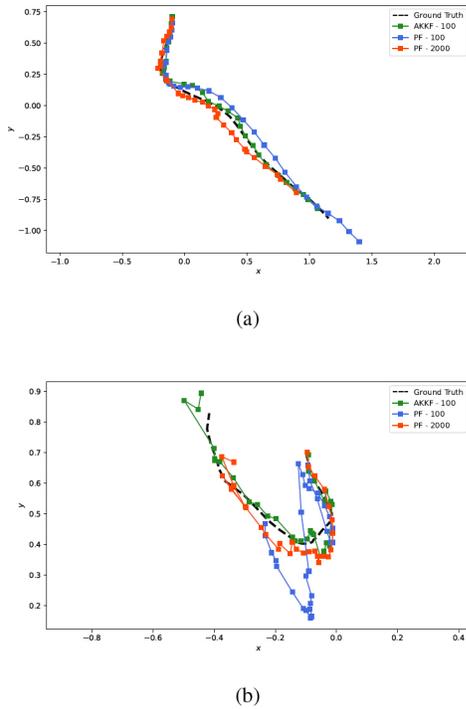


Fig. 2: BOT tracking of a moving target in two dimensions of Trajectory-1 (a) and Trajectory-2 (b) using the quartic kernel-based AKKF with 100 particles; the PF with 100 particles; and the benchmark performance of the PF with 2000 particles. The physical unit on the X-axis and Y-axis is the ‘metre’.

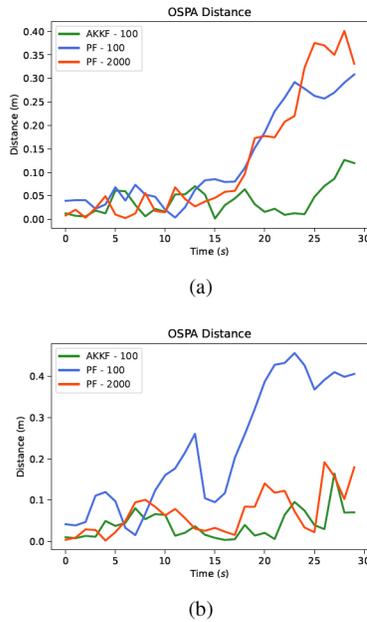


Fig. 3: Performance of BOT tracking of a moving target in two dimensions of Trajectory-1 (a) and Trajectory-2 (b) using the OSPA [13] distance.

2000 particles. Trajectory-1 (Fig 3a) shows that the AKKF performs better than both the PF and the benchmark, while Trajectory-2 (Fig 3b) shows that the AKKF performs alongside the benchmark and outperforms the PF. From the results, we

arrive at the following conclusions. The implementation of the AKKFs in Stone Soup works properly for the BOT problems, it shows improvement and robustness compared to the PF with the same number of particles.

V. CONCLUSIONS

This paper provides the complete implementation of the AKKF in the Stone Soup framework. We utilised and extended existing components to incorporate the AKKF. The new algorithm combines the PF’s non-linearity with the KF’s analytical properties via KMEs. This shows the compatibility of newer algorithms within the Stone Soup framework and establishes a workflow to support other algorithms being implemented in Stone Soup. The authors are interested in extending the AKKF further [6]–[8] by offering additional tutorials and demonstrations in the Stone Soup documentation [11].

REFERENCES

- [1] H. W. Sorenson, *Kalman Filtering: Theory and Application*. NJ: IEEE: Piscataway, 1985.
- [2] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., vol. 3068, International Society for Optics and Photonics. SPIE, 1997, pp. 182 – 193. [Online]. Available: <https://doi.org/10.1117/12.280797>
- [3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [4] M. Sun, M. E. Davies, I. K. Proudler, and J. R. Hoppood, “Adaptive kernel Kalman filter,” *IEEE Trans. Signal Process.*, vol. 71, pp. 713–726, 2023.
- [5] —, “Adaptive kernel Kalman filter,” in *2021 Sensor Signal Processing for Defence Conference (SSPD)*, 2021, pp. 1–5.
- [6] —, “Adaptive kernel Kalman filter based belief propagation algorithm for maneuvering multi-target tracking,” *IEEE Signal Process. Lett.*, vol. 29, pp. 1452–1456, 2022.
- [7] —, “Adaptive kernel Kalman filter multi-sensor fusion,” in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, 2021, pp. 1–8.
- [8] M. Sun, T. Horton, R. Hodgskin-Brown, M. E. Davies, I. K. Proudler, and J. R. Hoppood, “Adaptive kernel Kalman filter for magnetic anomaly detection-based metallic target tracking,” in *2023 Sensor Signal Processing for Defence Conference (SSPD)*, 2023, submitted, pp. 1–5.
- [9] J. Barr, O. Harrald, S. Hiscocks, N. Perree, H. Pritchett, S. Vidal, J. Wright, P. Carniglia, E. Hunter, D. Kirkland, D. Ravall, S. Zheng, A. Young, B. Balaji, S. Maskell, M. Hernandez, and L. Vladimirov, “Stone Soup open source framework for tracking and state estimation: enhancements and applications,” in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXXI*, I. Kadar, E. P. Blasch, and L. L. Grewe, Eds., vol. 12122, International Society for Optics and Photonics. SPIE, 2022, p. 1212205. [Online]. Available: <https://doi.org/10.1117/12.2618495>
- [10] S. Hiscocks, O. Harrald, J. Barr, N. Perree, L. Vladimirov, R. Green, E. Rogers, I. Dorrington, E. Hunter, J. Wright, O. Rosoman, H. Pritchett, J. Osborne, P. Carniglia, C. England, L. Flaherty, D. Kirkland, R. Davies, S. Naylor, and M. Campbell, “Stone Soup.” [Online]. Available: <https://doi.org/10.5281/zenodo.4663993>
- [11] “Stone Soup’s documentation,” 2023. [Online]. Available: <https://stonesoup.readthedocs.io/en/latest/>
- [12] J. Wright and M. Sun, “Stone Soup AKKF Tutorial,” 2023. [Online]. Available: https://stonesoup.readthedocs.io/en/latest/auto_tutorials/filters/AKKF.html
- [13] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.
- [14] J. H. Kotecha and P. M. Djuric, “Gaussian particle filtering,” *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, 2003.

A Lower Complexity Deep Learning Method for Drones Detection

Mohamad Kassab
Research Associate, Thales/SCAI
Mohamed bin Zayed University of AI
Abu Dhabi, UAE
mohamad.kassab@mbzuai.ac.ae

Amal El Fallah Seghrouchni
*Mohammed VI Polytechnic University-
Ai movement*
Rabat, Morocco
Sorbonne University
Paris, France
Amal.Elfallah@lip6.fr

Frederic Barbaresco
Thales Group
Paris, France
frederic.barbaresco@thalesgroup.com

Raed Abu Zitar
Senior Research Scientist, Thales/SCAI
Sorbonne University-Abu Dhabi
Abu Dhabi, UAE
raed.zitar@sorbonne.ae

Abstract—Detecting objects such as drones is a challenging task as their relative size and maneuvering capabilities can deceive machine learning models and cause them to misclassify drones as birds or other objects. In this work, we investigate applying several deep-learning techniques to benchmark real data sets of flying drones. A Deep learning paradigm is proposed for the purpose of mitigating the complexity of those systems. The proposed paradigm consists of a hybrid between the AdderNet deep learning paradigm and the SSD paradigm. The goal was to minimize multiplication operations numbers in the filtering layers within the proposed system and, hence, reduce complexity. Some standard machine learning techniques such as SVM is also tested and compared to other deep learning systems. The data sets used for training and testing were either complete or filtered in order to remove the images with small objects. The types of data were either RGB or IR data. Comparisons were made between all these types and conclusions are presented.

Index Terms—Drones Detection, Deep Learning, Birds versus Drones, Precision of Detection, AdderNet

I. INTRODUCTION

Drones and birds are similar in their flight altitude, velocity, and maneuverability [1]. Several machine learning models including Single Shot Detector (SSD), You Only Look Once (YOLO), Faster-RCNN, and Detection Transformer (DETR) have been applied in the literature to tackle the problem of real-time drone detection [2]. However, all those techniques rely heavily on Convolutional Neural Networks (CNN) which are computationally expensive. Furthermore, to the best of our knowledge, the data sets tested in the literature have only a single class (Drones). This study aims to investigate the use of deep learning/machine learning techniques for the task of airborne object detection. The objectives of this study include examining traditional machine learning techniques such as SVM to highlight the trade-offs between computational

complexity and accuracy for real-time drone detection. The convolutional filters in SSD were replaced with AdderNet filters proposed in [3] to examine the pros and cons of the modified SSD. All of the methods were compared with the replicated results of the performance benchmark in [2].

II. BACKGROUND

Object detection has gone through two milestones during the previous two decades, traditional object detection and deep learning object detection [4]. Traditional object detection techniques mainly depended on handcrafted features due to the lack of image representation techniques in the early days of object detection [4]. A very common and effective traditional object detector proposed by the authors of [5] is the Viola-Jones Detector. This detector uses a sliding window that passes through the whole image with different scales to detect the existence of the desired object [5]. The authors of [6] proposed a Histogram of Oriented Gradients (HOG) as a feature descriptor that can be used with traditional object detection techniques. The most attractive attribute of HOG feature descriptor is its ability to detect objects of different sizes [4]. On the other hand, Counter Unmanned Air Systems (C-UAS) solutions typically use multiple sensors such as radar, electro-optical, acoustic and RF, to reveal potential threats posed by small Unmanned Air Systems (sUASs) or drones to manned aviation, sensitive infrastructure, or military assets, etc. As an example of that is the Thales C-UAS system employs the Thales/Aveillant Gamekeeper staring radar and a camera (SYT 4k) to deliver reliable situational awareness at long ranges (up to 7.5 km).

III. DATA SETS USED

The data sets used in the experiments presented in this paper are Drone-Vs-Bird [7] and Anti-UAV [8].

*This research is supported by the Thales Chair of Excellence Project, Sorbonne Center of Artificial Intelligence, Sorbonne University-Abu Dhabi, UAE.

A. Drone-Vs-Bird Data Set

The data set has been downloaded by contacting wos-detc@googlegroups.com whom granted access to 77 videos. The videos have been converted into RGB images and COCO annotations were created by utilizing the tools available in [2].

One of the drawbacks of this data set is the size of the drones within the image. Moreover, the low quality of the images can cause a significant disadvantage when training a classifier. Furthermore, the drones in some images appear similar to the patterns in the bush of the tree behind it, hence, the classifier may end up learning the wrong patterns. Moreover, the images are created by converting the videos into a sequence of images, therefore, many of the images created from these data sets are replicates. Finally, the data set contains only one class which is drones. The description of the images obtained from the videos in the Drone-Vs-Bird data set is provided in table I. Nevertheless, this data set is a benchmark data set that is used by many researchers and we are going to partially rely on it in this work.

TABLE I: Number of Samples in Drone-Vs-Bird

Object Size	Training Samples	Validation Samples
Small	63295	3578
Medium	21124	1104
Large	2413	63
Background	13034	444
Total	99866	5189

B. Anti-UAV Data Set

The data set can be downloaded directly from [8]. The total number of infrared videos in this data set is 140 and by utilizing the tools in [2], the data set can be converted into infrared images with COCO annotations.

The size of the drones presented in Anti-UAV data set is better than Drone-Vs-Bird, however, the data set still contains small and unclear drone images. Similar to Drone-Vs-Bird, the images are sequences generated from videos, therefore, many repeated images can be found in this data set. Finally, the data set contains one class which is drones. The description of the images obtained from the videos in Anti-UAV data set is provided in table II.

TABLE II: Number of Samples in Anti-UAV

Object Size	Training Samples	Validation Samples
Small	58362	2373
Medium	38660	2670
Large	94	0
Background	1560	197
Total	98676	5240

The following section presents our proposed deep learning method that combines AdderNet with SSD in an attempt to reduce the complexity, followed by comparisons of experiments and results with the most common object detection models (SSD, YOLO, DETR, Faster-RCNN) tested with Drone-Vs-Bird data set and Anti-UAV data set. The AdderNet uses no multiplications in the convolution layers, hence the complexity in this type of deep learning paradigm is expected to be much

less. The SSD is one of the most successful one-stage deep learning paradigms, therefore, it was selected for hybridization with the AdderNet.

IV. DETECTION BASED ON DRONES SIZE

The size of an object is a factor that affects the performance of an object detector [9]. According to the authors of [9], the performance of an object detector can be significantly degraded if the resolution of the picture containing small objects is relatively low. Furthermore, images with extremely high resolution can add high computational costs [9]. The main problem with the Drone-Vs-Bird data set is the low resolution of small-sized objects. Hence, the choice of size and resolution presents a trade-off between performance and complexity. For the purpose of practical applications, complexity must not be increased dramatically as this means increasing the inference time. Henceforth, no matter how accurate the object detector is, it will not be greatly useful.

To determine the optimal object size for object detection without changing the resolution, three thresholds proposed by the authors of COCO data set [10] were used as described below:

- 1) Small = size < 32x32
- 2) Medium = 32x32 < size < 96x96
- 3) Large = size > 96x96

The reason behind doing this experiment is to determine the minimum object size to use in order to have meaningful results at reasonable complexity. This is important as it is desired to create a data set that contains both birds and drones in the future (not available in any of the current data sets). Hence, if the size of the object is too small, the model will not learn any useful information due to the similarities between drones and birds. The Drone-Vs-Bird data set (RGB) was filtered to eliminate small images (area < 32x32). SSD and Faster-RCNN were experimented with using the full data set and the filtered data set to observe the enhancements in mAP results.

TABLE III: Object Size Results SSD

Average Precision	Full Dataset	Filtered Dataset
IoU=0.50:0.95, area = all	0.270	0.525
IoU=0.50, area = all	0.620	0.960
IoU=0.75, area = all	0.172	0.490
IoU=0.50:0.95, area = small	0.164	-
IoU=0.50:0.95, area = medium	0.521	0.520
IoU=0.50:0.95, area = large	0.646	0.665

TABLE IV: Object Size Results Faster-RCNN

Average Precision	Full Dataset	Filtered Dataset
IoU=0.50:0.95 area= all	0.476	0.678
IoU=0.50, area = all	0.734	0.979
IoU=0.75, area = all	0.507	0.815
IoU=0.50:0.95, area = small	0.360	-
IoU=0.50:0.95, area = medium	0.792	0.676
IoU=0.50:0.95, area = large	0.765	0.725

As shown in tables III and IV, there is a significant improvement in mAP results between filtered and full data sets. Hence, this proves that selecting objects with reasonable size can significantly enhance the learning process of object detection

models. Moreover, table III suggests that SSD performance on small RGB objects is relatively low compared to the other methods which is also the conclusion of the authors in [11].

V. HYBRID SSD AND ADDERNET

According to [12] at least 75% of the papers published in popular related conferences target the accuracy of deep neural networks. Insufficient contributions targeting computational complexity were found in the literature. Moreover, improvement in neural network accuracy is a result of increasing the training examples dramatically, hence, accuracy is proportional to the computation complexity. Furthermore, the authors in [13] argue that performance must be judged in terms of both accuracy and computational complexity. This experiment aims to investigate an algorithm to reduce the complexity of deep neural networks. The main focus will be investigating the AdderNet algorithm proposed in [3] and the possibility of extending their findings to object detection.

For the purpose of decreasing the computational complexity (number of multiplications present in the convolution layers), the AdderNet filters proposed in [3] were used to replace the normal convolution filters in the architecture of SSD. Normal convolution filters rely mainly on multiplications, whereas AdderNet filters use addition only [3]. The aim of the hybrid model (SSD + AdderNet) is to decrease the inference time for real-time applications since addition is computationally cheaper than multiplication. The conventional structure of an SSD model as proposed in [14] is shown in Fig. 1. The SSD was chosen since it is a one-stage detector. It was chosen over YOLO-v3, which is also a one-stage detector because it has higher mAP results compared to YOLO-v3 as suggested in other studies [15] [16].

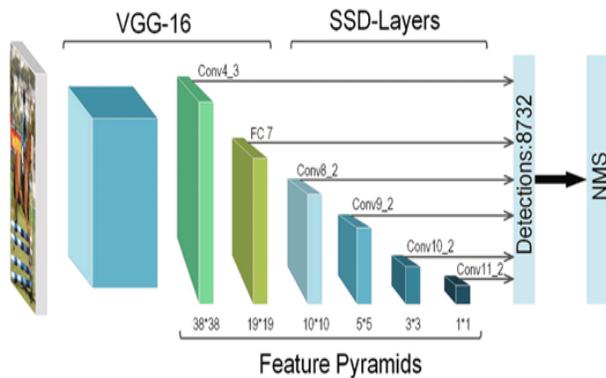


Fig. 1: SSD Model [14]

Initially, it was desired to remove all of the convolution layers present in all parts (backbone and SSD layers), however, this caused the hybrid model to become very unstable (loss reaching infinity while training). Therefore, the replacement was limited to include the VGG 16 layers only. The VGG 16 model introduced in [17] is shown in more detail in Fig. 2.

The blue-labeled layers in Fig. 2 were replaced with AdderNet filters. Unfortunately, the model was still unstable, and the loss reached infinity in the training phase. The AdderNet filters proved to be very sensitive to the change in weight at

each iteration, therefore, to make the model stable during the training phase, the learning rate was initiated from $2e-4$ and decayed to $2e-6$ using SGD optimizer with a momentum of 0.9. As a result of this constraint, the training of the model was possible.

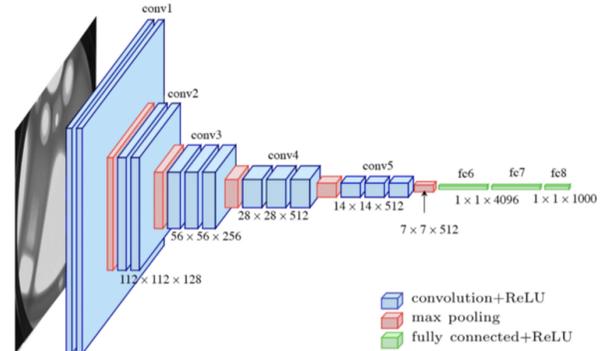


Fig. 2: SSD Backbone [17]

The number of filters, stride, padding, and multiplication of each of the convolutional layer per one RGB image is provided in table V. By implementing the proposed SSD + AdderNet, the number of multiplications has been reduced by 367.7785 M. Please, see table V again that depicts the number of multiplications eliminated.

TABLE V: Number of Multiplications

	Conv 1	Conv 2	Conv 3
Block 1	Number of filters : 64 Size of filter : 3x3 Padding : same Stride : 1 Input size : 224x224x3 Multiplications = 86.704 M	Number of filters : 64 Size of filter : 3x3 Padding : same Stride : 1 Input size : 224x224x3 Multiplications = 86.704 M	-
Block 2	Number of filters : 128 Size of filter : 3x3 Padding : same Stride : 1 Input size : 112x112x3 Multiplications = 44.35 M	Number of filters : 128 Size of filter : 3x3 Padding : same Stride : 1 Input size : 112x112x3 Multiplications = 44.35 M	-
Block 3	Number of filters : 256 Size of filter : 3x3 Padding : same Stride : 1 Input size : 56x56x3 Multiplications = 21.676 M	Number of filters : 256 Size of filter : 3x3 Padding : same Stride : 1 Input size : 56x56x3 Multiplications = 21.676 M	Number of filters : 256 Size of filter : 3x3 Padding : same Stride : 1 Input size : 56x56x3 Multiplications = 21.676 M
Block 4	Number of filters : 512 Size of filter : 3x3 Padding : same Stride : 1 Input size : 28x28x3 Multiplications = 10.838 M	Number of filters : 512 Size of filter : 3x3 Padding : same Stride : 1 Input size : 28x28x3 Multiplications = 10.838 M	Number of filters : 512 Size of filter : 3x3 Padding : same Stride : 1 Input size : 28x28x3 Multiplications = 10.838 M
Block 5	Number of filters : 512 Size of filter : 3x3 Padding : same Stride : 1 Input size : 14x14x3 Multiplications = 2.7095 M	Number of filters : 512 Size of filter : 3x3 Padding : same Stride : 1 Input size : 14x14x3 Multiplications = 2.7095 M	Number of filters : 512 Size of filter : 3x3 Padding : same Stride : 1 Input size : 14x14x3 Multiplications = 2.7095 M

The training was done for 24 epochs on both the DroneVs-Bird (RGB) and the Anti-UAV (IR). The following figure (Fig. 3) shows the training loss of the hybrid SSD model.

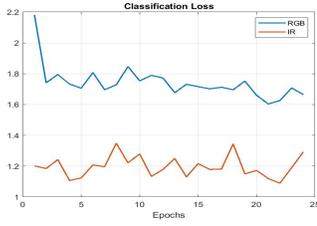


Fig. 3: Hybrid SSD Detection Learning Loss

For both the RGB and IR image groups. The IR group shows fewer values in all types of losses (localization and classification). The testing was done using the trained weights at different epochs. The following tables (VI, VII, and VIII) show the obtained testing mAP (using COCO evaluator) at epoch 3, epoch 12, and epoch 24 for Drone-Vs-Bird (RGB) and Anti-UAV (IR).

TABLE VI: Epoch 3

Average Precision	Results (RGB)	Results (IR)
IoU=0.50:0.95, area = all	0.165	0.538
IoU=0.50, area = all	0.458	0.838
IoU=0.75, area = all	0.064	0.617
IoU=0.50:0.95, area = small	0.085	0.372
IoU=0.50:0.95, area = medium	0.418	0.671
IoU=0.50:0.95, area = large	0.474	-

TABLE VII: Epoch 12

Average Precision	Results (RGB)	Results (IR)
IoU=0.50:0.95, area = all	0.187	0.539
IoU=0.50, area = all	0.508	0.839
IoU=0.75, area = all	0.087	0.620
IoU=0.50:0.95, area = small	0.099	0.376
IoU=0.50:0.95, area = medium	0.448	0.673
IoU=0.50:0.95, area = large	0.503	-

TABLE VIII: Epoch 24

Average Precision	Results (RGB)	Results (IR)
IoU=0.50:0.95, area = all	0.200	0.541
IoU=0.50, area = all	0.533	0.839
IoU=0.75, area = all	0.105	0.626
IoU=0.50:0.95, area = small	0.104	0.377
IoU=0.50:0.95, area = medium	0.473	0.673
IoU=0.50:0.95, area = large	0.495	-

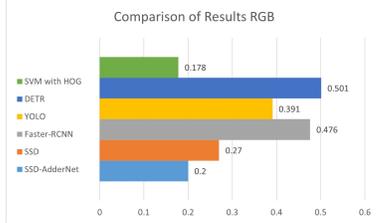


Fig. 4: Comparison of Results RGB

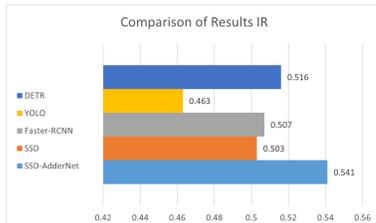


Fig. 5: Comparison of Results IR

The comparison of results as shown in Fig. 4 represent the mAP for the main metric (IoU = 0.50:0.95, area = all). Note that IoU means intersection over the union between the ground truth bounding boxes and predicted bounding boxes. The mAP results (main metric) of the hybrid SSD were compared with the mAP (main metric) of common object detectors (SSD, YOLO-v3, DETR, Faster-RCNN). The training for the common object detectors was done for 24 epochs as well for both RGB and IR images. Additionally, an object detector using SVM and HOG descriptor was trained and tested on RGB images only for the purpose of comparison as well. As shown in Fig. 13, the SSD-AdderNet scored an mAP of 0.20, while the SSD scored an mAP of 0.27. The decrease in testing results is associated with a decrease in computational complexity. Furthermore, the proposed hybrid SSD model scored higher than SVM with HOG when tested on RGB images.

Interestingly, on the IR images, the hybrid SSD model outperformed the other models with an mAP (main metric) of 0.541 as shown in Fig. 5. The performance of the proposed hybrid SSD was much better when tested on IR images as suggested in table VI. However, it can be also seen that there is a slow increase in the testing results as the training epochs increase. This is a consequence of the small learning rate used, however, increasing the learning rate was not possible as it makes the model unstable. Along with the enhanced results when tested on IR images, the hybrid SSD model as shown earlier has 367.7785 M less multiplications. Some of the images detected by the hybrid SSD-AdderNet are shown in Fig. 6(a and b).



(a) Sample Result 1: RGB



(b) Sample Result 2: IR

Fig. 6: Sample Results

All of the experimented object detection methods showed effectiveness in detecting drones, however, it is important to note that the quality of an image (resolution and the size of objects presented in it) are factors that affect the performance and complexity of the model.

VI. CONCLUSIONS

This paper summarizes efforts to improvise a new deep learning paradigm of one stage detector with much less complexity in the filtering layers. The reduced complexity is based on minimizing the number of multiplications in the convolutional layer in an SSD-AdderNet architecture. The training/testing data was extracted from real videos with moderate resolution and mixture of small/medium/large objects (drones) sizes. The goal was to detect the presence of a drone in the image. Despite the low precision achieved by our proposed SSD-AdderNet when trained/tested on RGB images compared to other well-known techniques, the reduction in the complexity was remarkable. However, the performance of the SSD-AdderNet outperformed other models when trained/tested on IR images. The proposed method is recommended to be used for large-to-medium-sized objects when dealing with RGB images where the results are acceptable. Furthermore, the SSD-AdderNet showed good performance when dealing with small images on IR data which is considered to be very promising. The users now have the privilege to switch between high precision detectors such as the DETR or FasterR-CNN for small RGB objects or our less complex and, hence, faster during inference, SSD-AdderNet for larger and higher resolution RGB objects that usually require more computations and time. Future work will focus more on increasing the precision of our proposed system for RGB objects, investigating other types of hybridization for different types of images, and creating a data set that combines the two classes (birds and drones).

REFERENCES

- [1] J. Sim, M. Jahangir, F. Fioranelli, C. J. Baker, H. Dale, Effective ground-truthing of supervised machine learning for drone classification, in: 2019 International Radar Conference (RADAR), IEEE, 2019, pp. 1–5.
- [2] B. K. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon, H. P. Shum, Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1223–1232.
- [3] H. Chen, Y. Wang, C. Xu, B. Shi, C. Xu, Q. Tian, C. Xu, Addernet: Do we really need multiplications in deep learning?, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1468–1477.
- [4] Z. Zou, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, arXiv preprint arXiv:1905.05055 (2019).
- [5] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, Ieee, 2001, pp. I–I.
- [6] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), volume 1, Ieee, 2005, pp. 886–893.
- [7] A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, M. Méndez, D. De la Iglesia, I. González, J.-P. Mercier, et al., Drone vs. bird detection: Deep learning algorithms and results from a grand challenge, *Sensors* 21 (2021) 2824.
- [8] N. Jiang, K. Wang, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, J. Zhao, G. Guo, Z. Han, Anti-uav: A large multi-modal benchmark for uav tracking, arXiv preprint arXiv:2101.08466 (2021).
- [9] Z. Liu, G. Gao, L. Sun, Z. Fang, Hrdnet: high-resolution detection network for small objects, in: 2021 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2021, pp. 1–6.
- [10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.
- [11] J. Park, D. H. Kim, Y. S. Shin, S.-h. Lee, A comparison of convolutional object detectors for real-time drone tracking using a ptz camera, in: 2017 17th International Conference on Control, Automation and Systems (IC-CAS), IEEE, 2017, pp. 696–699.
- [12] R. Schwartz, J. Dodge, N. A. Smith, O. Etzioni, Green ai, *Communications of the ACM* 63 (2020) 54–63.
- [13] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnornet: Imagenet classification using binary convolutional neural networks, in: European conference on computer vision, Springer, 2016, pp. 525–542.
- [14] L. Wei, W. Cui, Z. Hu, H. Sun, S. Hou, A single-shot multi-level feature reused neural network for object detection, *The Visual Computer* 37 (2021) 133–142.
- [15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.
- [16] L. Tan, T. Huangfu, L. Wu, W. Chen, Comparison of retinanet, ssd, and yolo v3 for real-time pill identification, *BMC Medical Informatics and Decision Making* 21 (2021) 1–11.
- [17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

Kalman Filter-Based Suspicious Object Tracking for Border Security and Surveillance Systems using Fixed Automotive Radar

Ji-il Park
Defense Reform Office
The Ministry of National Defense
 Seoul, Republic of Korea
 tinz64@kaist.ac.kr

SeungHyeon Jo
Automotive Business
DXC Luxoft
 Seoul, Republic of Korea
 coolwind@kaist.ac.kr

Hyung-Tae Seo
Department of Mechanical Engineering
Kyonggi University
 Suwon, Republic of Korea
 htseo@kyonggi.ac.kr

Keun Ha Choi
Daedong-KAIST Research Center
for Mobility, KAIST
 Daejeon, Republic of Korea
 choiha99@kaist.ac.kr

Jihyuk Park*
Department of Automotive Engineering
Yeungnam University
 Gyeongsan, Republic of Korea
 jihpark@yu.ac.kr

Kyung-Soo Kim*
Department of Mechanical Engineering
KAIST
 Daejeon, Republic of Korea
 kyungsookim@kaist.ac.kr

Abstract—With recent active research related to autonomous driving, object tracking technology using autonomous driving sensors such as LiDAR and radar has also undergone extensive development. Accordingly, attempts are being made to apply autonomous sensors not only on autonomous vehicles but also in various fields such as security and surveillance. However, since security and surveillance systems should be able to detect and track objects even under extreme environmental conditions such as snow, rain, and fog during the day or night, radar systems that meet the relevant requirements are essential. In South Korea, the distance of the Military Demarcation Line (MDL) is 250 km, and a considerable investment would be required to install more than 1,000 radars and PCs with built-in GPUs in all sections for a border security and surveillance system. Therefore, in this study, a Kalman filter-based object tracking system is explored rather than applying deep learning, which requires GPU processing. Additionally, most objects along the MDL are highly likely to be suspicious objects, so a radar sensor is most suitable because it provides coordinates, distance, and speed of movement without needlessly determining whether an object is an enemy or not. For accurate object detection and tracking performance, two motion models for the Kalman filter, a constant acceleration model (CAM) and a constant turn rate and acceleration model (CTRAM), are compared to identify a suitable model for each object movement state.

Index Terms—Radar tracking, Kalman filter, Security and surveillance, Scientific security system, Defense innovation

I. INTRODUCTION

Radar systems provide essential sensing capabilities for autonomous driving research because they are minimally influenced by environmental noise, such as snow, rain and fog,

This research was a part of the project titled 'Development of Smart Port Autonomous Ships Linkage Technology', funded by the Ministry of Oceans and Fisheries, Korea.

* Corresponding author e-mail: jihpark@yu.ac.kr, kyungsookim@kaist.ac.kr

compared to cameras and LiDAR. Another advantage is that the velocity and position of an object can be accurately measured based on the Doppler effect. For these reasons, radar is used not only in vehicles but also in various other applications, such as aviation and ground surveillance, particularly in the defense field.

In particular, recent attempts have been made to apply automotive radar technology in security and surveillance systems. This is possible due to the recent development of object detection and tracking technology in autonomous driving research, enabling the use of autonomous driving sensors in border security and surveillance systems. The long-range radar systems used in vehicles have a detection distance of 250 m, which is similar to the effective range of rifles; therefore, they are suitable for surveillance purposes in the defense field, and despite the narrow field of view (FOV) of radar, the advantage of being able to monitor a wide area through rapid rotation is important.

Some previous studies have used radar detection data to track objects. Here, object tracking technology refers to an algorithm that uses estimated dynamics to predict the new position of an object in the next frame and update it based on measurements. Research on object tracking using only radar sensors has been mainly conducted in the field of aviation. A study on the application of the EKF and UKF in radar systems for tracking target aircraft was conducted by U.K. Singh et al. [1]. However, in that study, the tracking performance of the EKF and UKF was compared only through simulations. In addition, various other studies have been conducted to improve the tracking performance of radar systems for detecting aircraft [2] [3] [4] [5].

Systems for detecting and tracking objects for surveillance purposes have been mainly developed based on cameras.

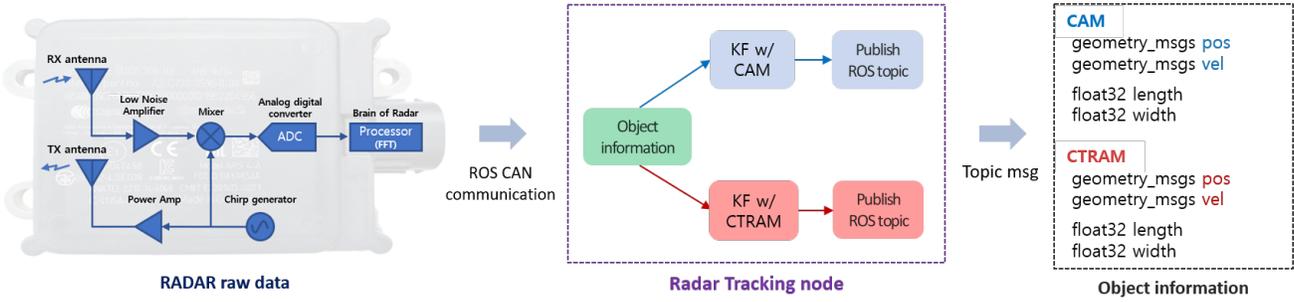


Fig. 1: ROS-based object tracking system configuration

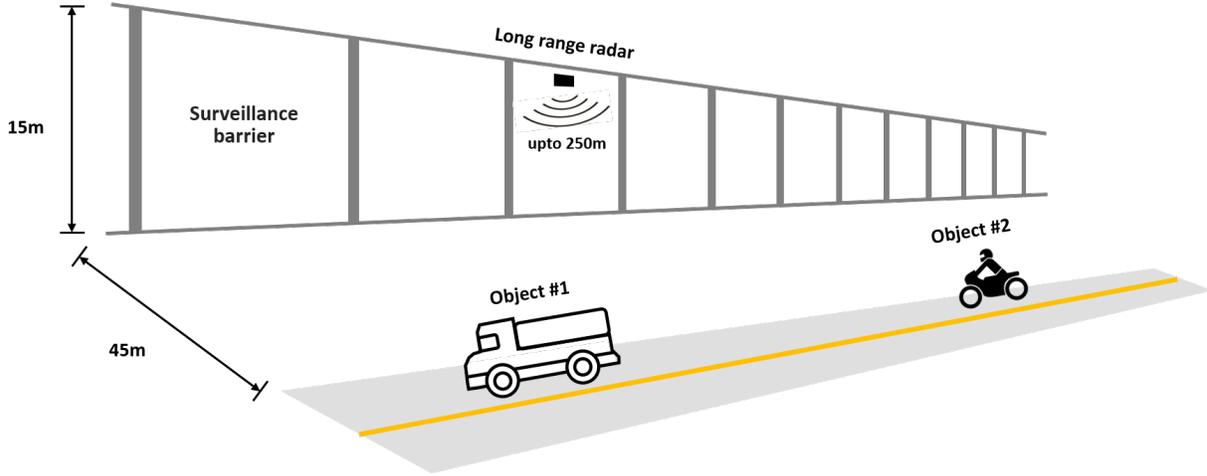


Fig. 2: Acquisition of object tracking data from an experimental environment similar to the real Freedom House area

Object tracking using a camera is a process in which a surveillance system finds an object in every frame of a video and tracks that object over time. The most representative method of camera-based object tracking is the optical flow approach. Sepehr Aslani et al. detected and tracked moving objects through the intensity changes in each video frame [6]. Various other studies have also used cameras to track objects for security and surveillance purposes [7] [8] [9] [10], but it is still difficult to apply this approach at some important security facilities due to the characteristics of cameras, whose performance is not robust against environmental changes [11].

Therefore, this study focuses on improving object tracking performance based on a radar sensor, which is robust to environmental changes and can detect objects over long distances. A comparison of Kalman filter-based object detection systems using two motion models (CAM and CTRAM) is presented. These are the most suitable models for a security and surveillance system to be applied in the border area of the Republic of Korea because they best correspond to the characteristics of the objects that are most likely to be detected in this area.

II. EXPERIMENTAL SETUP

A. Environment configuration

The radar tracking code was written in C++ and configured in a Robot Operating System (ROS) environment. As shown in Fig. 1, the raw data sensed by the radar were transmitted to the radar tracking node through ROS CAN communication, and then the tracking node generated object information from this raw data and published it as an ROS topic. The published topic messages included the position, velocity, length, and width information of the detected objects. Data collection was performed using a Continental ARS 408-21, which is a 77

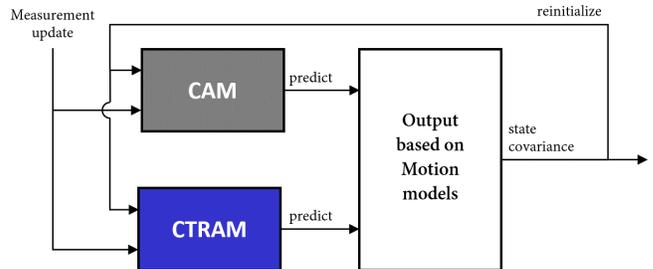


Fig. 3: System structure for a performance comparison of CAM and CTRAM

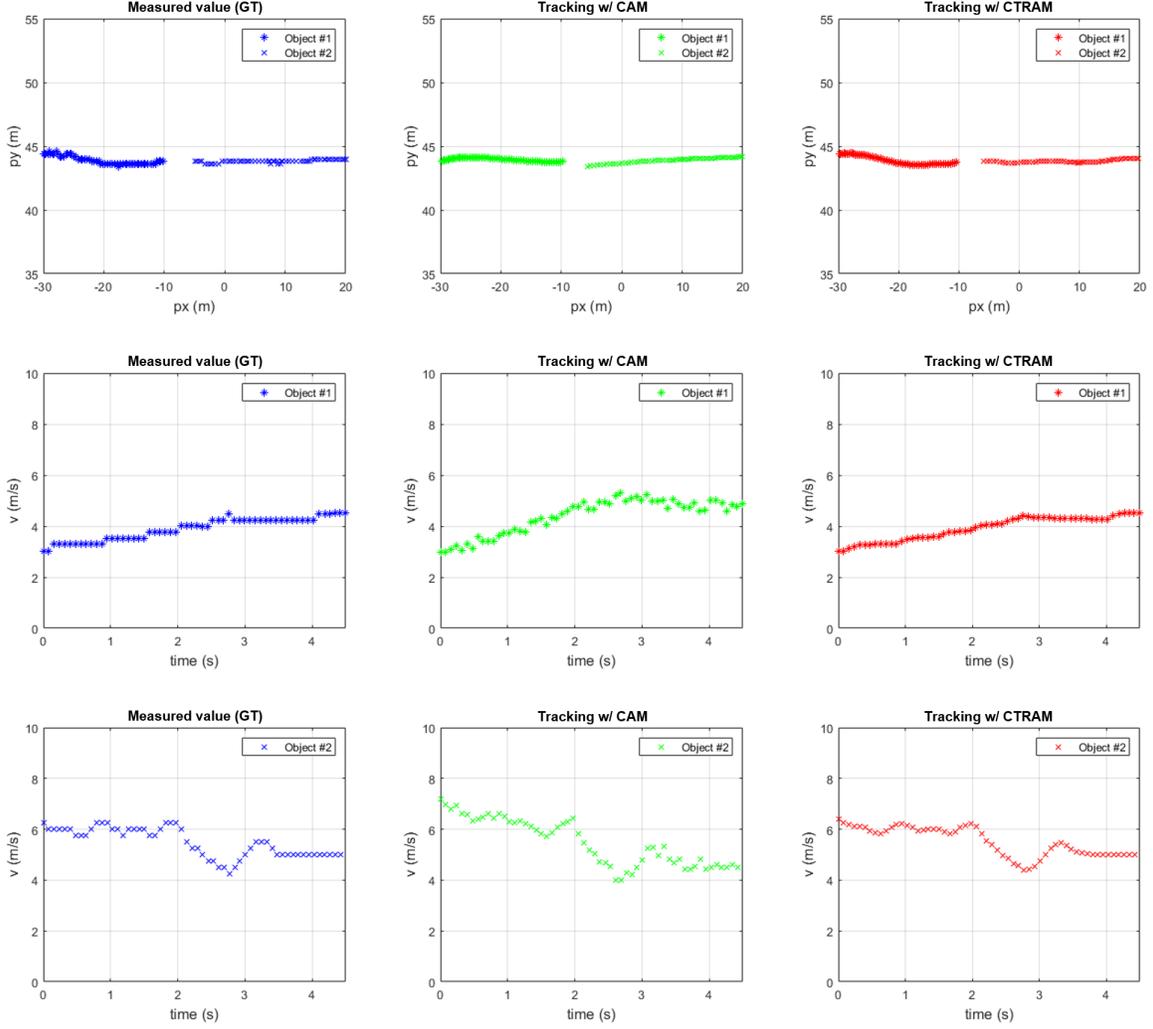


Fig. 4: Comparison of tracking performance with CAM and CTRAM (position and velocity)

GHz long-range radar system capable of sensing over distances of up to 250 m. Additionally, we used an Nvidia Jetson Xavier to obtain data.

$$\begin{aligned}
 p(x) &= p(x) + \frac{1}{\text{yaw}d^2} \{ (v \cdot \text{yaw}d + a \cdot \text{yaw}d \cdot dt) \cdot \sin(\text{yaw} + \text{yaw}d \cdot dt) \\
 &\quad + a \cdot \cos(\text{yaw} + \text{yaw}d \cdot dt) - v \cdot \text{yaw}d \cdot \sin(\text{yaw}) - a \cdot \cos(\text{yaw}) \} \\
 p(y) &= p(y) + \frac{1}{\text{yaw}d^2} \{ (-v \cdot \text{yaw}d - a \cdot \text{yaw}d \cdot dt) \cdot \cos(\text{yaw} + \text{yaw}d \cdot dt) \\
 &\quad + a \cdot \sin(\text{yaw} + \text{yaw}d \cdot dt) + v \cdot \text{yaw}d \cdot \sin(\text{yaw}) - a \cdot \sin(\text{yaw}) \} \\
 \text{yaw} &= \text{yaw} + \text{yaw}d \cdot dt \\
 v &= v + a \cdot dt \\
 \text{yaw}d &= \text{yaw}d \\
 a &= a
 \end{aligned}
 \tag{1}$$

As shown in Fig. 2, the radar system was installed on a 15-meter-high barrier to acquire a representative test dataset for security and surveillance systems. Then, data were acquired

for multiple vehicles on the road 45 m away from the installation point. The difference from previous studies is that the radar sensor in this study was installed at a height of 15 m rather than on the ground, and accordingly, objects were sensed diagonally instead of from the front or side.

B. Data acquisition

As shown in Fig. 2, the experimental site was constructed to be as similar as possible to the Freedom House area, where a North Korean soldier escaped in a vehicle in 2017. Then, object tracking data were obtained for vehicles rapidly approaching the MDL area along the road.

III. OBJECT TRACKING USING A KALMAN FILTER

A. Motion models

Two motion models (CAM and CTRAM) were applied in combination with a Kalman filter to compare the performance of the object tracking system with each model, as shown in

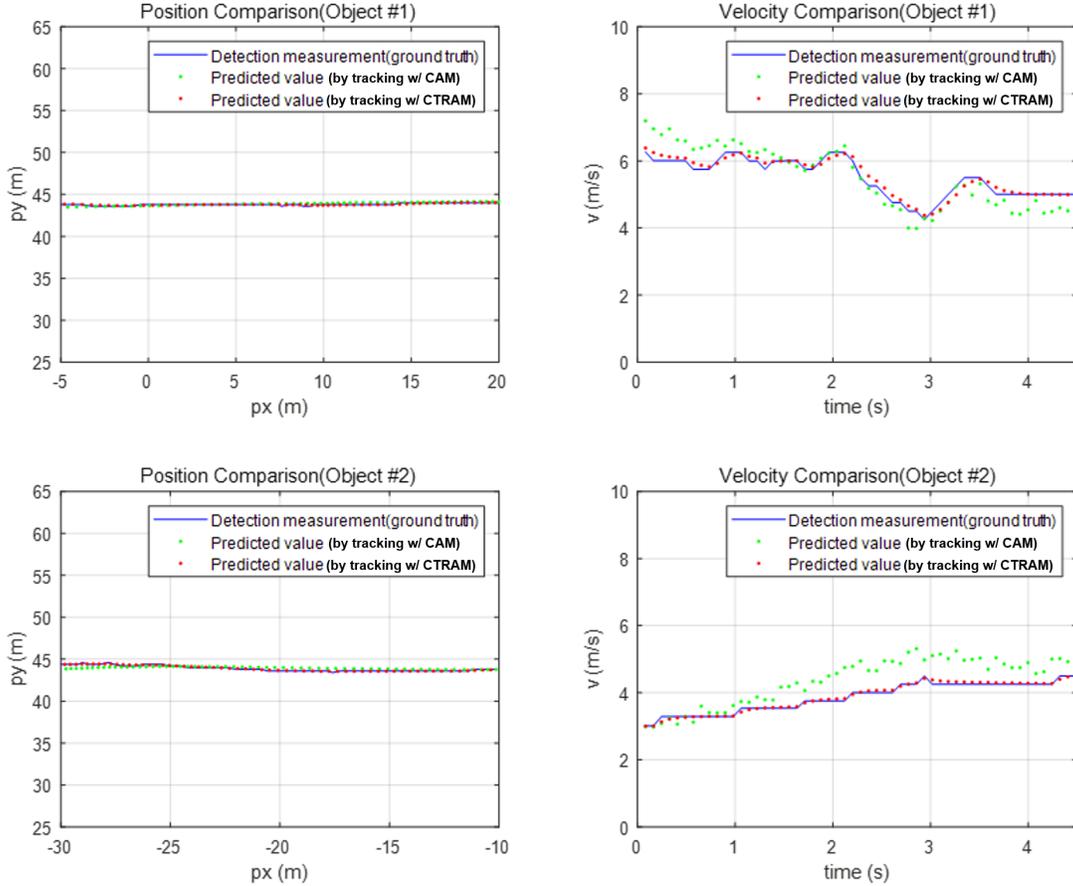


Fig. 5: Tracking performance evaluation for objects 1 and 2 (position and velocity)

Fig. 3. The equations for these two motion models are given in (1), where black font corresponds to CAM and blue font corresponds to CTRAM. Here, the state vectors for CAM and CTRAM are px , py , vel , acc , yaw , and $yawrate$, and the radar measurement noise parameters applied in each model are shown in Table 1.

B. Performance

The tracking performance was analyzed for two simultaneously acquired objects, and the CAM and CTRAM tracking results for the two approaching objects were compared. First, the position accuracy is shown in Fig. 4(a); the positions of both objects are displayed together in this figure. As

TABLE I: Standard deviations of radar measurement noise for CAM and CTRAM. From left to right (as listed for CTRAM), the units of the parameters are m , m/s , m/s^2 , rad , and rad/s , respectively.

Parameter	CAM				CTRAM				
	px, py	vel	yaw	$yawrate$	px, py	vel	acc	yaw	$yawrate$
Value	0.1	0.1	0.1	0.5	0.1	0.1	0.01	0.1	0.5

shown in the position trajectory plot, the tracking results obtained with the Kalman filter exhibit reduced instantaneous position variations compared to the measured values, and the trajectories are smoother. However, the CAM results slightly differ from the measured values, unlike those of CTRAM.

This tendency is even more pronounced for velocity. As with position, the Kalman filter smooths out rapid increases or decreases in velocity compared to the measured values. Moreover, CTRAM displays superior performance compared to CAM, with a smooth trajectory very similar to that of the measured values, as shown in Fig. 4(b).

The error of CAM likely arises because the turn rate is not considered. In the next section, the magnitude of the error associated with this linear approximation is quantitatively evaluated.

C. Evaluation

Object tracking performance should be evaluated by comparing the actual position of an object with its predicted position. However, it can be difficult to obtain the ground-truth position values of objects for reference. For this purpose, a radar system was installed on the ground 10 m from the

TABLE II: Root mean square errors (RMSEs) of CAM and CTRAM w.r.t. the measured data for objects 1 and 2

RMSE	Object 1		Object 2	
	CAM	CTRAM	CAM	CTRAM
Position	0.1660	0.0695	0.3009	0.0902
Velocity	0.7338	0.1264	0.8070	0.0584

TABLE III: Processing time comparison (s)

	Average time	Std. dev. time
Tracking w/CAM	0.00118015	0.00031656
Tracking w/CTRAM	0.00222676	0.00082623

road to obtain accurate object location data to be used as the ground-truth reference values.

The tracking performance of CAM and CTRAM was evaluated by comparing the measured and predicted data for multiple objects moving on the road based on the characteristics of the radar sensor. Fig. 5 shows the measured data (blue line), the CAM tracking results (green line), and the CTRAM tracking data (red line) for the positions and velocities of two objects (objects 1 and 2). The RMSEs between the measured data and the CAM and CTRAM tracking results for these two objects are shown in Table II. The results confirm that the RMSEs of CTRAM are smaller than those of CAM for both the positions and velocities of all objects.

The computational speed performance was compared by calculating the processing times of CAM and CTRAM. The processing time was calculated as the total time from when the measured value was input into the Kalman filter until the predicted value was obtained. Because CAM is based on a relatively simple motion model that does not consider the turn rate, its processing time was very low compared to that of CTRAM, as expected; however, the actual processing time of CTRAM was still very short, and its latency was negligible. This finding suggests that both CAM and CTRAM are capable of real-time object tracking.

IV. CONCLUSION

This paper focuses on tracking multiple target objects by means of a radar sensor for security and surveillance, and the tracking performance of two proposed models is evaluated by comparing the prediction results to measured object data. The data used for analysis in this paper include radar sensor data collected for two objects moving along a road, measured from a height of 15 m. At the time of measurement, the two objects were tracked simultaneously with a fixed radar sensor. The experimental results show that tracking with CTRAM is more robust against measurement errors than tracking with CAM. Notably, CTRAM provides better tracking accuracy for moving objects because it considers not only acceleration but also a constant turn rate. However, its accuracy may vary depending on the movement characteristics of a given object (for example, whether the object is moving in a completely straight line). Thus, either CAM or CTRAM could be op-

timal depending on the movement state of the target object (acceleration, turn rate, etc.). Combined filter-based models that apply various motion models depending on the movement of the detected objects are often used, but such models are complicated because they require motion models for all cases, and errors occur when the motion model changes (e.g., from CAM to CTRAM).

Considering the requirements of boundary monitoring systems (object tracking using a fixed radar sensor, the curvature of the roads in the monitoring area, etc.), applying a motion model suitable for each region can yield good tracking performance. However, in most cases, there will be many roads with slight curves, meaning that CTRAM is expected to be more effective than CAM, as confirmed in this study.

Although there are still many challenges to be solved, this study has confirmed the possibility of applying automotive radar in boundary monitoring systems. Currently, the South Korean military is attempting to implement a science and technology force by promoting “Defense Innovation 4.0”, and in this context, a highly efficient and effective detection system can be developed through the implementation of radar-based security technology using Kalman filters, as proposed in this study. In future research, we will develop optimal sensing hardware and tracking algorithms for application in 4D radar tracking systems for security and surveillance purposes, including military use.

REFERENCES

- [1] U. K. Singh, A. K. Singh, V. Bhatia, and A. K. Mishra, “EKF and UKF based estimators for radar system,” *Frontiers in Signal Processing*, vol. 1, p. 5, 2021.
- [2] J. Shen, Y. Liu, S. Wang, and Z. Sun, “Evaluation of unscented kalman filter and extended kalman filter for radar tracking data filtering,” in *2014 European Modelling Symposium*. IEEE, 2014, pp. 190–194.
- [3] G. Y. Kulikov and M. V. Kulikova, “The accurate continuous-discrete extended kalman filter for radar tracking,” *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 948–958, 2015.
- [4] —, “Accurate continuous-discrete unscented kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking,” *Signal Processing*, vol. 139, pp. 25–35, 2017.
- [5] H. Zhang, J. Xie, J. Ge, W. Lu, and B. Zong, “Adaptive strong tracking square-root cubature kalman filter for maneuvering aircraft tracking,” *IEEE Access*, vol. 6, pp. 10 052–10 061, 2018.
- [6] S. Aslani and H. Mahdavi-Nasab, “Optical flow based moving object detection and tracking for traffic surveillance,” *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 7, no. 9, pp. 1252–1256, 2013.
- [7] J.-C. Lai, S.-S. Huang, and C.-C. Tseng, “Image-based vehicle tracking and classification on the highway,” in *The 2010 International Conference on Green Circuits and Systems*. IEEE, 2010, pp. 666–670.
- [8] V. Reilly, H. Idrees, and M. Shah, “Detection and tracking of large number of targets in wide area surveillance,” in *European conference on computer vision*. Springer, 2010, pp. 186–199.
- [9] S. Zhang, C. Wang, S.-C. Chan, X. Wei, and C.-H. Ho, “New object detection, tracking, and recognition approaches for video surveillance over camera network,” *IEEE sensors journal*, vol. 15, no. 5, pp. 2679–2691, 2014.
- [10] B. Tian, Y. Li, B. Li, and D. Wen, “Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 597–606, 2013.
- [11] H.-Y. Cheng, P.-Y. Liu, and Y.-J. Lai, “Vehicle tracking in daytime and nighttime traffic surveillance videos,” in *2010 2nd International Conference on Education Technology and Computer*, vol. 5. IEEE, 2010, pp. V5–122.

JOINT LEARNING WITH SHARED LATENT SPACE FOR SELF-SUPERVISED MONAURAL SPEECH ENHANCEMENT

Yi Li¹, Yang Sun², Wenwu Wang³, Syed Mohsen Naqvi¹

¹ Intelligent Sensing and Communications Research Group, Newcastle University, UK

² Big Data Institute, University of Oxford, UK

³ Centre for Vision Speech and Signal Processing, University of Surrey, UK

ABSTRACT

Supervised learning has been used to solve monaural speech enhancement problem, offering state-of-the-art performance. However, clean training data is difficult or expensive to obtain in real room environments, which limits the training of supervised learning-based methods. In addition, mismatch conditions e.g., noises in the testing stages may be unseen in the training stage, present a common challenge. In this paper, we propose a self-supervised learning-based monaural speech enhancement method, using two autoencoders i.e., the speech autoencoder (SAE) and mixture autoencoder (MAE), with a *shared layer*, which help to mitigate mismatch conditions by learning a shared latent space between speech and mixture. To further improve the enhancement performance, we also propose phase-aware training and multi-resolution spectral losses. The latent representations of the amplitude and phase are independently learned in two decoders of the proposed SAE with only a very limited set of clean speech signals. Moreover, multi-resolution spectral losses help extract rich feature information. Experimental results on a benchmark dataset demonstrate that the proposed method outperforms the state-of-the-art self-supervised and supervised approaches. The source code is available at <https://github.com/Yukino-3/Complex-SSL-SE>.¹

Index Terms— monaural speech enhancement, self-supervised learning, multi-resolution spectral losses, phase-aware, joint training

1. INTRODUCTION

Monaural speech enhancement has attracted considerable research attention and deep learning techniques have significantly improved its performance with a supervised learning (SL) strategy [1, 2, 3, 4]. However, supervised training of the networks requires large sets of labelled paired data. Moreover, a trained model may suffer from performance degradation when deployed in previously unseen conditions e.g.,

¹For the purpose of open access, the authors have applied a creative commons attribution (CC BY) licence to any author accepted manuscript version arising.

a mismatch of room environments between the training and testing sets. To address the above limitations, self-supervised learning (SSL) techniques are applied as an effective alternative for monaural speech enhancement [5, 6, 7, 8].

The first SSL-based speech enhancement (SSE) method is proposed by Wang et al. [6], where an autoencoder is used to learn a latent representation of clean speech signals as the pre-task, and another autoencoder is used to learn the shared representation between the clean speech and its mixtures. However, the SSE method only learns a shared latent space with unseen speakers [6], its generalization ability to unseen noises and room environments is still limited. Moreover, the phase information of speech signals is ignored in [6]. To address the limitations in [6], we propose a joint training algorithm to improve the speech enhancement performance by using two autoencoders, namely, the speech autoencoder (SAE) and the mixture autoencoder (MAE). The SAE is trained with clean speech signals to learn their latent representations with the amplitude and phase information processed with two individual decoders. The MAE is trained with noisy mixtures recorded in real room environments, where a shared layer from the SAE and MAE is used to obtain a joint latent space of the learned clean speech and noisy mixture representations. The last layer of the encoder in the MAE is replaced by the one in the shared layer after the training stage is completed. To improve the generalization ability of the network model, the training data used for the MAE is unseen in the training data (i.e. unseen room environments) used for the SAE, which helps to train the shared layer to address the mismatch conditions between the training and testing stages.

2. PROPOSED METHOD

2.1. Network Architecture

The block diagram of the proposed method is shown in Fig. 1. Initially, multi-resolution features are extracted from the spectra S i.e., the input of the SAE. In order to preserve the desired information in the signal, in the encoder named E_S , each convolutional layer generates the feature map of a specific resolution, which is then scaled to produce the latent representation

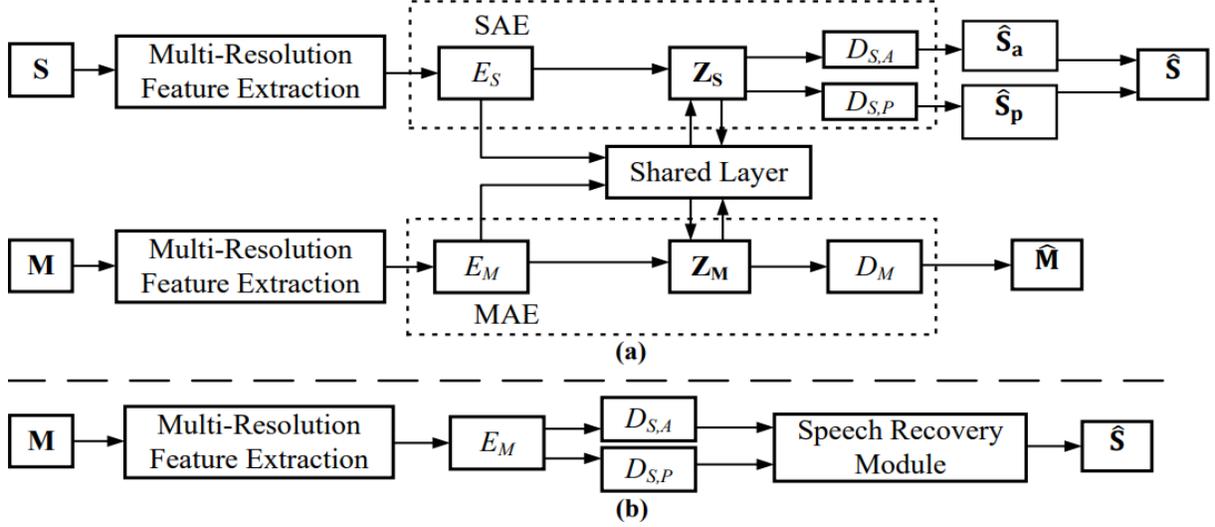


Fig. 1. The overall architecture of the proposed method. (a) Training: From speech spectra \mathbf{S} , the multi-resolution features are extracted with different window sizes as the input of E_S . Then, the latent representation of the speech feature \mathbf{Z}_S is learned via E_S . Then, the reconstructed amplitude and phase of clean spectra are independently obtained as the output by using $D_{S,A}$ and $D_{S,P}$. Similarly, from unseen noisy mixture \mathbf{M} , the multi-resolution features are extracted as the input of the E_M and the mixture feature map \mathbf{Z}_M is learned. Meanwhile, a shared latent space between \mathbf{Z}_S and \mathbf{Z}_M improves the generalization ability of the MAE. (b) Testing: Multi-resolution mixture spectra \mathbf{M} which are unseen with \mathbf{M} in (a) are fed into the trained E_M . The enhanced signal $\hat{\mathbf{S}}$ is obtained with the estimate spectrogram from the speech recovery module.

\mathbf{Z}_S with multi-resolutions. The optimal weights for combining the spectra with each resolution are learned with the target i.e., the feature map of the clean speech, during the training of E_S .

In the proposed method, two decoders $D_{S,A}$ and $D_{S,P}$ are applied in SAE to learn the amplitude and phase of speech, respectively. In detail, the latent representations of both the amplitude and phase are learned by minimizing the discrepancy between the input representation and the corresponding reconstruction. The multi-resolution spectra of the estimated speech signals are obtained.

Different from the SAE, the MAE only requires access to unseen noisy mixtures \mathbf{M} . The multi-resolution features are extracted from the noisy mixture and fed to E_M . Consequently, the latent representation of the mixture is obtained as the output of E_M and exploited to modify the loss functions. Then, the speech feature representation \mathbf{Z}_S and mixture representation \mathbf{Z}_M are used to learn a cross-domain latent space. To achieve that, we concatenate the last layer from both E_S and E_M and create the shared layer between two autoencoders. The mixture representation is passed through the decoders of the SAE to get the enhanced version of the mixture representation. Benefiting from the learned speech representation, a mapping relationship from the mixture to the target speech is learned through $D_{S,A}$ and $D_{S,P}$. The shared latent space between the SAE and MAE is used to further learn the latent representation of the unseen mixture spectra. The last layer of E_M is replaced by the one in the shared layer

after the training stage is completed.

In the testing stage, the feature of the noisy mixture is extracted and fed into the trained E_M to obtain the latent representation of the mixture feature. This representation is then used with the decoders $D_{S,A}$, and $D_{S,P}$ to decode the estimated amplitude and phase of the target speech spectra, respectively. Finally, in the speech recovery module, the phase is recovered by re-wrapping the estimated unwrapped phase of speech. Then, it is used with the recovered speech amplitude to reconstruct the estimated speech signal.

2.2. Loss Functions

Different from previous SSL methods [6, 7, 8, 9, 10], the proposed method exploits multi-resolution feature maps for the network training. Inspired by [11], we use the multi-resolution STFT loss as an auxiliary loss to improve the stability and efficiency for model training. The feature map is rescaled with the same frame shift (i.e. 32), but with different window sizes (1024, 512, 256, and 128). Each STFT loss term estimates the frame-level difference between the clean speech spectrogram and the corresponding reconstructed speech spectrogram.

For the SAE training, the loss \mathcal{L}_S is the sum of four multi-resolution losses defined on amplitude and phase between the clean speech feature and the reconstructed speech feature as:

$$\mathcal{L}_S = \sum_{i=1}^I (\|\mathbf{S}_a^i - \hat{\mathbf{S}}_a^i\|_2^2 + \|\mathbf{S}_p^i - \hat{\mathbf{S}}_p^i\|_2^2) \quad (1)$$

where i refers to the index of the multi-resolution feature maps, subscripts a and p denote the amplitude and phase, respectively. Once the loss function is minimized, we now use the trained SAE and noisy mixtures to train the MAE. The loss \mathcal{L}_M denotes the sum of the multi-resolution losses between the noisy mixture feature and the corresponding reconstruction as:

$$\mathcal{L}_M = \sum_{i=1}^I (\|M^i - \hat{M}^i\|_2^2) \quad (2)$$

Then, the shared layer between the two autoencoders is used to learn a shared latent representation to mitigate the mismatch between the training and testing conditions. To achieve this, the amplitude and phase of Z_M^i are enhanced by the trained $D_{S,A}$ and $D_{S,P}$, respectively. Then, the amplitude and phase of the enhanced spectra are mapped back by E_S to produce the estimated mixture representation \hat{Z}_M^i . The overall MAE loss with the hyper-parameter λ is given as:

$$\mathcal{L}_{MAE} = \mathcal{L}_M + \lambda \cdot \sum_{i=1}^I \left\| Z_M^i - \hat{Z}_M^i \right\|_2^2 \quad (3)$$

3. EXPERIMENTAL RESULTS

3.1. Datasets

The Device And Produced Speech (DAPS) dataset [12] is used in these experiments as [6]. The noisy data consists of 20 speakers (10 female and 10 male) each reading out 5 story excerpts in indoor environments with different real room impulse responses (RIRs). In addition, the clean raw data are collected in an acoustically treated low noise, low reflection vocal booth of a professional recording studio using a microphone with a flat frequency response [12]. Most non-speech sounds such as breaths and lip smacks were removed from the recordings by the sound engineer to create clean speech [12]. We cut 14 minutes of data from each speaker into 28 clips where each clip has 30 seconds long. To show the generalization ability of the proposed SSL method, we split utterances from different speakers in the data preprocessing stage. In the training stage, 420 clean utterances from 15 speakers are randomly selected. For each environment, we first randomly select 28 utterances from a speaker to generate the training data for the SAE. To train the MAE, 392 utterances from 14 speakers are used to generate the mixtures with three different background noises (*factory*, *babble*, and *cafe*) from the NOISEX dataset [13] with four SNR levels (-10, -5, 0, and 5 dB). Therefore, the data used for training MAE is unseen in the data used for training SAE. Moreover, in the testing stage, the remaining 140 utterances of 5 speakers, which are unseen from those in the training stage, are used to generate the mixtures with the same SNR levels but different background noise types and room environments as those in the training stage.

3.2. Experimental Setup and Performance Metrics

Similar to [6, 14, 15], the proposed autoencoders use variational autoencoders (VAEs) as the backbone. In the SAE, E_S , $D_{S,A}$, and $D_{S,P}$ all consist of four 1-D convolutional layers. In the MAE, E_M , $D_{M,A}$, and $D_{M,P}$ all consist of six 1-D convolutional layers. The proposed method is trained by using the Adam optimizer with a learning rate of 0.001 and batch size of 20. The coefficient λ is used in (3) to constraint loss terms and is set empirically with different experiments. For most of the experiments, it is set to 0.01 according to the grid search results by using 0.001, 0.01, 0.1, 1, and 10 as options for the parameter values. However, it is set to 0.1 because the latent representation loss plays a more important role in some specific experiments. The number of training epochs is 700 and 1500 for SAE and MAE, respectively.

Similar to [6], we use composite metrics that approximate the Mean Opinion Score (MOS) including COVL, i.e. the MOS predictor of overall signal quality, CBAK, i.e. the MOS predictor of background-noise intrusiveness, CSIG, i.e. the MOS predictor of signal distortion [16], and Perceptual Evaluation of Speech Quality (PESQ). Higher values of these performance metrics imply better enhancement performance.

3.3. Comparisons with SSL Methods

In this section, we compare the proposed method with three state-of-the-art SSL speech enhancement approaches [6, 7, 8]. The first method is SSE [6] which exploits two autoencoders to estimate speech and mixture, respectively. The second method is the pre-training vector quantization method (PT-VQ) [7], which combines WavLM [17] and Transformer encoder. The third method applies a cross-domain feature (CF) which integrates the SSL representation and spectrogram [8]. This baseline consists of 2 linear layers, two-layered bidirectional long short-term memory (BLSTM) of 256 hidden units and a sigmoid activation to generate the prediction mask. Table 1 shows the speech enhancement performance with PESQ, CSIG, CBAK, and COVL at different SNR levels.

It can be seen from Table 1 that the proposed method outperforms the state-of-the-art SSL methods in terms of all four performance measures. The proposed method and baselines are also compared at different SNR levels. From the experimental results, it can be seen that the proposed method outperforms the baselines even at a relatively low SNR level i.e., -5 dB. The proposed method has 7.6%, 7.0%, 7.8%, and 7.4% improvements compared with the CF method in terms of four performance measures at -5 dB SNR level.

3.4. Comparisons with SL Methods

In this section, we further compare the proposed method with state-of-the-art SL approaches [1, 2, 4]. The DBT-Net aims to recover the coarse- and fine-grained regions of the overall spectrogram in parallel [2]. An attention-in-attention

Table 1. Comparison with SSL methods. Each result is the average value of 1,260 (140 signals×3noise types×3 room environments) experiments. *Italic* shows the proposed methods. **Bold** indicates the best results.

SNR (dB)	PESQ			CSIG			CBAK			COVL		
	-5	0	5	-5	0	5	-5	0	5	-5	0	5
SSE [6]	1.32	1.33	1.34	1.97	2.04	2.09	1.74	1.76	1.77	1.59	1.65	1.68
PT-VQ [7]	1.68	1.70	1.71	2.24	2.27	2.29	1.76	1.79	1.80	1.72	1.77	1.81
CF [8]	1.71	1.74	1.77	2.29	2.30	2.35	1.80	1.89	1.96	1.76	1.80	1.86
<i>Proposed</i>	1.84	1.89	1.91	2.45	2.47	2.49	1.94	2.10	2.23	1.89	1.96	2.03

transformer-based network is adopted for better feature learning. The second method is frequency recurrent convolutional recurrent network (FRCRN) which boosts feature map along the frequency axis [4]. Moreover, in the spectrogram decomposition (SD) method, feature maps are composed of spectra containing evident speech components according to the mask value [1]. These feature maps make the boundary information of speech components clear by ignoring others, thus boosting the sensitivity of the model to input features. Table 2 shows the speech enhancement performance with PESQ, CSIG, CBAK, and COVL.

Table 2. Comparison with SL methods. Each result is the average value of 3,780 experiments (140 signals×3 noise types×3 room environments×3 SNR levels). *Italic* shows the proposed methods. **Bold** indicates the best results.

	PESQ	CSIG	CBAK	COVL
SD [1]	1.68	2.21	1.72	1.66
DBT-Net [2]	1.69	2.21	1.76	1.68
FRCRN [4]	1.72	2.25	1.83	1.74
<i>Proposed</i>	1.88	2.47	2.09	1.96

These SL methods are originally trained with large datasets e.g., VoiceBank [18] and DEMAND datasets [19] which contain 11,572 utterances in [2]. However, in these comparison experiments, we use only 420 utterances in the DAPS dataset to train all the methods because the clean speech data is difficult or expensive to obtain in real-world scenarios, e.g., talking in an office. The training of the supervised methods strongly relies on the large-scale data to facilitate the model to learn structural information [20]. Therefore, the speech enhancement performance of supervised methods suffers from significant degradation compared with its original implementation. In addition, different from the original implementation [2, 4, 1], unseen speakers, noises, and room environments are also used to generate noisy mixtures in the testing stage, which leads to a further drop in the reproduced performance results. In this work, the proposed method uses the shared layer to learn a joint latent space between the SAE and MAE in unseen cases. Thus, the speech enhancement performance is improved although the model is tested in unseen cases.

3.5. Ablation Study

In this section, we investigate the effectiveness of each contribution. Table 3 shows the speech enhancement performance with PESQ, CSIG, CBAK, and COVL.

Table 3. Ablation study of the three contributions in the proposed method. Each result is the average value of 3,780 experiments (140 signals×3 noise types×3 room environments×3 SNR levels). The shared layer and multi-resolution are abbreviated as S-L and M-R, respectively.

Phase	Ablation Settings		PESQ	CSIG	CBAK	COVL
	S-L	M-R				
✗	✗	✗	1.33	2.03	1.76	1.64
✓	✗	✗	1.43	2.15	1.83	1.68
✗	✓	✗	1.59	2.26	1.98	1.81
✗	✗	✓	1.39	2.10	1.79	1.65

From Table 3, it can be observed that the performance is improved by each contribution among all four performance metrics. The improvement of the proposed shared layer is more significant than the use of the phase-aware and multi-resolution spectral losses. Because the shared latent space between the two autoencoders is learned at the last layers of E_S and E_M , the speech signal can be estimated from unseen noisy mixtures using a network that is trainable without labelled training data.

4. CONCLUSION

In this paper, we have presented a self-supervised learning based method with complex spectra and limited training data to address the monaural speech enhancement problem. The cross-domain latent representation for unseen noisy mixtures was learned by using the proposed shared layer. To further improve the generalization ability, we proposed phase-aware decoders and multi-resolution spectral losses based on the multi-resolution feature maps. The experimental results showed that the proposed method outperformed the state-of-the-art approaches in a challenging case where the speakers, background noises, and room environments are unseen in the testing stage. Furthermore, the relationship between the amplitude and phase may be relevant to future studies.

5. REFERENCES

- [1] H. Shi, L. B. Wang, S. Li, J. W. Dang, and T. Kawahara, “Monaural speech enhancement based on spectrogram decomposition for convolutional neural network-sensitive feature extraction,” *Interspeech*, 2022.
- [2] G. C. Yu, A. D. Li, H. Wang, Y. T. Wang, Y. X. Ke, and C. S. Zheng, “DBT-Net: dual-branch federative magnitude and phase estimation With attention-in-attention Transformer for monaural speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.
- [3] Y. Xian, Y. Sun, J. A. Chambers, and S. M. Naqvi, “Geometric information based monaural speech separation using deep neural network,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [4] S. K. Zhao, B. Ma, K. N. Watcharasupat, and W.-S. Gan, “FRCRN: boosting feature representation using frequency recurrence for monaural speech enhancement,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [5] Y. Li, Y. Sun, and S. M. Naqvi, “Self-supervised learning and multi-task pre-training based single-channel acoustic denoising,” *IEEE International Conference on Multisensor Fusion and Integration*, 2022.
- [6] Y.-C. Wang, S. Venkataramani, and P. Smaragdis, “Self-supervised learning for speech enhancement,” *International Conference on Machine Learning (ICML)*, 2020.
- [7] X.-Y. Zhao, Q.-S. Zhu, and J. Zhang, “Boosting self-supervised embeddings for speech enhancement,” *arXiv preprint arXiv:2209.14150*, 2022.
- [8] K.-H. Hung, S.-W. Fu, H.-H. Tseng, H.-T. Chiang, Y. Tsao, and C.-W. Lin, “Boosting self-supervised embeddings for speech enhancement,” *Interspeech*, 2022.
- [9] L. Jing, P. Vincent, Y. LeCun, and Y. D. Tian, “Understanding dimensional collapse in contrastive self-supervised learning,” *arXiv preprint arXiv:2110.09348*, 2021.
- [10] Z. H. Du, M. Lei, J. Q. Han, and S. L. Zhang, “Self-supervised adversarial multi-task learning for vocoder-based monaural speech enhancement,” *Interspeech*, 2020.
- [11] H. Y. Kim, J. Yoon, S. J. Cheon, W. H. Kang, and N. Kim, “A multi-resolution approach to GAN-based speech enhancement,” *Applied Sciences*, vol. 11, no. 2, pp. 721, 2021.
- [12] G. J. Mysore, “Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges,” *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1006 – 1010, 2014.
- [13] A. Varga and H. J. M. Steeneken, “Assessment for automatic speech recognition: Ii. noisex-92: a database and an experiment to study the effect of additive noise on speech recognition systems,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 12, no. 3, pp. 247 – 251, 1993.
- [14] Y. Li, Y. Sun, W. Wang, and S. M. Naqvi, “U-shaped transformer with frequency-band aware attention for speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1511 – 1521, 2023.
- [15] Y. Li, Y. Sun, K. Horoshenkov, and S. M. Naqvi, “Domain adaptation and autoencoder based unsupervised speech enhancement,” *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 1, pp. 43 – 52, 2021.
- [16] Y. Hu and P. C. Loizou, “Evaluation of objective quality measures for speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229 – 238, 2008.
- [17] S. Y. Chen, C. Y. Wang, Z. Y. Chen, Y. Wu, S. J. Liu, Z. Chen, J. Y. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Z. Yu, and F. Wei, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *submitted to Journal of Selected Topics in Signal Processing*, 2022.
- [18] C. Veaux, J. Yamagishi, and S. King, “The voice bank corpus: design, collection and data analysis of a large regional accent speech database,” *IEEE Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, 2013.
- [19] J. Thiemann, N. Ito, and E. Vincent, “The diverse environments multi-channel acoustic noise database: a database of multichannel environmental noise recordings,” *The Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. 3591 – 3591, 2013.
- [20] S.-F. Huang, S.-P. Chuang, D.-R. Liu, Y.-C. Chen, G.-P. Yang, and H.-Y. Lee, “Stabilizing label assignment for speech separation by self-supervised pre-training,” *Interspeech*, 2021.

Underwater Passive Target Classification based on β Variational Autoencoder and MFCC

Adarsh Sunilkumar
Naval Physical and Oceanographic
Laboratory
DRDO
Kochi, India
adarsh.npol@gov.in

Shamju Joseph K
Naval Physical and Oceanographic
Laboratory
DRDO
Kochi, India
shamju.npol@gov.in

Manoj Kumar K
Naval Physical and Oceanographic
Laboratory
DRDO
Kochi, India
manojkk.npol@gov.in

Abstract—Underwater passive target classification is an open set classification problem, where quite often test data of those classes, which were not present during training phase is encountered and it is a challenging task due to the intrinsic complexity of the radiated noise from the target. Conventional classification architectures with spectral processing often fail miserably. Supervised learning methods like deep learning, offers higher success rate but they require enormous amount of data for training and their performance in open set classification is again a challenge. This paper presents an effective method for underwater target classification by the beta variational autoencoder (β - VAE) model with Mel Frequency Cepstral Coefficients (MFCCs) features. MFCC effectively utilises the non-linear auditory effect of the human ear with different frequencies. β - VAE, being one of the generative models, is capable of generalizing with less amount of data. Classification experiments on various underwater targets have been performed with the proposed method, and results indicate that the proposed method is effective in underwater passive target classification.

Keywords— *Open Set Classification, Supervised Learning, Deep Learning, Variational Autoencoder, Mel Frequency Cepstral Coefficients.*

I. INTRODUCTION

Classification of underwater passive target refers to processing of the radiated noise from the target and identifying the type of the target. This has utmost importance in many fields especially in the anti-submarine warfare. The classification of underwater passive target is highly challenging due to the low source level and non-availability of sufficient data. Recently, substantial improvements have been reported in the underwater passive target classification task, mainly due to the advancement in machine learning. Most of the work reported are based on supervised learning, where enormous amount of data is essential for appreciable level of performance. There are many supervised deep learning-based models available in the literature for passive underwater target classification, but they require large amount of data [1]-[3] and their performance in open set classification is not in appreciable level.

There are many open set architectures proposed for classification where the expected performance, when tested on an unseen data will be comparably better [4]-[7]. But most of the method proposed require tuning of many parameters with good amount of data, which is very difficult in underwater passive target classification. In areas like underwater passive field where availability of quality data is of real concern, a model which can generalise effectively with less amount of data is an ideal choice for classification tasks. Generative models are of this nature where it can generalise with less amount of data. Decebal *et al.* [8] proposed a one-shot learning method with a mixture of variational autoencoders for classification tasks. The method proposed above consists of variational autoencoder model for each class, and classification is performed based on the reconstruction error and it is able to achieve better accuracy. Shengchen Li *et al.* [9] proposed β -VAE based heart abnormality detection based on phonocardiogram. They proposed the model as an outlier detector with the requirement of only normal samples during training. Satheesh Chandran C *et al.* [10] proposed a method for underwater target classification using deep generative β -VAE. The above method assumes fixed number of classes for classification, so it does not come under open set architecture. Since this method involves large number of operations like convolution, it is computationally expensive as well. There are many models available in the literature where β -VAE deployed as an outlier detector and attained better performance [11]-[14]. β -VAE being a generative model with unsupervised learning capability is taken as the model in this proposed method. Even though vanilla VAE proven to provide satisfactory results in simple data sets, its beta version, having a hyperparameter beta on the VAE, produces better output even in complex data sets. Beta VAE imposes a limit on the reconstruction error, and it produces a better disentangled representation of the latent vector. Experiments have been performed with the above hyperparameter in the proposed model. Another crucial element for a successful classifier is extracting the suitable feature set. In this paper we propose

MFCC feature from the radiated noise of the target as the feature vector for the β -VAE model for underwater passive target classification. MFCC feature is one of the powerful, widely used feature for speech recognition. Yuze Tong *et al.* [15] proposed MFCC features with k-nearest neighbour (K-NN) algorithm for underwater target classification. The above paper has demonstrated that with combining MFCC features to classify different underwater targets, it can substantially reduce the dependence on sonar operator and there by the interaction of sonar operator with the computer system.

MFCC is capable of estimating human auditory response in a better way than other cepstral feature extraction techniques. This effective mechanism helps to minimize the interaction of sonar operators with the computer system in the actual combat environment. The proposed method is evaluated based on the recording performed during various expeditions conducted in the Indian Ocean region. This method labels a target as unclassified either when its data belongs to a new class or when there is no sufficient confidence to classify it as an existing class. This reduces the chance of misclassification and it provides an information to the operator that the new data can possibly be from a new class of target as well. Reduction of misclassifications and unknown class identifications are one of the major requirements in underwater passive target classification. The proposed method is scalable and can include models for new classes as well without retraining the entire system.

The rest of the paper is organised in the following way. First, the proposed method is discussed, followed by the results and discussions. Then the conclusion is presented.

II. MATERIALS AND METHODS

A. Mel Frequency Cepstral Coefficients

Human auditory system has a non-linear characteristics, and it is more sensitive to low frequencies. MFCC is based on Mel frequency, which can well characterise this non-linearity [15]. The process involved in obtaining MFCC feature vector from the sound signal is given below,

- a) In the pre-process step, framing and windowing is applied on the signal.
- b) Apply the Short Time Fourier Transform and perform the power spectrum calculation.
- c) Map the linear power spectrum into non-linear one on mel scale with the application of triangular filter banks.
- d) Apply the log of these spectrum values to obtain the log filter bank energies.
- e) Take the discrete cosine transform of this log filter bank energies.
- f) MFCCs are the amplitude of the resultant spectrum.

B. β -VAE

VAEs are deep generative networks which have both encoder and decoder networks similar to auto encoders. They learn to map their input X to latent representation z , by learning the probabilistic distribution $Q(z|X)$. VAE assumes that input X and z follow isotropic Gaussian distribution [10].

The encoder and decoder network of VAE is given by $Q_{\phi}(z|X)$ and $P_{\theta}(X|z)$ respectively. ϕ and θ are neural network parameters.

The objective function of VAE is given below in (1),

$$L_{VAE}(\phi, \theta; X) = L_{RC} + L_{KL} \quad (1)$$

where L_{RC} is the reconstruction loss and L_{KL} is the Kullback-Leibler (KL) divergence loss.

The same can be represented mathematically as in (2) below,

$$L_{VAE}(\phi, \theta; X) = E_{Q(z|X)}[\log(P(X|z))] - D_{KL}[Q(z|X) || P(z)] \quad (2)$$

ϕ and θ notations are avoided for simplicity. E and D_{KL} denote expected value and KL divergence respectively. In the above equation first term represents the marginal likelihood of the data, which denotes the reconstruction loss and the second term indicates the KL divergence loss.

β -VAE introduces the use of Lagrange multiplier β on the KL divergence term in the original VAE formulation. It seeks to discover disentangled latent representation. In general, higher value of β encourages learning of disentangled representation, but it is having a trade-off between reconstruction fidelity and quality of disentanglement within the latent representation.

The objective function of the β -VAE is denoted as shown in (3),

$$L_{VAE} = L_{RC} + \beta^* L_{KL} \quad (3)$$

Audio data of the target which is the input to the proposed classification system is split into multiple frames. Then the MFCC values of each frame are calculated. MFCC feature vector from each frame is given as input to the β -VAE model during training. For each class of target, a separate β -VAE is created. Once the training phase is over, the same MFCC feature vectors which are provided during the training phase is fed to the same model, and the reconstruction error is calculated for each frame. 99th percentile error has been noted for each class. Whenever a new test data comes, it passes through each classifier and the reconstruction error is noted, if the newly calculated error is within the 99th percentiles of that class, the method declares the data as an entity belongs to that class. If many such classifiers declare that the new test data belongs to it, the one which has least reconstruction error, will be reported as the class of the test data. If none of the classifiers declare that the test data belongs to it, the method declares that the class of the sample as unclassified.

The pseudocode of the proposed method is given below in Algorithm 1.

Let S be the sample audio signal to be classified, it is split into multiple frames. S_f represents MFCC feature vector of each frame of S . N represents total number of frames. T denotes the threshold, which is the minimum number of frames out of N , in which the proposed method demands that the frequency of any one of the models belongs to a particular class, to declare that the sample belongs to that particular class.

Let $clfr_t$ is a classifier created for some class t , its 99th percentile error is e_t^{99} and $clfr$ is the set of classifiers created for each class of targets.

Algorithm 1: Classifier Algorithm

Input: Audio of the target.

Output: Class of the target.

```

1: Create  $clfr_t, e_t^{99}, t = 1, 2, \dots, M$ ,  $M$  is the total number of
   trained classes.
2: For each  $S_f$  in  $S$  do
3:    $\{\text{minError}, \text{class}\} = \{\}$ 
4:   For each  $clfr_t$  in  $clfr$  do
5:      $e = \text{getReconstructionError}(clfr_t, S_f)$ 
6:     if ( $e \leq e_t^{99}$ ) then
7:        $\text{update}(\{\text{minError}, \text{class}\})$ 
8:     end if
9:   end for
10: end for
11:  $\{\text{class}, \text{frequency}\} = \text{getMostFrequentClass}()$ 
12: if ( $\text{frequency} < T$ ) then
13:    $\text{return " "}$ 
14: end if
15:  $\text{return class}$ 

```

Once the `getMostFrequentClass` function returns the class having most confidence for the majority of frames in the data, the algorithm imposes a condition that the frequency of that class should be at least equal to a threshold specified. If the above condition is not met, then the algorithm will declare the class as unclassified. Thereby it takes care of those samples on which proposed method does not have sufficient confidence and samples belongs to unknown classes as well, in that way it avoids the possibility of misclassification. Once the operator knows the class of those samples declared as unclassified, they can train the model with the actual label, in that case there exists two scenarios, first scenario consists of data belongs to existing class and the second one includes data belongs to a new class. In the first case user have to train the existing β -VAE model with the new data, and in the second case user have to create a new β -VAE and train with the new data, without retraining the entire models. Thereby the proposed method is

scalable, which can take into account of new classes as well, enabling it to perform training in online mode.

III. RESULTS AND DISCUSSIONS

Classification is performed on two kinds of audio data sets, first one consists of three underwater targets, one ship and two submarine targets, they are named as class A, B and C respectively. Second data set consist of five ship targets, namely class D, E, F, G and H. All data have been collected during various expeditions conducted in the Indian ocean with passive sonar systems. First data set is collected from sonar fitted on ship 1 and second set is collected from sonar fitted in ship 2. Data from underwater targets belong to classes other than the ones used for training, called unknown classes is also included in the test phase. U denotes unknown class data. Those data which are declared as unclassified by the proposed method is denoted by UC.

Two parameters have been studied for understanding their significance on the accuracy of the classification by the proposed method. The experiments have been conducted with various values of β , as well as with four different types of activation functions, which includes Rectified Linear Unit (ReLU), Scaled Exponential Linear Unit (SELU), Leaky RELU and Sigmoid activation functions. The tested β values include 0.001, 0.01, 1, 2, 4 and 8.

The proposed method is compared with one class Support Vector Machine (SVM) model and Isolation Forest model. For each class, a separate model is created with one class SVM and isolation forest during training as in the case of the proposed method. Test data passes through all the available models in both of the above methods. The model which declares most frames belong to it, its class will be declared as the class of the test data. Otherwise, the test data is declared as unclassified.

Two sets of MFCC features are also considered for classification. Out of the two sets of MFCC, one is of 40 MFCC coefficients, namely normal MFCC, another one is having the combination of first 12 (static) MFCC coefficients, and their 12 Delta (differential) MFCC coefficients and their 12 delta-delta (acceleration) MFCC coefficients, namely delta MFCC. First coefficient of MFCC has been discarded while selecting the features of both sets. The delta coefficients are computed using the following formula (4)

$$\Delta_t = \frac{\sum_{n=1}^N n(c_{t+n} - c_{t-n})}{2 \sum_{n=1}^N n^2} \quad (4)$$

Δ_t is delta coefficient from frame t , computed from static coefficients c_{t-n} to c_{t+n} . Delta-delta coefficients are also computed with above formula but from differential coefficients instead of static coefficients. N is taken as 2.

The metrics used to evaluate the models are accuracy, precision and recall. Precision used is macro precision, where precision of each class is calculated and average of the same is calculated, similarly macro recall is used for evaluating recall. When the methods declare as unclassified for the above two

data sets, mainly data set 1 and 2, it is counted as a wrong prediction, because the ground truth is known. The methods are also tested with data from unknown classes as well. The results obtained during the analysis phase is presented in the tables below.

A. Data Set 1

Each audio data in data set 1 consists of 512 millisecond duration. Data of classes A, B and C is split into train and test set with 60:40 ratio, samples of targets belong to unknown class is used only during testing phase, the details of the same is given below, in the table 1.

The experiment is repeated 30 times, by randomly splitting data into test and train set with stratified split and different random state and the average accuracy, precision and recall are calculated.

Out of the two sets of features delta MFCC provides better accuracy in all the three methods. Out of the three methods, the proposed method has better performance compared to both one class SVM and isolation forest.

The proposed method attained maximum accuracy, when the β value is 4 and the activation function used is SELU. The best results obtained for all the above methods with delta MFCC feature is given in table 2. Average confusion matrix obtained for the above set of β and activation function in case of the proposed method is given in table 3.

TABLE 1. TRAIN AND TEST OF DATA SET 1.

Class	Train Data Set Size	Test Data Set Size	Total Data Size
A	1080	720	1800
B	864	576	1440
C	768	512	1280
U	0	400	400

TABLE 2. RESULT OF DATA SET 1.

Method	Feature Used	Accuracy	Precision	Recall
Proposed Method	Delta MFCC	94.11%	93.61%	94.67%
One Class SVM	Delta MFCC	71.92%	84.8%	73.63%
Isolation Forest	Delta MFCC	68.45%	84.07%	73.68%

TABLE 3. AVERAGE CONFUSION MATRIX OF PROPOSED METHOD WITH DATA SET 1 IN PERCENTAGE (%).

	A	B	C	UC
A	92.77	0	3.05	4.16
B	0	90.97	0	9.02
C	1.1	0	94.92	3.90
U	0	0	0	100

B. Data Set 2

Each audio data in data set 2 is consists of 512 millisecond duration. Data of target class D, E, F, G and H is split into train and test set with 60:40 ratio. Sample of targets belong to unknown class is used only during testing phase, the details of the same is given below, in the table 4.

The experiment is repeated 30 times, by randomly splitting data into test and train set with stratified split and different random state and the average accuracy, precision and recall are calculated.

In this data set one class SVM and proposed method attained higher accuracy when the feature set used is delta MFCC. Isolation forest attained higher accuracy, when the feature used is normal MFCC. Clearly proposed method is having higher accuracy in the case of data set 2 also.

As in the case of data set 1, the proposed method attained maximum accuracy, when the value of β is 4 and the activation function used is SELU. The best results obtained for all the above methods is given in table 5. Average confusion matrix obtained for the above set of β and activation function in case of the proposed method is given below in table 6.

In the case of both of the datasets, the proposed method declared data belongs to unknown classes are as unclassified. There by the proposed method follows the open set architecture. The proposed method declares those sample, where sufficient confidence is unable to establish as also unclassified and thereby it reduces the chances of misclassification. By design the method is scalable and it is capable of performing online learning as well, thereby it is an ideal choice for underwater passive target classification, especially in the anti-submarine warfare.

TABLE 4. TRAIN AND TEST OF DATA SET 2.

Class	Train Data set size	Test Data set size	Total Data Size
D	720	480	1200
E	744	496	1240
F	696	464	1160
G	864	576	1440
H	888	592	1480
U	0	400	400

TABLE 5. RESULT OF DATA SET 2.

Method	Feature Used	Accuracy	Precision	Recall
Proposed Method	Delta MFCC	96.14%	95.91%	96.13%
One Class SVM	Delta MFCC	74.06%	84.74%	75.78%
Isolation Forest	Normal MFCC	80.11%	86.14%	81.41%

TABLE 6: AVERAGE CONFUSION MATRIX OF THE PROPOSED METHOD WITH DATA SET 2 IN PERCENTAGE (%).

	D	E	F	G	H	UC
D	91.25	2.91	1.67	0	0	4.16
E	0.80	94.35	1.61	0	0	3.22
F	0.86	0	95.68	0	0	3.44
G	0	0	0	97.56	0	2.43
H	0	0	0	0	97.97	2.02
U	0	0	0	0	0	100

IV. CONCLUSION

This paper proposes an effective method for underwater passive target classification by using MFCC features with β -VAE model. The proposed method is compared with one class SVM and isolation forest models and it performs better than both of the methods. The method attained maximum accuracy, when the feature used is delta MFCC, with β value 4 and SELU activation function. This classification method is quite suitable in anti-submarine warfare scenarios, where it is desirable to minimize dependency on human operator. The method is scalable and it can perform online learning without retraining the entire models. The method has less misclassification error and it is effective in identifying unknown classes as well, which is one of the major requirements of underwater passive target classification. Due to the generative potential of the model, the proposed method can provide better performance with less amount of data, which makes it ideal for applications in real combat environment.

ACKNOWLEDGMENT

The authors express sincere gratitude to Director, NPOL for the encouragement in publishing this work as well as for extending all the facility for carrying out this work.

REFERENCES

- [1] Gang Hu, Kejun Wang, Yuan Peng, Mengran Qiu, Jianfei Shi, and Liangliang Liu, "Deep Learning Methods for Underwater Target Feature Extraction and Recognition", Hindawi, Computational Intelligence and Neuroscience Volume 2018, Article ID 1214301. doi: 10.1155/2018/1214301.
- [2] Muhammad Irfan a, Zheng Jiangbin a, Shahid Ali b, Muhammad Iqbal c, Zafar Masood a, Umar Hamid, "DeepShip: An underwater acoustic benchmark dataset and a separable Convolution based autoencoder for classification", ELSEVIER, Expert Systems with Applications 183 (2021) 115270.
- [3] Xiaohan Yin, Xiaodong Sun, Peishun Liu, Liang Wang, Ruichun Tang, "Underwater Acoustic Target Classification Based on LOFAR Spectrum and Convolutional Neural Network", AIAM2020 - 2nd International Conference on Artificial Intelligence and Advanced Manufacture.
- [4] Abhijit Bendale, Terrance E. Boult, "Towards Open Set Deep Networks", IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016. doi: 10.1109/CVPR.2016.173.
- [5] Dan Hendrycks, Kevin Gimpel, "A Baseline For Detecting Misclassified Out-Of-Distribution Examples In Neural Networks", ICLR 2017.
- [6] Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult, "Reducing Network Agnostophobia", 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
- [7] Justin Leo and Jugal Kalita, "Incremental Deep Neural Network Learning using Classification Confidence Thresholding". IEEE Transactions on Neural Networks and Learning Systems, VOL 33, Issue 12, December 2022. Doi : 10.1109/TNNLS.2021.3087104.
- [8] Decebal Constantin Mocanu and Elena Mocanu. "One-Short Learning using Mixture of Variational Autoencoders: a Generalization Learning approach". 17th International Conference on Autonomous Agents and Multiagent systems (AAMAS 2018).
- [9] Shengchen Li, Ke Tian, Rui Wang, "Unsupervised Heart Abnormality Detection Based on Phonocardiogram Analysis with Beta Variational Autoencoders", 2021 IEEE International Conference on Accoustics, Speech and Signal Processing.
- [10] Sathesh Chandran C., Suraj Kamal, A. Mujeeb, and Supriya M. H, "Passive Sonar Target Classification Using Deep Generative β -VAE", IEEE Signal Processing Letters, VOL. 28, 2021.
- [11] Anqi He and Xiaoning Jin, "Deep Variational Autoencoder Classifier for Intelligent Fault Diagnosis Adaptive to Unseen Fault Categories", IEEE Transactions On Reliability, VOL. 70, NO. 4, December 2021.
- [12] Gaurav Dewangan and Seetaram Maurya, "Fault Diagnosis of Machines Using Deep Convolutional Beta-Variational Autoencoder", IEEE Transactions On Artificial Intelligence, VOL. 3, NO. 2, APRIL 2022.
- [13] Vijaya Kumar Sundar, Shreyas Ramakrishna, Zahra Rahiminasab, Arvind Easwaran, Abhishek Dubey, "Out-of-Distribution Detection in Multi-Label Datasets using Latent Space of β -VAE" 2020 Symposium on Security and Privacy Workshops (SPW).
- [14] Luca Bergamin, Tommaso Carraro Mirko Polato, Fabio Aioli, "Novel Applications for VAE-based Anomaly Detection Systems", 2022 International Conference on Neural Networks (IJCNN).
- [15] Yuze Tong, Xin Zhang, Yizhou Ge, "Classification and Recognition of Underwater Target Based on MFCC Feature Extraction", 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC).

Association based Feedback Aided Underwater Passive Target Tracking

Adarsh Sunilkumar
Naval Physical and Oceanographic
Laboratory
DRDO
Kochi, India
adarsh.npol@gov.in

Shamju Joseph K
Naval Physical and Oceanographic
Laboratory
DRDO
Kochi, India
shamju.npol@gov.in

Manoj Kumar K
Naval Physical and Oceanographic
Laboratory
DRDO
Kochi, India
manojkk.npol@gov.in

Abstract— Passive sonar target tracking is a challenging problem, especially due to the high ambient noise as well as multiple tracks originating from the same target. One of the features which is always available in a passive sonar system is the bearing of the target. So, passive bearing line tracking is a widely used tracking method in this field. Passive bearing line tracking is really challenged, in situations like low Signal to Noise Ratio (SNR) scenarios and when targets are spaced close to each other. This paper presents a track association system which associates sensor level tracks like passive bearing line tracks and multiple frequency line tracks, and provides feedback to sensor level tracking. This results in the performance enhancement of sensor level tracks like passive bearing line tracks. An enhanced multiple frequency line tracking (MFLT) method is also proposed in this paper, which will make the passive bearing line tracking more robust and accurate. The effectiveness of the proposed method is illustrated by various multi-target tracking simulation scenarios.

Keywords—Passive Bearing Line Tracking, Track Association, Multiple Frequency Line Tracking.

I. INTRODUCTION

Distributed multisensory systems have gained momentum in underwater sonar applications due to their ability to provide accurate results. In distributed multi sensor underwater sonar systems, individual sensor processing systems produces sensor level tracks and they send their tracks to a central system for associating the information to produce target level tracks. When the sensor level information is diverse, the challenge of associating the information to provide the target level picture increases. There are broadly two kinds of sonars - one is active sonar, where a specified signal is transmitted and the received echo from a target is processed and the other is passive sonar, where the radiated noise from a target is received and processed [1]. Transmitting active signal to detect the presence of a target is highly risky, especially from platforms like submarine, which are involved in the covert operations, because it reveals own identity. So, a passive sonar is an ideal sensor for platforms like submarines. However, passive sonar comes with many challenges, especially due to the non-ideal nature of the underwater environment and high ambient noise

level. Due to these, the noise originating from the targets are received in several paths with low SNR [2]. Passive tracking is one of the critical features present in sonar for analysing the movement of the target as well as for classifying it, especially in anti-submarine warfare.

Passive bearing line tracking is a common tracking mechanism deployed in this field, because azimuth (bearing) information is easy to obtain and not easily get disturbed. This method of tracking is really challenging in very low SNR conditions, as well as in scenarios where targets are closely spaced or when targets are crossing each other. Most of the time passive bearing line tracks fail to represent their targets, as they either degrade or fail to differentiate among themselves in above situations [3].

There are various methods available in the literature for solving of the situations like crossing of targets; however, their applications in passive sonar is less. Availability of ground truth is another challenge when the targets are close to each other. In those situations incorrect measurements can occur, which leads to wrong tracking even in methods with Kalman filter. Sonu Varghese et al. [3] proposed a method based on Nearest Neighbour Joint Probabilistic Data Association (NNJPDA) and Kalman filter. In this method, when targets crosses each other the track bearing depends only on predicted value and not on the measurement. The problem here is that when targets are spaced closely for a long period of time, the predicted values become invalid and this effects overall bearing tracking.

Many tracking methods based only on bearing information [4]-[7] are present in literature. In scenarios where multiple targets are close by or targets are crossing, a tracking mechanism based only on bearing information will fail most of the time. Implementation of those algorithms in the real sonar environment might require manual intervention. In such situations, sonar operator has to manually re-initiate tracks most of the time, and this is a tiresome job. Track fusion-based feedback to individual sensor level tracks has been tried in this field in different forms; one such method is proposed by GUAN Xin et al. [8]. After every update the fused track is

communicated to local sensor level tracks, which uses this information as a priori information. The above method depends only on bearing-based feedback, which may not be accurate always. Another parameter which can be accurately measured in a passive sonar system is frequency of the target. Radiated noise from the underwater target is mainly comprised of mechanical and electrical noise. A single target can emit multiple frequencies. Multiple Hypothesis Tracking (MHT) [9]-[10] approach used in narrow band processing is capable of tracking multiple frequencies of the same target. MFLT implemented with MHT can be used to separate targets closely spaced in bearing. The information present in the MFLT can be used to correctly obtain the measurements for passive bearing line tracking even in scenarios with closely spaced targets as well as during target crossing. In a low SNR scenario, the passive energy may be feeble, but frequency information of the target can be present, and here, the information obtained using MFLT can be used for enhancing passive bearing line tracking.

This paper presents a newly developed MFLT mechanism along with a track association method, which can give sufficient feedback to the passive bearing line tracking systems for correcting their tracking parameters. Track association mechanism is serving two important purposes here - one, it is able to produce target level identity by associating the sensor level tracks, and the other, it is able to provide feedback to individual sensor level tracking systems, which makes their function seamless and robust.

II. MATERIAL AND METHODS

The proposed track-association method, gets sensor level track data from various sensor level tracking systems, including passive bearing line tracking and MFLT. A single target can have multiple frequencies present in it. MFLT is a tracking method which tracks multiple frequencies of the same target, along with the bearing information [9]. Apart from associating the information, the association system provides feedback to individual sensor level track processing units (called trackers), about the correctness of their tracking parameters which is mainly bearing in this case. Individual trackers can assess the feedback and take necessary steps to make it accurate.

A. Multiple Frequency Line Tracking

A single target can have multiple frequencies. MFLT is a system developed to track multiple frequencies of the same target. Once Low Frequency Analyzer and Recorder (LOFAR) or Detection of Envelope Modulation on Noise (DEMON) processing of the signal has been performed, frequency vs. azimuth information will be obtained [1]. The proposed algorithm can be applied on the above information to obtain bearing-frequency tracking which can accommodate multiple frequencies from the same target. The algorithm assumes each target is having at least one frequency component which differentiate that target from another one.

The proposed method comprises of Kalman filter based tracking on the frequency and bearing based model. The method consists of initiation of the track, followed by

searching through the same bearing for identifying potential frequencies which are part of the current track. Each frequency component is taken as a separate track. After checking the consistency of the tracks, consistent tracks enter for further processing. All those near by tracks are identified and they are marked as in holding state. All those tracks which are in holding state, their measurements are not readily available, due to the presence of other tracks in the neighbouring area in terms of both frequency and bearing. Each target will be having at least one track in non-holding state, due to the assumption that at least one frequency component is present in each target which differentiate that target from another one and it can have zero or more tracks in holding state. Initially Kalman filter is applied only on non-holding tracks, once it is processed, Kalman filter will apply on holding tracks by taking the measurement values from their corresponding target's non-holding tracks.

The model used for Kalman Filter is given below in (1)–(6)

$$x(k) = Fx(k-1) + w(k-1) \quad (1)$$

is the state model.

$$y(k) = Hx(k) + v(k) \quad (2)$$

is the measurement model.

$x(k)$ denotes the state vector at time k , and is defined as below,

$$x(k) = \begin{bmatrix} \beta_k \\ \dot{\beta}_k \\ \ddot{\beta}_k \\ f_k \end{bmatrix} \quad (3)$$

β_k is the bearing at time k , $\dot{\beta}_k$ is the bearing rate, $\ddot{\beta}_k$ is the rate of change of bearing rate and f_k frequency.

F is the state transition matrix,

$$F = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 \\ 0 & 1 & T & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

T denotes the time. $w(k-1)$ is the process noise.

In matrix notation,

$$\begin{bmatrix} \beta_k \\ \dot{\beta}_k \\ \ddot{\beta}_k \\ f_k \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 \\ 0 & 1 & T & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_{k-1} \\ \dot{\beta}_{k-1} \\ \ddot{\beta}_{k-1} \\ f_{k-1} \end{bmatrix} + w(k-1) \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

H is called the measurement matrix, $v(k)$ is the measurement noise. Assuming process noise $w(k)$ and measurement noise $v(k)$ are zero mean, Gaussian distributed random variables with variance Q and R respectively, where Q is the process noise covariance matrix and R is the measurement noise covariance matrix.

Once the individual tracks are formed, tracks belonging to the same target has to be identified and need to be associated. Nearest neighbour criteria on the bearing parameter along with the history of association is considered for the above track to track association. The association process of MFLT produces set of sensor level tracks with each track consisting of bearing information and set of frequencies. Each sensor level track produced by the above association corresponds to a target.

The pseudo code for the proposed MFLT algorithm is given below in Algorithm 1.

Let Tr_i represents i^{th} track. Each Track is having a bearing and frequency information, for track Tr_i , they are $Tr_i.b$ and $Tr_i.f$ respectively. Tr_{ij} represents i^{th} track which is associated and it is a part of j^{th} sensor level track. $SLTr_j$ denotes j^{th} sensor level track. $updateMeasurements(Tr, Tr_i)$ function does the measurement update of Tr with values of Tr_i . Tr , Tr' and Tr'' denotes set of tracks, and $SLTr$ denotes set of sensor level tracks.

Algorithm 1: MFLT Algorithm

Input: Frequency and bearing information in a 2D matrix, obtained after the LOFAR or DEMON processing.

Output: Multiple Frequency Line Tracks

```

1: Initiate( $Tr_i$ )
2:  $Tr' = possibleTracksOverSameBearing(Tr_i)$ 
3:  $Tr = getConsistentTracks(Tr')$ 
4: while (  $tillDeletionOfTracks()$  ) do
5:    $Tr'' = getHoldingTracks(Tr)$ 
6:    $kalmanFilter(Tr-Tr'')$ 
7:    $updateMeasurements(Tr'', Tr-Tr'')$ 
8:    $kalmanFilter(Tr'')$ 
9:    $SLTr = association(Tr)$ 
10: end while

```

B. Track Association

In a multi-sensor system, a central entity is required to perform the task of associating the sensor level tracks to form target level tracks, this process is called track association. The proposed method models track to track association problem as an assignment problem. Sensor level tracks have to be associated with existing targets. Well-known Hungarian algorithm [11] is used for solving the above association. The cost matrix for the Hungarian algorithm is created with the frequency and bearing information from the sensor level tracks. The track association system performs one additional task of analysing individual tracks and provides sufficient feedback to the trackers about the accuracy of their reported parameters, mainly bearing in this case.

Let $C[M][N]$, be the cost matrix for the Hungarian algorithm, where M is the total number of tracks of a particular sensor type and N is the total number of targets present. $C[i][j]$ represents the cost of assigning i^{th} sensor level track to j^{th} target. Let $SLTr_i$ be the i^{th} sensor level track and Tg_j be the j^{th} target, having bearing $SLTr_i.b$ and $Tg_j.b$ respectively. $SLTr_i.f$ and $Tg_j.f$ are representing set of frequencies present in the

track and target respectively. Cost matrix will have cost from bearing as well from frequency matching as in (7)–(9).

$$C[i][j] = \text{Bearing Cost} + \text{Frequency Cost} \quad (7)$$

$$\text{Bearing Cost} = d_b(SLTr_i.b, Tg_j.b) \quad (8)$$

$$\text{Frequency Cost} = d_f(SLTr_i.f, Tg_j.f) \quad (9)$$

$d_b(b1, b2)$ represents bearing cost function, which is the bearing difference of $b1$ and $b2$. $d_f(f_1, f_2)$ represents the frequency cost function which is the cost of assigning set f_1 to set f_2 , and it can be obtained by performing Hungarian operation by modelling it as an assignment problem. Cost matrix has been formulated by taking into consideration the maximum bearing and frequency differences that can be allowed to associate a track to a target. If it is not in allowable range the corresponding entry can be filled with a very large value. Let $Tg_j.passive$ and $Tg_j.freqBearing$ represents set of passive sensor level tracks and average bearing of MLFT tracks present in the j^{th} target. Let $SLTr_i.tracker$ represent corresponding tracker for the i^{th} track. \bar{T} represents threshold for sending the feedback, which is decided based on the expected target dynamics. $SLTr_{ij}$ represents i^{th} sensor level track is associated to j^{th} target. Tg' denotes existing targets. $setSLTr$ represent set of sensor level tracks of various types, which include passive bearing line tracks, DEMON MFLT and LOFAR MFLT of various types of sensor arrays present.

Once the cost matrix has been formulated, Hungarian algorithm is applied on the matrix. Then the tracks as well as their associated targets are obtained. In scenarios where targets are close to each other and when they cross each other the passive bearing line tracking may produce incorrect bearing information. This issue can be identified by the track association system by analysing the reported bearings of passive bearing line tracks and multiple frequency line tracks. Based on this information, track association system gives feedback to trackers. The proposed track association algorithm is given below in Algorithm 2. Hungarian($SLTr, Tg$) gives the potential assignments of sensor level tracks to targets by minimizing the total cost. On top of that the proposed method applies some thresholding to validate the assignments. If the cost of assignment of a track to a target is within the threshold, then that track will be assigned to that target, otherwise a new target will be created. The update function present in the Algorithm 2 does the above work.

Algorithm 2: Track Association Algorithm

Input: Passive bearing line tracks and multiple frequency line tracks of various sensor arrays.

Output: Targets level tracks formed by associating the passive bearing line tracks and multiple frequency line tracks. Send feedback to passive bearing line trackers.

```

1:  $trackToTargetAssociation()$  do
2:   for each  $SLTr$  in  $setSLTr$  do
3:     Apply Hungarian ( $SLTr, Tg'$ )

```

```

4:     Update (Tg')
5:   end for
6:   return Tg'
7: end function
8:
9: Tg = trackToTargetAssociation()
10: for each target j in Tg do
11:   for each track i in Tgj.passive do
12:     if ( d( SLTrij.b, Tgj.freqBearing) > T̄ ) then
13:       actualBearing=Tgj.freqBearing
14:       SendFeedback(SLTri.tracker, actualBearing)
15:     end if
16:   end for
17: end for

```

III. RESULTS AND DISCUSSIONS

Simulations results for showing the performance of the proposed method is depicted in this section. The tracker subsystems for passive bearing line tracking taken for the simulation study, work based on the energy detection. The tracker subsystems are modified to incorporate the feedback from the track association subsystem. The passive bearing line tracking is tested in conditions with and without feedback from track association system. MFLT tracks are also assigned during the simulation scenario, by holding the assumption that each target is having at least one frequency which differentiate that target from another one. Five scenarios have been simulated and tested. First four scenarios, whose results are shown in Fig. 1-4 are tested with passive bearing line tracking method based on energy detection performs in a specified bearing window and the fifth scenario, depicted in Fig. 5 is tested with passive bearing line tracking based on Kalman filter. The first four scenarios consist of two moving targets crossing each other with fast and slow speeds, and a moving target crossing a stationary target with fast and slow speed. Each simulation is carried out N number of times, where N=50. Each simulation consists of M number of observations. Root Mean Square Error (RMSE) in kth observation is calculated as in (10),

$$RMSE_k = (1/N) \sqrt{\sum_{i=1}^N (y_k - \tilde{y}_k^i)^2}, \quad k \in [1, M] \quad (10)$$

Where y_k is the true value and \tilde{y}_k^i represents observed value in the kth observation in ith simulation trial. In Fig. 1 target starts crossing around 400th seconds, in Fig.2 it is around 600th seconds, similarly for Fig. 3 and Fig. 4 it is around 400th and 600th seconds respectively. In all the simulated scenarios it is observed that the error in the case of passive bearing line track is less when it incorporates feedback from the track association system. In MFLT the two targets are separated in the bearing vs. frequency space during entire analysis. Association system uses that information to provide feedback to passive bearing line trackers. Once the trackers start using the information provided by the association system it is getting closer to actual path of the target.

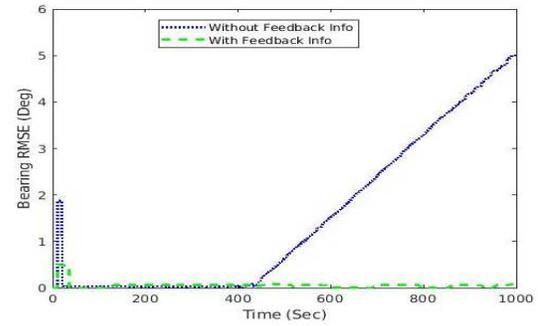


Fig. 1. Bearing RMSE plot showing for a scenario in which two fast moving targets crossing each other.

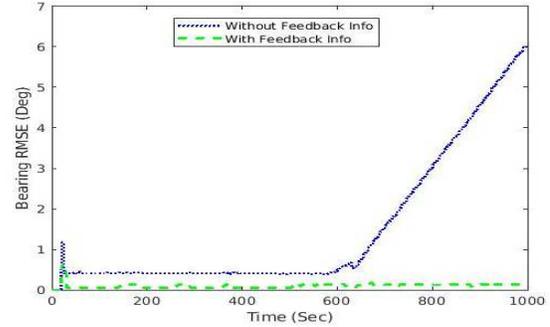


Fig. 2. Bearing RMSE plot showing for a scenario in which two slow moving targets crossing each other.

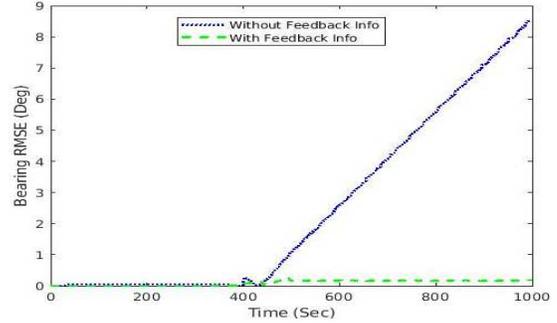


Fig. 3. Bearing RMSE plot showing for a scenario in which a fast moving target crossing a stationary target.

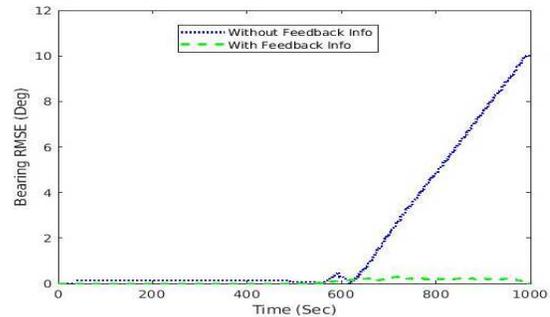


Fig. 4. Bearing RMSE plot showing for a scenario in which a slow moving target crossing a stationary target.

When the targets are spaced closely in terms of bearing for a long period of time, then the passive bearing line tracking which is based purely on bearing will fail most of the time. A

scenario depicting the above case is given in Fig. 5. T1 and T2 represents two targets and V1 represents own ship. Own ship is stationary in the above scenario, curved lines indicate the trajectories of the targets, both targets are moving away from the own ship. In the given scenario both of the targets are closely spaced in terms of bearing for a long period of time from the perspective of own ship. The passive bearing line tracking considered for the above scenario in Fig. 5 consists of Kalman filter having bearing and bearing rate in the state vector. The simulation result showing the passive bearing line tracking with and without feedback from track association system is shown in Fig. 6. From the Fig. 6 it is observed that error is less when the system incorporates feedback from the track association system.

From the simulation results it can be concluded that with the feedback information from the track association system the performance of passive bearing line track is improved and its operation becomes robust and seamless. When targets are crossing fast and when they are spaced closely for a small amount of time, passive bearing line tracking based on models like Kalman filter can function reasonably. But when targets take considerable amount of time for crossing or when they are spaced closely for a long period of time then an algorithm like the proposed method is essential. So, for attaining a reasonable performance in underwater passive target tracking an algorithm like the proposed method is essential, especially in the field of anti-submarine warfare.

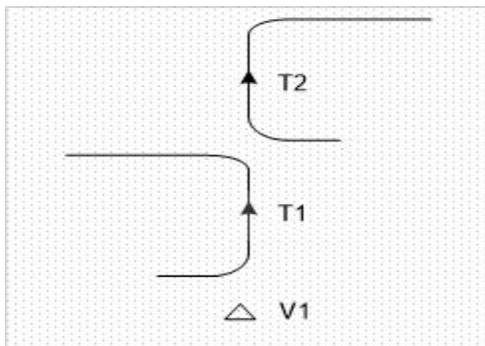


Fig. 5. Scenario showing closely spaced targets for a long period of time from the perspective of own ship.

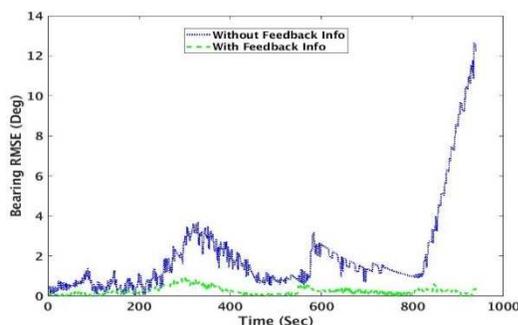


Fig. 6. Bearing RMSE plot showing for the scenario having closely spaced targets for a long period of time.

IV. CONCLUSION

Underwater passive target tracking with feedback from an association system and a newly developed MFLT is proposed in this paper. The passive bearing line tracking with and without feedback from the association system is discussed in detail and simulation studies have been performed. The simulation results show that, with feedback from the association system performance of passive bearing line tracking improves and it becomes robust and it's functions become seamless. Improved performance of passive bearing line tracking enhances overall performance of the association system as well as many other systems like underwater target classification system. The proposed method offers robustness to the passive sonar target tracking with less manual intervention, which is essential in the areas of anti-submarine warfare.

ACKNOWLEDGMENT

The authors express sincere gratitude to Director, NPOL for the encouragement in publishing this work as well as for extending all the facility for carrying out this work.

REFERENCES

- [1] A.D. Waite, "SONAR for Practising Engineers", John Wiley and Sons Ltd.
- [2] R. Urick, "Principles of Underwater Sound", McGraw Hill Inc. 1983.
- [3] Sonu Varghese, Sinchu P, Subhadra Bhai D, "Tracking crossing targets in passive sonars using NNJPDA", 6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India.
- [4] Jonghoek Kim, "Tracking Multiple Targets Using Bearing-Only Measurements in Underwater Noisy Environments", Sensors 2022. doi: 10.3390/s22155512.
- [5] Hadar Shalev and Itzik Klein, "BOTNet: Deep Learning-Based Bearings-Only Tracking Using Multiple Passive Sensors", Sensors, MDPI, 2021. doi: 10.3390/s21134457.
- [6] Yaan Li and Zhenyi Zhao, "Passive Tracking of Underwater Targets Using Dual Observation Stations", Proceedings of 2019 16th International Bhurban Conference on Applied Sciences & Technology (IBCAST), 2019.
- [7] Qiyun Peng, Wujun Li, Lingjiang Kong, "Multi-frame Track-before-detect Algorithm for Passive Sonar System", 2019 International Conference on Control, Automation and Information Sciences (ICCAIS).
- [8] GUAN Xin, YI Xiao and HE You, "Bearings-Only Underwater Track Fusion Solutions with Feedback Information", ICSP'04 Proceedings, 2004.
- [9] Kevin Brinkmann and Jörg Hurka, "Narrowband Passive Sonar Tracking", INFORMATIK 2010. Pages 812-817.
- [10] D. Pillon, N. Giordana, P. Blanc-Benon, S. Sitbon, "Association of narrow band sources in passive sonar", Proceedings of the Fifth International Conference on Information Fusion, pp. 1141 – 1146, vol.2, July, 2002.
- [11] H. W. Kuhn, "The Hungarian method for the assignment problem", doi:10.1002/nav.3800020109.

Computational Enhancement of Accumulated CA-CFAR Process in Side Scan Sonar Data

Ansila V. M.^{*}, Bibin Basheer[†], Sooraj K. Ambat[‡]

Naval Physical & Oceanographic Laboratory

Defence R&D Organization

Thrikkakara, Kochi - 682 021, India.

^{*}ansilavm123@gmail.com [†]bibinbasheer.npol@gov.in [‡]sooraj.npol@gov.in

Abstract—Segmentation plays a vital role in the Side Scan Sonar data processing as it leads to the identification and separation of different objects, features, or areas within the sonar data that have different characteristics or properties. This can make the data more interpretable and easier to analyze, and it can also improve the accuracy of object detection and classification. The Cell-Averaging Constant False Alarm Rate (CA-CFAR) algorithm and its variations are one of the most widely used techniques in processing SSS data. Accumulated CA-CFAR is a modification of the CA-CFAR algorithm proposed recently to improve computational efficiency, making it attractive for real-time SSS applications. We analyse the ACA-CFAR algorithm and propose various modifications to further improve the computational speed, without compromising on accuracy. The performance comparison experiments on different acoustic images show the computational advantage of the proposed method.

Index Terms—Side Scan Sonar (SSS), CA-CFAR, ACA-CFAR, acoustic image segmentation.

I. INTRODUCTION

Side Scan Sonar (SSS) is commonly used in various applications such as marine surveys, search, and recovery to locate and identify sunken vessels or other underwater objects, environmental monitoring, and coastal zone management [2]. SSS is also widely used in military and defence applications to detect and locate underwater mines and other objects that may pose a threat to naval operations.

High-resolution side scan images are characterized by visually distinct areas corresponding to *objects on the seabed*, *shadows*, and *background*. The objects on the seabed are seen as high-intensity, textured areas caused by the reflection of the acoustic wave. It is also known as a *highlight*. The shadows are seen as low-intensity, textured areas caused by the lack of acoustic reverberation from areas surrounding the objects. The background is seen as having distinct areas with strong textured characteristics.

Due to the nature of the environment, which causes erroneous shadows, multi-path returns, and side-lobe effects, object detection in sonar images is a challenging task [3]. The technique of classifying image pixels into different groups is known as image segmentation. The same labels are given to the pixels that belong to the same homogeneous regions. Segmentation is the common method used for separating the highlight, shadow, and background regions. This is due to the fact that pixels that represent objects have values that

are greater than the average pixel intensity in the image, and pixels that represent shadows created by objects have lower values. Choosing thresholds in pixel intensities to differentiate background, highlight, and shadow regions is the most direct method of segmenting the image. Segmentation is a popular and important pre-processing step in SSS data processing, as it allows for identifying and separating different objects, features, or areas within the sonar data with different characteristics or properties. This can make the data more interpretable and easier to analyze, and it can also improve the accuracy of object detection and classification. High-frequency SSS devices (typically 100-500 kHz) generate images with high resolution. In the case of real-time applications, the segmentation needs to be completed within the stipulated time.

Different techniques have been proposed in the literature for segmentation in SSS which includes thresholding, clustering, edge detection, morphological operations, watershed algorithms, level sets, and machine learning algorithms. Thresholding methods are commonly used for segmentation in SSS, which involves setting a threshold value, that separates the data into two or more segments depending on the threshold. CA-CFAR (Cell-Averaging Constant False Alarm Rate) is one of the most popular algorithms used for Segmentation. A computationally efficient modification of the CA CFAR algorithm *viz.* Accumulated CA-CFAR (ACA-CFAR) was recently proposed in the literature [1]. In this work, we propose a couple of improvements in ACA-CFAR to further improve the computational speed, without compromising accuracy.

II. BACKGROUND: CA-CFAR

CFAR (Constant False Alarm Rate) [4] is an algorithm used in radar systems for detecting targets in the presence of noise and clutter. It works by setting a threshold for detection that is adjusted based on the local noise and clutter level in order to maintain a constant false alarm rate. A false alarm is an erroneous radar target detection decision caused by noise or other interfering signals exceeding the detection threshold. This false alarm can be reduced by implementing a CFAR algorithm that maintains the false alarm probability constant such that the threshold value for each cell is updated in accordance with the estimated noise variance. This is achieved by estimating the average of the interference power values of the adjacent n cells, using which an adaptive detection

threshold is adjusted and the probability of a false alarm (α) is maintained. This ensures that contextual information is used in the detection process. This method is known as cell averaging-constant false alarm rate (CA-CFAR) [5]. CA-CFAR is used in sonar applications to detect targets in the presence of reverberation.

CA-CFAR is an adaptive technique that can effectively handle the varying signal characteristics encountered in side-scan sonar data by adapting the threshold based on the expected signal strengths for highlight and shadow regions. The detection task is facilitated by these abrupt variations. [1] proposed a computationally efficient variation of the CA-CFAR algorithm for sonar images. The algorithm segments the image into the acoustical highlight and seafloor reverberation areas. Without losing robustness, the method optimizes computational resources. The major highlight of their work is the simplicity of the algorithm. To reduce the computational cost, they introduced an accumulated matrix \mathbf{A} . The work was a 2-D extension of CA-CFAR, so the method is named accumulated cell averaging-constant false alarm rate in 2-D (ACA-CFAR 2-D). Our work is an extension of ACA-CFAR 2-D with a number of enhancements to further improve computational performance. The ACA-CFAR algorithm is discussed in the next section.

III. ACCUMULATED CA-CFAR IN 2D

It may be noted that the CA-CFAR algorithm discussed in the previous section is computationally very expensive. It analyses the acoustic intensity in each cell under test (CUT) and estimates the threshold to detect the presence or absence of a target in the cell. The detection threshold \hat{T} for each CUT is computed as,

$$\hat{T} = \frac{r}{N_c} \sum_{i=1}^{N_c} x_i,$$

where, x_1, x_2, \dots, x_{N_c} are the N_c neighboring cells of the CUT, and r denotes the constant multiplier chosen either to attenuate or enhance the detection threshold based on the probability of false alarm (α) explained in [1]. The multiplier r is computed as,

$$r = N_c(\alpha^{-1/N_c} - 1) \quad (1)$$

This threshold estimation requires $N_c = [2(N + G) + 1]^2$ memory accesses to calculate the sum of the reference cells, and additionally $G_c = (2G + 1)^2$ memory access for guard cell computations. Here N is the width of the reference window and G is the width of the guard window shown in Fig.1. A typical acoustic image may contain 200-2000 samples, and 2D CA-CFAR computation demands considerable computational resources and time to analyse the entire image. This requirement limits the usage of the 2D CA-CFAR algorithm in SSS systems with real-time requirements.

To alleviate these disadvantages, a computationally improved 2D CA-CFAR algorithm called the Accumulated CA-CFAR (ACA-CFAR) algorithm was proposed in [1]. ACA-

CFAR reduces the repeated computation by pre-calculating the summations of the reference and guard cells.

ACA-CFAR algorithm first computes an accumulation matrix, \mathbf{A} . The $(r, c)^{\text{th}}$ element of \mathbf{A} is calculated as

$$a_{r,c} = \sum_{i=1}^r \sum_{j=1}^c x_{i,j} \quad (2)$$

A graphical representation of accumulation matrix generation from acoustic sample data is given in Fig.3 of [1].

The computation of the threshold, \hat{T} , for each cell requires the summation of the values in the reference cells and guard cells given in Fig.1. In the reference window, the distance from CUT to the farthest cell is the reference distance $d_r = N + G$. And in the guard window, the distance from CUT to the farthest cell is the guard distance $d_g = G$.

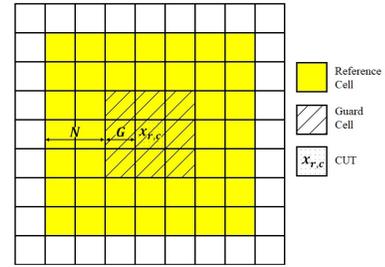


Fig. 1: Generic CFAR window 2-D [1].

Let Σ_R denote the sum of the reference cells of CUT. It may be observed that the sum of the reference cells Σ_R can be computed using only the algebraic sum of four elements of the pre-computed accumulation matrix \mathbf{A} as

$$\begin{aligned} \Sigma_R = & a_{r+d_r, c+d_r} - a_{r-(d_r+1), c+d_r} \\ & - a_{r+d_r, c-(d_r+1)} + a_{r-(d_r+1), c-(d_r+1)} \end{aligned} \quad (3)$$

Note that the above computation requires only four memory accesses from the pre-computed accumulated matrix.

The guard cell computation can also be improved in a similar way, by changing d_r by d_g in (3), we get

$$\begin{aligned} \Sigma_G = & a_{r+d_g, c+d_g} - a_{r-(d_g+1), c+d_g} \\ & - a_{r+d_g, c-(d_g+1)} + a_{r-(d_g+1), c-(d_g+1)} \end{aligned} \quad (4)$$

Detailed explanation for (3) and (4) is given in (17) and (18) of [1].

IV. PROPOSED ENHANCEMENTS IN ACA-CFAR

By analysing the ACA-CFAR method and the pseudo-code given in [1], we identified a couple of possible enhancements to further improve the computational performance of ACA-CFAR. The reasoning and the proposed improvements are explained below.

A. Zero-Padding

It may be noted that (3) and (4) are valid only for the general case in which the CUT is away from the edges of the image. In cases where the distance $d < d_r$ for left, right, below, or above on the CUT, [1] proposes different equations (equations

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1	36	42	48	54	60	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	60	54	48	42	36	
2	42	49	56	63	70	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	70	63	56	49	42	
3	48	56	64	72	80	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	80	72	64	56	48	
4	54	63	72	81	90	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	90	81	72	63	54	
5	60	70	80	90	100	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	100	90	80	70	60	
6	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66	
7	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
8	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
9	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
10	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
11	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
12	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
13	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
14	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
15	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
16	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
17	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
18	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
19	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
20	66	77	88	99	110	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	110	99	88	77	66
21	60	70	80	90	100	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	100	90	80	70	60
22	54	63	72	81	90	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	90	81	72	63	54
23	48	56	64	72	80	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	80	72	64	56	48
24	42	49	56	63	70	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	70	63	56	49	42
25	36	42	48	54	60	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	60	54	48	42	36

Fig. 2: N_c matrix for an image with size 25×30 and $d_r = 5$.

(19), (20) and (21) in [1]) to handle these special cases. These special cases appear as conditional branches in the Pseudocode of ACA-CFAR 2-D (Fig. 4 in [1]). A similar branching appears again while computing the accumulation matrix A .

Algorithm 1 ZeroPadding

Inputs: $X_{m \times n}$, d_r

Output: $Z_{m+2d_r+1, n+2d_r+1}$

$Z = \mathbf{0}_{m+2d_r+1, n+2d_r+1}$

$Z(d_r + 2 : m + d_r + 1, d_r + 2 : n + d_r + 1) = X_{m, n}$

Though modern processors use branch prediction and speculative execution to improve performance, it is well known that removing the branches in the algorithm is the best option to minimize the computation time.

To avoid this branching, we propose to pad an array of d_r+1 zeros to the left and top, and d_r zeros to the right and bottom portions of the input image with size $m \times n$. The *zeroPadding* algorithm is given in Algorithm 1.

Now, after zero-padding, the modified algorithm needs to execute only the general case. This avoids branching and leads to an improvement in the speed of execution.

During accumulated sample matrix (A) creation, the general case for the (i, j) th element is given by, $a_{i,j} = a_{i,j-1} + a_{i-1,j} - a_{i-1,j-1} + I_{i,j}$. The general case during the computation of the double sum of intensities is given in (3) and (4).

B. Pre-Computing the N_c matrix

We can also observe that the number of samples N_c also decreases and varies as the CUT approaches the edges of the image under consideration. This demands repeated calculations of N_c while computing the threshold for each CUT

cell, raising the computation requirement of the ACA-CFAR algorithm. [1] calls the *calculateSample* function for every CUT in the threshold computation step.

In [1], authors proposed to pre-compute the accumulated sample matrix A to reduce the computation. In a similar manner, if we pre-compute the number of samples N_c for each CUT and store it in a matrix, we can save the in situ computations of N_c while calculating the threshold for each CUT cell. For each CUT in (i, j) th locations, we pre-compute N_c and store it in the (i, j) th location of a matrix, say N_c . Since the N_c matrix depends only on the d_r , the same N_c matrix can be re-used for all acoustic images with the same dimension and the same d_r . This property will be advantageous for a typical side-scan sonar where the acoustic image size and d_r will be constant, leading to further computational improvement in online side-scan processing applications.

Computation of N_c matrix: We observed that the N_c matrix holds a nice structure using which we can further reduce the computational requirements of the N_c matrix. It is explained below.

N_c matrix contains the precomputed values of the number of sample cells surrounded by a test cell. For an image with size 25×30 and $d_r = 5$, the N_c matrix is shown in Fig.2. As we can observe from Fig.2, only a few elements in the N_c matrix need to be explicitly computed, and the rest can be copied by exploiting the structure of the matrix.

For efficient N_c matrix computation, the matrix is divided into 9 regions *viz.*, *Top*, *Bottom*, *Left*, *Right*, *Top-Left*, *Top-Right*, *Bottom-Left*, *Bottom-Right*, and *Center*. The regions are shown in Fig.3. It may be observed that the *Top-Left* portion is a symmetric matrix, and all the sub-matrices in the corners (*Top-Left*, *Top-Right*, *Bottom-Left*, *Bottom-Right*) are different

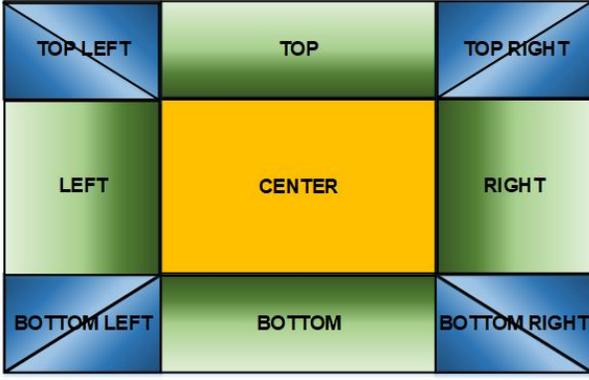


Fig. 3: The nine regions of N_c Matrix

combinations of this matrix. Once we compute the elements in the upper triangular matrix of the *Top-Left* matrix, we can easily find the rest of the elements in the *Top-Left* matrix, as it is a symmetric matrix. The *Bottom-Left* sub-matrix is just the flip of the *Top-Left* sub-matrix. Similarly, the *Top-Right* sub-matrix is a column-wise flip of *Top-Left* sub-matrix and *Bottom-Right* sub-matrix can be found by flipping *Top-Right* matrix row-wise. It may also be observed that the *Bottom sub-matrix* is flip of *Top sub-matrix*, *Left sub-matrix* is the transpose of *Top sub-matrix*, *Right sub-matrix* is column flip of *Left sub-matrix*. The *Center sub-matrix* is a constant matrix with all elements equal to $(2d_r + 1)^2$.

Algorithm 2 Modified ACA-CFAR in 2-D

Inputs: $\mathbf{X} \in \mathbb{R}^{m \times n}$, N , G , α , $N_c \in \mathbb{R}^{m \times n}$

Output: $\mathbf{Y} \in \mathbb{R}^{m \times n}$

$\mathbf{Y} \leftarrow \mathbf{0}_{m \times n}$

$\mathbf{Z} \leftarrow \text{ZeroPadding}(\mathbf{X}) \quad \triangleright \mathbf{Z} \in \mathbb{R}^{m+2d_r+1, n+2d_r+1}$

$\mathbf{A} \leftarrow \mathbf{0}_{m+2d_r+1, n+2d_r+1}$

for $i \leftarrow d_r + 2$ **to** $m + 2d_r + 1$ **do**

for $j \leftarrow d_r + 2$ **to** $n + 2d_r + 1$ **do**

$a_{i,j} \leftarrow a_{i,j-1} + a_{i-1,j} - a_{i-1,j-1} + z_{i,j}$

end for

end for

$B_l \leftarrow a_{m,n} / (m \times n)$

for $i \leftarrow d_r + 2$ **to** $m + d_r + 1$ **do**

for $j \leftarrow d_r + 2$ **to** $n + d_r + 1$ **do**

if $z_{i,j} > B_l$ **then** $\triangleright z_{i,j}$ is $(i,j)^{\text{th}}$ element of \mathbf{Z}

$c \leftarrow N_{c(i-d_r-1, j-d_r-1)}$

$r \leftarrow c \times (\alpha^{-1/c} - 1)$

$\Sigma_R \leftarrow \text{calculateSumCells}(\mathbf{A}, i, j, N + G)$

$\Sigma_G \leftarrow \text{calculateSumCells}(\mathbf{A}, i, j, G)$

$\hat{T} \leftarrow ((\Sigma_R - \Sigma_G) / c) \times r$

if $z_{i,j} > \hat{T}$ AND $\hat{T} > B_l$ **then**

$y_{(i-d_r-1, j-d_r-1)} \leftarrow 1$ (Highlight)

else

$y_{(i-d_r-1, j-d_r-1)} \leftarrow 0$ (Shadow)

end if

end for

end for

end for

C. Branch Optimization

By analysing the pseudo-code of the ACA-CFAR algorithm, it may be further observed that the threshold is computed for all CUT cells, and the cell is declared as a *highlight* if the value is greater than both the computed threshold, denoted by \hat{T} , and *lower Boundary*, denoted by B_l . The *lower Boundary* (B_l) of an image is defined as the average intensity of the image. For an image with size $m \times n$, B_l is given by, $B_l = \frac{a_{m,n}}{m \times n}$.

For any given image, B_l is constant and shall be pre-computed. Instead of checking the conditions $x(i, j) > \hat{T}$ and $\hat{T} > B_l$, if we check the condition $x(i, j) > B_l$ first, then those pixels that satisfy this condition are only required to be passed to the threshold \hat{T} calculation stage. In general, there will be numerous pixels that will be less than B_l , including the pixels in the shadow region, for any acoustic image obtained from a SSS. Thus, a large number of unwanted threshold calculation steps can be eliminated using this modification.

The modified ACA-CFAR is given in Algorithm 2.

Algorithm 3 calculateSumCells

Inputs: $\mathbf{A}, r, c, K \quad \triangleright K \in \{N + G, G\}$

Output: Σ_{rc}

$\Sigma_{rc} = a_{r+K, c+K} - a_{r-(K+1), c+K} - a_{r+K, c-(K+1)} + a_{r-(K+1), c-(K+1)}$

V. EXPERIMENTS AND RESULTS

To compare the computational advantages of our proposed modified ACA-CFAR algorithm with the ACA-CFAR algorithm, we conducted experiments on different SSS data available in the *SeabedObjects-KLSG-II* sonar image dataset [6]. The experiments were conducted using Matlab 2022b. Our comparison studies were performed on a PC with processor Intel® Core™ i7-10700 CPU @ 2.90GHz, 8 cores, 16 threads, memory 16 GB, with Ubuntu 22.04.1 LTS 64-bit OS.

We conducted the experiments in different phases so that the results of the various improvements we suggested could be independently analysed. The following options are analysed:

- 1) ACA-CFAR
- 2) ACA-CFAR with Zero Padding
- 3) ACA-CFAR with N_c Matrix
- 4) ACA-CFAR with Branch Optimization
- 5) ACA-CFAR with All the three Modifications

The measure used for performance analysis is the *average computation time*. The computation time is calculated as the *elapsed time* using the *tic* and *toc* functions available in Matlab. The results are given in Table I.

For the comparison study, the performance improvement of each modification over ACA-CFAR is considered. These results are shown in Table II. It may be observed that each of our modifications results in significant computation improvements over the ACA-CFAR algorithm. For all the images, the proposed method, Modified ACA-CFAR (ACA-CFAR with all three modifications), took 57% less time than ACA-CFAR. This shows the computational advantage of our proposed method.

Fig.	N	G	N_c	G_c	Prob. false alarm (α)	Average computation time (in seconds)				
						ACA-CFAR	ACA-CFAR with Zero Padding	ACA-CFAR with N_c Matrix	ACA-CFAR with Branch Optimization	Modified ACA-CFAR (with All three modifications)
4a	2	1	49	9	0.46019	0.0271	0.0250	0.0220	0.0135	0.0102
4b	3	1	81	9	0.46397	0.0114	0.0108	0.0094	0.0053	0.0045
4c	4	2	169	25	0.47412	0.0317	0.0289	0.0258	0.0152	0.0116
4d	2	1	49	9	0.45641	0.0062	0.0058	0.0050	0.0026	0.0023
4e	3	2	121	25	0.35439	0.0290	0.0263	0.0232	0.0163	0.0123

TABLE I: Comparison of average computation time between ACA-CFAR and Modifications of ACA-CFAR

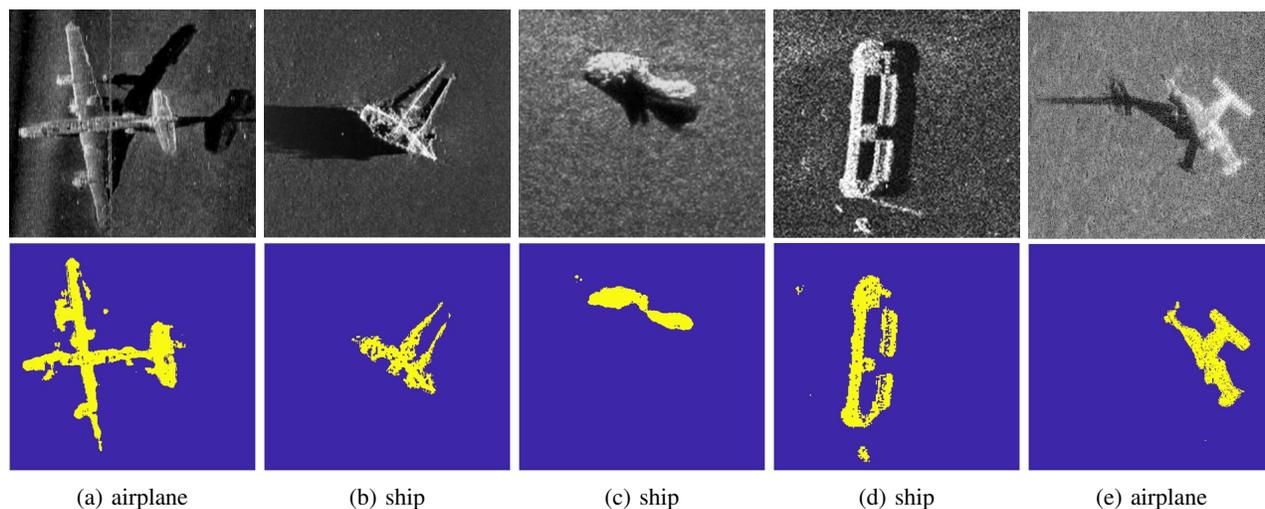


Fig. 4: Results of Modified ACA-CFAR

Fig.	Performance improvement in %			
	ACA-CFAR with Zero Padding	ACA-CFAR with N_c Matrix	ACA-CFAR with Branch Optimization	Modified ACA-CFAR (Algorithm 2)
4a	7.4510	18.5277	50.1214	62.2269
4b	5.1762	17.3903	53.5297	60.0680
4c	8.8587	18.5906	52.0281	63.4931
4d	7.5041	19.9137	58.0055	63.7565
4e	9.2643	19.9546	43.9394	57.7761

TABLE II: Computational improvements (in %) of the proposed modifications of ACA-CFAR compared to ACA-CFAR

The segmentation results of the experiment are shown in Fig. 4. Here, the first row shows the input images and the second row shows the output segmented images. It may be noted that the ACA-CFAR algorithm and our modified ACA-CFAR algorithm give the same output in terms of segmentation. However, our proposed algorithm gives over 57% improvement in computation time, making it more attractive for real-time segmentation applications in SSS.

VI. CONCLUSIONS

Segmentation in SSS data is a vital task in the automatic detection and classification of underwater objects as it divides the data into smaller, meaningful segments or regions. CA-CFAR algorithm is an efficient algorithm used widely for segmentation tasks. However, the 2D version of the algorithm is computationally very expensive and a computationally improved version *viz.* ACA-CFAR was proposed in [1]. We proposed enhancements to the ACA-CFAR algorithm to improve

computational efficiency further. The computational advantage of the proposed algorithm is shown using real-world sonar images available in public datasets. The experiments showed a significant improvement (more than 57%) in the computation time over the ACA-CFAR algorithm. Hence, the proposed algorithm is more attractive and suitable for a real-time SSS system.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Ajithkumar K., Director, NPOL, Mrs. Manjula Rani P., Group Director (ES), and Mrs. Rani Gopakumar, Division Head (IP), for their support during the work.

REFERENCES

- [1] Acosta, G.G, Sebastian A. Villar, "Accumulated CA-CFAR Process in 2-D for Online Object Detection From Sidescan Sonar Data," *IEEE J. Ocean. Eng.*, July 2015, 40, 558–569.
- [2] T. Celik and T. Tjahjadi, "A Novel Method for Sidescan Sonar Image Segmentation," *IEEE J. Ocean. Eng.*, vol. 36, no. 2, pp. 186–194, Apr. 2011.
- [3] Reed, S.; Petillot, Y.; Bell, J. "An automatic approach to the detection and extraction of mine features in sidescan sonar," *IEEE J. Ocean. Eng.*, 2003, 28, 90–105.
- [4] D. K. Barton and S. A. Leonov, *Radar Technology Encyclopedia*. Reading, MA, USA: Artech House, 1998, pp. 91–93.
- [5] M. A. Richards, *Fundamental of Radar Signal Processing*. New York, NY, USA: McGraw-Hill, 2005, ch. VI and VII.
- [6] Huo, G.; Wu, Z.; Li, J., "Underwater Object Classification in Sidescan Sonar Images Using Deep Transfer Learning and Semisynthetic Training Data", *IEEE Access* 2020, 8, 47407–47418

Multi-Target Tracking Using a Swarm of UAVs by Q-learning Algorithm

Seyed Ahmad Soleymani¹, Shidrokh Goudarzi², Xingchi Liu³,
Lyudmila Mihaylova³, Wenwu Wang¹, and Pei Xiao⁴

¹Centre for Vision Speech and Signal Processing (CVSSP), The University of Surrey, UK

²School of Computing and Engineering, University of West London, UK

³Department of Automatic Control and Systems Engineering, The University of Sheffield, UK

⁴Institute for Communication Systems (5GIC), The University of Surrey, UK

Abstract—This paper proposes a scheme for multiple unmanned aerial vehicles (UAVs) to track multiple targets in challenging 3-D environments while avoiding obstacle collisions. The scheme relies on Received-Signal-Strength-Indicator (RSSI) measurements to estimate and track target positions and uses a Q-Learning (QL) algorithm to enhance the intelligence of UAVs for autonomous navigation and obstacle avoidance. Considering the limitation of UAVs in their power and computing capacity, a global reward function is used to determine the optimal actions for the joint control of energy consumption, computation time, and tracking accuracy. Extensive simulations demonstrate the effectiveness of the proposed scheme, achieving accurate and efficient target tracking with low energy consumption.

Index Terms—Multi-target tracking, UAV, Q-Learning, Edge Computing.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have emerged as a highly promising platform for target tracking systems, primarily owing to their exceptional mobility, adaptable deployment capabilities, and cost-effectiveness [1]. The versatility of UAVs lies in their ability to cover vast areas across different altitudes and locations, while also offering superior Line-of-Sight (LoS) links compared to ground Base Stations (BSs), courtesy of their elevated altitude. Consequently, UAVs stand as an ideal choice for target tracking applications. Especially in challenging scenarios where ground service agents are unavailable, UAVs play a pivotal role, diligently and precisely tracking targets [2].

However, limited communication range, battery capacity, and computing capacity are the main challenges of UAVs in a target tracking system. To deal with these challenges, a swarm of autonomous UAVs can be effective. A swarm of UAVs can be used to ensure effective communication coverage in the long term. The utilization of Edge Computing (EC) is also a promising solution to tackle the challenges faced by UAVs. For example, by leveraging the computational capacity of the edge, compute-intensive operations of UAVs can be offloaded to Edge Nodes (ENs) and as a result, enhance both computing quality and the lifetime of the UAVs network [3], [4]. As shown in [5], UAV-enabled EC has been conceptualized as a viable option to enhance the target tracking process.

In recent years, UAV-aided target detection and tracking has been studied. In [6], a Deep Q-Network (DQN) was constructed, with a finite action space, to deal with the limited field of view (FOV) of the camera equipped on the UAV, where a reward function was designed to take into account whether a target is within the FOV. In [7], authors introduced a motion planning algorithm based on the unscented Kalman filter (UKF) for UAVs to estimate the state of the target. The motion planner determines the UAV trajectory, which includes acceleration and turn rate. In [8], a reinforcement learning (RL) technique is used to train a swarm of UAVs to determine the optimal routes that maximize the probability of observing the targets. Existing works on target tracking employed different technologies and methods. However, it is still an open research problem. According to [9], mobile target tracking is a challenging problem due to the uncontrollable motion of the target, making the task even more complicated.

In this work, we focus on addressing the challenge of controlling multiple UAVs to track multiple targets, with the constraints of communication and computing resources of UAVs. To this end, we present a new approach where RSSI is used, due to its low cost and power consumption, hardware simplicity, and the ability to use simple receivers. More specifically, a Q-learning-based algorithm for UAV control action selection is proposed, along with a novel reward function that encourages UAVs to learn an optimal policy for improved tracking with maximum expected cumulative reward while considering accuracy, latency, and energy consumption. The key contributions of the paper are as follows:

- 1- We present a scheme using the QL algorithm that controls multiple UAVs in 3-D environments to achieve optimal tracking of multiple targets.
- 2- We develop an efficiency-maximizing reward function that accounts for joint optimization of accuracy, delay, energy consumption, and obstacle avoidance.

The paper is organized as follows: Section II presents the system model, Section III explains the proposed scheme, Section IV analyzes the scheme through simulations, and Section V provides concluding remarks.

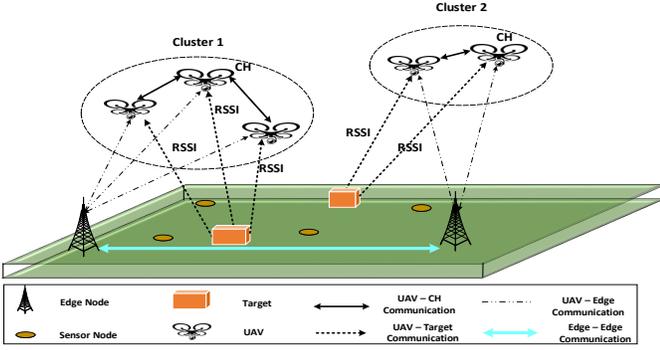


Fig. 1. Network Model.

II. SYSTEM MODEL

In this section, we discuss the target and UAV trajectory models along with the channel model between UAVs and the target, for the scenario shown in Fig. 1, which includes the targets, ENs, and UAVs equipped with RSS sensors.

A. Target Trajectory Model

In the system, there are M targets that have mobility on the ground. Each target has a start point (x_m^s, y_m^s) and endpoint (x_m^e, y_m^e) , where $m = 1, \dots, M$. Each target chooses a path between these two points for its movement by considering obstacle avoidance. The initial location of m -th Radio Frequency (RF) target is fixed at $pos_m^{tar} = [x_m^{tar}(0) = x_m^s, y_m^{tar}(0) = y_m^s]$ and the time-varying location of target is denoted as $pos_m^{tar}(t) = [x_m^{tar}(t), y_m^{tar}(t)]$ at time t . Here, the target movement velocity is defined as $v_m^{tar}(t) = [v_{x,m}^{tar}(t), v_{y,m}^{tar}(t)]$.

B. UAV Trajectory Model

In this system, there exists N UAVs in which each UAV flies at different altitudes. We assume that the initial location of the UAV at time $t = 0$ is $pos_n^{uav}(0) = [x_n^{uav}(0), y_n^{uav}(0), z_n^{uav}(0)]$, where $n = 1, \dots, N$. The time-varying location of the n -th UAV at time t is denoted as $pos_n^{uav}(t) = [x_n^{uav}(t), y_n^{uav}(t), z_n^{uav}(t)]$ and flight velocity of UAV is defined as $v_n^{uav}(t) = [v_{x,n}^{uav}(t), v_{y,n}^{uav}(t), v_{z,n}^{uav}(t)]$. Let $pos_n^{uav}(t)$ be the coordinate of the n -th UAV at time t . Hence, the sequence of points $L_n = \{pos_n^{uav}(0), \dots, pos_n^{uav}(T_n)\}$ can be used to express the trajectory of the n -th UAV where T_n is the total time that n -th UAV flies during its trajectory, which depends on the trajectory length and velocity of the UAV, and can be obtained as follows [10]:

$$T_n = \sum_{t=0}^{T-1} \frac{\|pos_n^{uav}(t+1) - pos_n^{uav}(t)\|}{v_n^{uav}(t+1)} \quad (1)$$

C. Channel Model

The received power captured by the RSS sensor mounted on the n -th UAV at time t can be mathematically expressed as [11]:

$$rssi_n^{uav}(t) = P_{TX} - PL_n(t) - \rho_n, \quad (2)$$

here, P_{TX} represents the constant transmit power of the RF target, while $PL_n(t)$ denotes the path loss between the n -th UAV and the target at time t . ρ_n is an exponential random variable with a unit mean incorporating the effect of Rayleigh fading. The RSS measurements in each UAV can be denoted by $RSSI_n = [rssi_n^{uav}(0), \dots, rssi_n^{uav}(T_n)]$.

III. DESIGN OF MULTI-TARGET TRACKING BY MULTI-UAV BASED ON Q-LEARNING AND MULTILATERATION

In this section, we outline our scheme for the multi-target tracking problem. In this work, Q-learning, normalization, and multilateration form the core of our scheme.

A. Q-Learning

The Q-learning algorithm is a value-based Reinforcement Learning (RL) technique that is specifically designed for deterministic policies. In RL algorithms, the primary goal is to identify the optimal policy π^* that maximizes the cumulative reward over the long term. During each time slot, the QL algorithm determines an action to be performed by the UAV. Upon taking an action a , the UAV receives a reward $r(s, a)$ and transitions to a new state s' . Following each decision, the Q-value of the state-action pair is updated as:

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha \left[r(s, a) + \gamma \max_{a' \in A} Q(s', a') \right] \quad (3)$$

where $\gamma \in (0, 1]$ is a discount factor that determines the importance of future rewards, and α is the learning rate that controls the extent to which new information overrides old information. The optimal policy can be learned through interactions with the environment and recording the corresponding experiences (s, a, r, s') .

B. Multilateration

Multilateration is the process of determining the unknown position coordinates of a point of interest. In target tracking, using multilateration method for locating the m -th target with position pos_m^{tar} , the distance from $r_{m,n}$ to n -th UAV with position pos_n^{uav} is given as

$$r_{m,n} = \sqrt{(x_m^{tar} - x_n^{uav})^2 + (y_m^{tar} - y_n^{uav})^2} \quad (4)$$

C. Multi Target Tracking Using a Swarm of UAVs

In this work, a swarm of UAVs was considered to track each target. Once the position of the detected target is estimated, the edge node (EN) selects a swarm of nearby UAVs to track the target. These UAVs form a cluster consisting of a Cluster Head (CH) and other UAVs that are directly and wirelessly connected to the CH. Since each UAV is limited by its battery capacity, the EN selects a UAV with the highest battery capacity as the CH. It is worth noting that the number of UAVs in each cluster should be at least two.

Since the Q-learning algorithm utilized in UAVs is a state-action algorithm, we considered some allowable control actions for UAVs that can be taken at each state. In this work,

the number of actions is equal to $\eta = 8$. These actions denote the flight direction along the x , y , and z -axis. UAVs determine the flight direction by choosing one action from discrete action space $AS = \{a_1, a_2, \dots, a_\eta\}$. We assumed that UAVs have only horizontal movement, hence, the UAV dynamics are formulated as follows:

$$pos_n^{uav}(t) = pos_n^{uav}(t-1) + \begin{bmatrix} d * \cos(\theta_i) \\ d * \sin(\theta_i) \\ z_n^{uav}(t-1) \end{bmatrix} \quad (5)$$

where d is the velocity of the target at time t , $\theta_i = i * \frac{2\pi}{|AS|}$ for $i \in [1, \eta]$.

The Q-learning algorithm considers three parameters, namely accuracy, delay, and energy, to optimize the target tracking performance of UAVs. To account for these parameters, we designed a reward function that aims to minimize energy and delay while maximizing accuracy. Thus, the reward function can be expressed as follows:

$$reward = w_1 * (1 - E^*) + w_2 * (1 - D^*) + w_3 * A^* \quad (6)$$

The weights assigned to energy, delay, and accuracy are denoted by w_1 , w_2 , and w_3 , respectively. The normalized values of consumed energy, delay, and accuracy, obtained through Min-Max normalization [12], are represented by E^* , D^* , and A^* , respectively.

In this work, the energy consumption of UAVs E is determined by the energy consumed during flight of UAV E_{flight} as follows [13]:

$$E = E_{flight} \quad (7)$$

where

$$E_{flight} = (W_{uav} \times g \times dist) + (F_p \times v_n^{uav} \times dist) \quad (8)$$

Here, E_{flight} be the total energy consumption during flight (in joules), W_{uav} be the weight of the UAV (in kilograms), $g = 9.81m/s^2$ is the acceleration due to gravity, $dist$ be the total distance traveled during the flight (in meters), F_p be the average propulsion force required to maintain flight (in newtons), v_n^{uav} be the average flight speed (in meters per second).

The delay between the target and UAV is directly proportional to the time taken for the signal to propagate between them. Hence, we can express the delay D as a function of propagation time D_{prop} as follows:

$$D = D_{prop} \quad (9)$$

where $D_{prop} = Distance/Speed$, ($Speed = 3 \times 10^8 m/s$) As the distance between the target and UAV increases, the propagation time also increases, leading to an increase in the overall delay.

To compute accuracy A , the distance between UAV and the target is considered as follows:

$$A = dist_{mn} = \|pos_n^{uav} - pos_m^{tar}\| \quad (10)$$

where $\|\cdot\|$ is the Euclidean distance.

The parameters E , D , and A are measured and then subjected to Min-Max normalization to accommodate their different ranges of values and units. The normalized values are denoted as E^* , D^* , and A^* in the output. Additionally, since accuracy is considered more important than energy consumption and delay, we assigned it a higher weight. Specifically, we set $w_1 = \frac{1}{4}$, $w_2 = \frac{1}{4}$, and $w_3 = \frac{1}{2}$.

In each state, every UAV selects the optimal action from a set of η possible actions (i.e., flight directions) using the Q-learning algorithm and leveraging the reward function. To achieve the final objective, avoiding obstacle collisions, we assign a reward value of $reward = 0$ to each action where the probability of obstacle collision is high. Then, UAV measures the RSSI from the power level of a received signal of the target m . The RSSI $rss_i^{uav}(t)$ measured by UAV n as well as current position $pos_n^{uav}(t)$ of UAV available in the cluster will be sent to the CH. Next, CH executes the Multilateration function and estimates the position of the target m . Finally, CH sends the estimated position to all UAVs in the cluster. Once UAVs receive the position of the target, UAVs run the Q-learning algorithm for selecting the next state. This process will be repeated until the target (e.g. m) reaches the endpoint (e.g. (x_m^e, y_m^e)). Fig. 2 represents the process of target position estimation as well as target tracking by our scheme. This figure also shows the process of data communication between a UAV and a CH.

As mentioned above, each UAV is limited by its battery capacity. The energy consumption of each UAV is affected by several factors such as weight, aerodynamics, flight speed and altitude, and environmental conditions. Computation, communication, and task complexity also contribute to power consumption. The UAV's onboard computing system includes a processor, memory, and other components that consume energy. The processing load is primarily determined by the task complexity, dataset size, and algorithm used. Communication between UAVs and ground stations requires the use of communication systems, such as radios or transceivers, which also consume energy. The energy consumption of the communication system depends on the amount of data transmitted or received, the distance, and the quality of the communication link. Higher data rates or longer distances typically require higher transmit powers, leading to higher energy consumption.

To address this issue, we established two threshold values for the battery power of UAVs. These thresholds are utilized to monitor the state of the battery during the UAV's flight. By setting these threshold values, we can ensure that the UAV operates within a predetermined energy budget, which not only prolongs its flight time and range but also enhances its reliability and lowers the likelihood of battery depletion during a mission. Whenever the battery power of a UAV, such as UAV_n , falls below the first threshold value, it sends a warning message to the nearby EN to report its status. It also stops measuring RSSI and sends a request message to CH asking for the target's position until its battery power is sufficient for target tracking. Concurrently, the EN attempts

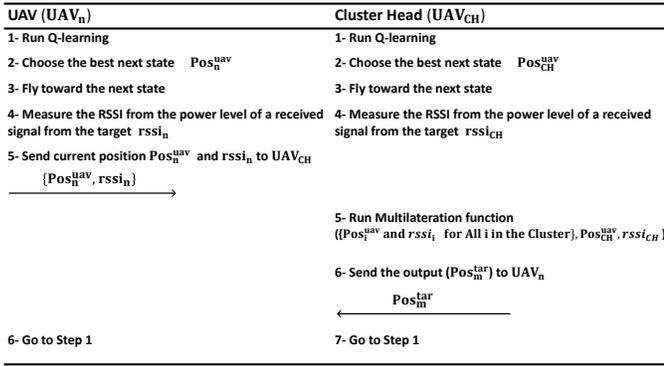


Fig. 2. The process of data communication between UAV and CH.

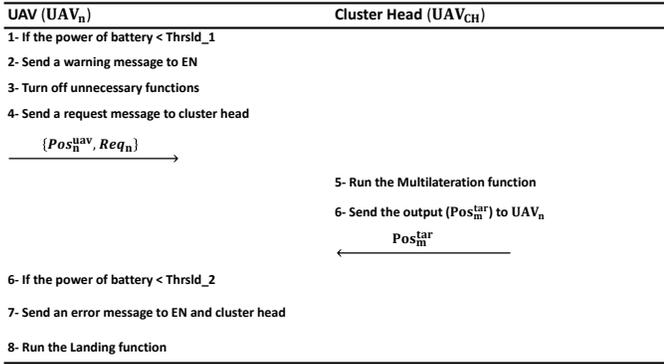


Fig. 3. The process of UAV battery power monitoring.

to find a replacement UAV to swap with UAV_n . If UAV_n 's battery power falls below the second threshold value, it sends an error message to the nearby EN and CH, and then initiates the landing function. This process is illustrated in Fig. 3.

IV. NUMERICAL RESULTS

This section presents the numerical results of our scheme, specifically tracking accuracy and energy consumption. MATLAB was used as the simulation platform, and an obstacle-filled environment was created using a matrix with cylinders and cones representing the obstacles. The simulation involved five UAVs tracking two targets in this environment. Here, the tracking of target 1 is performed by three UAVs, namely $\{UAV_1, UAV_2, UAV_3\}$, while target 2 is tracked by two UAVs, namely $\{UAV_4, UAV_5\}$. We also included three edge nodes (ENs) in the simulation. Each UAV is capable of communicating with an EN that is within its communication range. In order to assess the effectiveness of our scheme, we established three separate scenarios, outlined as follows:

- **Cluster 1:** In this scenario, three UAVs are organized into a cluster, and a single UAV is designated as the cluster head (CH). The two remaining UAVs communicate with the CH and nearby EN and do not directly communicate with each other.
- **Cluster 2:** In this scenario, two UAVs are grouped into a cluster, and one UAV is elected as a cluster head (CH).

Another UAV is able to communicate with CH and nearby ENs.

- **Non-Clustered:** In this scenario, there is no clustering of UAVs. Instead, three individual UAVs are assigned to track a target and are able to communicate with each other as well as nearby ENs.

The initial positions of each UAV and target were defined as previously explained. Target 1 has a starting position of $pos_1^{tar}(0) = [2, 1]$ meters, while target 2 has a starting position of $pos_2^{tar}(0) = [1, 6]$ meters. Both targets have a designated endpoint of $[30, 15]$ meters. To move toward the endpoint while avoiding obstacles, each target randomly selects a path between its start point and the endpoint. Additionally, we have defined the initial positions of five UAVs as $pos_1^{uav}(0) = [1, 2, 2.5]$, $pos_2^{uav}(0) = [3, 4.5, 3]$, $pos_3^{uav}(0) = [6, 1, 2]$, $pos_4^{uav}(0) = [4.5, 6, 3]$, and $pos_5^{uav}(0) = [2, 10, 4]$ meters. Each target is initially assigned a velocity, and their velocities can vary from $1m/s$ to $5m/s$ during their movement along the trajectory. The UAVs adjust their velocity during target tracking based on the velocity of the target. Here, we assume that each UAV will receive information about obstacles from nearby ENs to avoid the collision. The information includes the dimensions of the obstacles such as length, width, height, diameter, and other relevant details.

The root means square error (RMSE) can be an effective metric for assessing the accuracy of the scheme's performance [14]. To this end, we consider the actual position of the target and the estimated position of the target by the UAVs. For all positions that the target passed during its trajectory, we measured the RMSE. We carried out this procedure for each of the aforementioned scenarios individually. The RMSE was computed using the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^K (x_i^{tar} - \hat{x}_i^{tar})^2 + (y_i^{tar} - \hat{y}_i^{tar})^2}{K}} \quad (11)$$

where K is the number of positions that the target passed during its trajectory, and $[x_i^{tar}, y_i^{tar}]$ and $[\hat{x}_i^{tar}, \hat{y}_i^{tar}]$ represent the actual and estimated positions of the target at the i -th position, respectively. Fig. 4 presents a comparison of the RMSE for each scenario. It is observed that the accuracy of the proposed scheme in scenario 1 is superior to those in other scenarios. This can be attributed to the higher number of UAVs present in scenario 1, as compared to scenario 2 in cluster-based scenarios. In real-time applications like target-tracking, both the computation and communication delay have a significant impact on application performance accuracy. The reduced number of connections and communications for sending/receiving information between UAVs in scenario 1 compared to scenario 3 leads to higher accuracy in the former.

We conducted experiments to measure the total energy consumption by each UAV during the target tracking process. The results, as shown in Fig. 5, indicate that the energy consumed by UAVs in the cluster-based scenario is less than that in the non-clustered scenario. This is due to the reduced

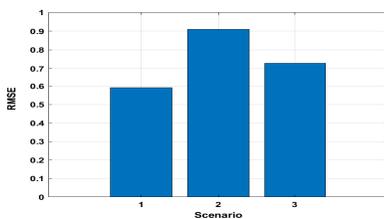


Fig. 4. Comparison of measured RMSE in each scenario

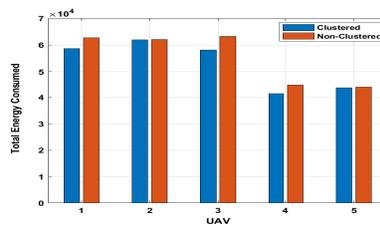


Fig. 5. Comparison of total energy consumption by each UAV in clustered and non-clustered scenarios

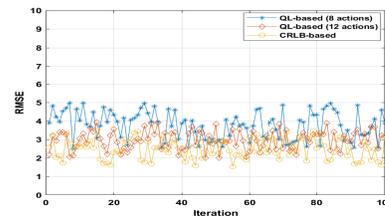


Fig. 6. Comparison of our scheme with CLRB-based scheme over 100 Monte Carlo experiments.

communication and computation requirements in the cluster-based scenario.

In addition, we conduct a comparative analysis of our scheme with a Cramér–Rao Lower Bound (CRLB) based scheme proposed in [15] over 100 Monte Carlo experiments. The CRLB is a fundamental concept used in target tracking to estimate the accuracy of any unbiased estimator. It serves as a benchmark for assessing the quality of target tracking algorithms. The comparison focuses on assessing the performance in terms of Root Mean Squared Error (RMSE). For our scheme, we evaluate its performance under the first scenario with different numbers of allowable control actions ($\eta = 8$ and 12). As depicted in Fig. 6, the QL-based control demonstrates tracking performance comparable to the CRLB-based control, which is considered the optimal control scheme.

V. CONCLUSION

In this study, a scheme based on RSSI has been proposed for tracking multiple targets using multiple UAVs. The QL algorithm and Multilateration are the core of the proposed scheme. Due to the limitation of power capacity and the computing capacity of UAVs and in addition, the importance of delay in the target tracking, energy consumption, delay, and accuracy have been considered as three main parameters in the reward function of the QL algorithm. We have analyzed our scheme in cluster-based and non-cluster-based scenarios. The obtained results showed that our scheme based on clustering has provided a more accurate and efficient target-tracking solution with lower energy.

ACKNOWLEDGMENT

This research was sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing and was accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government, or the U.K. Government. The U.S. Government and U.K. Government are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

- [1] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, “Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones,” *Ieee vehicular technology magazine*, vol. 12, no. 3, pp. 73–82, 2017.
- [2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.
- [3] S. Goudarzi, M. H. Anisi, H. Ahmadi, and L. Musavian, “Dynamic resource allocation model for distribution operations using sdn,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 976–988, 2020.
- [4] S. Goudarzi, S. A. Soleymani, W. Wang, and P. Xiao, “Uav-enabled mobile edge computing for resource allocation using cooperative evolutionary computation,” *IEEE Transactions on Aerospace and Electronic Systems*, 2023.
- [5] J. Wang, K. Liu, and J. Pan, “Online uav-mounted edge server dispatching for mobile-to-mobile edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1375–1386, 2019.
- [6] S. Bhagat and P. Sujit, “Uav target tracking in urban environments using deep reinforcement learning,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 694–701.
- [7] L. Wang, Y. Li, H. Zhu, and L. Shen, “Target state estimation and prediction based standoff tracking of ground moving target using a fixed-wing uav,” in *IEEE ICCA 2010*. IEEE, 2010, pp. 273–278.
- [8] T. Wang, R. Qin, Y. Chen, H. Snoussi, and C. Choi, “A reinforcement learning approach for uav target searching and tracking,” *Multimedia Tools and Applications*, vol. 78, pp. 4347–4364, 2019.
- [9] X. Deng, J. Li, P. Guan, and L. Zhang, “Energy-efficient uav-aided target tracking systems based on edge computing,” *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2207–2214, 2021.
- [10] Y.-J. Chen, D.-K. Chang, and C. Zhang, “Autonomous tracking using a swarm of uavs: A constrained multi-agent reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 702–13 717, 2020.
- [11] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, “A location estimation algorithm based on rssi vector similarity degree,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371350, 2014.
- [12] M. F. Aslan, A. Durdu, and K. Sabanci, “Visual-inertial image-odometry network (viinet): A gaussian process regression-based deep architecture proposal for uav pose estimation,” *Measurement*, vol. 194, p. 111030, 2022.
- [13] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, “Joint computation and communication design for uav-assisted mobile edge computing in iot,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5505–5516, 2019.
- [14] S. Goudarzi, S. Ahmad Soleymani, M. H. Anisi, D. Ciuonzo, N. Kama, S. Abdullah, M. Abdollahi Azgomi, Z. Chaczko, and A. Azmi, “Real-time and intelligent flood forecasting using uav-assisted wireless sensor network,” *Computers, Materials and Continua*, vol. 70, no. 1, pp. 715–738, 2021.
- [15] S. Papaioannou, S. Kim, C. Laoudias, P. Kolios, S. Kim, T. Theocharides, C. Panayiotou, and M. Polycarpou, “Coordinated crlb-based control for tracking multiple first responders in 3d environments,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1475–1484.

Generalised Sequential Matrix Diagonalisation for the SVD of Polynomial Matrices

Faizan A. Khattak, Ian K. Proudler, John G. McWhirter, and Stephan Weiss

Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, Scotland

{faizan.khattak,ian.proudler,stephan.weiss}@strath.ac.uk

Abstract—To extend the singular value decomposition (SVD) to matrices of polynomials, an existing algorithm — a polynomial version of the Kogbetliantz SVD — iteratively targets the largest off-diagonal elements, and eliminates these through delay and Givens operations. In this paper, we perform a complete diagonalisation of the matrix component that contains this maximum element, thereby transferring more off-diagonal energy per iteration step. This approach is motivated by — and represents a generalisation of — the sequential matrix diagonalisation (SMD) method for parahermitian matrices. In simulations, we demonstrate the benefit of this generalised SMD over the Kogbetliantz approach, both in terms of diagonalisation and the order of the extracted factors.

I. INTRODUCTION

The singular value decomposition (SVD) is a standard linear algebraic tool for the diagonalisation of a rectangular matrix [1]. It has proven central in signal processing to provide solutions to many different challenges [2]. Often solutions can be optimal in various respects; for example for such as for precoding and equalisation for the diagonalisation of multiple-input multiple-output (MIMO) channels [3], where the SVD leads to optimality in least squares and channel capacity senses. Such matrices typically describe narrowband systems.

In the broadband case, where impulse responses rather than complex-valued gain factors between sources and sensors have to be considered, a multiple-input multiple-output system created by N transmitters and M receivers becomes a matrix of transfer functions. For example, the system $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M \times N}$,

$$\mathbf{A}(z) = \begin{bmatrix} a_{1,1}(z) & \dots & a_{1,N}(z) \\ \vdots & \ddots & \vdots \\ a_{M,1}(z) & \dots & a_{M,N}(z) \end{bmatrix}, \quad (1)$$

contains in its m th row and n th column the z -transform $a_{m,n}(z) = \sum_{\tau} a_{m,n}[\tau]z^{-\tau}$, or for short $a_{m,n}(z) \bullet\circ a_{m,n}[\tau]$, where $a_{m,n}[\tau]$ is the impulse response between the n th transmitter and the m th receiver. If these impulse responses are finite and causal, then $\mathbf{A}(z)$ in (1) is a polynomial matrix. For such matrices, the standard SVD can diagonalise (1) for only one specific value of z , or equivalently $\mathbf{A}[\tau]$ for one one value of τ .

The work of Faizan Khattak is supported by a Commonwealth Scholarship. This work was also in part supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1 and the MOD University Defence Research Collaboration in Signal Processing.

Therefore a different SVD factorisation is required for (1), that can simultaneously diagonalise $\mathbf{A}(z)$ for all z , or equivalently $\mathbf{A}[\tau]$ for all τ , such that

$$\mathbf{A}(z) \approx \mathbf{U}(z)\mathbf{\Sigma}(z)\mathbf{V}^P(z). \quad (2)$$

The approximation sign is due to the potential truncation of infinite series and other effects that we will briefly review in Sec. II. Decompositions such as (2) have in the past been realised via two polynomial eigenvalue decompositions (PEVDs) applied to two parahermitian matrices $\mathbf{A}(z)\mathbf{A}^P(z)$ and $\mathbf{A}^P(z)\mathbf{A}(z)$ [6]. The parahermitian transposition $\mathbf{A}^P(z) = \{\mathbf{A}(1/z^*)\}^H$ involves a Hermitian transposition and time reversal; if a matrix $\mathbf{R}(z)$ satisfies $\mathbf{R}^P(z) = \mathbf{R}(z)$, it is also termed a parahermitian matrix, for which a number of eigenvalue factorisations have been reported [4], [5], [6], [7], [8], [9], [10], [11]. To avoid the route via two PEVDs, a polynomial QR decomposition has been exploited in [12]. Further, a direct polynomial SVD has been created by a Kogbetliantz-type approach to the SVD [13]. Such algorithms can enable a number of applications ranging from e.g. MIMO communications [14], [15], beamforming [16], to filter bank design and paraunitary matrix completion [17].

The Kogbetliantz method in [13] is a powerful approach that generally yields better diagonalisation and lower order factors than those achieved via two PEVDs. The approach is a generalisation of the second order sequential best rotation (SBR2) algorithm, which calculates the PEVD of parahermitian matrices [6], [8]. It is an iterative algorithm, that in every step eliminates the largest off-diagonal element by transferring its energy onto the diagonal. For the EVD of parahermitian matrices, SBR2 performs a similarity transform, where an elementary paraunitary matrix and its parahermitian transpose are left- and right-multiplied against the result from the previous iteration. In the Kogbetliantz approach in [13], this approach is modified to permit different paraunitary matrices for the left- and right-multiplications. In this paper, we want to explore whether performance improvements can be attained by borrowing ideas from a sequential matrix diagonalisation (SMD) algorithm for parahermitian matrices. In SMD, more energy is transferred per iteration, whereby for the PEVD of parahermitian matrix significant advantages have been reported [18], [19].

Therefore, below we review some aspects of a polynomial SVD in Sec. II before introducing the proposed generalised

SMD algorithm for a direct polynomial SVD in Sec. III. Its performance is explored in Sec. IV.

II. POLYNOMIAL SVD

In this section, we highlight the properties of a polynomial SVD as in (2), followed by a brief review of the Kogbetliantz approach in [13].

A. Analytic SVD

For an analytic $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M \times N}$ that is not tied to a multiplexing operation, there exists an analytic, diagonal $\Sigma(z)$ that is real-valued on the unit circle, as well as analytic left- and right-singular vectors as in the columns of $\mathbf{U}(z)$ and $\mathbf{V}(z)$ that are unique up to common allpass filters [20], [21]. However, the singular values in $\Sigma(z)$, when evaluated on the unit circle, must be permitted to intersect and even take on negative values — an observation previously made also for the case of matrices in a continuous time parameter [22], [23]. In this case, (2) holds with equality.

Time domain polynomial matrix factorisation methods such as [6], [12], [13], [8], [19], [24] typically encourage spectrally majorised eigen- or singular values, such that for $\Sigma(z) = \text{diag}\{\sigma_1(z), \dots, \sigma_K(z)\}$ with $K = \min M, N$,

$$\sigma_k(e^{j\Omega}) \geq \sigma_{k+1}(e^{j\Omega}) \quad \forall \Omega, \quad k = 1, \dots, (K-1) \quad (3)$$

is satisfied, i.e. they must not intersect. In case of the polynomial EVD, the SBR2 algorithm in [6], [8] has been explicitly proven to yield a spectrally majorised result [25]. Thus, algorithmic solutions may deviate from the analytic solution, such that, either due to falsely assumed spectral majorisation or due to the approximation of infinite order factors by Laurent polynomials, (2) does not hold with equality. In practice however, due to estimation errors, singular values will be spectrally majorised with probability one [26], such that only approximation errors impact on the precision of (2).

B. Polynomial Kogbetliantz Algorithm

The polynomial Kogbetliantz approach in [13] is a generalisation of the SBR2 algorithm, that extends the application of the latter from parahermitian to general matrices. The algorithm starts with the initialisation $\mathbf{S}^{(0)}(z) = \mathbf{A}(z)$. This initialisation may involve a unitary phase correction to the entire matrix, such that for $\mathbf{S}^{(0)}[\tau] \circ \bullet \mathbf{S}^{(0)}(z)$, the coefficient matrix of order zero, $\mathbf{S}^{(0)}[0]$, is real valued. We will refer to this coefficient matrix for $\tau = 0$ of a polynomial matrix as the ‘zero plane’ below.

At the i th iteration, the Kogbetliantz approach generates an enhanced diagonalised $\mathbf{S}^{(i)}(z)$ in two steps. Firstly, the maximum off-diagonal component is brought to the zero-plane by delay operations. Secondly, a Givens rotation transfers the energy of this component onto the diagonal. For the first step,

$$\{m_i, n_i, \tau_i\} = \arg \max_{\substack{m, n, \tau \\ j \neq k}} |s_{m, n}^{(i-1)}[\tau]| \quad (4)$$

determines the location of the maximum off-diagonal element, where $s_{m, n}^{(i-1)}[\tau]$ is the element in the m th row and n th

column of $\mathbf{S}^{(i-1)}[\tau]$. With the identified parameters, two delay matrices

$$\mathbf{B}_r^{(i)}(z) = \text{blockdiag}\{\mathbf{I}_{n_i}, z^{-\tau_i}, \mathbf{I}_{M-n_i}\} \quad (5)$$

$$\mathbf{B}_l^{(i)}(z) = \text{blockdiag}\{\mathbf{I}_{n_i}, z^{\tau_i}, \mathbf{I}_{N-n_i}\} \quad (6)$$

are formed. Thus,

$$\mathbf{S}^{(i-\frac{1}{2})}(z) = \mathbf{B}_l^{(i)}(z) \mathbf{S}^{(i-1)}(z) \mathbf{B}_r^{(i)}(z) \quad (7)$$

contains in its zero plane the same diagonal elements as $\mathbf{S}^{(i-1)}(z)$, but the maximum off-diagonal component and also some other elements in the n_i th row and n_i th column have been transferred such that $s_{m_i, n_i}^{(i-\frac{1}{2})}[0] = s_{m_i, n_i}^{(i-1)}[\tau_i]$. This is accomplished by $\mathbf{B}_r^{(i)}(z)$ delaying the n_i th column by τ_i samples, while $\mathbf{B}_l^{(i)}(z)$ advances the n_i th row.

For the second step, a Givens rotation via matrices $\mathbf{G}_r^{(i)}$ and $\mathbf{G}_l^{(i)}$ transfers the the energy of $s_{m_i, n_i}^{(i-\frac{1}{2})}[0]$ onto the diagonal, such that

$$\mathbf{S}^{(i)}(z) = \mathbf{G}_l^{(i)} \mathbf{S}^{(i-\frac{1}{2})}(z) \mathbf{G}_r^{(i)} \quad (8)$$

Since this operation is applied across the entire matrix and not just to the zero plane, it may undo some of the efforts in previous iterations. Overall, since the energy in the diagonal monotonously increases while the overall energy in $\mathbf{S}^{(i)}(z)$ remains unaltered from $\mathbf{S}^{(i-1)}(z)$, the algorithm can be proven to converge [13]. The iterative process is terminated when either the maximum off-diagonal element falls below a given threshold or a specified maximum number of iterations has been reached. Then after K iterations, for the singular values, $\Sigma(z)$ is extracted as the diagonal of $\mathbf{S}^{(K)}(z)$, and

$$\mathbf{U}(z) = \prod_{i=1}^K \left\{ \mathbf{B}_l^{(i)}(z) \right\}^P \left\{ \mathbf{G}_l^{(i)} \right\}^H \quad (9)$$

$$\mathbf{V}(z) = \prod_{k=0}^{K-1} \mathbf{B}_r^{(K-k)}(z) \mathbf{G}_r^{(K-k)} \quad (10)$$

will provide the left- and right-singular vectors. In order to limit the polynomial order of the extracted factors, it may be advantageous to apply trimming during the iterative process, or after the algorithm has terminated [27], [6], [28], [24], [29].

III. GENERALIZED SEQUENTIAL MATRIX DIAGONALIZATION

The Kogbetliantz approach in Sec. II-B represents a generalisation of SBR2; it eliminates one largest off-diagonal element per iteration. The cost is moderate, but the polynomial order of the result grows with every iteration. Therefore, in this section, we generalise the SMD algorithm, which in general is capable of transferring more energy per iteration than SBR2. This generalised SMD algorithm is outlined in this section.

A. Initialisation

The proposed algorithm starts with $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M \times N}$. Without loss of generality, we assume $M \geq N$, as otherwise

we can operate with $\mathbf{A}^P(z)$ instead. We first perform a diagonalisation of its zero plane matrix $\mathbf{A}[0]$ via an SVD,

$$\mathbf{A}[0] = \mathbf{U}^{(0)}\mathbf{S}^{(0)}[0]\mathbf{V}^{(0),H}. \quad (11)$$

Based on this factorisation, the initial step for the algorithm is

$$\mathbf{S}^{(0)}(z) = \mathbf{U}^{(0),H}\mathbf{A}(z)\mathbf{V}^{(0)}. \quad (12)$$

Note that $\mathbf{U}^{(0)}$ and $\mathbf{V}^{(0)}$ are applied to the entirety of $\mathbf{A}(z)$, but will ensure that the zero plane matrix $\mathbf{S}^{(0)}[0]$ is indeed diagonal and real-valued. We also record initial estimates for the left- and right-singular vectors as $\mathbf{U}^{(0)}(z) = \mathbf{U}^{(0)}$ and $\mathbf{V}^{(0)}(z) = \mathbf{V}^{(0)}$.

B. Iterative Procedure

Following the initialisation in (12), any subsequent iterations $i = 1, 2, \dots$ repeat the three steps below. Firstly, in the i th iteration, we transfer the n_i th column and the n_i th row of $\mathbf{A}^{(i-1)}(z)$ to the zero plane. This step exploits the matrices $\mathbf{B}_r^{(i)}(z)$ and $\mathbf{B}_l^{(i)}(z)$ defined in (5) and (6), and generates a shifted version $\mathbf{S}^{(i-\frac{1}{2})}(z)$ according to (7). The particular row and delay selection for this step can differ from (4), and we define a general column norm that excludes any diagonal elements, such that

$$\|\hat{\mathbf{s}}_n^{(i-1)}[\tau]\|_p = \left\{ \sum_{m=1, m \neq n}^M |s_{m,n}^{(i-1)}[\tau]|^p \right\}^{\frac{1}{p}}. \quad (13)$$

For $p \rightarrow \infty$, the norm picks the maximum element, and the selection is identical to (4). The resulting algorithm version is termed the maximum element generalised SMD (ME-GSMD). Since we ultimately want to perform a complete diagonalisation of the zero plane matrix, it appears promising to shift more energy to the zero plane than with ME-GSMD. This can be accomplished for $p = 2$, and we term the resulting procedure the GSMD algorithm.

Secondly, we now diagonalise the zero plane of the shifted matrix $\mathbf{S}^{(i-\frac{1}{2})}(z) = \mathbf{B}_l^{(i)}(z)\mathbf{S}^{(i-1)}(z)\mathbf{B}_r^{(i)}(z)$. By computing an SVD of its zero plane matrix $\mathbf{S}^{(i-\frac{1}{2})}[0] = \mathbf{U}^{(i)}\mathbf{D}^{(i)}\mathbf{V}^{(i),H}$, we determine

$$\mathbf{S}^{(i)}(z) = \mathbf{U}^{(i),H}\mathbf{S}^{(i-\frac{1}{2})}(z)\mathbf{V}^{(i)}. \quad (14)$$

This operation diagonalises the zero plane matrix, but also modifies all other entries in $\mathbf{S}^{(i-\frac{1}{2})}(z)$.

Thirdly, we update the left- and right-singular values as

$$\mathbf{U}^{(i)}(z) = \mathbf{U}^{(i-1)}(z)\mathbf{B}_l^{(i),P}(z)\mathbf{U}^{(i)} \quad (15)$$

$$\mathbf{V}^{(i)}(z) = \mathbf{V}^{(i-1)}(z)\mathbf{B}_r^{(i)}(z)\mathbf{V}^{(i)}, \quad (16)$$

based on the previous estimates, the delay matrices, and the unitary matrices obtained from the application of an SVD in (14).

At each iteration, $\mathbf{S}^{(i)}(z)$ grows in order by $2|\tau_i|$, and $\mathbf{U}^{(i)}(z)$, and $\mathbf{V}^{(i)}(z)$ each grow in order by $|\tau_i|$. It may therefore be opportune to apply trimming [27], [6], [28], [24], [29] at each iteration step, thus stemming the order growth

and somewhat arresting the computational complexity of the algorithm.

C. Convergence and Termination

It can be shown that with each iteration step, the overall energy within $\mathbf{A}(z)$ remains unaltered while the energy on the diagonal monotonously increases. However, such a proof is beyond the scope of this paper; for the special case of $\mathbf{A}(z)$ being a parahermitian matrix, the GSMD algorithm reduces to the SMD algorithm, for which an explicit convergence proof is reported in [19].

The iterations continue until either a sufficiently low threshold for the off-diagonal elements defined via (13) is attained, or until a predefined maximum number of iterations is reached. Thus, after L iterations, we can extract the approximate polynomial SVD factors of (2) as

$$\hat{\mathbf{U}}(z) = \mathbf{U}^{(L)}(z), \quad \hat{\mathbf{V}}(z) = \mathbf{V}^{(L)}(z), \quad (17)$$

while

$$\hat{\mathbf{\Sigma}}(z) = \mathbf{S}^{(L)}(z). \quad (18)$$

Note that $\hat{\mathbf{\Sigma}}(z)$ may still contain some non-zero off-diagonal components albeit of small magnitude.

IV. SIMULATIONS AND RESULTS

To compare the two proposed approaches, ME-GSMD and GSMD, against the polynomial Kogbetliantz approach in [13] — a generalisation of the SBR2 algorithm [6], [8] — this section presents some simulation results.

A. Performance Metrics

In order to assess the performance of the various polynomial SVD algorithms, we utilise the diagonalisation η ,

$$\eta = \frac{\sum_{\tau} \|\bar{\mathbf{\Sigma}}[\tau]\|_{\mathbb{F}}^2}{\sum_{\tau} \|\hat{\mathbf{\Sigma}}[\tau]\|_{\mathbb{F}}^2}, \quad (19)$$

where $\bar{\mathbf{\Sigma}}[\tau]$ is same as $\hat{\mathbf{\Sigma}}[\tau]$ but with its off-diagonal elements set to zero. For a completely diagonalised $\hat{\mathbf{\Sigma}}[\tau]$, this metric η would be unity. To assess the algorithmic performance, we also measure the execution time of the algorithms.

For gauging the computational complexity of a practical implementation in general communications or signal processing application, the order of the achieved polynomial SVD factors is important. Therefore we assess the orders of both right and left singular vectors, denoted by $\mathcal{O}\{\hat{\mathbf{U}}(z)\}$ and $\mathcal{O}\{\hat{\mathbf{V}}(z)\}$ respectively.

B. Numerical Example

Before conducting extensive simulations, we demonstrate the performance of the proposed method using a numerical example. We create a 4×3 polynomial matrix $\mathbf{A}(z)$ of order 2, with coefficients drawn from a normal distribution with zero mean and unit variance. The generated matrix is characterised in Fig. 1.

Prior to executing the algorithm, the diagonalisation ratio η is 0.2452. Running GSMD, we achieve a diagonalisation ratio

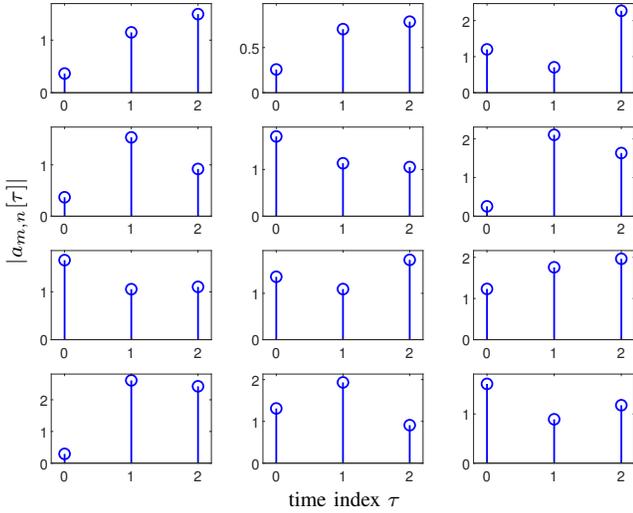


Fig. 1. Matrix $\mathbf{A}[\tau] \circ \bullet \mathbf{A}(z)$ for numerical example, showing the moduli of its elements, $|a_{m,n}[\tau]|$, $m = 1, \dots, 4$ and $n = 1, 2, 3$.

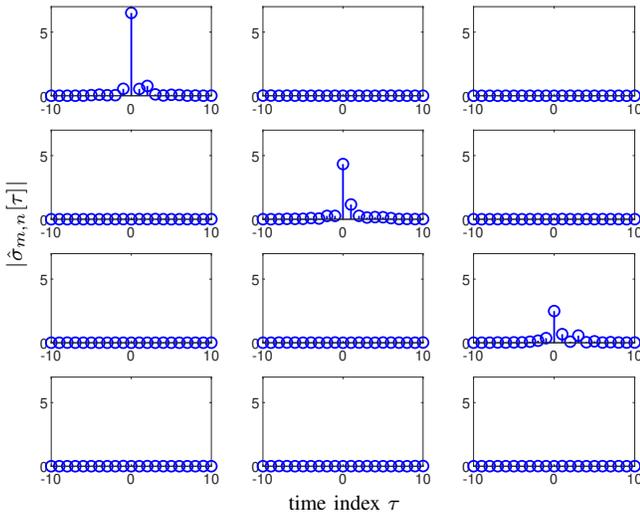


Fig. 2. Approximately diagonalised matrix $\hat{\Sigma}[\tau]$ derived from $\mathbf{A}[\tau]$ in Fig. 1 using the GSMD algorithm with $L = 100$ iterations.

of $\eta = 0.9998$ after 100 iterations. The resulting diagonalised matrix is shown in Fig. 2 where only 10 central lags are displayed, and trailing values close to zero are suppressed. As a comparison, the generalised SBR2 approach that forms the polynomial Kogbetliantz method in [13] requires 205 iterations to reach a similar diagonalisation ratio as GSMD.

C. Ensemble Results

We construct an ensemble of 500 random instantiations of $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{5 \times 3}$ of order $\mathcal{O}\{\mathbf{A}(z)\} = 2$, whose coefficients are drawn from a normal distribution with zero mean and unit variance. We compare the proposed two algorithms — GSMD and ME-GSMD — against the polynomial Kogbetliantz approach, which implements a generalised SBR2 (GSBR2) algorithm. The various results below represent averages across the ensemble.

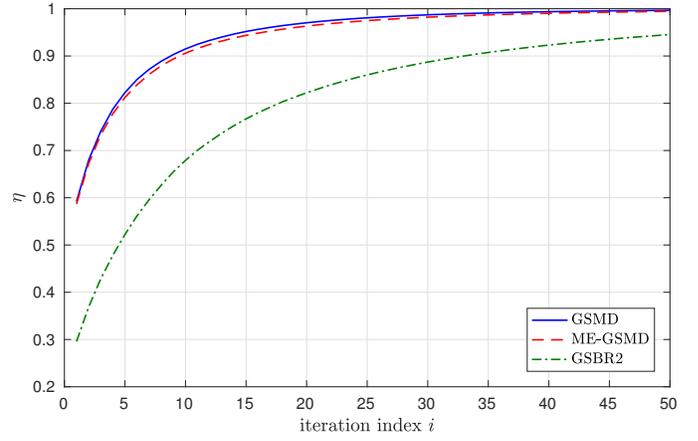


Fig. 3. Ensemble average of η versus iteration number.

First, we compare the diagonalisation performance η of the algorithms over a run of $L = 50$ iteration. For this test, the order of the polynomial SVD factors are of a lesser importance, and in order to ensure that algorithms perform the desired number of iterations, the trimming threshold is set to zero. The average diagonalisation is shown in Fig. 3. All algorithms converge towards $\eta = 1$ as the number of iterations L increases. The GSMD algorithm, utilising the L_2 norm in (13) and hence transferring the column with maximum power in each iteration, provides a slightly faster convergence than the ME-GSMD, which looks for the maximum off-diagonal element via the L_∞ norm in (13). Both of these proposed methods converge significantly faster than the benchmark, polynomial Kogbetliantz approach in [13], here referred to as GSBR2.

The execution times over 50 iterations are $0.042 \pm 0.08s$ for GSMD, 0.044 ± 0.009 for ME-GSMD, and 0.04 ± 0.01 for GSBR2. Hence, the algorithms exhibit very similar overall complexities, and Fig. 3 can also be taken as a rough indication of what a comparison of diagonalisation versus execution time would provide.

The order growth of the paraunitary matrices $\hat{U}(z)$ and $\hat{V}(z)$ is illustrated in Figs. 4 and 5, displaying the achievable diagonalisation η versus the orders of the left-singular vectors, $\mathcal{O}\{\hat{U}(z)\}$, in Fig. 4 and the orders of the right-singular vectors, $\mathcal{O}\{\hat{V}(z)\}$, in Fig. 5. The results demonstrate that GSMD and ME-GSMD yield paraunitary matrices that can be more economically applied in order to achieve good diagonalisation, with an advantage for GSMD over ME-GSMD. The former tends to transfer more energy per iteration, and since the order of the polynomial SVD factors are likely to grow with each iteration, the smaller number of iterations indicated in Fig. 3 also translate into lower orders in Figs. 4 and 5. Both methods yield significantly lower order paraunitary matrices compared to the benchmark, GSBR2. Note that this difference in performance is more pronounced for the left-singular vectors in Fig. 4 with $\hat{U}(z) : \mathbb{C} \rightarrow \mathbb{C}^{5 \times 5}$ compared to the shorter right-singular vectors in the smaller $\hat{V}(z) : \mathbb{C} \rightarrow \mathbb{C}^{3 \times 3}$ in Fig. 5.

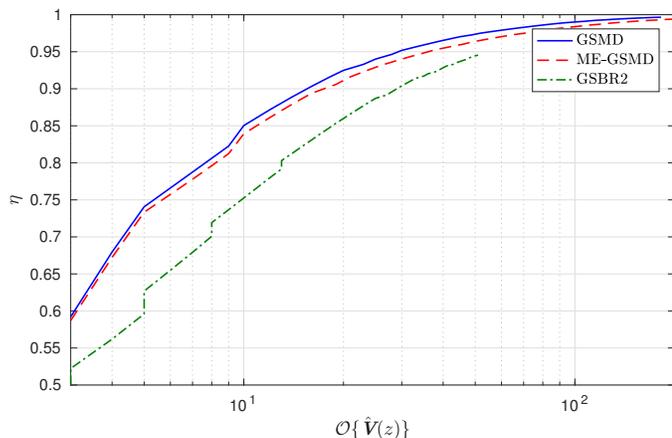


Fig. 4. Ensemble average of η versus ensemble median of order of $\hat{U}(z)$.

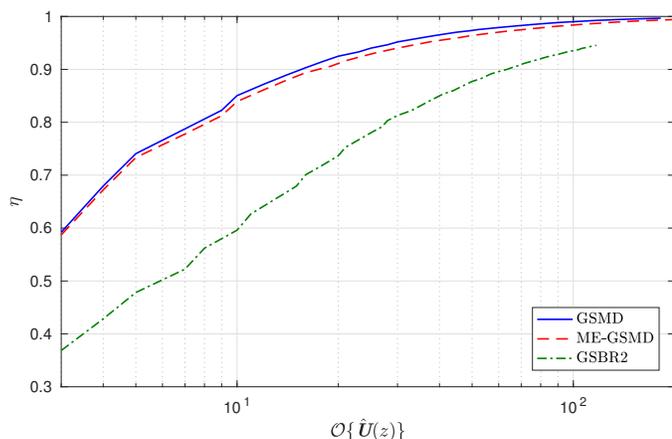


Fig. 5. Ensemble average of η versus ensemble median of order of $\hat{V}(z)$.

V. CONCLUSION

In this paper, we have proposed a generalised sequential matrix diagonalization algorithm for the SVD of polynomial matrices. This is an extension of the SMD algorithm [18], [19], which is applicable to parahermitian matrices, to the more general case of rectangular matrices of transfer functions. This generalisation is akin to the way a benchmark algorithm, the polynomial Kogbetliantz approach in [13] extends the SRB2 algorithm [6], [8]. Ensemble simulations show that GSMD can achieve better diagonalisation with lower order polynomial matrices compared to this benchmark.

REFERENCES

- [1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.
- [2] M. Moonen and B. de Moor, Eds., *SVD and Signal Processing, III: Algorithms, Architectures and Applications*. Elsevier, 1995.
- [3] M. Vu and A. Paulraj, "MIMO wireless linear precoding," *IEEE Signal Processing Magazine*, **24**(5):86–105, Sep. 2007.
- [4] A. Tkacenko and P. Vaidyanathan, "On the spectral factor ambiguity of FIR energy compaction filter banks," *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 380–385, Jan. 2006.
- [5] —, "Iterative greedy algorithm for solving the FIR paraunitary approximation problem," *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 146–160, Jan. 2006.
- [6] J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif, and J. Foster, "An EVD algorithm for para-Hermitian polynomial matrices," *IEEE Trans. Signal Processing*, **55**(5):2158–2169, May 2007.
- [7] A. Tkacenko, "Approximate eigenvalue decomposition of para-hermitian systems through successive FIR paraunitary transformations," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, TX, Mar. 2010, pp. 4074–4077.
- [8] S. Redif, J. McWhirter, and S. Weiss, "Design of FIR paraunitary filter banks for subband coding using a polynomial eigenvalue decomposition," *IEEE Trans. Signal Processing*, **59**(11):5253–5264, Nov. 2011.
- [9] M. Tohidian, H. Amindavar, and A. M. Reza, "A DFT-based approximate eigenvalue and singular value decomposition of polynomial matrices," *EURASIP J. Advances Sig. Processing*, **2013**(1):1–16, 2013.
- [10] V. W. Neo and P. A. Naylor, "Second order sequential best rotation algorithm with householder reduction for polynomial matrix eigenvalue decomposition," in *IEEE ICASSP*, Brighton, UK, pp. 8043–8047, May 2019.
- [11] S. Weiss, I. K. Proudler, and F. K. Coutts, "Eigenvalue decomposition of a parahermitian matrix: Extraction of analytic eigenvalues," *IEEE Trans. Signal Processing*, **69**:722–737, 2021.
- [12] J. Foster, J. McWhirter, M. Davies, and J. Chambers, "An algorithm for calculating the qr and singular value decompositions of polynomial matrices," *IEEE Trans. Signal Processing*, **58**(3):1263–1274, Mar. 2010.
- [13] J. G. McWhirter, "An algorithm for polynomial matrix SVD based on generalised Kogbetliantz transformations," in *EUSIPCO*, Aalborg, Denmark, pp. 457–461, Aug. 2010.
- [14] A. Mertins, "MMSE design of redundant FIR precoders for arbitrary channel lengths," *IEEE Trans. Signal Processing*, **51**(9):2402–2409, Sep. 2003.
- [15] N. Moret, A. Tonello, and S. Weiss, "MIMO precoding for filter bank modulation systems based on PSVD," in *IEEE VTC-Spring*, May 2011.
- [16] D. Hassan, S. Redif, J.G. McWhirter, and S. Lambotharan, "Polynomial GSVD beamforming for two-user frequency-selective mimo channels," *IEEE Trans. Signal Processing*, **69**:948–959, 2021.
- [17] S. Weiss and S. J. Schlecht, "Polynomial procrustes problem: Paraunitary approximation of matrices of analytic functions," in *EUSIPCO*, Helsinki, Finland, 2023.
- [18] J. Corr, K. Thompson, S. Weiss, J. McWhirter, S. Redif, and I. Proudler, "Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices," in *IEEE SSP*, Gold Coast, Australia, pp. 312–315, June 2014.
- [19] S. Redif, S. Weiss, and J. McWhirter, "Sequential matrix diagonalization algorithms for polynomial EVD of parahermitian matrices," *IEEE Trans. Signal Processing*, **63**(1):81–89, Jan. 2015.
- [20] G. Barbarino and V. Noferini, "On the Rellich eigendecomposition of para-Hermitian matrices and the sign characteristics of *-palindromic matrix polynomials," *arXiv:2211.15539*, 2022.
- [21] S. Weiss, I. K. Proudler, G. Barbarino, J. Pestana, and J. G. McWhirter, "On properties and structure of the analytic singular value decomposition," submitted.
- [22] B. De Moor and S. Boyd, "Analytic properties of singular values and vectors," KU Leuven, Tech. Rep., 1989.
- [23] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nicols, "Numerical computation of an analytic singular value decomposition of a matrix valued function," *Numer. Math.*, **60**:1–40, 1991.
- [24] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter, "Row-shift corrected truncation of paraunitary matrices for PEVD algorithms," in *EUSIPCO*, Nice, France, pp. 849–853, Aug. 2015.
- [25] J. G. McWhirter and Z. Wang, "A novel insight to the SBR2 algorithm for diagonalising para-Hermitian matrices," in *11th IMA Conf. Maths in Signal Processing*, Birmingham, UK, Dec. 2016.
- [26] F.A. Khattak, S. Weiss, I.K. Proudler, and J.G. McWhirter, "Space-time covariance matrix estimation: Loss of algebraic multiplicities of eigenvalues," in *56th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2022.
- [27] J. Foster, J.G. McWhirter, and J. Chambers, "Limiting the order of polynomial matrices within the SBR2 algorithm," in *IMA Int. Conf. Maths in Signal Processing*, Cirencester, UK, Dec. 2006.
- [28] C.H. Ta and S. Weiss, "Shortening the order of paraunitary matrices in SBR2 algorithm," in *ICICSP*, Singapore, Dec. 2007.
- [29] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter, "Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms," in *Sensor Signal Processing for Defence*, Edinburgh, Scotland, Sep. 2015.

A Novel Adaptive Architecture: Joint Multi-targets Detection and Clutter Classification

Linjie Yan^{*}, Carmine Clemente[†], Sudan Han[‡], Chengpeng Hao^{*}, Danilo Orlando[§], and Giuseppe Ricci[¶]

^{*} *Institute of Acoustics, Chinese Academy of Sciences, Beijing, China*

[†] *Department of Electronic and Electrical Engineering, University of Strathclyde, G1 1XW Glasgow, U.K.*

[‡] *Academy of Military Sciences, Beijing, China*

[§] *Università degli Studi “Niccolò Cusano”, Roma, Italy*

[¶] *Dipartimento di Ingegneria dell’Innovazione, Università del Salento, Lecce, Italy*

E-mail: yanlinjie16@163.com

Abstract—A novel adaptive architecture is conceived to face with the problem of multiple point-like targets detection buried in Gaussian disturbance with the lacking of targets information, including their positions, number, and angles of arrival. To this end, a target-rich scenario where the clutter properties vary over the range profile is considered. Such distinct clutter properties are modeled in terms of different interference covariance matrices that provide the basis to jointly classify clutter and targets over the range. Specifically, suitable estimates of the unknown parameters are figured out by adopting the expectation-maximization algorithm together with a grid search approach. Then, a decision scheme based upon the Likelihood Ratio Test is exploited along with the estimated values. The performance assessment, conducted resorting to simulated data, highlights the effectiveness of the proposed scheme in heterogeneous interference.

Index Terms—Clutter classification, expectation-maximization, heterogeneous environment, multiple targets, radar detection

I. INTRODUCTION

Adaptive radar detection embedded in Gaussian interference is a ubiquitous task, and due to the advances in electronic technology and digitalization, it has received a great attention and continuous performance enhancement over the past few decades. The complexity of operating scenarios is proportional to the increase in computational resources that allows for miniaturized high performance sensors and processing boards.

From this point of view, the heterogeneity assumption for the radar returns, namely the statistical properties of the interference vary over the range bins due to various types of terrain, clutter discrete, or outliers, is becoming a very common situation of practical interest. In this context, several interesting solutions have been proposed with novel contributions on clutter classification. In [1], [2], adaptive architectures are designed to localize the clutter edge positions present in the radar reference window. Precisely, a sliding window moves over the entire radar window and for each position a test on the presence of a clutter edge is performed. Another approach to deal with the problem of clustering clutter returns is provided by [3]. At the design stage, a classification procedure exploiting the Expectation-Maximization (EM) algorithm [4] and the latent variable model [5] has been proposed with

the capability of partitioning the heterogeneous dataset into homogeneous clusters. More recently, classification schemes are proposed to identify the data homogeneity, accounting for homogeneous, partially-homogeneous environments with possible clutter edges [6], [7].

Another challenging scenario in modern radar systems is the target-rich environment, where the structured echoes may contaminate training data leading to a high risk of incorporating target components into the covariance matrix estimate with a consequent reduction of the receiver sensitivity. In [8], a tangible example of this problem is provided by the so-called Adaptive Matched Filter with De-emphasis (AMFD) that with the de-emphasis parameter equal to 0 is equivalent to the Adaptive Matched Filter (AMF) derived in [9]. However, the AMFD can return poor detection performance with respect to the AMF and Kelly’s detector [10]. To overcome this drawback, architectures not only capable of identifying heterogeneous clutter returns having different statistical properties but also of jointly detecting multiple point-like targets within each clutter region are urgently needed.

In this work, we elaborate on the systematic framework proposed in [11] to develop algorithms to jointly classify the clutter echoes and detect multiple point-like targets whose positions, number, and Angle of Arrival (AoA) are unknown. To this end, we leverage the contaminated Gaussian model for the interference¹ and introduce hidden random variables that represent different operating situations at range bin level. The unknown interference and target parameters are estimated by a suitable modification of the EM algorithm in conjunction with a grid search approach to seek the most likely estimates of the AoAs. Then, a decision scheme based upon the Likelihood Ratio Test (LRT) is built up to accomplish the detection task. Finally, the numerical examples obtained over synthetic data highlight the effectiveness of the proposed architecture.

The remainder of this work is organized as follows. In the next section, the formal statement of the addressed problem is provided whereas in Section III, the proposed architecture is

¹In what follows, we use the term interference to denote the overall disturbance affecting radar data that, generally speaking, is the sum of thermal noise and clutter. When we use the term clutter to denote the interference, it is understood that the clutter component is the most significant.

This work was supported by the National Natural Science Foundations of China under Grant No. 61971412 and No. 62201564.

designed. Section IV is devoted to the performance assessment and, finally, concluding remarks are drawn in Section V.

II. PROBLEM FORMULATION

Let us denote by $\mathbf{z}_1, \dots, \mathbf{z}_K$ the N -dimensional vectors representing the returns from K range bins of the region under surveillance, in which the statistical characterization of the clutter component is assumed to be range-dependent [11]. The set $\Omega = \{1, \dots, K\}$ denotes the cardinality of $\mathbf{z}_k, k = 1, \dots, K, \Omega_l^c \subseteq \Omega, l = 1, \dots, L^2$ represents the unknown indices of range bins sharing the same interference covariance matrix with L the known number of homogeneous subsets coming from the a priori information about the terrain types of the interested region. Moreover, $\Omega_l^t \subseteq \Omega_l^c$ indexes vectors containing target components within Ω_l^c . Notice that once Ω_l^t has been estimated, the positions and number of targets within the entire data window can be obtained. With the above remarks in mind, the detection problem for the multiple deterministic targets can be formulated as the following binary hypothesis test where $l = 1, \dots, L$

$$\begin{cases} H_0 : \mathbf{z}_k \sim \mathcal{CN}_N(\mathbf{0}, \mathbf{M}_l), k \in \Omega_l^c, \\ H_1 : \begin{cases} \mathbf{z}_k \sim \mathcal{CN}_N(\mathbf{0}, \mathbf{M}_l), k \in \Omega_l^c \setminus \Omega_l^t, \\ \mathbf{z}_k \sim \mathcal{CN}_N(\alpha_k \mathbf{v}(\theta_t), \mathbf{M}_l), k \in \Omega_l^t, \end{cases} \end{cases} \quad (1)$$

where H_0 is the noise-only hypothesis, H_1 denotes the signal-plus-interference hypothesis, $\mathcal{CN}_N(\mathbf{0}, \mathbf{M}_l)$ denotes the N -dimensional circular complex Gaussian distribution with mean $\mathbf{0}$ and unknown positive definite covariance matrix in the l th clutter region \mathbf{M}_l . Moreover, $\alpha_k \in \mathbb{C}$ is an unknown deterministic factor accounting for target response and channel effects, $\mathbf{v}(\theta_t) \in \mathbb{C}^{N \times 1}$ denotes the spatial steering vector pointed along θ_t [12], which is the unknown AoA of each target, $\Omega_l^c \setminus \Omega_l^t$ represents the index vector containing the homogeneous returns in terms of their covariance matrix except for the targets components, and \mathbf{z}_k s are statistically independent.

For future reference, let us denote by $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_K] \in \mathbb{C}^{N \times K}$ the overall data matrix. $\mathcal{P}_{0,k} = \{\Omega_l^c, \mathbf{M}_l : l = 1, \dots, L\}$ and $\mathcal{P}_{1,k} = \{\Omega_l^t, \Omega_l^c, \theta_t, \alpha_k, \mathbf{M}_l : l = 1, \dots, L\}$ the sets of unknown parameters associated with the distribution of \mathbf{z}_k under H_0 and H_1 , respectively. Based on these, the Probability Density Functions (PDFs) of \mathbf{z}_k under H_0 are

$$f_0(\mathbf{z}_k; \mathcal{P}_{0,k}) = \frac{\exp\{-\text{tr}[\mathbf{M}_l^{-1} \mathbf{z}_k \mathbf{z}_k^\dagger]\}}{\pi^N \det(\mathbf{M}_l)}, k \in \Omega_l^c, \quad (2)$$

and under H_1

$$f_1(\mathbf{z}_k; \mathcal{P}_{1,k}) = \begin{cases} \frac{\exp\{-\text{tr}[\mathbf{M}_l^{-1} \mathbf{z}_k \mathbf{z}_k^\dagger]\}}{\pi^N \det(\mathbf{M}_l)}, k \in \Omega_l^c \setminus \Omega_l^t, \\ \frac{\exp\{-\text{tr}[\mathbf{M}_l^{-1} \mathbf{u}_k \mathbf{u}_k^\dagger]\}}{\pi^N \det(\mathbf{M}_l)}, k \in \Omega_l^t. \end{cases} \quad (3)$$

where $\mathbf{u}_k = \mathbf{z}_k - \alpha_k \mathbf{v}(\theta_t)$, $\text{tr}(\cdot)$, $\det(\cdot)$, and $(\cdot)^\dagger$ denote the trace, determinant, and conjugate transpose, respectively.

²We have $\cup_{l=1}^L \Omega_l^c = \Omega$.

III. PROPOSED ARCHITECTURE DESIGN

The aftermentioned problem can be viewed as a classification problem of heterogeneous clutter as well as a detection problem of multiple targets. Let us remind that the clutter distribution parameters are unknown and, hence, must be estimated from data. However, obtaining the Maximum Likelihood Estimates (MLEs) of such parameters is a difficult task from the standpoint of mathematics [3], [13]. Thus, we firstly solve the estimation problem by introducing the hidden random variables and devise iterative procedures based upon the EM algorithm and a grid search method. Finally, an adaptive LRT detector is built up based on the estimates.

A. Estimation procedures for deterministic targets

Let us assume that K independent and identically distributed discrete random variables, c_k s, $k = 1, \dots, K$ say, which have unknown Probability Mass Function (PMF), $P(c_k = n_{s,l}) = p_{s,l}$ with $n_{s,l} \in \{1, \dots, L_c\}$ coding different operating situations, $L_c = L_s + L$, $s = 0, 1$, accounting for the number of clutter covariance classes and the presence of a possible target controlled by L_s , $s = 0, 1$. As a consequence, $L_c = L$ under H_0 and $L_c = 2L$ under H_1 . Accordingly, the logarithm of \mathbf{z}_k under $H_i, i = 0, 1$ can be rewritten as

$$g_1(\mathbf{z}_k; \mathcal{P}'_{1,k}) = \sum_{s=0}^1 \sum_{l=1}^L p_{s,l} f_1(\mathbf{z}_k | c_k = n_{s,l}; \mathcal{P}_{1,k,l}), \quad (4)$$

$$g_0(\mathbf{z}_k; \mathcal{P}'_{0,k}) = \sum_{l=1}^L p_{0,l} f_0(\mathbf{z}_k | c_k = n_{0,l}; \mathbf{M}_{n_{0,l}}), \quad (5)$$

where $\mathcal{P}_{1,k,l} = \{\mathbf{M}_l, \alpha_k, \theta_t : l = 1, \dots, L\}$, $\mathcal{P}'_{1,k} = \mathcal{P}_{1,k,l} \cup \mathcal{A}$, $\mathcal{P}'_{0,k} = \mathbf{M}_{n_{0,l}} \cup \mathcal{A}$ with $\mathcal{A} = \{p_{s,l}\}_{s=0,1}^{l=1, \dots, L}$ and $n_{0,l}$ the indices of clutter region under H_0 . Note that when there is no target in the k th range bin under H_1 , namely $s = 0$, we set $\alpha_{0,k} = 0$ and ignore the target AOA. Notice that the estimation procedures under H_0 have been mostly developed in [3] and, hence, are omitted here for brevity.

Applying the EM algorithm under H_1 , we can write the joint log-likelihood of \mathbf{Z} as follows

$$\begin{aligned} \mathcal{L}(\mathbf{Z}; \mathcal{P}_1) &= \sum_{k=1}^K \log g_1(\mathbf{z}_k; \mathcal{P}'_{1,k}) \\ &= \sum_{k=1}^K \log \left[\sum_{s=0}^1 \sum_{l=1}^L p_{s,l} f_1(\mathbf{z}_k | c_k = n_{s,l}; \mathcal{P}_{1,k,l}) \right], \end{aligned} \quad (6)$$

where $\mathcal{P}_1 = \cup_{k=1}^K \mathcal{P}'_{1,k}$. In order to mitigate the overestimation inclination of MLEs, we borrow likelihood approximations resorting to the Model Order Selection (MOS) rules [14]. Before moving on, notice that the joint estimation of $\mathbf{M}_l, \alpha_{1,k}$ and θ_t is a difficult task due to the intractable mathematics. A suboptimum solution is to apply EM algorithm first on the basis of a given $\theta_t \in \{\theta_1, \dots, \theta_T\}$. Then, the target AoA is

estimated through a grid search. Update rule at the E-step is

$$q_k^{(h-1)}(Ls+l) = \frac{f_1\left(\mathbf{z}_k|c_k=n_{s,l};\widehat{\mathcal{P}}_{1,k,l}^{(h-1)}\right)e^{-u(s)}\widehat{p}_{s,l}^{(h-1)}}{\sum_{j=0}^1\sum_{i=1}^L f_1\left(\mathbf{z}_k|c_k=n_{j,i};\widehat{\mathcal{P}}_{1,k,l}^{(h-1)}\right)e^{-u(j)}\widehat{p}_{j,i}^{(h-1)}}, \quad (7)$$

where $\widehat{\mathcal{P}}_{1,k,l}^{(h-1)}$ and $\widehat{p}_{s,l}^{(h-1)}$ are the estimates of $\mathcal{P}_{1,k,l}$ and $p_{s,l}$ in the $(h-1)$ th iteration with known AoA, $u(j)$, $j=0,1$ is the penalty term borrowed from the MOS rules, whose expression is $u(j) = (N^2 + 2j)(1 + \rho)/2$ with $\rho \geq 1$ for Generalized Information Criterion (GIC) [15].

The second step is the M-step leading to the following maximization problem with respect to \mathcal{P}_1

$$\widehat{\mathcal{P}}_1^{(h)} = \arg \max_{\mathcal{P}_1} \left\{ \sum_{k=1}^K \sum_{s=0}^1 \sum_{l=1}^L q_k^{(h-1)}(Ls+l) \times \log f_1(\mathbf{z}_k|c_k=n_{s,l};\mathcal{P}_{1,k,l}) + \sum_{k=1}^K \sum_{s=0}^1 \sum_{l=1}^L q_k^{(h-1)}(Ls+l) \log p_{s,l} \right\}. \quad (8)$$

Observing that the estimation procedure with respect to $p_{s,l}$, $s=0,1$, $l=1,\dots,L$, is independent of that over M_l , $\alpha_{1,k}$ and θ_t . Thus, we start from the optimization over $p_{s,l}$ solved by using the method of Lagrange multipliers to take into account the constraint $\sum_{s=0}^1 \sum_{l=1}^L p_{s,l} = 1$. We obtain

$$\widehat{p}_{s,l}^{(h)} = \frac{1}{K} \sum_{k=1}^K q_k^{(h-1)}(Ls+l), \quad s=0,1, \quad l=1,\dots,L. \quad (9)$$

Ignoring the terms related to $p_{s,l}$ in (8) and replacing the PDFs with their expressions, the maximization problem with respect to the rest parameters of interest is tantamount to

$$\min_{\alpha_{1,k}} \min_{M_l} \sum_{k=1}^K \sum_{s=0}^1 \sum_{l=1}^L q_k^{(h-1)}(Ls+l) \left\{ \log \det(M_l) + \text{tr} \left[M_l^{-1} (\mathbf{z}_k - \alpha_{s,k} \mathbf{v}(\theta_t)) (\mathbf{z}_k - \alpha_{s,k} \mathbf{v}(\theta_t))^\dagger \right] \right\}. \quad (10)$$

A cyclic optimization procedure is used to estimate M_l and $\alpha_{1,k}$. More specific, assume $\alpha_{1,k}$ is known and estimate M_l in the $(m-1)$ th iteration of the inner procedure. Then, set the M_l s to the values obtained at the previous step and estimate the $\alpha_{1,k}$ s. To this end, we solve the following problem

$$\min_{M_l} \left\{ -\log \det(M_l^{-1}) \sum_{k=1}^K \sum_{s=0}^1 q_k^{(h-1)}(Ls+l) + \text{tr} \left[M_l^{-1} \sum_{k=1}^K \sum_{s=0}^1 q_k^{(h-1)}(Ls+l) \times (\mathbf{z}_k - \widehat{\alpha}_{s,k}^{(h-1),(m-1)} \mathbf{v}(\theta_t)) (\mathbf{z}_k - \widehat{\alpha}_{s,k}^{(h-1),(m-1)} \mathbf{v}(\theta_t))^\dagger \right] \right\}. \quad (11)$$

where $\widehat{\alpha}_{1,k}^{(h-1),(m-1)}$, $k=1,\dots,K$, denote the estimates of the $\alpha_{1,k}$ s at the $(h-1)$ th EM step and $(m-1)$ th step of this inner procedure. The minimizer can be obtained by resorting to the following inequality [16] $\log \det(\mathbf{A}) \leq \text{tr}[\mathbf{A}] - N$, where \mathbf{A} is any N -dimensional matrix with nonnegative eigenvalues, and, hence, we come up with

$$\widehat{M}_l^{(h-1),(m)} = \frac{\sum_{s=0}^1 \sum_{k=1}^K q_k^{(h-1)}(Ls+l)}{\sum_{s=0}^1 \sum_{k=1}^K q_k^{(h-1)}(Ls+l)} \times (\mathbf{z}_k - \widehat{\alpha}_{s,k}^{(h-1),(m-1)} \mathbf{v}(\theta_t)) (\mathbf{z}_k - \widehat{\alpha}_{s,k}^{(h-1),(m-1)} \mathbf{v}(\theta_t))^\dagger. \quad (12)$$

Now, assuming that in (10) $M_l = \widehat{M}_l^{(h-1),(m)}$, $l=1,\dots,L$, and setting to zero the first derivative of the objective function with respect to $\alpha_{1,k}$ leads to

$$\widehat{\alpha}_{1,k}^{(h-1),(m)} = \frac{\sum_{l=1}^L q_k^{(h-1)}(L+l) \mathbf{v}(\theta_t)^\dagger \left(\widehat{M}_l^{(h-1),(m)} \right)^{-1} \mathbf{z}_k}{\sum_{l=1}^L q_k^{(h-1)}(L+l) \mathbf{v}(\theta_t)^\dagger \left(\widehat{M}_l^{(h-1),(m)} \right)^{-1} \mathbf{v}(\theta_t)}. \quad (13)$$

The above inner procedure continues until a convergence criterion is satisfied. The final estimates of $\alpha_{1,k}$ and M_l are denoted by $\widehat{\alpha}_{1,k}^{(h)}$ and $\widehat{M}_l^{(h)}$, respectively. All the results obtained from the maximization step are used in the next cycle of the EM-based procedure. Finally, the optimization with respect to AoA of targets is tantamount to

$$\begin{aligned} \widehat{\theta}_t &= \max_{\theta_t \in \{\theta_1, \dots, \theta_T\}} \mathcal{L}(\mathbf{Z}; \widehat{\mathcal{P}}_1^{(h_{max})}) \\ &= \max_{\theta_t \in \{\theta_1, \dots, \theta_T\}} \sum_{k=1}^K \log \left[\sum_{s=0}^1 \sum_{l=1}^L \widehat{p}_{s,l}^{(h_{max})} \right. \\ &\quad \left. \times f_1\left(\mathbf{z}_k|c_k=n_{s,l};\widehat{\mathcal{P}}_{1,k,l}^{(h_{max})}\right) \right]. \end{aligned} \quad (14)$$

where $\widehat{\mathcal{P}}_1^{(h_{max})}$, $\widehat{p}_{s,l}^{(h_{max})}$ and $\widehat{\mathcal{P}}_{1,k,l}^{(h_{max})}$ are the estimates of \mathcal{P}_1 , $p_{s,l}$ and $\mathcal{P}_{1,k,l}$ for the maximum number of EM iterations and for known AoA. The workflow of the estimation procedure based on EM and grid-search are summarized in Algorithm 1 for the reader ease.

B. Adaptive detector with classification capabilities

The adaptive detector devised in this section is grounded on the LRT where the unknown parameters are replaced by the previously-obtained estimates, namely

$$\prod_{k=1}^K \frac{g_1(\mathbf{z}_k; \widehat{\mathcal{P}}_{1,k}^{\prime})}{g_0(\mathbf{z}_k; \widehat{\mathcal{P}}_{0,k}^{\prime})} \underset{H_0}{\overset{H_1}{>}} \eta \quad (15)$$

where η is the generic detection threshold to be set according to the required probability of false alarm (P_{fa}). For classification purposes, the unknown quantities that have been estimated

Algorithm 1: Estimation procedure based on EM and grid-search.

Input: $L, \mathbf{Z}, \theta_t, t = 1, \dots, T$

Output: $\hat{\Omega}_l^c, \hat{\Omega}_l^t, l = 1, \dots, L, \hat{\mathcal{P}}'_{1,k}, k = 1, \dots, K$

Latent Variable Model: introduce the hidden random variables $c_k, k = 1, \dots, K$ accounting for different clutter types and the presence of a possible target;

for $\theta_t, t = 1, \dots, T$ **do**

Parameters initialization: set $h = 0,$

$\hat{\mathcal{P}}'_{1,k}^{(0)}, k = 1, \dots, K;$

E-step: compute the conditional expectation of \mathbf{z}_k and obtain update rule of $q_k^{(h)}(L_s + l);$

M-step: maximize the log-likelihood to get updates for $\hat{\mathcal{P}}'_{1,k}^{(h+1)}, k = 1, \dots, K$ with the inner cyclic iterations $m = m_{max};$

if $h = h_{max}$ or convergence criterion is satisfied

then

 | set $t = t + 1$ and continue;

else

 | set $h = h + 1$ and go to E-step;

end

end

Estimate $\theta_t: \hat{\theta}_t = \max_{\theta_t \in \{\theta_1, \dots, \theta_T\}} \mathcal{L}(\mathbf{Z}; \hat{\mathcal{P}}_1^{(h_{max})}).$

allow us to separate the target response from the heterogeneous clutter by exploiting the following rule under H_1

$$\mathbf{z}_k \sim \begin{cases} \mathcal{CN}_N(\mathbf{0}, \widehat{\mathbf{M}}_{\hat{l}_k}^{(h_{max})}), & 1 \leq \hat{l}_k \leq L, \\ \mathcal{CN}_N(\hat{\alpha}_k^{(h_{max})} \mathbf{v}(\hat{\theta}_t), \widehat{\mathbf{M}}_{\hat{l}_k - L}^{(h_{max})}), & L + 1 \leq \hat{l}_k \leq 2L, \end{cases} \quad (16)$$

with $\hat{l}_k = \arg \max_{l=1, \dots, 2L} q_k^{(h_{max})}(l)$, where $\widehat{\mathbf{M}}_{\hat{l}_k}^{(h_{max})}$ and $\hat{\alpha}_k^{(h_{max})}$ denote the final estimates of \mathbf{M}_l and $\alpha_{1,k}$ at the h_{max} th step corresponding to $\hat{\theta}_t$.

IV. ILLUSTRATIVE EXAMPLES

This section is devoted to the investigation of the classification and detection performance of the proposed architecture using simulated data. Specifically, we resort to standard Monte Carlo counting technique by evaluating the detection threshold over $100/P_{fa}$ independent runs with $P_{fa} = 10^{-3}$ and the Probability of Detection (P_d) over 1000 independent trials. All the illustrative examples assume $N = 8, L = 3, h_{max} = 15, \rho = 3$, and the maximum number of the inner cyclic iterations $m_{max} = 5$. As for the parameters initialization of the EM iterations under H_0 and H_1 , we initialize the $p_{s,l}$ s with $p_{s,l} = 1/L_c$; the initial value of \mathbf{M}_l , namely $\widehat{\mathbf{M}}_l^{(0)}$, is generated in the same way as in Section IV.A of [3]. A possible

choice for $\hat{\alpha}_{s,k}^{(0)}, s = 0, 1, k = 1, \dots, K$, is

$$\hat{\alpha}_{s,k}^{(0)} = \begin{cases} 0, & s = 0, \\ \max_{l=1, \dots, L} \left(\frac{\mathbf{v}(\theta_t)^\dagger (\widehat{\mathbf{M}}_l^{(0)})^{-1} \mathbf{z}_k}{\mathbf{v}(\theta_t)^\dagger (\widehat{\mathbf{M}}_l^{(0)})^{-1} \mathbf{v}(\theta_t)} \right), & s = 1. \end{cases}$$

As for targets' AoA estimation, for simplicity, we sample the angular sector under surveillance ranging from -20° to 20° with a sampling interval of 5° . Synthetic targets assuming the same Signal-to-Interference-plus-Noise Ratio (SINR) value lie on the sampling grid with the AoA of 0° . Specifically, SINR is defined as $\text{SINR} = |\alpha_{k,l}|^2 \mathbf{v}(\theta_n)^\dagger \boldsymbol{\Sigma}_l^{-1} \mathbf{v}(\theta_n)$, where $\boldsymbol{\Sigma}_l$ is the interference covariance matrix consisting of a thermal noise component and heterogeneous clutter, $\alpha_{k,l}$ is the amplitude associated with a target in the k th range bin of the l th region, $\mathbf{v}(\theta_n)$ is the nominal steering vector with θ_n the nominal AoA.

Let us focus on the scenario that consists of three clutter regions, namely, $K_1 = K_2 = K_3 = 32$ range bins in each region comprise the scenario of interest with $\text{CNR}_1 = 20$ dB, $\text{CNR}_2 = 30$ dB and $\text{CNR}_3 = 40$ dB, respectively. $\text{CNR}_l = \sigma_{c,l}^2 / \sigma_n^2$ denotes the Clutter-to-Noise Ratio in the l th, $l = 1, \dots, L$ clutter region, where $\sigma_{c,l}^2 > 0$ is the clutter power and $\sigma_n^2 = 1$ is the thermal noise power. Five targets appear at the 6th, 15th, 36th, 42th, and 80th range bin in the considered scenario. With these remarks in mind, this operating scenario yields $L_c = 6$ considered classes, that is classes 1-3 where the generic vector of the l th region does not contain any target component and classes 4-6 where the generic vector of the l th region contains target components.

In Fig. 1, we plot a snapshot of the classification results over one Monte Carlo trial under H_1 for $\text{SINR} = 20$ dB. From the figure inspection, the classifier can correctly classify most of returns containing multiple targets and heterogeneous clutter. In Fig. 2, we evaluate the P_d to verify the effectiveness of the proposed algorithm in terms of target detection in comparison with Kelly's Generalized Likelihood Ratio Test (Kelly's GLRT) in [17] under multi-targets situation. In particular, in Fig. 2 (a) we plot the P_d curve of Kelly's GLRT under the condition that the target under test is in the 6th range bin and other targets are regarded as jammings with $\text{SINR} = 15$ dB and $L = 3$, in other words, the number of secondary data is $K_{glrt} = 95$, while in Fig. 2 (b) we plot it assuming $K_{glrt} = 16$ and the cell under test is the number 9. Notice that, in this case, secondary data used by Kelly's GLRT are affected by one interferer only. The results show that the proposed detector exhibits much better detection performance than Kelly's GLRT, which can be motivated by the fact that the secondary data contamination caused by the redundant targets leads to the performance degradation of the classical detector. In order to measure the error in target AoA estimation, we estimate the Root Mean Square Error (RMSE) values with the true target AoA of 0° being on-grid and of 2° being off-grid, respectively, over 1000 trials in Fig. 3. Note that the RMSE metric decreases as the SINR increases, and in Fig. 3 (a) it becomes 0 degree for SINR values greater than 14

dB whereas the performance degradation occurs in Fig. 3 (b) where the minimum RMSE value is approximately 2 degrees for $\text{SINR} > 30$ dB.

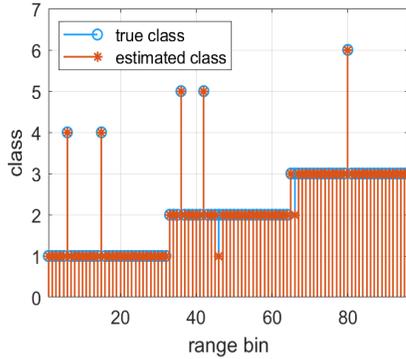


Fig. 1: Classification snapshot for $\text{SINR} = 20$ dB under H_1 .

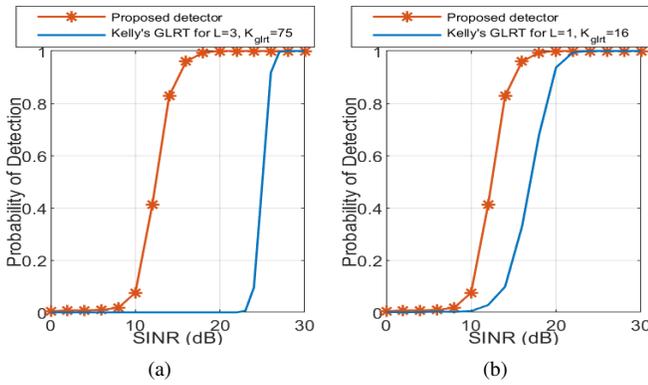


Fig. 2: P_d of the proposed detector and Kelly's GLRT versus SINR under H_1 assuming $P_{fa} = 10^{-3}$.

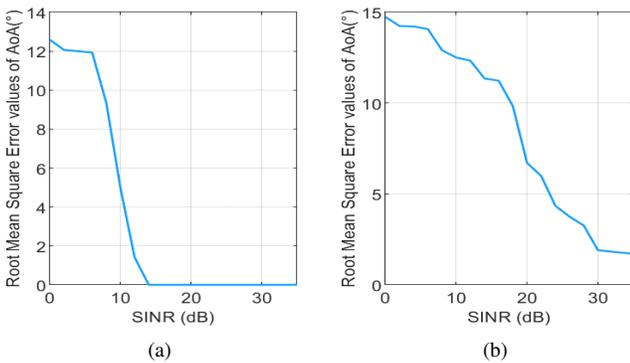


Fig. 3: RMSE values of AoA under H_1 for different SINRs: (a) true target AoA is on-grid; (b) true target AoA is off-grid.

V. CONCLUSIONS

This paper has addressed the problem of multiple point-like targets detection from an unknown AoA and in the presence

of heterogeneous Gaussian clutter and the ubiquitous thermal noise. At the design stage, we account for the heterogeneity of the operating scenario modeled as a variation of covariance matrices over the range cells. Within this framework, the EM algorithm in conjunction with grid search technique are used to estimate the unknown distribution parameters. Hence, the clutter region classification results, targets positions over range and AoA can be obtained. Finally, an adaptive detector is introduced resorting to the LRT criterion, and the superiority with respect to a conventional detector is verified based on simulated data. Possible future research can extend the proposed framework to the scenarios that consider the joint presence of point-like as well as range-spread targets.

REFERENCES

- [1] D. Xu, P. Addabbo, C. Hao, J. Liu, D. Orlando, and A. Farina, "Adaptive strategies for clutter edge detection in radar," *Signal Processing*, vol. 186, p. 108127, 2021.
- [2] T. Wang, D. Xu, C. Hao, P. Addabbo, and D. Orlando, "Clutter Edges Detection Algorithms for Structured Clutter Covariance Matrices," *IEEE Signal Processing Letters*, vol. 29, pp. 642–646, 2022.
- [3] P. Addabbo, S. Han, D. Orlando, and G. Ricci, "Learning Strategies for Radar Clutter Classification," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1070–1082, 2021.
- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society (Series B - Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [5] K. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2012.
- [6] C. Yin, L. Yan, C. Hao, S. L. Ullo, G. Giunta, A. Farina, and D. Orlando, "Classification schemes for the radar reference window: Design and comparisons," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–18, 2023.
- [7] C. Yin, L. Yan, C. Hao, S. L. Ullo, D. Orlando, and C. Hou, "Novel classification schemes for radar scenarios: Design and comparison," in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 1861–1865.
- [8] C. Richmond, "The theoretical performance of a class of space-time adaptive detection and training strategies for airborne radar," in *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No.98CH36284)*, vol. 2, 1998, pp. 1327–1331 vol.2.
- [9] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 1, pp. 208–216, 1992.
- [10] F. Bandiera, D. Orlando, and G. Ricci, *Advanced Radar Detection Schemes Under Mismatched Signal Models*. San Rafael, US: Synthesis Lectures on Signal Processing No. 8, Morgan & Claypool Publishers, 2009.
- [11] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar: Basic Principles*. Raleigh, NC: Scitech Publishing, 2010.
- [12] L. Yan, P. Addabbo, C. Hao, D. Orlando, and A. Farina, "New ECCM Techniques Against Noise-like and/or Coherent Interferers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1172–1188, 2020.
- [13] P. Addabbo, J. Liu, D. Orlando, and G. Ricci, "Novel Parameter Estimation and Radar Detection Approaches for Multiple Point-Like Targets: Designs and Comparisons," *IEEE Signal Processing Letters*, vol. 27, pp. 1789–1793, 2020.
- [14] P. Stoica and Y. Selen, "Model-order selection: A review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, 2004.
- [15] S. Yan, F. Lotfi, S. Chen, C. Hao, and D. Orlando, "Innovative two-stage radar detection architectures in adverse scenarios using two training data sets," *IEEE Signal Processing Letters*, vol. 28, pp. 1165–1169, 2021.
- [16] H. Lütkepohl, *Handbook of Matrices*. Wiley, 1997.
- [17] E. J. Kelly, "An adaptive detection algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, no. 2, pp. 115–127, 1986.

Consensus-based Distributed Variational Multi-object Tracker in Multi-Sensor Network

Qing Li, Runze Gan, Simon Godsill

Engineering Department

University of Cambridge

Cambridge, UK

{ql289, rg605, sjg30}@cam.ac.uk

Abstract—The growing need for accurate and reliable tracking systems has driven significant progress in sensor fusion and object tracking techniques. In this paper, we design two variational Bayesian trackers that effectively track multiple targets in cluttered environments within a sensor network. We first present a centralised sensor fusion scheme, which involves transmitting sensor data to a fusion center. Then, we develop a distributed version leveraging the average consensus algorithm, which is theoretically equivalent to the centralised sensor fusion tracker and requires only local message passing with neighbouring sensors. In addition, we empirically verify that our proposed distributed variational tracker performs on par with the centralised version with equal tracking accuracy. Simulation results show that our distributed multi-target tracker outperforms the suboptimal distributed sensor fusion strategy that fuses each sensor’s posterior based on arithmetic sensor fusion and an average consensus strategy.

Index Terms—distributed sensor fusion, multiple object tracking, variational inference, average consensus, data association

I. INTRODUCTION

The distributed multi-sensor multi-object tracker has emerged as a promising approach due to its potential for reduced communication costs and increased robustness against single-node faults when compared to centralised fusion solutions. Several optimal algorithms for distributed data fusion have been developed, relying solely on local message passing [1], [2]. However, these techniques necessitate specific network topologies, such as fully connected and tree-connected networks, and often come with a high computational burden that limits their applicability in certain situations.

To overcome the limitations, several approximate methods have been studied. One popular fusion strategy is geometric average fusion, such as the widely used generalised covariance intersection method proposed in [3], with the aim of avoiding double counting of common information while fusing multiple multi-object densities with unknown correlations among sensors. An alternative approach is the arithmetic average fusion, which has been shown to perform better when fusing random variables or point estimates [4]. Consensus-based algorithms [5]–[7] have been introduced to enable geometric or arithmetic average fusion in a fully distributed manner. The analysis and comparison of these two fusion strategies can be found in [4], [8]. However, these methods that fuse the local posteriors of each sensor are suboptimal and can result in degraded tracking performance. In [9], a consensus-based method was designed

to obtain an approximation of the joint likelihood function by distributing the likelihood functions of each sensor. This likelihood consensus method was then developed to implement distributed particle filters and distributed Gaussian particle filters for multiple target tracking applications. Nevertheless, the joint likelihood function is approximate, and the estimation accuracy and fusion efficiency can be affected by the choice of the basis functions.

Here we propose a solution for distributed sensor fusion and object tracking by leveraging the non-homogeneous Poisson process (NHPP) measurement model and the recently-developed NHPP trackers [10]. The original NHPP tracker in [11] successfully avoids the data association problem, but its particle filter implementation is limited by the curse of dimensionality. To address this issue, an association-based NHPP measurement model was introduced in [12] to enable efficient parallel computing and a tractable structure. Additionally, a fast Rao-Blackwellised sequential Markov chain Monte Carlo sampling scheme was developed in [13] with improved efficiency compared to [12] for linear Gaussian models. While sampling-based methods like those presented in [12]–[14] can theoretically converge to optimal Bayesian filters, their computational requirements can be intensive when the number of targets and measurements increases. Therefore, a high-performance variational inference implementation was designed in [10], which achieves comparable tracking accuracy with sampling-based implementations [12] while offering faster processing speeds.

This paper develops an extension of the variational Bayes multi-object tracker presented in [10] to multi-sensor cases, as it has demonstrated superior tracking performance in terms of both accuracy and implementation efficiency. Our key contribution is the development of a variational filtering framework for tracking multiple objects in a distributed sensor network. This is accomplished by leveraging the average consensus algorithm, which, when successfully converged, allows the distributed version to be theoretically equivalent to the centralised sensor fusion tracker. In particular, each sensor in the network runs locally using its own measurements while communicating with its neighbouring sensors to obtain global statistics for the local coordinate ascent update. Once the average consensus algorithm has converged, the local estimates for each sensor are updated using the global statistics obtained from the con-

sensus. Therefore, this approach only requires communication with neighbouring sensors and does not require complete knowledge of the network topology. Overall, the proposed approach allows for distributed sensor fusion and tracking that can achieve tracking accuracy equivalent to centralised fusion while being more efficient in communication costs. The simulation results show that compared to the arithmetic fusion method that fuses the local posteriors of each sensor, the proposed distributed variational tracker exhibits superior tracking accuracy and efficiency.

II. PROBLEM FORMULATION AND MODELLING

This paper considers tracking multiple targets in clutter under a distributed sensor network where the communication links between sensors can be time-varying. Assume that there are K targets in the surveillance area. At each discrete time step n , their joint state is $X_n = [X_{n,1}^\top, X_{n,2}^\top, \dots, X_{n,K}^\top]^\top$, where each vector $X_{n,k}, k \in \{1, \dots, K\}$ denotes the kinematic state for the k -th target. Suppose that the targets are observed by a sensor network consisting of N_s sensors, each capable of observing the entire tracking area. The time-varying sensor network at time t can be modelled as a graph $G(t) = \{\mathcal{S}, \mathcal{E}(t)\}$ at any given continuous time t , where the sensor set is denoted by $\mathcal{S} = \{1, 2, \dots, N_s\}$, and $\mathcal{E}(t)$ is the set of edges with the existence of edge (i, j) meaning that the i -th sensor can communicate with the j -th sensor at time t . The set of neighbours of sensor i is denoted by $\mathcal{N}_i(t) = \{j \mid (i, j) \in \mathcal{E}(t)\}$. The degree $d_i(t)$ of the i -th sensor represents the number of its neighbouring sensors with which it can communicate, i.e., $d_i(t) = |\mathcal{N}_i(t)|$. In a sensor network, the measurements received from all sensors at time step n can be denoted by $Y_n = [Y_n^1, Y_n^2, \dots, Y_n^{N_s}]$. Each Y_n^s includes measurements acquired by the s -th sensor, and $Y_n^s = [Y_{n,1}^s, \dots, Y_{n,M_n^s}^s]$, where M_n^s is the total number of measurements received at the s -th sensor ($s = 1, \dots, N_s$). Subsequently, $M_n = [M_n^1, \dots, M_n^{N_s}]$ records the total number of measurements received from all sensors at time step n .

A. Dynamical model

We assume that targets move in a 2D surveillance area with each $X_{n,k} = [x_{n,k}^1, \dot{x}_{n,k}^1, x_{n,k}^2, \dot{x}_{n,k}^2]^\top$, where $x_{n,k}^d$ and $\dot{x}_{n,k}^d$ ($d = 1, 2$) indicate the k -th target's position and velocity in the d -th dimension, respectively. We assume an independent linear Gaussian transition density for each target's states:

$$p(X_n | X_{n-1}) = \prod_{k=1}^K \mathcal{N}(X_{n,k}; F_{n,k} X_{n-1,k}, Q_{n,k}). \quad (1)$$

where $F_{n,k} = \text{diag}(F_{n,k}^1, F_{n,k}^2)$, $Q_{n,k} = \text{diag}(Q_{n,k}^1, Q_{n,k}^2)$. For a constant velocity (CV) model, $F_{n,k}^d, Q_{n,k}^d$ ($d = 1, 2$) are

$$F_{n,k}^d = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, Q_{n,k}^d = \sigma_k^2 \begin{bmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{bmatrix}, \quad (2)$$

where τ is the time interval between time steps.

B. NHPP measurement model and association prior

Here, we assume each sensor independently detects targets in accordance with the NHPP measurement model described in [11]. Notably, the NHPP model may vary for each sensor. Denote the set of Poisson rates for all sensors as $\Lambda = [\Lambda^1, \Lambda^2, \dots, \Lambda^{N_s}]$. For each sensor s , the Poisson rate vector is defined by $\Lambda^s = [\Lambda_0^s, \Lambda_1^s, \dots, \Lambda_K^s]$, where Λ_0^s is the clutter rate and Λ_k^s is the k -th target rate, $k = 1, \dots, K$. For each sensor s , each target k generates measurements by a NHPP with a Poisson rate Λ_k^s , and the total measurement process is also a NHPP from the superposition of the conditional independent NHPP measurement process from K targets and clutter. The total number of measurements from the s -th sensor follows a Poisson distribution with a rate of $\Lambda_{sum}^s = \sum_{k=0}^K \Lambda_k^s$.

Our independent measurement model assumption signifies that given X_n , the measurements of each sensor are conditionally independent, i.e., $p(Y_n | X_n) = \prod_{s=1}^{N_s} p(Y_n^s | X_n)$. We denote the associations of all measurements Y_n by $\theta_n = [\theta_n^1, \theta_n^2, \dots, \theta_n^{N_s}]$, with each $\theta_n^s = [\theta_{n,1}^s, \theta_{n,2}^s, \dots, \theta_{n,M_n^s}^s]$ ($s = 1, \dots, N_s$) representing the association vector for the s -th sensor's measurements. Each component $\theta_{n,j}^s$ ($j = 1, \dots, M_n^s$) gives the origin of the measurement $Y_{n,j}^s$; $\theta_{n,j}^s = 0$ indicates that $Y_{n,j}^s$ is generated by clutter, and $\theta_{n,j}^s = k$ ($k = 1, \dots, K$) means that $Y_{n,j}^s$ is generated from the target k . The adopted conditionally independent NHPP measurement model leads to the following properties: the joint association prior are conditionally independent give all the measurement numbers

$$p(\theta_n | M_n) = \prod_{s=1}^{N_s} p(\theta_n^s | M_n^s), \quad (3)$$

and given all associations, the joint likelihood $p(Y_n | \theta_n, X_n)$ remains conditionally independent for each sensor, i.e.,

$$p(Y_n | \theta_n, X_n) = \prod_{s=1}^{N_s} p(Y_n^s | \theta_n^s, X_n). \quad (4)$$

Finally, for each sensor s , the NHPP measurement model implies the following according to [10]: measurements are conditionally independent given associations and target states

$$p(Y_n^s | \theta_n^s, X_n) = \prod_{j=1}^{M_n^s} \ell^s(Y_{n,j}^s | X_{n,\theta_{n,j}^s}), \quad (5)$$

where M_n^s is implicitly known from θ_n^s since $M_n^s = |\theta_n^s|$, and ℓ^s is the probability density function of a single measurements received in sensor s given its originator's state. Here we assume the target originated measurement follows a linear and Gaussian model while the clutter measurement is uniformly distributed in the observation area of volume V^s :

$$\ell^s(Y_{n,j}^s | X_{n,k}) = \begin{cases} \mathcal{N}(HX_{n,k}, R_k^s), & k \neq 0; \quad (\text{object}) \\ \frac{1}{V^s}, & k = 0; \quad (\text{clutter}) \end{cases} \quad (6)$$

where H is the observation matrix, and R_k^s indicates the s -th sensor noise covariance. Moreover, the joint prior $p(\theta_n^s | M_n^s)$ can be factorised as the product of M_n^s independent association priors, i.e., $p(\theta_n^s | M_n^s) = \prod_{j=1}^{M_n^s} p(\theta_{n,j}^s)$, where the prior for

each association $p(\theta_{n,j}^s)$ is a categorical distribution with support $\theta_{n,j}^s \in \{0, \dots, K\}$

$$p(\theta_{n,j}^s) = \frac{\sum_{k=0}^K \Lambda_k^s \delta[\theta_{n,j}^s = k]}{\Lambda_{sum}^s}. \quad (7)$$

III. COORDINATE ASCENT VARIATIONAL FILTERING FOR CENTRALISED SENSOR FUSION

This section develops a coordinate ascent variational filtering framework for tracking multiple objects in clutter in a centralised sensor network where there exists a central hub for collecting the measurements from multiple sensors and using them to track the targets. The parameters K , Λ , and $R_{1:K}^s$ in Section II are assumed to be known and are therefore always implicitly conditioned in our derivations. The objective is to sequentially estimate the posterior $p(X_n, \theta_n | Y_{1:n})$ given observations $Y_{1:n}$ from all sensors in the network. Accordingly, the exact optimal filtering can be recursively expressed as follows,

$$p(X_n, \theta_n | Y_{1:n}) \propto p(Y_n | \theta_n, X_n) p(\theta_n | M_n) \times \int p(X_n | X_{n-1}) p(X_{n-1} | Y_{1:n-1}) dX_{n-1}, \quad (8)$$

However, this exact filtering recursion is intractable, prompting us to replace $p(X_{n-1} | Y_{1:n-1})$ with a tractable approximate filtering prior. According to [10], a natural choice of this tractable prior is the approximate filtering result/posterior from the previous time step. In our context, where variational Bayes is employed to approximate the target distribution, this corresponds to using the converged variational distribution $q_{n-1}^*(X_{n-1})$ obtained from the approximate filtering at time step $n-1$. Therefore, the target posterior distribution of our current approximate filtering step is

$$\hat{p}_n(X_n, \theta_n | Y_n) \propto p(Y_n | \theta_n, X_n) p(\theta_n | M_n) \hat{p}_n(X_n), \quad (9)$$

where the predictive prior $\hat{p}_n(X_n)$ is written as

$$\hat{p}_n(X_n) = \int p(X_n | X_{n-1}) q_{n-1}^*(X_{n-1}) dX_{n-1}. \quad (10)$$

A. Coordinate ascent update

We assume a mean-field family of variational distributions that satisfy the factorisation $q_n(X_n, \theta_n) = q_n(X_n) q_n(\theta_n)$. Then, the variational distribution $q_n^*(X_n, \theta_n)$ is chosen from the posited family that minimises the KL divergence $\text{KL}(q_n(X_n) q_n(\theta_n) || \hat{p}_n(X_n, \theta_n | Y_n))$. This optimisation with respect to q_n can be done by the following coordinate ascent algorithm that ensures convergence. We start by setting the initial association distribution $q_n(\theta_n)$ as $q_n^{(0)}(\theta_n)$; afterwards, we iteratively update $q_n(X_n)$ while keeping $q_n(\theta_n)$ fixed, and update $q_n(\theta_n)$ while keeping $q_n(X_n)$ fixed, repeating these steps until convergence is achieved. The converged variational distribution $q_n^*(X_n, \theta_n)$ is then used to approximate the target distribution $\hat{p}_n(X_n, \theta_n | Y_n)$. We now present these updates.

1) *update for $q_n(X_n)$* : First we present the update for X_n

$$q_n(X_n) \propto \hat{p}_n(X_n) \prod_{k=1}^K \mathcal{N}(\bar{Y}_n^k; H X_{n,k}, \bar{R}_n^k), \quad (11)$$

where

$$\bar{R}_n^k = \left(\sum_{s=1}^{N_s} \Omega_{k,1}^s \right)^{-1}, \quad \Omega_{k,1}^s = (R_k^s)^{-1} \sum_{j=1}^{M_n^s} q_n(\theta_{n,j}^s = k),$$

$$\bar{Y}_n^k = \bar{R}_n^k \sum_{s=1}^{N_s} \Omega_{k,2}^s, \quad \Omega_{k,2}^s = (R_k^s)^{-1} \sum_{j=1}^{M_n^s} q_n(\theta_{n,j}^s = k) Y_{n,j}^s.$$

Such an update can be considered as updating the predictive prior $\hat{p}_n(X_n)$ in (10) with K pseudo-measurements $\bar{Y}_n^k, k = 1, 2, \dots, K$. Given an independent initial Gaussian prior $p(X_0) = \prod_{k=1}^K p(X_{0,k})$ and the transition in (1), the updated variational distribution can always be in an independent Gaussian form, i.e., $q_n(X_n) = \prod_{k=1}^K q_n(X_{n,k})$. Denote the converged variational distribution for the k -th target at time step $n-1$ as $q_{n-1}^*(X_{n-1,k}) = \mathcal{N}(X_{n-1,k}; \mu_{n-1|n-1}^{k*}, \Sigma_{n-1|n-1}^{k*})$, then we denote its predictive prior according to (10) by

$$\hat{p}_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu_{n|n-1}^{k*}, \Sigma_{n|n-1}^{k*}). \quad (13)$$

The variational distribution $q_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu_{n|n}^k, \Sigma_{n|n}^k)$ can then be updated by Kalman filtering. Such an update can be independently carried out for all targets.

2) *update for $q_n(\theta_n)$* : Since $q_n(\theta_n) = \prod_{s=1}^{N_s} q_n(\theta_n^s)$, $q_n(\theta_n)$ can be updated by individually evaluating $q_n(\theta_n^s)$ for each sensor, where the update can be performed in parallel:

$$q_n(\theta_n^s) \propto \prod_{j=1}^{M_n^s} q_n(\theta_{n,j}^s), \quad (14)$$

$$q_n(\theta_{n,j}^s) \propto \frac{\Lambda_0^s}{V_s^s} \delta[\theta_{n,j}^s = 0] + \sum_{k=1}^K \Lambda_k^s l_k^s \delta[\theta_{n,j}^s = k], \quad (15)$$

$$l_k^s = \mathcal{N}(Y_{n,j}^s; H \mu_{n|n}^k, R_k^s) \exp(-0.5 \text{Tr}((R_k^s)^{-1} H \Sigma_{n|n}^k H^\top)),$$

where each $q_n(\theta_{n,j}^s)$ is a categorical distribution and the updates for θ_n can be independently carried out for each $\theta_{n,j}$.

B. Initialisation

We adopt the initialisation strategy in [10]: at time step n , the algorithm starts the recursive updates from $q_n(X_n)$ and the initial variational distribution $q_n^{(0)}(\theta_n^s)$ for each sensor s is

$$q_n^{(0)}(\theta_n^s) \propto \frac{\Lambda_0^s}{V_s^s} \delta[\theta_{n,j}^s = 0] + \sum_{k=1}^K \Lambda_k^s l_k^{s,0} \delta[\theta_{n,j}^s = k], \quad (16)$$

$$l_k^{s,0} = \mathcal{N}(Y_{n,j}^s; H \mu_{n|n-1}^{k*}, H \Sigma_{n|n-1}^{k*} H^\top + R_k^s),$$

IV. CONSENSUS-BASED DISTRIBUTED VARIATIONAL MULTI-OBJECT TRACKER

In this section, we present distributed variational filtering frameworks for a sensor network without a fusion center. The aim is to achieve the same converged variational distribution $q_n^*(X_n, \theta_n)$, as obtained in the centralised variational filtering framework for approximating the target posterior $\hat{p}_n(X_n, \theta_n | Y_n)$ in (9), by solely relying on local processing and communications between neighbouring sensors. To this end, we assume at the initial time step 0, an identical target state prior $p(X_0)$ is given to all sensors $s = 1, \dots, N_s$. To ensure the variational distribution at each sensor s converges

Algorithm 1: Consensus-based Distributed Variational Multi-object Tracker at time step n

Input: $q_{n-1}^*(X_{n-1}), Y_n, M_n$, maximum iteration I_{max} .

Initialisation: Set $\hat{p}_n(X_n)$ according to (13).

At each sensor s :

Initialise $q_n(\theta_n^s)$ according to (16).

for $i = 1, 2, \dots, I_{max}$ **do**

 Compute $\Omega_{k,1}^s, \Omega_{k,2}^s, k = 1, 2, \dots, K$ using (12).

Perform average consensus with $\hat{\Omega}_{k,1}$ and $\hat{\Omega}_{k,2}$.

for $k = 1, 2, \dots, K$ **do**

 Evaluate \bar{R}_n^k, \bar{Y}_n^k according to (19).

 Update $q_n(X_{n,k})$ by Kalman filtering.

end

 Update $q_n(\theta_{n,j}^s), j = 1, 2, \dots, M_n^s$ using (15).

end

Set $q_n^*(X_n) = \prod_{k=1}^K q_n(X_{n,k})$, and

$q_n^*(\theta_n^s) = \prod_{j=1}^{M_n^s} q_n(\theta_{n,j}^s)$.

to the same $\hat{p}_n(X_n, \theta_n|Y_n)$ at the time step n , according to (11), it requires the local sensor has access to the pseudo-measurements \bar{Y}_n^k and $\bar{R}_n^k, k = 1, 2, \dots, K$ calculated using all values of $\{\Omega_{k,1}^s, \Omega_{k,2}^s\}_{s=1}^{N_s}$ computed at each sensor.

The sum expressions of (12) can be computed at each sensor by using a distributed, iterative consensus algorithm. Specifically, we adopt the distributed average consensus algorithm introduced in [7], which is guaranteed to converge provided that the sensor network is connected, even under time-varying communication links. For our application, given the initial value of $\Omega_{k,1}^s, \Omega_{k,2}^s$ at each sensor s , each sensor can converge to the same average value $\hat{\Omega}_{k,1}^s = \frac{1}{N_s} \sum_{s=1}^{N_s} \Omega_{k,1}^s$ and $\hat{\Omega}_{k,2}^s = \frac{1}{N_s} \sum_{s=1}^{N_s} \Omega_{k,2}^s$. As an example, the distributed average consensus for computing $\hat{\Omega}_{k,1}^s$ at sensor s can be described as follows.

- At initial iteration $m = 0$, each sensor node s initialises its state as $\hat{\Omega}_{k,1}^{(s,0)} = \Omega_{k,1}^s$.
- For $m = 0, 1, 2, \dots$ until convergence each sensor s updates its state by using its own state and the states of instantaneous neighbours $\mathcal{N}_s(m)$:

$$\hat{\Omega}_{k,1}^{(s,m+1)} = W_{ss}^{(m)} \hat{\Omega}_{k,1}^{(s,m)} + \sum_{j \in \mathcal{N}_s(m)} W_{sj}^{(m)} \hat{\Omega}_{k,1}^{(j,m)} \quad (17)$$

where $W_{sj}^{(m)}$ is the linear weight on $\hat{\Omega}_{k,1}^{(j,m)}$ at node s . Here we adopt the Metropolis weight in [7]:

$$W_{sj}^{(m)} = \begin{cases} \frac{1}{1 + \max\{d_s^{(m)}, d_j^{(m)}\}} & \text{if } j \in \mathcal{N}_s^{(m)}, \\ 1 - \sum_{s,k \in \mathcal{E}(m)} W_{sk}^{(m)} & \text{if } j = s \end{cases} \quad (18)$$

In the same way, we can obtain the $\hat{\Omega}_{k,2}^s$ by the same distributed average consensus algorithm. After obtained the converged value of $\hat{\Omega}_{k,1}^s$ and $\hat{\Omega}_{k,2}^s$, at each sensor s , we can compute the required pseudo-measurements \bar{Y}_n^k and $\bar{R}_n^k, k = 1, 2, \dots, K$, by the following expressions:

$$\bar{R}_n^k = (N_s \hat{\Omega}_{k,1}^s)^{-1}, \quad \bar{Y}_n^k = \bar{R}_n^k (N_s \hat{\Omega}_{k,2}^s) \quad (19)$$

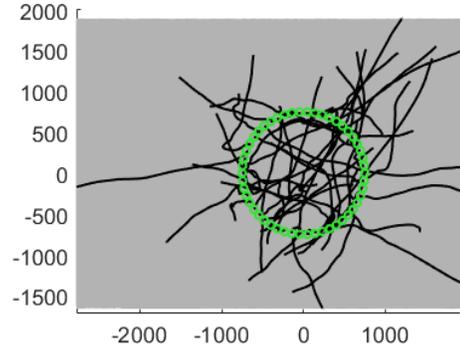


Fig. 1: Measurements, ground truth tracks of 50 targets; grey dots are measurements covering the whole background, black lines are the trajectories and green circles are starting points

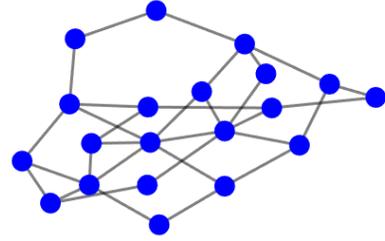


Fig. 2: Simulated sensor network; blue circles are sensors and lines indicate the communication links between sensors

In this way, during each iteration of the update for $q_n(X_n)$, every sensor s updates $q_n(X_n)$ locally based on the centralised pseudo-measurements calculated at all sensors using the distributed consensus algorithm, such that each sensor can behave equivalently to the fusion center in the centralised version. The overall distributed implementation of the variational tracker is summarised in Algorithm 1.

V. RESULTS

In this section, we will conduct a performance comparison between different versions of the variational multi-object tracker: the centralised fusion in Section III, the optimal distributed fusion in Section IV, and the suboptimal distributed fusion with an arithmetic average (AA) fusion strategy. Specifically, the suboptimal distributed fusion adopts the similar approximation in the literature, e.g., [8], in which each sensor infers a multi-object posterior distribution based on local measurements and, then a distributed average consensus algorithm is implemented to fuse the derived multi-object posteriors from each sensors using the AA fusion principle.

In the simulated dataset, the network consists of 20 sensors as shown in Fig. 2, all observing 50 targets in the surveillance area. The general parameter settings are as follows. For all datasets, the total time steps are 50, and the time interval between observations is $\tau = 1$ s. The parameters in the CV model are $\sigma_k = 5$ and $R_k^s = 100I$ where I is a 2-D identity matrix. The target Poisson rates are set to 1; the clutter rate is 100. To evaluate the robustness of the algorithm, we generate

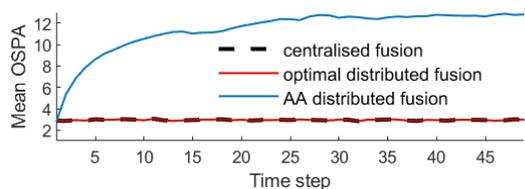


Fig. 3: Mean OSPA of different fusion strategies over 50 time steps, averaged over 20 Monte Carlo runs (average consensus iteration for both proposed optimal distributed fusion and the AA distributed fusion is 20)

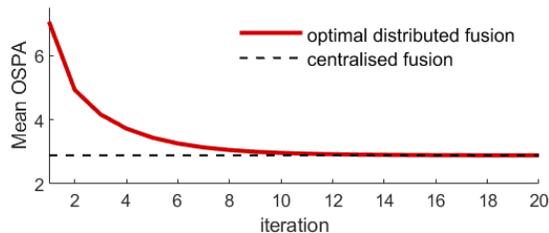


Fig. 4: Mean OSPA of the optimal distributed fusion over different iterations of average consensus algorithm, averaged over 20 Monte Carlo runs

20 different measurement sets under the same parameter settings. One sample measurement set is shown in Figure 1. We use the optimal sub pattern assignment (OSPA) [15] metric to evaluate the tracking performance of all methods. For the OSPA metric, the order is set to $p = 1$ and the distance cut-off value is $c = 50$. For both datasets, we calculate the mean OSPA metric over all the sensors and Monte Carlo runs.

Figure 3 shows the three variational multi-target trackers' mean OSPA of each time step calculated over all the sensors and Monte Carlo runs. Specifically, for both proposed optimal distributed fusion and the suboptimal AA fusion, we set the average consensus iteration to 20 to obtain the results in Figure 3. It is observed that the proposed optimal distributed fusion has a much lower mean OSPA value compared to the suboptimal AA fusion. The estimation results also confirm the equivalence of our proposed optimal distributed variational tracker with the centralised variational tracker when the distributed average consensus reaches convergence. Figure 4 shows mean OSPA over all the sensors, Monte Carlo runs, and 50 time steps versus the number of iterations used in the distributed average consensus algorithm for proposed optimal distributed variational tracker. We can see that as the number of iterations increases, the performance of the optimal distributed fusion approaches the performance of the centralised fusion within approximately 10 iterations.

VI. CONCLUSION

The paper presents a novel optimal distributed variational multi-target tracker for sensor networks that only require communication between neighbouring sensors. Our method achieves equivalent tracking performance to centralised fusion while retaining a decentralised processing architecture and reducing communication costs. The simulation results demon-

strate the equivalence of the proposed optimal distributed fusion and the centralised fusion in terms of tracking accuracy. In the future, we will extend the current distributed variational tracker to handle unknown target numbers and heterogeneous sensor networks with varying coverage.

VII. ACKNOWLEDGEMENTS

This research is sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing under the SIGNeTS project. It is accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions in this document are of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government or the U.K. Government. The U.S. Government and U.K. Government are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

- [1] C.-Y. Chong, "Distributed multitarget multisensor tracking," *Multitarget-multisensor tracking: Advanced applications*, pp. 247–296, 1990.
- [2] —, "Forty years of distributed estimation: A review of noteworthy developments," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, 2017, pp. 1–10.
- [3] R. P. Mahler, "Optimal/robust distributed data fusion: a unified approach," in *Signal Processing, Sensor Fusion, and Target Recognition IX*, vol. 4052. SPIE, 2000, pp. 128–138.
- [4] M. Kayaalp, Y. Inan, E. Telatar, and A. H. Sayed, "On the arithmetic and geometric fusion of beliefs for distributed inference," *arXiv preprint arXiv:2204.13741*, 2022.
- [5] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [7] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. IEEE, 2005, pp. 63–70.
- [8] T. Li and F. Hlawatsch, "A distributed particle-phd filter using arithmetic-average fusion of gaussian mixture parameters," *Information Fusion*, vol. 73, pp. 111–124, 2021.
- [9] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4334–4349, 2012.
- [10] R. Gan, Q. Li, and S. Godsill, "A variational Bayes association-based multi-object tracker under the non-homogeneous Poisson measurement process," in *2022 25th International Conference on Information Fusion (FUSION)*. IEEE, 2022, pp. 1–8.
- [11] K. Gilholm, S. Godsill, S. Maskell, and D. Salmund, "Poisson models for extended target and group tracking," in *Signal and Data Processing of Small Targets 2005*, vol. 5913. International Society for Optics and Photonics, 2005, p. 59130R.
- [12] Q. Li, J. Liang, and S. Godsill, "Scalable data association and multi-target tracking under a Poisson mixture measurement process," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5503–5507.
- [13] Q. Li, R. Gan, J. Liang, and S. J. Godsill, "An adaptive and scalable multi-object tracker based on the non-homogeneous Poisson process," *IEEE Transactions on Signal Processing*, 2023.
- [14] Q. Li, B. I. Ahmad, and S. J. Godsill, "Sequential dynamic leadership inference using Bayesian Monte Carlo methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2039–2052, 2021.
- [15] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE transactions on signal processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

Joint Sensor Scheduling and Target Tracking with Efficient Bayesian Optimisation

Xingchi Liu^{*1}, Chenyi Lyu^{*1}, Seyed Ahmad Soleymani², Wenwu Wang², Lyudmila Mihaylova¹

¹Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK

²Centre for Vision Speech and Signal Processing (CVSSP), University of Surrey, Guildford, UK.

Email: xingchi.liu@sheffield.ac.uk, clyu5@sheffield.ac.uk, s.soleymani@surrey.ac.uk,
w.wang@surrey.ac.uk, l.s.mihaylova@sheffield.ac.uk

Abstract—The received signal strength measurement has been widely used in search and tracking applications and its benefit is linked with the distance between the transmitter and receiver. This paper proposes an online Bayesian optimisation-based approach that relies on signal strength measurements to schedule multiple sensors for searching and tracking a moving target, without any prior knowledge of the target’s state or motion model. A unique contribution lies in incorporating the Gaussian processes factorisation method into the Bayesian optimisation framework, which enhances the effectiveness of the proposed approach. Numerical results obtained from different sizes of measurements demonstrate that the proposed approach can efficiently schedule two unmanned aerial vehicles. Particularly, it achieves at most 21% lower computational time for deciding measurement locations and 79% lower time for updating the surrogate model as compared to the benchmark approach.

Index Terms—Active sensing, Bayesian optimisation, factorised Gaussian process, target tracking, sensor management, unmanned aerial vehicles, hierarchical off-diagonal low-rank (HODLR) factorisation

I. INTRODUCTION

Target tracking is crucial for applications including sea surveillance, autonomous vehicles, and traffic monitoring. Model-based and data-driven approaches have been proposed for this challenge, dealing with data association, group/extended object, and sensor management. However, many approaches rely on informative prior state beliefs which may be unavailable in scenarios like search and rescue or wildlife monitoring. In such cases, the active position estimation [1] becomes a significant challenge as the active sensing platform must locate and track the target simultaneously.

One way to detect and track targets is by analysing the received signal strength (RSS). By measuring the RSS, the distance between the sensor and the target can be estimated. This distance can then be used to track the target over time. Moreover, changes in the RSS signal can provide additional information about the target, such as its velocity and direction of movement [2]. Therefore, active sensing using RSS signals has become an important research area in target tracking and has demonstrated promising results in various applications.

There are three primary types of active sensing techniques for searching and tracking targets, based on the RSS-distance

relationship. Geometric approaches [3] rely on model inversion and trilateration, which requires at least three receivers to estimate the target’s position accurately. Statistical strategies [4], [5] are designed to account for both model inaccuracies and measurement noise by treating RSS measurements as random variables and applying statistical filtering techniques, such as the Kalman filter and particle filter, to refine raw data points. Data-driven methods leverage machine learning models, such as neural networks [6] or Gaussian processes (GP) [7], to model the RSS-distance relationship, and optimise their parameters during the training process. In this paper, we follow the data-driven idea to design a Bayesian optimisation (BO)-based probabilistic search and tracking approach.

BO is a machine learning-based optimisation method that involves building a surrogate model for the objective function, along with prediction uncertainty quantification using GP. BO iteratively locates the global optimum by using an acquisition function (AF) defined over the surrogate. BO has been applied to solve active sensing and path planning problems [8]–[10]. In these works, GP represents the RSS-distance relationship and AF is used to decide where to place the sensors to collect new measurements and plan the path of the unmanned aerial vehicles (UAVs). However, most existing works focused on modelling stationary processes (e.g., searching for static targets or planning based on static environments). BO can also be used to solve dynamic and non-stationary optimisation problems [11]–[13] by designing kernel functions to account for non-stationary and time-varying processes.

While BO shows promise for solving active sensing problems, its computational complexity cannot be ignored, especially when a large number of measurements are collected by the sensing platform. This is due to the $n \times n$ covariance matrix of GP, which incurs a significant cubic computational cost ($\mathcal{O}(n^3)$) with respect to the number of measurements n . This poses challenges for real-time active sensing since both GP model updates and AF optimisation can be time-consuming, which is related to inversion and determinant evaluation of the covariance matrix. To reduce the computational complexity, this paper explores the idea of factorising the dense covariance matrix into data-sparse hierarchical off-diagonal matrices [14], [15]. This structure can provide a very close approximation to the Cholesky factorisation method with only $\mathcal{O}(n \log n^2)$ cost.

*Equal contribution.

Moreover, the computational complexity was further reduced to $\mathcal{O}(n \log n)$ with a designed low-rank approximation method.

This paper proposes a BO-assisted approach for active sensing management to search and track a moving target. In contrast to our previous work [16] that focused on distributed tracking, this study emphasises active tracking without any prior position information. The main contribution is two-fold. First, a spatial-temporal composite kernel function is designed to account for the non-stationary and time-varying nature of the RSS map. Moreover, several techniques are introduced to reduce the computational cost of the GP used in BO. The proposed approach can schedule multiple UAVs to perform efficient area search and can also be applied to activate sensors over time in sensor management problems [17].

The paper is structured as follows: In Section II, the problem formulations and the fundamentals of BO are introduced. Section III presents the proposed search and tracking approach. Section IV presents the simulation results, while Section V summarises the conclusions.

II. PROBLEM FORMULATION

We first model the RSS as a black-box function of the coordinates of the measuring location and the time. Define the location of measuring the RSS in time t as $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^2$, where \mathcal{X} is the area of interest. Denote y as the measurement, a black-box dynamic function can be represented as

$$y = f(\mathbf{x}_t, t) + \epsilon, \quad (1)$$

where ϵ is the measurement noise that is assumed to follow a zero-mean Gaussian distribution with variance σ^2 . Since the expected value of an RSS measurement is related to the proximity between the target and the sensor, the location with the highest expected value of RSS measurements is identified as the target location. The task of searching and tracking a target over time is equivalent to solving a dynamic optimisation problem [18], specifically finding the maximum of this function. This optimisation problem can be formulated as

$$\max f(\mathbf{x}_t, t), \quad (2)$$

$$\text{s.t. } \mathbf{x}_t \in \mathcal{X}, t \in \mathcal{T}, \quad (3)$$

where \mathcal{X} and \mathcal{T} are the spatial and temporal search spaces, respectively. Next, we describe the GP that represents the black-box function. The UAVs positions are optimised based on the objective AF function.

A. Gaussian Process Regression

The unknown function $f(\mathbf{x}_t, t)$ is a black-box function lacking an analytical form. Therefore, a surrogate model, namely GP, is utilised to represent this function for two reasons. First, GP can quantify the uncertainty of the learned RSSs in a principled way, aiding the exploration-exploitation (EE) tradeoff for maximisation (see more details in the next section). Second, GP functions well with small volumes of data and is particularly useful in the early stages of the search process where few RSS measurements are available

for building the surrogate. The GP that is placed as a prior distribution of the function $f(\mathbf{x}_t, t)$ can be written as

$$f(\mathbf{x}_t, t) \sim \mathcal{GP}(m(\mathbf{x}_t, t), k((\mathbf{x}_t, t), (\mathbf{x}'_t, t'))), \quad (4)$$

where (\mathbf{x}_t, t) and (\mathbf{x}'_t, t') are either the training or the testing input data. $m(\mathbf{x}_t, t)$ and $k((\mathbf{x}_t, t), (\mathbf{x}'_t, t'))$ denote the mean and the covariance functions of GP, respectively.

Suppose that by the time t , n_t RSS measurements have been received with time stamps t_1, t_2, \dots, t_{n_t} . Define \mathbf{x}_{t_i} as the location associated with the measurement at t_i , where $t_i \leq t$. In addition, define y_{t_i} as the measurement at t_i . Therefore, at any t , we can have a set of 3-tuple that can be denoted as $\mathcal{D}_t = \{\mathbf{x}_{t_i}, t_i, y_{t_i}\}_{i=1}^{n_t}$. Given \mathcal{D}_t , define \mathbf{K}_t as a covariance matrix with the (i, j) th entry as $k((\mathbf{x}_{t_i}, t_i), (\mathbf{x}_{t_j}, t_j))$. In addition, define \mathbf{k}_* as a vector with the j th entry as $k((\mathbf{x}_{t_j}, t_j), (\mathbf{x}_*, t_*))$. Denote the set of measurements received until t by $\mathbf{y}_t = [y_{t_1}, y_{t_2}, \dots, y_{t_{n_t}}]^\top$. The GP predictive distribution at a new input (\mathbf{x}_*, t_*) can be written as

$$\mu_* = m(\mathbf{x}_*, t_*) + \mathbf{k}_*^\top (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} (\mathbf{y}_t - m(\mathbf{x}_*, t_*)), \quad (5)$$

$$\sigma_*^2 = k((\mathbf{x}_*, t_*), (\mathbf{x}_*, t_*)) - \mathbf{k}_*^\top (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*, \quad (6)$$

where μ_* and σ_*^2 denote the posterior predictive mean and variance of the unknown function at (\mathbf{x}_*, t_*) , respectively.

The hyperparameters of GP are learned from the data by maximising the log marginal likelihood that can be written as

$$\log p(\mathbf{y}_t | \mathcal{D}_t, \boldsymbol{\theta}) = -1/2 \mathbf{y}_t^\top (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t - 1/2 \log |\mathbf{K}_t + \sigma^2 \mathbf{I}| - n_t/2 \log 2\pi, \quad (7)$$

where $\boldsymbol{\theta}$ represents the set of hyperparameters.

B. Acquisition Function

Selecting measuring points to evaluate the unknown function sequentially and locate the moving target efficiently with minimum measurements is challenging. There exists an EE dilemma in this decision-making process: Exploring the unknown function provides knowledge but may lead to low search efficiency. However, exploiting the learned knowledge may miss the opportunity to measure higher RSS from under-explored areas. To address this, an AF [19] is optimised to determine measuring points while balancing the EE.

The selection strategy of the next measuring point depends on the type of AF. Here we apply the expected improvement (EI) function [20]. The objective is to find the next measuring point with the highest EI as compared to the incumbent measurement that can be defined as $\tau_{n_t} = \max_{i \in \{1, 2, \dots, n_t\}} \mathcal{Y}_{t_i}$. The EI function can be written as

$$\begin{aligned} \alpha_{\text{EI}}(\mathbf{x}_t, t) &:= \mathbb{E}[[f(\mathbf{x}_t, t) - \tau_{n_t}]^+], \\ &= \sigma(\mathbf{x}_t, t) \phi\left(\frac{\Delta(\mathbf{x}_t, t)}{\sigma(\mathbf{x}_t, t)}\right) + \Delta(\mathbf{x}_t, t) \Phi\left(\frac{\Delta(\mathbf{x}_t, t)}{\sigma(\mathbf{x}_t, t)}\right), \end{aligned} \quad (8)$$

where $\mathbb{E}(\cdot)$ represents the mathematical expectation operation. $\Delta(\mathbf{x}_t, t) = \mu(\mathbf{x}_t, t) - \tau_{n_t}$ is the expected difference between the predicted RSS at a point and the incumbent target. Here $\phi(\cdot)$ and $\Phi(\cdot)$ denote the probability density and cumulative density functions, respectively. In (8), the predictive standard

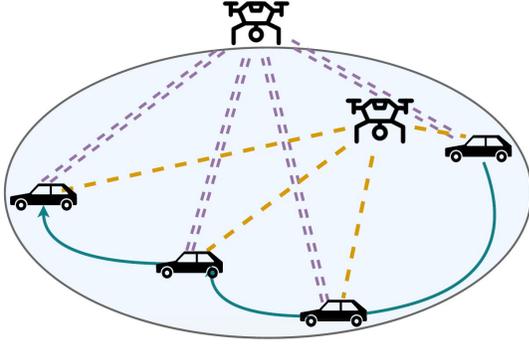


Fig. 1: UAV-based searching and tracking

deviation affects the first term and the predictive mean affects the second term. By maximising the EI function, the EE trade-off can be well-balanced. In this work, the AF is maximised using the grid search method.

III. EFFICIENT BO-ASSISTED SEARCH AND TRACKING

Assuming UAVs can fly over the area of interest to measure RSS from the moving target, building an expressive and scalable surrogate (GP) for the dynamic function is a critical challenge. This section describes the proposed composite kernel and GP factorisation to address these challenges.

A. Kernel Design

Inspired by [13], a spatial-temporal composite kernel

$$k((\mathbf{x}_t, t), (\mathbf{x}'_t, t')) = k_S(\mathbf{x}_t, \mathbf{x}'_t) \cdot k_T(t, t'), \quad (9)$$

is introduced to capture both the spatial and temporal correlations in the unknown time-varying function, where $k_S(\cdot, \cdot)$ represents the spatial kernel, used for characterising the one-time RSS map, $k_T(\cdot, \cdot)$ represents the temporal kernel which considers the temporal correlations of the RSS.

The *designed GP spatial composite kernel* is a sum of a constant kernel $k_{S,Con}$ and a squared exponential (SE) kernel $k_{S,SE}$. The constant kernel is added considering the fact that the RSS values in a specific area are above a certain level. The SE kernel represents the smooth changes of RSS. For the *temporal kernel*, we choose the Matérn kernel $k_{T,Mat}$ since it includes a large class of kernels and is proven to be very useful for matching physical processes realistically.

The kernel function (9) can be rewritten as

$$k((\mathbf{x}_t, t), (\mathbf{x}'_t, t')) = (k_{S,Con}(\mathbf{x}_t, \mathbf{x}'_t) + k_{S,SE}(\mathbf{x}_t, \mathbf{x}'_t)) \cdot k_{T,Mat}(t, t'), \quad (10)$$

$$k_{S,Con}(\mathbf{x}_t, \mathbf{x}'_t) = \Phi, \quad (11)$$

$$k_{S,SE}(\mathbf{x}_t, \mathbf{x}'_t) = \sigma_m^2 \exp(-\|\mathbf{x}_t - \mathbf{x}'_t\|^2 / l^2), \quad (12)$$

$$k_{T,Mat}(t, t') = \sigma_m^2 \frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2v}\|t-t'\|}{l} \right)^v K_v \left(\frac{\sqrt{2v}\|t-t'\|}{l} \right) \quad (13)$$

with σ_m^2 and l being the amplitude and length scale parameters, respectively, Φ represents a constant. $K_v(\cdot)$ is a modified Bessel function and $\Gamma(\cdot)$ is a Gamma function. Moreover, v is a smoothness parameter of Matérn kernel. Different functions belonging to the Matérn kernel can be built with varying v .

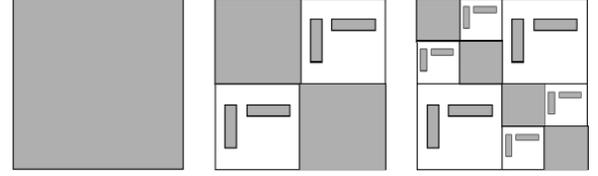


Fig. 2: HODLR matrix at different levels

B. GP Factorisation

This section introduces a novel approach that enhances the BO efficiency by integrating the hierarchical off-diagonal low-rank (HODLR) structure [21] into the GP surrogate model of the BO framework. The HODLR structure divides recursively the covariance matrix and applies rank-revealing lower-upper factorisation hierarchically to certain sub-matrices in the off-diagonal section, while retaining the diagonal parts. Consequently, a subset of columns, rows, or entries is formed, which is much quicker to calculate than the entire matrix. Generally, constructing the HODLR matrix requires a cost of $\mathcal{O}(n \log^2(n))$. Fig. 2 gives a graphical representation of the HODLR matrices. A real symmetric matrix $\mathbf{K}_t \in \mathbb{R}^{n_t \times n_t}$ can be decomposed to a two-level HODLR matrix:

$$\mathbf{K}_t = \begin{bmatrix} \mathbf{K}_{t,1} & \mathbf{U}_1 \mathbf{V}_1^T \\ \mathbf{V}_1 \mathbf{U}_1^T & \mathbf{K}_{t,2} \end{bmatrix}, \quad (14)$$

with the diagonal blocks decomposed into

$$\mathbf{K}_{t,1} = \begin{bmatrix} \mathbf{K}_{t,1}^{(2)} & \mathbf{U}_1^{(2)} \mathbf{V}_1^{(2)T} \\ \mathbf{V}_1^{(2)} \mathbf{U}_1^{(2)T} & \mathbf{K}_{t,2}^{(2)} \end{bmatrix}, \quad (15)$$

$$\mathbf{K}_{t,2} = \begin{bmatrix} \mathbf{K}_{t,3}^{(2)} & \mathbf{U}_2^{(2)} \mathbf{V}_2^{(2)T} \\ \mathbf{V}_2^{(2)} \mathbf{U}_2^{(2)T} & \mathbf{K}_{t,4}^{(2)} \end{bmatrix}.$$

The $\mathbf{K}_{t,1}$ and $\mathbf{K}_{t,2}$ are the $n/2^j \times n/2^j$ diagonal block matrices from the original matrix \mathbf{K}_t and $\mathbf{U}^{(j)}$, $\mathbf{V}^{(j)}$ matrices are $n/2^j \times r$ matrices with $r \ll n$. j denotes the level of decomposition, which are 2 in this example, and rank r depends on the desired accuracy of the low-rank approximation. A higher rank results in less precision loss and higher computation cost.

Given an HODLR-type factorisation, rapid computation of the inverse of the entire matrix is allowed via the Sherman-Morrison-Woodbury formula [14]. This method has a computational complexity of $\mathcal{O}(n \log n)$ for both solving linear systems and computing the determinant, satisfying the requirements of real-time implementation.

C. Algorithm Overview

The proposed approach requires an initial set of data \mathcal{D}_0 and a GP prior $\mathcal{G}\mathcal{P}_0$ as the surrogate of the unknown dynamic function. The initial input data can be randomly sampled from the search space which is an area of the location \mathbf{x} and a period of time t . The GP is used to construct the AF that leads to searching for the maximum RSS location over time. Besides, we introduce a superscript $k \in \{1, 2, \dots, K\}$ to represent different UAVs, where K is the number of UAVs. In addition, at any time t_i , define the spatial search space as the whole area

of interest and the temporal search space as $\mathbf{t} = [t_s, t_s + \gamma]$, where $t_s = t_i + \psi$. A large γ value means a large time interval for one UAV to decide when to collect a measurement, thereby reducing the number of measurements when the total search time is fixed. The proposed algorithm works in an iterative process, the UAVs are scheduled to collect measurements and send them to an edge node which then updates GP and determines new points for UAVs to measure. The proposed algorithm will terminate after a certain period of time T .

The detailed process is described as Algorithm 1.

Algorithm 1 BO-assisted joint sensor scheduling and tracking

Require: Prior surrogate model \mathcal{GP}_0 , initial data \mathcal{D}_0 , UAV number K

- 1: **while** $t_i \leq T$ **do**
- 2: Receive the K RSS measurements
- 3: Set the time stamp $t_i = \max\{t_i^1, t_i^2, \dots, t_i^K\}$
- 4: Augment data $\mathcal{D}_i \leftarrow \mathcal{D}_{i-1} \cup \{\mathbf{x}_{t_i}^k, t_i^k, y_{t_i}^k\}_{k=1}^K$
- 5: Generate the HODLR structure for covariance matrix
- 6: Update \mathcal{GP}_i by maximising (7)
- 7: Set the start time stamp $t_s \leftarrow t_i + \psi$
- 8: Update search bound of time interval as $\mathbf{t} = [t_s, t_s + \gamma]$
- 9: Determine $\{\mathbf{x}_{t_{i+1}}^k\}_{k=1}^K$ and $\{t_{i+1}^k\}_{k=1}^K$ by sequentially maximising AF as follows:

$$\{\mathbf{x}_{t_{i+1}}^k, t_{i+1}^k\} = \arg \max_{\mathbf{x}_t \in \mathcal{X}, t \in \mathbf{t}} \alpha^k(\mathbf{x}_t, t)$$

- 10: Send the UAVs to measure the RSSs at $\{\mathbf{x}_{t_{i+1}}^k, t_{i+1}^k\}_{k=1}^K$
- 11: $i \leftarrow i + 1$
- 12: **end while**

IV. NUMERICAL RESULTS

A. Simulation Settings

The log-distance path-loss model [22]

$$y_{t_i} = y_{0,t_i} - \eta \log_{10}(d_{t_i}) + \epsilon, \tag{16}$$

is used to generate RSS measurements, where y_{0,t_i} is a constant characterising the transmission power at t_i with the unit of dBm. An RSS y_{t_i} of a target measured by a UAV. The distance between the target and the UAV at t_i is defined as d_{t_i} . Sensor distortions and environmental interference are represented by ϵ , which is assumed to be a zero-mean Gaussian noise. The proposed algorithm is validated by setting the standard deviation of the Gaussian noise as one dB. Here η is the attenuation gain. We tested the proposed approach in a 400m x 400m area with a target trajectory based on the constant velocity model and an initial state vector of $[50m, 1m/s, 50m, 1m/s]$. Two UAVs are used for tracking, and two BO-based approaches are implemented with different factorisation methods. We also study the impact of the level of HODLR factorisation determined by the block size parameter (it controls the size of the dense diagonal grey block shown in Fig. 2). As a general rule, a smaller block size leads to a higher level of factorisation, which brings faster computation

TABLE I: One-step computational time

γ	Factorisation method	GP update (sec)	AF maximisation (sec)	
			1st UAV	2nd UAV
$\gamma = 1$	HODLR	0.86	0.61	0.62
	Cholesky	0.97	0.77	0.77
$\gamma = 2$	HODLR	0.29	0.72	0.73
	Cholesky	0.49	0.84	0.85
$\gamma = 4$	HODLR	0.04	0.79	0.80
	Cholesky	0.19	0.96	0.97

at a cost of lower accuracy than the full GP. All the results are averaged over 100 Monte Carlo simulations.

B. Computational Time

The proposed HODLR factorisation-based BO is compared to Cholesky factorisation-based BO in terms of computational time. Changing γ controls measurement size and evaluates approach efficiency. A large γ value reduces the frequency of collecting measurements and also means not very frequent AF maximisation and GP updates, leading to long search intervals for one-step operation (based on lines 3, 8, and 9 in Algorithm 1). The average received numbers of measurements in the three cases are 424, 268 and 162, respectively. To ensure fairness, we calculate the time required for one-step AF maximisation and GP update. Table I shows that increasing γ decreases GP update time, while AF maximisation time increases due to less efficient grid searching with more grid points required. The AF maximisation time for each UAV is presented separately, as it can be done asynchronously. The proposed HODLR factorisation-based approach outperforms the Cholesky-based approach in terms of shorter time for both AF maximisation and GP update. Specifically, the HODLR factorisation reduces 79% of surrogate model update time when $\gamma = 4$ and up to 21% of AF maximisation time.

C. Tracking Error

The tracking errors of both approaches are presented in Fig. 3. The proposed approach with HODLR factorisation achieves slightly higher errors than the Cholesky factorisation due to a more sparse representation of the covariance matrix. However, it still performs competitively well as compared to the Cholesky factorisation. Particularly, the gap between the two approaches becomes smaller with less training data.

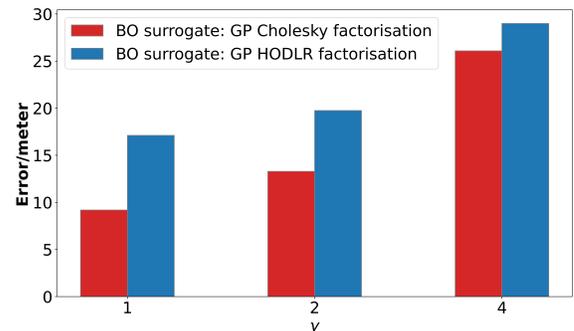


Fig. 3: Tracking error versus γ time step in the BO search (as in Algorithm 1)

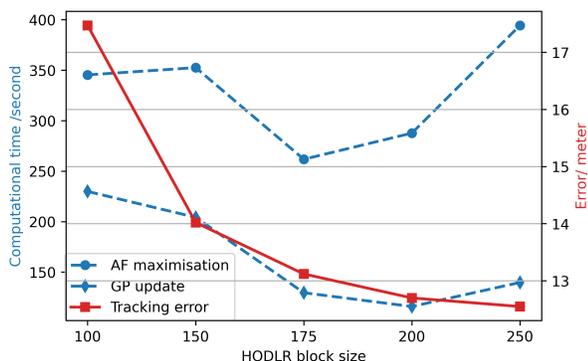


Fig. 4: Tracking error and computational time versus HODLR block size

D. HODLR Factorisation Level

The total computational time and the tracking error affected by the HODLR factorisation level are presented in Fig. 4. Here the HODLR block size is changed while γ is fixed to be one. The implementation of the algorithms was performed in Python, on a PC with i7-12700h CPU. The results reveal that although smaller block sizes do incur higher errors, the algorithm still has efficient computational time. High computational times at small block sizes are due to the computational overhead in constructing the HODLR matrix. The results highlight the importance of choosing the proper HODLR block size to achieve the best computational efficiency. Noted that better computational efficiency results could be achieved by different implementations of the proposed approach.

V. CONCLUSION

This paper proposes a novel joint sensor scheduling and target tracking approach to send multiple UAVs to track a moving target using RSS measurements. A spatial-temporal composite kernel comprised of a constant kernel, a squared exponential and a Matérn kernel is designed. Then a GP surrogate model for the latent process of RSS generation is constructed that varies over time. Particularly, the HODLR factorisation is integrated into the proposed algorithm to improve its efficiency. Numerical results confirm that the proposed HODLR factorisation-based BO reduces the running time as compared to the standard Cholesky factorisation-based BO while achieving competitive tracking accuracy. Future work will focus on developing non-myopic strategies to solve the sensor scheduling problem.

ACKNOWLEDGMENTS

The research was sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing, and was accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government or the U.K. Government. The U.S. Government and U.K. Government are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. We acknowledge the support from the EPSRC through EP/T013265/1 project NSF-EPSRC: “ShiRAS. Towards Safe

and Reliable Autonomy in Sensor Driven Systems” and the support for ShiRAS by the USA National Science Foundation under Grant NSF ECCS 1903466.

REFERENCES

- [1] L. Varotto, A. Cenedese, and A. Cavallaro, “Active sensing for search and tracking: A review,” *arXiv preprint arXiv:2112.02381*, 2021.
- [2] O. Kaltiokallio, R. Hostettler, and N. Patwari, “A novel Bayesian filter for RSS-based device-free localization and tracking,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 780–795, 2021.
- [3] V. C. Paterna, A. C. Auge, J. P. Aspas, and M. P. Bullones, “A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and Kalman filtering,” *Sensors*, vol. 17, no. 12, p. 2927, 2017.
- [4] J. Röbesaat, P. Zhang, M. Abdelaal, and O. Theel, “An improved BLE indoor localization with Kalman-based fusion: An experimental study,” *Sensors*, vol. 17, no. 5, p. 951, 2017.
- [5] A. Cenedese, G. Ortolan, and M. Bertinato, “Low-density wireless sensor networks for localization and tracking in critical environments,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2951–2962, 2010.
- [6] X. Liu, P. Li, and Z. Zhu, “Bayesian optimisation-assisted neural network training technique for radio localisation,” in *Proceedings of the 2022 95th IEEE Vehicular Technology Conference*, 2022, pp. 1–5.
- [7] F. Yin and F. Gunnarsson, “Distributed recursive Gaussian processes for RSS map applied to target tracking,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 492–503, 2017.
- [8] M. Carpin, S. Rosati, M. E. Khan, and B. Rimoldi, “UAVs using Bayesian optimization to locate WiFi devices,” *arXiv:1510.03592*, 2015.
- [9] R. Marchant and F. Ramos, “Bayesian optimisation for intelligent environmental monitoring,” in *Proceedings of the IEEE/RISJ International conf. on Intelligent Robots and Systems*. IEEE, 2012, pp. 2242–2249.
- [10] A. A. Meera, M. Popović, A. Millane, and R. Siegwart, “Obstacle-aware adaptive informative path planning for UAV-based target search,” in *Proceedings of the International Conference on Robotics and Automation*, 2019, pp. 718–724.
- [11] J. Snoek, K. Swersky, R. Zemel, and R. Adams, “Input warping for Bayesian optimization of non-stationary functions,” in *Proceedings of the 31st International Conference on Machine Learning*, 2014.
- [12] R. Martinez-Cantin, “Bayesian optimization with adaptive kernels for robot control,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017, pp. 3350–3356.
- [13] F. M. Nyikosa, M. A. Osborne, and S. J. Roberts, “Bayesian optimization for dynamic problems,” *arXiv preprint arXiv:1803.03432*, 2018.
- [14] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O’Neil, “Fast direct methods for Gaussian processes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 252–265, 2016.
- [15] A. Aminfar, S. Ambikasaran, and E. Darve, “A fast block low-rank dense solver with applications to finite-element matrices,” *Journal of Computational Physics*, vol. 304, pp. 170–188, 2016.
- [16] X. Liu, L. Mihaylova, J. George, and T. Pham, “Gaussian process upper confidence bounds in distributed point target tracking over wireless sensor networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 295–310, 2023.
- [17] A. O. Hero and D. Cochran, “Sensor management: Past, present, and future,” *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, 2011.
- [18] C. Cruz, J. R. González, and D. A. Pelta, “Optimization in dynamic environments: a survey on problems, methods and measures,” *Soft Computing*, vol. 15, no. 7, pp. 1427–1448, 2011.
- [19] P. I. Frazier, “A tutorial on Bayesian optimization,” *arXiv preprint arXiv:1807.02811*, 2018.
- [20] J. Mockus, V. Tiesis, and A. Zilinskas, “The application of Bayesian methods for seeking the extremum,” *Towards Global Optimization*, vol. 2, pp. 117–129, 1978.
- [21] C.-T. Pan, “On the existence and computation of rank-revealing LU factorisations,” *Linear Algebra and its Applications*, vol. 316, no. 1-3, pp. 199–222, 2000.
- [22] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. N. Canagarajah, and B. Ristic, “Mobility tracking in cellular networks using particle filtering,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 10, pp. 3589–3599, 2007.

Simulation of Anisoplanatic Turbulence for Images and Videos

D. Vint, G. Di Caterina, P. Kirkland

Centre for Signal & Image Processing

University of Strathclyde

Glasgow, United Kingdom

{david.vint, gaetano.di-caterina, paul.kirkland}@strath.ac.uk

R. A. Lamb, D. Humphreys

Airborne & Space Systems Division

Electronics Division, Leonardo UK Ltd

Edinburgh, United Kingdom

robert.lamb@leonardo.com

Abstract—Turbulence is a common phenomenon in the atmosphere and can generate a variety of distortions in an image. This can cause further image processing tasks to struggle due to lack of detail in the resulting turbulence affected imagery. It is therefore useful to attempt to remove such distortions as a post processing step. However, the development of such algorithms is difficult due to the complex nature of turbulence data acquisition. To alleviate these issues, this paper presents the development of a turbulence simulator that is capable of imparting the effects of a turbulent atmosphere onto clean images and videos. This work also provides a large, publicly available dataset that can be used as a benchmark. The simulator and dataset will be valuable resources in the field of turbulence mitigation. Indeed, the simulator allows researchers to simulate specific turbulent conditions for any application as required; while the dataset provides the ability to make use of turbulent data without the expensive time commitment of simulation.

Index Terms—Atmospheric Turbulence, Simulation, Anisoplanatic, Wave propagation, Dataset

I. INTRODUCTION

Turbulence within the atmosphere is caused by random perturbations in the refractive index of air [1], which can cause light to divert from its intended path as it travels. When incident on a camera sensor, the consequences of these diversions are angle of arrival fluctuations and phase alterations. The resulting image is therefore degraded by spatially varied warping and blur. The processing of such an image can therefore be rather challenging, as the combined warping and blurring result in a loss of image clarity and high frequency detail. Therefore, a common post-acquisition step consists in the attempt to recover such detail [2]–[4]. However, in order to design such post processing algorithms, example data is needed. Very often such post processing methods require not only the turbulence affected images, but also the clean, turbulent free images as reference. This is especially true for the case of deep learning approaches, as paired data is necessary for training purposes. The practical acquisition of such a dataset however is an extremely difficult task, as the lack of control over imaging conditions makes the capture of corresponding ground truth images nearly impossible. Such ground truth images would need to be acquired when turbulence is not present, likely at a different time of day. The

D.V. acknowledged support from the UK EPSRC and Leonardo UK Ltd, Edinburgh

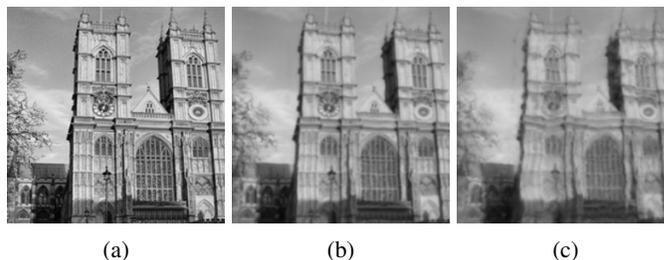


Fig. 1: Example images from simulation. (a) Input Image (b) $C_n^2 = 0.25 \times 10^{-15} \text{m}^{-2/3}$ (c) $C_n^2 = 1.5 \times 10^{-15} \text{m}^{-2/3}$

challenge then becomes the spatial alignment of the camera such that the exact same image is taken.

The ability to simulate the effects of turbulence in software is therefore desirable, as any atmospheric conditions can be evaluated, simulated, and applied to an image. This paper presents the development and implementation of such a simulator for generating anisoplanatic turbulent imagery. Based on the works Schmidt [5] and Hardie et al. [6], the proposed simulator models the turbulent atmosphere as a series of phase screens. A point source is then propagated through the screens to provide a Point Spread Function (PSF), which describes how the point source has spread throughout the atmosphere. This is then applied to a clean image to provide the final turbulent output. The simulator is controlled by a series of input parameters including propagation distance, camera aperture, and the strength of the turbulence, where the turbulence strength is defined by the refractive index structure parameter, C_n^2 . Example outputs of the simulator can be seen in Fig.1. The simulator presented in this paper was developed in MATLAB and is able to produce turbulent images as well as videos, where the speed and direction of the turbulence can also be defined. This work also presents a novel dataset generated through the proposed simulator. This dataset is available for use by the general public and will be a useful resource for future research on turbulence mitigation algorithms¹.

The remainder of the paper is organised as follows. Section

¹All data underpinning this publication are openly available from the University of Strathclyde KnowledgeBase at <https://doi.org/10.15129/1adfbc5c-68f0-49f1-9bad-e64872f9f582>.

II provides related works in the field of turbulence simulation. Section III-A details the underlying theory of turbulence and its characterisation. Section III-B then goes into the details of the simulator, describing the generation of the phase screens (III-B1) and the propagation process (III-B2). Section III-C describes the dataset presented with this work, with details on the simulator settings. Finally, Section IV provides the conclusion to the paper.

II. RELATED WORKS

When simulating the effects of turbulence, there is a trade-off between simulation accuracy and processing time. To ensure accurate simulation, a large number of calculations are needed, therefore increasing simulation time. Other simulation methods instead make use of prior knowledge regarding turbulence theory that allows an estimation of image degradation. These implementations are faster, however they have limited accuracy. The distortion of an image can be described by

$$I_{out}(x, y) = H[I_{in}(x, y)] + n(x, y) \quad (1)$$

where I_{in} and I_{out} are the clean input and distorted output images respectively. Here n is additive noise and $H[\cdot]$ is a function that describes the type of distortion. In the case of turbulent imagery, the function $H[\cdot]$ can be represented by a spatially varying warp, as well as a blurring operation [7], [8]. The simplest method of turbulence simulation can be achieved by applying (1) as a set of random functions, where the distortion function can be represented by a random shift of pixels, and the noise is represented by Gaussian blur [9]. The most common approach to the simulation of turbulence is that of estimating the PSF caused by a turbulent atmosphere [7], [10]. This can be done using an optical transfer function (OTF), which describes how a source of light is affected by the turbulent atmosphere and camera system [11].

Whilst the theoretical OTF can provide an indication of a turbulent volume, the most accurate method of depicting the path of light is by implementing a propagation simulation [12]. These methods model the atmosphere using a series of complex planes called phase screens, which represent how the light wave changes path as it travels [13], [14]. Although the propagation of a single light wave can be achieved with minimal computation, the aggregate time required to perform a propagation for each pixel in an image can result in a computationally expensive simulation.

As previously described, the practical acquisition of turbulent data is a challenging task. However, by combining practical acquisition with computer simulation, real data can be utilised. A popular method of simulation is the work of Repasi et al. [15], which was able to extract typical distortions caused by different levels of turbulence. This allows their simulation method to draw upon real turbulent experience when processing an image. This method has also been used in [16], [17]. Computer simulation is not the only method of recreating turbulence. Multiple works have made use of real life phenomena and real time photo capture in order to obtain turbulent data. This includes imaging through a heating vent

[18], gas hobs [3] and hair dryers [19]. Whilst these methods introduce a form of control over the underlying turbulent effects, they are not true simulations of turbulence over a long propagation path and are therefore only a simplistic imitation of the real effects of turbulence.

Despite the challenges, attempts have been made to produce a real turbulence dataset, such as the work of Gilles et al. [20], which formulates ground truth images via a downsampling and registration process of the pristine image. Anantrasirichai et al. [3] have collated real data for the purpose of turbulence mitigation. To overcome the lack of ground truth, a no reference image quality metric is used to evaluate the performance of their methods. Turbulent data can also be acquired for the purpose of atmosphere analysis, such as in [1], where scintillometers are used to provide a reference value of C_n^2 for the current atmosphere.

III. TURBULENCE SIMULATION

A. Turbulence Theory

Any viscous fluid can be categorised into one of two states of motion: laminar or turbulent. In the case of laminar flow, no mixing of the fluid occurs, resulting in a predictable flow that has known characteristics. The flow becomes turbulent when mixing occurs, causing the flow to break into subflows called turbulent eddies, where any predictable characteristics are therefore lost [21]. The point at which a fluid can transition from laminar to turbulent flow can be determined by the Reynolds number, Re , defined as

$$Re = \frac{Vl}{\nu}$$

where V , l and ν are the velocity, length and kinematic viscosity of the fluid respectively. At low values of the Reynolds number, fluids tend towards laminar flow, whilst high Reynolds numbers characterise turbulent flow [21]. By considering the atmosphere as a fluid, the same principles can be applied in the case of turbulence modelling.

Once a fluid transitions from laminar into turbulent flow, the resulting turbulent subflows can then be described as an energy cascade [22]. This begins with large eddies forming due to an injection of energy; these then proceed to break up into smaller eddies and continue to reduce until the eddies dissipate completely as heat. This cascade begins at a size L_0 and reduces in size to l_0 . These two values are known as the inner and outer scales of turbulence. The eddies that lie within these two scales form what is known as the ‘inertial subrange’ [21]. Within this inertial subrange, assumptions can be made with regard to the statistics of the atmospheric field, which led Kolmogorov to build his statistical model of turbulence [23]. By using these assumptions and dimensional analysis, Kolmogorov derived a power spectral density (PSD) for the changes in refractive index in air:

$$\Phi_n^K(\kappa) = 0.033C_n^2\kappa^{-11/3}, \quad 1/L_0 \ll \kappa \ll 1/l_0 \quad (2)$$

where κ is the angular spatial frequency in rad/m. Typical values of C_n^2 range from 1×10^{-16} (weak) to 1×10^{-13}

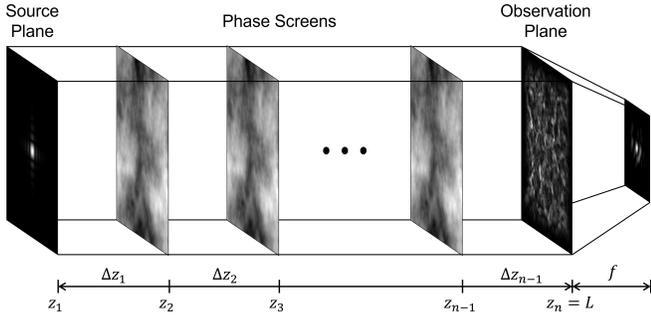


Fig. 2: Illustration of a single propagation. The point source is propagated through each phase screen until the observation plane is reached. The lens operation then converts the complex plane into a Point Spread Function.

(strong). Other models for the refractive PSD introduce additional control parameters to better align the theoretical and analytical experiments [5]. Such a model is the modified Von Karman PSD, which is the PSD used for this simulator. It is evaluated as

$$\Phi_{\phi_i}^{mvK}(f) = \frac{0.023e^{-f^2/f_m^2}}{r_{0_i}^{5/3}(f^2 + f_0^2)^{11/6}} \quad (3)$$

where f is the angular spatial frequency in cycles/m. $f_m = 5.92/2\pi l_0$ and $f_0 = 1/L_0$. Unlike (2), this is evaluated with respect to the Fried parameter of the i^{th} screen, r_{0_i} , which is a measure of optical transmission quality [24].

B. Simulator Model

To simulate the effects of a turbulent atmosphere on images and videos, the atmosphere must first be modelled in 3D space. A point source can then be propagated from the source plane, along a distance L until it reaches the camera sensor (Observation Plane). To represent a volume of turbulent atmosphere, it is common to treat the atmosphere as a series of discrete layers [5], where each layer is represented by a 2 dimensional phase screen. The result of propagation through these phase screens is a complex matrix at the observation plane. Using a lens operation, this complex plane is transformed into a single PSF [6], which can then be applied to an image via convolution. This process of propagation is illustrated in Fig.2. The propagation of a single point source through phase screens is known as isoplanatic simulation. In such a case, the resultant PSF is applied to each pixel within the source image (i.e. spatially invariant). This therefore assumes that each pixel in the image has passed through the same volume of turbulence (or that the turbulence is identical in all points in the 3D space). This however is not the case in real imagery as each source of light traces a different path through the atmosphere. The modelling of such an environment is known as anisoplanatic simulation, in which each pixel has a specific PSF based on its optical path through the atmosphere. To achieve this, the phase screens are generated at an extended size, as seen in Fig.3. Once the trajectory of a pixel (pencil ray)

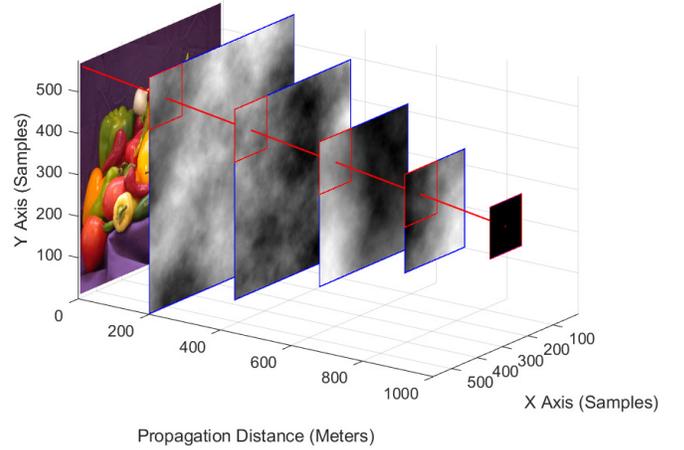


Fig. 3: Geometry of 3D space in which phase screens are situated. For each pixel, a pencil ray is traced towards the observation plane. The intersections of the pencil rays correspond to the centre points for the cropped screens used for propagation.

is traced through 3D space, the intersections with the phase screens are taken as centre points for a cropping operation. The cropped screens are then used for propagation, as in Fig.2. For the simulation of video sequences, the phase screens are again generated to an extended size. The video is then simulated one frame at a time, where the phase screens are translated laterally, to simulate turbulence motion, by a number of samples in a given direction for each new frame, resulting in frames that are temporally correlated.

In order to sample the planes such that the simulation is accurate, the actual physical dimensions of the simulation are used. In the case of this simulator, the size of the planes is related to the diameter D of the camera aperture. From this, the width X of the point source and cropped phase screens is defined as $X = sD = \Delta x N$, where s is a scaling parameter, Δx is the grid spacing and N is the sample count. Δx is calculated such that it is able to accurately represent each screen without undersampling. To ensure this, the Voel critical sampling rule is applied [5], as

$$\Delta x = \left(\frac{\lambda L}{N} \right)^{\frac{1}{2}} \quad (4)$$

allowing the sample count N to be evaluated as

$$N = \frac{X^2}{\lambda L} \quad (5)$$

The scaling parameter s is chosen such that the resulting value of N is a power of two and the screen width is at least 4 times that of the aperture diameter. The image at the source plane is sampled according to Nyquist as $\delta_o = \lambda L/(2D)$ [6].

1) *Generation of Phase Screens*: Each phase screen imparts an optical phase change to the incoming wave. Assuming that

this phase is a Fourier transformable function, it can be written as a Fourier series [5] as

$$\phi(x, y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{c}_{n,m} e^{i2\pi(f_{x_n}x + f_{y_m}y)}$$

where f_{x_n} and f_{y_m} are the spatial frequencies. $\hat{c}_{n,m}$ is the random realisation of the Fourier series coefficients $c_{n,m}$, which are multiplied by a Gaussian random variable with zero mean and unit variance as

$$\hat{c}_{n,m} = \mathcal{N}(f_{x_n}, f_{y_m} | 0, 1) c_{n,m}$$

The values of $c_{n,m}$ are generated from the modified von Karman PSD (3) as

$$c_{n,m} = \sqrt{\Phi_{\phi_i}(f_{x_n}, f_{y_m})\Delta f}$$

where the frequency spacing is $\Delta f = 1/(N\Delta)x$. For the evaluation of (3), the values of r_{0_i} are first calculated using the optimisation method described in [6]. The limitation of this method is that of sampling the modified von Karman PSD at Δf , as most of the power lies at the low spatial frequencies. To access these frequencies, a larger spatial sampling rate, Δx , would be required. To overcome this, more phase screens are generated at subharmonics of Δf ($\Delta f_p = \Delta f/3^p$). These are then combined with the base screen, resulting in accurate phase representations of the turbulent atmosphere.

In the case of anisoplanatic simulation, (4) and (5) are evaluated with respect to a single propagation, resulting in a cropped screen size of X . The extended screens have a width of $\tilde{X}_i = \Delta x \tilde{N}_i$, where \tilde{N}_i is evaluated based on the position of the screen along the propagation path, the largest of which (i.e. closest to the source plane) is sampled such that it is N samples larger than the source image (sampled at δ_o) for the accommodation of the corner pencil rays. Note that the PSDs for these extended screens are sampled at a frequency of $\Delta f = 1/\Delta x \tilde{N}$. An example phase screen can be seen in Fig.4a.

2) *Propagation and Image Generation:* The point source used for propagation is modelled as a 2D Gaussian windowed sinc function [5], defined as

$$U_{pt}(x, y) = \lambda L \alpha^2 e^{-i\frac{k}{2L}(x^2+y^2)} \text{sinc}[\alpha x, \alpha y] e^{-\frac{\alpha^2}{16}(x^2+y^2)}$$

where $\alpha = (4D)/(\lambda L)$. This source is designed such that if propagated through a turbulent free atmosphere, the result is a $4D \times 4D$ patch of uniform amplitude on the observation plane.

Split-step propagation of the point source is performed using the Fresnel diffraction equation; for the purposes of this paper, the derivation of this equation is omitted, and the reader is redirected to [5]. This equation begins with the point source and propagates it to the first phase screen, at which point the phase is altered as defined. This process is then repeated until the observation plane is reached. This is known as split-step propagation. An attenuating border is also introduced to the phase screens, in order to reduce any signal energy that is tending towards the simulation boundary. The split-step propagation produces a complex field $U_0(x, y)$ at the

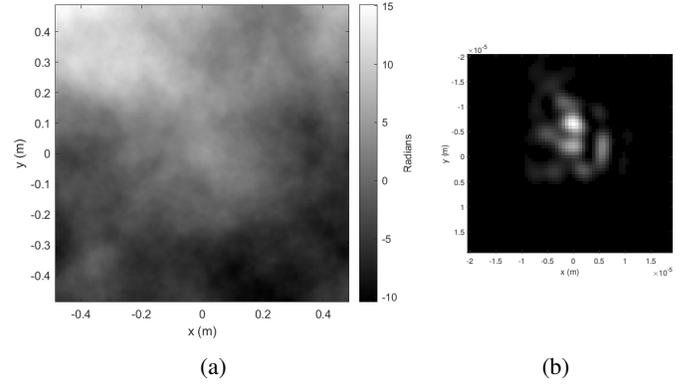


Fig. 4: (a) Example Modified Von Karman Phase Screen with subharmonics (b) PSF after propagation and lens functions.

observation plane. This complex field represents a spherical wave with a quadratic phase. In order to simulate the lens function, this spherical wave first needs to be converted into a plane wave. This is done using a collimation operation [6]. This stage also includes a pupil function, $a(x, y)$, which masks the complex plane according to the camera aperture, D . These two operations provide a complex plane, $p(x, y)$, which represents the amplitude distribution behind the lens, and is given as

$$p(x, y) = a(x, y) U_0(x, y) \exp\left[\frac{-i\pi(x^2 + y^2)}{\lambda R}\right]$$

The final PSF can then be found using Fourier optics principles [25] as

$$h(x, y) = (|\text{FT}\{p(x, y)\}|^2) \Big|_{u=\frac{x}{\lambda L}, v=\frac{y}{\lambda L}}$$

which is then resampled to Nyquist sample spacing [6], and normalised to have a sum of 1, where $\text{FT}\{\cdot\}$ denotes the forward Fourier transform. An example PSF is shown in Fig. 4b. PSFs are obtained for each pixel in the image, allowing the final turbulent image to then be formulated as

$$y[m, n] = \sum_j \sum_i x[i, j] \cdot h_{m,n}[m-i, n-j]$$

where x is the clean input image.

A performance gain for the simulation process is possible at this stage. In fact, due to the nature of the pencil rays, a pixel will have a very similar path through the atmosphere as its immediate neighbours. Therefore, a skip parameter allows the pencil rays to be traced using a sparser grid of pixels. The PSFs of the remaining pixels are then estimated using bilinear interpolation.

C. Dataset Generation

The dataset generated for this work makes use of the Places dataset [26]. This is a dataset of 1,469,737 scene images covering 205 separate categories, of which 31 have been chosen as categories that could be prone to turbulent interference, such as outdoor scenes. By isolating the data to

TABLE I: Simulator settings for provided dataset.

Description	Dataset Values
Camera aperture diameter	$D = 0.1m$
Propagation distance	$L = 5km$
Number of phase screens	$n_{scr} = 8$
Inner scale	$l_0 = 0.01m$
Outer Scale	$L_0 = 300m$
Pixel Skip	$skip = 4$
wavelength of light	$\lambda = 525 \times 10^{-9}$
Image Size	Image_Pixels = 257×257
Video frames	frame_count = 15
No. of Subharmonics	Sub_Count = 2
Subharmonic grid size	$N_{subharm} = 4$

TABLE II: Dataset variables.

Variable	Values
C_n^2 [$m^{-2/3}$]	$(0.25, 0.6525, 0.875, 1.1875, 1.5) \times 10^{-15}$
Turbulence Speed	1, 2, 3, 4
Turbulence Direction	$\rightarrow \downarrow \swarrow \nwarrow \leftarrow \nearrow \uparrow$

these categories, a total of 148,884 images were selected for the simulation process.

The simulation parameters used for the resultant dataset are detailed in Table I, where the resulting grid spacing and sample count of the cropped phase screens were $\Delta x = 0.0064$, $N = 64$. To ensure a comprehensive dataset of diverse data, three variables were used during the simulation process. For each video, the values of C_n^2 , turbulence speed and turbulence direction were randomly selected from a predefined list of values, shown in Table II. Such values allow for a total of 160 different potential turbulent simulations within the dataset. The random choice is made such that the final dataset has a uniform distribution of these 160 different classes. The range of values for the atmospheric structure parameter C_n^2 is chosen such that a range of low to high turbulence is represented, the two extremes of which are shown in Fig.1. The speed and direction of the turbulence are given as an integer value and an angle of movement. Each details the nature of movement of the enlarged phase screens in between each frame of the simulation. The speed denotes how many pixels the screens are to move, whilst the direction provides the angle in which the screens should translate. This can result in slow to fast turbulence in all the cardinal directions as well as diagonals.

IV. CONCLUSION

This paper has presented the details of an accurate method of turbulence simulation for optical images. This simulator is capable of imparting the realistic effects of turbulence onto datasets of clean images. Such data can then be further utilised in the development of turbulence mitigation algorithms. This paper also presents a dataset that is available for public use, allowing the slow process of propagation simulation to be avoided.

REFERENCES

- [1] A. Tunick, N. Tikhonov, M. Vorontsov, and G. Carhart, "Characterization of optical turbulence (cn2) data measured at the arl a_lot facility," tech. rep., US Army Research Laboratory Adelphi United States, 2005.

- [2] D. Vint, G. Di Caterina, J. Soraghan, R. Lamb, and D. Humphreys, "Analysis of deep learning architectures for turbulence mitigation in long-range imagery," in *Artificial Intelligence and Machine Learning in Defense Applications II*, vol. 11543, p. 1154303, SPIE, 2020.
- [3] N. Anantrasirichai, A. Achim, N. G. Kingsbury, and D. R. Bull, "Atmospheric turbulence mitigation using complex wavelet-based fusion," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2398–2408, 2013.
- [4] N. Anantrasirichai, A. Achim, and D. Bull, "Mitigating the effects of atmospheric distortion on video imagery: A review," 2011.
- [5] J. D. Schmidt, *Numerical Simulation of Optical Wave Propagation with examples in MATLAB*. SPIE, 2010.
- [6] R. C. Hardie, J. D. Power, D. A. LeMaster, D. R. Droege, S. Gladysz, and S. Bose-Pillai, "Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis," *Optical Engineering*, vol. 56, no. 7, pp. 071502–071502, 2017.
- [7] G. Chen, Z. Gao, Q. Wang, and Q. Luo, "U-net like deep autoencoders for deblurring atmospheric turbulence," *Journal of Electronic Imaging*, vol. 28, no. 5, pp. 053024–053024, 2019.
- [8] M. Van Iersel and A. M. Van Eijk, "Estimating turbulence in images," in *Free-Space Laser Communications X*, vol. 7814, pp. 197–206, SPIE, 2010.
- [9] W. H. Chak, C. P. Lau, and L. M. Lui, "Subsampled turbulence removal network," *arXiv preprint arXiv:1807.04418*, 2018.
- [10] J. Gao, N. Anantrasirichai, and D. Bull, "Atmospheric turbulence removal using convolutional neural network," *arXiv preprint arXiv:1912.11350*, 2019.
- [11] M. C. Roggemann, B. M. Welsh, and B. R. Hunt, *Imaging through turbulence*. CRC press, 1996.
- [12] J. P. Bos and M. C. Roggemann, "Technique for simulating anisoplanatic image formation over long horizontal paths," *Optical Engineering*, vol. 51, no. 10, pp. 101704–101704, 2012.
- [13] N. Chimmitt, Z. Mao, G. Hong, and S. H. Chan, "Rethinking atmospheric turbulence mitigation," *arXiv preprint arXiv:1905.07498*, 2019.
- [14] J. Beck, C. Bekins, and J. P. Bos, "Wavepy: a python package for wave optics," in *Long-Range Imaging*, vol. 9846, pp. 8–17, SPIE, 2016.
- [15] E. Repasi and R. Weiss, "Computer simulation of image degradations by atmospheric turbulence for horizontal views," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII*, vol. 8014, pp. 279–287, SPIE, 2011.
- [16] R. Nieuwenhuizen and K. Schutte, "Deep learning for software-based turbulence mitigation in long-range imaging," in *Artificial Intelligence and Machine Learning in Defense Applications*, vol. 11169, pp. 153–162, SPIE, 2019.
- [17] C. Zhang, B. Xue, F. Zhou, and W. Xiong, "Removing atmospheric turbulence effects in unified complex steerable pyramid framework," *IEEE Access*, vol. 6, pp. 75855–75867, 2018.
- [18] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 607–614, IEEE, 2010.
- [19] H. van der Elst and J. van Schalkwyk, "Modelling and restoring images distorted by atmospheric turbulence," in *Proceedings of COMSIG'94-1994 South African Symposium on Communications and Signal Processing*, pp. 162–167, IEEE, 1994.
- [20] J. Gilles and N. B. Ferrante, "Open turbulent image set (otis)," *Pattern Recognition Letters*, vol. 86, pp. 38–41, 2017.
- [21] L. C. Andrews, *Laser Beam Propagation Through Random Media*. SPIE, 2005.
- [22] A. McFadden, "Creating a simulator of visual turbulence in long distance imaging," Master's thesis, University of Strathclyde, 2019.
- [23] A. N. Kolmogorov, "The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 434, no. 1890, pp. 9–13, 1991.
- [24] N. Chimmitt and S. H. Chan, "Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated zernike coefficients," *Optical Engineering*, vol. 59, no. 8, pp. 083101–083101, 2020.
- [25] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company publishers, 2005.
- [26] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," *Advances in neural information processing systems*, vol. 27, 2014.

Investigation of an end-to-end neural architecture for image-based source term estimation

A. Abdulaziz, Y. Altmann, S. McLaughlin
School of Engineering and Physical Sciences
Heriot-Watt University
 Edinburgh, United Kingdom
 a.abdulaziz@hw.ac.uk

M. E. Davies
School of Engineering
University of Edinburgh
 Edinburgh, United Kingdom

Abstract—Rapid and accurate estimation of hazardous material release parameters, including source location, release time, and quantity of material released, is crucial for protecting assets and facilitating timely and effective emergency response. In this paper, we present a first artificial neural network (ANN) approach for end-to-end source term estimation (STE) using time-series of multispectral satellite images. The architecture consists of two successive ANNs. The first-stage ANN estimates the hazardous material release rate over time, producing a 3D concentration map, while the second-stage ANN utilizes the generated concentration map to estimate the 2D source location, release time, and easterly and northerly wind speeds. By leveraging the inherent nonlinearity of ANNs and advances in parallel computing, our proposed method aims to eventually overcome the limitations of existing optimization and Bayesian inference techniques in handling the nonlinear STE problem. In this preliminary study, we validate the performance of our approach on a simulated dataset, demonstrating its potential for enhancing the accuracy and speed of STE in real-world applications.

Index Terms—source term estimation, artificial neural networks

I. INTRODUCTION

The increasing threat of hazardous material releases resulting from accidents, terrorism [1], or natural disasters, such as the Bhopal gas leak accident [2], the Fukushima nuclear accident [3], and the Eyjafjallajökull volcanic eruption [4], highlights the crucial need for quick and precise estimation of the emission source location, time of release, and quantity of material released. This information is vital to protect public health and enable effective emergency response. Atmospheric dispersion simulation (ADS) models are usually employed to predict the spread of contaminants, assisting in efficient response and post-emergency assessment. Researchers have developed numerous ADS modeling methods, with the Gaussian model being a typical and efficient tool for atmospheric dispersion prediction due to its simple expression [5]. The Gaussian dispersion models (such as the Gaussian puff model and the Gaussian plume model) are particularly suitable for emergency management because of their simplicity and efficiency [6]. Accurate forecasting requires several input variables for the model, including meteorological data, release

strength, and location. While meteorological data are generally available from local weather stations or global sources, the strength, location, and release timing often remain unknown and must be inferred from relevant sensor measurements. The development of methods to address this challenge is referred to as inverse dispersion modeling or source term estimation (STE). Incorporating meteorological variables as parameters can account for spatial variations in meteorological conditions, leading to a more accurate overall source estimation. Most STE problems consider individual sensors on the ground or sparse sensor networks. This requires the deployment of assets in the region of interest which may not be possible in remote locations. In this paper, we adopt alternative approach where STE is solved using multi/hyperspectral satellite images which are becoming increasingly available. This, however, assumes that the source of interest is observable using an imaging modality.

STE presents computational challenges due to the intrinsically nonlinear nature of radionuclide diffusion processes in the atmosphere [7]. Two primary approaches are employed to tackle this problem: optimization methods [8]–[15] and probabilistic techniques based on Bayesian inference [16]–[23]. Regardless of the approach, the inferred source parameters are input into a forward ADS model to generate predicted concentrations, which are then compared to observed data using a data fidelity term or likelihood function. The primary objective of these methods is to identify the optimal or most probable match between the predicted and observed data [24]. Both sets of approaches have demonstrated promising results in simulations; however, it has been found that there is substantial potential for improvement when applied to experimental data [24], [25]. Bayesian methods offer the advantage of generating a final estimate with confidence levels and incorporating prior information into the algorithm via probability distributions. This allows for the potential accounting of inaccuracies due to modeling errors or sensor noise, although perfect characterization of these distributions may be challenging, especially in real-world scenarios. On the other hand, optimization methods yield a single point estimate for the source parameters but do not provide confidence intervals for prior information or the final estimate. Despite this drawback, optimization methods are typically less com-

This work was supported by the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing.

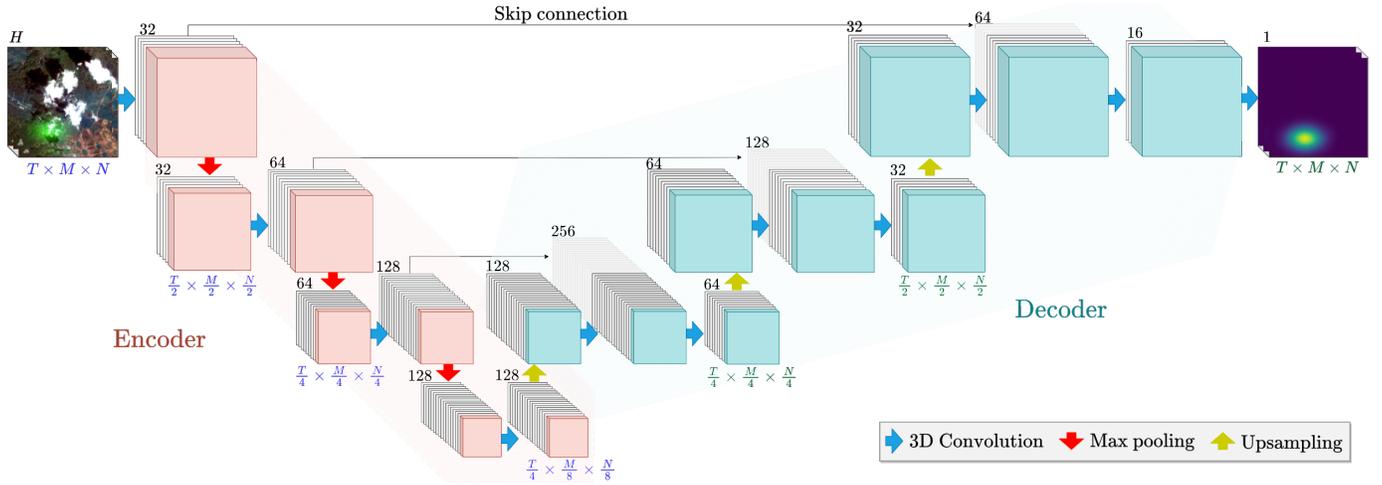


Fig. 1: Schematic of the first-stage of the proposed ANN pipeline

. Arrows of different colors indicate various types of neural network layers, as explained by the color-coding in the legend above.

putationally demanding and may converge more rapidly than Bayesian techniques. Furthermore, they benefit from requiring little or no prior information, although the availability of such information could lead to improved performance.

In recent years, artificial neural networks (ANNs) have emerged as a promising approach to enhance STE [6], [7], [26]–[30]. ANNs can capture complex, nonlinear relationships between inputs and outputs, making them suitable for tackling the inherent nonlinearities of the STE problem. Additionally, deep learning methods can benefit from large amounts of training data, which can lead to more accurate and robust models. Furthermore, advances in parallel computing and hardware accelerators, such as GPUs, have enabled ANNs to achieve faster convergence and real-time performance, making them an attractive option for STE in critical applications. However, existing ANN-based approaches for STE are designed to estimate a subset of the source term parameters such as the release rate [6], [7], [27], [28], [30], the release rate and release time [26], or the source 2D coordinates [29], while the other parameters are assumed known.

In this paper, we introduce a two-stage ANN pipeline designed for estimating source term parameters from time-series hyperspectral satellite images. The first stage of the pipeline focuses on calculating the hazardous material release rate over time, subsequently generating a 3D concentration map derived from the time-series hyperspectral satellite images. The second stage utilizes the 3D concentration map to estimate the 2D source location, the release time, along with the easterly and northerly wind speeds. The effectiveness of the proposed approach is thoroughly validated using a simulated dataset. It is important to highlight that in this study, we assume an instantaneous 2D release simulated utilizing the Gaussian puff model, which is adapted from [14].

The remainder of this paper is organized as follows. In Section II, we introduce the two-stage ANN pipeline adopted

in this work for solving the STE problem. The experimental setup and results are discussed in Section III, and finally, we conclude with the key findings and future directions in Section IV.

II. METHODOLOGY

Unlike conventional Bayesian and optimization algorithms commonly employed in STE, ANNs offer several significant benefits that make them a more attractive solution for rapid response scenarios. One key advantage is their ability to learn highly non-linear dispersion models directly from training data, eliminating the need for an analytical expression of the forward model, i.e., the cloud formation/evolution model. Moreover, a well-trained ANN can compute predictions in a fraction of a second, providing high computational efficiency that is crucial in time-sensitive situations.

Our objective in this preliminary study is to accurately estimate the following source parameters using ANNs: release rate, 2D spatial source location, release time, and both easterly and northerly wind speeds. The direct prediction of these parameters from hyperspectral satellite images poses a considerable challenge, given the images encompass a diverse range of geographical areas. To address this, we adopt a two-stage strategy. In the first stage, we extract the release rates of hazardous materials from time-series hyperspectral satellite images, generating a 3D concentration map. Subsequently, in the second stage, we utilize the concentration map to estimate the source location, release time, and the two wind speed components. This novel method demonstrates the potential of ANNs to enhance STE in emergency situations, ultimately aiding in more efficient response and mitigation efforts.

A. Two-stage ANN pipeline

The first-stage ANN takes as input a time-series of hyperspectral satellite images with dimensions $T \times M \times N \times H$, containing hazardous material dispersion. Here, T denotes

the temporal dimension, M and N represent the two spatial dimensions, and H corresponds to the channel dimension (i.e., $H = 3$ for RGB but H can be larger for multi/hyperspectral images). The output is a 3D cloud with dimensions $T \times M \times N$, which conveys the spatial concentration of hazardous material over time. As illustrated in Fig. 1, the architecture of the first-stage ANN follows a 3D U-Net design, comprising an encoder and a decoder with skip connections [31], [32]. The encoder consists of four 3D convolutional layers, with the number of feature channels set at 32, 64, 128, and 128 for each layer, respectively. The first three convolutional layers are followed by a 3D max-pooling operation for down-sampling, with pool sizes of $(2 \times 2 \times 2)$ after the first two layers and $(1 \times 2 \times 2)$ after the third layer. The decoder features three levels. Each level begins with 3D up-sampling of the feature map, using a size of $(1 \times 2 \times 2)$ for the first step and $(2 \times 2 \times 2)$ for the remaining two steps. Every up-sampling operation is succeeded by a 3D convolution, with the number of feature channels set to 128, 64, and 32, respectively. To enable localization, the output of each convolution is concatenated with the corresponding feature map from the encoder (skip connection). A subsequent 3D convolutional layer is then employed to learn to assemble a more precise output based on the concatenated features, with the number of feature channels set to 64, 32, and 16, respectively. All convolutional layers utilize a kernel size of $(3 \times 3 \times 3)$ and a rectified linear unit (ReLU) activation function. Finally, a $(1 \times 1 \times 1)$ convolution maps each 16-component feature vector to the desired concentration rate. It is important to note that the first-stage ANN can be modified to accommodate more complex observational phenomena, such as changes in illumination or registration errors. Ultimately, it could be integrated as a module into the second-stage ANN.

The second-stage ANN input is the first-stage output, specifically, the 3D extracted cloud with dimensions $T \times M \times N$.

The output of this stage consists of the source term parameters, including release time, the 2D spatial position of the source, and the easterly and northerly wind speeds. As depicted in Fig. 2, the architecture of the second-stage ANN commences with an encoder that shares the same design as the encoder in the first-stage. After the encoder, the two spatial dimensions and the channel dimension are flattened into a single feature dimension, while maintaining the temporal dimension. A Long Short-Term Memory (LSTM) layer containing 2048 units is employed to capture global temporal information. Subsequently, three additional LSTM layers, each consisting of 64 units, are utilized to capture the release time information, source coordinates information, and wind speed information, respectively. The number of LSTM units have been obtained through cross-validation. Finally, three dense layers are used to predict the release time, the 2D spatial coordinates of the release source, and the 2D components of the wind speed.

It is worth noting that the proposed architecture is highly adaptable and can accommodate time-series hyperspectral satellite images of any dimensions $T \times M \times N \times H$. This flexibility allows the model to process a wide range of input sizes, making it suitable for various applications and scenarios.

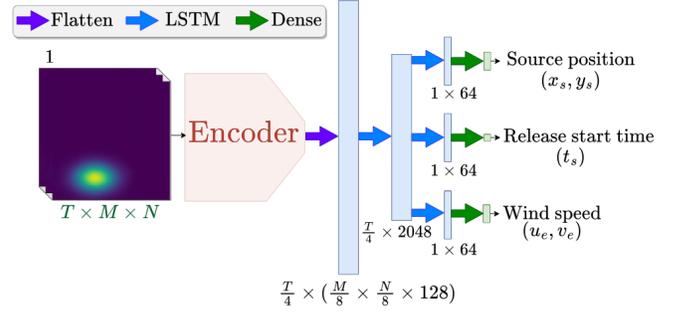


Fig. 2: Schematic of the second-stage of the proposed ANN pipeline. Arrows of different colors indicate various types of neural network layers, as explained by the color-coding in the legend above. The encoder has the same architecture as the encoder of the first-stage with output size $(\frac{T}{4} \times \frac{M}{8} \times \frac{N}{8} \times 128)$.

B. Two-stage ANN training

The two stages of the proposed ANN pipeline were trained sequentially, with the second-stage ANN utilizing the extracted 3D concentration map generated by the first-stage ANN. More precisely, we first trained the first-stage ANN and then the second-stage ANN by considering the full pipeline and freezing the first-stage ANN. Both ANNs were trained for 100 epochs using the Adam optimizer [33], with a learning rate of 10^{-3} and a batch size of 30.

The loss function employed for this process is the mean squared error (MSE) loss. In the first branch, the MSE loss is calculated between the true concentration cloud and the predicted cloud, while in the second branch, the MSE loss is determined between the true source term parameters and the predicted parameters. This approach ensures the optimization of both branches for accurate and effective source term estimation.

III. SIMULATIONS AND RESULTS

An essential aspect of the proposed ANN approach is data preparation, which involves the creation of training and testing datasets. We utilize 320 RGB satellite images with dimensions $968 \times 937 \times 3$ and a 4 m resolution from the Pleiades ESA archive [34]. It is worth noting that in this study we use $H = 3$ (RGB images) for the proof of principle and computational concerns. However, H can be larger for multi/hyperspectral images. To augment the dataset size, we randomly crop 10 sections of size $128 \times 128 \times 3$ from each high-resolution image, resulting in a total of 3200 satellite images. Of these, 3000 images are designated for training, while the remaining 200 are allocated for testing.

To simulate instantaneous 2D release of hazardous material over time, we employ the 2D Gaussian puff model, which is adapted from [14] and takes the following form:

$$c(x, y, t) = \frac{q_s}{4\pi\sqrt{\sigma_x\sigma_y}} \exp\left[-\frac{0.25}{(t-t_s)} \times \left(\frac{(x-x_s-u_e(t-t_s))^2}{\sigma_x} + \frac{(y-y_s-v_e(t-t_s))^2}{\sigma_y}\right)\right], \quad (1)$$

where q_s is the source mass in kg, t_s is the release time, x_s and y_s are the spatial coordinates of the source, u_e and v_e are the easterly and northerly wind speed components, and σ_x and σ_y are the dispersion coefficients. In this context, x and y correspond to a 2D grid with a range of $[-63, 64]$ pixels, covering a range of $512 \times 512 \text{ m}^2$. We set the dispersion coefficients to $\sigma_x = 9$ and $\sigma_y = 4$, which control the dispersion profile. The time t is an integer number in the range $[1, T]$ frames. Here, we assume $T = 20$ frames and the time between two successive frames is 2 sec.

For each of the 3200 RGB images, we generate a random concentration map over time following (1). We randomly select the emission source coordinates, x_s and y_s , within the $[-50, 50]$ pixel range, ensuring the source remains within the field of view. The release time, t_s , is randomly chosen within the $[1, 20]$ frame range. The easterly and northerly wind components, u_e and v_e , are randomly determined within the $[-2, 2]$ pixel range, corresponding to 0 mps, 2 mps, or 4 mps wind speeds. To add the concentration map to the normalised $[0, 1]$ range RGB image, we simulate a mostly green RGB spectrum for the source (Fig. 3), using the same spectrum across the dataset, and set the source mass q_s to 50 kg for a peak source intensity of 0.66 in the green channel. We assume static images over time. Following this, we simulate 3200 cubes of hazardous material dispersion, each with $20 \times 128 \times 128 \times 3$ dimensions, where $T = 20$, $M = N = 128$, and $H = 3$.

Recall that the first-stage ANN processes a time-series of RGB satellite images containing hazardous material dispersion and generates a 3D cloud representing the spatial concentration of the hazardous material over time. To assess the performance of the first-stage ANN, we present three time frames of one of the testing emission scenarios in Fig. 3. For this particular scenario, the release time, t_s , is frame 3. As shown in Fig. 3 (top row), the concentration map is empty at frame $t = 1$ since there is no emission. At frame $t = t_s = 3$, the ANN successfully extracts the concentration map from the RGB satellite image. Moreover, the model effectively captures the concentration map’s evolution over time, as depicted in Fig. 3 (bottom row). Notably, the model can differentiate between the hazardous material release and clouds in the sky, suggesting that the proposed approach is robust under adverse weather conditions. The average MSE between the predicted and true clouds over the 200 testing emission scenarios is 1×10^{-6} , which demonstrates the high fidelity of the ANN approach.

To assess the performance of the second-stage ANN, which processes the extracted 3D concentration map and generates the source term parameters, we present the average MSE between the estimated source parameters and the true values over the testing dataset of 200 emission scenarios in Table I. The results demonstrate that the release time is predicted effectively, with an average MSE of 0.09 frames, indicating the model’s efficiency for emergency response applications. Additionally, the 2D spatial coordinates of the emission source are predicted with high accuracy, as evidenced by an average MSE of approximately 1 pixel. Lastly, the easterly and northerly wind speed components are also estimated with high

precision, with the average MSE being less than half a pixel. These results underscore the effectiveness and reliability of the second-stage ANN in determining the source term parameters.

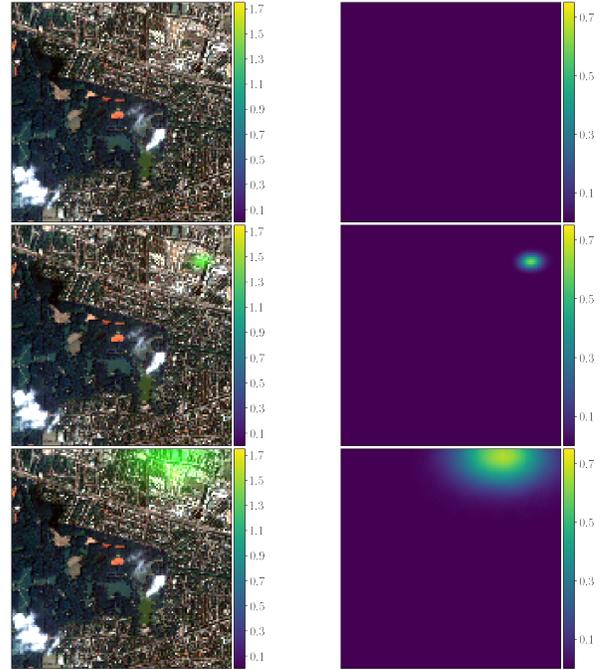


Fig. 3: Estimated concentration maps over time (right panels) obtained from the corresponding RGB satellite images (left panels) using the first-stage ANN. Displayed from top to bottom are the results for frames 1, 3, and 20, with the release time, t_s , set at frame 3.

TABLE I: Average MSE results between the predicted source term parameters from the second-stage ANN and the true values for the testing dataset of 200 emission scenarios.

source term parameter	MSE
x_s	1.16 ± 2.04 (pixels)
y_s	0.99 ± 1.67 (pixels)
t_s	0.09 ± 0.15 (frames)
u_e	0.4 ± 1.52 (pixels)
v_e	0.42 ± 1.65 (pixels)

IV. CONCLUSIONS

In conclusion, this study presents a novel two-stage ANN pipeline for source term estimation (STE) using time-series multispectral satellite images, addressing the inherent non-linearity of the problem. By harnessing the power of ANNs and advances in parallel computing, the proposed approach offers a promising solution for rapid and precise estimation of hazardous material release parameters, which is vital for public health protection and effective emergency response management. We have not conducted a comparison of our proposed method with other STE methods due to the challenge of designing a fair comparison. However, this is an important aspect for future work. To improve our pipeline for real-world applications, several areas require further investigation and improvement. Conducting an uncertainty analysis is a crucial

step to evaluate the pipeline's reliability. The performance can be significantly enhanced by refining the encoder-decoder architecture, possibly by incorporating a Variational Auto-encoder (VAE) architecture. It's also important to assess how our pipeline performs with irregularly timed and misregistered images, to improve the detection of faint clouds in the imagery, and to work on more realistic data simulations. Reevaluating the training approach for the artificial neural networks (ANNs) is another area worth considering. Instead of training the two-stage ANNs separately, exploring an end-to-end training approach or merging the two successive ANNs into a single network could be beneficial.

REFERENCES

- [1] A. Gunatilaka, B. Ristic, and R. Gailis, "On localisation of a radiological point source," in *2007 Information, Decision and Control*, 2007, pp. 236–241.
- [2] D. F. VARMA and S. MULAY, "The bhopal accident and methyl isocyanate toxicity," *Toxicology of organophosphate & carbamate compounds*, pp. 79–88, 2006.
- [3] T. J. Yasunari, A. Stohl, R. S. Hayano, J. F. Burkhart, S. Eckhardt, and T. Yasunari, "Cesium-137 deposition and contamination of japanese soils due to the fukushima nuclear accident," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. 19 530–19 534, 2011.
- [4] A. Stohl, A. J. Prata, S. Eckhardt, L. Clarisse, A. Durant, S. Henne, N. I. Kristiansen, A. Minikin, U. Schumann, P. Seibert, K. Stebel, H. E. Thomas, T. Thorsteinsson, K. Tørseth, and B. Weinzierl, "Determination of time- and height-resolved volcanic ash emissions and their use for quantitative ash dispersion modeling: the 2010 eyjafjallajökull eruption," *Atmospheric Chemistry and Physics*, vol. 11, no. 9, pp. 4333–4351, 2011. [Online]. Available: <https://acp.copernicus.org/articles/11/4333/2011/>
- [5] N. S. Holmes and L. Morawska, "A review of dispersion modelling and its application to the dispersion of particles: an overview of different dispersion models available," *Atmospheric environment*, vol. 40, no. 30, pp. 5902–5928, 2006.
- [6] S. Qiu, B. Chen, R. Wang, Z. Zhu, Y. Wang, and X. Qiu, "Atmospheric dispersion prediction and source estimation of hazardous gas using artificial neural network, particle swarm optimization and expectation maximization," *Atmospheric environment*, vol. 178, pp. 158–163, 2018.
- [7] W. Cui, B. Cao, Q. Fan, J. Fan, and Y. Chen, "Source term inversion of nuclear accident based on deep feedforward neural network," *Annals of Nuclear Energy*, vol. 175, p. 109257, 2022.
- [8] S. E. Haupt, G. S. Young, and C. T. Allen, "Validation of a receptor–dispersion model coupled with a genetic algorithm using synthetic data," *Journal of applied meteorology and climatology*, vol. 45, no. 3, pp. 476–490, 2006.
- [9] L. C. Thomson, B. Hirst, G. Gibson, S. Gillespie, P. Jonathan, K. D. Skeldon, and M. J. Padgett, "An improved algorithm for locating a gas source using inverse methods," *Atmospheric environment*, vol. 41, no. 6, pp. 1128–1134, 2007.
- [10] C. T. Allen, G. S. Young, and S. E. Haupt, "Improving pollutant source characterization by better estimating wind direction with a genetic algorithm," *Atmospheric Environment*, vol. 41, no. 11, pp. 2283–2289, 2007.
- [11] X. Zheng and Z. Chen, "Back-calculation of the strength and location of hazardous materials releases using the pattern search method," *Journal of hazardous materials*, vol. 183, no. 1-3, pp. 474–481, 2010.
- [12] S. K. Singh and R. Rani, "A least-squares inversion technique for identification of a point release: Application to fusion field trials 2007," *Atmospheric environment*, vol. 92, pp. 104–117, 2014.
- [13] Y. Liu, S. Fang, H. Li, J. Qu, and D. Fang, "Source term estimation in nuclear accidents using 4d variational data assimilation for heterogeneous atmospheric condition," in *International Conference on Nuclear Engineering*, vol. 45936. American Society of Mechanical Engineers, 2014, p. V003T06A021.
- [14] P. E. Bieringer, L. M. Rodriguez, F. Vandenberghe, J. G. Hurst, G. Bieberbach Jr, I. Sykes, J. R. Hannan, J. Zaragoza, and R. N. Fry Jr, "Automated source term and wind parameter estimation for atmospheric transport and dispersion applications," *Atmospheric Environment*, vol. 122, pp. 206–219, 2015.
- [15] S. Fang, S. Zhuang, X. Li, and H. Li, "Automated release rate inversion and plume bias correction for atmospheric radionuclide leaks: a robust and general remediation to imperfect radionuclide transport modeling," *Science of The Total Environment*, vol. 754, p. 142140, 2021.
- [16] P. Robins, V. Rapley, and N. Green, "Realtime sequential inference of static parameters with expensive likelihood calculations," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 58, no. 5, pp. 641–662, 2009.
- [17] R. Lane, M. Briers, and K. Copley, "Approximate bayesian computation for source term estimation," *Mathematics in Defence*, vol. 2009, 2009.
- [18] E. Yee, "Inverse dispersion for an unknown number of sources: model selection and uncertainty analysis," *International Scholarly Research Notices*, vol. 2012, 2012.
- [19] H. Rajaona, F. Septier, P. Armand, Y. Delignon, C. Olry, A. Albergel, and J. Moussafir, "An adaptive bayesian inference algorithm to estimate the parameters of a hazardous atmospheric release," *Atmospheric Environment*, vol. 122, pp. 748–762, 2015.
- [20] B. Ristic, A. Gunatilaka, and R. Gailis, "Localisation of a source of hazardous substance dispersion using binary measurements," *Atmospheric Environment*, vol. 142, pp. 114–119, 2016.
- [21] D. D. Lucas, M. Simpson, P. Cameron-Smith, and R. L. Baskett, "Bayesian inverse modeling of the atmospheric transport and emissions of a controlled tracer release from a nuclear power plant," *Atmospheric Chemistry and Physics*, vol. 17, no. 22, pp. 13 521–13 543, 2017.
- [22] P. B. Westoby, L. Delle Monache, and D. Silk, "18th international conference on harmonisation within atmospheric dispersion modelling for regulatory purposes 9-12 october 2017, bologna, italy," 2017.
- [23] M. Hutchinson, C. Liu, and W.-H. Chen, "Source term estimation of a hazardous airborne release using an unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 4, pp. 797–817, 2019.
- [24] M. Hutchinson, H. Oh, and W.-H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Information Fusion*, vol. 36, pp. 130–148, 2017.
- [25] N. Platt and D. DeRiggi, "Comparative investigation of source term estimation algorithms using fusion field trial 2007 data: linear regression analysis," *International Journal of Environment and Pollution*, vol. 48, no. 1-4, pp. 13–21, 2012.
- [26] B. Wang, B. Chen, and J. Zhao, "The real-time estimation of hazardous gas dispersion by the integration of gas detectors, neural network and gas dispersion models," *Journal of hazardous materials*, vol. 300, pp. 433–442, 2015.
- [27] R. Wang, B. Chen, S. Qiu, Z. Zhu, Y. Wang, Y. Wang, and X. Qiu, "Comparison of machine learning models for hazardous gas dispersion prediction in field cases," *International journal of environmental research and public health*, vol. 15, no. 7, p. 1450, 2018.
- [28] R. Wang, B. Chen, S. Qiu, L. Ma, Z. Zhu, Y. Wang, and X. Qiu, "Hazardous source estimation using an artificial neural network, particle swarm optimization and a simulated annealing algorithm," *Atmosphere*, vol. 9, no. 4, p. 119, 2018.
- [29] A. Fanfarillo, "Quantifying uncertainty in source term estimation with tensorflow probability," in *2019 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*. IEEE, 2019, pp. 1–6.
- [30] Y. Ling, Q. Yue, C. Chai, Q. Shan, D. Hei, and W. Jia, "Nuclear accident source term estimation using kernel principal component analysis, particle swarm optimization, and backpropagation neural networks," *Annals of Nuclear Energy*, vol. 136, p. 107031, 2020.
- [31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
- [32] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.
- [33] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," in *ICLR*, 2015, pp. 1–15.
- [34] E. S. Agency, "Pleiades." [Online]. Available: <https://earth.esa.int/eogateway/missions/pleiades>

Random Sampling for Robust Detection of Data modulated LFM Waveforms

Kaiyu Zhang, Fraser K. Coutts, and John Thompson

Institute for Digital Communications, University of Edinburgh, Edinburgh, EH9 3FG, UK

Email: Kaiyu.Zhang@ed.ac.uk

Abstract—Integrated sensing and communication (ISAC) waveforms have emerged as a prominent area of investigation for the next generation of communications. The joint communication and radar application is one significant use case for the ISAC waveform. In this paper, receiver detection of a phase shift keying modulated linear frequency modulation (LFM) waveform is considered as an example ISAC waveform signal processing task. The non-coherent discrete chirp-Fourier transform (NC-DCFT) method is discussed for detecting the parameters of these waveforms in a blind system with time synchronisation errors. Furthermore, the random sampling process (RSP) is proposed to counteract the effect of time synchronisation errors. The simulations show for the NC-DCFT method with the RSP performs well when the SNR is over -10 dB even in the present of time synchronisation errors.

I. INTRODUCTION

Following the development of the 5G communication system, research towards the next phase of 5G and 6G technologies has attracted substantial interest, and has revealed a desire to tackle specific applications, including sensing tasks [1]. Furthermore, [2] elaborates the applications and the approaches of integrated sensing and communications (ISAC) in 6G communications such as remote sensing, simultaneous localisation and mapping and area imaging. Joint communication and radar (JCR) sensing [3] is one of the applications of ISAC waveform enabling both a communication task and radar function simultaneously. Research in [4] has introduced two research approaches to design JCR waveforms that satisfy radar-communication coexistence (RCC) and dual-functional radar-communication (DFRC) system requirements.

As a general ISAC waveform, research in [5] discusses biphasic codes and polyphase codes for the data design in the phase-modulated waveforms. Furthermore, as one of the traditional radar waveforms, the linear frequency modulation (LFM) waveform is discussed in [6] and [7]. In some DFRC systems [8], the waveform is implemented to form the data modulated LFM waveform. Then, separate receivers are designed for both the radar function and the communication task separately. Furthermore, [6] and [7] discuss the significance of parameter estimation for the LFM waveforms in the radar system to capture the distance and velocity of the target. Therefore, [9] proposes the non-coherent discrete chirp-Fourier transform (NC-DCFT) method to estimate the chirp frequency and the offset frequency for the data modulated LFM waveform. However, due to real world limitations, the receiver may exhibit deteriorated performance due to syn-

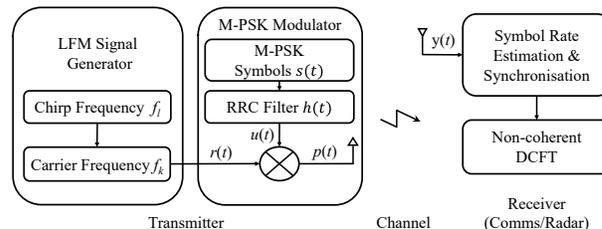


Fig. 1. System Model.

chronisation errors. As an alternative approach to counteract the effect of the synchronisation errors, this paper discusses the implementation of the NC-DCFT method and proposes the random sampling process (RSP).

The novel contributions of this paper are as follows. This paper discusses the implementation of the NC-DCFT method for the multiple-phase shift keying (M-PSK) modulated LFM waveform and proposes the RSP to counteract the adverse effect of the synchronisation. Firstly, a RSP is proposed in the NC-DCFT method for a blind receiver system when the symbol rate is unknown. Secondly, the RSP is applied to a scenario where synchronisation errors are present to improve the robustness of the NC-DCFT method.

The layout of this paper is as follows: Section II presents the system model of this paper; Section III introduces the NC-DCFT method and its application for the data modulated LFM waveform; Section IV elaborates the RSP and the inaccurate synchronisation scenario for the NC-DCFT method; in Section V, simulation results are shown to compare the normalised mean squared error (NMSE) performance for scenarios with inaccurate symbol rates and various random sampling methods, and Section VI provides conclusions to the paper.

II. SYSTEM MODEL

In this section, the system model, the data modulated LFM waveform, and the received waveform are introduced as shown in Fig. 1. The complex LFM waveform $r(t)$ in Fig. 1 at time t during the transmission period $[0, T]$ is given by

$$r(t) = \exp(j(\pi f_l t^2 + 2\pi f_k t)), \quad (1)$$

where j is $\sqrt{-1}$, f_l represents the chirp frequency, and f_k denotes the offset frequency. The quadrature phase shift keying (QPSK) symbol $s(t)$ is expressed as

$$s(t) = \exp(j(2\iota - 1)\pi/4), \quad (2)$$

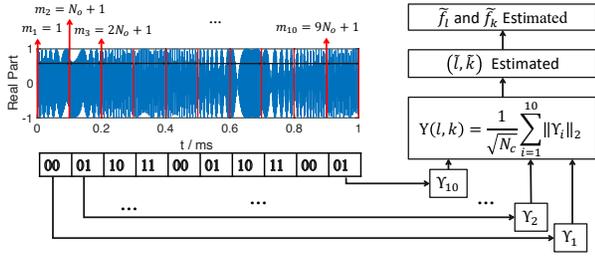


Fig. 2. Illustration of the NC-DCFT method when $N_s = 10$.

where $\iota \in \{0, 1, 2, 3\}$. When the symbol rate is R_s symbols per second and the sample frequency is f_s , then the number of symbols to be transmitted is $N_s = T \times R_s$, the total data record size is $N_c = T \times f_s$, and the oversampling rate is $N_o = f_s/R_s$. The data symbols $s[n]$ contain N_s samples obtained by sampling $s(t)$. Subsequently, the root raised-cosine (RRC) filter $h(t)$ from [10] is applied to $s[n]$ to limit the signal bandwidth and to attenuate the intersymbol interference. The pulse shape $h[n]$ is characterised by a roll-off factor β and a spanning period of N_r symbols and T_R period, resulting in $(N_r \times N_o + 1)$ samples for $h[n]$. Correspondingly, the symbol group $u[n]$ to be transmitted is given by

$$u[n] = h[n] * s[n], \quad (3)$$

where $*$ represents the convolution operator and the size of $u[n]$ is $(N_c + N_r \times N_o)$. The transmission period $[0, T]$ for $r(t)$ is modified into $[-T_R/2, T_R/2 + T]$ for $r(t)$ in (1) to compatible with $u[n]$. Then the data modulated LFM waveform $p[n]$ to be transmitted shown in Fig. 1 is formed as

$$p[n] = u[n]r[n], \quad (4)$$

where $p[n]$ comprises $(N_c + N_r \times N_o)$ samples. When the channel is subject to additive white Gaussian noise (AWGN), the received waveform $y[n]$ is denoted as

$$y[n] = p[n] + w[n], \quad (5)$$

where the size of $y[n]$ and $w[n]$ are $(N_c + N_r \times N_o)$ and $w[n]$ is AWGN distributed as $\mathcal{CN}(0, \sigma^2)$ and σ^2 is the power of the AWGN. The signal-to-noise ratio (SNR) in this paper is defined as $(E(p[n])^2/\sigma^2)$, where $E[\cdot]$ is the expectation operation. Prior to processing, the receiver truncates the first and last $(N_r \times N_o)/2$ samples in $y[n]$ denoted as $\tilde{y}[n]$ with N_c samples. As shown in Fig. 1, the symbol rate is known or is estimated by a suitable algorithm, e.g. [11]. Subsequently, we assume the receiver has synchronised to the start of the chirp waveform, except for some small timing error.

III. THE NON-COHERENT DCFT METHOD

To estimate the parameters of the data modulated waveform, the NC-DCFT method and relevant parameter recovery steps are proposed in the [9] and discussed in this section.

When the receiver has the prior knowledge on the potential detection range of f_l and f_k , these ranges are set for \tilde{f}_l and

\tilde{f}_k as $[f_l^{\min}, f_l^{\max}]$ and $[f_k^{\min}, f_k^{\max}]$. The coefficients a, c for \tilde{f}_l , and b, d for \tilde{f}_k are defined for the K length NC-DCFT as

$$a = \frac{K(f_l^{\max} - f_l^{\min})}{2f_s^2(K-1)}, \quad c = \frac{Kf_l^{\min}}{2f_s^2}, \quad (6)$$

$$b = \frac{K(f_k^{\max} - f_k^{\min})}{f_s(K-1)}, \quad d = \frac{Kf_k^{\min}}{f_s}. \quad (7)$$

Through these coefficients, the size K NC-DCFT is defined in [9] as

$$X[l, k] = \frac{1}{\sqrt{N}} \sum_{i=1}^M \left\| \sum_{n=m_i}^{m_i+n_i-1} x[n] W_K^{(al+c)n^2+(bk+d)n} \right\|_2, \quad (8)$$

where $\|\cdot\|_2$ is the L2-norm operator, $W_K = \exp(-2\pi j/K)$ is a twiddle factor, l and k are integers in the range $[0, K-1]$, M is the number of symbols in the time domain signal $x[n]$, m_i is the initial sample number and n_i , which is usually set to a constant in [9], is the number of samples for the i th symbol block in $x[n]$. Through the output matrix \mathbf{X} , the corresponding coordinate (\tilde{l}, \tilde{k}) is obtained when

$$\mathbf{X}[\tilde{l}, \tilde{k}] = \max \mathbf{X}[l, k] \quad l, k = 0, \dots, K-1. \quad (9)$$

The values of estimated results for \tilde{f}_l and \tilde{f}_k , \tilde{f}_l and \tilde{f}_k , are therefore expressed as

$$\tilde{f}_l = \frac{1}{K} 2f_s^2(a\tilde{l} + c), \quad \tilde{f}_k = \frac{1}{K} f_s(b\tilde{k} + d). \quad (10)$$

In the system model of this paper as shown in Fig. 1, the received signal $\tilde{y}[n]$ is the resource signal whose parameters are to be estimated. Thus, $\tilde{y}[n]$ is the input $x[n]$ in (8), and each element $Y[l, k]$ in the output NC-DCFT matrix \mathbf{Y} is

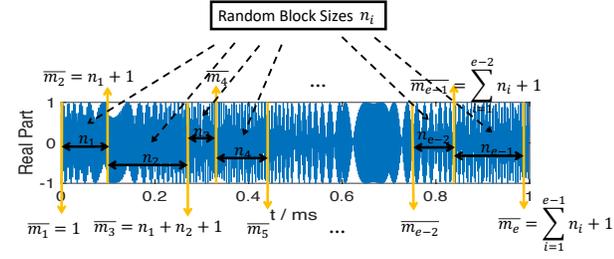
$$\begin{aligned} Y[l, k] &= \frac{1}{\sqrt{N_c}} \sum_{i=1}^{N_s} \|\Upsilon_i\|_2 \\ &= \frac{1}{\sqrt{N_c}} \sum_{i=1}^{N_s} \left\| \sum_{n=(i-1)N_o+1}^{iN_o} \tilde{y}[n] W_K^{(al+c)n^2+(bk+d)n} \right\|_2. \end{aligned} \quad (11)$$

Then \tilde{f}_l and \tilde{f}_k are estimated via the process in (9) and (10).

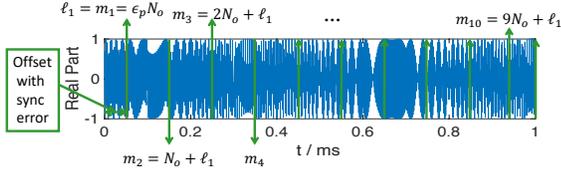
Fig. 2 shows the example process of the NC-DCFT method when $N_s = 10$. From the method shown in (11) and Fig. 2, the accuracy of the NC-DCFT method depends on the number of symbols N_s , the length of samples N_o , and the initial point m_1 in each symbol. These parameters correspond to the symbol rate estimation, the sampling process, and the synchronisation procedure.

IV. RANDOM SAMPLING AND SYNCHRONISATION ERRORS

Due to the nature of the NC-DCFT method, certain parameters are unknown during the implementation. Thus, this section proposes a RSP and discusses the presence of synchronisation errors as the motivating scenario for this approach.



(a) Sampling process with e random sampling termination points (RSTPs) \bar{m}_p .



(b) Synchronisation process with the starting point ℓ_1 .

Fig. 3. Example waveforms showing the effect of the random sampling process and the impact of synchronisation errors.

A. Random Sampling Process (RSP)

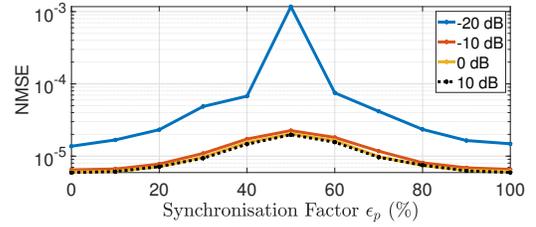
In the NC-DCFT method, the number of samples N_o in (11) is fixed for the i th symbol block, while the number of samples n_i in (8) can vary according to the implementation of the NC-DCFT. The procedure to apply a fluctuating length of samples n_i in each NC-DCFT segment is the key concept for the RSP. Fig. 3(a) shows the RSP with random sampling termination points (RSTPs) \bar{m}_p and random block size n_{p-1} for $p-1$ th block as black arrowed lines. The first sample of input is the first RSTP as well as the start point in the RSP, denoted as $\bar{m}_1 = 1$. The varying length of samples between \bar{m}_1 and \bar{m}_2 is set as n_1 and between \bar{m}_2 and \bar{m}_3 is expressed as n_2 , then the RSTPs are correspondingly $\bar{m}_2 = n_1 + 1$ and $\bar{m}_3 = n_2 + n_1 + 1$. In the following RSP, each RSTP in the RSP is $\bar{m}_p = \sum_{i=1}^{p-1} n_i + 1$ using the chosen values n_i . For the varying value of n_i , the group $R \in \mathbb{Z} \cap [R_{\min}, R_{\max}]$ is introduced. In the RSP, n_i is not fixed but uniformly and randomly selected from the group R . Based on (8), the final RSTP \bar{m}_e are introduced as the symbol transition from the $(e-1)$ th symbol to adjacent symbols. The random block size n_{e-1} is the number of input samples for the e th symbol in the NC-DCFT method. The yellow lines in Fig. 3(a) show examples of the RSTP values \bar{m}_p that form the symbol blocks for the basic NC-DCFT process. Due to the random value of n_i , the number of RSTPs e of the RSTP \bar{m}_e is defined that \bar{m}_e is no more than N_c while \bar{m}_{e+1} is greater than N_c . As a result, the number of RSTPs e is not directly to the number of symbols N_s . When applying the above process to the NC-DCFT in (8), this paper defines the whole above process as the random sampling NC-DCFT (RS-NC-DCFT).

B. Synchronisation Problem

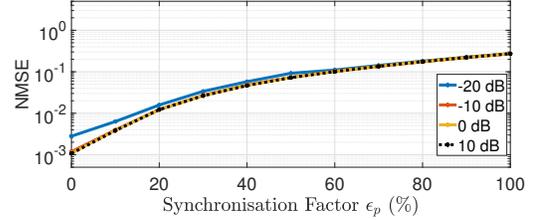
In previous research, [9] shows the influence of synchronisation error for the NC-DCFT method. With perfect syn-

TABLE I
KEY PARAMETER SETTINGS

Name and Notation	Value or Definition
Default symbol rate R_s	50 k symbols per second
Default oversampling N_o	200 samples per symbol
Roll-off factor β for RRC filter	0.5
RRC Filter span N_r	6 symbols
Number of samples N_c	10^4
Time period T	10^{-3} s
Sample frequency f_s	10^7 Hz
Number of Monte Carlo runs	10^4
Estimation range $[f_l^{\min}, f_l^{\max}]$	$[10^9, 10^{10}]$ Hz
Estimation range $[f_k^{\min}, f_k^{\max}]$	$[0, 5 \times 10^5]$ Hz
Ground truth range \mathbf{g}_l for f_l	$[10^9, 10^{10}]$ Hz
Ground truth range \mathbf{g}_k for f_k	$[0, 3 \times 10^5]$ Hz
Length of the NC-DCFT method K	256



(a) NMSE for f_l



(b) NMSE for f_k

Fig. 4. Performance of the NC-DCFT method with different synchronisation factors ϵ_p .

chronisation, the initial point of the input of $x[n]$ or $\bar{y}[n]$ is $m_1 = 1$ in Fig. 2. However, loss of synchronisation leads to a mismatch and forces the initial point to change from 1 to ℓ_1 .

To describe the scale of the inaccurate synchronisation, the synchronisation factor ϵ_p is introduced to calculate the relationship of ℓ_1 and N_o . For example, Fig. 2 shows the ideal process of the synchronisation in the red dashed line with $\ell_1 = m_1 = 1$ and $\epsilon_p = 0\%$ while Fig. 3(b) demonstrates a worst-case synchronisation error with $\ell_1 = m_1 = N_o/2$ and $\epsilon_p = 50\%$. The missed samples between the initial waveform and ℓ_1 are the offset with a synchronisation error as illustrated in Fig. 3(b). Fig. 4 shows NMSE results for various synchronisation factors ϵ_p . To evaluate the performance of the NC-DCFT, this paper utilises the NMSE metric, which is

defined as follows:

$$J_{\text{NMSE}} = \frac{\mathbf{d}\mathbf{d}^T}{\mathbf{g}\mathbf{g}^T}, \quad (12)$$

where the operation denoted by $\{\cdot\}^T$ is the transpose, each element of the vector \mathbf{d} represents the difference between $(\hat{f}_l - g_l)$ or $(\hat{f}_k - g_k)$ from each Monte Carlo simulation run, and the elements in the vector \mathbf{g} correspond to the ground truth values of g_l or g_k . The ground truth vector \mathbf{g}_l for f_l is calculated by two steps. First, the value of α is randomly generated in the interval of $[0, 1]$, and each entry of \mathbf{g}_l as g_i

$$g_i = 10^{\alpha_i} \times 10^9, \quad i = 1, 2, \dots, 10^4. \quad (13)$$

The ground truth vector \mathbf{g}_k for f_k is randomly generated in the interval of $[0, 3 \times 10^9]$. Other parameter settings of the simulations in this paper are shown in Table I.

Fig 4(a) shows when the synchronisation is perfect, namely $\epsilon_p = 0\%$, and $\epsilon_p = 100\%$, the NMSE shows similar and best performance. Furthermore, when $\ell_1 = m_1 = N_o/2$, namely $\epsilon_p = 50\%$, it performs worst for f_l estimation. For ϵ_p in the interval of $[0\%, 50\%]$, the accuracy of the f_l estimation is decreasing but decreases symmetrically in the interval of $[50\%, 100\%]$. This result is mainly caused by the composition of samples in each block of samples. Samples belonging to different symbols cancel each other out and deteriorate the phase information for computing f_l . Therefore, the NC-DCFT method in the above condition loses coherent processing gain and the recovery result is not as accurate as the ground truth. With $\epsilon_p = 100\%$, all samples of the first symbols is excluded and the similar performance with $\epsilon_p = 0\%$ is shown. Additionally, the $\epsilon_p = 50\%$ case performs worst as each data block contains equal number of samples in two different symbols. Fig 4(b) shows the higher NMSE with a higher ϵ_p , since inappropriate synchronisation loses initial information of $\bar{y}[n]$. Thus, the actual starting synchronisation point ℓ_1 experiences delay, which forces the value of f_k to change, thereby affecting the estimation accuracy.

V. SIMULATIONS AND DISCUSSION

This section compares the performance of the NC-DCFT method and the proposed RS-NC-DCFT method in different application settings. We consider scenarios with different preset symbol rates and symbol synchronisation states. Key parameter settings are shown in Table I and the performance is evaluated based on the NMSE formula in (12).

A. Comparison of Preset Symbol Rate

Since the symbol rate R_s affects the number of symbols N_s and the oversampling value N_o , both of them are significant parameters in (11). Furthermore, the receiver may preset R_s or N_s due to the prior knowledge and then process the received waveform based on these preset values, therefore this subsection discusses the impact of preset values of R_s or N_s . Assuming the preset symbol rates R_s stored in \mathbf{S} are [10 k, 25 k, 40 k, 50 k, 100 k, 200 k] while the ground truth symbol rate is 50 k in Table I. Correspondingly, the preset

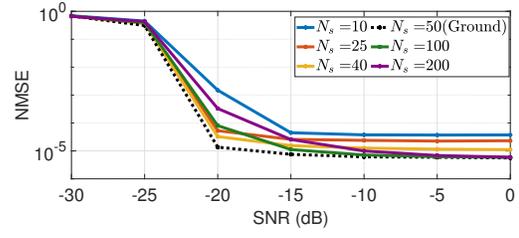


Fig. 5. NMSE results for f_l via different preset symbol rates.

TABLE II
PARAMETERS SETTINGS FOR THE RS-NC-DCFT METHOD

Label	R1	R2	R3	R4	R5
RS range	[1, 200]	[1, 400]	[40, 100]	[100, 300]	[200, 400]

values of N_s deposited in \mathbf{S}_s are [10, 25, 40, 50, 100, 200] while N_o stored in \mathbf{S}_o are [1000, 400, 250, 200, 100, 50].

The NMSE trends for both f_l and f_k is similar and therefore Fig. 5 only presents the NMSE result of f_l . In Fig. 5, the ground truth performs best as expected. However, estimating a higher symbol rate performs slightly worse but with similar results at high SNRs while estimating a lower symbol rate performs much more poorly. The performances in Fig. 5 can be attributed to incorrect sets of N_s for use in (11). Since the received signal is divided into N_s regions as part of the NC-DCFT, an incorrect estimate of N_s yields blocks of data that do not optimally capture a single symbol. When N_s is less than the ground truth, a larger number of samples are calculated for Υ_i in i th block. Such samples from different symbols again cancel each other out and therefore deteriorate the NC-DCFT performance. When N_s is larger than the ground truth, the smaller number of samples are included for Υ_i and the truncated length of samples shows poorer gain versus noise performance, especially at high SNRs.

B. Performance of the RS-NC-DCFT method

This subsection compares the performance of the RS-NC-DCFT method in Sec. IV-A and the NC-DCFT method where the symbol rate is known or estimated by [11]. Five possible settings of the RS-NC-DCFT method are shown in Table II with their own labels R1-R5 in Fig. 6.

For each Monte Carlo run, the ground truth of the length of N_o is fixed as 200. Fig. 6 only shows the NMSE result of f_l as the NMSE trends are similar for above methods for both f_l and f_k . Fig. 6 shows the NMSE results of R1 and R3 are close to the ground truth while R2, R4, and R5 performs worse. Since in the R group such as R1 and R3, the majority of the samples in the each block come from one symbol with a high probability and therefore the sample cancellation effect is avoided. Thus, when R_{\max} is larger, the RS-NC-DCFT method is unable to estimate f_l and f_k of the QPSK-LFM waveforms. Furthermore, when R_{\min} is too small such as R1, the RS-NC-DCFT method performs worse than with a medium choice of R_{\min} such as R3. Therefore, the RS-NC-DCFT methods works well with a specific and appropriate R range.

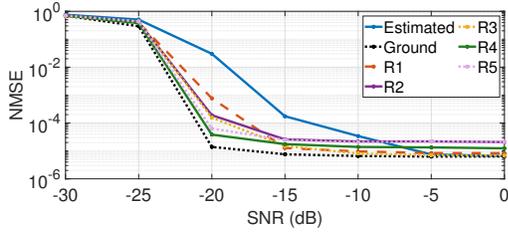
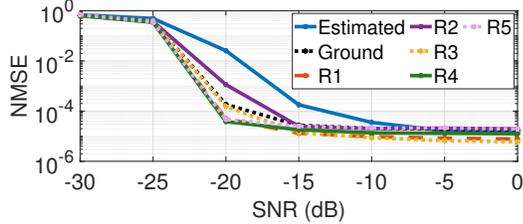
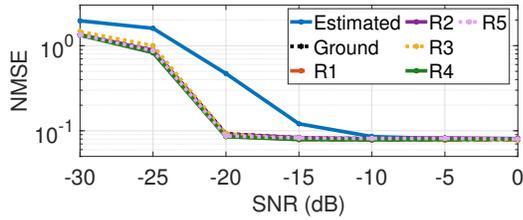


Fig. 6. NMSE results for f_l of the NC-DCFT method via different methods.



(a) NMSE for f_l under different SNRs.



(b) NMSE for f_k under different SNRs.

Fig. 7. Performance of the NC-DCFT method via different methods with $\epsilon_p = 50\%$.

Additionally, Fig. 6 shows the estimated method performs well when SNR is above -5 dB. Furthermore, the RS-NC-DCFT methods with R1 and R3 ranges outperform the estimated method when the SNR is lower than -5 dB and shows similar performance of the estimated method when the SNR is above 5 dB. Therefore, Fig. 6 shows the RS-NC-DCFT method with a specific R range is a viable scheme for the implementation of the NC-DCFT. Fig. 5 shows us that the estimated symbol rate is important for parameter estimation with the NC-DCFT. Fig. 6 confirms that, by using the RS-NC-DCFT, we can achieve a good level of performance without knowing the symbol rate exactly.

C. Synchronisation Problem

For the synchronisation process illustrated by Fig. 3(b), this subsection discusses the implementation of methods in Table II with the worst synchronisation scenario $\epsilon_p = 50\%$ shown in Fig. 4(a). For the f_l recovery results, Fig. 7(a) shows the NC-DCFT method with the known symbol rate (labelled Ground) performs poorly compared to Fig. 6. Furthermore, Fig. 7(a) shows the performance of the RS-NC-DCFT methods is similar as shown in Fig. 6 and outperforms the performance of the “Ground” approach. Therefore, the results show the RS-NC-DCFT method is able to resist the influence of inappropriate

synchronisation. However, the f_k recovery in Fig. 7(b) of all methods performs poorly due to the loss of data samples, which is similar to the effect shown in Fig. 4(b). Therefore, the result shows the RS-NC-DCFT method with a specific R range is an alternative implementation method to maintain good NMSE performance when there are synchronisation errors.

VI. CONCLUSION

This paper discusses one application of the ISAC waveform based on the data modulated LFM waveform. To estimate waveform parameters, the RS-NC-DCFT method is proposed and applied in a blind system with time synchronisation errors. For the NC-DCFT method, when the symbol rate is overestimated, the performance is found to approach that of the ground truth case in the absence of timing errors.

Moreover, the RS-NC-DCFT method with a suitable symbol rate range R performs well and shows the similar performance to the ground truth case. Furthermore, in the presence of time synchronisation errors, the RS-NC-DCFT method can perform better than the NC-DCFT method using the correct symbol rate. Scenarios with unknown symbol rates and synchronisation errors are highly likely to occur in practice. In general, the RS-NC-DCFT method proposed in this paper is a robust choice for real-world scenarios when the symbol rate is unknown and waveform synchronisation is subject to timing errors.

ACKNOWLEDGMENT

This work was supported by the Engineering and Physical Sciences Research Council of the UK (EPSRC) Grant number EP/S000631/1 and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing. The authors gratefully acknowledge insightful discussions with Prof. Bernie Mulgrew and Dr Navneet Garg.

REFERENCES

- [1] W. Saad *et al.*, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2020.
- [2] F. Liu *et al.*, “Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, 2022.
- [3] J. A. Zhang *et al.*, “An overview of signal processing techniques for joint communication and radar sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 6, pp. 1295–1315, 2021.
- [4] F. Liu *et al.*, “Joint radar and communication design: Applications, state-of-the-art, and the road ahead,” *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3834–3862, 2020.
- [5] M. A. Richards, *Fundamentals of radar signal processing*, 2nd ed. Chicago, Ill: McGraw-Hill Education LLC., 2014.
- [6] N. Levanon, *Radar signals*, ser. Wiley InterScience electronic collection. John Wiley & Sons, 2004.
- [7] F. Gini *et al.*, *Waveform Design and Diversity for Advanced Radar Systems*, ser. Radar, Sonar and Navigation. IET, 2012, vol. 22.
- [8] G. N. Saddik *et al.*, “Ultra-wideband multifunctional communications/radar system,” *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 7, pp. 1431–1437, 2007.
- [9] K. Zhang *et al.*, “Non-coherent discrete chirp Fourier transform for modulated LFM parameter estimation,” in *2022 SSPD*, 2022, pp. 1–5.
- [10] J. G. Proakis, *Digital communications / John G. Proakis, Masoud Salehi*, 5th ed. New York: McGraw-Hill, 2008.
- [11] A. A. Kuchumov, V. I. Lipatkin, and E. M. Lobov, “Advanced algorithms for automatic symbol rate estimation of M-FSK and M-PSK signals,” in *2017 SINKHROINFO*, 2017, pp. 1–4.

Generalized Polynomial Power Method

Faizan A. Khattak, Ian K. Proudler, and Stephan Weiss

Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, Scotland
 {faizan.khattak,ian.proudler,stephan.weiss}@strath.ac.uk

Abstract—The polynomial power method repeatedly multiplies a polynomial vector by a para-Hermitian matrix containing spectrally majorised eigenvalue to estimate the dominant eigenvector corresponding to the dominant eigenvalue. To limit the order of the resulting vector, truncation is performed in each iteration. This paper extends the polynomial power method from para-Hermitian matrices to a general polynomial matrix for determining its dominant left- and right-singular vectors and the corresponding singular value. The proposed extension assumes that the dominant singular is positive on the unit circle. The resulting algorithm is compared with a state-of-the-art PSVD algorithm and provides better accuracy with reduced computation time and lower approximation orders for the decomposition.

I. INTRODUCTION

The algebra of polynomial matrices has proven to be useful in solving a variety of problems related to broadband sensor arrays, particularly through the application of two main operations: the polynomial eigenvalue decomposition (PEVD) [1–8] and the polynomial singular value decomposition (PSVD) [9–11]. The PEVD is restricted to para-Hermitian matrices only, where the matrix is equal to its transpose-conjugate time-reversed version [12]. In contrast, the PSVD can be applied to any polynomial matrix. Thus, PSVD algorithms find applications in a variety of problems such as MIMO design [13, 14], paraunitary filter design [7], or beamforming [15].

Typically, the PSVD is calculated using two PEVDs [10] or via a polynomial QR decomposition (PQRD) [9], both of which are computationally expensive. However, a dedicated PSVD algorithm exists which is the generalization of the second-order sequential best rotation (SBR2)[4], and it exploits the Kogbetliantz transformation[16]. This method iteratively transfers the energy onto the diagonal. While [16] performs only an approximate diagonalization, its performance is still better than that achieved via PEVD or PQRD approaches. An SVD with analytic factors exists [17, 18], such that there are unique singular values that are real on the unit circle, and left- and right-singular vectors that share a common ambiguity w.r.t. arbitrary allpass functions. The above algorithm ignore this coupled ambiguity, and hence typically yield complex-valued approximations of the singular values.

In order to overcome the deficiencies of the above PSVD algorithms, in this document, we extend the polynomial power method [19] from a para-Hermitian matrix to a general

polynomial matrix for the computation of the dominant left- and right-singular vector and the singular value. The polynomial power method is an extension of the ordinary power method [20] to para-Hermitian matrices where a polynomial vector is repeatedly multiplied by a para-Hermitian matrix, and the resulting vector converges to the dominant eigenvector provided that the matrix is spectrally majorised. Similar to the power method, the generalised polynomial power method can be coupled with deflation in order to compute an entire PSVD.

II. POLYNOMIAL SINGULAR VALUE DECOMPOSITION

For an analytic, non-multiplexed polynomial matrix $\mathbf{A}(z) \in \mathbb{C}^{M \times N}$, $M \geq N$, the analytic SVD exists [18]

$$\mathbf{A}(z) = \mathbf{U}(z)\mathbf{\Sigma}(z)\mathbf{V}^P(z), \quad (1)$$

such that $\mathbf{\Sigma}(z) = \text{diag}\{\sigma_1(z), \dots, \sigma_N(z)\} \in \mathbb{C}^{M \times N}$ contains the analytic singular values and the matrices $\mathbf{U}(z) \in \mathbb{C}^{M \times M}$, $\mathbf{V}(z) \in \mathbb{C}^{N \times N}$ are paraunitary i.e. $\mathbf{U}(z)\mathbf{U}^P(z) = \mathbf{I}$, $\mathbf{V}(z)\mathbf{V}^P(z) = \mathbf{I}$, and contains the left- and right analytic singular vectors, respectively. Note that the parahermitian operation, $\{\cdot\}^P$, involves a Hermitian transposition and time reversal of its arguments, such that e.g. $\mathbf{U}^P(z) = (\mathbf{U}(1/z^*))^H$. Unlike singular values of constant matrices, which must be real and positive semi-definite [20], the analytic singular values evaluated on unit circle for $z = e^{j\Omega}$ must be permitted to take on negative values. This is similarly known for matrices that depend analytically on a continuous, real parameter on some interval [21, 22].

Generally, the analytic singular values of a matrix $\mathbf{A}(z)$ may intersect. However, if $\mathbf{A}(z)$ is estimated from finite data e.g. via system identification [23], it will have spectrally majorised singular values

$$\sigma_i(e^{j\Omega}) \geq \sigma_{i+1}(e^{j\Omega}) \quad \forall \Omega, \quad i = 1, \dots, (N-1) \quad (2)$$

with probability one [23]. We therefore assume the property (2) to hold for the remainder of this paper.

III. POLYNOMIAL POWER METHOD FOR PARAHERMITIAN MATRIX

The underlying idea is to extend the polynomial power method, proposed in [19] for para-Hermitian matrices, to general polynomial matrices for the extraction of the dominant singular vectors and singular value. Thus this section provides a brief summary of the polynomial power method. We denote a para-Hermitian matrix by $\mathbf{R}(z)$, which must therefore must satisfy $\mathbf{R}^P(z) = \mathbf{R}(z)$ [12].

Faizan Khattak is supported by the Commonwealth Scholarship Commission. This work was also supported in parts by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1 and the MOD University Defence Research Collaboration in Signal Processing.

A. Overall Rationale

This method is an extension of the ordinary power iteration [20] from Hermitian matrices to para-Hermitian matrices where an arbitrary polynomial vector $\mathbf{x}^{(0)}(z)$ is repeatedly multiplied with a para-Hermitian matrix $\mathbf{R}(z)$ to obtain a sequence of polynomial vectors. After k iterations, we have

$$\mathbf{x}^{(k)}(z) = \mathbf{R}(z)\mathbf{x}^{(k-1)}(z) = \mathbf{R}^k(z)\mathbf{x}^{(0)}(z). \quad (3)$$

In each iteration, $\mathbf{x}^{(k)}(z)$ has to be normalized, such that $\mathbf{x}_{\text{norm}}^{(k)}(z)$ satisfies $\mathbf{x}_{\text{norm}}^{(k)}(z)\mathbf{x}_{\text{norm}}^{(k)}(z) = 1$. This normalization can be performed on the unit-circle, due to analyticity, with ease. To limit the order growth of $\mathbf{x}^{(k)}(z)$, truncation can be applied. This can be achieved by either limiting the order to the estimated support of the eigenvector obtained from [24] by shifted-truncation [25], or removing any trailing coefficients below a small threshold. The iterations are stopped once a suitable defined difference between consecutive polynomial vectors falls below some threshold. The overall analysis that connects the polynomial power method with the ordinary power method is given next.

B. Polynomial Power Method Analysis

The initial $\mathbf{x}^{(0)}(z)$ can be represented as a linear combination of the eigenvectors of $\mathbf{R}(z)$

$$\begin{aligned} \mathbf{x}^{(0)}(z) &= \mathbf{Q}(z)\mathbf{c}(z) \\ &= c_1(z)\mathbf{q}_1(z) + \dots + c_M(z)\mathbf{q}_M(z), \end{aligned} \quad (4)$$

where $\mathbf{q}_m(z)$, $m = 1 \dots, M$, is the m th analytic eigenvector of $\mathbf{R}(z)$ and $\mathbf{c}(z) = [c_1(z), \dots, c_M(z)]^T \in \mathbb{C}^M$ is a vector of analytic weighting factors. Due to analyticity, we can restrict the analysis to the unit circle. Therefore, z can be substituted with $e^{j\Omega}$ to evaluate and iterate on the unit circle. Combining (4) and (3) with the fact that $\mathbf{R}(z)$ has spectrally-majorised eigenvalues due to being estimated from finite data, we have

$$\begin{aligned} \mathbf{x}^{(k)}(e^{j\Omega}) &= \lambda_1^k(e^{j\Omega}) \left[c_1(e^{j\Omega})\mathbf{q}_1(e^{j\Omega}) \right. \\ &\quad \left. + \sum_{m=2}^M c_m(e^{j\Omega}) \left(\frac{\lambda_m(e^{j\Omega})}{\lambda_1(e^{j\Omega})} \right)^k \mathbf{q}_m(e^{j\Omega}) \right]. \end{aligned} \quad (5)$$

The summation term will converge towards zero for $k \rightarrow \infty$. This permits us to re-write (5) as

$$\mathbf{x}^{(k)}(e^{j\Omega}) = \lim_{k \rightarrow \infty} \lambda_1^k(e^{j\Omega}) c_1(e^{j\Omega}) \mathbf{q}_1(e^{j\Omega}). \quad (6)$$

The term $\mathbf{x}^{(k)}(e^{j\Omega})$ in (6) is normalized in each iteration such that it has unit norm on the unit circle. This normalisation is carried out in the DFT domain. If $c_1(e^{j\Omega})$ possesses any spectral nulls for some Ω , the resulting division by zero in the normalisation process can be avoided by regularization [19]. For a sufficiently large k , the normalized vectors become

$$\mathbf{x}_{\text{norm}}^{(k)}(e^{j\Omega}) = \hat{\mathbf{q}}_1(z) = g_1(e^{j\Omega})\mathbf{q}_1(e^{j\Omega}), \quad (7)$$

where $g_1(e^{j\Omega}) = c_1(e^{j\Omega})/|c_1(e^{j\Omega})|$ is an allpass filter. This allpass filter generalises the phase ambiguity of the eigenvectors of a standard matrix.

Algorithm 1: PPM Algorithm [19]

Input: $\mathbf{R}(z), \epsilon, k_{\text{max}}$

Output: $\hat{\mathbf{q}}_1(z), \hat{\lambda}_1(z)$

$\mathbf{x}^{(0)}(z) \in \mathbb{C}^M, k \leftarrow 0, \gamma = \infty;$

$\tilde{\mathbf{x}}_{\text{norm}}^{(0)}(z) \leftarrow$ normalise & order limit $\mathbf{x}^{(0)}(z);$

while $\gamma > \epsilon$ & $k < k_{\text{max}}$ **do**

$k \leftarrow k + 1;$

$\mathbf{x}^{(k)}(z) \leftarrow \mathbf{R}(z)\tilde{\mathbf{x}}_{\text{norm}}^{(k-1)}(z);$

$\mathbf{x}_{\text{norm}}^{(k)}(z) \leftarrow$ normalisation $\mathbf{x}^{(k)}(z);$

$\tilde{\mathbf{x}}_{\text{norm}}^{(k)}(z) \leftarrow$ order limitation of $\mathbf{x}_{\text{norm}}^{(k)}(z);$

update γ

end

$\hat{\mathbf{q}}_1(z) = \tilde{\mathbf{x}}_{\text{norm}}^{(k)}(z);$

$\hat{\lambda}_1(z) = \tilde{\mathbf{x}}_{\text{norm}}^{(k),P}(z)\mathbf{R}(z)\tilde{\mathbf{x}}_{\text{norm}}^{(k)}(z);$

The stopping criterion for the polynomial power method is to measure the overall deviation

$$\gamma = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\alpha(\Omega)|^2 d\Omega \quad (8)$$

of the Hermitian angle $\alpha(\Omega)$, defined as

$$\begin{aligned} \alpha(\Omega) &= \angle \{ \mathbf{x}_{\text{norm}}^{(k)}(e^{j\Omega}), \mathbf{x}_{\text{norm}}^{(k-1)}(e^{j\Omega}) \} = \\ &= \text{acos} \left(\frac{|\mathbf{x}_{\text{norm}}^{(k),H}(e^{j\Omega})\mathbf{x}_{\text{norm}}^{(k-1)}(e^{j\Omega})|}{\|\mathbf{x}_{\text{norm}}^{(k)}(e^{j\Omega})\|_2 \cdot \|\mathbf{x}_{\text{norm}}^{(k-1)}(e^{j\Omega})\|_2} \right). \end{aligned}$$

Once γ falls below a threshold ϵ , the process can be terminated. The corresponding dominant eigenvalue can be extracted as

$$\hat{\lambda}_1(z) = \tilde{\mathbf{x}}_{\text{norm}}^{(k),P}(z)\mathbf{R}(z)\tilde{\mathbf{x}}_{\text{norm}}^{(k)}(z). \quad (9)$$

This constitutes the polynomial version of the power method for spectrally majorised matrices, with the overall procedure summarised in Algorithm 1. For further details, please refer to [19].

IV. ORDINARY GENERALISED POWER METHOD

We know that the conventional reduced SVD of $\mathbf{A} \in \mathbb{C}^{M \times N}$ with $M \geq N$, given as $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H$ with $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{C}^{M \times N}$, $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_N] \in \mathbb{C}^{N \times N}$, can be obtained through the ordinary power method [20]. In order to determine the right-singular vectors, the power method can be applied to $\mathbf{A}^H\mathbf{A} \in \mathbb{C}^{N \times N}$ as its eigenvectors are in fact the right-singular vectors of \mathbf{A} . After $\hat{\mathbf{v}}_i = e^{j\phi}\mathbf{v}_i$, $i = 1, \dots, N$ has been found, the singular values and left-singular vectors can be obtained via

$$\sigma_i = \|\mathbf{A}\hat{\mathbf{v}}_i\|_2, \Rightarrow \hat{\mathbf{u}}_i = \frac{\mathbf{A}\hat{\mathbf{v}}_i}{\sigma_i} = e^{j\phi}\mathbf{u}_i, \quad i = 1, \dots, M, \quad (10)$$

where $e^{j\phi}$ is an arbitrary phase shift. Note that the phase ambiguity of the left- and right-singular vectors is coupled.

Alternatively, the left-singular vector can be determined by applying the power method to $\mathbf{A}\mathbf{A}^H$ and then the singular value can be computed later. Since left- and right-singular

vectors are determined independently, their phase ambiguities are no longer coupled. Hence, if $\hat{\mathbf{u}}_1 = e^{j\alpha}\mathbf{u}_1$ and $\hat{\mathbf{v}}_1 = e^{j\phi}\mathbf{v}_1$, the resulting estimated singular value will be $\hat{\sigma}_1 = \hat{\mathbf{u}}_1^H \mathbf{A} \hat{\mathbf{v}}_1 = e^{-j\alpha} \sigma_1 e^{j\phi}$ i.e. it will not in general be real-valued. However, real-valuedness, and therefore phase coupling of the left- and right-singular vectors, can be achieved by adjusting the phase of $\hat{\sigma}_i$. Such a procedure may not be possible in the case of polynomial matrices as will become clear in the following section.

This shows that the power iteration is not restricted to Hermitian matrices, but can indeed be applied to any matrix. This motivates us to combine the above concept with the already established polynomial power iterations, however. The aim is to drop the restriction to para-Hermitian matrices, such that the dominant singular vectors may be computed in first instance, with the option of later performing a full SVD of a polynomial matrix through deflation.

V. GENERALIZED POLYNOMIAL POWER METHOD

We now aim to extend the polynomial power method reviewed in Sec. III to generalise the SVD approach summarised in Sec. IV to the case of a polynomial matrix $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M \times N}$.

A. Polynomial Iterations Analysis

For an initial $\mathbf{x}^{(0)}(z) = \mathbf{V}(z)\mathbf{c}(z)$, the polynomial iteration can be applied to a para-Hermitian matrix $\mathbf{A}^P(z)\mathbf{A}(z)$ where after k iterations, with z substituted by $e^{j\Omega}$, we obtain

$$\begin{aligned} \mathbf{x}^{(k)}(e^{j\Omega}) &= \mathbf{A}^H(e^{j\Omega})\mathbf{A}(e^{j\Omega})\mathbf{x}^{(k-1)}(e^{j\Omega}) \\ &= \sum_{n=1}^N \mathbf{v}_n(e^{j\Omega})\sigma_n^{2k}(e^{j\Omega})\mathbf{v}_n^H(e^{j\Omega})\mathbf{V}(e^{j\Omega})\mathbf{c}(e^{j\Omega}), \end{aligned} \quad (11)$$

which can be re-written as

$$\begin{aligned} \mathbf{x}^{(k)}(e^{j\Omega}) &= \sigma_1^{2k}(e^{j\Omega}) \left[c_1(e^{j\Omega})\mathbf{v}_1(e^{j\Omega}) \right. \\ &\quad \left. + \sum_{m=2}^M c_m(e^{j\Omega}) \left(\frac{\sigma_m(e^{j\Omega})}{\sigma_1(e^{j\Omega})} \right)^{2k} \mathbf{v}_m(e^{j\Omega}) \right]. \end{aligned} \quad (12)$$

Since the singular values of $\mathbf{A}(z)$ are spectrally-majorised, so are the eigenvalues of the para-Hermitian matrix $\mathbf{A}^P(z)\mathbf{A}(z)$ i.e. $\sigma_n^2(e^{j\Omega}) \geq \sigma_{n+1}^2(e^{j\Omega})$ $n = 1, \dots, N-1$. Hence $\mathbf{x}^{(k)}(e^{j\Omega})$ converges to a scaled version of $\mathbf{v}_1(e^{j\Omega})$ for sufficiently large k similar to (6). Similarly, after normalization, we have

$$\mathbf{x}_{\text{norm}}^{(k)}(e^{j\Omega}) = \hat{\mathbf{v}}_1(e^{j\Omega}) = g_1(e^{j\Omega})\mathbf{v}_1(e^{j\Omega}) \quad \forall \Omega, \quad (13)$$

where $g_1(z)$ is an allpass filter. Similar to the power method for para-Hermitian matrix, this generalised approach also includes truncation and normalization in each iteration which can be found in the original algorithm reported in [19]. Similarly, the problem of singularities in $c_1(e^{j\Omega})$ can be handled either through regularization or modification to the initialization if a spectral zero is encountered.

The estimation of the dominant singular value and the corresponding left-singular vector is not straightforward and

needs careful consideration. Assuming, we follow the first method described in Sec. IV in (10), which is to determine the singular value and then the left-singular vector. For this method, the frequency dependent version for extracting the dominant singular value will be

$$\hat{\sigma}_m(e^{j\Omega}) = \|\mathbf{A}(e^{j\Omega})\hat{\mathbf{v}}_m(e^{j\Omega})\|_2, \quad m = 1, \dots, N, \quad (14)$$

which forces $\sigma_m(e^{j\Omega})$ to be positive $\forall \Omega$ due to the norm operator whereas the theory behind the analytic decomposition existence shows that the singular value can be negative on the unit-circle [21, 22]. Forcing the singular values to be positive violates this condition, thus with this method, the obtained decomposition might differ from the decomposition given in [18]. Alternatively, if the matrix $\mathbf{A}(z)$ is known to be positive semi-definite, the singular values are guaranteed to be real and positive and so this method gives the correct decomposition. Once the singular value is obtained with an acceptable accuracy, which we discuss further below, via (14) the dominant left-singular vector can be obtained as

$$\hat{\mathbf{u}}_1(e^{j\Omega}) = \mathbf{A}(e^{j\Omega})\hat{\mathbf{v}}_1(e^{j\Omega})/\hat{\sigma}_1(e^{j\Omega}) = g_1(e^{j\Omega})\mathbf{u}_1(e^{j\Omega}). \quad (15)$$

The allpass factor $g_1(e^{j\Omega})$ is the same as that of the right-singular vector, such that their ambiguities are coupled. This coupling results in the singular value being real-valued on the unit-circle. Both (14) and (15) can be implemented in DFT domain. Adjusting the size of this DFT is discussed further below.

The second method, described in Sec. IV, determines the left-singular vector by applying the polynomial power method to $\mathbf{A}(z)\mathbf{A}^P(z)$ and then computes the singular value as $\hat{\sigma}_1(z) = \hat{\mathbf{u}}_1^P(z)\mathbf{A}(z)\hat{\mathbf{v}}_1(z)$. This method does not impose the condition of singular value being positive on the unit circle, and so it can allow the analytic decomposition given in (1) to be achievable for any $\mathbf{A}(z)$. However, to retain real-valuedness for the singular values on the unit circle, the left- and right-singular vector have to have a common allpass factor. Thus if both the left- and right-singular vectors are independently extracted by applying the polynomial power method to $\mathbf{A}^P(z)\mathbf{A}(z)$ and $\mathbf{A}(z)\mathbf{A}^P(z)$, respectively, the allpass factor in $\hat{\mathbf{u}}_1(z)$ and $\hat{\mathbf{v}}_1(z)$ will, in general, not be coupled. Hence, the second method may not be desirable to be used unless a common phase shift can be found. Another reason for avoiding this method is invoking two polynomial power methods is computationally more expensive compared to the first approach.

To optimize the computational efficiency of the first method, we estimate the right singular vector using the polynomial power method in case of $M \geq N$. For the case $M \leq N$, all of the above approaches can be used instead to factorise $\mathbf{A}^P(z)$.

B. Sufficient DFT Size

1) *Dominant Singular Value:* Once $\mathbf{v}_1(z)$ is determined with Algorithm 1, the singular value can be determined via (14) in the DFT domain. To determine a sufficient DFT size,

time-domain aliasing can be utilized [2]. Thus (14) can be evaluated at increasing DFT sizes until

$$\xi_{\hat{\sigma}} = \sum_{\tau} \frac{|\hat{\sigma}_1^{(K)}[\tau] - \hat{\sigma}_1^{(K/2)}[\tau]|^2}{|\hat{\sigma}_1^{(K)}[\tau]|^2}, \quad (16)$$

where $\hat{\sigma}_1^{(K)}[\tau]$ represents the time-domain equivalent of (14) obtained with a K -point inverse DFT, until $\xi_{\hat{\sigma}}$ falls below a certain low threshold ε_1 . A small value of $\xi_{\hat{\sigma}}$ indicates that $K/2$ can be considered sufficient for approximate the dominant singular. If $\hat{v}_i(z)$ is extracted with satisfactory accuracy, a DFT size of $K = \mathcal{O}\{\mathbf{A}(z)\hat{v}_i(z)\} + 1$ should generally suffice, where $\mathcal{O}\{\cdot\}$ measures the polynomial order of its argument.

2) *Left Singular Vector*: Similarly, to determine a sufficient DFT size for (15), time-domain aliasing may be captured via the error w.r.t. normality in the time-domain as

$$\xi_u = \sum_{\tau} |\hat{\mathbf{u}}_1^H[-\tau] * \hat{\mathbf{u}}_1[\tau] - \delta[\tau]|_2^2, \quad \tau \in \mathbb{Z}. \quad (17)$$

A similar criterion has been utilised for the DFT size in [3]. There, it is shown as a necessary criterion; while sufficiency has not been proven, in practise it has generally been shown to suffice in all simulations.

It follows that for a sufficient DFT size ξ_u will be small since $\hat{\mathbf{u}}_1[\tau]$ should be normal. Thus (15) is implemented at increasing DFT size until ξ_u falls below a some given threshold ε_u .

VI. SIMULATIONS AND RESULTS

A. Numerical Example

To demonstrate the potential of the generalized polynomial power method, we assume a simple case of $\mathbf{A}(z)$ where we know the ground truth factorisation according to (1). For these factors, $\boldsymbol{\Sigma}(z) \in \mathbb{C}^{3 \times 2}$ contains

$$\sigma_1(z) = \frac{1}{2}z + 4 + \frac{1}{2}z^{-1}, \sigma_2(z) = \frac{1}{4}z + 1 + \frac{1}{4}z^{-1}, \quad (18)$$

which are spectrally majorised. The left-singular vector are constructed via elementary paraunitary operation given as [12]

$$\mathbf{U}(z) = \prod_{i=1}^2 \{\mathbf{I} - (1 - z^{-1})\mathbf{e}_i\mathbf{e}_i^H\}, \quad (19)$$

where $\mathbf{e}_{i=1,2} = [1, 1, \mp 1]^T / \sqrt{3} \in \mathbb{C}^3$ are unit-norm vectors. The right-singular vectors in $\mathbf{V}(z) \in \mathbb{C}^{2 \times 2}$ of order 2 are generated by the same approach with $\mathbf{e}_1 = [1, -1]^T / \sqrt{2}$ and $\mathbf{e}_2 = [-1, 0]^T$. The polynomial matrix $\mathbf{A}(z)$ is then defined as $\mathbf{U}(z)\boldsymbol{\Sigma}(z)\mathbf{V}^P(z)$.

Algorithm 1 is executed with $\epsilon = 10^{-12}$, $k_{\max} = 10^3$, $\mathbf{R}(z) = \mathbf{A}^P(z)\mathbf{A}(z)$ and $\mathbf{x}^{(0)}(z) = 1$. The truncation method employed is the order limitation [19] where the order of $\mathbf{x}^{(k)}(z)$ post-normalisation is limited to the estimated support obtained from [24]. Algorithm 1 converges in 44 iterations resulting in $\xi_v = 1.4 \times 10^{-11}$. Once the left-singular vector is estimated, the corresponding singular value is estimated via (14). With $K = 16$, the time-domain aliasing

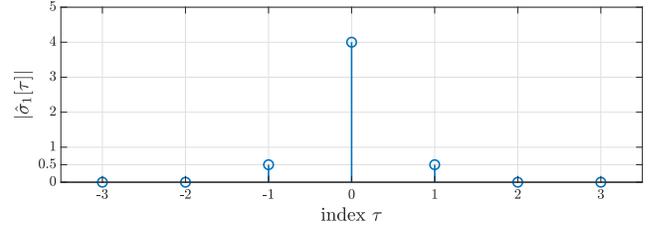


Fig. 1. Polynomial Power method based estimated dominant singular value coefficients for the numerical example.

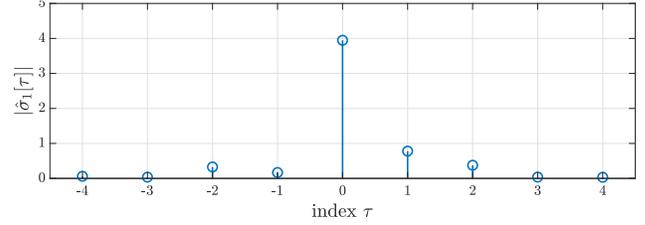


Fig. 2. GSR2 based estimated dominant singular value coefficients for the numerical example.

$\xi_{\lambda} = 8 \times 10^{-28}$. The trailing coefficients of $\hat{\sigma}_1[\tau]$ are truncated on either side of $\tau = 0$ via a threshold of 10^{-10} . This results in order of 6 whereas the ground-truth singular value has an order of 2. The coefficients are illustrated in Fig. 1 where the coefficients at $\tau = 0, \pm 1$ exactly match the ground-truth coefficients in (18), whereas the coefficients at $\tau = \pm 2, \pm 3$ are smaller than 10^{-5} . The normalized squared difference between the estimated and the ground-truth singular value, which can be defined similar to (16) as

$$\xi_{\sigma} = \sum_{\tau} \frac{|\sigma_1[\tau] - \hat{\sigma}_1[\tau]|^2}{|\sigma_1[\tau]|^2}, \quad (20)$$

is 3.5×10^{-12} . The corresponding left-singular vector is then obtained from (15) with a DFT of size $K = 16$. Thereafter, the order is limited by a shifted-truncation to 3, which achieves a metric of $\xi_v = 9 \times 10^{-12}$.

The GSR2 is executed with $\mu_{PU} = 10^{-5}$, $\epsilon = 10^{-5}$ and $\mu_{PH} = 10^{-5}$ for 1000 iterations and results in $\xi_v = 9.6 \times 10^{-5}$, $\xi_u = 1.5 \times 10^{-5}$ and $\xi_{\lambda} = 2.7 \times 10^{-2}$. The dominant singular value estimated with GSR2 has order 8 whose coefficients are illustrated in Fig. 2. It is evident that the estimated singular value is neither conjugate symmetric and nor the coefficients match the ground truth coefficients except at $\tau = 0$ where $\hat{\sigma}[0] = 3.95 \approx 4$. This loss of conjugate symmetry may be the cause of large value of ξ_{λ} .

B. Ensemble Test

In a more extensive test, we evaluate the proposed method against the Kogbetliantz transformation-based method [11] which we refer to as the generalised second order sequential best rotation (GSR2) algorithm via an ensemble consisting of 500 randomised instantiations of $\mathbf{A}(z) \in \mathbb{C}^{3 \times 2}$ such that each instance has $\mathcal{O}\{\mathbf{U}(z)\} = \mathcal{O}\{\mathbf{V}(z)\} = 10$ and $\mathcal{O}\{\boldsymbol{\Sigma}(z)\} = 20$. All the instantiations have spectrally majorised singular values.

TABLE I
PERFORMANCE COMPARISON OF GSR2 AND GPPM

Metrics	GSR2	GPPM
$\mathcal{O}\{\hat{\mathbf{u}}_1(z)\}$	966 ± 185	10
$\mathcal{O}\{\hat{\mathbf{v}}_1(z)\}$	422 ± 126	10
$\mathcal{O}\{\hat{\sigma}_1(z)\}$	96 ± 38	57 ± 4
ξ_v	$(1.2 \pm 0.8) \times 10^{-3}$	$(5.5 \pm 4.5) \times 10^{-5}$
ξ_u	$(1.6 \pm 0.85) \times 10^{-3}$	$(5.5 \pm 4.5) \times 10^{-5}$
ξ_σ	0.09 ± 0.07	$(1.5 \pm 1.3) \times 10^{-5}$
time(s)	0.67 ± 0.15	0.44 ± 0.19

For the proposed method, Algorithm 1 is simulated with $\epsilon = 10^{-10}$, $k_{\max} = 10^3$, $\mathbf{R}(z) = \mathbf{A}^P(z)\mathbf{A}(z)$ and $\mathbf{x}^{(0)}(z) = 1$. The order of the product vector is limited to 10, with its order estimated through the method in [24], followed by shifted-truncation [19, 25]. The corresponding singular value and the left-singular vector are extracted at $K = 2^{\lceil \log_2(\mathcal{O}\{\mathbf{A}(z)\hat{\mathbf{u}}_1(z)\}) \rceil}$ where $\lceil \cdot \rceil$ denotes ceiling operation. The trailing coefficients of the estimated left-singular are truncated below a threshold of 10^{-10} while the right-singular vector is similarly order-limited to its estimated support. GSR2 is simulated with $\mu_{\text{PU}} = 10^{-4}$, $\mu_{\text{PH}} = 2 \times 10^{-10}$ employing the original truncation method of SBR2/SMD [4, 5]. The algorithm is allowed to perform a maximum of 200 iteration; however, the execution is terminated if the off-diagonal terms fall below 10^{-6} .

The ensemble average for all the metrics is shown in Table I. It is evident that the proposed method provides a more compact order approximation for both the left- and right-singular vectors and the singular value compared to the GSR2. Moreover, the error metrics ξ_u and ξ_v of the proposed method's extracted singular vectors reach orders of magnitude below those obtained with GSR2. Likewise, the normalized squared difference between the estimated and ground-truth singular value is orders of magnitude lower for the polynomial method than GSR2. The potential reason for the large deviation of the GSR2's estimated singular value is likely the imperfect conjugate symmetry due to the uncoupled allpass ambiguity of the estimated left- and right-singular vectors. Also, SBR2-type algorithms are known to only achieve a relatively poor diagonalisation compared to their DFT-domain counterparts in e.g. [2, 3].

VII. CONCLUSION

The polynomial power method, which was initially proposed for para-Hermitian matrices, has been extended into the generalized polynomial power method for computing the dominant left- and right-singular vectors and their corresponding singular value of a polynomial matrix. The proposed extension provides better estimation of the singular vectors with lower order approximation as compared to the only direct PSVD algorithm based on the Kogbetliantz method. The proposed method promises better results and can be further utilized to compute the PSVD of a polynomial matrix through the polynomial matrix deflation analogous to ordinary matrix deflation.

REFERENCES

- [1] S. Weiss, J. Pestana, and I. K. Proudler, "On the existence and uniqueness of the eigenvalue decomposition of a parahermitian matrix," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2659–2672, 2018.
- [2] S. Weiss, I. K. Proudler, and F. K. Coutts, "Eigenvalue decomposition of a parahermitian matrix: Extraction of analytic eigenvalues," *IEEE Transactions on Signal Processing*, vol. 69, pp. 722–737, 2021.
- [3] S. Weiss, I. Proudler, F. Coutts, and F. Khattak, "Eigenvalue decomposition of a parahermitian matrix: extraction of analytic eigenvectors," *IEEE Transactions on Signal Processing*, vol. 71, pp. 1642–1656, 2023.
- [4] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster, "An EVD algorithm for para-Hermitian polynomial matrices," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2158–2169, 2007.
- [5] S. Redif, S. Weiss, and J. G. McWhirter, "Sequential matrix diagonalization algorithms for polynomial EVD of parahermitian matrices," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 81–89, 2015.
- [6] V. W. Neo, C. Evers, and P. A. Naylor, "Speech enhancement using polynomial eigenvalue decomposition," in *IEEE WASPAA*, 2019, pp. 125–129.
- [7] S. Redif, J. G. McWhirter, P. D. Baxter, and T. Cooper, "Robust broadband adaptive beamforming via polynomial eigenvalues," in *OCEANS*, 2006, pp. 1–6.
- [8] S. Redif, S. Weiss, and J. G. McWhirter, "Relevance of polynomial matrix decompositions to broadband blind signal separation," *Signal Processing*, vol. 134, pp. 76–86, 2017.
- [9] J. A. Foster, J. G. McWhirter, M. R. Davies, and J. A. Chambers, "An algorithm for calculating the QR and singular value decompositions of polynomial matrices," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1263–1274, 2010.
- [10] Z. Wang, A. Sandmann, J. G. McWhirter, and A. Ahrens, "Multiple shift SBR2 algorithm for calculating the SVD of broadband optical MIMO systems," in *Int. Conf. TSP*, 2016, pp. 433–436.
- [11] J. G. McWhirter, "An algorithm for polynomial matrix SVD based on generalised kogbetliantz transformations," in *EUSIPCO*, 2010, pp. 457–461.
- [12] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. USA: Prentice-Hall, Inc., 1993.
- [13] N. Moret, A. Tonello, and S. Weiss, "MIMO precoding for filter bank modulation systems based on PSVD," in *IEEE VTC*, 2011, pp. 1–5.
- [14] W. Al-Hanafy, A. P. Millar, C. H. Ta, and S. Weiss, "Broadband SVD and non-linear precoding applied to broadband MIMO channels," in *Asilomar Conf. SSC*, 2008, pp. 2053–2057.
- [15] S. Weiss and I. Proudler, "Comparing efficient broadband beamforming architectures and their performance trade-offs," in *DSP*, vol. 1, pp. 417–423, 2002.
- [16] G. E. Forsythe and P. Henrici, "The cyclic jacobi method for computing the principal values of a complex matrix," *Transactions of the American Mathematical Society*, vol. 94, no. 1, pp. 1–23, 1960.
- [17] G. Barbarino and V. Noferini, "On the Rellich eigendecomposition of para-hermitian matrices and the sign characteristics of *-palindromic matrix polynomials," *Linear Algebra & Its Appl.*, vol. 672:1–27, 2023.
- [18] S. Weiss, I. K. Proudler, J. P. G. Barbarino, and J. G. McWhirter, "On properties and structure of the analytic singular value decomposition," submitted 2023.
- [19] F. A. Khattak, I. K. Proudler, and S. Weiss, "Extension of power method to para-hermitian matrices: Polynomial power method," in *EUSIPCO*, accepted 2023.
- [20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.
- [21] B. De Moor and S. Boyd, "Analytic properties of singular values and vectors," KU Leuven, Tech. Rep., 1989.
- [22] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nicols, "Numerical computation of an analytic singular value decomposition of a matrix valued function," *Numer. Math.*, vol. 60, pp. 1–40, 1991.
- [23] F. A. Khattak, S. Weiss, I. K. Proudler, and J. G. McWhirter, "Space-time covariance matrix estimation: Loss of algebraic multiplicities of eigenvalues," in *Asilomar Conf. SSC*, Pacific Grove, CA, Oct. 2022.
- [24] F. A. Khattak, I. K. Proudler, and S. Weiss, "Support estimation of analytic eigenvectors of parahermitian matrices," in *2022 Int. Symp. RAEE & CS*, 2022.
- [25] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter, "Row-shift corrected truncation of paraunitary matrices for PEVD algorithms," in *EUSIPCO*, Nice, France, Aug. 2015, pp. 849–853.

Joint optimization of sonar waveform selection and sonobuoy placement

Christopher M Taylor*, Simon Maskell, Alexey Narykov and Jason F Ralph
 Department of Electrical Engineering and Electronics, University of Liverpool, U.K.
 Email: *christopher.taylor2@liverpool.ac.uk

Abstract—We propose a method to jointly optimize placement of a sonobuoy field and selection of appropriate active sonar waveforms in a complex undersea environment, using deep reinforcement learning. To this end, we develop a custom simulation environment and train an online algorithm that uses updated information as sonobuoys are placed and sonar measurements made. The approach may be termed multiple bistatic, though it shares features with multistatic approaches. We develop benchmarks using conventional fixed sonobuoy placement patterns together with fixed composite sonar pulse trains and present results that show our new approach can outperform the best conventional combination of fixed placement pattern and pulse train.

I. INTRODUCTION

The problems of optimizing sonobuoy placement patterns and schedules [1], [2], [3], [4], and of optimizing waveform selection and pulse train construction [5], [6], [7], [8], [9] for sonar and radar, both have a long history in the literature. Our new work suggests that there is a relationship between the problems and that optimizing both sonobuoy placement and waveform selection jointly can have performance benefits.

The sonobuoy placement problem in a complex environment was examined in [10], [11], using a two-stage process, combining an evolutionary algorithm with tabular reinforcement learning. We now address the more complex problem in which an optimal combination of sonobuoy placement schedule and waveform sequence is sought online using updated information throughout a mission. To achieve this, we use deep reinforcement learning (DRL) [12] to track an uncooperative and stealthy target in a simulation of complex and noisy underwater environment. The environment is defined and parametrized in such a way that tracking is difficult, and there are several stochastic elements, adding to the difficulty in learning optimal actions.

The rest of this paper is organized as follows. In Section II we describe the scenario and the modelling used to construct the simulations and the custom environment we have developed to simulate the problem. In Section III we describe simulations performed with fixed sonobuoy placement patterns and sonar pulse trains, to investigate the joint effects of varying these. In Section IV, we describe our custom environment and the DRL algorithm, and present initial results.

II. SCENARIO AND MODELLING

Consider a scenario in which we wish to localize a single underwater target of interest (TOI), given a fixed tolerance

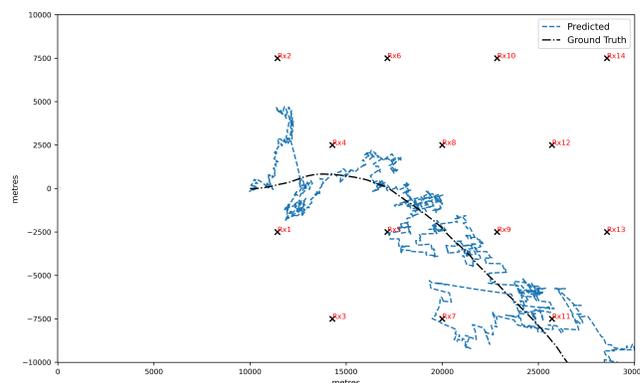


Figure 1: An example of randomly generated ground truths (in black) and predicted track (in blue). The positions of 14 sonobuoys in a staggered lattice pattern of 5 columns are each marked with an x . The base and sonar transmitter are located at (0,0). The placing agent uses a predetermined striping flight plan, starting from the nearest column to the base; the numbering indicates the order in which the sonobuoys are placed. The predicted track may be more or less noisy depending on the combination of ground truth, noise/clutter map, sonobuoy pattern and pulse train.

for uncertainty around the target position. At the outset, we know that the TOI is present within a defined rectangular area of interest (AOI), the nearest edge of which is a significant distance from a static base of operations which is located at the origin in our coordinate system. This base is the starting location of a sensor placing agent (such as a helicopter or UAV), and active sonar pulses are transmitted from the base. We assume that the TOI travels at a constant shallow depth.

During a mission, the agent moves to place the sonobuoys in selected locations. The agent has a limited payload of sonobuoys which it can place anywhere within the AOI, subject to completing placement and returning to base within a given maximum mission time. Depending on the schedule chosen, the agent may not be able to place its entire payload of sonobuoys. Rather than a full flight schedule being fixed in advance, as considered in [10], [11], updated information is used after placement of each sensor to decide on the next placement location.

Once the first sensor is placed, at fixed intervals a single high-powered transmitter located at the static base can send out either a continuous wave (CW) waveform or a linear frequency modulated (LFM) active sonar waveform. At time t each of

the N_t sonobuoys already placed provides information in the form of a state vector:

$$\mathbf{z}_{i,t} = [\theta_{i,t}, r_{i,t}, r_{i,t}]^T, \quad i = 1 \dots N_t, t = 1 \dots T_{max},$$

the elements of which represent bearing, range and range-rate, respectively. LFM pulses provide superior range performance but inferior range-rate performance compared to CW pulses [13], [14], [15]. The use of a composite pulse train allows the sensors to optimise the information extracted from the TOI.

For the examples shown in this paper, the chosen mission goal is to maximize the amount of time during which position estimates $(\hat{x}_{1,t}, \hat{x}_{2,t})$ have errors below a certain threshold value η , expressed in metres. The true constrained optimization problem can be stated as:

Maximize

$$\sum_{t=1}^T \mathbf{1} \left(\sqrt{(x_{1,t} - \hat{x}_{1,t})^2 + (x_{2,t} - \hat{x}_{2,t})^2} \leq \eta \right), \quad (1)$$

s.t.

$$\begin{aligned} N_t &\leq N_{max}, \\ T &\leq t_{max}, \end{aligned} \quad (2)$$

where N_{max} is the maximum payload of the placing agent and T and t_{max} are the actual time that passes during the mission and the maximum time for the mission, respectively, and $\mathbf{1}(x)$ is the indicator function, which returns 1 if errors are below the threshold and 0 otherwise. Hence the goal is to maximize the number of time periods in which the error comes within the threshold tolerance. The mission will terminate early if the target leaves the AOI before the maximum time, and the total number of sensors placed may not reach the maximum payload if the total travel time does not allow placement of all the sonobuoys. Total travel time includes the time spent travelling to and from the base, the time travelling between placement locations, and the time spent at each location during the placement process. However, since we only have access to $(\hat{x}_{1,t}, \hat{x}_{2,t})$ but not to ground truth, we cannot address Equation 1 directly during training of the algorithm, and so later we will substitute a measure of uncertainty over the measurements into the optimization problem as a proxy for the true objective during training of the algorithm. Equation 1 will still be used in validation, however.

In our initial modelling, N passive receivers, such as DIFAR sonobuoys [16], have been placed in a predetermined grid pattern of N sensors, optimized for equal coverage for the entire AOI. The approach may be termed multiple bistatic, though it shares features with multistatic approaches. The sonobuoys are assumed to be equipped with GPS or other geolocation system of similar accuracy, and to have an operational life limited only by a scuttling time that exceeds the maximum mission time [17]. The transmitter has a choice at each timestep t to transmit one of two waveforms, LFM or CW, and the receivers each report measurements in the form of estimated speed, bearing and distance of the target, together with measures of uncertainty. We assume that with high-powered active sonar, probability of detection approaches 1.

Within the simulations, we use an unscented Kalman filter [18] (UKF)-based tracker for sensor fusion and localization. Ground truth is generated using a 2D linear Gaussian constant velocity model. Initial position, course and speed are also random, with the target initially placed within a large central area inside the AOI. The measurement state space is calculated using the bistatic Doppler shift equation in [19], calculated for each sonobuoy as follows:

$$\mathbf{z}_{i,t} = \begin{bmatrix} \frac{\text{atan2}(x_{1,t}, x_{2,t})}{C} \cdot \left(\sqrt{x_{1,t}^2 + x_{2,t}^2} + \sqrt{(x_{1,t} - p_{i,1})^2 + (x_{2,t} - p_{i,2})^2} \right) \\ - \frac{\lambda}{2} \cdot \frac{\sqrt{(x_{1,t} - p_{i,1})^2 + (x_{2,t} - p_{i,2})^2} \cdot \sqrt{(x_{1,t})^2 + (x_{2,t})^2}}{\sqrt{x_{1,t}^2 + x_{2,t}^2}} \end{bmatrix}, \quad (3)$$

where atan2 is the standard 2-input arctangent function, $\mathbf{p}_{i,t}$, $i = 1 \dots N_t$ is the position vector of the i th placed sonobuoy, λ is the wavelength of the carrier signal and C is a constant representing the local speed of sound in water. As in [10], [11], a random, spatially-varying environment noise/clutter map is generated using D bivariate Gaussians with random means and covariance matrices. This map is used to calculate line integrals along the paths taken by the sonar pulses, but in this scenario these are used to add noise to the measurements used by the filter, as well as acting as a proxy for transmission loss. The measurement covariance parameters reflect the superiority of CW over LFM for Doppler resolution and superiority of LFM over CW for range resolution. The tracker then produces a state space estimate $\hat{\mathbf{x}}_t = [\hat{x}_{1,t}, \hat{x}_{1,t}, \hat{x}_{2,t}, \hat{x}_{2,t}]$ for the target after updating with all available measurements from the sensors that have been placed.

Several simplifying assumptions have been used in the modelling and simulation, in particular the following:

- Sonobuoys are identical, do not drift, do not fail, and their position is known with certainty;
- There is a single TOI which is always detected by all sonobuoys, albeit with possibly very high uncertainty, with no false alarms or sensor failure.

Note that even with these simplifications, sensors that are a long distance away from the target will supply little useful information.

III. INITIAL SIMULATIONS

To investigate behaviour with traditional approaches to sonobuoy placement and waveform selection and establish benchmarks for performance, we first considered a scenario where an agent (such as a helicopter) requires a fixed flight plan in advance and places sonobuoys in a fixed, staggered lattice pattern; see Figure 1. The nearest edge of the AOI is placed 10km away from the base at the origin where the placing agent is initially located. The transmitter at the base uses a repeating pulse train which is a predetermined sequence of CW and LFM pulses; for example, the pulse train $pt = [CW, LFM, LFM, CW, LFM, LFM]$ has length 6 and mean $\mu_{pt} = 0.66$, calculated using a value of 1 for each LFM

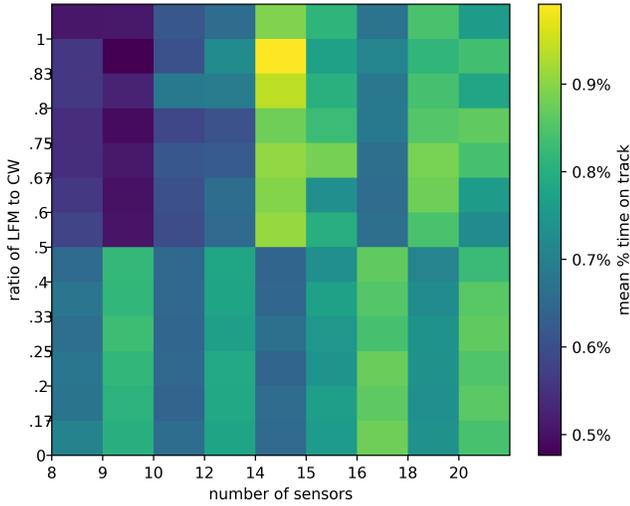


Figure 2: Heat map of aggregated mean time-on-track across 544,500 combinations of fixed sonobuoy placement patterns and composite sonar pulse trains. The colour represents the mean time-on-track percentage, that is, the percentage of time steps for which the localization error is below the required threshold. This generally increases with the number of sonobuoys, but whilst pulse trains with a higher proportion of LFM do better for patterns with larger numbers of sonobuoys, pulse trains with a lower proportion of LFM do better for patterns with larger numbers of sonobuoys.

pulse and 0 for each CW pulse, so that μ_{pt} represents the proportion of LFM pulses within the pulse train. Sonobuoy placement patterns based on staggered lattices were generated for numbers of sonobuoys in the range $N = 8 \dots 20$. We use a fixed set of $GT = 30$ randomly generated ground truth paths, which start at a central point with initial values within the state vector \mathbf{x}_t drawn from a Gaussian distribution but subject to several random changes during the course of the mission. We also generated a fixed set of $NC = 30$ random generic noise/clutter maps. We then ran all possible combinations of ground truth path, noise/clutter map, pulse train, and sonobuoy lattice placement pattern.

If the number of pulses in the repeating pattern is $k = 2, 3, \dots, K$, then in general there are $(K^3 + 3 \cdot K^2 + 2 \cdot K - 6) / 6$ combinations of two waveforms to consider; we used $K = 6$, so that there were 55 combinations in total. We considered a single lattice configuration for each non-prime number $\in 2, \dots, 20$; we omit prime numbers to ensure all columns of the lattice have the same number of sensors. If N_s represents the number of sonobuoy patterns, then the total number of unique scenarios is $GT \cdot NC \cdot N_s \cdot (K^3 + 3 \cdot K^2 + 2 \cdot K - 6) / 6$, so we conducted 544,500 experiments in total. We then calculated time-on-track, that is, the amount of time the tracker error is less than $\eta = 200\text{m}$, for each scenario, using Equation 1. Figures 2 and 3 show heat maps of mean scores and score variance respectively, aggregated with respect to the ratio of LFM within the pulse train and the number of sonobuoys in the placed pattern.

One conclusion from the simulation results is that fixed

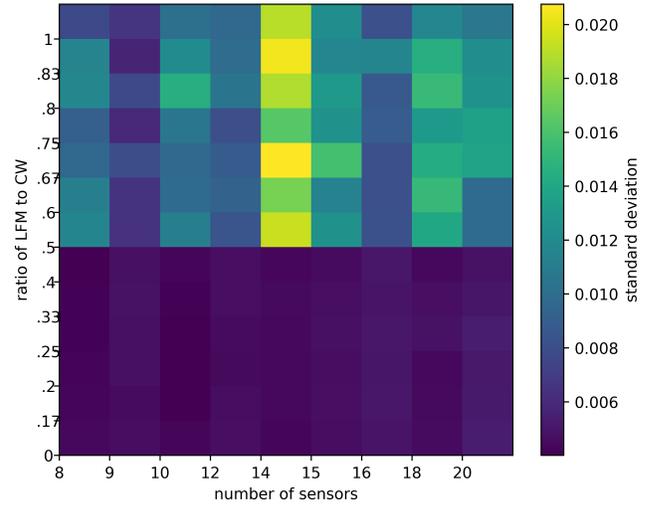


Figure 3: Heat map of aggregated time-on-track standard deviations across all simulations. The standard deviation of time-on-track is higher for LFM proportions above 0.5, but with respect to the number of sensors, shows high correlation with the means shown in Figure 2, suggesting a trade-off between mean and variance.

patterns with higher numbers of placed sensors are not always superior to ones with lower N . Although patterns with higher N will give more information once placed, patterns with smaller N are deployed faster over the whole AOI, potentially giving more information to the tracker sooner, depending on the initial location of the target and its subsequent movements. The results also suggest a link between waveform selection and sonobuoy placement pattern, in particular that different patterns benefit from different proportions of LFM and CW pulses.

IV. EXPERIMENTS WITH DEEP REINFORCEMENT LEARNING

To facilitate experiments using DRL, we refactored our simulation software as a custom simulation environment using the OpenAI Gym [20] paradigm. In designing such an environment, the choice of observation space and action space is key. In many standard test environments, such as Atari games, the observation space is a preprocessed version of what is seen on screen when the game is rendered, but we chose to construct the observation space directly from available information, so that it consists of the following:

- x -position of each sensor (size: N)
- y -position of each sensor (size: N)
- bearing prediction from each sensor (size: N)
- range from each sensor (size: N)
- range-rate from each sensor (size: N)
- predicted state vector for the target from the tracker (size:4)
- position covariance from the tracker (size:4)
- prediction variance from the tracker (size:2)
- steps remaining to time limit (size:1)

The total size of the observation space is $5 \cdot N + 11$. The inputs are largely self-explanatory; the final input is necessary because episode lengths are variable, as an episode will terminate before the time limit if the target leaves the AOI before this time, and the algorithm needs some concept of steps remaining to time limit in order to learn to adjust observed episode rewards [21]. The information from sensors starts as all zeros before any sonobuoys are placed and readings are added only after each sonobuoy is placed. The algorithm has access only to the resultant noisy and uncertain measurements from the sensors.

For the purpose of deciding sonobuoy placement locations, the environment is discretized into a grid determined by the dimensions of the AOI and a scaling factor. We used a $20\text{km} \times 10\text{km}$ rectangle and a 200m scale, meaning there are initially 5,000 locations available for sensor placement, and 6.89×10^{40} possible placement patterns of up to 14 sonobuoys. A Gaussian error is added to the actual placement location when the sensor is placed. The nearest edge of the AOI is placed 10km away from the base, as in Section III.

The total action space consists of these locations plus two waveform selection choices and a choice of no action. To reduce this very large action space, and to deal with the fact that the sonobuoy placement actions and waveform selection actions have different timescales, we employ action masking [22], [23], meaning that only actions which can be taken at a given step are available to the algorithm at that step. At the first step, the algorithm selects the first sonobuoy placement location, and subsequently only the ‘no action’ option is available until that first sensor is placed. Thereafter at steps immediately after a sensor is placed, all locations other than ones at which sensors are already placed are available to schedule sensor placement until all sensors are placed or the episode ends; at all other steps, the action space reduces to a binary decision between transmitting one of the two waveforms (CW or LFM). The choice of placement locations is also constrained by the requirement that the placing agent be able to return to base from the next placement location before the maximum time for the episode is reached.

For training we used the Proximal Policy Optimization (PPO) [24] algorithm, which makes use of clipping of the value function to avoid large updates and balance exploitation and exploration. Policy gradient methods [25] compute an estimator of the policy gradient and pass this to a stochastic gradient descent algorithm; we used the Adam optimizer [26] for this. PPO is easier to implement than many other DRL algorithms because it requires only first-order gradients, employs multiple workers and does not require a replay buffer. We trained the algorithm with an initial payload of 14 sonobuoys. A new ground truth and noise/clutter map was randomly generated for each episode. We cannot use time-on-track for the reward function in training, as this requires access to ground truth, so as a proxy at each timestep the reward is set at 1 if the norm of the prediction covariance matrix from the tracker is below a given threshold, and zero otherwise.

For validation, we randomly generated a new set of 100

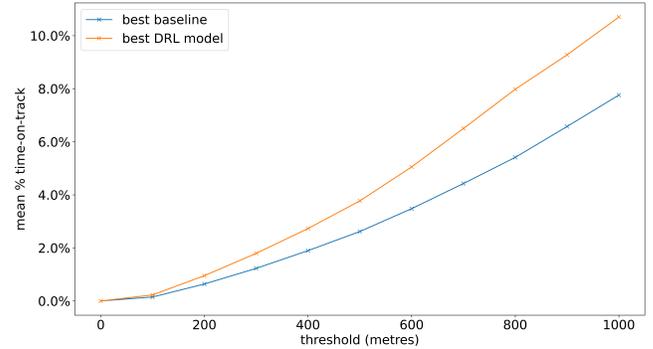


Figure 4: Mean % time-on-track for the model at different localization error thresholds. Time-on-track is calculated using different values up for the threshold value η averaged over 100 different environments. The best DRL model outperforms the best baseline at all error thresholds.

environments, each again with a unique ground truth path and noise/clutter map; ran each of the checkpoint models using each of these environments; and recorded time-on-track as per the methodology in Section III. We compared results to the baseline combination from Section III that showed best results, namely a fixed pattern of 14 sonobuoys combined with a fixed repeating pulse train $pt = [LFM, LFM, CW, LFM, LFM, CW]$.

We found that the DRL model trained for 8 million steps performed best, and outperformed the baseline at all thresholds; see Figure 4. Training improves performance up to this checkpoint, but out-of-sample performance deteriorates thereafter, as further training leads to overfitting [27].

V. DISCUSSION AND FUTURE WORK

We have shown that jointly optimizing sonobuoy placement patterns and composite sonar pulse trains based on updated information from sensors can improve localization performance in a complex and noisy environment where tracking is difficult. Results from simulations with fixed sonobuoy patterns and composite sonar pulse trains show different patterns benefit from different proportions of LFM and CW pulses, and that patterns with higher numbers of sensors are not always superior, because of the increased deployment time. Preliminary results for the DRL algorithm running with our custom simulation environment show promising improvements in performance compared to the untrained model.

In future work we will look to improve the performance of the algorithm further by investigating alternatives to PPO, increasing training time and changing the configuration of the observation space. We also envisage adding further elements that increase the realism of the problem. This may include: modelling initial sensor payload as a random variable; other stochastic elements such as imprecise placement and drift of sonobuoys; accounting for the attitude of the target to the sensors; sensor variability or failure; and multiple targets and false alarms.

ACKNOWLEDGEMENT

The authors would like to thank DSTL Grant no. 1000143726 for financial support.

REFERENCES

- [1] D. R. DelBalzo, D. N. McNeal, and D. P. Kierstead, "Optimized multi-static sonobuoy fields," in *Proceedings of OCEANS 2005 MTS/IEEE*, IEEE, 2005, pp. 1193–1198.
- [2] D. R. Delbalzo and K. C. Stangl, "Design and performance of irregular sonobuoy patterns in complicated environments," in *OCEANS 2009*, IEEE, 2009, pp. 1–4.
- [3] R. Grasso, M. Cococcioni, B. Mourre, J. Osler, and J. Chiggiato, "A decision support system for optimal deployment of sonobuoy networks based on sea current forecasts and multi-objective evolutionary optimization," *Expert Systems with Applications*, vol. 40, no. 10, pp. 3886–3899, 2013.
- [4] M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, "Multisensor resource deployment using posterior Cramér-Rao bounds," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 399–416, 2004.
- [5] D. J. Kershaw, S. Member, and R. J. Evans, "Optimal Waveform Selection for Tracking Systems," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1536–1550, 1994.
- [6] A. I. Larsson and C. Gillard, "On waveform selection in a time varying sonar environment," in *ACOUSTICS 2004*, 2004, pp. 73–78.
- [7] V. S. Veeramachaneni, "Intelligent Selection of Waveform Based on Predicted Target State for Active Sonar," MSc Thesis, George Mason University, 2019.
- [8] C. E. Thornton, R. M. Buehrer, H. S. Dhillon, and A. F. Martone, "Universal Learning Waveform Selection Strategies for Adaptive Target Tracking," 2022. [Online]. Available: <http://arxiv.org/abs/2202.05294>
- [9] X. Cao, Z. Zheng, and D. An, "Adaptive waveform selection algorithm based on reinforcement learning for cognitive radar," in *AUTEEE 2019*, 2019, pp. 208–213.
- [10] C. M. Taylor, S. Maskell, and J. F. Ralph, "Optimizing sonobuoy placement using multiobjective machine learning," in *2022 Sensor Signal Processing for Defence Conference (SSPD)*, no. August. IEEE, 2022, pp. 86–90.
- [11] —, "Using hybrid multiobjective machine learning to optimize sonobuoy placement patterns," *IET Radar, Sonar and Navigation*, vol. 17, no. 3, pp. 374–387, 2023.
- [12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," 2013.
- [13] O. P. Azar, H. Amiri, and F. Razzazi, "Enhanced target detection using a new combined sonar waveform design," *Telecommunication Systems*, vol. 77, no. 2, pp. 317–334, 2021. [Online]. Available: <https://doi.org/10.1007/s11235-021-00761-6>
- [14] M. Erdem, E. Evrigen, and Z. Özcan, "Performance analysis of coded waveform sonar," in *UDT 2019*, no. 1. Meteksan Defence, 2019, p. 5.
- [15] A. Waite, *Sonar for Practising Engineers*, third edit ed. Wiley, 2002.
- [16] R. Holler, "Evolution of the Sonobuoy from World War II to the Cold War," *U.S. Navy Journal of Underwater Acoustics*, vol. 25, no. January, pp. 322–346, 2014. [Online]. Available: <http://www.dtic.mil/dtic/tr/fulltext/u2/a597432.pdf>
- [17] S. Rankin, B. Miller, J. Crance, T. Sakai, and J. L. Keating, "Sonobuoy Acoustic Data Collection during Cetacean Surveys," Tech. Rep. April, 2019.
- [18] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *IEEE AS-SPCC 2000*, 2000, pp. 153–158.
- [19] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond The Kalman Filter*. Artech House, 2003.
- [20] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," 2016. [Online]. Available: <http://arxiv.org/abs/1606.01540>
- [21] F. Pardo, A. Tavakoli, V. Levdiuk, and P. Kormushev, "Time limits in reinforcement learning," *35th International Conference on Machine Learning, ICML 2018*, vol. 9, pp. 6443–6452, 2018.
- [22] S. Huang and S. Ontañón, "A Closer Look at Invalid Action Masking in Policy Gradient Algorithms," in *FLAIRS*, vol. 35, 2022, pp. 1–10.
- [23] T. Tao, D. Reda, and M. van de Panne, "Evaluating Vision Transformer Methods for Deep Reinforcement Learning from Pixels," 2022. [Online]. Available: <http://arxiv.org/abs/2204.04905>
- [24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017. [Online]. Available: <http://arxiv.org/abs/1707.06347>
- [25] V. Mnih, A. P. Badia, L. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML 2016*, vol. 4, 2016, pp. 2850–2869.
- [26] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*, 2015, pp. 1–15.
- [27] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, "A Study on Overfitting in Deep Reinforcement Learning," *arXiv preprint*, pp. 1–25, 2018. [Online]. Available: <http://arxiv.org/abs/1804.06893>

Development of the Line Scatterer Model for Bistatic Wind Turbine Clutter

Juhani Nissilä*, Pasi Pertilä[†], Minna Väilä[‡], Juha Jylhä[§]
Emerging Technology Research (ETR), Systems & Integrations Operations
Patria Aviation Oy

ORCID: 0000-0001-6908-0406*, 0000-0003-1413-6091[†], 0000-0002-4786-3474[‡], 0000-0002-6731-3939[§]

Abstract—The scattering of electromagnetic signals from wind turbines is a potential cause of clutter especially for passive radars which may use for example commercial Digital Video Broadcasting – Terrestrial (DVB-T) or frequency modulation (FM) broadcasting transmissions as transmitters of opportunity. These transmitters typically have a nearly constant gain in azimuth and thus they will illuminate the wind turbines at all times if they are visible to the transmitter mast. Understanding the wind turbine clutter is a first step in mitigating its negative effects. A previously published analytic line scatterer model for wind turbines is developed further in this work. Elevation angles are taken into account and thus the transmitter, wind turbine and receiver locations are no longer limited to the plane of same height. Furthermore, the amplitude of the scattered signal in the line scatterer model is investigated in detail and compared to radar cross-section (RCS) simulations from a 3D model of a wind turbine rotor. Based on this comparison, frequency dependent models for the amplitude are developed. The model development shows that wind turbine RCS can be highly dependent on frequency and view angle, but simplified models can also give quite good predictions. One possible application of the model is demonstrated by simulating a passive radar signal processing scenario where the effect of the blade flashes can be seen in the cross-ambiguity function (CAF) computation.

Index Terms—Electromagnetic scattering, Wind energy, Wind turbine, Bistatic radar, Passive radar, RCS simulation

I. INTRODUCTION

Modern wind turbines are seen as large reflective objects by radars and sources of both stationary and doppler shifted clutter. For this reason, considerable amount of both simulations and experimental research work has been done to both understand and mitigate the problem during the past decade. The accelerating shift towards greener energy production means that the number and size of wind turbines will continue to increase in the future.

Wind turbine scattering has been modeled using for example a finite number of point scatterers placed at regular intervals on the blades' locations [1]. This kind of modeling is typical for studying various Doppler modulation effects caused by for example the target's vibration or rotation and they are referred to as micro-Doppler effects [2]. An analytic formula for the scattered response of wind turbine blades was developed in [3] by integrating point scatterers along the length of the blades. It was assumed in [3] that the transmitter, wind turbine, and receiver are all located at the same height. There was also no mention about scaling the amplitude of the scattered signal based on RCS simulations or measurements. This model is

further developed in this study by including the elevation angles into the formula and developing a model for scaling the scattering amplitudes realistically. We also note that a similar model was developed earlier in [4] by considering the far-field electromagnetic field of infinitesimal dipoles placed along the blade and integrating along the length of the blades for the complete field. The simulation results were carried out using numerical integration, so no closed form formula was presented. In [5], the micro-Doppler modeling was combined with simple multipath considerations as possible sources of more irregular scattering returns from wind turbines.

Spectrograms calculated from radar returns of wind turbines have been reported in multiple papers during the previous decades. Several measured time series of wind turbine clutter in the S- and X-band are documented in [6]. Measurements of wind park clutter recorded by a passive radar is documented in [7]. Also the RCS of a scaled down laboratory model is studied in [8] and compared to simulations and measurements.

Some interesting methods for removing wind turbine clutter as post processing from weather radar CAFs have been presented for example in [9] and [10]. In [9], sparse signal processing is used by assuming that the weather data have a groups sparse property and the wind turbine data have a sparse time derivative. In [10], the CAFs are analysed as images via the Radon transform and the vertical bands are mainly caused by the wind turbines. These can be removed in the Radon domain by filtering the low angle data. It should be noted that although these post processing methods may remove the blade flashes, they will not remove the base noise level caused by the blade flashes in the CAF computation. This phenomena of larger target echo sidelobes masking the detection of smaller ones across all ranges and velocities is described in [11, Ch. 5.1]. The sidelobes of wind turbine flashes may cause such masking of targets especially if a passive radar is placed so close to the wind turbine that it is in its line of sight.

Typically in passive radar processing one needs to digitally remove at least the zero-frequency components before the computation of the CAF. This removes the multipath interferences from stationary objects. But wind turbine clutter can be present at quite large Doppler frequencies, and to remove it, would mean extending the filtering to certain range-Doppler areas away from the zero-frequency line. Adaptive digital clutter removal algorithms are discussed for example in [12] and especially for Orthogonal Frequency Division

Multiplexing (OFDM) modulation signals in [13], [14].

The strong radar returns from wind turbines are not always an undesired effect, as there has also been research aimed at utilizing dedicated radar measurements of wind turbine scattering for online monitoring of wind turbine operation and their structural health [15], [16]. In [15], it was demonstrated that from scattered signal returns from DVB-T signals it was possible to extract the structural resonance frequency of the the tower and the blade rotation frequency and its multiples. In [16], it was shown that C- and K-band measurements can reveal the shape (straight or curved) of the turbine blades from the calculated spectrograms of signal returns.

The above examples highlight the needs for improved models for the scattering of electromagnetic signals from wind turbines and that these models have potentially multiple use cases. The structure of the rest of this paper is as follows. In Section II, the analytical equation for the line scatterer model is developed further based on [3] by adding elevation angles into the model. In Section III, the amplitude in the line scatterer equation is discussed in detail and a frequency-dependent model is developed for it based on numerical RCS simulations of a wind turbine. In Section IV, examples of the model applied to simulating realistic passive coherent location (PCL) radar CAF calculations are shown and in Section V conclusions from these modeling improvements are drawn.

II. LINE SCATTERER MODEL

The model is based on [3]. Since we generalize it to include all possible elevation angles, it is necessary to formally develop the equation. The scattered signal from a single point scatterer at distance R_T from a transmitter (Tx) and R_R from a receiver (Rx) is

$$s_{\text{Scatterer}}(t, \mathbf{r}) = A \cdot e^{j(2\pi f_0 t - ck\tau + \Phi_0)}, \quad (1)$$

where $\tau = (R_T + R_R)/c$ is path delay, c is the speed of light, A is amplitude, f_0 is sampling frequency, $k = \frac{2\pi}{\lambda}$ is wavenumber, λ is wave length and Φ_0 is a constant phase. The origin for spherical coordinates is located at the rotor center and a single scatterer's location \mathbf{r} is thus (r, ϕ_S, Θ_S) . The rotational angular frequency ω of the wind turbine changes the inclination of the scatterer ϕ_S as a function of time $\phi_S = \omega t$ measured from the z-axis in the clockwise direction. This differs from the elevation angles ϕ_T and ϕ_R , which are measured from the xy-plane and limited to $[-\pi/2, \pi/2]$. Locations of the transmitter and receiver are $\mathbf{r}_T = (r_T, \phi_T, \Theta_T)$ and $\mathbf{r}_R = (r_R, \phi_R, \Theta_R)$. The geometry is demonstrated for the scatterer and transmitter in Fig. 1 and a corresponding geometry is valid for the receiver as well in the bistatic scenario. We need the azimuth angle differences between the level of the rotor hub and the transmitter and receiver

$$\theta_{\{R,T\}} = \Theta_S - \Theta_{\{R,T\}}, \quad (2)$$

We make the same simplification as was done in [3], namely, we observe that the length of the blades L is insignificant compared to R_T or R_R , but we extend this observation to the

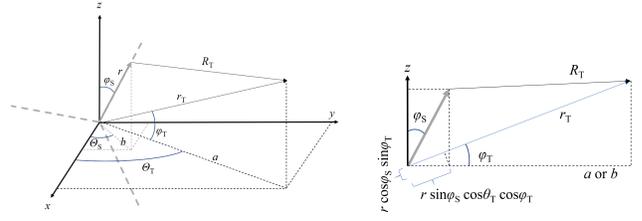


Fig. 1. Coordinate system (left) for a single point scatterer at a distance r from the rotor hub and R_T from the transmitter and a view of a vertical plane (right) from it to illustrate the approximate distance changes. The horizontal line in the latter is a when the horizontal change $r \sin \phi_S \cos \theta_T \cos \phi_T$ is considered and b when the vertical change $r \cos \phi_S \sin \phi_T$ is considered. The vector \mathbf{r} is first projected into a or b and then into r_T or perpendicular to it. An equivalent geometry will also apply for the point scatterer and the receiver, but the corresponding distances are not shown here for clarity. The left-hand panel is inspired by Fig. 1 in [3] with updated notation and coordinates.

z-direction as well. We thus observe the change in distance in horizontal and vertical directions separately for given elevations of the transmitter and receiver. This observation is demonstrated in Fig. 1 and the equations for the distances are

$$R_T = \|\mathbf{r} - \mathbf{r}_T\| \approx r_T - r \sin \phi_S \cos \theta_T \cos \phi_T - r \cos \phi_S \sin \phi_T, \quad (3)$$

$$R_R = \|\mathbf{r} - \mathbf{r}_R\| \approx r_R - r \sin \phi_S \cos \theta_R \cos \phi_R - r \cos \phi_S \sin \phi_R. \quad (4)$$

At this point we note that this approximation gives the exact distance difference when the blade is pointing directly in the direction of Tx/Rx or away from it and $\theta_R = \theta_T = 0$. In the first situation $\phi_T = \phi_R = \pi/2 - \phi_S$ and thus $\sin(\phi_T) = \sin(\pi/2 - \phi_S) = \cos(\phi_S)$ and the same holds for cosines. This means that

$$\|\mathbf{r} - \mathbf{r}_T\| \approx r_T - r \sin^2 \phi_S - r \cos^2 \phi_S = r_T - r. \quad (5)$$

When the blade is pointing in the opposite direction, the result would likewise be the exact $r_T + r$. In the earlier model of [3], this exact distance would only occur when the blade is parallel to zero elevation. Combining the two approximations (3) and (4) to (1) gives

$$s_{\text{Scatterer}}(t, \mathbf{r}) = A \cdot \exp \left(j \left[2\pi f_0 t + kr \sin(\omega t) (\cos \theta_R \cos \phi_R + \cos \theta_T \cos \phi_T) + kr \cos(\omega t) (\sin \phi_R + \sin \phi_T) - k(r_R + r_T) + \Phi_0 \right] \right). \quad (6)$$

Now $\Phi_1 = -k(r_R + r_T) + \Phi_0$ and then we integrate over the length of the blade

$$s_{\text{Blade}}(t) = \int_0^L s_{\text{Scatterer}}(t, \mathbf{r}) dr = A \cdot e^{j(2\pi f_0 t + \Phi_1)} \cdot \left[\left(e^{jkL (\sin(\omega t) (\cos \theta_R \cos \phi_R + \cos \theta_T \cos \phi_T) + \cos(\omega t) (\sin \phi_R + \sin \phi_T))} - 1 \right) / \left(jk (\sin(\omega t) (\cos \theta_R \cos \phi_R + \cos \theta_T \cos \phi_T) + \cos(\omega t) (\sin \phi_R + \sin \phi_T)) \right) \right]. \quad (7)$$



Fig. 2. A 3D model adapted from [17] and used for RCS simulations and studying the maximum, mean and median values of blade flashes. The blade lengths were scaled to 85 m.

By observing the limit $\lim_{a \rightarrow 0} \frac{1}{jab} (e^{jabL} - 1) = L$, we see that the result inside the brackets is L when the denominator is zero. The final result is a sum of N equally spaced blades

$$s_{\text{Rotor}}(t) = A \cdot e^{j(2\pi f_0 t + \Phi_1)} \cdot \sum_{n=1}^N \left[\left(\exp \left(jkL \left(\sin(\omega t + \frac{2\pi n}{N}) (\cos \theta_R \cos \phi_R + \cos \theta_T \cos \phi_T) + \cos(\omega t + \frac{2\pi n}{N}) (\sin \phi_R + \sin \phi_T) \right) \right) - 1 \right) / \left(jk \left(\sin(\omega t + \frac{2\pi n}{N}) (\cos \theta_R \cos \phi_R + \cos \theta_T \cos \phi_T) + \cos(\omega t + \frac{2\pi n}{N}) (\sin \phi_R + \sin \phi_T) \right) \right) \right], \quad (8)$$

where $N = 3$ for typical wind turbines.

The spectrum of the signal is a box function. Increasing the rotor frequency flattens the spectrum, i.e., makes it lower in amplitudes and wider in frequencies.

We can also note some special values of the model. Inserting $\phi_T = 0$ and $\phi_R = 0$ removes all the terms related to elevations and returns the simpler model where everything is located on the same plane. With $\phi_T = \pm\pi/2$ and $\phi_R = \pm\pi/2$, the Tx and Rx are located directly above or below the wind turbine on the same vertical line and all the terms related to azimuth angles disappear.

III. AMPLITUDE OF THE LINE SCATTERER

The amplitude A describes the amplitude of the scattered signal per length of the blade. Because the maximum absolute value inside the brackets in (7) can be L , the maximum absolute value of (7) is AL . The RCS of a blade is proportional to the square of (7) and its maximum is thus $(AL)^2$. From these considerations we also see that A is unitless. For realistic simulations, we need a good way to estimate A , since the mathematics of the model developed so far do not give any clues to its desired behavior. To this end, we performed RCS simulations for a 3D model of a wind turbine shown in Fig. 2 using a method described in [18]. The simulations were

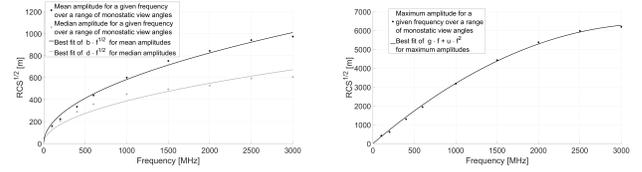


Fig. 3. Mean and median (left) and maximum (right) values of the scattered amplitude for a 85 m long blade over monostatic azimuth view angles $0^\circ, 10^\circ, \dots, 60^\circ$ and zero elevation and best fits of the function $bf^{1/2}$ and $df^{1/2}$ (left) or $gf + uf^2$ (right) to those values.

performed with the assumption that the blades are perfectly conducting. Of course, this is not completely true, since the blades are constructed from composite materials, which typically contain a lot of fiberglass, for example. Simulations between fiberglass and perfectly conducting blades were compared in [19] and it was found that the differences in absolute RCS values were negligible.

The estimated RCS values from our simulations showed, that the mean and median values of blade flash maximum amplitudes across a range of different azimuth view angles and zero elevation increased roughly as a square root of frequency. These mean and median values and their square root fits are shown in Fig. 3. R^2 value for the mean value fit is 0.991 and for the median values it is 0.925. The maximum values of the same range of view angles on the other hand increased almost linearly for most of the frequency interval and this is also shown in Fig. 3. R^2 for the fit of the maximum values of the form $cf + df^2$ is 0.999 and adjusted R^2 is 0.998. Since the actual shape of the blade is mostly like a cone plus a flat section, it is not a surprise that the RCS depends on frequency and view angle. The maximum values were all attained from a single view angle (50°) when the flat part of the blade was roughly parallel to the transmitted and received beam during the top flash. Note that the goal was to find simple analytical formulas which fit the data well and hopefully correspond to some known RCS formulas of some simple shapes. For this reason, we required that the fits must converge to 0 at 0 MHz. We draw two conclusions from these observations.

- 1) The mean or median behavior of the RCS over different view angles can be approximated with a cylinder model.
- 2) The maximum possible RCS occur at very specific view angles but can be huge. These can be approximated with a flat plate model at least up to 1 GHz.

We also note that increasing main flashes with frequency can also be seen in the simulations in [19], even though it was not the main focus in that study and most figures also contain the effect of the tower. For passive radar applications the tower is not such a big concern, because its effect is removed in the typical signal processing when the zero-Doppler is filtered.

Fig. 3 tells us that the mean and median monostatic RCS values of the blade flashes increase roughly linearly with frequency. A suitable simple model for this frequency dependency is thus a cylinder, for which the physical optics monostatic RCS approximation when viewed head on from

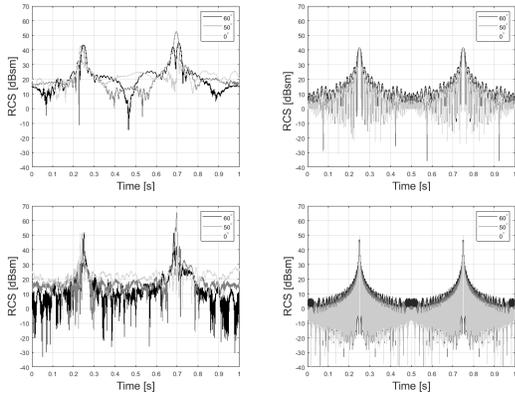


Fig. 4. Scattered RCS time series at the frequencies 100 MHz (top row) and 600 MHz (bottom row) for a wind turbine with 85 m long blades of monostatic view angles 0° , 50° and 60° calculated using the RCS simulation results (left column) and the line scatterer model (right column).

the broadside [20, Ch. 5.3] is

$$\text{RCS}_{\text{cylinder}} = \frac{\pi D L^2}{\lambda} = \frac{\pi D L^2 f}{c}, \quad (9)$$

and then the amplitude per length of blade is

$$A_{\text{cylinder}} = \frac{\text{RCS}_{\text{cylinder}}^{1/2}}{L} = \left(\frac{\pi D}{\lambda}\right)^{1/2} = \left(\frac{\pi D f}{c}\right)^{1/2}. \quad (10)$$

Using the fitted curve value $b = 0.018442 \text{ s}^{1/2}\text{m}$ from Fig. 3, we can solve what is the width of the modelled cylinder which causes the mean RCS behavior when the blade length is $L = 85 \text{ m}$

$$D = \frac{(0.018442 \text{ s}^{1/2}\text{m}/85 \text{ m})^2 \cdot 299792458 \text{ m/s}}{\pi} = 4.492 \text{ m}, \quad (11)$$

and using the fitted value of $d = 0.012267 \text{ s}^{1/2}\text{m}$ from Fig. 3, the median behavior is achieved by a cylinder of width $D = 1.988 \text{ m}$.

The physical optics monostatic RCS approximation of a flat plate of length L and width W when viewed head on [20, Ch. 5.3] is

$$\text{RCS}_{\text{plate}} = \frac{4\pi W^2 L^2}{\lambda^2} = \frac{4\pi W^2 L^2 f^2}{c^2}, \quad (12)$$

and the amplitude per length of blade would be

$$A_{\text{plate}} = \frac{\text{RCS}_{\text{plate}}^{1/2}}{L} = \frac{2\pi^{1/2} W}{\lambda} = \frac{2\pi^{1/2} W f}{c}. \quad (13)$$

Using the fitted curve value g for the linear part of the maximum value fit from Fig. 3, the width of the modelled plate causing the maximum RCS behavior when the blade length is $L = 85 \text{ m}$

$$W = \frac{(3.7469 \cdot 10^{-6} \text{ sm}/85 \text{ m}) \cdot 299792458 \text{ m/s}}{2\pi^{1/2}} = 3.728 \text{ m}. \quad (14)$$

These are all reasonable widths compared to the blade length of 85 m. The maximum width of the blades in the 3D model of Fig. 2 is about 11.6 m and the width of the cone's widest part (at the hub connection) is 5.8 m.

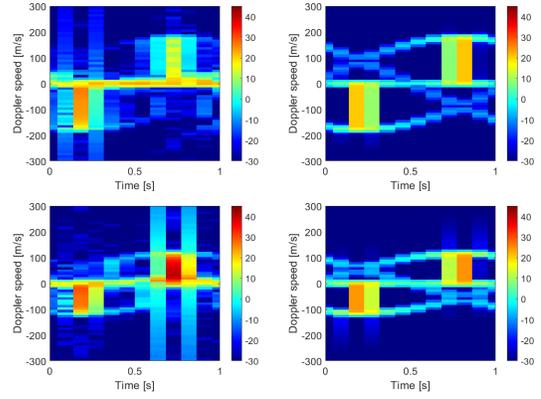


Fig. 5. Short time Fourier transforms of the scattered signals at 100 MHz and monostatic view angles 0° (top row) and 50° (bottom row) and using RCS simulations (left column) and the line scatterer model (right column). Hamming window length is 512 samples and sampling frequency is 3333 Hz.

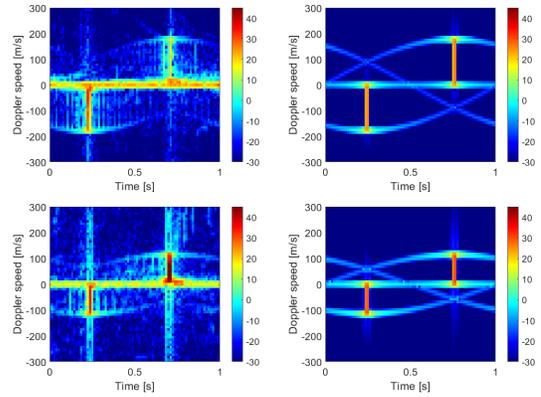


Fig. 6. Short time Fourier transforms of the scattered signals at 600 MHz and monostatic view angles 0° (top row) and 50° (bottom row) and using RCS simulations (left column) and the line scatterer model (right column). Hamming window length is 512 samples and sampling frequency is 20 kHz.

Figure 4 compare the RCS time series at 100 MHz and 600 MHz and at three monostatic view angles of the RCS simulations and the line scatterer model when the line scatterer amplitude has been scaled using the fit of the median values. The rotor rotates at 20 rpm. Figs. 5 and 6 shows the short time Fourier transforms (STFT) of the corresponding signals at two monostatic view angles using window lengths 0.1536 s and 0.0256 s respectively (giving the same Doppler scale) and 50% overlap. The color scale (from -30 to 45) for each time-Doppler-cell is in dBsm. The line scatterer signals and their STFTs are more regular and less noisy than the RCS simulations, but the average behavior is captured quite well.

IV. CAF SIMULATIONS

First thing to note when applying the line scatterer model to realistic signal models is that the (8) returns the scattered response of a single frequency f . A straightforward way to apply the model to a signal which has some bandwidth around a carrier frequency is to use the center frequency to represent the response of the whole band and give it the full

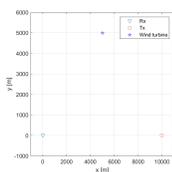


Fig. 7. Geometry for PCL simulation.

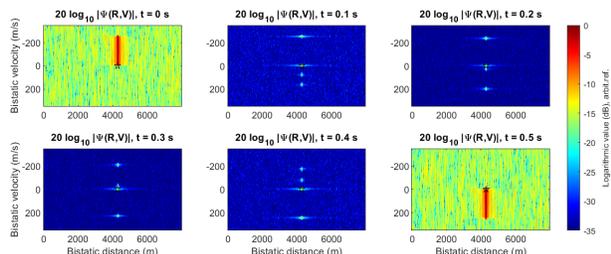


Fig. 8. Time sequence of CAF matrices with a wind turbine reflection.

transmit power. A more sophisticated method would compute (8) for every discrete frequency separately with their share of the transmit power and add the results. The first method is demonstrated in this study.

The PCL simulation geometry was as follows. The receiver was at the origin (0,0,0) km, transmitter at (10,0,0) km, and the three blade wind turbine (marked by a \star in Figs. 7 and 8) at (5,5,1) km (xyz)-coordinates. The wind turbine blades are aligned in the x-direction. The bistatic radar equation [11, Ch. 2.3] was used to calculate the received signal gain without other sources of signal attenuation. The signals were noiseless and no direct path interference or clutter was added. The wind turbine speed was set to 20 rpm (0.33 Hz) and blade length was 85 m. Tx signal's I/Q components were WGN. The transmission power was 30 kW with center frequency of $f_c = 600$ MHz. The baseband's bandwidth was 10 MHz. The CAF matrices were obtained with $T = 0.1$ s integration time [11, p. 132]. Fig. 8 shows the CAF matrices with their corresponding start time of the integration period. The first CAF displays the flash of the approaching blade, while the last CAF shows the flash of the adjacent receding blade. When the main flash of the blade occurs, a vertical line is visible with sidelobes in the CAF and also a noticeable noise level increase.

V. CONCLUSIONS

A simple and fast line scatterer model for bistatic wind turbine scattering was developed further in this paper to include all possible elevation angles. Numerical RCS simulations were also used to develop a frequency dependent model for realistically scaling the amplitude of the line scatterer. It was observed that the mean and median RCS values over many view angles increase roughly linearly in frequency and thus a cylinder model was found suitable for describing it.

The line scatterer model was applied in a passive radar signal processing simulation. The simulated CAFs demonstrate that the blade flashes occur as dynamically changing clutter.

Unfiltered main flashes may cause significant sidelobes and increase the noise level in the CAF.

The line scatterer model can be applied, for example, to quickly estimate the performance of different radar systems in various different scenarios and geometries with wind turbines. The passive radar signal processing simulation can be used in the development and testing of different adaptive digital clutter removal algorithms, for example.

REFERENCES

- [1] R. Mangain, A. Rathi, D. Deb, R. Jain, and D. Seshagiri, "Wind turbine clutter spectrum analysis of radar data and mitigation at tracker for airborne surveillance radar," in Proc. 2018 IEEE MTT-S IMaRC, Kolkata, India, 28–30 November 2018.
- [2] V.C. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: phenomenon, model, and simulation study," IEEE Trans. Aerosp. Electron. Syst., vol. 42, no. 1 January 2006.
- [3] M. Ummerhofer, "Validation of wind turbine Doppler signatures in a passive bistatic radar with a point scatterer model," 2019 ACES, Miami, FL, USA, 14–19 April 2019.
- [4] O.A. Krasnov, and A.G. Yarovoy, "Radar micro-Doppler of wind-turbines simulation and analysis using slowly rotating linear wired constructions," 2014 11th EuRAD, Rome, Italy, 08–10 Oct. 2014.
- [5] C. Dai, L. Huang, B. Tang, G. Liu, and H. Xie, "Simulation and characteristic analysis of wind turbine radar echo based on improved multipath model," 2020 IEEE ICHVE, Beijing, China, 06–10 Sept. 2020.
- [6] F. Fioranelli, M. Ritchie, H. Griffiths, and A. Balleri, "Analysis of multiband monostatic and bistatic radar signatures of wind turbines," 2015 IEEE RadarConf, Johannesburg, South Africa, 27–30 Oct. 2015.
- [7] H. Kuschel, J. Heckenbach, J. Schell, and W. Wissmann, "Effects of wind power plants on passive radar operation," RADAR 2009, Bordeaux, France, 12–16 Oct. 2009.
- [8] F. Kong, Y. Zhang, R. Palmer, and Y. Bai, "Wind turbine radar signature characterization by laboratory measurements," 2011 IEEE RadarCon (RADAR), pp. 162–166, 2011.
- [9] F. Uysal, I. Selesnick, and B.M. Isom, "Mitigation of wind turbine clutter for weather radar by signal separation," IEEE Trans. Geosci. Remote Sens., vol. 54, no. 5, May 2016.
- [10] B. Gallardo-Hernando, F. Pérez-Martínez, and F. Aguado-Encabo, "Wind turbine clutter," chapter 5 in Radar Technology, Guy Kouemou, Ed., IntechOpen, 2010.
- [11] M. Malanowski, Signal Processing for Passive Bistatic Radar, Norwood, MA: Artech House, 2019, ISBN 13: 978-1-63081-662-9.
- [12] T. Pető, and R. Seller, "Adaptive clutter cancellation techniques for passive radars," chapter 6 in Topics in Radar Signal Processing, Graham Weinberg, Ed., IntechOpen, 2018.
- [13] C. Schwark, and D. Cristallini, "Advanced multipath clutter cancellation in OFDM-based passive radar systems," 2016 IEEE RadarConf, Philadelphia, PA, USA, 02–06 May 2016.
- [14] S. Searle, D. Gustainis, B. Hennessy, and R. Young, "Aspects of delay-Doppler filtering in OFDM passive radar," 2018 RADAR, Brisbane, QLD, Australia, 27–31 Aug. 2018.
- [15] M. Ummerhofer, C. Schwark, C. Kress, S. Mechlers, N. Denecke, A. Rohr, M. Fredebohm, M. Koch, and A. Friedmann, "Practical investigation of using passive radar for structural health monitoring of wind farms," 2018 IEEE EE, 12–14 March 2018.
- [16] J.-M. Muñoz-Ferreras, Z. Peng, Y. Tang, R. Gómez-García, D. Liang, and C. Li, "Short-range Doppler-radar signatures from industrial wind turbines: theory, simulations, and measurements," IEEE Trans. Instrum. Meas., vol. 65, no. 9, pp. 2108–2119, 2016.
- [17] <https://3dwarehouse.sketchup.com/model/cc036208d537d6f98967f3aa7f40c33/Wind-Turbine>
- [18] H. Perälä, M. Väilä, J. Jylhä, and A. Visa, "ESPRESS—On efficient bistatic characterization of radar targets," 2015 IEEE RadarCon, Arlington, VA, USA, 10–15 May. 2015.
- [19] T. Cuong, "Radar cross section (RCS) simulation for wind turbines," Masters Thesis, Monterey, CA: Naval Postgraduate School, 2013.
- [20] E.F. Knott, J.F. Shaeffer, and M.T. Tuley, Radar Cross Section, 2nd ed., Raleigh, NC: SciTech Publishing, Inc, 2004. ISBN: 1-891121-25-1.

DB-Drift: Concept drift aware density-based anomaly detection for maritime trajectories

Amelia Henriksen

Sandia National Laboratories

Albuquerque, United States

aahenri@sandia.gov

Abstract—Anomalies in maritime surveillance operations are often high-risk, and need to be detected quickly from real-world, incoming data sources. Hence it is critical that we develop unsupervised anomaly detection algorithms that both operate on a data stream and adapt to it. Real-world maritime data streams involve multiple, intersecting forms of concept drift, meaning that the underlying data distribution changes over time. We introduce DB-Drift, a novel algorithm for adapting existing density-based unsupervised anomaly detection pipelines to handle gradual and seasonal drift simultaneously.

Index Terms—Concept drift, maritime trajectories, anomaly detection

I. MOTIVATION

Illegal activity plagues the world’s oceans. Illegal, unreported and unregulated (IUU) fishing [1]; dumping and pollution [2]; and poorly managed eco-tourism [3] are a major threat to the environment. Drug trafficking and other forms of smuggling are a constant concern, and maritime piracy and armed robbery menace shipping routes around the world [4]. Perhaps most alarming are terrorist activities in coastal waters: smuggling fighters, weapons and other materials in spite of the best efforts of nations world wide [5]. The high-risk nature of these threats means that identifying anomalous vessel behaviors at sea is critical to national and international security, however the sheer volume of maritime traffic data means that maritime surveillance cannot be executed by domain experts alone. The first step in many maritime surveillance pipelines is unsupervised anomaly detection (UAD) algorithms.

Creating robust UAD algorithms in this sphere is a lively field of research—excellent surveys of recent advances can be found in [6] [7]. In spite of these advances, a distinct maritime component has been left out of many models for UAD at sea: concept drift.

Fundamentally, concept drift describes how the underlying distribution of data changes over time. Capturing these changes is critical for unsupervised anomaly detection algorithms, because UAD is typically executed by creating an

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

underlying “normal” model, and then identifying outliers with respect to this norm. If the data majority changes and the associated normal model does not, outliers captured with respect to this inaccurate normal model are less likely to be truly anomalous—and true anomalies are less likely to be captured. In maritime settings, the underlying distribution of vessels changes constantly in a variety of ways. Hence UAD models in maritime settings must also evolve in a variety of ways. This is a critical gap in the state-of-the-art. In this paper we fill this gap by introducing DB-Drift, a density based UAD algorithm that accounts for both seasonal and gradual drift. This is the first algorithm that addresses multiple forms of concept drift simultaneously for UAD at sea.

II. BACKGROUND

A. Density-based clustering for maritime outlier detection

The goal of this work is to improve existing maritime UAD pipelines by incorporating concept drift. To do this, we target a specific component common to many such pipelines: DBSCAN (Density-based spatial clustering of applications with noise) [8]. DBSCAN is a ubiquitous tool because it can automatically identify outliers, has few hyperparameters, and does not need a pre-set number of clusters (unlike k-means). In the vessel tracking sphere, DBSCAN is typically incorporated in one of two ways. First, clustering individual vessel points to extract standard geographic sea routes (and outliers with respect to those routes) [9] [10]. Second, detecting anomalous vessel trajectories by using DBSCAN’s automatic outlier detection on trajectory feature vectors [11] [12] [13]. We focus on the the latter application of DBSCAN.

B. Maritime Concept Drift

Many different forms of concept drift arise and intermingle in real-world data [14]. Important study [15] demonstrated the impact of gradual and seasonal drifts on vessel traffic data. These two types of drift are the focus of our work.

Gradual drift is the most well-understood form of concept drift. It refers to the small, incremental shifts in the underlying distribution of normal data over time. Seasonal drift refers to distributional behaviors that reoccur in a periodic fashion. Because the ocean is affected by literal meteorological seasons, [15] showed that vessel traffic behavior is also highly seasonal—particularly on a monthly scale.

C. Density-based clustering and concept drift: prior work

Much state-of-the-art development for UAD under concept drift has been benchmarked on cybersecurity applications, which are strongly impacted by gradual concept drift [16]. This has given gradual drift adaption for density-based clustering a solid foundation of work. One of the simplest and most commonly used versions compensates for gradual drift by retraining DBSCAN over a sliding window model. More sophisticated techniques use the damped window model, which re-weights and fades the sample data at a user defined rate. Examples include many incremental streaming models for density-based clustering, such as DenStream [17], D-stream [18], DB-stream [19], and TSF-DBSCAN [20].

However, state-of-the-art is biased toward addressing drifts specific to its driving applications. Thus, there has been little work addressing seasonal drift for density-based clustering UAD. We are aware of one work that supports our preliminary algorithmic adaptation in [21]. This work focuses on atmospheric data and briefly discusses the advantages of assigning a separate clustering model to each 28-day period (a recurrent season) for subsequent years of data. The general idea of assigning each seasonal concept its own model is also described in [22]. Developing UAD algorithms under drift for highly seasonal applications—such as maritime surveillance—is critical to advancing the state of the art.

III. THE DB-DRIFT ALGORITHM

A. Understanding DenStream

We now discuss the mechanics of our proposed algorithm. To understand DB-Drift, we must first understand a critical building block: DenStream [17]. DenStream is particularly valuable in the UAD context because it is able to identify outliers in real-time [23].

DenStream is executed by creating and updating microclusters called p-microclusters (the normal model) and o-microclusters (the outlier model). We denote the sets of p-microclusters and o-microclusters at time t by P_t and O_t respectively. For a given microcluster o , we denote $T_0(o)$ to be the microcluster initialization time. These microclusters evolve over time as samples are added and as the points are reweighted according to the damped window model. The rate at which samples fade is determined by fade factor λ . If the input time for a given sample x is given by $T(x)$ then the weight of sample x at time t is $2^{-\lambda(t-T(x))}$. In the original DenStream algorithm, an offline step for microcluster merging is periodically executed. Because this step is irrelevant to the *online* outlier detection component, we do not include it.

We summarize the overarching procedure for DenStream for outlier detection in Algorithm 1. Details for computing the microcluster radius and weight can be found in [17].

B. DB-Drift

The DB-drift algorithm is summarized in figure 1. In the first step, vessel data points are received from maritime surveillance data and fed as a stream into a trajectory assembler (such as Sandia National Laboratories' Tracktable module

Algorithm 1 DenStream for outlier detection at time t

Parameters: Max radius ϵ , minimum p-microcluster weight $\mu\beta$, pruning stepsize T_p
 $\lambda \leftarrow 1/T_p * \log_2(\beta\mu/(\beta\mu - 1))$

for each sample x s.t. $T(x) = t$ **do** ▷ Merge step
 Find the nearest p-microcluster $p_t^* \in P_t$.
 if radius of $\{p_t^*, x\} \leq \epsilon$ **then**
 Add x to p_t^*
 else
 Report the outlier score as $\min_{p_t^* \in P_t} \|x - c(p_t^*)\|$
 Find the nearest o-microcluster $o_t^* \in O_t$
 if radius of $\{o_t^*, x\} \leq \epsilon$ **then**
 Add x to o_t^*
 if weight $w(o_t^*, t) > \mu\beta$ **then**
 Move o_t^* from O_t to P_t
 else
 Add $\{x\}$ to O_t as a new o-microcluster.

if $t \% T_p == 0$ **then** ▷ Pruning step
 for $p \in P_t$ **do**
 if $w(p, t) \leq \mu\beta$ **then**
 Remove p from P_t

for $o \in O_t$ **do**
 if $w(o, t) \leq \xi(t, T_p, o) = \frac{2^{-\lambda(t-T_0(o)+T_p)-1}}{2^{-\lambda T_p}-1}$ **then**
 Remove o from O_t

[24]) which processes the data points to form a trajectory (see section IV-B).

In step two, we extract n-features (see section IV-C for examples) from the input trajectory and feed them as a sample vector x at time $T(x)$ into our outlier detector. We split our model into two DenStream based outlier-scorers: one model under gradual drift and one model under seasonal drift.

Our critical contribution comes from steps three and four. In step three, we evolve multiple layers of cluster-sets via separate DenStream models which each capture a different kind of drift over time. In the current version of our algorithm, we consider two sets of evolving clusters.

For the gradual model layer, we set a small $T_{p,g}$ (corresponding to fade factor λ_g) so that recent samples are weighted more heavily (and older samples fade more quickly).

The seasonal model layer assigns a separate clustering to each season in our recurrent drift model. In our initial implementation of this drift model—here for actual meteorological seasons—a separate DenStream clustering is specifically assigned to each month of the year, and the outlier score for a new trajectory is reported only with respect to the month to which it belongs. It is important that each month has its own DenStream model since seasons themselves can also evolve over time—albeit at a typically slower rate. Hence we set $T_{p,s}$ (corresponding to fade factor λ_s) to be larger. While this algorithm is designed to take advantage of known seasonal behavior, an ongoing area of work is incorporating season discovery for unknown underlying seasonal trends.

In step four, the combined outlier score for sample x at time

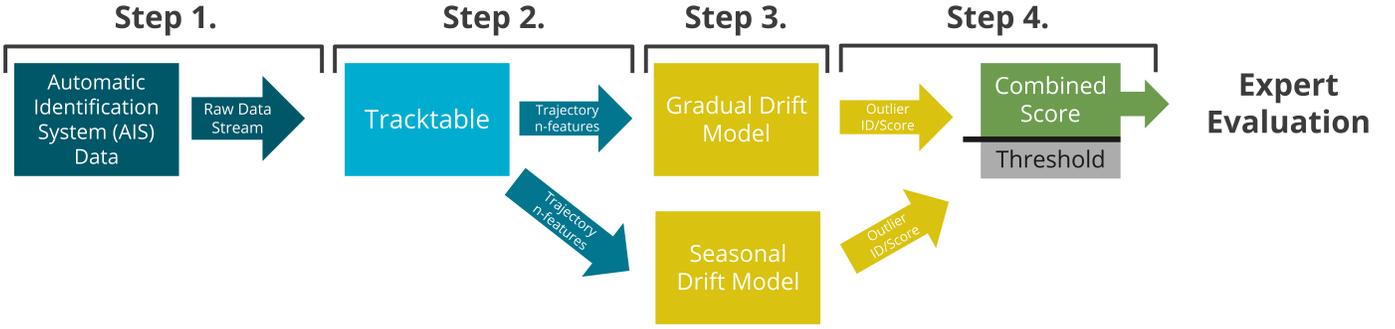


Fig. 1. The structure for the DB-Drift Algorithm.

$t = T(x)$ is given by

$$\hat{O}(x) = w_g \min_{g_t, i \in G_t} \|x - c(g_t, i)\| + w_s \min_{s_t, i \in S_t} \|x_t - c(s_t, i)\| \quad (1)$$

Here w_g and w_s are the outlier-score weights for the gradual and seasonal models. G_t is the set of gradual p-microclusters at time t , S_t is the set of seasonal p-microclusters at time t , and $c(\cdot)$ is the center of a given microcluster. In our examples we set $w_s = 2/3, w_g = 1/3$ because we are interested in seasonal anomalies.

The outlier score is then compared to a threshold θ , s.t. if $\hat{O}(x) > \theta$ then x is included as part of the output anomaly subset, which is passed to an expert for further assessment. This threshold is important, because it allows us to consider the most anomalous points with respect to both seasonal and gradual drift. If an anomaly is captured by only the seasonal model, it must be a strong seasonal anomaly to be included in the anomaly subset. This principle similarly applies to anomalies identified by only the gradual model.

C. Setting hyperparameters

In real-world pipelines, anomaly detection is used to make decisions—for example, deploying law enforcement to intercept illegal activity or developing overarching policy to counter illegal behaviors. Thus real-world UAD pipelines must almost always involve domain expertise. We note that by domain expertise, we refer to not only domain expert researchers but expert algorithms that may be prohibitively expensive to run on large maritime data sets.

This means that the goal of the UAD algorithm is twofold: to find anomalies from data that an expert would otherwise be unable to identify; and to process quantities of data that are too large for domain expert or expert algorithm to parse. The goal for our algorithm is to return an expert-tractable "anomaly subset" that contains the samples of interest. Hyperparameter tuning in this context can be simplified to choosing hyperparameters that deliver a certain size or percentage anomaly subset. It is then the role of feature selection to maximize the number of "true anomalies" captured by the anomaly subset of the desired size.

For this algorithm, we consider these hyperparameters:

1) ϵ_s, ϵ_g : The maximum p-microcluster radius for the gradual and seasonal models respectively. In a true online setting, we would choose these epsilon values using prior historical or burn-in data that yield anomaly subsets of the desired size (for full historical data we tune ϵ_s, ϵ_g post hoc). We then set the ϵ_s, ϵ_g to be slightly smaller than these derived values (leading to a slightly larger percentage of outlier scores > 0), and focus on updating our minimum threshold θ to maintain the appropriate output percentage of outliers.

2) μ, β : The minimum weight for a microcluster to be considered a p-microcluster. Following other implementations of DenStream, we set $\mu = 10, \beta = .25, \mu\beta = 2$ for both the seasonal and gradual models.

3) θ : The minimum threshold for sample x at time $T(x)$ to be considered an outlier (namely $\hat{O}(x) \geq \theta$).

Let r be the desired percentage of the data set to return as an anomalous subset. Let $Q(X, q)$ be the q 'th sample quantile for a set of scalar values X . Select a sample time period $[t_i, t_j]$ to determine θ for incoming points. Denote $n_{t_i, t_j} = |\{x\}_{T(x) \in [t_i, t_j]}|$ to be the number of samples during period $[t_i, t_j]$. Let $\hat{n}_{t_i, t_j} = |\{\hat{O}(x) > 0\}_{T(x) \in [t_i, t_j]}|$ be the number of samples x with outlier score $\hat{O}(x) > 0$ input during period $[t_i, t_j]$. Then we define θ_{t_j}

$$\theta_{i,j} = \begin{cases} Q(\{\hat{O}(x) > 0\}_{T(x) \in [t_i, t_j]}, 1 - q_{i,j}) & q_{i,j} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$q_{i,j} = (n_{t_i, t_j} r) / \hat{n}_{t_i, t_j}$$

This $\theta_{i,j}$ can then be used as a threshold for incoming points until the next update. The length of the periods between θ updates and the amount of historical data used to determine θ are set by the user. For our experiments, we set θ for each incoming month using the samples from the month prior.

4) $T_{p,g}, T_{p,s}$: Pruning periods for the gradual and seasonal models respectively. These are the smallest possible amounts of time before a microcluster can be pruned, and are used to determine the corresponding fade factors λ_g, λ_s for each model (see Algorithm 1). Because we want the gradual model to evolve rapidly, we recommend setting $T_{p,g} = 7$ days. Because we want to retain monthly cluster information from one year to the next, we recommend setting $T_{p,s} = 30$ days.

IV. EXPERIMENTING WITH DB-DRIFT

We now describe the details for experimenting with DB-Drift on real-world data, and present a promising initial result.

A. Automatic Identification System Data

While maritime vessel trajectories can be extracted from many different signals and sources, one of the most common is via the Automatic Identification System (AIS). Ships fitted with AIS consistently broadcast static vessel information (MMSI, vessel type, IMO, length, and width) and dynamic vessel information (time, latitude, longitude, course over ground (COG), speed over ground (SOG), and heading). AIS was designed to prevent collisions at sea, and is publicly broadcast by almost all large vessels. Public research for anomaly detection at sea is frequently performed on AIS data alone or AIS data fused with other data sources.

B. Data Processing

For our experiments, we consider historical AIS data from 2017-2020 off of the coast of Hawaii, compiled from MarineCadastre.gov [25]. We define a bounding box from -161.70558 to -152.98331 longitude and 24.07175 to 18.13869 latitude. UAD models are typically tailored to a particular vessel class. In this example we consider all fishing vessels.

We form trajectories from these AIS points in the following way. First, we group all AIS points by their unique 9 digit maritime identifier (MMSI) and sort them by timestamp. We remove all points associated with non-valid MMSI values and all duplicate points. If the time between two successive points is greater than 30 minutes, the trajectory is split. This trajectory set is then pruned—removing all trajectories with < 5 points or with a convex hull area $< .2$ km (stationary trajectories). After processing, our data set consists of 13,370 fishing trajectories with a total of 20,814,228 AIS points.

C. Feature extraction

A study in [26] analyzes the many geometric and kinematic features that can be used to classify trajectories. We use the following features for our trajectories: sinuosity, number of stops, $k=1$ distance geometry, convex hull perimeter, maximum speed, and medium-low speed proportion. See tables 2 and 3 of [26] for the explicit definitions for these features.

To calculate the number of stops we set the max stop speed $m_s = 1$ knots and the minimum stop segment length to be $m_t = 5$ minutes. All speed values were computed using speeds derived from the location and time-stamp data rather than reported speed over ground data (which is highly prone to noise and errors). Because these features have inherently different scales, they must be standardized (to mean 0 and variance 1) before input.

In this example, features are extracted from trajectories as they are completed (when a subsequent AIS point has not been received for 30 minutes). Depending on real-world user needs and detection scenarios, this pipeline can be adjusted to extract features for incoming trajectory *segments*—where the corresponding trajectories are ongoing.

D. Comparison

1) *Sliding Window DBSCAN*: The goal of this work is to show how maritime UAD pipelines with a DBSCAN or density based clustering component can be improved by incorporating concept drift into that component. In our results, we consider the outlier-detection behavior of DB-Drift verses its commonly used counterpart: sliding window DBSCAN.

Sliding window DBSCAN works as follows: trajectory feature vectors over a given training window size are clustered using static DBSCAN. For a subsequent prediction window, points are labeled outliers if their distance to the closest DBSCAN cluster is $> \epsilon$, where ϵ is the neighborhood radius of the static DBSCAN cluster set. When the end of the prediction window is reached, the training window shifts forward by the length of the prediction window, and the static DBSCAN clustering is retrained. For our experiments we varied the training window sizes at 2 weeks, 3 weeks, 4 weeks, and 8 weeks and kept a constant prediction window of 1 week.

Note that as we increase the size of the sliding window, the memory cost of storing and processing the associated training points increases and can become prohibitive, especially for feature vectors with high dimensionality. Because DB-Drift is built using DenStream, it only needs to store the micro-cluster attributes—centers, radii, and total weights—associated with the gradual and seasonal models respectively. This has significantly lower memory cost.

2) *Choosing Hyperparameters*: We choose the hyperparameters for DB-Drift following section III-C. To update threshold θ we set desired subset percentage $r = 7\%$. We chose ϵ_s, ϵ_g s.t. the models returned around 11% of the data in the anomaly subset. Using our threshold updating technique and combined outlier score, the final "anomaly subset" constituted $\approx 7.5\%$ of the data set. This shows that our threshold technique correctly regulated the output.

To be able to compare sliding window DBSCAN and DB-Drift, we chose DBSCAN ϵ values that returned anomaly subsets of the same size as as DB-Drift (approximately 7.5% of the data). The minimum cluster sample number for sliding window DBSCAN was set to 4.

E. Results

Evaluating anomaly detection algorithms for AIS data is uniquely difficult because there are no publicly available vessel track data sets with labeled anomalies [27] [6]. As such, we have used publicly available reports from the US Coast Guard and various news sources to compile a case-study-based subset of 105 fishing trajectories corresponding to 74 real maritime incidents. This case-study subset does not capture all anomalies, but gives a starting point to analyze the improvements of our algorithm on a form of ground truth. These incidents were primarily derived from US Coast Guard incident investigation reports (IIR) [28], which cover a wide range of incidents at sea. Thus we do not expect every anomaly to be captured by the limited feature set used in our examples—UAD trajectory feature selection based on USCG IIR would be an interesting subject for another paper.

TABLE I
ALGORITHM BEHAVIOR ON REAL-INCIDENT CASE STUDIES

Method	# of real incidents captured	Intersection with DB-Drift
DB-Drift	22	22
DBSCAN 2w	14	13
DBSCAN 3w	17	15
DBSCAN 4w	17	14
DBSCAN 8w	18	14

We obtained the following results in table I. In this experiment, we see that increasing the size of the window for sliding-window DBSCAN allows it to capture more real incidents—albeit at the cost of a longer burn in period and higher memory overheads. DB-drift still captures almost all of the same real-world incidents discovered by sliding window DBSCAN, with lower memory requirements and a burn in period of typically only a few days. We also see that DB-Drift is overall able to capture more real-world incidents than sliding window DBSCAN for all window sizes. This suggests that adding multiple forms of drift can indeed improve density based clustering methods for maritime surveillance.

V. CONCLUSION AND FUTURE WORK

For high risk applications like maritime surveillance, it is critical that UAD pipelines be robust. This means that for data where multiple forms of concept drift are present, multiple forms of concept drift must be addressed. We introduced DB-Drift, the first algorithm to incorporate both gradual and seasonal concept drift into density based clustering for anomaly detection on vessel trajectories. We demonstrated how DB-Drift can be used on real-world data and showed a promising first result. Future work includes demonstrating the behavior of DB-Drift on additional data sets and also demonstrating the behavior of DB-Drift when used in place of DB-SCAN in existing pipelines.

We are also currently investigating ways to incorporate another important form of drift—abrupt drift—into DB-Drift. Abrupt drift describes a sudden shift in the underlying distribution that typically manifests as an outlier itself but is characterized as drift when the shift persists. By adding an explicit abrupt drift detector for our model, DB-Drift would not only be robust to gradual and seasonal drifts but would be able to quickly compensate for sudden shifts in the underlying distribution of the data.

REFERENCES

- [1] G. Hosch, B. Soule, M. Schofield, T. Thomas, C. Kilgour, and T. Huntington, "Any port in a storm: vessel activity and the risk of iuu-caught fish passing through the world's most important fishing ports," *Journal of Ocean and Coastal Economics*, vol. 6, no. 1, p. 1, 2019.
- [2] "Marine pollution: thousands of serious offences exposed in global operation," *Interpol News and Events*, December 2019. [Online]. Available: <https://www.interpol.int/en/News-and-Events/News/2019/Marine-pollution-thousands-of-serious-offences-exposed-in-global-operation>
- [3] J. Almunia, P. Delponti, and F. Rosa, "Using automatic identification system (ais) data to estimate whale watching effort," *Frontiers in Marine Science*, vol. 8, p. 635568, 2021.
- [4] Interpol. (2023) Maritime crimes have a huge economic and human cost. [Online]. Available: <https://www.interpol.int/en/Crimes/Maritime-crime/The-issues>
- [5] B. of Counterterrorism, "Country reports on terrorism 2021," U.S. Department of State, Tech. Rep., 2021.
- [6] M. Riveiro, G. Pallotta, and M. Vespe, "Maritime anomaly detection: A review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 5, p. e1266, 2018.
- [7] K. Wolsing, L. Roepert, J. Bauer, and K. Wehrle, "Anomaly detection in maritime ais tracks: A review of recent approaches," *Journal of Marine Science and Engineering*, vol. 10, no. 1, p. 112, 2022.
- [8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [9] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 6, pp. 2218–2245, 2013.
- [10] M. D. Ferreira, J. N. Campbell, and S. Matwin, "A novel machine learning approach to analyzing geospatial vessel patterns using ais data," *GIScience & Remote Sensing*, vol. 59, no. 1, pp. 1473–1490, 2022.
- [11] T. Zhang, S. Zhao, B. Cheng, and J. Chen, "Atedlw: Intelligent detection of abnormal trajectory in ship data service system," in *2021 IEEE International Conference on Services Computing (SCC)*. IEEE, 2021, pp. 401–406.
- [12] M. D. Rintoul, "MI for trajectories: Bridging the gap from computer to analyst with tracktable." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2020.
- [13] J. Huang, F. Zhu, Z. Huang, J. Wan, and Y. Ren, "Research on real-time anomaly detection of fishing vessels in a marine edge computing environment," *Mobile Information Systems*, vol. 2021, pp. 1–15, 2021.
- [14] V. M. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. Batista, "Challenges in benchmarking stream learning algorithms with real-world data," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1805–1858, 2020.
- [15] E. Osekowska, H. Johnson, and B. Carlsson, "Maritime vessel traffic modeling in the context of concept drift," *Transportation research procedia*, vol. 25, pp. 1457–1476, 2017.
- [16] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms—the numenta anomaly benchmark," in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, 2015, pp. 38–44.
- [17] F. Cao, M. Estert, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proceedings of the 2006 SIAM international conference on data mining*. SIAM, 2006, pp. 328–339.
- [18] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 133–142.
- [19] M. Hahsler and M. Bolaños, "Clustering data streams based on shared density between micro-clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1449–1461, 2016.
- [20] A. Bechini, F. Marcelloni, and A. Renda, "Tsf-dbscan: a novel fuzzy density-based approach for clustering unbounded data streams," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 3, pp. 623–637, 2020.
- [21] R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," *Information Sciences*, vol. 382, pp. 96–114, 2017.
- [22] I. Katakis, G. Tsoumakas, and I. Vlahavas, "An ensemble of classifiers for coping with recurring contexts in data streams," in *ECAI 2008*. IOS Press, 2008, pp. 763–764.
- [23] D. Choudhary, A. Kejariwal, and F. Orsini, "On the runtime-efficacy trade-off of anomaly detection techniques for real-time streaming data," *arXiv preprint arXiv:1710.04735*, 2017.
- [24] A. Wilson, M. Rintoul, B. Newton, K. Ward, P. Baxley, C. Valicka, and P. Schrum, "Tracktable," oct 2020.
- [25] B. of Ocean Energy Management (BOEM), N. Oceanic, and A. A. (NOAA). (2017-2020) Marinecadastre.gov. [Online]. Available: marinecadastre.gov/data
- [26] Y. Tavakoli, L. Peña-Castillo, and A. Soares, "A study on the geometric and kinematic descriptors of trajectories in the classification of ship types," *Sensors*, vol. 22, no. 15, p. 5588, 2022.
- [27] I. Obradović, M. Miličević, and K. Žubričić, "Machine learning approaches to maritime anomaly detection," *Naše more: znanstveni časopis za more i pomorstvo*, vol. 61, no. 5-6, pp. 96–101, 2014.
- [28] U. S. C. Guard, "Uscg maritime information exchange incident investigation reports," United States Coast Guard, Tech. Rep., 2017-2020. [Online]. Available: <https://cgmix.uscg.mil/IIR/Default.aspx>

Author Index

A		L	
Ansila V.M.	36	Lamb, R.A.	66
Abdulaziz, A.	71	Li, Qing.....	56
Altmann, Y.	71	Li, Yi.....	21
Ambat, Sooraj K.	36	Liu, Xingchi	41,61
B		Lyu, Chenyi.....	61
Barbaresco, Frederic	11	M	
Basheer, Bibin	36	Maskell, Simon.....	86
C		McLaughlin, S.	71
Choi, Keun Ha	16	McWhirter, John G.	46
Clemente, Carmine.....	51	Mihaylova, Lyudmila	41,61
Coutts, Fraser K.	76	N	
D		Naqvi, Syed Mohsen.....	21
Di Caterina, G.	66	Narykov, Alexey.....	86
Davies, Mike E.	1,6,71	Nissilä, Juhani.....	91
E		O	
El Fallah Seghrouchni, Amal	11	Orlando, Danilo	51
G		P	
Gan, Runze.....	56	Park, Jihyuk	16
Godsill, Simon.....	56	Park, Ji-Il	16
Goudarzi, Shidrokh	41	Pertilä, Pasi.....	91
H		Proudler, Ian K.	1,6,46,81
Han, Sudan.....	51	R	
Hao, Chengpeng.....	51	Ralph, Jason F.....	86
Henriksen, Amelia.....	96	Ricci, Giuseppe.....	51
Hodgskin-Brown, Richard.....	1	S	
Hopgood, James R.	1,6	Seo, Hyung-Tae.....	16
Humphreys, D.	66	Soleymani, Seyed Ahmad	41,61
J		Sun, Mengwei	1,6
Jo, SeungHyeon	16	Sun, Yang	21
Joseph, K. Shamju.....	26,31	Sunilkumar, Adarsh.....	26,31
Jylhä, Juha.....	91	T	
K		Taylor, Christopher M	86
Kassab, Mohamad.....	11	Thompson, John	76
Khattak, Faizan A.	46,81	V	
Kim, Kyung-Soo.....	16	Väilä, Minna	91
Kirkland, P.	66	Vint, D.	66
Kumar, K. Manoj	26,31		

W

Wang, Wenwu21,41,61
Weiss, Stephan.....46,81
Wright, James S.6

X

Xiao, Pei41

Y

Yan, Linjie51

Z

Zhang, Kaiyu.....76
Zitar, Raed Abu.....11