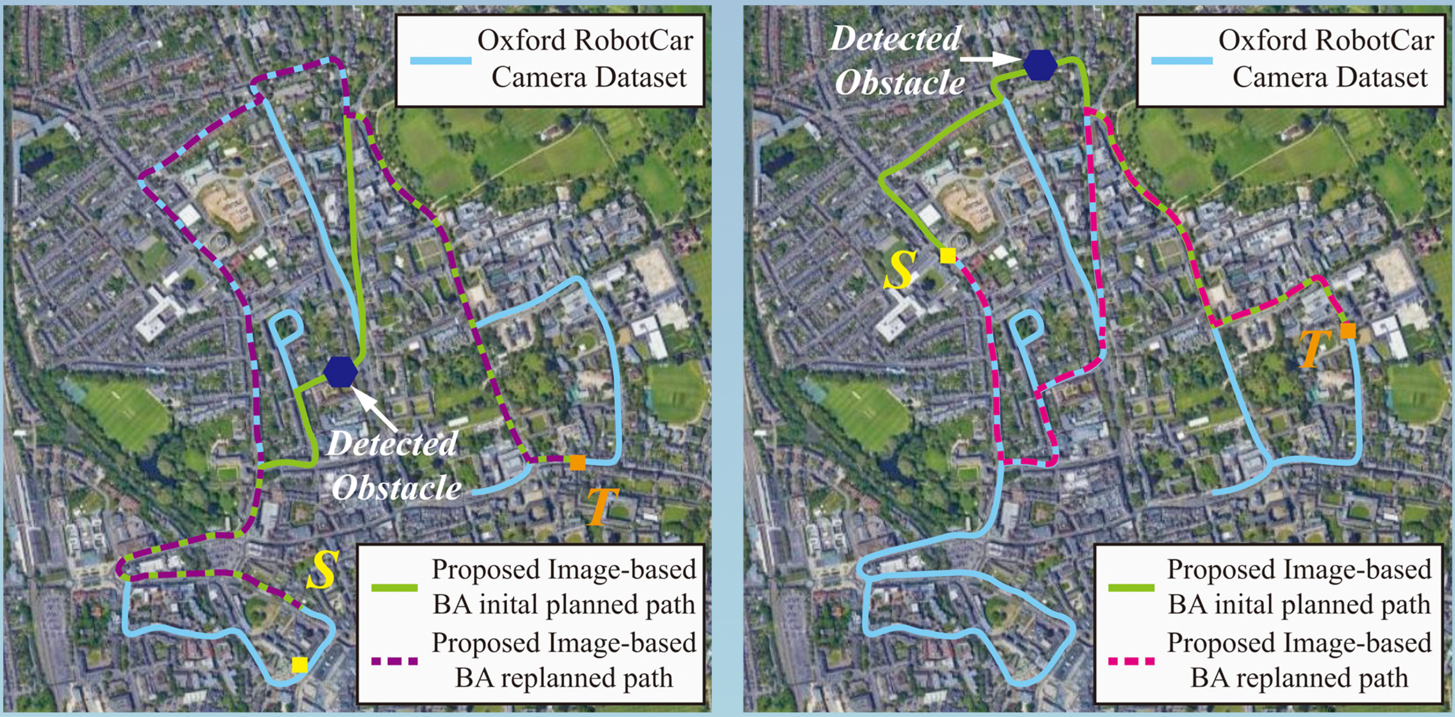
Volume 3, Issue 2, June 2023

INTELLIGENCE & ROBOTICS


Oxford RobotCar Camera Dataset
Detected Obstacle
S
T
Proposed Image-based BA inital planned path
Proposed Image-based BA replanned path
Detected Obstacle
Oxford RobotCar Camera Dataset
S
T
Proposed Image-based BA inital planned path
Proposed Image-based BA replanned path

A bio-inspired algorithm in image-based path planning and localization using visual features and maps

Daniel Short, Tingjun Lei, Chaomin Luo, Daniel W. Carruth, Zhuming Bi

Open Access
ISSN 2770-3541 (Online)

oae
www.intellrobot.com

Editor-in-Chief

Simon X. Yang

Prof. Simon X. Yang is currently the Head of the Advanced Robotics and Intelligent Systems Laboratory at the University of Guelph. His research interests include artificial intelligent, robotics, sensors and multi-sensor fusion, wireless sensor networks, control systems, bio-inspired intelligence, machine learning, neural networks, fuzzy systems, and computational neuroscience.

Our Features

(1) Gold Open Access
(2) Strong Editorial Board
(3) Rigorous Peer-review
(4) Discounted English language Editing Service
(5) Online First Once Accepted
(6) Wide Promotion (Twitter\LinkedIn\WeChat\Facebook)

Editorial Board

- 1 Editor-in-Chief
- 2 Executive Editor
- 6 Advisory Editorial Members
- 49 Associate Editors
- 24 Youth Editorial Board Members

Scope

Top-quality unpublished original technical and non-technical application-focused articles are welcome from intelligence and robotics, particularly on the interdisciplinary areas of intelligence and robotics, including but not limited to the following areas:

- biological, bio-inspired, and artificial intelligence;
- neural networks, fuzzy systems, and evolutionary algorithms;
- sensing, multi-sensor fusion, localization, data analysis, modeling, planning, and control for various mobile, aerial, and underwater robotic systems;
- robot cooperation, teleoperation, and human-machine interactions;
- development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

Journal Home

https://intellrobot.com/

Submission Link

https://oaemesas.com/login?JournalId=ir

EDITORIAL BOARD

EDITORIAL BOARD

GENERAL INFORMATION

About the Journal

Intelligence & Robotics (*IR*), ISSN 2770-3541 (Online), publishes top-quality unpublished original technical and non-technical application-focused articles on intelligence and robotics, particularly on the interdisciplinary areas of intelligence and robotics. The Journal seeks to publish articles that deal with the theory, design, and applications of intelligence and robotics, ranging from software to hardware. The scope of the Journal includes, but is not limited to, biological, bio-inspired, and artificial intelligence; neural networks, fuzzy systems, and evolutionary algorithms; sensing, multi-sensor fusion, localization, data analysis, modeling, planning, and control for various mobile, aerial, and underwater robotic systems; and robot cooperation, teleoperation and human-machine interactions. The Journal would be interested in distributing development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

Information for Authors

Manuscripts should be prepared in accordance with Author Instructions.
Please check https://intellrobot.com/pages/view/author_instructions for details.
All manuscripts should be submitted online at https://oaemesas.com/login?JournalId=ir.

Copyright

Permissions

For information on how to request permissions to reproduce articles/information from this journal, please visit
www.intellrobot.com.

Disclaimer

The information and opinions presented in the journal reflect the views of the authors and not of the journal or its Editorial Board or the Publisher. Publication does not constitute endorsement by the journal. Neither the *IR* nor its publishers nor anyone else involved in creating, producing or delivering the *IR* or the materials contained therein, assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information provided in the *IR*, nor shall they be liable for any direct, indirect, incidental, special, consequential or punitive damages arising out of the use of the *IR*. *IR*, nor its publishers, nor any other party involved in the preparation of material contained in the *IR* represents or warrants that the information contained herein is in every respect accurate or complete, and they are not responsible for any errors or omissions or for the results obtained from the use of such material. Readers are encouraged to confirm the information contained herein with other sources.

Published by

OAE Publishing Inc.
245 E Main Street Ste 107, Alhambra CA 91801, USA
Website: www.oaepublish.com

Contacts

E-mail: editorial@intellrobot.com
Website: www.intellrobot.com

Intelligence & Robotics

CONTENTS

Volume 3, Issue 2

Research Article

113-30 **Operational control with set-points tuning-application to mobile robots**
Xiaomo Yan, Hong Wang
Intell Robot 2023;3(2):113-30. http://dx.doi.org/10.20517/ir.2023.06

131-43 **Semi-supervised joint adaptation transfer network withconditional adversarial learning for rotary machine fault diagnosis**
Chun Liu, Shaojie Li, Hongtian Chen, Xianchao Xiu, Chen Peng
Intell Robot 2023;3(2):131-43. http://dx.doi.org/10.20517/ir.2023.07

Editorial

144-7 **ChatGPT in connected and autonomous vehicles: benefits and challenges**
Lei Lei, Hao Zhang, Simon X. Yang
Intell Robot 2023;3(2):144-7. https://dx.doi.org/10.20517/ir.2023.08

Research Article

148-60 **Application of distributed and decentralized technologies in the management of intelligent transport systems**
Luba Eremina, Anton Mamoiko, Guo Aohua
Intell Robot 2023;3(2):148-60. https://dx.doi.org/10.20517/ir.2023.09

161-75 **Reinforcement learning with parameterized action space and sparse reward for UAV navigation**
Shiying Feng, Xiaofeng Li, Lu Ren, Shuiqing Xu
Intell Robot 2023;3(2):161-75. http://dx.doi.org/10.20517/ir.2023.10

176-89 **Adaptive backstepping control of high-order fully actuated nonlinear systems with event-triggered strategy**
Chengyuan Yan, Jianwei Xia, Xinru Liu, Huarong Yue, Chong Li
Intell Robot 2023;3(2):176-89. http://dx.doi.org/10.20517/ir.2023.11

Review

190-212 **Intelligent flood forecasting and warning: a survey**
Yue Zhang, Daiwei Pan, Jesse Van Griensven, Simon X. Yang, Bahram Gharabaghi
Intell Robot 2023;3(2):190-212. http://dx.doi.org/10.20517/ir.2023.12

Intelligence & Robotics

CONTENTS

Volume 3, Issue 2

Research Article

213-21 **The cooperatability of the first-order multi-agent systems consisting of a leader and a follower with multiplicative noises under Markov switching topologies**
Dianqiang Li, Tao Li
Intell Robot 2023;3(2):213-21. http://dx.doi.org/10.20517/ir.2023.13

222-41 **A bio-inspired algorithm in image-based path planning and localization using visual features and maps**
Daniel Short, Tingjun Lei, Chaomin Luo, Daniel W. Carruth, Zhuming Bi
Intell Robot 2023;3(2):222-41. http://dx.doi.org/10.20517/ir.2023.14

Intelligence & Robotics

Research Article

Open Access

Operational control with set-points tuning - application to mobile robots

Xiaomo Yan[1], **Hong Wang**[2]

[1]The School of Electrical Engineering and Electronics, The University of Manchester, Manchester M6- 1QD, UK.
[2]Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA.

Correspondence to: Prof. Hong Wang, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. E-mail: mikewanguk@yahoo.co.uk; ORCID: 0000-0002-9876-0176

How to cite this article: Yan X, Wang H. Operational control with set-points tuning - application to mobile robots. *Intell Robot* 2023;3(2):113-30. http://dx.doi.org/10.20517/ir.2023.06

Received: 30 Jan 2023 **First Decision:** 7 Mar 2023 **Revised:** 27 Mar 2023 **Accepted:** 4 Apr 2023 **Published:** 21 Apr 2023

Academic Editor: Simon X. Yang **Copy Editor:** Ying Han **Production Editor:** Ying Han

Abstract

This paper proposes a novel method for the optimal tuning of set points for multiple-layered control system structure widely seen in robotics and other complex industrial processes composed of a number of subsystems. The terminal sliding mode control (SMC) is used as the low-level control strategy to ensure the stability of subsystems. When uncertainties exist, it can be shown that the deteriorated system performance will be improved by the outer loop with set points tuning. For this purpose, the learning of the new set point is designed to compensate for the effects caused by uncertainties during the system operation. At the same time, the system is proven to stay with the original set point when the compensation is introduced. A practical application to a holonomic mobile robot system is given to illustrate the presented method. Desired results have been obtained.

Keywords: Set point reselection, mobile robot, terminal sliding mode control, hamilton–jacobi–bellman(HJB) equation

1. INTRODUCTION

Conventional optimal control has focused on improving system performance by optimizing controller parameters. As the development of control theory grows and the demand for industrial automation strengthens, controllers are embedded in industrial equipment universally. However, change to controller parameters becomes more difficult and costly as a result of universal use. Instead of changing controller parameters, system performance can also be improved by reselecting the set point[1–4]. This is particularly true for the multi-layered operational structure of control systems widely seen in robotics and complex industrial processes. Such systems generally consist of two layers - one is the low-level control systems with a number of subsystems, and the other is the operational control layer that generate set points to the subsystems so as to ensure a desired overall system performance. The new set point is designed based on prior knowledge of the system structure. The set point reselection method is achieved by using a cascade structure, which is widely used in complex industrial control strategy design[5–7]. The inherent control loop is considered to be the inner loop, which guarantees the system's stability. Then system's performance will be be improved by the outer loop.

Sliding mode control (SMC) is known for its outstanding robustness[8–15]. By choosing the proper sliding manifold, a sliding mode controller is capable of guaranteeing system convergence. The chattering phenomenon has long been a notable issue in the SMC design[16–18]. The high-frequency components will not only affect the system response but also damage the actuators. To eliminate the chattering phenomenon, smooth functions, such as the saturation function, are usually used in the SMC design[19]. However, when the sliding manifold moves within the scope of a chosen linear portion, the amplifier from the saturation function slows the response. In 2012, Polyakov pointed out a certain type of function[20] that guarantees a fast arrival with both large and small variances. In the conventional SMC design, a fast response relies on a large control gain, which is another incentive for the chattering issue. Instead of using a linear function, nonlinear sliding manifolds give a more flexible response[21–24], but it is necessary to check if singular points exist in the system.

To overcome the above difficulties, in this work we focus on the tuning of the set-points to all the subsystems in the considered multi-layered operational system structure. The idea is to tuning these set points so that the system can still achieve the originally targeted performance when the system is subjected to unexpected uncertainties.

Therefore the novelties and contributions are as follows:

(1) The system considered is of a multi-layered structure that consists of 1) subsystem layers which are low-level control system, and 2) operational layer that generate set points to these subsystems by optimizing a global system performance;

(2) The proposed method aims at achieving optimal tuning of the set-points so as to guarantee the original optimized performance when the system is subjected to unexpected uncertainties rather than ask the subsystems to follow the newly updated set-points;

(3) Application to a mobile robot system has been made - showing encouraging results in terms of performance guarantee.

This paper is organized in the following sections. In Section 2, the discrete time kinematic system is described. The inherent control strategy is terminal SMC. The stability of inner loops is also checked. This description is followed by the augmented representation of the state variables and system uncertainties. An optimal outer loop design is proposed in Section 3. Finally, the method proposed in this paper is tested on a holonomic mobile robot in Section 4 to validate system dynamics and quantitatively compare the system's performance with and without an outer loop. Conclusions are given in section 5.

2. METHODS

Because most of the industrial processes are controlled by computers, discrete system dynamics are considered in the design.

2.1. System model

Consider the system described by the following expression:

$$\begin{cases} \boldsymbol{x}_{1,k+1} = \boldsymbol{x}_{1,k} + h\boldsymbol{x}_{2,k}, \\ \boldsymbol{x}_{2,k+1} = \boldsymbol{x}_{2,k} + h(\boldsymbol{f}_k + \boldsymbol{g}_k \boldsymbol{u}_k), \end{cases} \tag{1}$$

where h is the system sampling interval, $\boldsymbol{x}_{i,k} = \boldsymbol{x}_i(kh)$, $\boldsymbol{x}_k = \begin{pmatrix} \boldsymbol{x}_{1,k} & \boldsymbol{x}_{2,k} \end{pmatrix}^T \in \mathbb{R}^{2n}$ is the system states vector, $\boldsymbol{f}_k \in \mathbb{R}^n$ represents the nonlinear parts of the system, and $\boldsymbol{g}_k \in \mathbb{R}^n$ is an invertible matrix, which represents the relationships between control input $\boldsymbol{u}_k$ and system states $\boldsymbol{x}_k$.

It can be seen that such system expression would represent a wide-range of dynamics systems in practice, and examples are robotic systems and complicated industrial processes such as paper making, mineral processing, chemnical plant and car manufacturing systems.

The errors between system states and a set point vector can be expressed as

$$\boldsymbol{e}_{1,k} = \boldsymbol{x}_{1,k} - \boldsymbol{r}^*_k, \tag{2}$$

where $\boldsymbol{r}^*_k$ is the set point vector. Then,

$$\boldsymbol{e}_{1,k+1} = \boldsymbol{x}_{1,k+1} - \boldsymbol{r}^*_{k+1}. \tag{3}$$

The difference between errors of $k+1$ and k can be expressed as

$$\boldsymbol{e}_{1,k+1} - \boldsymbol{e}_{1,k} = h\boldsymbol{x}_{2,k} - (\boldsymbol{r}^*_{k+1} - \boldsymbol{r}^*_k). \tag{4}$$

In system expression Equation (1), $\boldsymbol{e}_{2,k}$ describes the change from $\boldsymbol{e}_{1,k}$ to $\boldsymbol{e}_{1,k+1}$. Let

$$\boldsymbol{e}_{2,k} = \boldsymbol{x}_{2,k} - \frac{1}{h}(\boldsymbol{r}^*_{k+1} - \boldsymbol{r}^*_k). \tag{5}$$

Then, the error dynamics can be written as

$$\begin{cases} \boldsymbol{e}_{1,k+1} = \boldsymbol{e}_{1,k} + h\boldsymbol{e}_{2,k}, \\ \boldsymbol{e}_{2,k+1} = \boldsymbol{e}_{2,k} + h(\boldsymbol{f}_k + \boldsymbol{g}_k \boldsymbol{u}_k) + \dfrac{1}{h}(-\boldsymbol{r}^*_{k+2} \\ \qquad + 2\boldsymbol{r}^*_{k+1} - \boldsymbol{r}^*_k), \end{cases} \tag{6}$$

which is going to be used in the following control strategy design. In this context, the system in (1) represents the subsystems in a multi-layered system operational structure. The set points grouped in the set point vector is generated by the operational layer as shown in the following figure.

2.2. Terminal sSliding mode controller design

In this subection, we will focus on the control system design for subsystems represented bt equation (1). This requires the selection of control input $\boldsymbol{u}_k$ to ensure the tracking of the system state to the set point. For this purpose, the well-known sliding mode control (SMC) strategy is used to guarantee the stability of the system. Based on the contribution made by Man *et al.* [25], to achieve the terminal convergence, the nonsingular sliding manifold is defined as

$$\boldsymbol{s_k} = \begin{pmatrix} s_{1,k} \\ s_{2,k} \\ \vdots \\ s_{n,k} \end{pmatrix} = \begin{pmatrix} e_{2,1,k} + \beta_1 e_{1,1,k}^{\frac{q_1}{p_1}} \\ e_{2,2,k} + \beta_2 e_{1,2,k}^{\frac{q_2}{p_2}} \\ \vdots \\ e_{2,n,k} + \beta_n e_{1,n,k}^{\frac{q_n}{p_n}} \end{pmatrix}, \tag{7}$$

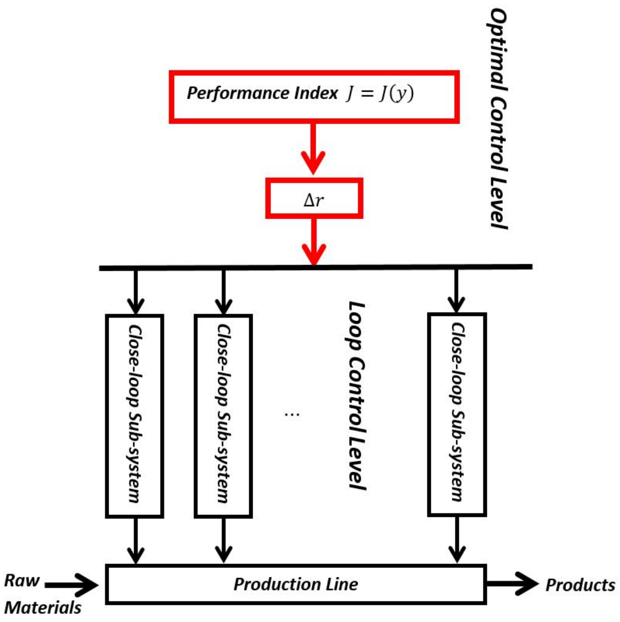


Figure 1. Block diagram of two-layer control structure.

where s_i is the ith sliding surface for the subsystem, $\beta_i \in \mathbb{R}$ is a positive definite, p_i and q_i are positive definite odd integers satisfying $\frac{q_i}{p_i} \in (1, \infty)$, and $e_{i,j,k}$ denotes the kth sample of the jth component in the ith state. The sliding manifold is a constraint to the system states, which drives the system's states along a certain trajectory to an origin point in finite time. System states move along the sliding manifold redproduced

$$s_{i,k+1} = s_{i,k} = 0. \tag{8}$$

Then, by combining Equation (7) and Equation (8) together, the equivalent control law to keep system states on the sliding manifold (7) is given by

$$\begin{aligned} u_{i,k} =& - g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} \\ & + h e_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (r_{i,k+2}^* - 2 r_{i,k+1}^* \\ & + r_{i,k}^*), \end{aligned} \tag{9}$$

where $u_{i,k}$ represents the ith controller component based on the states value at the kth sampling time. However, if a system's initial values (x_0) are not placed on the manifold ($\boldsymbol{s}_k = 0$), the controller shown in Equation (9) cannot transport system states to the sliding manifold. Based on[26], the new controller is written as

$$\begin{aligned} u_{i,k} =& - g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} \\ & + h e_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (r_{i,k+2}^* - 2 r_{i,k+1}^* \\ & + r_{i,k}^*) - g_{i,k}^{-1} \frac{1}{h} (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \tag{10}$$

where $m_{1,i}, m_{2,i}, n_{1,i}, n_{2,i}$ are positively defined odd integers that satisfy $\frac{n_{1,i}}{m_{1,i}} \in (0,1)$ and $\frac{n_{2,i}}{m_{2,i}} \in (1,\infty)$. When the controller described in Equation (10) is implemented into the system, error dynamics change to

$$\begin{cases} e_{1,i,k+1} = e_{1,i,k} + he_{2,i,k}, \\ e_{2,i,k+1} = e_{2,i,k} + \beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} + he_{2,i,k})^{\frac{q_i}{p_i}} \\ \qquad - (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}). \end{cases} \tag{11}$$

To prove the stability of the inner closed-loop system (i.e., the subsystems), the convergence of system states must be analyzed. This is summarized in the following lemma.

Lemma 1 *For the error dynamics shown in Equation* (6), *if the controller is designed as*

$$\begin{aligned} u_{i,k} = & - g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} \\ & + he_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (r_{i,k+2}^* - 2r_{i,k+1}^* \\ & + r_{i,k}^*) - g_{i,k}^{-1} \frac{1}{h} (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \tag{12}$$

where $m_{1,i}, m_{2,i}, n_{1,i}, n_{2,i}$ are positive definite odd integers that satisfy $\frac{n_{1,i}}{m_{1,i}} \in (0,1)$ and $\frac{n_{2,i}}{m_{2,i}} \in (1,\infty)$, then the system state will converge to zero within a finite time.

Proof 1 *By using the sliding manifold described in Equation* (7), *system states go along the sliding manifold (s_i = 0), and when they arrive at it,*

$$s_{i,k+1} = s_{i,k} = 0. \tag{13}$$

However, the system states are not always on the manifold. To make the sliding manifold an attractor, the system states must be ensured to move toward the manifold when the system states are outside the sliding manifold ($s_i \neq 0$), which means that

$$s_{i,k}(s_{i,k+1} - s_{i,k}) < 0. \tag{14}$$

If it can be proved that $(s_{i,k+1} - s_{i,k})$ has an opposite sign to the sliding manifold $(s_{i,k})$, then system stability can be proved. Combining the sliding manifold with Equation (14),

$$\begin{aligned} & s_{i,k}(s_{i,k+1} - s_{i,k}) \\ = & (e_{2,i,k+1} + \beta_i e_{1,i,k+1}^{\frac{q_i}{p_i}} - e_{2,i,k} - \beta_i e_{1,i,k}^{\frac{q_i}{p_i}}) s_{i,k} \\ = & [e_{2,i,k} + \beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} + he_{2,i,k})^{\frac{q_i}{p_i}} \\ & - (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}) + \beta_i (e_{1,i,k} \\ & + he_{2,i,k})^{\frac{q_i}{p_i}} - e_{2,i,k} - \beta_i e_{1,i,k}^{\frac{q_i}{p_i}}] s_{i,k} \\ = & - (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}+1} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}+1}). \end{aligned} \tag{15}$$

As described previously, m_1, m_2, n_1, and n_2 are positive definite odd integers, which guarantees the numerators of $\frac{n_{j,i}}{m_{j,i}} + 1$ are even. An even power,$(n_{j,i} + m_{j,i})$, will erase the sign. $\alpha_{1,i}$ and $\alpha_{2,i}$ are also positive definite real numbers. The derivative of the Lyapunov candidate presented here is less than zero as long as $s_{i,k} \neq 0$; so,

$$s_{i,k}(s_{i,k+1} - s_{i,k}) \leq 0. \tag{16}$$

Zero is the locally stable equilibrium point of the system. Basing on these results, $|s_{i,k+1}| < |s_{i,k}|$ when $|s_{i,k}| \neq 0$, and $s \rightarrow 0$ when $t \rightarrow 0$. Then,

$$e_{2,i,k} = -\beta_i e_{1,i,k}^{\frac{q_1}{p_1}}. \tag{17}$$

The system is globally asymptotically stable.

2.3. Set point reselection method

2.3.1. System model with distribution

The system dynamics with uncertainties are

$$\begin{cases} \boldsymbol{x}_{1,k+1} = \boldsymbol{x}_{1,k} + h\boldsymbol{x}_{2,k} + \boldsymbol{d}_1, \\ \boldsymbol{x}_{2,k+1} = \boldsymbol{x}_{2,k} + h(\boldsymbol{f}_k + \boldsymbol{g}_k\boldsymbol{u}_k) + \boldsymbol{d}_2, \end{cases} \tag{18}$$

where $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are uncertainties in the system. To improve the performance of the system, an additional reselected set point vector $\boldsymbol{\Delta r}_\zeta$ is introduced. The new reference signal is

$$\boldsymbol{r}_k = \boldsymbol{\Delta r}_\zeta + \boldsymbol{r}_k^*, \tag{19}$$

where $\boldsymbol{r}_k^*$ is the origin reference vector that system states want to track, $\boldsymbol{\Delta r}$ is the new compensatory set point to be designed, and $(\bullet)_\zeta$ shows a different sampling rate with the system states. Because of the introduction of a new set point vector, some new vectors need to be defined to emphasize the difference between new states error and the one used in Section 2.

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k} = \boldsymbol{x}_{1,k} - \boldsymbol{r}_k \\ \boldsymbol{\varepsilon}_{2,k} = \boldsymbol{x}_{2,k} - \frac{1}{h}(\boldsymbol{r}_{k+1} - \boldsymbol{r}_k). \end{cases} \tag{20}$$

Based on the new vectors defined above, system dynamics change to

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \boldsymbol{d}_1, \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + h(\boldsymbol{f}_k + \boldsymbol{g}_k\hat{\boldsymbol{u}}_k) + \dfrac{1}{h}(-\boldsymbol{r}_{k+2}, \\ \qquad + 2\boldsymbol{r}_{k+1} - \boldsymbol{r}_k) + \boldsymbol{d}_2, \end{cases} \tag{21}$$

where $\boldsymbol{\varepsilon} = \begin{pmatrix} \boldsymbol{\varepsilon}_1 & \boldsymbol{\varepsilon}_2 \end{pmatrix}^T$ is the new state of the error dynamics. $\hat{u}$ is used to represent the controller based on the new error dynamics described in Equation (26). h is the length of the system states' sampling interval. Based on the error dynamics described in Equation (26), controller $(\hat{u}_{i,k})$ is expressed as

$$\begin{aligned} \hat{u}_{i,k} =& - g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1}\frac{1}{h}[\beta_i \varepsilon_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i(\varepsilon_{1,i,k} \\ & + h\varepsilon_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1}\frac{1}{h^2}(r_{i,k+2}^* - 2r_{i,k+1}^* \\ & + r_{i,k}^*) - g_{i,k}^{-1}\frac{1}{h}(\alpha_{1,i}\hat{s}_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i}\hat{s}_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \tag{22}$$

where $\hat{s}_{i,k} = \varepsilon_{2,i,k} + \beta_i \varepsilon_{1,i,k}^{\frac{q_i}{p_i}}$ is the sliding manifold related to the new system state vector $\boldsymbol{\varepsilon}$. $f_{i,k} = f_i(\boldsymbol{\varepsilon}_k)$ and $\varepsilon_{j,i,k} = \varepsilon_{j,i}(kh)$. Similar to the previous steps, by combining system dynamics described in Equation (27) and controller $\hat{u}_{i,k}$, the system dynamics become

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \boldsymbol{d}_1 \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + \beta\varepsilon_{1,k}^{\frac{q}{p}} - \beta(\varepsilon_{1,k} + h\varepsilon_{2,k})^{\frac{q}{p}} + \boldsymbol{d}_2 \\ \qquad + \dfrac{1}{h}(-\Delta r_{k+2} + \Delta 2r_{k+1} - \Delta r_k). \end{cases} \tag{23}$$

2.4. System performance optimization

This section details a new method of selecting an alternative set point to improve the system performance. In this context, the system model with disturbances will be presented first. This is then followed by the proposed method on the set point tuning that guarantees the achievement of the originally targeted system performance.

2.4.1. System model with distribution

The system dynamics with uncertainties or disturbances can be expressed by adding extra terms in equation (1) as follows.

$$\begin{cases} \boldsymbol{x}_{1,k+1} = \boldsymbol{x}_{1,k} + h\boldsymbol{x}_{2,k} + \boldsymbol{d}_1, \\ \boldsymbol{x}_{2,k+1} = \boldsymbol{x}_{2,k} + h(\boldsymbol{f}_k + \boldsymbol{g}_k\boldsymbol{u}_k) + \boldsymbol{d}_2, \end{cases} \tag{24}$$

where $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are uncertainties or disturbances in the system.

To maintain and improve the performance of the system when the system is subjected to uncertainties and disturbances, we propose to tune the set points rather than changing the control parameters. For this purpose, an additionally reselected set point vector $\boldsymbol{\Delta r}_\zeta$ is introduced. The means that new set point vector should be of the following form

$$\boldsymbol{r}_k = \boldsymbol{\Delta r}_\zeta + \boldsymbol{r}_k^*, \tag{25}$$

where $\boldsymbol{r}_k^*$ is the original reference vector that system states want to track, $\boldsymbol{\Delta r}$ is the new compensatory set point to be designed as an additional term onto the original set point vector, and $(\bullet)_\zeta$ shows a different sampling rate with the system states. Because of the introduction of a new set point vector, some new vectors need to be defined to emphasize the difference between error from new states and the one used in Section 2.

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k} = \boldsymbol{x}_{1,k} - \boldsymbol{r}_k \\ \boldsymbol{\varepsilon}_{2,k} = \boldsymbol{x}_{2,k} - \frac{1}{h}(\boldsymbol{r}_{k+1} - \boldsymbol{r}_k). \end{cases} \tag{26}$$

Based on the new vectors defined previously, system dynamics change to

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \boldsymbol{d}_1, \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + h(\boldsymbol{f}_k + \boldsymbol{g}_k\hat{\boldsymbol{u}}_k) + \dfrac{1}{h}(-\boldsymbol{r}_{k+2}, \\ \qquad\qquad + 2\boldsymbol{r}_{k+1} - \boldsymbol{r}_k) + \boldsymbol{d}_2, \end{cases} \tag{27}$$

where $\boldsymbol{\varepsilon} = \begin{pmatrix} \boldsymbol{\varepsilon}_1 & \boldsymbol{\varepsilon}_2 \end{pmatrix}^T$ is the new state of the error dynamics, $\hat{u}$ is used to represent the controller based on the new error dynamics described in Equation (26), and h is the length of the system states sampling interval. Based on the error dynamics described in Equation (26), controller $(\hat{u}_{i,k})$ is expressed as

$$\begin{aligned} \hat{u}_{i,k} =& - g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1}\frac{1}{h}[\beta_i \varepsilon_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i(\varepsilon_{1,i,k} \\ & + h\varepsilon_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1}\frac{1}{h^2}(r_{i,k+2}^* - 2r_{i,k+1}^* \\ & + r_{i,k}^*) - g_{i,k}^{-1}\frac{1}{h}(\alpha_{1,i}\hat{s}_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i}\hat{s}_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \tag{28}$$

where $\hat{s}_{i,k} = \varepsilon_{2,i,k} + \beta_i \varepsilon_{1,i,k}^{\frac{q_i}{p_i}}$ is the sliding manifold related to the new system state vector $\boldsymbol{\varepsilon}$, $f_{i,k} = f_i(\boldsymbol{\varepsilon}_k)$, and $\varepsilon_{j,i,k} = \varepsilon_{j,i}(kh)$. Similar to the previous steps, by combining system dynamics described in Equation (27) and controller $\hat{u}_{i,k}$, the system dynamics become

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \boldsymbol{d}_1 \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + \beta\varepsilon_{1,k}^{\frac{q}{p}} - \beta(\varepsilon_{1,k} + h\varepsilon_{2,k})^{\frac{q}{p}} + \boldsymbol{d}_2 \\ \qquad\qquad + \dfrac{1}{h}(-\Delta r_{k+2} + \Delta 2r_{k+1} - \Delta r_k). \end{cases} \tag{29}$$

2.4.2. Performance optimization

To eliminate the effect of uncertainties, a new compensatory set point is designed in this section. Since the objective is still to ensure that all the subsystems can track their original set points when the system is subjected to uncertainties or disturbances, the following performance function is used.

$$J = \boldsymbol{e}_{1,N}^T e_{1,N} + \sum_{\zeta=0}^{N-1} \boldsymbol{e}_{1,\zeta}^T \boldsymbol{e}_{1,\zeta}, \tag{30}$$

where N is the bound of sampling times. The performance index previously chosen tries to guarantee the minimum position error between the robot and inherent set point $\boldsymbol{r}^*$. The outer loop is designed based on this performance index, where the idea is to select $\boldsymbol{\Delta r}$ so that J is minimized. Based on the chosen performance index, the Hamiltonian becomes

$$\begin{aligned} H^\zeta =& \boldsymbol{e}_{1,\zeta}^T \boldsymbol{e}_{1,\zeta} + \boldsymbol{\lambda}_{\zeta+1}^T \left({}^{\boldsymbol{\varepsilon}_{1,\zeta} + h\boldsymbol{\varepsilon}_{2,\zeta} + \boldsymbol{d}_1}_{\boldsymbol{\varepsilon}_{2,\zeta} + h(\boldsymbol{f}_\zeta + \boldsymbol{g}_\zeta \hat{\boldsymbol{u}}_\zeta) + \boldsymbol{d}_2} \right) \\ =& \sum_{i=1}^{n} \{ \varepsilon_{1,i,\zeta}^2 + 2\varepsilon_{1,i,\zeta} \Delta r_{i,\zeta} + \Delta^2 r_{i,\zeta} \\ & + \lambda_{1,i,\zeta+1} (\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta} + d_1) \\ & + \lambda_{2,i,\zeta+1} [\varepsilon_{2,i,\zeta} + \beta_i \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}} - \beta_i (\varepsilon_{1,i,\zeta} \\ & + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}} - (\alpha_{1,i} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}}) \\ & - \frac{1}{h} (\Delta r_{i,\zeta+2} - 2\Delta r_{i,\zeta+1} + \Delta r_{i,\zeta}) + d_2] \}, \end{aligned} \tag{31}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{2\times n}$ is the costate.

Remark 1 *Because of the multi-rate method used in the design, more than one sampling period of system states*$((\bullet)_k)$ *may arrive during the sampling period of a reselected set point* $((\bullet)_\zeta)$*. If the reselected set point remains the same during the period considered in the Hamilton equation* $(\Delta r_{i,k+2} = \Delta r_{i,k+1} = \Delta r_{i,k})$*, the reselected set point is unrelated to the system states during this period. In the Hamilton equation,*$\Delta r_{i,\zeta}$ *takes the place of* $\Delta r_{i,k}$ *to make sure the new strategy is sensitive to the change of system state.*

The basic idea of the Hamilton equation is to find the minimum value by using the partial derivative related to the performance index. To get the dynamics of the cost function, the partial derivative of each state is needed.

$$\begin{aligned} \lambda_{1,i,\zeta} =& \frac{\partial H^\zeta}{\partial \varepsilon_{1,i,\zeta}} \\ =& \lambda_{2,i,\zeta+1} [\beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} - \beta_i \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} \\ & + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \\ & + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}) \beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1}] \\ & + 2\varepsilon_{1,i,\zeta} + 2\Delta r_{i,\zeta} + \lambda_{1,i,\zeta+1}, \\ \lambda_{2,i,\zeta} =& \frac{\partial H^\zeta}{\partial \varepsilon_{2,i,\zeta}} \\ =& \lambda_{1,i,\zeta+1} h + \lambda_{2,i,\zeta+1} [1 - \beta_i h \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} \\ & + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i} \frac{n_1}{m_1} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \\ & + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1})]. \end{aligned} \tag{32}$$

To make sure that the Hamilton equation has an extremum, Δr_i needs to satisfy the stationary condition, which is

$$\begin{aligned} 0 &= \frac{\partial H^{\zeta}}{\partial \Delta r_{i,\zeta}} \\ &= 2\varepsilon_{1,i,\zeta} + 2\Delta r_{i,\zeta} - \frac{1}{h}\lambda_{2,i,\zeta+1}. \end{aligned} \tag{33}$$

Owing to the nonlinear controller expression used in this equation, it is necessary to check if singular points exist in the Hamilton equation is necessary. This is described in the following subsection.

2.4.3. Singular problem analysis

Because the component that relates to $\hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}$ appears in costate dynamics and the parameter selections $\frac{n_1}{m_1} \in (0,1)$, the singular problem has to be considered when $\varepsilon_1 = 0$ and when the system states arrive at the surface. To solve the singular problem, a new coefficient is introduced.[27]

$$\mu_{i,\zeta} = \begin{cases} \sin\left(\frac{\pi}{2}\frac{|\hat{s}_{i,\zeta}^{1-\frac{n_{1,i}}{m_{1,i}}}|}{\tau}\right) & \text{if } |\hat{s}_{i,\zeta}^{1-\frac{n_{1,i}}{m_{1,i}}}| \le \tau\ , \\ 1 & \text{otherwise} \end{cases} \tag{34}$$

where τ is the enclosing scope of the sinusoidal function, and $\tau \ll 1$. When $\hat{s}_{i,\zeta} \to 0$, $\mu_{i,\zeta}\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \to 1$. After introducing the new coefficient, the costates dynamics change into

$$\begin{aligned} \lambda_{1,i,\zeta} =& \frac{\partial H^{\zeta}}{\partial \varepsilon_{1,i,\zeta}} \\ =& \lambda_{2,i,\zeta+1}[\beta_i\frac{q_i}{p_i}\varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} - \beta_i\frac{q_i}{p_i}(\varepsilon_{1,i,\zeta} \\ &+ h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i}\frac{n_{1,i}}{m_{1,i}}\mu_{i,\zeta}\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \\ &+ \alpha_{2,i}\frac{n_{2,i}}{m_{2,i}}\hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1})\beta_i\frac{q_i}{p_i}\varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1}] \\ &+ 2\varepsilon_{1,i,\zeta} + 2\Delta r_{i,\zeta} + \lambda_{1,i,\zeta+1}, \\ \lambda_{2,i,\zeta} =& \frac{\partial H^{\zeta}}{\partial \varepsilon_{2,i,\zeta}} \\ =& \lambda_{2,i,\zeta+1}[1 - \beta_i h\frac{q_i}{p_i}(\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} \\ &- (\alpha_{1,i}\frac{n_{1,i}}{m_{1,i}}\mu_{i,\zeta}\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} + \alpha_{2,i}\frac{n_{2,i}}{m_{2,i}}\hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1})] \\ &+ \lambda_{1,i,\zeta+1}h. \end{aligned} \tag{35}$$

The reselected set point is

$$\Delta r_{i,\zeta} = -\varepsilon_{1,i,\zeta} + \frac{1}{2h}\lambda_{2,i,\zeta+1}. \tag{36}$$

The above equation gives the tuning of the set point vector when the uncertainties or disturbances exist. It is also a feedback control for the outer loop of the system shown in Fig. 1. This differs from the most existing methods where the optimization was performed for the tuning of control parameters in the low-level control systems.

Remark 2 *It is possible to avoid the singular problem by choosing some other method to use in the design of the arriving part of the sliding mode controller. The singular problem is considered in this paper to provide a more adaptable solution.*

Theorem 1 *Suppose that the system (6) with the controller shown in Equation (28) experiences uncertainties denoted by $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$. Then, under the reselected set point $\boldsymbol{\Delta r}_\zeta$, the performance index described in Equation (30) will be optimized, and system states will also track the original set point $\boldsymbol{r}^*$.*

Proof 2 *Because the set point update law was introduced, $\boldsymbol{r}_\zeta = \boldsymbol{r}^*_\zeta + \boldsymbol{\Delta r}_\zeta$, where $\boldsymbol{\Delta r}_\zeta$ is shown in Equation (36). With the uncertainties, control inputs change from $\boldsymbol{e}$ to $\boldsymbol{\varepsilon}$. Because the changes are in the set point, the SMC controller can guarantee the stability of the system. If the convergence of $\boldsymbol{\varepsilon}$ to $\boldsymbol{e}$ can be proved, then the system states will track $\boldsymbol{r}^*$.*

By combining Equation (35) and Equation (36), the relationships between costates are

$$\begin{cases}\lambda_{1,i,\zeta+1} =\lambda_{1,i,\zeta} - A\lambda_{2,i,\zeta+1},\\ \lambda_{2,i,\zeta+1} =\lambda_{2,i,\zeta} - \lambda_{1,i,\zeta+1}h + B\lambda_{2,i,\zeta+1},\end{cases} \tag{37}$$

where

$$\begin{aligned}A =&\frac{1}{h} - \beta_i\frac{q_i}{p_i}\varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} + \beta_i\frac{q_i}{p_i}(\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1}\\ &+ (\alpha_{1,i}\frac{n_{1,i}}{m_{1,i}}\mu_{i,\zeta}\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1}\\ &+ \alpha_{2,i}\frac{n_{2,i}}{m_{2,i}}\hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1})\beta_i\frac{q_i}{p_i}\varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1}],\end{aligned} \tag{38}$$

and

$$\begin{aligned}B =&\beta_i h\frac{q_i}{p_i}(\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1}\\ &+ (\alpha_{1,i}\frac{n_{1,i}}{m_{1,i}}\mu_{i,\zeta}\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} + \alpha_{2,i}\frac{n_{2,i}}{m_{2,i}}\hat{s}_{i,\zeta}^{\frac{n_2}{m_2}-1}).\end{aligned} \tag{39}$$

If the system has equilibrium points, then system states will remain at one of the equilibrium points at the end. Assume the system described in Equation (37) has equilibrium points. Based on the properties of equilibrium points,

$$\boldsymbol{\lambda}_{i,\zeta} = \boldsymbol{\lambda}_{i,\zeta+1}. \tag{40}$$

Combining with Equation (37) produces

$$\begin{cases}0 = \boldsymbol{A\lambda}_{2,i,\zeta+1}\\ 0 = -\boldsymbol{\lambda}_{1,i,\zeta+1}h + \boldsymbol{B\lambda}_{2,i,\zeta+1}.\end{cases} \tag{41}$$

If $\boldsymbol{A\lambda}_{2,i,\zeta+1} = 0$ is satisfied, either $\boldsymbol{A}$ or $\boldsymbol{\lambda}_{2,i,\zeta+1}$ should be zero. When $\boldsymbol{A} = 0$ and $\boldsymbol{\varepsilon}_{i,\zeta} \neq 0$, the costate equation has an equilibrium point. However, when the sliding mode controller is applied to the system, $\boldsymbol{\varepsilon}_{i,\zeta} \neq 0$ is not an equilibrium point for the system. According to Equation (38), a constant $\frac{1}{h}$ exists in $\boldsymbol{A}$, which means if $\boldsymbol{A} = 0$ is desired, the system states have to compensate for the constant.

*Based on Equation (37), if $\boldsymbol{\lambda}_{2,\zeta+1} = 0$, the system will stay at $\boldsymbol{\lambda}_{i,\zeta+1} = 0$.
When $\boldsymbol{\lambda} = 0$,*

$$\begin{cases}0 =\boldsymbol{x}_{1,i,\zeta} - \boldsymbol{r}^*_{i,\zeta}\\ 0 =\boldsymbol{x}_{2,i,\zeta} - \dfrac{1}{h}(\boldsymbol{r}^*_{\zeta+1} - \boldsymbol{r}^*_\zeta).\end{cases} \tag{42}$$

This equation shows that $\boldsymbol{x} \rightarrow \boldsymbol{r}^*$.

Remark 3 *The structure in Figure 1 is generic in the sense that many industrial systems have exhibited such a multiple-layered structure[28,29] in their operations, where the top-layer generates the optimized set-points and the lower-layer consists of a number of closed loop tracking systems. This indicates that the method described above can be readily applied to a wide-range of complex systems.*

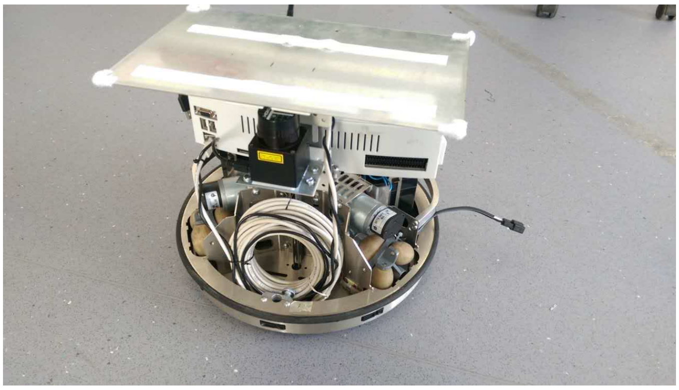


Figure 2. Robotino mobile robot.

The procedures of the algorithm design and implementation in practice is given in the following.

- **Step 1:** Design the paths based upon the destinations for the actuators. The path used in this paper is a direct line. As the hardware structure of Robotino is stated, the motion is factorized in the X and Y direction, whilst with the rotation in the Z direction;
- **Step 2:** Design the controller with the MCU in the actuators as the SMC described in this paper. This is for the actuators to guarantee the stability of the whole motion system;
- **Step 3:** Add/simulate faults in the system. Indeed, faults can be added in any loop such as the position, velocity or current loops. Faults used in this paper are sinusoidal functions added in position and velocity loops.
- **Step 4:** Record the reading from the encoders as the input for optimization;
- **Step 5:** Run the optimal algorithm with the encoder reading to obtain the re-selection of the set-points array, then add the array to the designed path. In this phase the limited computing resource of Robotino has to be considered. The optimal algorithm are therefore executed in a PC instead of the MCU in the Robot;
- **Step 6:** Repeat Step 3 to Step 5 until the system operational performance reaches its expectation.

3. RESULTS

With increasing interest in industrial automation, mobile robots have been playing an important role in transportation, rescue, and other fields. Mobile robots always work in complex situations in which uncertainties are inevitable. Most of those applications rely on accurate information about the robot's location; however, the original controller may not be able to guarantee that accuracy when uncertainties occur.

The Robotino mobile robot in Figure 2 was used in this experiment. The programming environment was Ubuntu Linux. The Robotino is a holonomic mobile robot in 2D. The wheels were driven by direct current (DC) motors. Figure 3 shows the block diagram of the inner closed-loop design of the Robotino. The trajectory planning agency generated the set points based on the requirements. The control signals were passed to the Robotino through its inherent WLAN-Link. The embedded system in the Robotino transferred the velocity setting values into torques. To guarantee the primary stability of the mobile robot, the Robotino had a PI controller inside.

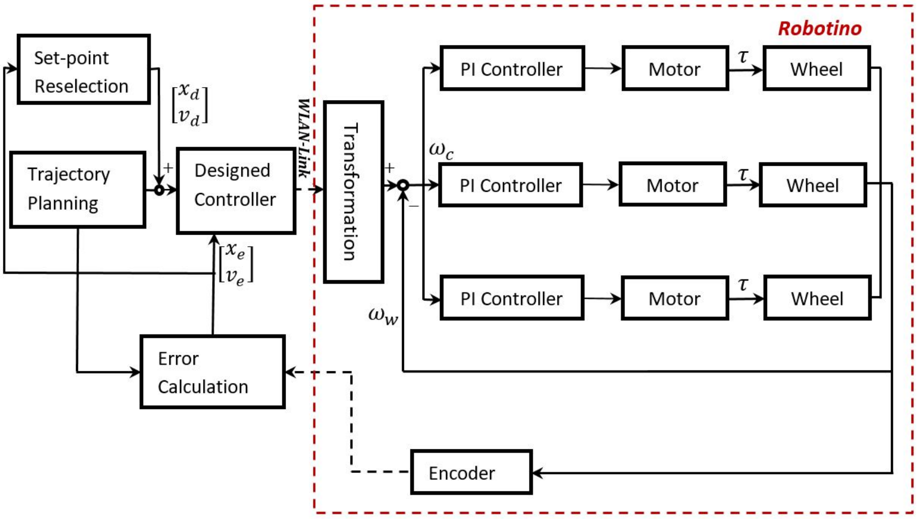


Figure 3. Block diagram of Robotino control.

To demonstrate this method, the sliding manifold picked for x and y was

$$s = v + 2p^{\frac{11}{9}}, \tag{43}$$

where v represents the velocity, and p is the position feedback from encoders. To make sure that the Robotino could be initialled at any position,

$$u_a = -(0.02s^{\frac{9}{11}} + 0.05s^{\frac{11}{9}}). \tag{44}$$

The performance index used in this study is

$$J = \boldsymbol{e}_{1,N}^T \boldsymbol{e}_{1,N} + \sum_{k=0}^{N-1} \boldsymbol{e}_{1,k}^T \boldsymbol{e}_{1,k}. \tag{45}$$

Figure 4 and Figure 5 are system state errors. Encoder faults were added onto the controller inputs. The controller's inputs that were affected by faults are represented by solid curves. The figures also show that when the system worked without the optimal loop, the robot's stability was still guaranteed. According to the hardware structure and programming logic, the optimized control signal may not have been able to follow the system if their sampling frequencies remained the same. The multirate sampling method was used in this experiment. Dotted curves represent the resampled signals. The sampling frequency of the resampled signal was one-tenth of the system's clocking frequency.

Velocities of the mobile in each direction are shown in Figures 6 and 7. The resampled velocities are represented with dotted curves.

Figure 8 and Figure 9 are the comparisons of optimized and original position errors relating to r^* and r. From the figures, the reselected set points increased the feedback error at the beginning, which offered a faster response. Δr also decreased the steady errors, and the reselected set point did not affect the original set point tracking. The mobile robot not only had a faster response but also improved accuracy. Figure 10 is the performance index of the mobile robot. The performance index with the reselected set point was better than the original one.

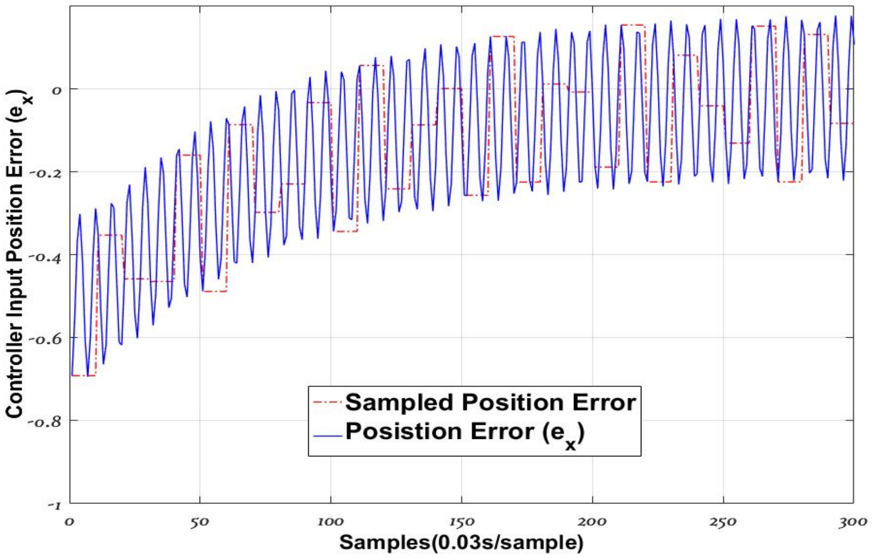


Figure 4. Sampled position error in x direction.

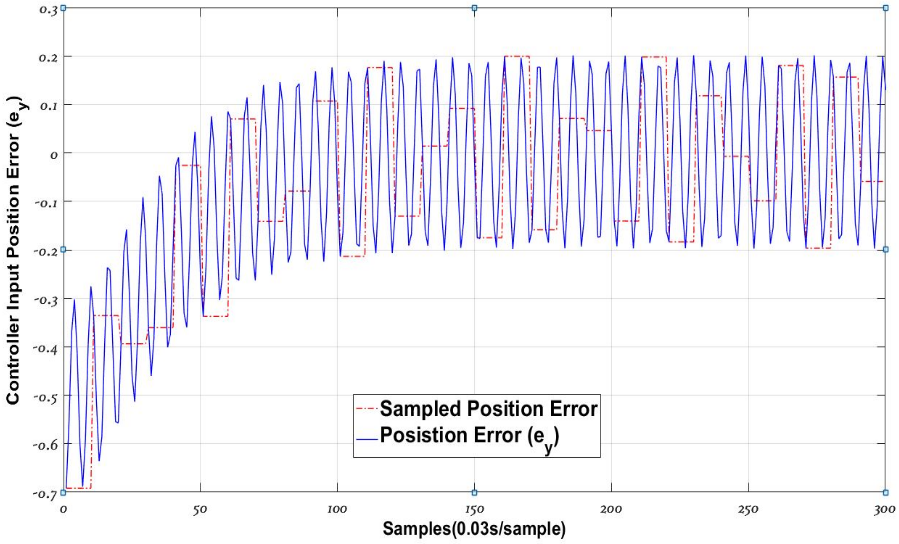


Figure 5. Sampled position error in y direction.

On the other hand, it can be seen that the possible comparison can be made. However, this would mean that we need to compare the results of the set point resection with respect to the case when the set point are not tuned at all regardless of the presence of uncertainties. As the proposed algorithm is an optimization based design. It is believed that the set point tuning would generally produce a better result in terms of minimizing the impact of the uncertainties.

4. CONCLUSIONS

In this paper, a novel set point reselection method was proposed. A sliding mode controller was applied onto the inner loop as the basic controller, which guaranteed the system stability. To eliminate the effects of uncer-

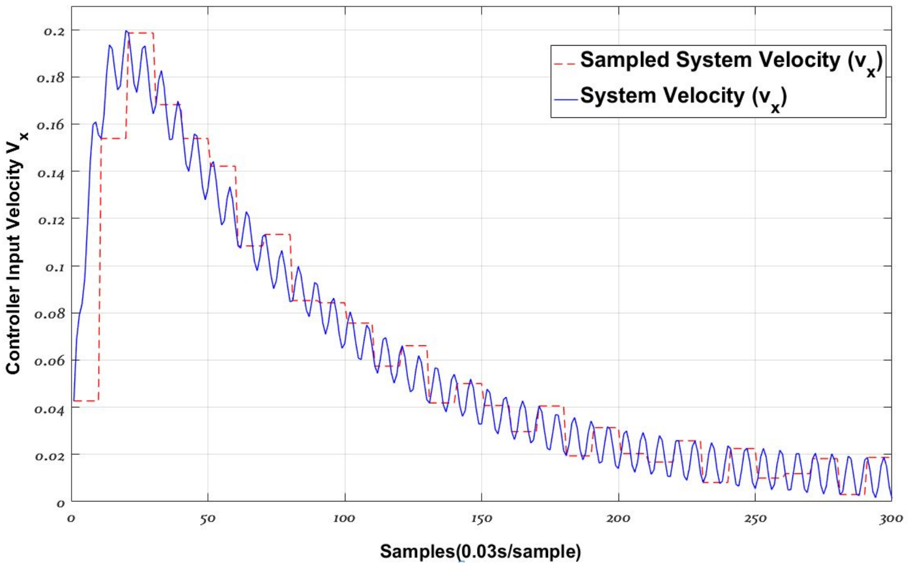


Figure 6. Sampled velocity in x direction.

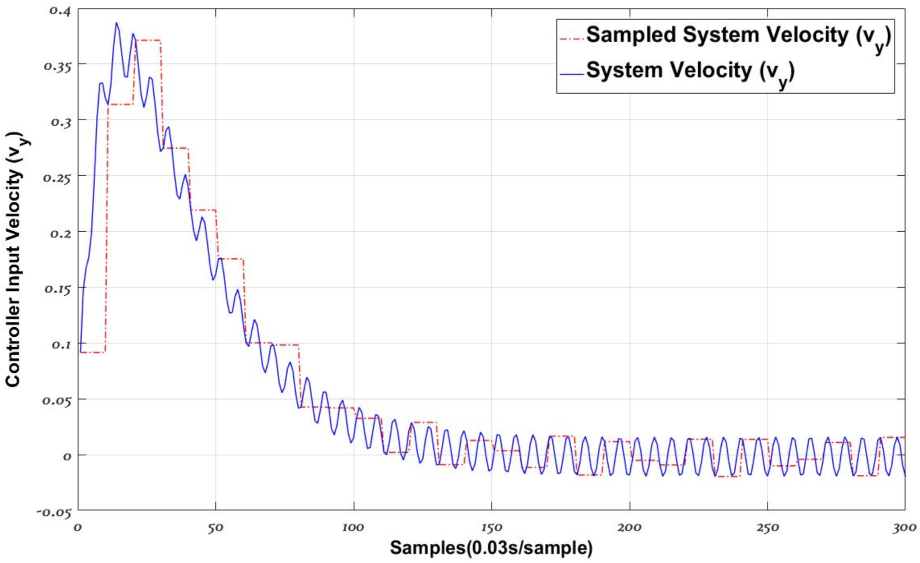


Figure 7. Sampled velocity in y direction.

tainties, the outer loop was designed based on the Hamilton equation used in the optimality principle, which focused on optimizing the chosen system performance index. However, as the nonlinear sliding manifold was introduced, the singular problem existed in the Hamilton equation. To solve the singular problem, an extra sine function was added to the Hamilton equation. The singular problem could also be avoided by carefully choosing the value of the parameters in the sliding manifold. To show a more general result, singular problem was still considered in this paper. Because the new set point may have changed the stability of this system, the convergence to e was proved. The proposed algorithm has been applied to a mobile robot and encouraging results have been obtained.

The system considered in this paper is assumed having known dynamics for f_k and g_k. However, in practice

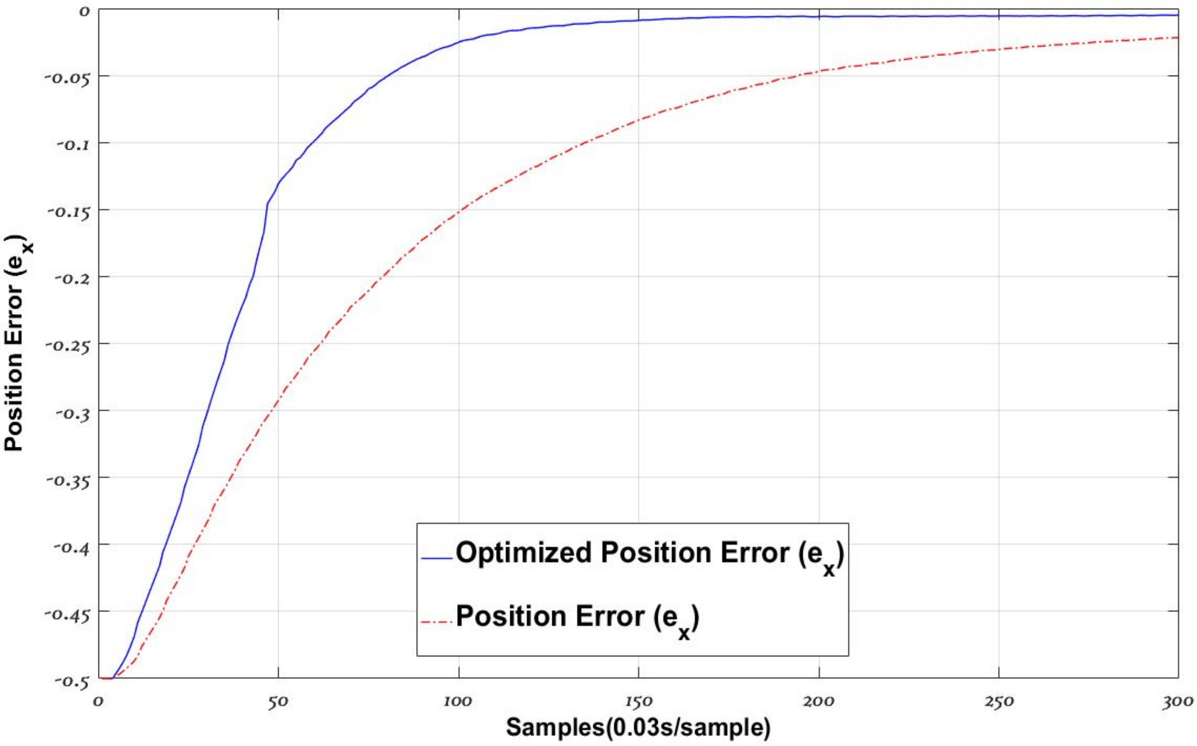


Figure 8. Optimal system output error (e_x).

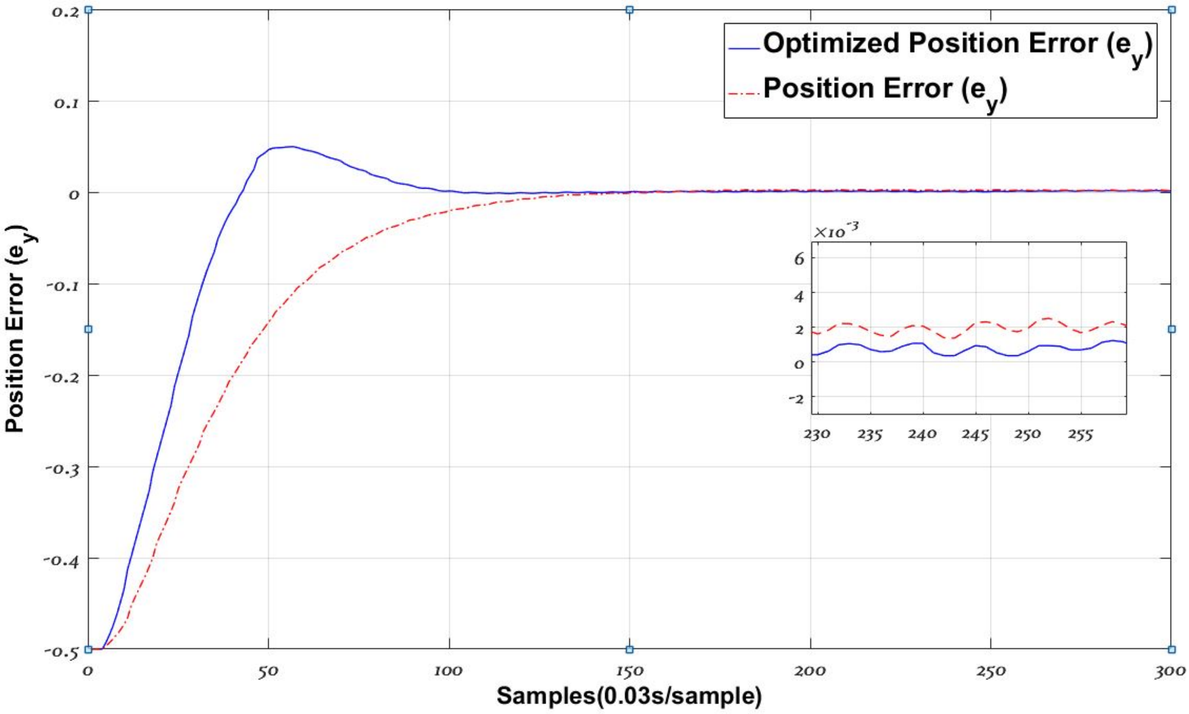


Figure 9. Optimal system output error (e_y).

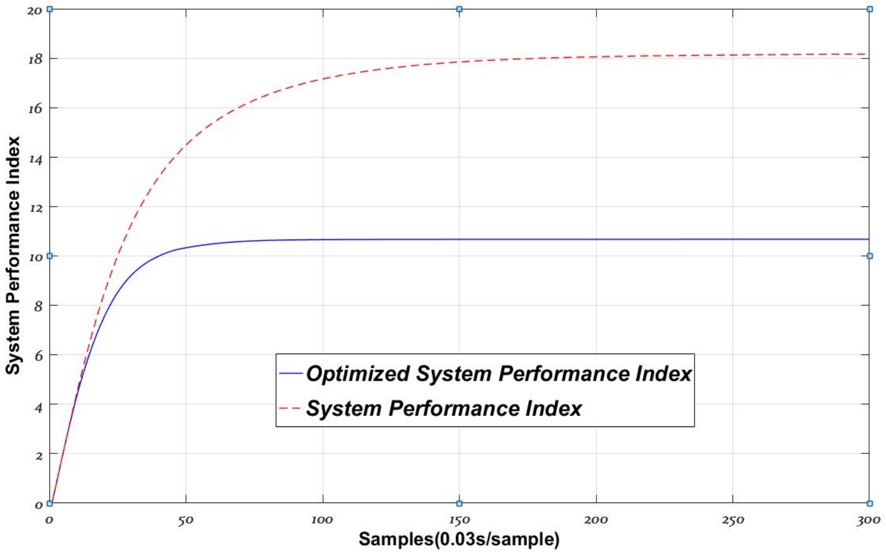


Figure 10. System performance index.

there are many systems of which their dynamics would be subjected to unknown changes in operation environment and other unpredictable factors. As a result, for the future work, we need to consider the adaptive control as well where the model parameters and system dynamics will be learnt using data-driven approaches such as neural networks to estimate dynamics f_k and g_k in equation 1), This would lead to an extra adaptive tuning loop for the closed loop system as shown in Figure 1.

DECLARATIONS

Acknowledgement

The authors would like to thank Dr. Eduard A. Codres and Dr. Mohamed Mustafa from Control Center at the University of Manchester and Dr. Liping Yin from Nanjing University of Information Science and Technology. They provided insight and expertise that greatly assisted this research. The work was performed when Dr. Xiaomo Yan was a PhD student under the supervision of Professor Hong Wang (the second author) between 2013 and 2016. Support from the University of Manchester is gratefully acknowledged.

Authors' contributions

Formulation of detailed algorithm together with testing on mobile robot: Yan X
Main idea on the operational control with set-point tuning: Wang H

Availability of data and materials

The experimental data are not available for open-source usage.

Financial support and sponsorship

The work is partly supported by the University of Manchester UK.

Conflicts of interest

All authors declared that there are no other conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Chai T, Qin SJ, Wang H. Optimal operational control for complex industrial processes. *Annu Rev Control* 2014;38:81-92. DOI
2. Dai W, Chai T, Yang SX. Data-driven optimization control for safety operation of hematite grinding process. *IEEE Trans Ind Electron* 2015;62:2930-41. DOI
3. Wang A, Zhou P, Wang H. Performance analysis for operational optimal control for complex industrial processes under small loop control errors. In: Proceedings of the 2014 international conference on advanced mechatronic systems. IEEE, 2014:159-64. DOI
4. Wu M, Xu C, She J, Yokoyama R. Intelligent integrated optimization and control system for lead–zinc sintering process. *Control Eng Pract* 2009;17:280-90. DOI
5. Zhou P, Chai T, Sun J. Intelligence-based supervisory control for optimal operation of a DCS-controlled grinding system. *IEEE Trans Contr Syst Technol* 2013;21:162-75. DOI
6. Dai W, Zhou P, Zhao D, Lu S, Chai T. Hardware-in-the-loop simulation platform for supervisory control of mineral grinding process. *Powder Tech* 2016;288:422-34. DOI
7. Chai T, Ding J, Wu F. Hybrid intelligent control for optimal operation of shaft furnace roasting process. *Control Eng Pract* 2011;19:264-75. DOI
8. Haq IU, Khan Q, Ullah S, et al. Neural network-based adaptive global sliding mode MPPT controller design for stand-alone photovoltaic systems. *PLoS One* 2022;17:e0260480. DOI
9. Ilyas M, Iqbal J, Ahmad S, Uppal AA, Imtiaz WA, Riaz RA. Hypnosis regulation in propofol anaesthesia employing super-twisting sliding mode control to compensate variability dynamics. *IET Syst Biol* 2020;14:59-67. DOI
10. Anjum MB, Khan Q, Ullah S, et al. Maximum power extraction from a standalone photo voltaic system via neuro-adaptive arbitrary order sliding mode control strategy with high gain differentiation. *Appl Sci* 2022;12:2773. DOI
11. Iqbal J. Modern control laws for an articulated robotic arm: modeling and simulation. *Eng Technol Appl Sci Res* 2019;9:4057-61. DOI
12. Yan X, Zuo Z, Yin L, Wang A, Wang H. Chattering-free sliding mode control for mimo nonlinear manipulator systems based on adaptive neural networks. In: 2015 54th IEEE Conference on Decision and Control (CDC). Dec 2015, pp. 6300–5. DOI
13. Li S, Zhou M, Yu X. Design and implementation of terminal sliding mode control method for PMSM speed regulation system. *IEEE Trans Ind Inf* 2013;9:1879-91. DOI
14. Zhang J, Lin Y, Feng G. Analysis and synthesis of memory-based fuzzy sliding mode controllers. *IEEE Trans Cybern* 2015;45:2880-9. DOI
15. Naouar MW, Monmasson E, Naassani AA, Slama-belkhodja I. FPGA-based dynamic reconfiguration of sliding mode current controllers for synchronous machines. *IEEE Trans Ind Inf* 2013;9:1262-71. DOI
16. Zhang X, Sun L, Zhao K, Sun L. Nonlinear speed control for PMSM system using sliding-mode control and disturbance compensation techniques. *IEEE Trans Power Electron* 2013;28:1358-65. DOI
17. Lee JD, Khoo S, Wang Z. DSP-based sliding-mode control for electromagnetic-levitation precise-position system. *IEEE Trans Ind Inf* 2013;9:817-27. DOI
18. Basin MV, Rodriguez-ramirez PC. Sliding mode controller design for stochastic polynomial systems with unmeasured states. *IEEE Trans Ind Electron* 2014;61:387-96. DOI
19. Khalil HK. Noninear Systems. New Jersey: Prentice-Hall, 1996.
20. Polyakov A. Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Trans Automat Contr* 2012;57:2106-10. DOI
21. Feng Y, Yu X, Han F. High-order terminal sliding-mode observer for parameter estimation of a permanent-magnet synchronous motor. *IEEE Trans Ind Electron* 2013;60:4272-80. DOI
22. Janardhanan S, Bandyopadhyay B. On discretization of continuous-time terminal sliding mode. *IEEE Trans Automat Contr* 2006;51:1532-6. DOI
23. Lin C. Nonsingular terminal sliding mode control of robot manipulators using fuzzy wavelet networks. *IEEE Trans Fuzzy Syst* 2006;14:849-59. DOI
24. Zou AM, Kumar KD, Hou ZG, Liu X. Finite-time attitude tracking control for spacecraft using terminal sliding mode and chebyshev Neural Network. *IEEE Trans Syst Man Cybern B Cybern* 2011;41:950-63. DOI
25. Man Zhihong, Paplinski A, Wu H. A robust MIMO terminal sliding mode control scheme for rigid robotic manipulators. *IEEE Trans Automat Contr* 1994;39:2464-9. DOI
26. Wang H, Wang YJ, Kabore P. Time-varying controller for known nonlinear dynamic systems with guaranteed stability. *Int J Syst Sci* 2002;33:931-8. DOI
27. Zuo Z. Nonsingular fixed-time consensus tracking for second-order multi-agent networks. *Automatica* 2015;54:305-9. DOI
28. Wang A, Wang H, Sheng N, Yin X. Performance analysis for operational optimal control for complex industrial processes — the square

impact principle. In: 2015 21st International Conference on Automation and Computing (ICAC), Glasgow, UK, 2015, pp. 1-6. DOI

29. Yin L, Wang H, Yan X, Zhang H. Disturbance observer-based dynamic optimal setting control. *IET Contr Theo Appl* 2018;12:2423-32. DOI

Liu *et al*. *Intell Robot* 2023;3(2):131-43
DOI: 10.20517/ir.2023.07

Intelligence & Robotics

Research Article

Open Access

Semi-supervised joint adaptation transfer network with conditional adversarial learning for rotary machine fault diagnosis

Chun Liu[1,2], Shaojie Li[2], Hongtian Chen[3], Xianchao Xiu[1], Chen Peng[1]

[1]School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China.
[2]Institute of Artificial Intelligence, Shanghai University, Shanghai 200444, China.
[3]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China.

Correspondence to: Dr. Chun Liu, the School of Mechatronic Engineering and Automation, and also the School of Artificial Intelligence, Shanghai University, Shanghai 200444, China. E-mail: Chun_Liu@shu.edu.cn

How to cite this article: Liu C, Li S, Chen H, Xiu X, Peng C. Fault diagnosis, joint adaptation transfer network, conditional adversarial learning, rotary machine. *Intell Robot* 2023;3(2):7. http://dx.doi.org/10.20517/ir.2023.07

Received: 28 Feb 2023 **First Decision:** 17 Apr 2023 **Revised:** 27 Apr 2023 **Accepted:** 5 May 2023 **Published:** 20 May 2023

Academic Editor: Simon X. Yang **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

At present, artificial intelligence is booming and has made major breakthroughs in fault diagnosis scenarios. However, the high diagnostic accuracy of most mainstream fault diagnosis methods must rely on sufficient data to train the diagnostic models. In addition, there is another assumption that needs to be satisfied: the consistency of training and test data distribution. When these prerequisites are not available, the effectiveness of the diagnosis model declines dramatically. To address this problem, we propose a semi-supervised joint adaptation transfer network with conditional adversarial learning for rotary machine fault diagnosis. To fully utilize the fault features implied in unlabeled data, pseudo-labels are generated through threshold filtering to obtain an initial pre-trained model. Then, a joint domain adaptation transfer network module based on conditional adversarial learning and distance metric is introduced to ensure the consistency of the distribution in two different domains. Lastly, in three groups of experiments with different settings: a single fault with variable load, a single fault with variable speed, and a mixed fault with variable speed and load, it was confirmed that our method can obtain competitive diagnostic performance.

Keywords: Fault diagnosis, joint adaptation transfer network, conditional adversarial learning, rotary machine

1. INTRODUCTION

The intelligent development of modern industrial technology leads to the gradual complexity and systematization of machinery and equipment[1]. As essential equipment in modern industrial applications, rotary machines play a vital role in ensuring efficient and reliable operations. Key components, such as bearings and gears, are critical to the proper functioning of these machines, and any faults can disrupt the normal rotating mechanism. In engineering practice, bearings and gears are prone to faults due to improper assembly, corrosion, overload, poor lubrication, etc[2]. If the equipment fault is not detected in time, it may affect the regular operation of the equipment and cause economic losses. In more serious cases, it may even put the lives of operators at risk. The early detection and prediction of bearing and gear faults in rotary machines will significantly enhance the safety of machinery production and avoid the loss of lives and property caused by mechanical faults. Based on the literature[3,4], fault diagnosis methods for rotary machines are divided into two main categories: traditional fault diagnosis methods that rely on manual signal analysis and newer methods that use neural network diagnostic models to mine fault features.

For the past few years, deep learning techniques have made significant breakthroughs in artificial intelligence fields, and the advantages of automatically learning and extracting valid information from data are gaining increasing attention. By using sensors to acquire vibration signals and other relevant data and processing the data with deep learning algorithms to extract features that correspond to fault data, it becomes feasible to recognize and rectify potential faults[5]. Unlike traditional fault diagnosis methods that use signal processing techniques combined with machine learning classifiers to perform fault diagnosis[6], deep learning-based fault diagnosis models can automatically mine and analyze the underlying mechanisms of faults to obtain accurate fault classification performance with sufficient data[7]. However, in practical engineering scenarios, mechanical equipment mainly operates normally, and failures are relatively rare. Therefore, the amount of fault data collected is usually limited. Furthermore, the distribution of data collected under changing operating conditions, such as speed, load, and surrounding environment of rotary machines, can vary considerably, which may affect the reliability and stability of diagnostic results[8].

Transfer learning is a machine learning technique that allows for the transfer of knowledge learned from one task to another, with the aim of improving the performance of the latter task[9]. In the context of diagnostic tasks, transfer learning allows for the simultaneous application of diagnostic knowledge learned from pre-trained data to relevant diagnostic tasks in order to achieve good diagnostic results[10]. In this strategy, the core problem is distribution alignment, which enables the models to be constrained by the objective function so that it satisfies the assumption of distributional consistency to achieve good diagnostic results[11]. Domain adaptation is the core technique for achieving distribution alignment. It essentially ensures that the feature spaces of the two tasks are aligned through some kind of transformation[12]. In real-world scenarios, the feature space of the source and target tasks can vary greatly, and distance metric minimization is often utilized for alignment. Metrics for differences in distribution between domains include Kullback–Leibler (KL) divergence[13], maximum mean difference (MMD)[14], Wasserstein distance[15], and CORAL loss[16]. Additional loss measures are introduced into the loss function and then optimized by gradient descent. Notably, it is acknowledged that this strategy can obtain effective alignment with little difference in data distribution.

However, these methods mainly focus on aligning the marginal probability distributions, which only capture the variation of global characteristics and ignore differences in the conditional distribution probabilities in different domains. This makes it challenging to handle scenarios where the differences in data distribution between different domains are more complex. Based on the recent literature[17–19], transfer learning fault diagnosis techniques are preferred by a wide range of researchers. Qian *et al.* use DenseNet as the baseline model, combined with a joint distribution adapted regularization term to get the metastable features. In this way, diagnostic capabilities are effectively migrated[17]. Li *et al.* using the representational capabilities learned in supervised learning to obtain target domain feature representations by minimizing the multi-kernel maximum mean discrepancy (MKMMD) in different feature layers between the two domains[18]. Wang *et al.* propose a method that uses multi-scale convolution to extract fault features while combining adversarial training to achieve effective migration effects. The effect of this method is close to 100% on the bearing dataset[19]. The above-mentioned studies demonstrate the effectiveness of deep transfer learning in rotary machine fault diagnosis. However, there are still some problems that have not been taken seriously: (i) Most transfer learning methods only perform domain-adaptive alignment from a global perspective. This alignment effect is greatly reduced when the data distribution varies dramatically[20]; (ii) During the validation of transfer learning algorithms, the effectiveness of transfer effects for mixed fault types on different devices is rarely considered, which is quite difficult due to the significant differences in data distribution.

To address the aforementioned challenges, this paper proposes a semi-supervised joint adaptation transfer network with conditional adversarial learning for rotary machine fault diagnosis, which introduces the following main innovative aspects.

(1) To efficiently transfer the diagnostic power learned on a large amount of data in the source domain, a pre-trained model is trained on the labeled data in the source domain and then used to generate pseudo-labels for the unlabeled target domain data. This effectively utilizes unlabeled data to boost the performance of the diagnostic model. Then, to reduce domain shifts and align the joint distribution of the source and target domains, we take into account both the global feature variation and the intra-class similarity between different domains. This enables the alignment of both the conditional probability distributions and the marginal probability distributions in different domains. This method can effectively capture both the global and local differences between the two domains and align the distributions to reduce the domain shift. This can significantly improve the diagnostic performance on the target domain and enable the use of diagnostic models in real-world scenarios where the labeled data may be scarce.

(2) Considering the mutual influence between different devices of modern rotary machines, the difficulty of fault diagnosis is significantly increased. Our method can be used in single-type fault diagnosis and produce highly reliable results. More importantly, our method has shown great improvement in diagnostic tasks involving mixed fault types, which has led to more accurate diagnostic results.

The remaining sections of this paper are organized as follows: Section 2 provides an introduction to the related definitions of transfer learning. Section 3 elaborates on the proposed method in detail. In section 4, we present experimental results on three different types of settings to showcase the effectiveness of our method. Finally, section 5 summarizes the contributions of this work and discusses potential avenues for future research.

2. TRANSFER LEARNING PROBLEM

Having sufficient annotated data is a requirement for a well-performing supervised model; however, the process of annotating data can be tedious and time-consuming. Therefore, transfer learning is a proven way to make use of a previously pre-trained model on a new task while ensuring optimal performance. The main goal is to transfer the capabilities learned in the source domain data to the target domain data, thus solving the pain point that it is difficult to obtain sufficient knowledge in the target domain with limited data[21].

From Figure 1, we can see that the traditional intelligent fault diagnosis method gives an accurate diagnosis in the case where the data distribution of the training and test sets is similar. Therefore, transfer learning is unnecessary in such cases. In general, when their data distributions are inconsistent, the generalization ability of the model is poor. In these situations, transfer learning can exploit the diagnostic power learned from the training data by reducing the difference between the two distributions.

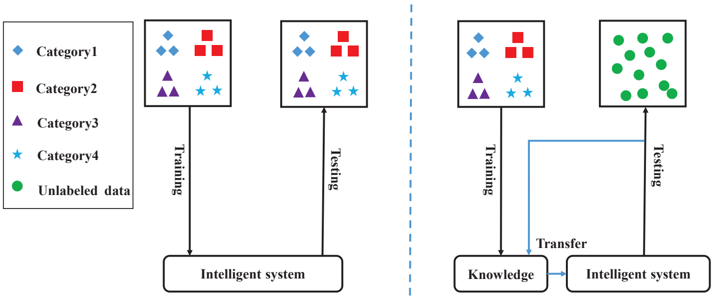


Figure 1. Traditional intelligent methods and transfer learning-based intelligent methods.

Formally, we define $D^s = (x_i^s, y_i^s)$ as labeled training data, $D^t = x_i^t$ as unlabeled test data, where s denotes the source domain task, t denotes the target domain task, and x_i and y_i represent the vectorized representation of the ith sample and the corresponding label. In addition, it is worth noting that the target domain task has no corresponding y_i^t, which means that the available labeled data in the training phase can only rely on the labeled data in the source domain, which will increase the difficulty of transfer. Since there is a great difference between the task data in two different fields, transfer learning can minimize the difference between them by finding a mapping relationship, thus realizing the reusable diagnostic ability. When the data distributions of the two domains are close, we can satisfy the assumptions on which the existing intelligent faults depend and realize an effective diagnosis.

3. THE PROPOSED ARCHITECTURE

In order to efficiently transfer the diagnostic power learned from the labeled data, a pre-trained model is obtained by generating pseudo-labels for training. A domain adaptation network, using the joint maximum mean deviation (JMMD) criterion and conditional domain adversarial (CDA) learning, is then used to learn a mapping relationship that reduces the variation in the distribution of different domains. The joint distribution between the aligned features and the predicted labels is aligned through multiple domain adaptation approaches. Meanwhile, the information from the unlabeled data is incorporated in the pre-training phase, thus resulting in maximum category differentiation and domain adaptation under multimodal conditions.

As depicted in Figure 2, the primary architecture of the proposed method is structured as follows: First, enough labeled data in the source domain are collected to train a pre-trained model. After that, the unlabeled data in the target domain are predicted to obtain pseudo-labels. Then these data are combined to extract more effective fault features. Second, the feature vectors and label vectors are linearly transformed several times to jointly model the implied relationships between them. Finally, a domain adaptation module is used to align the differences between the two data through loss function optimization. The optimization objectives include the CDA loss, the label classification loss, and the JMMD loss, respectively, in order to perform a joint optimization training of the three components.

3.1. Pre-training

The pre-trained model structure using convolutional neural networks (CNN) with bi-directional long short-term memory (BILSTM). Detailed information on the model structure is given in Table 1. To accelerate computational efficiency, the raw signal is first downsampled and then fed into the CNN. After that, the features obtained from CNN are fed again into the BILSTM to better extract the temporal information of the vibration

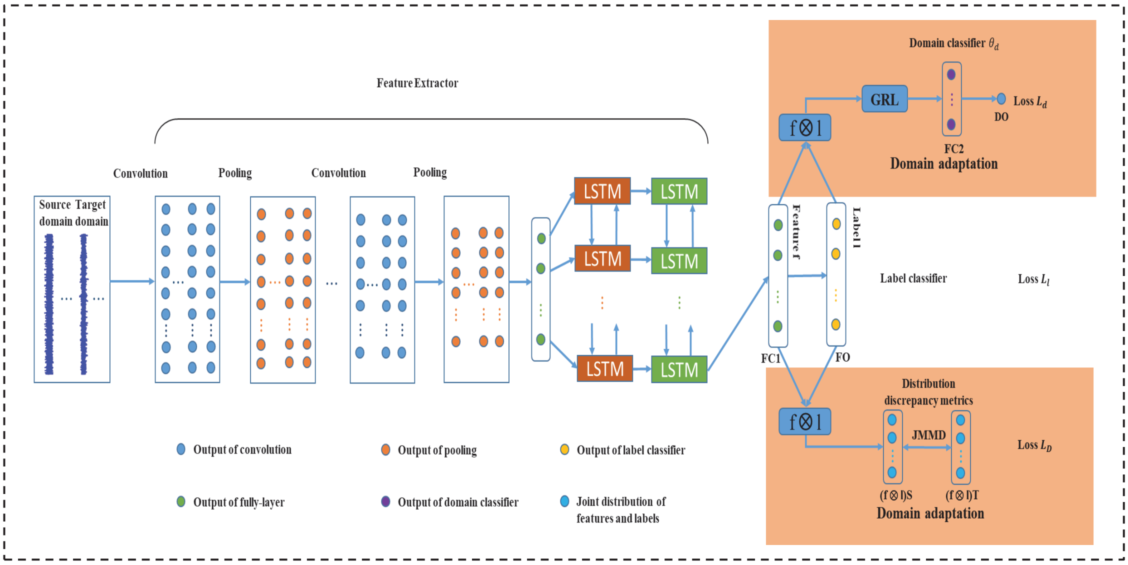


Figure 2. The structure illustration of the proposed rotary machine fault diagnosis method.

Table 1. The architecture of the one-dimensional CNN-BILSTM

Layer	Symbol domain	Operator	Parameter size
1	Input	Input Signal	1024
2	C1	Convolution	(16, kernel size = 15)
3	P1	Pooling	kernel size = 2
4	C2	Convolution	(32, kernel size = 3)
5	P2	Pooling	/
6	C3	Convolution	(64, kernel size = 3)
7	P3	Pooling	/
8	C4	Convolution	(128, kernel size = 3)
9	P4	AdaptiveMaxPooling	/
10	BILSTM	BILSTM	hidden_dim=64

signal. A large kernel size = 15 for CNN/1 is used to get low-frequency information, while CNN/2, CNN/3, and CNN/4 extract high-frequency signals and, therefore, use a smaller kernel size = 3.

In the pre-training phase, we first train with data labeled with the source domain. Unlabeled data with predicted probabilities above a threshold of 0.8 are filtered out and added to the training until convergence. Here, we simply rely on the empirical values of the task threshold above 0.8, which are relatively reliable pseudo-labels.

3.2. Domain adaptation

In order to achieve effective alignment, while the label classifier ensures the basic diagnostic ability, the domain classifier and a distance discrepancy metric module are additionally designed to further improve the effect. They correspond to the following three objective functions: (1) Minimize the classification loss of fault classification on the labeled data; (2) Maximize the domain classification error on two different domains. 3)

Minimize the JMMD distance between the two dissimilar distributions.

3.2.1. Loss-function L_l

To migrate the diagnostic capability to the target task, it is first necessary to ensure that the model has learned enough diagnostic knowledge in the source domain data. Thus, the first loss function L_l of our method is to minimize the classification loss of fault classification on the labeled data. The required objective function L_l for data with k fault classes is the standard softmax loss function.

$$L_l = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} I\left[y_i = k\right]\log\frac{e^{\left((w_j)^T x+b\right)}}{\sum_{l=1}^{k} e^{\left((w_l)^T x+b\right)}}\right] \tag{1}$$

where n is the batch size and k is the number of fault classes.

3.2.2. Loss-function L_{d}

The primary role of the domain adaptation module is to guide the network to extract domain invariant features under the constraint of the loss function. Borrowing ideas from generative adversarial networks, an adversarial domain-based training approach is added to learn the domain-invariant features. By setting a gradient reverse layer (GRL) in front of the domain classifier, the target domain data is confounded with the source domain data, thus maximizing the classification loss between the two domains. The domain classifier and feature extractor struggle with each other and finally reach a balance. Thus, domain-invariant features are learned. However, if we just align the marginal distribution between two data and ignore the correlation between labels and features, the final alignment results are poor. The conditional domain adversarial network is used to capture the cross-covariance between features and labels, thus improving the discrimination[22]. Considering the non-linear and non-smooth nature of fault signals, the joint distributions of fault features and corresponding labels need to be aligned as closely as possible to effectively transfer the diagnostic capability. Therefore, we train CDA as a second objective function here. Subsequently, the loss function L_d is shown below.

$$w(H(p)) = 1 + e^{-H(p)}, H(p) = -\sum_{k=0}^{k-1} p_k \log p_k \tag{2}$$

$$\begin{aligned} L_{\mathrm{d}} = &-\frac{1}{n_s} w\left(H\left(p_i^s\right)\right)\sum_{i=1}^{n_s}\log\left[1 - D\left(F\left(x_i^s;\theta_f\right);\theta_d\right)\right] \\ &-\frac{1}{n_t} w\left(H\left(p_i^t\right)\right)\sum_{i=1}^{n_t}\log\left[D\left(F\left(x_j^t;\theta_f\right);\theta_d\right)\right] \end{aligned} \tag{3}$$

where θ_f is the model parameter corresponding to the feature extraction module, θ_d is the parameter of the domain classifier, and k denotes the number of fault types, $H(p)$ denotes the uncertainty of the sample classification result, and $w(H(p))$ denotes the weight of each sample.

3.2.3. Loss-function L_{D}

Compared with the CDA method, spatial metric distance minimization is another approach to learning domain invariant features. The MMD method is used by Borgwardt *et. al*[23] to measure the variability of distributions. However, the effectiveness of aligning different distributions with MMD in complex multimodal conditions is limited. To address this problem, Long *et al.*[24] proposes the JMMD method to de-align the joint distribution in the feature space and label space, where the loss function L_D is defined as

$$L_{\mathrm{D}} = \left\|\mathbb{E}_S\left(z^{sf}\otimes z^{sl}\right) - \mathbb{E}_{\mathrm{T}}\left(z^{tf}\otimes z^{tl}\right)\right\| \tag{4}$$

where z^{sf} and z^{tf} represent the output of the fault feature, and z^{sl} and z^{tl} denote the vector representation of label. Unlike the standard JMMD, we add $f \otimes l$ to align the joint distribution of two domains, $f \otimes l$ refers

Table 2. The accuracy of different domain adaptation methods in CWRU datasets(%)

Method	No-TL	AdaBN	MKMMD	CORAL	JMMD	DA	CDA	OURS
Task 0-1	98.77	99.68	**100**	98.38	99.68	99.35	**100**	**100**
Task 0-2	96.49	99.48	96.43	**100**	**100**	99.03	99.68	**100**
Task 0-3	94.43	98.38	92.88	**100**	99.03	99.35	92.56	99.68
Task 1-0	97.55	95.40	98.08	98.85	**100**	99.23	97.32	**100**
Task 1-2	98.70	99.87	**100**	98.35	**100**	99.03	**100**	**100**
Task 1-3	94.82	99.03	98.71	99.68	**100**	99.68	**100**	**100**
Task 2-0	96.02	94.64	98.47	96.55	97.32	98.47	95.79	**99.23**
Task 2-1	98.18	99.29	98.05	97.08	**100**	99.03	96.43	**100**
Task 2-3	98.77	99.22	**100**	99.35	**100**	99.03	99.68	**100**
Task 3-0	87.82	90.04	84.67	**99.23**	98.08	95.79	97.70	98.85
Task 3-1	88.56	93.18	92.86	99.35	98.38	90.26	95.45	**100**
Task 3-2	87.98	95.32	96.10	**100**	99.68	97.40	98.70	**100**

to introducing two learnable weight matrices, w_1 and w_2, to unify f and l into the same dimension and add them together to represent the joint distribution of features and labels.

4. EXPERIMENTAL VERIFICATION

In this section, the proposed semi-supervised joint adaptation transfer network with adversarial learning is evaluated by examining vibration signal data from different rotary machine types, such as motor bearings, wind turbine bearings, and gearbox bearings and gears. The three datasets were used to evaluate the diagnostic capability of our method under different loads, speeds, and mixed fault-type scenarios. We conducted comparative experiments across multiple tasks using six existing transfer methods and analyzed the diagnostic effectiveness of no migration. We then demonstrate that our proposed semi-supervised method exhibits good diagnostic capability. This plays a crucial role in situations where obtaining fault data is difficult.

4.1. Case 1: CWRU bearing datasets under different loads

4.1.1. Data description

In this case, the bearing dataset is from the CWRU laboratory[25]. The experimental setup mainly consists of a dependent motor, a torque sensor/encoder, and a load motor. The bearing dataset is collected at four loads (0, 1, 2, and 3 HP). Single point faults are arranged on the bearings using electrical discharge machining (EDM) to simulate inner race faults (IF), rolling element faults (RF), and outer race faults (OF). Twelve transfer tasks are designed by migrating between the four load states. In addition, 1000 samples of length 1024 are provided for each data type. The sampling rate for our task is selected as 12 kHz. To obtain the diagnostic model, 80% of the data is used, while 20% is used to verify its validity.

A dataset of bearings with variable load conditions from the CWRU laboratory is applied to illustrate that the model could accurately classify fault types. To assess the diagnostic capability of the model, comparative tests with some commonly used domain adaptation algorithms such as MKMMD, CORAL, JMMD, domain adversarial (DA), and CDA are performed. In our article, average accuracy is a key indicator to evaluate the diagnosis results of different methods.

4.1.2. Experimental results and analysis

The comparative results of the eight different methods on the bearing dataset are shown in Table 2. Our method is still the best performer among the eight methods on this dataset, with an average accuracy of 100% for 9 out of the 12 migration tasks. Some other domain adaptation methods, including JMMD, have also achieved positive results, probably because this dataset is relatively simple and the differences in the distribution are

Table 3. The accuracy of different domain adaptation methods in JNU datasets (%)

Method	No-TL	AdaBN	MKMMD	CORAL	JMMD	DA	CDA	OURS
Task 0-1	97.44	96.96	97.27	89.08	98.12	96.08	94.54	**99.15**
Task 0-2	91.60	96.55	97.61	88.91	97.95	95.39	96.25	**98.63**
Task 1-0	85.46	91.88	88.57	88.40	**96.93**	83.62	86.69	94.88
Task 1-2	97.41	97.51	98.29	97.61	98.29	94.54	97.27	**99.32**
Task 2-0	85.02	91.16	98.46	97.44	98.81	94.54	97.10	**99.49**
Task 2-1	97.95	97.68	97.61	96.25	98.63	96.08	97.95	**99.49**

relatively small. However, the effectiveness of many such methods remains unclear. For the CWRU dataset, it is evident that the diagnostic results of the various methods are good. This is mainly due to the fact that the faults in this dataset are artificially set and have a more pronounced fault signature. In addition, the relatively small differences in the distribution of the bearing datasets collected under different loads reduce the difficulty of the migration task. In some migration tasks, such as 3-0, 3-1, and 3-2, the accuracy is only around 85%. It further implies that directly using a pre-trained model from the source domain task to predict the target domain data still produces significant errors. Thus, it is also demonstrated that this domain migration strategy is still essential. These comparative experiments provide a preliminary validation of the effectiveness of our proposed method.

4.2. Case 2: JNU bearing datasets under different speeds

4.2.1. Data description

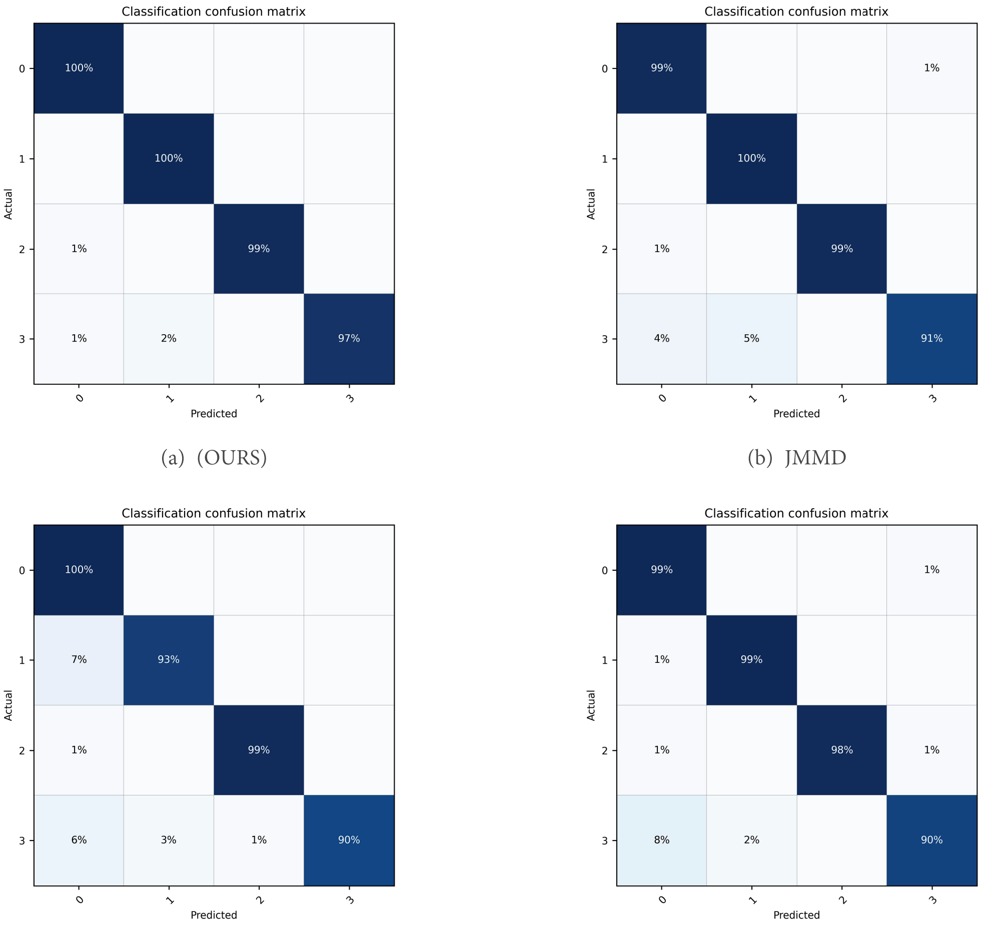
In this case, the bearing dataset is obtained from the Jiangnan University (JNU) laboratory[26]. Vibration signals are collected from wind turbine bearings at three speeds of 600, 800, and 1000 rpm, including normal bearings, rolling element failures, outer ring failures, and inner ring failures. These four faults are simulated by hand machining tiny scars on the inner ring, outer ring, and rolling element of the bearing by wire cutting. The scar size of the bearing faults is 0.25 mm × 0.7 mm. The total length of the collected data is 2,000,000, and the amount of data for each state is 500,000.

In this experiment, we labeled the settings at three different speeds of 600, 800, and 1000 rpm as tasks 0, 1, and 2, respectively. We designed a total of six transfer tasks by combining the vibration signals of the three states in a two-by-two manner. We cut the length of each sample to 1024 and then performed a comparative test with some commonly used domain adaptation algorithms, such as MKMMD, CORAL, JMMD, DA, and CDA. Again, average accuracy is a key assessment metric.

4.2.2. Experimental results and analysis

The comparative results of the eight different methods on the bearing dataset are shown in Table 3. The comparative results for the six transfer tasks under different speed conditions validate that our method still outperforms the other seven methods on this dataset, achieving the best average accuracy in five of the six transfer tasks. The result that no method could achieve 100% accuracy on this dataset implies that there are still some discrepancies in the data distribution related to this task. Other domain adaptation methods, such as JMMD, have also been found to give perfect results. These comparative experiments demonstrate the adaptability of our approach to variations in this domain in different speed scenarios.

Figure 3 illustrates the details of the four best diagnostic results using the confusion matrices. From the visualization of the confusion matrix, we know that JMMD, CDA, and MKMMD have a larger classification error on the fourth fault type. It can be attributed to the fact that the data distribution of the failure types varies considerably under different speed conditions. The diagnostic effectiveness of a single strategy is quite limited.



(a) (OURS)

(b) JMMD

(c) CDA

(d) MKMMD

Figure 3. Confusion Matrix of four different methods on gearbox dataset

By using a joint domain adaptation migration network to de-target the alignment to reduce the joint distribution differences between two different domains, the accuracy of our proposed method in this fault type has been dramatically improved. At the same time, a conditional confrontation training module was introduced to help improve the alignment effect to deal with domain drift. Finally, the most significant differences between the different categories were obtained. The above-mentioned results provide sufficient evidence of the transferability of our proposed fault diagnosis method.

4.3. Case 3: SEU gearbox datasets with mixed fault

4.3.1. Data description

We use the bearing and gearbox dataset from Southeast University in China in this experiment[27]. The experimental platform, DDS, consists mainly of a motor, a planetary gearbox, and a parallel gearbox. The fault signals are obtained under two different working conditions, 20Hz-0V and 30Hz-2V. The dataset for the gearbox includes the fault signal of the planetary gearbox in the X, Y, and Z directions. There are four types of faults: broken teeth, missing teeth, root faults, and surface faults, and one normal type for healthy working conditions. The bearing data are available for four types of faults: inner ring, outer ring, rolling element, and mixed inner and outer rings. In order to evaluate the performance of our approach when dealing with mixed fault types, gear and bearing fault data from the SEU dataset were combined into a mixed dataset. There are nine fault types in this mixed dataset, including four gear faults, four bearing faults, and one normal data. There

Table 4. The accuracy of different domain adaptation methods in SEU datasets (%)

Method	No-TL	AdaBN	MKMMD	CORAL	JMMD	DA	CDA	OURS
Task 0-1(TD)	45.43	49.21	59.97	50.59	65.40	54.40	59.53	**75.95**
Task 1-0(TD)	56.16	57.89	67.45	58.44	68.62	58.80	65.54	**72.87**
Task 0-1(FD)	35.19	41.38	44.57	42.52	45.45	43.70	43.26	**50.15**
Task 1-0(FD)	42.99	49.53	44.28	51.17	61.29	53.96	52.93	**62.90**

are 1000 samples for each data type, and each sample is 1024 in length. Thus, this dataset consists of 9,000 data samples. Finally, we use 80% of the data to obtain the diagnostic model and 20% of the data to verify its effectiveness.

In this experiment, to demonstrate the transfer effectiveness of the proposed method under different load and velocity operating conditions, we collected vibration signals for two different states, 20Hz-0V and 30Hz-2V, and named Task 0 and Task 1, respectively. We validated the model by combining the vibration signals for the two states in a two-by-two fashion. In addition, two additional different signal forms were set up, with both time and frequency domain signals considered as inputs, and a total of four different transfer tasks were designed to validate the model. In order to evaluate the performance of our method in this case of widely varying data distributions with different load and speed conditions, comparative tests were carried out with some commonly used domain adaptation algorithms, such as MKMMD, CORAL, JMMD, DA, and CDA. In this case, we also choose average accuracy as a key assessment metric.

4.3.2. Experimental results and analysis

The comparative results of the eight different methods on the bearing dataset are shown in Table 4. The results of the four transfer tasks under different load and speed conditions show that our method still performs the best of the eight different methods on this dataset, with the best average accuracy in all four transfer tasks. It must be noted that a high level of accuracy is not achieved on this dataset, and it is evident that there are significant differences in the data distribution between the two domains on this task. The main reason lies that the vibration signals collected at different speeds and loads are inherently different. In addition, there are a number of mixed fault types in this task, such as mixed inner and outer ring faults and both bearing and gearbox faults, which can affect the final transfer results. It is worth noting that the JMMD method, which performs quite effectively in the first two tasks, differs from the best results by around 7-8% on this task. Since the data distribution is complex and varies significantly, domain adaptation strategies alone are not sufficient to align the distribution well enough to achieve good diagnostic performance.

On the one hand, domain adaptation is performed at the feature extraction and classification layers via JMMD by exploiting the differences in the joint distribution. On the other hand, adversarial domain training is performed by adjusting the joint distribution to reduce domain drift. These two modules achieve maximum category differentiation and domain adaptation in multimodal conditions. Finally, the advantages and disadvantages of the diagnostic approaches are verified in two cases: using the original time domain signal directly as inputs versus transforming the data into the frequency domain and using that as inputs. It turns out that in this task, the time domain signal is used directly as input to obtain better diagnostic results. The reason for this phenomenon may be that the time-domain representation is more capable of intuitively reflecting the amplitude, frequency, and phase information of the signal over time and can better display the waveform shape of the signal, which is very helpful for detecting short-term signal changes and analyzing signal shape. Additionally, frequency domain representation mainly provides information about the frequency components and relative strengths of the signal but may not be able to fully reflect all the information about the signal, especially if the signal is very complex or contains multiple frequencies. In addition, the interpretation of frequency domain representation may be more difficult to understand and may require higher levels of professional knowledge for analysis. Although frequency domain representation can provide valuable information about the frequency components of the signal, it may not be as effective in capturing the complex time characteristics of the signal. Therefore, time domain representation is more prominent in terms of intuitiveness and practicality. Through these comparative tests, it is demonstrated that the transfer effects of our proposed method are practical for different speed scenarios.

In summary, the improvement in diagnostic performance achieved by our method can be attributed to the combination of JMMD and adversarial domain training modules, which effectively address the challenges of domain adaptation in multimodal conditions. Additionally, the use of the time domain signal as input also contributes to the improvement in diagnostic performance.

5. CONCLUSION

This paper proposes a novel semi-supervised joint adaptation transfer network with conditional adversarial learning for fault diagnosis of the rotary machine, which can effectively solve the problem of poor diagnosis due to insufficient data in the target domain. The proposed fault diagnosis method first incorporates information from unlabeled target domain data by introducing a pre-trained model. Two domain adaptation modules are then used to close the distance between the distributions of different domains, thereby improving the effectiveness of the diagnostics of mutual migrations in the two different domains. Ultimately, our approach is validated to achieve reliable results for variable loads, variable speeds, and mixed fault-type diagnostic tasks in three different experimental settings. However, the method we proposed has not been validated using fault data obtained from real scenarios, where the fault patterns are typically more complex, and the data often contains a significant amount of noise. As a result, there is a possibility that the performance of this method could be affected.

In this work, we focus more on domain adaptation between data in different domains so that pseudo-labels use only empirical thresholds to filter reliable labels. In future investigations, we will focus on how to filter for more reliable pseudo-labels in order to make the best possible use of unlabeled data and further improve the diagnosis of tasks with insufficient labeling data.

DECLARATIONS

Authors' contributions

Made substantial contributions to the conception and design of the study and performed data analysis and interpretation: Liu C, Li S

Performed data acquisition and provided administrative, technical, and material support: Chen H, Xiu X, Peng C

Availability of data and materials

CWRU:[25] JNU:[26] SEU:[27]

Financial support and sponsorship

This work was supported by the National Natural Science Foundation of China (62103250, 62273223, and 62173218); Shanghai Sailing Program (21YF1414000); Project of Science and Technology Commission of Shanghai Municipality, China (22JC1401401).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Xu X, Cao D, Zhou Y, Gao J. Application of neural network algorithm in fault diagnosis of mechanical intelligence. *Mech Syst Signal Pr* 2020;141:106625. DOI
2. Qiao W, Lu D. A survey on wind turbine condition monitoring and fault diagnosis—part I: components and subsystems. *IEEE Trans Ind Electron* 2015;62:6536–45. DOI
3. Hoang DT, Kang HJ. A survey on Deep Learning based bearing fault diagnosis. *Neurocomputing* 2019;335:327–35. DOI
4. Lei Y, Yang B, Jiang X, et al. Applications of machine learning to machine fault diagnosis: a review and roadmap. *Mech Syst Signal Pr* 2020;138:106587. DOI
5. Tran MQ, Amer M, Dababat A, Abdelaziz AY, Dai HJ, et al. Robust fault recognition and correction scheme for induction motors using an effective IoT with deep learning approach. *Measurement* 2023;207:112398. DOI
6. Gong W, Chen H, Zhang Z, et al. A novel Deep Learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. *Sensors* 2019;19:1693. DOI
7. Pandey SK, Janghel RR. Recent Deep Learning techniques, challenges and its applications for medical healthcare system: a review. *Neural Process Lett* 2019;50:1907–35. DOI
8. Liu R, Yang B, Zio E, Chen X. Artificial intelligence for fault diagnosis of rotating machinery: a review. *Mech Syst Signal Pr* 2018;108:33–47. DOI
9. Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning. *Proceedings of the IEEE* 2021;109:43–76. DOI
10. Qian C, Zhu J, Shen Y, Jiang Q, Zhang Q. Deep transfer learning in mechanical intelligent fault diagnosis: application and challenge. *Neural Process Lett* 2022;54:2509–31. DOI
11. Yang X, Chi F, Shao S, Zhang Q. Bearing fault diagnosis under variable working conditions based on deep residual shrinkage networks and transfer learning. *J Sensors* 2021;2021:1–13. DOI
12. Kouw WM, Loog M. A review of domain adaptation without target labels. *IEEE Trans Pattern Anal Mach Intell* 2021;43:766–85. DOI
13. Hershey JR, Olsen PA. Approximating the kullback leibler divergence between gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07. IEEE; 2007. DOI
14. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:14123474* 2014.
15. Shen J, Qu Y, Zhang W, Yu Y. Wasserstein Distance Guided Representation Learning for Domain Adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence* 2018;32. DOI
16. Sun B, Saenko K. Deep CORAL: correlation alignment for deep domain adaptation. In: Lecture Notes in Computer Science. Springer International Publishing; 2016. pp. 443–50. DOI
17. Qian C, Jiang Q, Shen Y, Huo C, Zhang Q. An intelligent fault diagnosis method for rolling bearings based on feature transfer with improved DenseNet and joint distribution adaptation. *Meas Sci Technol* 2021;33:025101. DOI
18. Li X, Zhang W, Ding Q, Sun JQ. Multi-Layer domain adaptation method for rolling bearing fault diagnosis. *Signal Process* 2019;157:180–97. DOI
19. Wang Y, Ning D, Lu J. A novel transfer capsule network based on domain-adversarial training for fault diagnosis. *Neural Process Lett* 2022;54:4171–88. DOI
20. Li W, Huang R, Li J, et al. A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mech Syst Signal Process* 2022;167:108487. DOI
21. Yao S, Kang Q, Zhou M, Rawa MJ, Abusorrah A. A survey of transfer learning for machinery diagnostics and prognostics. *Artificia Intell Rev* 2022;56:2871–922. DOI
22. Long M, CAO Z, Wang J, Jordan MI. Conditional adversarial domain adaptation. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, et al., editors. Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc.; 2018. Available from: https://proceedings.neurips.cc/paper_files/paper/2018/file/ab88b15733f543179858600245108dd8-Paper.pdf.
23. Borgwardt KM, Gretton A, Rasch MJ, et al. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics* 2006;22:e49–57. DOI

24. Long M, Zhu H, Wang J, Jordan MI. Deep transfer learning with joint adaptation networks. In: International conference on machine learning. PMLR; 2017. pp. 2208–17.
25. "Case Western Reserve University Bearing Data Center Website";. https://engineering.case.edu/bearingdatacenter.
26. Li K, Ping X, Wang H, Chen P, Cao Y. Sequential fuzzy diagnosis method for motor roller bearing in variable operating conditions based on vibration analysis. *Sensors* 2013;13:8013-41. DOI
27. Shao S, McAleer S, Yan R, Baldi P. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Trans Indus Inf* 2019;15:2446–55. DOI

Lei *et al. Intell Robot* 2023;3:145-7
DOI: 10.20517/ir.2023.08

Intelligence & Robotics

Editorial

Open Access

ChatGPT in connected and autonomous vehicles: benefits and challenges

Lei Lei[1], **Hao Zhang**[2], **Simon X. Yang**[1]

[1]School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada.
[2]College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China.

Correspondence to: Dr. Lei Lei, School of Engineering, University of Guelph, 50 Stone Road East, Guelph ON N1G 2W1, Canada. E-mail: leil@uoguelph.ca

How to cite this article: Lei L, Zhang H, Yang SX. ChatGPT in connected and autonomous vehicles: benefits and challenges. *Intell Robot* 2023;3:145-7. https://dx.doi.org/10.20517/ir.2023.08

Received: 20 Apr 2023 **Accepted:** 8 May 2023 **Published:** 25 May 2023

Academic Editor: Jinguo Liu **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

The OpenAI chatbot ChatGPT has achieved unprecedented success since its launch in November 2022. The Artificial Intelligence (AI) technologies behind ChatGPT are expected to have far-reaching effects on various technological fields beyond natural language processing. This editorial discusses the potential benefits and challenges that ChatGPT may bring to the connected and autonomous vehicles (CAVs). CAVs have been heavily researched in both the automotive and communications industries in recent years, where the AI technologies have played an indispensable role. Exploring how and to what extent ChatGPT will affect this field is an interesting and timely research topic.

Keywords: ChatGPT, connected and autonomous vehicles

1. BACKGROUND

OpenAI has announced a major update to the software that powers ChatGPT. While the program previously ran on Generative Pre-trained Transformer 3 (GPT-3) technology, OpenAI has now officially launched GPT-4. ChatGPT's popularity has had a broad impact and may even disrupt many traditional methods in various technological fields[1]. This brings up the question of whether and how ChatGPT will change the development direction of connected and autonomous vehicles (CAVs).

As we all know, CAVs have undergone major technological breakthroughs in recent years. Both automakers and communication companies have high expectations for their adoption. However, whether they are entertainment or safety-related applications for autonomous driving, the development and commercialization of CAVs have not been entirely satisfactory.

As a result, when ChatGPT appears, automakers are eager to incorporate it into their vehicles. For example, general motors (GM) is developing a new in-car assistant based on the technology behind OpenAI's massively popular ChatGPT[2]. Simultaneously, discussions began about how to use ChatGPT technology to further the development of CAVs and what problems and challenges it would bring to this field[3-7]. This editorial will discuss the above issues briefly.

2. POTENTIAL BENEFITS

ChatGPT is a human-machine interaction tool based on natural language processing (NLP) that can effectively improve communication and interaction between drivers and vehicles. The underlying technology of ChatGPT is a large-scale deep learning network, making it suitable for processing massive amounts of data and driving decisions in CAVs. Therefore, ChatGPT may improve the performances of CAVs in the following ways.

2.1. In-vehicle voice assistants

Until self-driving cars become more common, the majority of vehicle networking services offered are for entertainment or driving assistance. Both types of services require drivers to interact with the vehicle and remote servers in a timely and effective manner. Voice is the most appropriate interaction method in terms of safety. ChatGPT's strong language processing capabilities make it ideal for use as an in-car voice assistant. ChatGPT has the potential to improve the state-of-the-art in-car interaction system by better understanding human instructions and making more intelligent responses.

2.2. Complex environment perception

Vehicles can perceive increasing amounts of data from their surroundings by using various in-vehicle sensors, such as cameras, radar, and Lidar. However, the processing capabilities of the vehicle are ineffective in processing this massive amount of data in a timely manner. We can efficiently process various perceptual data of the vehicle, obtain comprehensive and in-depth environmental awareness knowledge and provide a solid base for the implementation of autonomous driving technologies, if we deploy GPT-4 models to edge computing devices closer to the vehicle and combine them with the vehicle's local machine learning model. Moreover, although the Transformer architecture behind ChatGPT was initially proposed for NLP, researchers have successfully applied it in computer vision for improving the accuracy of vision processing[3]. This is an optimistic development for autonomous driving, as several high-profile incidents in CAVs are mainly caused by the failure of AI (Artificial intelligence)-based computer vision models.

2.3. Driving behavior decision-making

The performance of ChatGPT is enhanced through the reinforcement learning from human feedback (RLHF) technique, where a reward model is first trained based on human feedback. The reward model is then used in reinforcement learning (RL) to fine-tune the pre-trained language models. This learning paradigm is highly appealing for decision-making in autonomous driving, as human feedback will help CAVs learn from human drivers' driving behaviors and habits to enhance driving safety and comfort. Although imitation learning has been studied for CAVs to learn from human drivers, the lack of labels or human driving behaviors has impeded its practical applications. With the trained reward models in RLHF, human feedback can be simulated to train the decision-making models in CAVs and thus effectively deal with the scarcity of labels.

2.4. Intelligent anomaly detection

ChatGPT can provide intelligent detection for abnormal data and external threats received while the vehicle is moving based on its own powerful learning capabilities. This not only enhances the intelligence of traditional detection mechanisms but also increases the detection rate of unknown 0-day anomalies in the vehicle. This enables the intelligent anomaly detection mechanism based on ChatGPT enhancements to improve the security of CAVs.

2.5. Adversarial defense

ChatGPT is highly susceptible to being utilized by hackers to design highly stealthy and destructive attack strategies against CAVs. Due to this potential threat, it is a potential research direction to equally utilize ChatGPT for smart vehicles to design adversarial defense strategies that can guarantee the safe driving of vehicles.

3. CHALLENGES

Any new technology, including ChatGPT, usually brings with it both opportunities and challenges.

3.1. Data privacy issue

To ensure continuous training of the model and consistent accuracy and performance, ChatGPT must frequently interact with users. This necessitates the collection of a significant amount of user-related data. However, currently, OpenAI has not provided effective notification on collecting and processing user information. As a result, there is a lack of legal basis for collecting and storing personal information.

3.2. Security issue

ChatGPT will experience similar challenges to other deep learning models, such as the production of inaccurate output following a malicious assault. As CAVs are connected through wireless communications, they are highly vulnerable to network attacks. Consequently, if such an issue were to occur under certain circumstances, it could lead to serious traffic accidents with decision-making in automatic driving based on ChatGPT.

3.3. Real-time issue

ChatGPT is based on a large-scale network with a massive number of parameters that require a large amount of computing power. However, computing power is relatively scarce in the current vehicle network, especially within the vehicle itself. The most suitable place for the deployment of the GPT-4 model is still on remote cloud servers. As a result, how to transmit the output of model processing back in time will pose a serious challenge to vehicle-to-everything (V2X) communication.

3.4. Ethical issue

As an AI-based technology, ethical issues are a big challenge for ChatGPT-based decision-making of autonomous vehicles. Some ethical considerations should be addressed to avoid possible dilemmas, such as various biases resulting from the training datasets, the responsibility of decision-making that may result in huge losses, and possible misuse for malicious purposes. Similar to AI-based medical applications, the ethical decisions should be made by humans who operate the autonomous vehicles, although the assistance by ChatGPT can significantly facilitate and improve the efficiency and effectiveness of those decisions.

4. CONCLUSIONS

This editorial has briefly discussed the benefits and challenges that ChatGPT may bring to the CAVs. The AI technologies behind ChatGPT have the potential to improve the performance of CAVs in several aspects, including in-vehicle voice assistants, complex environment perception, driving behavior decision-making, and intelligent anomaly detection. Meanwhile, there are several challenges that must be addressed when introducing ChatGPT into CAVs, including data privacy, security, real-time, and ethical concerns.

It is foreseeable that ChatGPT will inevitably be applied to future CAVs. However, it is still not clear how and to what extent it will affect this field. Related research in this area is in urgent demand.

DECLARATIONS

Authors' contributions

Wrote the first draft: Lei L
Made additions to the first draft: Zhang H, Yang SX

Availability of data and materials

Not applicable

Financial support and sponsorship

None

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Abdullah M, Madain A, Jararweh Y. ChatGPT: fundamentals, applications and social impacts. 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), Milan, Italy, 2022, pp. 1-8. https://ieeexplore.ieee.org/document/10062688 [Last accessed on 10 May 2023].
2. Morrison R. TechMonitor, GM bringing ChatGPT-like assistant to its cars. Available from: https://techmonitor.ai/technology/ai-and-automation/gm-bringing-chatgpt-like-assistant-to-its-cars [Last accessed on 10 May 2023].
3. Yu YC. VicOne, Is ChatGPT the next AI milestone for autonomous vehicles? Available from: https://www.vicone.com/blog/is-chatgpt-the-next-ai-milestone-for-autonomous-vehicles [Last accessed on 10 May 2023].
4. Synopsys, From ChatGPT to computer vision processing: how deep-learning transformers are shaping our world. Available from: https://medium.com/@synopsys/from-chatgpt-to-computer-vision-processing-how-deep-learning-transformers-are-shaping-our-world-6f58b86d1346 [Last accessed on 10 May 2023].
5. Gao Y, Tong W, Wu EQ, et al. Chat with ChatGPT on interactive engines for intelligent driving. In: IEEE Transactions on Intelligent Vehicles, vol. 8, no. 3, pp. 2034-2036. Available from: https://ieeexplore.ieee.org/document/10059220 [Last accessed on 10 May 2023].
6. Zhang J. HiVeGPT: human-machine-augmented intelligent vehicles with generative pre-trained transformer. In: IEEE Transactions on Intelligent Vehicles, vol. 8, no. 3, March 2023, pp. 2027-2033. Available from: https://ieeexplore.ieee.org/document/10068744 [Last accessed on 10 May 2023].
7. Du H, Teng S, Chen H, et al. Chat with ChatGPT on intelligent vehicles: an IEEE TIV perspective. In: IEEE Transactions on Intelligent Vehicles, vol. 8, no. 3, March 2023, pp. 2020-2026. Available from: https://ieeexplore.ieee.org/abstract/document/10061356 [Last accessed on 10 May 2023].

Intelligence & Robotics

Research Article

Open Access

Application of distributed and decentralized technologies in the management of intelligent transport systems

Luba Eremina, Anton Mamoiko, Guo Aohua

[1]Department of Transport and Traffic Management, Don State Technical University, Rostov on Don 344003, Russia.

Correspondence to: Dr. Luba Eremina, Department of Transport and Traffic Management, Don State Technical University, Gagarina sq.1, Rostov on Don 344003, Russia. E-mail: runa666.6@mail.ru

How to cite this article: Eremina L, Mamoiko A, Aohua G. Application of distributed and decentralized technologies in the management of intelligent transport systems. *Intell Robot* 2023;3(2):149-60. https://dx.doi.org/10.20517/ir.2023.09

Received:13 Feb 2023 **First Decision:** 20 Apr 2023 **Revised:12** May 2023 **Accepted:** 18 May 2023 **Published:** 31 May 2023

Academic Editor: Simon X. Yang **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

Shifting focus from the field of distributed and decentralized technologies in the management of intelligent transportation systems (ITS), we now delve into the specific application of blockchain in transportation management. Blockchain is a fundamental component of distributed and decentralized technologies. The research paper discusses the utilization of blockchain technology in managing transportation systems through multi-agent systems. Specifically, the use of blockchain technology is examined in the context of the quick road system (QRS) in ITS to provide a service for obtaining a special fare status. This service aims to establish a decentralized network that facilitates real-time road lane sharing. The study indicates that depending on traffic situations, drivers can share their lane space with other vehicles traveling on the same route by exchanging incentives via blockchain with other private car owners, thereby allowing for faster travel for individuals in a hurry or those requiring priority access to fast lanes. The paper also addresses the increasing number of connected devices in ITS due to the development of the internet of things (IoT) technology. It highlights the importance of efficiently utilizing large datasets and identifies the internet of vehicles (IoV) as a crucial area of integration for existing IoT technologies to address smart traffic within multi-agent systems.

Keywords: Traffic, blockchain, intelligent transport systems, Internet of Things, smart contracting, end-to-end technologies, digitalization

1. INTRODUCTION

Intelligent transportation systems (ITS) have become increasingly important in recent years as they provide a range of services aimed at improving the efficiency, safety, and sustainability of transportation networks. However, traditional ITS solutions often suffer from several limitations, such as security vulnerabilities, data silos, and inefficiencies in data exchange and processing. To address these challenges, blockchain technology has emerged as a promising solution for the transportation industry.

Blockchain technology is characterized by its decentralized, secure, and immutable nature, which makes it ideal for creating secure and transparent platforms for exchanging data between different stakeholders in a transportation network, such as vehicles, roadside devices, and traffic management systems. In addition, blockchain technology can be used to create decentralized platforms for managing various transportation processes, such as payment for transportation services, vehicle identity management, and supply chain management.

Before delving into the subject, let us examine the characteristics of distributed decentralized technologies. The ITS and the Vehicular Ad Hoc Network (VANET) are closely related since VANET is a specific type of ITS that uses wireless communication technology to allow vehicles to interact with each other and with the infrastructure around them.

In an ITS, various technologies are employed to gather and analyze data from different sources, such as sensors and cameras, to provide real-time information to drivers, traffic managers, and other stakeholders. This information is used to optimize traffic flow, enhance safety, and minimize congestion. Additionally, ITS can facilitate communication between vehicles and the surrounding infrastructure, such as traffic lights and road signs, to provide additional information to drivers and improve traffic management.

VANET is an essential technology used in ITS that enables communication between vehicles and infrastructure through wireless networks. VANET allows vehicles to exchange information on traffic conditions, road hazards, and other crucial data. This information enables vehicles to make informed decisions about their routes and speed, leading to enhanced traffic flow and safety.

Overall, ITS and VANET work together to create a smarter and more connected transportation system. By integrating wireless communication technology into the infrastructure and vehicles, ITS can provide real-time information and communication that can improve traffic flow, reduce accidents, and enhance the overall driving experience.

Task offloading in VANET refers to the allocation of computational tasks between vehicles and roadside infrastructure[1]. Vehicles may have limited processing power and storage capacity, while roadside infrastructure can provide additional computational resources. One solution to this issue is a privacy-aware multi-agent deep reinforcement learning approach for task offloading in VANET. This approach involves a deep reinforcement learning algorithm to learn the optimal task offloading strategy, a multi-agent system to coordinate the task offloading process between the vehicles and roadside infrastructure, and a privacy-aware mechanism to ensure that the privacy of agents is preserved during the task offloading process.

Although the proposed approach has the potential to improve the efficiency of task offloading in VANET and preserve the privacy of the agents involved, it has its limitations. The approach involves multiple components, which may add complexity to the task offloading process. Furthermore, the use of deep reinforcement learning may require significant computational resources, which may limit the feasibility of the approach in real-world VANET systems with limited resources. Additionally, there is always a risk that sensitive data could be leaked or exposed during the task offloading process, even with the privacy-aware mechanism in place.

In recent years, cloud-assisted VANETs and internet of things (IoT) applications have become increasingly popular. However, these systems have also raised concerns about privacy, energy efficiency, and resource allocation. Two recent research papers, "PPVF: privacy-preserving protocol for vehicle feedback in cloud-assisted VANET"[2] and "AFED-EF: An energy-efficient virtual machine (VM) allocation algorithm for IoT applications in a cloud data center"[3], propose solutions to these challenges.

The PPVF protocol presented in the first paper aims to enable privacy-preserving feedback collection in cloud-assisted VANET systems. The protocol utilizes a group signature scheme for anonymous authentication of the vehicle, a zero-knowledge proof for proving the correctness of the feedback without revealing the actual data, and a homomorphic encryption scheme for computation on encrypted data. While the PPVF protocol has the potential to improve privacy, it may also introduce complexity, computational requirements, communication overhead, and security risks. Therefore, careful consideration is necessary before implementing the protocol in practical applications.

The algorithm AFED-EF is designed for energy-efficient VM allocation in cloud data centers that host IoT applications. The algorithm is based on the Ant Colony Optimization (ACO) algorithm and includes energy-saving techniques such as dynamic voltage and frequency scaling (DVFS) and VM consolidation on physical servers. AFED-EF considers the workload characteristics of IoT applications and the energy efficiency of the physical servers in the cloud data center. The algorithm offers several advantages, including improved energy efficiency, reduced operational costs, and improved performance for IoT applications. However, the algorithm may have limitations and require further testing and refinement in practical applications.

Overall, both papers present promising solutions to challenges in cloud-assisted VANETs and IoT applications. However, as with any research paper, further testing and refinement is necessary before implementing the proposed solutions in practical applications. These papers demonstrate the importance of considering privacy, energy efficiency, and resource allocation in designing and implementing cloud-assisted VANETs and IoT applications.

The integration of blockchain technology in ITS has the potential to revolutionize the transportation industry by addressing key challenges faced by traditional ITS solutions. By providing secure and decentralized platforms for exchanging data, managing transportation processes, and facilitating payment systems, blockchain technology can help to create more efficient, secure, and transparent transportation networks. In this paper, we provide a comprehensive review of the current state of the art in the use of blockchain technology in ITS, with a focus on its applications in secure data exchange, decentralized platforms, and payment systems.

The potential of blockchain technology in managing transport systems lies in its ability to create decentralized ITS[4]. However, to fully realize this potential, there are several key issues that must be addressed. The objective of this study was to explore the use of blockchain technology in managing transport systems within multi-agent systems that account for the independence and limited view of the participants of the system and its decentralized nature.

The study focused on utilizing the decentralized features of blockchain technology and IoT technologies in the transport system. The rapid growth of connected devices in the IoT is expected to continue, creating significant opportunities for enhancing transport management[5].

The optimal utilization of large amounts of data has become a significant research area in multiple fields. The internet of vehicles (IoV) is a key area of focus for integrating existing IoT technologies to address the issue of intelligent traffic. The prevention of road incidents is crucial, and regulations are more effective than quick reactions. ITS indirectly improve traffic safety and reduce accidents. While significant effort has been invested in creating ITS, safety issues must be addressed. Blockchain technology can solve potential safety problems associated with traditional centralized security systems[6].

Data protection has two critical aspects: trust and privacy. The decentralization of a blockchain system ensures a reliable level of security. Personal data can be shared only with users, leading to an intelligent transport system monitoring system. The trust and privacy of blockchain align with the concept that personal rating of system participants is closely linked to their behavior on the road. Therefore, a safer driver will have a reputation as a trustworthy driver.

To coordinate real-time traffic monitoring and control, individual user ratings can be connected to road behavior to incentivize safe driving. While a single traffic regulation cannot solely improve the road environment, efforts should be made by the community and social environment. By monitoring and managing intelligent traffic, precise action can be taken against dangerous drivers, ultimately improving the traffic situation during rush hours[7].

A credit token, a type of blockchain, could be used to record traffic violations and exchange big data between agencies, preventing the isolation of intelligent transport systems. A credit token transaction system based on blockchain technology could be created and integrated within the road platform. The credit token represents personal reputation and works only within the road platform. To reduce accidents and create a better urban transport environment, the token payment process is linked to the correct lane change and the current selection of a safe speed limit[8].

2. BLOCKCHAIN TECHNOLOGY AS A PLATFORM ITS

2.1. Parallel transport management

Let us outline the main research questions and potential ideas.

2.2.1. Decentralized autonomous transportation systems

Decentralized autonomous transportation systems leverage the core features of blockchain technology. Peer-to-peer (P2P) networks, based on distributed consensus and economic incentives, provide a natural approach to modeling complex transportation systems. Each computing node (such as IoT devices, vehicles, or other objects with computing power) can function as an autonomous agent in this system[9].

A multitude of such nodes can be linked to a shared network and interact with each other through diverse block-based decentralized applications (Dapps), leading to the creation of decentralized autonomous organizations (DAOs) geared towards specific requirements and tasks. Moving to the macro-level, this can result in the development of a decentralized autonomous system and even a community of systems (DAS)[10].

2.1.2. Creating incentives for crowdsourcing

The process of distributed consensus in blockchain-based systems can be seen as a crowdsourcing task performed by a large number of nodes that contribute their computing power to verify the blockchain data[11]. Since these nodes act as independent agents, any crowdsourcing mechanisms and incentives must ensure that the individual behavior of each node, motivated to maximize its income, is aligned with the overall goal of maintaining system reliability and security. The block technology has the potential to bring together all available computing resources in ITS to tackle hitherto unsolvable problems, such as more precise real-time transportation management and control. However, further research is needed to design effective incentives for crowdsourcing in decentralized ITS.

2.1.3. Development of software for establishing trust within ITS

The trust that stems from blockchain technology plays a critical role in creating decentralized ITS and enabling its application to tasks such as P2P commerce, payment, and communication[12]. This trust is secured by the code and the validation of the majority of the process participants. The technology has the potential to simplify structurally complex systems, thereby reducing social issues. It will allow for the transfer of currency and assets between legal entities and individuals without hindrance. For example, P2P trust could enable cars to be resold and registered directly through blockchain applications rather than centralized bodies or platforms.

Further research is needed to address the fundamental principles underlying trust and trust management in this domain.

2.1.4. Intelligent contracts for intelligent transport systems

Smart contracts act as a catalyst for blockchain by utilizing various algorithms (such as machine learning and big data analysis) and high-level logic programs to construct an ITS software ecosystem and enhance its application intelligence[13]. These self-executing contracts significantly reduce social complexity by minimizing the importance of human involvement, serving as software agents acting on behalf of their creators or even themselves. Therefore, there is a pressing need to explore the development and implementation of specific smart contracts and their management and control of ITS.

2.1.5. Data security and privacy protection

While blockchain has demonstrated high reliability and security, the encryption structure must be further strengthened in ITS with numerous devices to safeguard against attacks.

Several researchers[14] have proposed the concept of PTMS (Parallel Transportation Management and Control System), which optimizes the real transport system by simultaneously interacting with its artificial or virtual counterparts. Blockchain is considered one of the most secure and reliable architectures for PTMS, making it an important step toward the realization of this concept. Figure 1 below illustrates one of the possible applications of blockchain technology in PTMS.

A PTMS, based on blockchain, will encompass all physical objects, such as IoT devices, vehicles, and assets, which can be easily digitized through the "blockchain of things" and registered in the blockchain online. The transmission of large data in cyberspace can also be integrated into the blockchain. Furthermore, using the Ethereum platform and its programmable scripts to support complex modeling and computation[15], one or more artificial transport systems can be created in the code space of smart contracts.

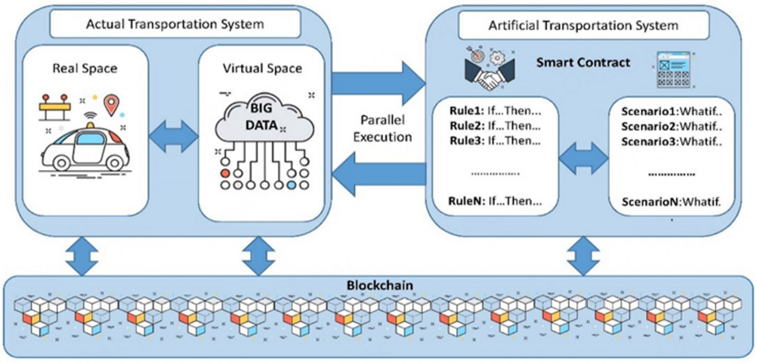


Figure 1. Parallel transport management[4].

Using these jointly developed real and artificial transport systems, multi-directional computational experiments can be designed and conducted to assess and validate specific behavior, mechanisms, and strategies in ITS (e.g., to evaluate traffic conditions). These experiments can be designed as "What If" scenarios and modeling based on predefined "If Then" rules. The optimal solution will be developed through a large number of computational experiments and returned to real transport systems. This process will be repeated indefinitely until the actual transport system approaches its optimal artificial counterparts[16].

2.2. Quick road system

One possible scenario for utilizing blockchain in intelligent transport systems is to create a service that provides a way for drivers to obtain unimpeded passage, referred to as quick road system (QRS)[4]. This service aims to establish a decentralized network of road lane sharing in real time. If a driver is in a rush or desires priority in utilizing the speed lane, they can designate a special status and share their place in the lane with other vehicles traveling the same route by exchanging incentives through the blockchain with other private car owners.

To function as one of the QRS computing nodes, a special application can be installed on the smartphone of a driver or integrated into the vehicle software, referred to as "Road miners". Real-time data is verified and stored on a P2P network supported by the community. All lane and payment sharing behavior is coordinated and performed through this network, with the road miners connected to the P2P network without any central authority.

One possible solution for incentivizing road miners to contribute to the QRS is to introduce an innovative consensus algorithm called "proof of traffic". This algorithm encourages road miners to use the QRS application while driving and share their traffic data, thereby building a local network of lane social use. As a reward, QRS generates new tokens called "QRS" for road miners, which can be used to pay for travel and other transport services. The more road miners drive in slow lanes and contribute to the community, the more QRS tokens they earn. Additionally, drivers who use fast lanes will have to pay for the service from their own funds or from previously accumulated QRS tokens [as shown in Figure 2].

In addition to the proof of traffic algorithm, various decision-making algorithms will also be developed to automate specific processes in QRS. For example, these algorithms can determine whether QRS can be used in a particular location or activate a service in an area where the number of active users exceeds a "critical

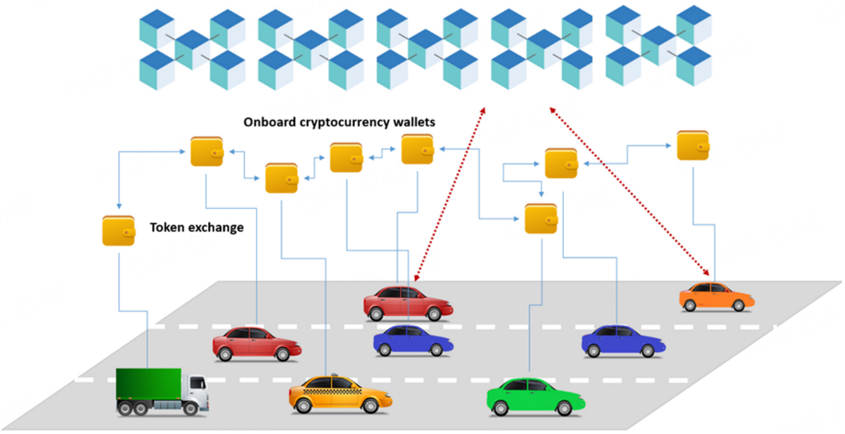


Figure 2. The service of getting a free passage[4].

mass". By automating these processes, QRS can function more efficiently without requiring human involvement.

The QRS could be a pioneer in the trend of decentralized, self-directed systems or DAOs. Alongside similar services, it may alter the sharing economy of urban transport and become a leading model for social transport in the future[17-20].

Another potential use case for blockchain technology is a distributed transport service that operates using proprietary cryptocurrencies. This network would be constructed on the Bitcoin blockchain and adopt a new concept called "proof of motion" to mine coins.

The transport network is open to everyone, and becoming a miner is as easy as downloading the app on your smartphone and turning on the GPS when traveling at a speed of over 20 km/h. Alternatively, users can buy coins by contributing to the software or design of the app or by inviting others to join the network. This approach allows early adopters to establish the foundation for the network, with rewards paid in proprietary cryptocurrencies issued through mechanisms such as Mastercoin, Counterparty, or similar protocols[21]. As the number of participants grows, the entire system can run autonomously. This concept opens up new possibilities for the future of transportation and creates a decentralized ecosystem for mobility services.

However, in the pursuit of profit, various sources may generate false information about a particular service request, making it difficult to identify reliable information from multiple sources in a reliable QRS platform. To address these issues, we propose the use of QRR-chain (Quick Road Reputation chain), a reputation-based blockchain platform for crowdsourcing. The QRR-chain uses advanced blockchain platforms such as Cosmos, Polkadot, and Avalanche, which are decentralized, transparent, and irreversible, to manage crowdsourcing activities. Each crowdsourced exchange is recorded as a blockchain transaction governed by predefined criteria specified in a smart contract that ensures a secure trading environment and prevents malicious user behavior. Furthermore, the QRR-chain incorporates a reputation model created using the smart contract, which improves the quality of crowdsourcing services. This model evaluates the level of trust of both service providers and consumers based on their actions, including the quality of missions and payment fulfillment. A quantitative reputation score helps users select potential service providers and allows providers to choose appropriate missions. Each reputation score of a service provider is updated and recorded on the blockchain after each crowdsourced transaction, with a warning issued for any provider with a negative reputation. QRR-chain is the first solution to combine blockchain-based crowdsourcing management with reputation evaluation, and it addresses the aforementioned issues while offering a secure and efficient platform for VCS scenarios.

In the future, there are plans to develop a reputation-based blockchain crowdsourcing system called the QRR-chain to address the issues mentioned above. The QRR-chain will record each traffic lane seat purchase as a transaction on the blockchain, which will prevent malicious behavior and ensure the security of the trading environment. Furthermore, there are plans to establish a reputation model using smart contracts to evaluate the trustworthiness of QRS users based on their activities, such as mission quality and payment transactions. Quantifying reputation can help consumers choose potential vendors and vendors select missions. The QRR-chain will merge the traffic flow seat selling and reputation evaluation components. Queuing theory will be utilized to optimize the configuration and performance of the QRR-chain.

In the future, there are plans to develop a reputation-based blockchain crowdsourcing system called the QRR-chain to address the issues mentioned above. The QRR-chain will record each traffic lane seat purchase as a transaction on the blockchain, which will prevent malicious behavior and ensure the security of the trading environment. Furthermore, there are plans to establish a reputation model using smart contracts to evaluate the trustworthiness of QRS users based on their activities, such as mission quality and payment transactions. Quantifying reputation can help consumers choose potential vendors and vendors select missions. The QRR-chain will merge the traffic flow seat selling and reputation evaluation components. Queuing theory will be utilized to optimize the configuration and performance of the QRR-chain.

The QRR-chain is a new blockchain-based platform that aims to enhance trust in the crowdsourcing system. Its smart contracts will define all trading rules, and every seat purchase in the traffic flow will be registered as a QRR-chain transaction.

To elevate the quality of crowdsourcing services, the QRR-chain will integrate a novel reputation model. This model will thwart malicious or irresponsible behavior by diminishing the reputation of unscrupulous service providers or consumers. Additionally, the model will leverage mission-based reputation scores to detect untrustworthy service providers, helping consumers find high-quality ones.

To optimize the performance of the QRR-chain platform, an efficient configuration scheme will be developed using queuing theory. The effectiveness of the proposed scheme will be evaluated based on key performance indicators such as transaction confirmation time and throughput.

The convergence of two game-changing ideas, decentralized transport and cryptocurrencies, is inevitable. Our aim is to leverage the technology behind cryptocurrencies to achieve the necessary critical mass of users and create a truly decentralized transport system in cities[22,23]. The proceeds from the coin sales will be used to enhance and refine the transport system, and the coins will be redeemable for rides once the network is operational.

3. RESULTS

Blockchain is a revolutionary technology that enables the transfer of value between peers without requiring a central intermediary. It achieves this through the use of cryptographic hashing functions, consensus protocols, and decentralized data storage to guarantee secure, decentralized trust, immutability, and transparency of transactions. By pairing blockchain technology with smart contracts, businesses can automate rules and processes in a reliable, efficient, and trustworthy manner. A smart contract is a self-executing code that runs on a blockchain network. It translates predefined rules between participating organizations into functions that establish trust between them.

Let us explore the main opportunities and potential ideas for blockchain technology in the transport sector.

3.1. Decentralized autonomous transport systems

Decentralized autonomous transport systems embody the core elements of blockchain. P2P networks, which rely on a consensus of distributed coordination and economic incentives, are a natural way to model the complex transport system. In this system, each computational node, such as IoT devices, vehicles, or other objects with computational power, can act as an autonomous agent[4].

These nodes can be linked to a common network and interact with each other through various types of blockchain-based Dapps, resulting in a DAO that is subject to specific requirements and tasks. Eventually, this can lead to the creation of a decentralized autonomous system and even a DAS[24].

To advance in this area, it is important to investigate both the micro-level of individual behaviors and interactions among autonomous system agents and the macro-level of modeling, self-organization, self-improvement, and self-adjustment of systems.

3.2. Create an incentivization framework for crowdsourcing

In blockchain-based systems, distributed consensus competition can already be viewed as a crowdsourcing task where numerous nodes contribute their computing power to validate blockchain data[8]. Since these nodes are individual agents, the incentivization and crowdsourcing mechanisms must ensure that individual actions of a node in the pursuit of profit maximization align with the system-wide objective of safeguarding the protection and dependability of the system. The blockchain technology applied can aggregate all available computational resources in an ITS to improve ITS services and enhance decision-making speed.

3.3. Implementation of software-defined trust in ITS systems

Trust is a crucial aspect of creating a decentralized ITS using blockchain technology, which can facilitate services such as commercial transportation, public transport, and electronic payments in transport[9]. This type of trust is established through code and the verification of the majority of process participants, potentially reducing structural complexity and addressing social problems. For instance, P2P trust can enable the direct sale and registration of used cars through blockchain applications without relying on centralized authorities or platforms. Further research is needed to explore the underlying rationale for trust and fiduciary formation in this direction.

3.4. Leveraging smart contracts for intelligent transport systems

Smart contracts act as a catalyst for blockchain technology by enabling the dynamic execution of processes based on pre-set rules and conditions. Through various algorithms such as machine learning and big data analysis, smart contracts can enhance the intelligence of ITS and its applications, resulting in a more efficient and effective transport ecosystem. They also decrease social complexity by minimizing the involvement of humans in executing the contracts. Thus, it is important to study the creation and deployment of smart contracts in ITS and the management and monitoring of the system built upon them.

3.5. Ensuring data security and privacy protection

Despite its high reliability and security, blockchain needs to have a stronger encryption framework when used in an ITS that involves a large number of devices to safeguard against attacks.

Researchers have proposed the concept of PTMS[24], which optimizes the real transportation system by enabling parallel interactions with artificial or virtual counterparts. Blockchain is a secure and reliable architecture for PTMS and can be considered an important step toward its implementation.

A blockchain-based PTMS can include all physical objects, such as IoT devices, vehicles, and assets, which can be easily digitized and registered online in a "blockchain of things." The transmission of big data in cyberspace can also be integrated into blockchain. It is also possible to create one or more virtual transport systems in the code space of smart contracts using the Ethereum platform, which provides programmable scripts to support modeling and computation. Therefore, it is crucial to ensure the security and privacy of data in such systems to maintain the trust of the participants of the system[25].

The co-development of real and virtual transport systems enables the design and execution of computational experiments to evaluate and verify specific behaviors, scenarios, and strategies in ITS. These experiments can take the form of 'What If' scenarios and simulation output based on predefined 'If Then' rules. The optimal solution is derived from a large number of computational experiments, which are then applied to the actual transport systems. This cyclic process is repeated until the actual transport system approximates its optimal virtual counterparts. This approach offers an efficient means of evaluating traffic conditions and improving the overall performance of ITS[26].

4. DISCUSSION

In order to use blockchain as a reliable ledger for transactional data, a user needs to be able to verify on the blockchain whether a transaction has been made to their address or electronic wallet, where cryptocurrencies are stored. If this information was stored on a single computer or server, the availability of the data would be dependent on that particular device. In the event that the device goes down or fails, the data becomes inaccessible. However, this is where blockchain technology proves useful. The current state of the blockchain is downloaded, synchronized, and made available to a large number of computers around the world, known as nodes. These nodes operate in a P2P network, working together to keep the blockchain secure and up-to-date. Each node stores a complete and updated version of the blockchain, and when a new block is added, all nodes update their respective blockchains.

Using a P2P network to store the blockchain has several advantages over relying on a single node. First, information is available at any time because there are multiple nodes.

Second, it makes the blockchain highly secure because the data is distributed across many computers. In order to change the data, all the nodes must be updated simultaneously, which is practically impossible. This makes the data tamper-proof and immutable.

Third, the decentralized nature of the blockchain means that an attacker would need to gain control of thousands of nodes at the same time to manipulate the data. This would require an enormous amount of processing power, making such an attack highly unlikely.

Finally, once data is stored on the blockchain, it cannot be edited or deleted, ensuring the integrity and authenticity of the recorded transactions.

The consensus mechanism is a critical component of a P2P network, which is responsible for validating transactions and blocks. When a new block is added to the blockchain, it must be accepted by all nodes in the network to be considered valid.

In contrast, in centralized systems, a single administrator controls the database and makes decisions on which files to keep and how to update them. Each transaction is approved, edited, or deleted by a single person or computer, and other users in the system do not have real-time visibility into the changes being made. This lack of transparency makes the system vulnerable to fraud, misuse, and errors.

Decentralized public resources, such as blockchain, do not have administrators. No single node can approve, edit, or delete any transaction without the consensus of all blockchain nodes. This presents a challenge for achieving real-time consensus and updating the state of the public ledger. The solution is found in mining, which is a mechanism that checks and updates transactions in real-time to maintain consensus among all nodes.

In conclusion, implementing information technologies is generally cheaper and easier than building infrastructure. Several available technologies, such as GPS and intelligent vehicles, can be utilized to improve performance[18,19].

The adoption of information technology can enhance facility performance by reducing transaction delays, improving productivity, and minimizing check processing time. It can also increase transit speed and decrease operational costs, thereby improving the overall capabilities of ITS. As the development of artificial intelligence and 5G wireless connectivity continues to progress, intelligent traffic is expected to become a comprehensive solution that involves numerous IoT devices that connect and communicate with one another. This will enable the prediction of traffic conditions and the anticipation of optimal solutions. Moreover, due to significant advancements in intelligent software development, a more sophisticated system has been implemented.

Integrating ITS with blockchain technology is expected to make the system even more powerful and efficient. This integration can help to mitigate the influence of human factors and external malicious attacks, allowing vehicles to communicate and self-organize based on traffic conditions through an intelligent contract in a blockchain.

In the coming years, there is an expectation of developing machine learning algorithms for predicting traffic conditions and determining traffic intensity. Moreover, a consensus block algorithm is being developed to improve tolerance and transaction rates in the real-time system.

The integration of machine learning systems in the process of data retrieval, loading, and conversion is not just about creating a more efficient system, but it also has the potential to create a self-improving and self-managing service that is capable of continuous learning. The results of sensory data and extensive data analysis of road modules will be processed and added to the knowledge base, leading to subsequent quantitative analysis that is more accurate than the previous one. This implies that road modules, which are primarily responsible for accumulating large data streams, can adapt their data collection strategies according to the results of statistical analysis.

The impact of blockchain technology is extending beyond its initial use in cryptocurrency and is poised to revolutionize a wide range of industries. Among these is the logistics sector, where blockchain holds the potential to enhance the efficiency of business operations and enable the development of new services and business models.

In conclusion, the integration of blockchain technology in ITS within multi-agent systems holds great promise for revolutionizing the transportation industry by addressing key challenges faced by traditional ITS solutions. By providing secure and decentralized platforms for exchanging data, managing transportation processes, and facilitating payment systems, blockchain technology has the potential to create more efficient, secure, and transparent transportation networks. Further research is needed to fully realize the potential of blockchain technology in ITS within multi-agent systems and to explore new applications and use cases in this field.

5. CONCLUSIONS

In conclusion, integrating blockchain technology into the transport sector has the potential to revolutionize the industry by addressing key challenges and unlocking new opportunities. Decentralized autonomous systems of blockchain can model complex transport networks, while incentivization frameworks can enhance crowdsourcing efforts. Software-defined trust and smart contracts offer secure and efficient transactions, minimizing reliance on centralized authorities. Data security and privacy protection are crucial considerations, especially in parallel transportation management systems. By integrating real and virtual transport systems, computational experiments can optimize traffic conditions and improve overall performance. The adoption of blockchain technology can enhance efficiency, security, and transparency in transportation networks. Further research is needed to fully explore and exploit the potential of blockchain in multi-agent systems within the transport sector, unlocking its full benefits and discovering new applications.

DECLARATIONS

Authors' contributions

Contributed significantly to the development of the concept of blockchain integration with ITS: Eremina L, Mamoiko A, Aohua G
Working out the Quick Road System (QRS): Eremina L, Mamoiko A, Aohua G
Developed the concept of performance measurement indicators and provided technical support: Eremina L, Mamoiko A, Aohua G

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Wei D, Zhang J, Shojafar M, Kumari S, Xi N, Ma J. Privacy-aware multiagent deep reinforcement learning for task offloading in VANET. *IEEE Trans Intell Transport Syst* 2022:1-15. DOI
2. Cheng H, Shojafar M, Alazab M, Tafazolli R, Liu Y. PPVF: privacy-preserving protocol for vehicle feedback in cloud-assisted VANET. *IEEE Trans Intell Transport Syst* 2022;23:9391-403. DOI
3. Zhou Z, Shojafar M, Alazab M, Abawajy J, Li F. AFED-EF: an energy-efficient VM allocation algorithm for IoT applications in a cloud data center. *IEEE Trans on Green Commun Netw* 2021;5:658-69. DOI
4. Eremina L, Mamoiko A, Bingzhang L, Kudriavtcev S, Murgul V. Use of blockchain technology in planning and management of transport systems. *E3S Web Conf* 2020;157:04014. DOI
5. Zhang J, Wang F, Wang K, Lin W, Xu X, Chen C. Data-driven intelligent transportation systems: a survey. *IEEE Trans Intell Transport Syst* 2011;12:1624-39. DOI
6. Wang F. Parallel control and management for intelligent transportation systems: concepts, architectures, and applications. *IEEE Trans Intell Transport Syst* 2010;11:630-8. DOI
7. Li L, Li X, Li Z, Zeng DD, Scherer WT. A bibliographic analysis of the IEEE transactions on intelligent transportation systems literature. *IEEE Trans Intell Transport Syst* 2010;11:251-5. DOI
8. Swan M. Blockchain: blueprint for a new economy. O'Reilly Media; Inc., 2015.
9. Antonopoulos AM. Mastering bitcoin. O'Reilly Media; Inc., 2014.
10. Yuan Y, Wang FY. Blockchain: the state of the art and future trends. *cta Automatica Sinica* 2016;42:481-94. DOI
11. Kong Q, Li L, Yan B, Lin S, Zhu F, Xiong G. Developing parallel control and management for urban traffic systems. *IEEE Intell Syst* 2013;28:66-9. DOI
12. Wang FY. Parallel system methods for management and control of complex systems. Control and Decision 2004;19:485-9. (in Chinese). DOI
13. Wang FY, Dai RW, Zhang SY, et al. A complex system approach for studying sustainable and integrated development of metropolitan transportation, logistics and ecosystems. *Complex Systems and Complexity Science* 2004;1:60-9. Available from: http://en.cnki.com.cn/Article_en/CJFDTOTAL-FZXT200402008.htm [Last accessed on 23 May 2023].
14. S. Nakamoto. Bitcoin: a peer-to-peer electronic cash system. Available from: http://bitcoin.org/bitcoin.pdf [Last accessed on 23 May 2023].
15. Ferrag MA, Ferrag M, Mukherjee M, et al. Blockchain technologies for the internet of things: research issues and challenges. *IEEE Internet of Things Journal* 2019;6:2188-204. DOI
16. Lakshman TV, Agrawala AK. Efficient decentralized consensus protocols. *IIEEE Trans Software Eng* 1986;SE-12:600-7. DOI
17. Kitahara F, Kera K, Bekki K. Autonomous decentralized traffic management system. In: Proceedings of International Workshop on Autonomous Decentralized Systems, 2000 Sep 87-91. DOI
18. Mori K. Autonomous decentralized systems technologies and their application to a train transport operation system. In: Winter VL, Bhattacharya S, editors. High integrity software. Boston: Springer US; 2001. pp. 89-111. DOI
19. Wen D, Yuan Y, Li X. Artificial societies, computational experiments, and parallel systems: an investigation on a computational theory for complex socioeconomic systems. *IEEE Trans Serv Comput* 2013;6:177-85. DOI
20. Lv YS, OU Y, Tang SM, Zhu FH, Zhao HX. Computational experiements of evaluating road network traffic conditions based on artificial transportation systems. *Journal of Jilin University* 2000;39:87-90. Available from: http://qikan.cqvip.com/Qikan/Article/Detail?id=1003866541 [Last accessed on 23 May 2023] (in chinese).
21. Kim HM, Laskowski M. Toward an ontology-driven blockchain design for supply-chain provenance. *Intell Syst Account Financ Manag* 2018;25:18-27. DOI
22. Crosby M, Pattanayak P, Verma S, Kalyanaraman V. Blockchain technology: beyond bitcoin. *Appl Innov* 2016;2:6-10. Available from: https://scet.berkeley.edu/wp-content/uploads/AIR-2016-Blockchain.pdf [Last accessed on 31 May 2023].
23. Morley HR. Industry skeptical of pace of logistics tech sdoption. JOC: New York, NY, USA, 2017.Available from: https://www.joc.com/article/industry-skeptical-pace-logistics-tech-adoption_20170620.html [Last accessed on 31 May 2023]
24. Lehmacher W. Why blockchain should be global trade's next port of call. World Economic Forum: Geneva, Switzerland, 2017.Available from: https://www.weforum.org/agenda/2017/05/blockchain-ports-global-trades/ [Last accessed on 31 May 2023]
25. Yang C, Lirn T. Revisiting the resource-based view on logistics performance in the shipping industry. *IJPDLM* 2017;47:884-905. DOI
26. Sun L, Yang Q, Chen X, Chen Z. RC-chain: reputation-based crowdsourcing blockchain for vehicular networks. *J Netw Comput Appl* 2021;176:102956. DOI

Intelligence & Robotics

Research Article

Open Access

Reinforcement learning with parameterized action space and sparse reward for UAV navigation

Shiying Feng[1], Xiaofeng Li[2], Lu Ren[2], Shuiqing Xu[3]

[1]Institute of Physical Science and Information Technology, Anhui University, Hefei 230601, Anhui, China.
[2]School of Artificial Intelligence, Anhui University, Hefei 230601, Anhui, China.
[3]School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, Anhui, China.

Correspondence to: Lu Ren, School of Artificial Intelligence, Anhui University, Hefei 230601, Anhui, China. E-mail: penny_lu@ahu.edu.cn

How to cite this article: Feng S, Li X, Ren L, Xu S. Reinforcement learning with parameterized action space and sparse reward for UAV navigation. *Intell Robot* 2023;3:161-75. http://dx.doi.org/10.20517/ir.2023.10

Received: 23 Feb 2023 **First Decision:** 27 Apr 2023 **Revised:** 11 May 2023 **Accepted:** 18 May 2023 **Published:** 27 Jun 2023

Academic Editor: Simon X. Yang **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

Autonomous navigation of unmanned aerial vehicles (UAVs) is widely used in building rescue systems. As the complexity of the task increases, traditional methods based on environment models are hard to apply. In this paper, a reinforcement learning (RL) algorithm is proposed to solve the UAV navigation problem. The UAV navigation task is modeled as a Markov Decision Process (MDP) with parameterized actions. In addition, the sparse reward problem is also taken into account. To address these issues, we develop the HER-MPDQN by combining Multi-Pass Deep Q-Network (MP-DQN) and Hindsight Experience Replay (HER). Two UAV navigation simulation environments with progressive difficulty are constructed to evaluate our method. The results show that HER-MPDQN outperforms other baselines in relatively simple tasks. Especially for complex tasks involving relay operations, only our method can achieve satisfactory performance.

Keywords: Deep reinforcement learning, parameterized action space, sparse reward

1. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs) have been widely used in emergency rescue fields within the framework of Industry 4.0[1–3]. The key technology behind these applications is UAV attitude control[4,5] and autonomous navigation[6]. Traditional approaches to address navigation challenges using modeling techniques[7–9] and simultaneously localization-and-mapping techniques[10–12]. While these methods have demonstrated satisfactory performance, they heavily rely on prior knowledge of the environment, limiting their applicability in complex and dynamic navigation scenarios.

Deep reinforcement learning (DRL) has developed rapidly and achieved remarkable success in recent years. DRL aims at deriving a policy by maximizing a long-term cumulated reward of a Markov decision process (MDP). MDP characterizes a process in which an agent at some state takes action, transits to another state, and obtains rewards. For continuous decision problems, Silver *et al*.[13] develop a hybrid DRL system that defeated a human world champion in Go. In robotics, DRL successfully solves optimal control problems[14,15]. For this reason, researchers turn their attention to reinforcement learning (RL)-based methods. For instance, Yan *et al*.[16] propose an improved Q-learning algorithm to handle the path planning problem without prior knowledge. However, it is difficult to deal with tasks in the real world because only discrete action sets are considered. In contrast, Bouhamed *et al*.[17] propose a path planning framework for handling continuous actions based on DRL.

Recent work on UAV navigation based on DRL has been successful, but there are still some potential issues that have been overlooked. On the one hand, the behavior of UAVs is usually defined as single-type actions in existing studies. However, UAVs often need both discrete decision-making and continuous control when performing tasks. In[18], Masson *et al*. call the action space consisting of two control signals as parameterized action space. Hausknecht and Stone[19] deal with this problem by relaxing the parameterized action space into a continuous one. However, they do not exploit the structural information of this action space. For that matter, Xiong *et al*.[20] propose Parametrized Deep Q-Networks (P-DQN), which can directly learn from parameterized actions. Since P-DQN couples all continuous parameters to each discrete action, the prediction of the Q network is influenced by unrelated parameters. In[21], Bester *et al*. develop the Multi-Pass Deep Q-Network (MP-DQN) algorithm by changing the Q network architecture of P-DQN to eliminate this effect.

On the other hand, the studies above train agents with custom reward functions to speed up network convergence[22,23]. However, overly complex reward signals may cause the agent to get stuck in local optima. Moreover, it is hard to give correct rewards according to whether the UAV is moving away from the target due to the presence of obstacles. This problem can be avoided by adopting more general sparse reward schemes. While defining sparse reward is simple, the potential learning problem is much harder to solve (i.e., lack of intermediate rewards hinders the learning of agents). Numerous studies propose curiosity-based approaches to encourage agents to explore previously rare states[24,25]. These methods introduce additional fitting models of environmental dynamics to measure curiosity. However, the model will reduce efficiency when the dynamics are unpredictable. Andrychowicz *et al*.[26] develop the Hindsight Experience Replay (HER) algorithm by introducing the goal mechanism. The effectiveness of HER has been demonstrated in many robotics applications with sparse reward[27,28].

Parameterized action space and sparse reward inspire us to rethink new forms of RL problems. In this paper, our contributions can be summarized as follows:

1) We model UAV navigation as a parameterized action MDP to better suit the task requirements. At the same time, sparse rewards are considered to improve the generality of the algorithm in various tasks. To handle these challenges, an off-policy algorithm called HER-MPDQN is developed by incorporating HER with MP-DQN.
2) To address the issue of extended invalid experiences encountered in traditional methods, we propose a goal-switching mechanism. This mechanism effectively reduces the invalid expansion for experience and improves the rationality of the expansion experience.
3) We compare our algorithm with baselines in experiments with high randomness and varying difficulty. Experimental results show that our method is capable of learning better policies in solving navigation tasks

with sparse rewards. It can be successfully generalized to any position in space and significantly outperforms existing RL algorithms.

The rest of this article is outlined below. PAMDP and the sparse reward problem are described in Section 2. In Section 3, the navigation problem is modeled as a PAMDP and uses a sparse reward scheme. Our proposal to address the problem is elaborated in Section 4. In Section 5, we compare HER-MPDQN with other baselines in two UAV navigation simulation environments, followed by our discussion in Section 6 and conclusions in Section 7.

2. BACKGROUND

In this section, we briefly introduce MDP and PAMDP, followed by the issue of sparse reward.

2.1. MDP and PAMDP

MDP is built based on a set of interactive objects, namely agents and environments. MDP consists of a state space S, an action space A, a state transition probability distribution $P(s_{t+1}|s_t, a_t)$, and a reward function $R : r_t = r(s_t, a_t)$. Where P satisfies the Markov property, and R represents the immediate reward r_t obtained by executing an action a_t in a given state s_t. RL, which learns the optimal policy based on trial and error, provides a way to solve the MDP problem. The policies learned based on RL are divided into deterministic policy and stochastic policy. For discrete action spaces, the deterministic policy is expressed as $a_t = \pi(s_t)$, which means that a certain action a_t can be obtained for a given state s_t. The stochastic policy is denoted as $a_t \sim \mu(\cdot|s_t)$, which means to select an action a_t from the probability distribution after a given state s_t. If the action space is continuous, the corresponding policies above will be parameterized as functions $a_t = \pi(s_t, \theta)$ and $a_t \sim \mu(\cdot|s_t, \theta)$, where θ refers to the parameters of the function. To uniform representation, we all implicitly mean that policy π (or μ) is a function of θ, and all the gradients are with respect to θ in the following contexts. RL involves estimating state-value functions $V(s)$ and action-value functions $Q(s, a)$. Taking deterministic policy as an example, the state-value function is defined as

$$V_\pi(s_t) = \mathbb{E}_\pi\left[\sum_{m=0}^{T} \gamma^m r(s_{t+m}, a_{t+m})|s_t)\right], \tag{1}$$

where $\gamma \in [0, 1]$ is a discount factor, and the action-value function is defined as

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi\left[\sum_{m=0}^{T} \gamma^m r(s_{t+m}, a_{t+m})|s_t, a_t)\right]. \tag{2}$$

The agent learns an optimal policy by maximizing the expected discounted reward (target function) as follows

$$J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\right]. \tag{3}$$

If π is a stochastic policy, the corresponding state-value function, action-value function, and target function can be obtained by replacing $a_t = \pi(s_t)$ by $a_t \sim \mu(\cdot|s_t)$ in (1), (2), and (3).

PAMDP is an extension of MDP on the action space, allowing decision-making using parameterized actions. Parameterized actions flexibly integrate discrete actions and continuous actions and provide richer expressiveness. In tasks such as UAV navigation, which demand precise parameter control, using parameterized actions enables finer-grained control. Moreover, from an interpretability standpoint, the structural characteristics of parameterized actions make the decision-making process of agents more understandable and explainable. The action space of PAMDP can be expressed as $\mathcal{H} = \{(k, x_k)\,|x_k \in \mathcal{X}_k\}\ \ for\ all\ k \in K$, where $k = \{1, ..., K\}$. K denotes the number of discrete actions. k refers to a specific discrete action (e.g., $k = 1$ means movement and

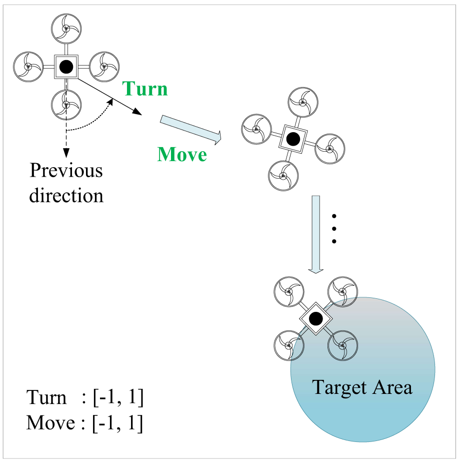


Figure 1. The direct navigation task. The UAV only needs to fly from the initial point to the target area.

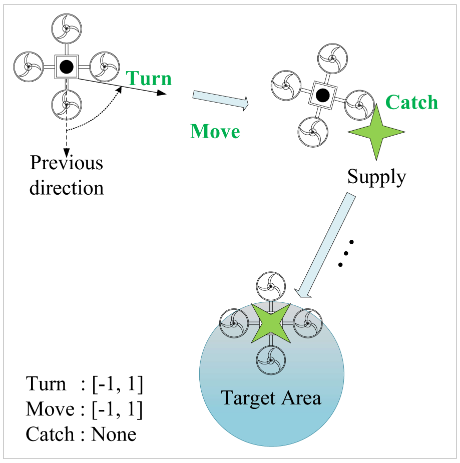


Figure 2. The relay navigation task. The UAV needs to operate on "catching supply" during the navigation process.

$k = 2$ means turning). x_k represents the continuous parameter (e.g., acceleration or angle) associated with the discrete action k. $\mathcal{X}_k$ is the set of all continuous parameters. In the PAMDP, the agent first selects a discrete action k, then obtains the corresponding x_k from $\mathcal{X}$ according to k, and finally executes the action (k, x_k). Therefore, PAMDP has subtle differences in the interaction process compared to standard MDP. Assuming that at step t, PAMDP is in state s_t, the agent executes an action by policy π: $s_t \rightarrow (k_t, x_{k_t})$ and receives an immediate reward $r(s_t, k_t, x_{k_t})$ and the next state $s_{t+1} \sim P(s_{t+1}|s_t, k_t, x_{k_t})$. The target function of the agent becomes as follows

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r(s_t, k_t, x_{k_t}) \right]. \tag{4}$$

2.2. Sparse reward problems

The sparse reward scheme is a reward mechanism that uses only binary values to indicate whether a task is eventually completed. Specifically, the agent can get a positive reward when completing the task and a negative reward during exploration. Although this mechanism reduces the effort required for human design, it brings potential learning problems. The lack of effective rewards prevents the agent from judging the pros and cons of its behavior and thus cannot optimize the policy π correctly. Under the influence of the sparse reward problem, the agent learns slowly or even fails to learn. Solving the harmful interference brought by the sparse reward scheme is one of the focuses of this paper.

3. PROBLEM FORMULATION

In this section, a UAV navigation task from an initial to a target position is first formulated. The UAV will learn a policy by mapping internal state information to action sequences. Then the PAMDP modeling process with sparse reward is described in detail.

3.1. UAV navigation tasks

In this paper, UAV navigation tasks are divided into the direct navigation task [Figure 1] and the relay navigation task [Figure 2]. The distinction between these tasks lies in the presence of intermediate operations that the UAV needs to perform. As a result, the flight requirements for UAVs in relay navigation tasks are more demanding. Nevertheless, the simple task is also considered to verify the effectiveness and generality of the algorithm.

Typically, the position, direction, and motion of the UAV are determined by both the earth and the body coordinate frame. The earth coordinate frame is used to describe the position and direction of the UAV, denoted as $\phi = [x, y, z, \theta_x, \theta_y, \theta_z]$. The linear and angular velocities of the UAV are denoted as $\psi = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]$ in the body coordinate frame.

For simplicity, the UAV is assumed to fly at a fixed altitude, i.e., the motion of the UAV is constrained within the x-y plane. Moreover, the steering action of the UAV takes effect immediately because the momentum is ignored. As a result, the vector describing the motion information of the UAV is simplified to $\xi = [x, y, v, \theta]$, and the motion formula is expressed as follows

$$\begin{aligned} \theta_{t+1} &= \theta_t + \Delta\theta \\ v_{t+1} &= v_t + \Delta a \\ x_{t+1} &= x_t + v_{t+1}\cos(\theta_{t+1}) \\ y_{t+1} &= y_t + v_{t+1}\sin(\theta_{t+1}) \end{aligned} \tag{5}$$

where $\Delta\theta$ is the steering signal, Δa is the acceleration signal, v is the speed of the UAV, and θ is the direction angle between the UAV and the target.

3.2. State representation specification

State representation is essential for the agent to perceive the surrounding environment and reach the target position. In our setting, the state vector is reduced as much as possible to avoid interference from irrelevant information and speed up training. Specifically, the observable information of the UAV has three sources. The first is the internal state of UAVs, which is represented in terms of position $[x, y]$, velocity v, and direction θ. The second is the relationship between UAVs and the environment. Since obstacles are not considered, the number of actions the UAV performs is used to be the unique representation, which is recorded as n_{step}. Taking n_{step} as the state can encourage the UAV to complete the task with fewer steps. The last is the relationship information between the UAV and the target. Also, to simplify the representation, the distance between the UAV and the target is only used to describe this relationship, denoted as d_{target}. By combining the three kinds of information, the final form of the state vector becomes: $s = [x, y, v, \theta, d_{target}, n_{step}]$. For relay navigation tasks, UAVs need additional information about supplies. The location information of the supply can be directly obtained from the simulation environment. In this paper, the distance d_{supply} between the UAV and the supply is included as one of the state representations to fulfill this requirement. As a result, the state vector in the relay task is modified as follows: $s = [x, y, v, \theta, d_{target}, d_{supply}, n_{step}]$.

3.3. Parameterized action design

Considering that the flight altitude of the UAV has been set as a constant above, its movement in the vertical direction can be ignored. For the direct navigation task, the optional discrete actions of the UAV are defined in the discrete set $\mathbb{D} = \{k_1, k_2\}$. Where k_1 represents the "MOVE" behavior and k_2 represents the "TURN" behavior. In addition, the parameter set $\mathbb{C} = \{(c_1), (c_2)\}$ defines the continuous parameters corresponding to each action in $\mathbb{D}$, where c_1 represents the acceleration value and c_2 represents the rotation angle. When the UAV acts, it not only needs to select the discrete action but also needs to determine the corresponding parameter values. To sum up, the parameterized action space of UAV is represented as $\mathcal{H} = \{(k_1, (c_2)), (k_2, (c_2)\}$. The value range of c_1 and c_2 are scaled to $[-1 \sim 1]$. Since the UAV needs to perform intermediate operations for the relay navigation task, a new discrete action k_3 will be added to $\mathbb{D}$ to represent the "CATCH" behavior. It should be pointed out that action k_3 does not set continuous parameters. This is because we focus more on learning the navigation policy rather than the specific control process. When the UAV gets close enough to the supply, it performs the "CATCH" action, allowing the supply to move with itself. However, when the UAV is far away from the supply, it will not affect the supply when it performs the "CATCH" action. In this case, the UAV advances one step based on the current speed and angle. The primary goal of the UAV is to transport the supply to the target area.

3.4. Sparse reward function

Although sparse reward setting is simple and universal, different tasks need to set corresponding reward functions, which will bring different degrees of sparsity. The direct navigation task aims to get the UAV to reach the target area, which is a single-stage task with a relatively small reward sparsity. In the relay navigation task that contains supplies, an effective reward can be obtained only when the UAV finds supplies and carries them to the target area. This task includes relay operations which significantly increase the reward sparsity. Existing baselines are not effective at handling the relay task. This phenomenon and the solution are described in detail in Section 4.2. According to the above task requirements, the reward function of the direct navigation task is defined as

$$R = \begin{cases} 0, & if\ UAV\ in\ target\ area \\ -1, & otherwise \end{cases} \tag{6}$$

and the reward function of the relay navigation task is defined as

$$R = \begin{cases} 0, & if\ supply\ in\ target\ area \\ -1, & otherwise\ . \end{cases} \tag{7}$$

4. DEEP REINFORCEMENT LEARNING AGENT

To solve the navigation task presented in Section 3, the MPDQN algorithm is considered first, which has been proven effective in PAMDP. Besides, we introduce how to extend HER to solve the sparse reward problem in the relay task, followed by the implementation and training process of HER-MPDQN.

4.1. Multi-Pass Deep Q-Network (MP-DQN)

MP-DQN is an off-policy RL algorithm that deals with parameterized action space. In MP-DQN, the goal of the agent is to optimize the policy $\pi : S \rightarrow A$, which maximizes the long-term cumulative discounted rewards:

$$R_t = \sum_{i=t}^{T} \gamma^{t-i} r(s_t, k_t, x_{k_t}). \tag{8}$$

Due to the integration of DQN[29] and DDPG[30] algorithms, MP-DQN has a network architecture similar to the Actor-Critic architecture. In specific, the MPDQN agent has an actor-parameter network π_{θ_x} (similar to Actor) and a Q-value network Q_{θ_Q} (similar to Critic). Where θ_x and θ_Q are parameters of the network. The discrete action policy is implicitly learned when approximating the Q-value function.

For the actor-parameter network, the MP-DQN agent learns a deterministic mapping policy $\pi_{\theta_x}(s)$ from state to continuous parameters vector as follows

$$\mathcal{X}_k^t = \pi_{\theta_x}(s_t) \tag{9}$$

where $\mathcal{X}_k^t = [x_1^t, x_2^t, ..., x_K^t]$ contains all continuous parameters corresponding to the discrete actions k. In order to decouple discrete actions and irrelevant continuous parameters, the $\mathcal{X}_k^t$ becomes

$$\mathcal{X}_k^t = \begin{bmatrix} x_1^t & 0 & \cdots & 0 \\ 0 & x_2^t & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & x_K^t \end{bmatrix} = \begin{bmatrix} x_{e_1}^t \\ x_{e_2}^t \\ \vdots \\ x_{e_K}^t \end{bmatrix}$$

Each row of the matrix is fed into the Q-value network separately, thereby eliminating the influence of irrelevant parameters on the estimation of the Q-value function.

For the Q-value network, MP-DQN evaluates actions using an action-value function similar to DDPG. Since MPDQN needs to optimize both discrete actions and continuous parameters, the Bellman equation becomes

$$Q_{\theta_Q}(s_t, k_t, x_{k_t}) = \mathop{\mathbb{E}}_{r_t, s_{t+1}} \left[r_t + \gamma \max_{k \in [K]} \sup_{x_k \in \mathcal{X}_k} Q_{\theta_Q}(s_{t+1}, k, x_k) | s_t, k_t, x_{k_t} \right], \tag{10}$$

where k_t is the discrete action selected at time t, and x_{k_t} is the associated continuous parameter. To avoid taking supremum over continuous space $\mathcal{X}_k$, equation (10) is rewritten as:

$$Q_{\theta_Q}(s_t, k_t, x_{k_t}) = \mathop{\mathbb{E}}_{r_t, s_{t+1}} \left[r_t + \gamma \max_{k \in [K]} Q_{\theta_Q}(s_{t+1}, k, \pi_{\theta_x}(s_{t+1})) | s_t \right], \tag{11}$$

where $\pi_{\theta_x} : S \rightarrow \mathcal{X}_k$ represents the mapping relationship in equation (9). This means that approximating the Q-value function needs to fix θ_Q first and find θ_x such that

$$Q_{\theta_Q}(s, k, \pi_{\theta_x}(s)) \approx \sup_{x_k \in \mathcal{X}_k} Q_{\theta_Q}(s, k, x_k) \qquad for\ each\ k \in [K] \tag{12}$$

Then, similar to DQN, θ_Q is estimated by minimizing the mean-squared Bellman error. The loss function of the Q-value network is:

$$\mathcal{L}(\theta_Q) = \frac{1}{2} \left[y_t - Q_{\theta_Q}(s_t, k_t, \pi_{\theta_x}(s_t)) \right]^2, \tag{13}$$

where

$$y_t = r_t + \gamma \max_{k \in [K]} Q_{\theta'_Q}(s_{t+1}, k, \pi_{\theta'_x}(s_{t+1})). \tag{14}$$

The loss of the actor-parameter network is given by the negative sum of Q-values as

$$\mathcal{J}(\theta_x) = -\sum_{k=1}^{K} Q_{\theta_Q}(s_t, k, \pi_{\theta_x}(s_t)). \tag{15}$$

4.2. Hindsight experience replay

HER is another important baseline algorithm that we consider in our method. The core idea is to expand the experience by constructing the goal variable as shown in Figure3 (a). There are two ways to construct the goal variable:

1) Direct construction: The agent uses the target information feedback by the environment as the goal variable at step t, which is recorded as $desired\_goal_t$. Additionally, the location of itself at step t is recorded as $achieved\_goal_t$. In the relay navigation task, the traditional algorithm will always record the location of the supply as $achieved\_goal_t$. In our method, the information represented by $achieved\_goal_t$ changes with the state of the agent; see the end of this section for the specific process.
2) Replacement construction: Through direct construction, an experience can be obtained at step t, simplified as $(achieved\_goal_t, desired\_goal_t)$. The HER algorithm will randomly sample 4 items from the experience obtained from step $t+1$ to step T and then uses the $achieved\_goal$ as the goal variable to replace the $desired\_goal$ in the experience at step t. T is the maximum step that allows the agent to act in a single task.

The new goals constructed in the above way have the potential to be generalized to unseen real goals. Although the agent fails to achieve a given goal in the current episode, it still learns action sequences that may achieve different given goals in a future episode. Therefore, the original failure transition can be transformed into the virtual success transition by selecting a new goal from the state experienced to replace the initial goal.

However, HER tends to be less efficient in some relay tasks. As shown in Figure 3 (b), the goal of the agent is to deliver the block to the target area. It should be noted that the $achieved\_goal$ (i.e., the location of the block)

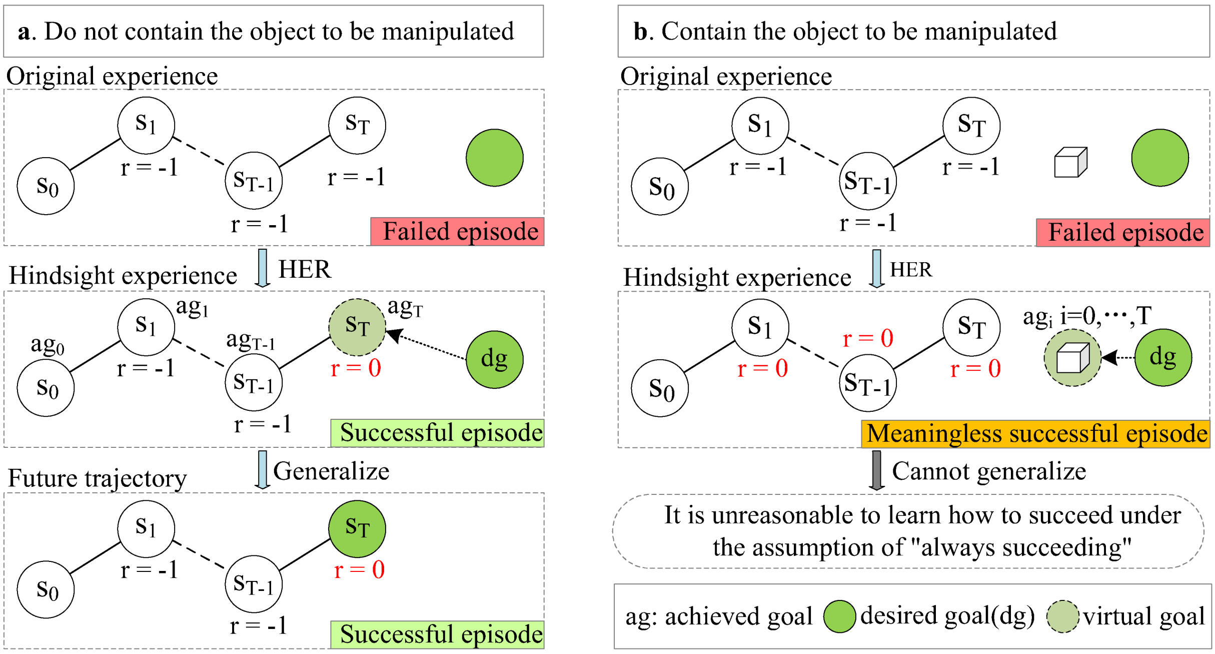


Figure 3. Illustration of positive reward sparsity for HER. (a) In the task that does not contain the manipulated object, achieved goal is directly affected by the behavior of the agent and constantly changes in each rollout. In this case, HER can generate valuable learning experiences. (b) For the task containing the manipulated object, achieved goal remains unchanged until the agent comes into contact with the object. In this case, all the experience generated by HER includes positive rewards but has no substantial help to the learning of the agent.

defined in HER has not changed when the agent is not in contact with the block. Any hindsight goal selected by HER is labeled as a "success episode". However, such a "successful episode" cannot bring meaningful guidance to the agent. This, in part, leads to another kind of sparsity, positive reward sparsity.

To solve this problem, we propose the goal-switch mechanism (GSM). The principle of the GSM aims to generate meaningful goal variables by dynamically assigning goals during experience expansion. In the relay navigation task, the conventional HER approach always designates the target area as the *desired_goal* and the supply point as the *achieved_goal*. GSM assists the UAV in determining the goal based on the current state: (1) Prior to acquiring the supply, the supply point is marked as the *desired_goal*, and the own position of the UAV is labeled as the *achieved_goal*. (2) After acquiring the supply, the target area is marked as the *desired_goal*, while the supply point remains as the *achieved_goal*. By assigning different goals, GSM enables the UAV to construct more effective goal variables and hindsight experiences, which helps mitigate reward sparsity. This simple idea provides an effective solution and is proved in Section 5.

4.3. HER-MPDQN

MP-DQN samples a fixed batch from an offline experience reply buffer to update the network. HER expands the original experience by goal replacement. On this basis, we further eliminate the hindsight experience without guidance significance. This means that our proposal can be effectively integrated with MP-DQN. As shown in Figure 4, the input vector of the neural network is extended in MP-DQN. In specific, the input vector of the actor-parameter network becomes

$$V_{\theta_x} = (s_t, g_t). \tag{16}$$

Correspondingly, the input vector of the Q-value network becomes

$$V_{\theta_Q} = (s_t, g_t, xe_{k_t}). \tag{17}$$

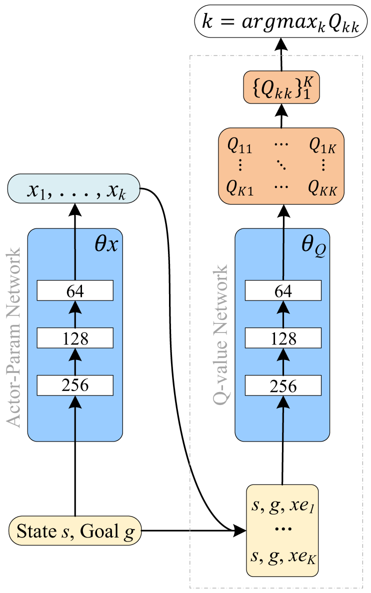


Figure 4. The network architecture of HER-MPDQN. For the symbols in the figure, s denotes the observed state, g represents the goal of the agent, θ_x and θ_Q refer to network parameters, and (256, 128, 64) indicates the number of neurons in the network. x_k is the continuous parameter corresponding to the kth discrete action, where $k = 1, 2, ..., K$. K is the total number of discrete actions. xe_k represents the expanded continuous parameter vector derived from x_k. Q_{kk} denotes the Q value associated with the kth discrete action. The selection of the discrete action k is determined based on the largest Q value.

In our method, the original experience is stored together with the hindsight experience in the replay buffer. A fixed batch of data is sampled from this buffer when the neural network needs to update. The training process in Algorithm 1 is generally divided into three steps: (1) The agent interacts with the environment to accumulate real experience; (2) Capture the original experience and hindsight experience of each time step and store all of them in the replay buffer; (3) Update the parameters of the network according to the length of each episode.

5. RESULTS

To evaluate the effectiveness of HER-MPDQN, a UAV is fully trained and tested in two environments. A relatively simple direct navigation environment and experimental results are first presented in 5.1. After that, a more complex relay navigation task is considered. The related experiments and result analysis are in 5.2.

5.1. The direct navigation task

To our knowledge, open-source benchmark environments for UAV navigation tasks with parameterized action spaces and sparse rewards are currently lacking. Therefore, we simulate a large-scale environment inspired by the existing simulation[1]. Figure 1 shows a top view of the navigation environment. The UAV can fly within four square kilometers ($2km \times 2km$). The same settings for the environment randomness are used to have a fair comparison. In each episode, the coordinate of the UAV and the target area are sampled from the uniform distribution of the entire environment. The default initial velocity of the UAV is 0. The goal of the UAV is to reach the target area in a limited number of steps. The optional actions are MOVE (acceleration) and TURN (angle). MOVE indicates that the UAV moves along the current direction with the given acceleration. TURN means that the UAV rotates at the given angle. The episode ends if the UAV reaches the target area (winning state) or the time limit exceeds.

[1] https://github.com/thomashirtz/gym-hybrid.

```
Algorithm 1: HER-MPDQN
Input: Minibatch size $B$, exploration parameter $\epsilon$, soft update parameter $\tau$, learning rate $\alpha_{\theta_Q}$ and $\alpha_{\theta_x}$.
Initialize the weight of Q-value network and actor-parameter network.
Initialize target networks by hard update.
for $episode = 0, ..., max\_episode$ do
    Select an initial state $s_0$ and an initial goal $g_0$
    for $t = 0, ..., max\_step$ do
        Compute action parameters $x_{k_t}$ by $\pi_{\theta_x}$:
        $x_{k_t} = \pi_{\theta_x}(s_t||g_t)$
        Select action $a_t = (k_t, x_{k_t})$ by $\epsilon$-greedy policy
        Recieve $r_t$, $s_{t+1}$ and $g_{t+1}$ by excute action $a_t$
    end
    for $t = 0, ..., episode\_length$ do
        Store the transition $(s_t, a_t, r_t, s_{t+1}, g_t)$ in $R$
        Sample additional goals $G$ for current transition
        for $g' \in G$ do
            $r' := r(s_t, a_t, g')$
            Store the transition $(s_t, a_t, r', s_{t+1}, g')$ in $R$
        end
    end
    for $t = 0, ..., episode\_length * U$ do
        Sample a minibatch from $R$
        Compute the target $y_i$ according to (5)
        Optimize parameters $\pi_{\theta_Q}$ and $\pi_{\theta_x}$
        Update the weights of target networks by:
            $\theta_Q^{'} \leftarrow \tau * \theta_Q + (1 - \tau) * \theta_Q^{'}$
            $\theta_x^{'} \leftarrow \tau * \theta_x + (1 - \tau) * \theta_x^{'}$
    end
end
```

We chose Python 3.7 as our development language due to its simplicity, flexibility, and efficient development capabilities. For algorithm development and testing, we utilize the OpenAI Gym library, which is an open-source RL library that offers convenient tools for creating custom environments. In terms of hardware, we employ an AMD Ryzen 7 5800H processor and 16GB of RAM. This configuration is relatively new and provides sufficient computing resources to support the training and testing of the algorithms developed in this study.

In the context of sparse rewards, the agent ends the current episode either when the task completion or the agent goes out of bounds which results in higher rewards. Therefore, relying solely on "episode rewards" is inadequate to assess the learning effectiveness. To evaluate algorithm performance, we utilize the "success rate" as a metric, which directly reflects the number of times the agent completes the task. Also, a higher "success rate" implies a higher "reward". The "success rate" is defined as follows:

$$\mathcal{SR} = \frac{CT}{TT} \tag{18}$$

where $\mathcal{SR}$ means "success rate", CT is the number of times the agent completes the task, and TT is the number of times the agent performs the task.

We evaluated the performance of HER-MPDQN in the above environment. Besides, we also implement and test the following three baselines: HER-PDQN[31], MP-DQN,[21] and P-DQN[20]. Each algorithm uses the same network with hidden layer sizes (128,64). The step size of each environment is limited to 100, the size of

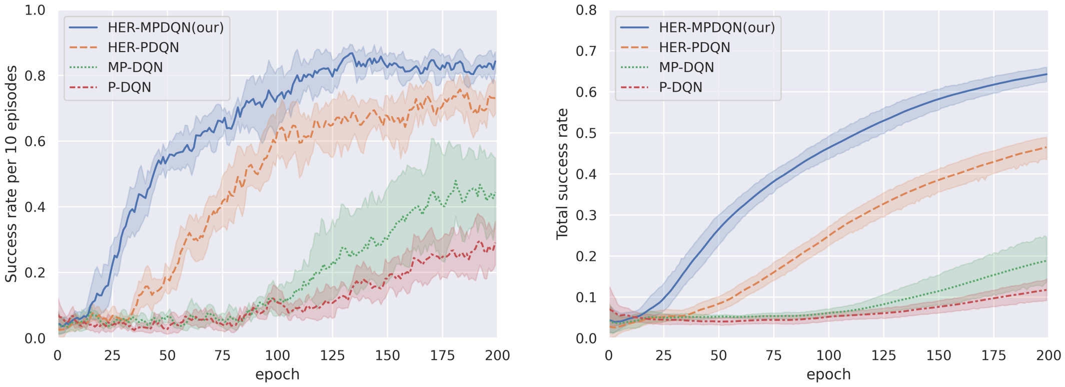


Figure 5. Comparisons of the results using HER-MPDQN and other baselines. The left figure shows the periodic calculation of the task completion rate of the last 10 episodes, and the right figure shows the total success rate during the entire training process. The shaded area is the variance in multiple experiments. Smaller shading indicates that the algorithm is less sensitive to random seeds.

Table 1. The average performance over 1000 episode evaluations

	The direct navigation task	
	Success rate	Mean reward
HER-MPDQN(our)	**0.810** ± **0.030**	**-40.095** ± **4.861**
HER-PDQN	0.725 ± 0.082	-43.141 ± 11.917
MP-DQN	0.412 ± 0.165	-70.504 ± 15.897
P-DQN	0.305 ± 0.102	-84.504 ± 13.125

the replay buffer R is $50,000$, the mini-batch size B is 128, and the update frequency U of the network is 40. The learning rate of the Q-value network α_{θ_Q} and actor-parameter network α_{θ_x} is 10^{-2} and 10^{-3} respectively. Adam optimizer with an exponential decay rate of ($\beta_1 = 0.9$, $\beta_2 = 0.999$) is used for optimization. All algorithms run seven independent experiments and train 2,000 episodes in each experiment.

Figure 5 provides the training process of the UAV in the direct navigation task. Table 1 shows the evaluation performance for each algorithm in 1000 episodes. The results show that HER-MPDQN has a faster convergence speed and higher average success rate. Compared to HER-PDQN, our method evaluates the Q-value more accurately and updates the actor-parameter network without bias. At the same time, the agent learns effective experience early by introducing HER, which brings stronger learning ability than the original MP-DQN. Since P-DQN has no additional mechanism to eliminate the influence of irrelevant parameters and deal with sparse rewards, its performance is not satisfactory. In addition, it can be seen that the learning curve of HER-MPDQN has a smaller shaded area. This means that HER-MPDQN has better robustness in various experiments.

5.2. The relay navigation task

Considering the need for UAVs to deliver supplies, a relay navigation environment is further simulated. Compared with the previous environment setting, the following changes are made: (1) Add a supply point that requires the UAV to perform relay operations. (2) Introduce a new discrete action CATCH to represent grabbing operations. As shown in Figure 2, the UAV must deliver supply to the target area in a limited number of steps. CATCH is valid only when the UAV is in contact with the supply. Each episode ends when the supply has been transported to the target area (winning state) or the time limit exceeds.

Due to the increased complexity, the hyperparameters for this task are tuned as follows. The hidden layer size of all networks is set to (256, 128, 64). The size of the replay buffer R is $150,000$. The update frequency U

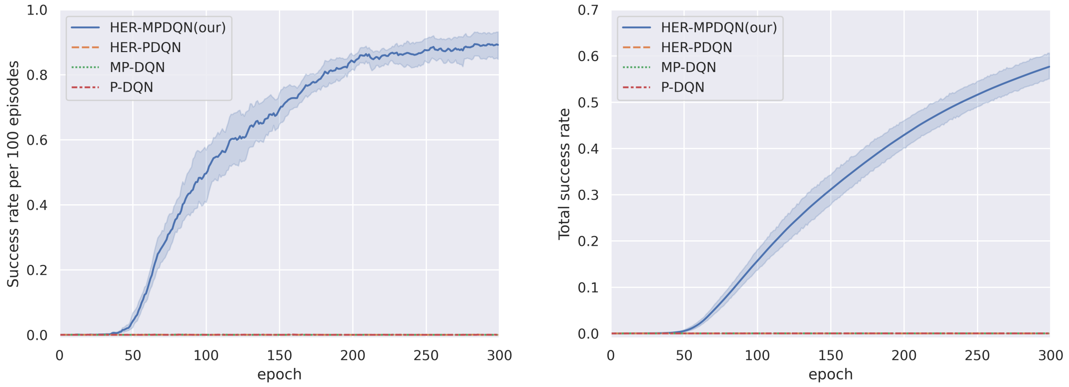


Figure 6. Comparisons of the results using HER-MPDQN and other baselines. The left figure shows the periodic calculation of the task completion rate of the last 100 episodes; other descriptions for evaluating algorithm performance are the same as Figure 5.

Table 2. The average performance over 1000 episodes evaluations

	The relay navigation task	
	Success rate	Mean reward
HER-MPDQN(our)	**0.885** ± **0.059**	**-57.758** ± **6.294**
HER-PDQN	0.000 ± 0.000	-00.000 ± 0.000
MP-DQN	0.000 ± 0.000	-00.000 ± 0.000
P-DQN	0.000 ± 0.000	-00.000 ± 0.000

changes according to the episode length and is limited to [1∼10]. The learning rate of the Q-value network α_{θ_Q} and actor-parameter network α_{θ_x} is 10^{-3} and 10^{-5}, respectively. Each algorithm is trained with 30,000 episodes in each experiment.

Figure 6 shows the learning curve in the relay task. The evaluation results are given in Table2. It is obvious that HER-MPDQN is far superior to the other algorithms. On the one hand, more sparse rewards make it impossible for the UAV to complete tasks through random actions. Therefore, the MP-DQN and P-DQN without HER are challenging to learn. On the other hand, although many positive experiences are generated by introducing HER, they do not have guide significance. For this reason, HER-PDQN cannot learn policy effectively. By treating the relay task as a continuous multi-stage task, the goal is automatically allocated by HER-MPDQN according to the current state. Thus, the hindsight experience of different stages has the correct guiding significance. Figure 7 shows the flight trajectories of the trained UAV when performing specific relay navigation tasks. It can be seen that the UAV has learned the correct action policy. Not only that, HER-MPDQN exhibits good scalability in our experiments. The multi-goal relay navigation task can be completed by expanding the goal space. We leave this research for future work.

6. DISCUSSION

Existing baselines can learn effective policies in the direct navigation task but fail to handle the relay navigation task. In contrast, HER-MPDQN achieves satisfactory results in both simple and complex tasks. This means that HER-MPDQN is versatile in solving different types of navigation tasks. General-purpose agents are one of the critical directions of DRL research, which can avoid repeated algorithm design. HER-MPDQN has excellent advantages in tasks that are difficult to design reward functions and require flexible rescue strategies. However, the algorithm in environments with obstacles has yet to test, and the flying altitude of the UAV is fixed. These limitations make the transfer of simulation to reality more difficult. Future research can consider addressing the sparse reward problem in more realistic simulation conditions, which can narrow the gap

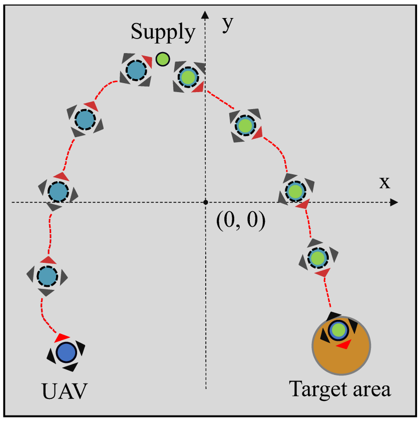


Figure 7. Illustration of flight trajectory map of the trained UAV in the relay navigation task. The red arrow represents the current direction of the UAV, and the red dotted line is the flight path.

between "sim-to-real".

7. CONCLUSIONS

In this paper, the HER-MPDQN algorithm is developed to address UAV navigation tasks with parametrized action space and sparse reward. In addition, a goal-switching method is proposed to correct meaningless hindsight experiences in the relay navigation task. The experiments show that HER-MPDQN outperforms baselines regarding training speed and converged value. Especially in the relay task, only the agent trained by HER-MPDQN learns effectively due to reasonable experience expansion. Further research could consider the energy consumption model of UAVs and test real UAVs in high-dimensional environments containing obstacles.

DECLARATIONS

Authors' contributions

Made contributions to the conception and design of the work, developed most associated code for the simulation environment and the reinforcement learning method, and performed data analysis and visualizations, manuscript writing, and related tasks: Feng S

Contributed to parts of the conception and design and performed the validation of the experimental design and related tasks: Li X

Contributed to the research supervision and guidance, provided environmental equipment, and performed analysis and transition of formal specifications, manuscript writing, and related tasks: Ren L

Participated in part of the experimental data analysis and visualizations and performed data collation and related tasks: Xu S

Availability of data and materials

This work is based on the Gym platform. All codes can be found at https://github.com/No-human-is-limited/HER-MPDQN. There are no additional data or materials associated with this study.

Financial support and sponsorship

This work was supported in part by the Open Project Program of Fujian Provincial Key Laboratory of Intelligent Identification and Control of Complex Dynamic System No:2022A0001; in part by the National Natural Science Foundation of China No:62203002; and in part by the Natural Science Foundation of Anhui Province No:2208085QF204.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Mourtzis D. Simulation in the design and operation of manufacturing systems: state of the art and new trends. *Int J Prod Res* 2020;58:1927-49. DOI
2. Mourtzis D. Design and operation of production networks for mass personalization in the era of cloud technology. Amsterdam: Elsevier; 2021;1-393. DOI
3. Atif M, Ahmad R, Ahmad W, Zhao L, Rodrigues JJ. UAV-assisted wireless localization for search and rescue. *IEEE Syst J* 2021;15:3261-72. DOI
4. Zhao L, Liu Y, Peng Q, Zhao L. A dual aircraft maneuver formation controller for mav/uav based on the hybrid intelligent agent. *Drones* 2023;7:282. DOI
5. Mourtzis D, Angelopoulos J, Panopoulos N. Unmanned aerial vehicle (UAV) manipulation assisted by augmented reality (AR): the case of a drone. *IFAC-PapersOnLine* 2022;55:983-8. DOI
6. Walker O , Vanegas F , Gonzalez F ,et al. A deep reinforcement learning framework for UAV navigation in indoor environments. 2019 IEEE Aerospace Conference. 2019;1-14. DOI
7. Wang Q, Zhang A, Qi L. Three-dimensional path planning for UAV based on improved PSO algorithm. In: The 26th Chinese Control and Decision Conference (2014 CCDC). IEEE; 2014;3981-5. DOI
8. Yao P, Xie Z, Ren P. Optimal UAV route planning for coverage search of stationary target in river. *IEEE Trans Contr Syst Technol* 2017;27:822-9. DOI
9. Shin J, Bang H, Morlier J. UAV path planning under dynamic threats using an improved PSO algorithm. *Int J Aerospace Eng* 2020;2020:1-17. DOI
10. Wen C, Qin L, Zhu Q, Wang C, Li J. Three-dimensional indoor mobile mapping with fusion oftwo-dimensional laser scanner and RGB-D camera data. *IEEE Geosci Remote Sensing Lett* 2013;11:843-7. DOI
11. Mu B, Giamou M, Paull, et al. Information-based active SLAM via topological feature graphs. In: 2016 IEEE 55th Conference on decision and control (Cdc). IEEE; 2016. pp. 5583-90. DOI
12. Weiss S, Scaramuzza D, Siegwart R. Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments. *J Field Robot* 2011;28:854-74. DOI
13. Silver D, Huang A, Maddison CJ, et al. Mastering the game of Go with deep neural networks and tree search. *Nature* 2016;529:484-9. DOI
14. Levine S, Finn C, Darrell T, Abbeel P. End-to-end training of deep visuomotor policies. *J Mach Learn Res* 2016;17:1334-73. DOI
15. Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Ind Robot* 2018;37:421-36. DOI
16. Yan C, Xiang X. A path planning algorithm for uav based on improved q-learning. In: 2018 2nd international conference on robotics and automation sciences (ICRAS). IEEE; 2018.;1-5. DOI
17. Bouhamed O, Ghazzai H, Besbes H, Massoud Y. Autonomous UAV navigation: A DDPG-based deep reinforcement learning approach. In: 2020 IEEE International Symposium on circuits and systems (ISCAS). IEEE; 2020. pp. 1-5. DOI
18. Masson W, Ranchod P, Konidaris G. Reinforcement learning with parameterized actions. In: Thirtieth AAAI Conference on Artificial Intelligence; 2016. DOI
19. Hausknecht M, Stone P. Deep reinforcement learning in parameterized action space. In: Proceedings of the International Conference on Learning Representations. 2016. DOI
20. Xiong J, Wang Q, Yang Z, et al. Parametrized deep q-networks learning: reinforcement learning with discrete-continuous hybrid action space. *arXiv preprint arXiv:181006394.* 2018. DOI
21. Bester CJ, James SD, Konidaris GD. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces. *arXiv preprint arXiv:190504388.* 2019. DOI
22. Wang W, Luo X, Li Y, Xie S. Unmanned surface vessel obstacle avoidance with prior knowledge-based reward shaping. *Concurrency Computat Pract Exper* 2021;33:e6110. DOI
23. Okudo T, Yamada S. Subgoal-based reward shaping to improve efficiency in reinforcement learning. *IEEE Access.* 2021;9:97557-68. DOI
24. Burda Y, Edwards H, Storkey A, Klimov O. Exploration by random network distillation. *arXiv preprint arXiv:181012894.* 2018. DOI

25. Badia AP, Sprechmann P, Vitvitskyi A, et al. Never give up: learning directed exploration strategies. *arXiv preprint arXiv:200206038.* 2020. DOI
26. Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay. *Adv Neural Inf Process Syst* 2017;30. DOI
27. Lanka S, Wu T. Archer: aggressive rewards to counter bias in hindsight experience replay. *arXiv preprint arXiv:180902070.* 2018. DOI
28. Schramm L, Deng Y, Granados E, Boularias A. USHER: unbiased sampling for hindsight experience replay. *arXiv preprint arXiv:220701115.* 2022. Available from: https://proceedings.mlr.press/v205/schramm23a/schramm23a.pdf [Last accessed on 15 Jun 2023]
29. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *nature.* 2015;518:529-33. DOI
30. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:150902971* 2015. DOI
31. Liu C, Van Kampen EJ. HER-PDQN: a reinforcement learning approach for uav navigation with hybrid action spaces and sparse rewards. In: AIAA SCITECH 2022 Forum; 2022;0793. DOI

Intelligence & Robotics

Research Article

Open Access

Check for updates

Adaptive backstepping control of high-order fully actuated nonlinear systems with event-triggered strategy

Chengyuan Yan, Jianwei Xia, Xinru Liu, Huarong Yue, Chong Li

School of Mathematics Science, Liaocheng University, Liaocheng 252000, Shandong, China.

Correspondence to: Prof. Jianwei Xia, School of Mathematics Science, Liaocheng University, Liaocheng 252000, Shandong, China. E-mail: njustxjw@126.com

How to cite this article: Yan C, Xia J, Liu X, Yue H, Li C. Adaptive backstepping control of high-order fully actuated nonlinear systems with event-triggered strategy. *Intell Robot* 2023;3(2):176-89. http://dx.doi.org/10.20517/ir.2023.11

Received: 19 Mar 2023 **First Decision:** 6 May 2023 **Revised:** 23 May 2023 **Accepted:** 26 May 2023 **Published:** 27 Jun 2023

Academic Editor: Simon X. Yang **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

This paper investigates the problem of adaptive event-triggered fuzzy control for nonlinear high-order fully actuated systems. In this paper, a completely unknown nonlinear function is considered, and its prior knowledge is unknown. To solve this problem, the fuzzy logic system technology is applied to approximate the unknown nonlinear function. In order to save communication resources, a novel high-order event-triggered controller is proposed under backstepping control. With the help of Lyapunov stability theory, it is proved that all signals of the closed-loop system are bounded. Finally, the theoretical results are applied to the robot system to verify their validity.

Keywords: fuzzy logic system, event-triggered strategy, high-order fully actuated nonlinear systems, adaptive control

1. INTRODUCTION

With the development of modern society and modern industry, linear system theory has become relatively well-established and sophisticated[1–3]. Many scholars have proposed various powerful analysis tools for linear systems. However, with the progress of science and technology and the improvement of the accuracy of measuring tools, the understanding of the actual system is gradually deepened, and the requirements for its control performance are also increasingly high. Ignoring some objective factors, some practical systems are modeled as linear systems and controller designs are carried out, but the designed controllers have not met the requirements for the control performance of practical systems. In such cases, it is particularly necessary to model some practical systems into nonlinear systems. This includes systems such as unmanned vehicle systems[4], unmanned aerial vehicle systems[5], robot systems, and manipulator systems[6]. Therefore, nonlinear systems have received extensive attention from scholars at home and abroad and have proposed various tools to handle the control problem of nonlinear systems, such as adaptive backstepping control[7,8], sliding mode control[9], etc. Among them, the combination of backstepping recursive design and adaptive control has produced a large number of excellent results[10–14].

The high-order fully actuated system possesses unparalleled control characteristics compared to other systems. Its fully actuated characteristics enable the elimination of all dynamic characteristics of the open-loop system while establishing new and desired closed-loop dynamic characteristics. About high-order fully actuated systems, there have been some excellent results[15–23]. Among them, The work[19] proposed the direct parametric approach of fully-actuated high-order systems. A constrained cooperative control is proposed[22] for high-order fully actuated multiagent systems with prescribed performance.

Since the beginning of this century, networked control systems[24–29] have been widely used in remote operation, industrial automation, building energy conservation, and other fields. This is due to their low maintenance cost and high flexibility. In the networked control system, the actuator, controller, sensor, and other components transmit information through the shared network channel. Therefore, it is necessary to reduce the occupation of shared communication by single subsystem control to achieve the purpose of saving cost and energy. The traditional sampling control[30–33]is based on the system signal sampling value instead of continuous value and takes different constant values periodically, which has relatively high communication efficiency compared with continuous time control. Sampling control requires information transmission and control update at a conservative fixed frequency regardless of obvious changes in system performance, so it is not suitable for networked control systems with high integration, which leads to the emergence of more efficient control of resource utilization, namely event-triggered control. The key point of the event-triggered control design is to build an event-triggered mechanism. The most basic types are absolute threshold type, relative threshold type, and mixed threshold type. The construction of an event-triggered mechanism depends not only on the system structure but also on the expected control objectives. Even with the increase in system complexity and performance requirements, additional dynamic and online adjustment parameters need to be introduced. Over the past decade, significant progress has been made in the research of event-triggered control for nonlinear systems[34–43].

Inspired by the above excellent results and combined with the reality of the lack of event-triggered control results of the high-order fully activated system, this paper studies the adaptive fuzzy event-triggered control for the high-order fully activated system. The contribution of this paper is reflected in two aspects:

1) For the uncertain high-order fully actuated nonlinear system, the unknown nonlinear function is considered, and the fuzzy logic system (FLS) is used to approximate the nonlinear function without a priori condition of the nonlinear function.

2) The proposed event-triggered scheme for the uncertain high-order fully nonlinear system can effectively eliminate the continuous update of the designed controller, thus saving communication resources.

The organization of this article is arranged as follows. The second section includes problem formulas and preliminary knowledge. The third section introduces an event-triggered controller design scheme. The fourth section shows the simulation. The fifth section is the summary.

Notation

I_n represents the identity matrix and

$$x^{(0\sim n)} = \begin{bmatrix} x \\ \dot{x} \\ \vdots \\ x^{(n)} \end{bmatrix},$$

$$A^{0\sim n-1} = \begin{bmatrix} A_0 & A_1 & \dots & A_{n-1} \end{bmatrix},$$

$$I_n^\circ = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \ddots & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$$x_{i\sim j}^{(0\sim n)} = \begin{bmatrix} x_i^{(0\sim n)} \\ x_{i+1}^{(0\sim n)} \\ \vdots \\ x_j^{(0\sim n)} \end{bmatrix}, j \geq i$$

$$\Phi(A^{0\sim n-1}) = \begin{bmatrix} 0 & I & & \\ & & \ddots & \\ & & & I \\ -A_0 & -A_1 & \dots & A_{n-1} \end{bmatrix}.$$

2. PROBLEM FORMULAS AND PRELIMINARY KNOWLEDGE

2.1. Problem statement

Consider the following uncertain high-order fully nonlinear system:

$$\begin{cases} x_1^{(p_1)} = g_1(x_1^{(0\sim p_1-1)})x_2 + f_1(x_1^{(0\sim p_1-1)}), \\ x_j^{(p_j)} = g_j(x_i^{(0\sim p_i-1)}|_{i=1\sim j})x_{j+1} + f_j(x_i^{(0\sim p_i-1)}|_{i=1\sim j}), \\ x_n^{(p_n)} = g_n(x_i^{(0\sim p_i-1)}|_{i=1\sim n})u + f_n(x_i^{(0\sim p_i-1)}|_{i=1\sim n}), \end{cases} \tag{1}$$

where $p_i \in \mathbb{N}^+$, and u denotes the system input. $f_i(x_i^{(0\sim p_i-1)}|_{i=1\sim j})$ are sufficiently smooth unknown nonlinear functions, $g_i(x_i^{(0\sim p_i-1)}|_{i=1\sim j})$ are control gain functions, and satisfy full-actuation conditions.

Remark 1 *The above-mentioned high-order fully nonlinear system is the general form of a second-order fully nonlinear system. For practical examples, such as robotic systems, it is no longer necessary to transform a high-order system into a first-order system. Instead, we can deal with it directly.*

2.2. Preliminaries knowledge

Assumption 1 [37] *There are two constants that the control gain functions* $0 < \underline{g}_j \leq |g_j(x_i^{(0\sim p_i-1)}|_{i=1\sim j})| \leq \bar{g}_j, j = 1, \cdots, n.$

Remark 2 *The above assumption is a common standard condition that ensures the controllability of the uncertain high-order fully nonlinear system. This is derived from modeling real systems, and it makes perfect sense.*

Lemma 1 [38] *The unknown nonlinear continuous function* $\zeta(\xi)$ *is defined on a compact set. And there is an FLS satisfying the following inequality*

$$\zeta(\xi) = W^{*T}S(\xi) + \delta(\xi), \tag{2}$$

where $\delta(\xi)$ indicates the any estimation error which satisfies $|\delta(x)| \leq \bar{\delta}$.

Lemma 2 [38] *For $\forall\epsilon > 0$ and $\sigma \in R$, it can be concluded that*

$$0 \leq |\sigma| - \sigma \tanh(\frac{\sigma}{\epsilon}) \leq 0.2785\epsilon. \tag{3}$$

Lemma 3 [18] *Design the matrix $A_i^{0\sim p_i-1} \in \mathbb{R}^{1\times p_i}$ so that the matrix $\Phi(A_i^{0\sim p_i-1}) \in \mathbb{R}^{p_i\times p_i}$ is stable. Moreover, according to Lyapunov Theorem, there is a matrix $P_i(A_i^{0\sim p_i-1}) \in \mathbb{R}^{p_i\times p_i}$, which is positive definite, satisfying*

$$\Phi(A_i^{0\sim p_i-1})^T P_i(A_i^{0\sim p_i-1}) + P_i(A_i^{0\sim p_i-1})\Phi(A_i^{0\sim p_i-1}) = -\rho_i I_i, \tag{4}$$

where $\rho_i > 0$ ($i = 1, \cdots, n$) are design parameters.

3. CONTROLLER DESIGN AND STABILITY ANALYSIS

3.1. Adaptive event-triggered controller design

To facilitate the calculation, we first give some necessary coordinate transformations:

$$\tilde{P}_i(A_i^{0\sim p_i-1}) = I_2^\circ P_i^T(A_i^{0\sim p_i-1}),$$

$$P_i(A_i^{0\sim p_i-1}) = \begin{bmatrix} P_{iF}(A_i^{0\sim p_i-1}) & \cdots & P_{iL}(A_i^{0\sim p_i-1}) \end{bmatrix},$$

$$\tilde{P}_i^{-1}(A_i^{0\sim p_i-1}) = \begin{bmatrix} Q_{i11}(A_i^{0\sim p_i-1}) & Q_{i12}(A_i^{0\sim p_i-1}) & Q_{i13}(A_i^{0\sim p_i-1}) \\ Q_{iF}(A_i^{0\sim p_i-1}) & Q_{iM}(A_i^{0\sim p_i-1}) & Q_{iL}(A_i^{0\sim p_i-1}) \end{bmatrix},$$

where $Q_{iL}(A_i^{0\sim p_i-1}) \neq 0$ ($i = 1, \cdots, n$).

Step 1: Let

$$s_1^{(0\sim p_1-1)} = x_1^{(0\sim p_1-1)}, \tag{5}$$

and

$$\tilde{P}_2(A_2^{0\sim p_2-1}) s_2^{(0\sim p_2-1)} = x_2^{(0\sim p_2-1)} - \begin{bmatrix} \alpha_1 \\ 0 \end{bmatrix}.$$

With the help of the notations, one has

$$P_{2L}^T(A_2^{0\sim p_2-1}) s_2^{(0\sim p_2-1)} = x_2 - \alpha_1.$$

Choose virtual controller α_1 as

$$\alpha_1 = -\frac{1}{g_1(x_1^{(0\sim p_1-1)})}(A_1^{(0\sim p_1-1)} s_1^{(0\sim p_1-1)} + \frac{1}{2a_1^2} P_{1L}^T(A_1^{0\sim p_1-1}) s_1^{(0\sim p_1-1)} \hat{\theta}_1 S_1^T S_1 + \frac{1}{2} P_{1L}^T(A_1^{0\sim p_1-1}) s_1^{(0\sim p_1-1)}). \tag{6}$$

The Lyapunov candidate function V_1 is designed as

$$V_1 = (s_1^{(0\sim p_1-1)})^T P_1(A_1^{0\sim p_1-1}) s_1^{(0\sim p_1-1)} + \frac{1}{2}\tilde{\theta}_1^2, \tag{7}$$

where $\theta_1 = \max\{\|W_1\|^2\}$, $\tilde{\theta}_1 = \theta_1 - \hat{\theta}_1$, and $\hat{\theta}_1$ is the estimation of θ_1.

With the help of FLS and Young's inequality, one gets

$$\begin{aligned}&P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}F_1(X_1)=P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}(W_1^TS_1(X_1)+\delta_1)\\ \leq&\frac{(P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)})^2\theta_1S_1^T(X_1)S_1(X_1)}{2a_1^2}+\frac{1}{2}a_1^2+\frac{1}{2}P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}+\frac{1}{2}\bar{\delta}_1^2,\end{aligned} \tag{8}$$

where $F_1(X_1)=f_1(x_1^{(0\sim p_1-1)})$, $X_1=[x_1^{(0\sim p_1-1)}]$ and a_1 is a constant.

The adaptive law $\hat{\theta}_1$ is chosen as

$$\dot{\hat{\theta}}_1=\frac{1}{a_1^2}P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}S_1^TS_1-l_1\hat{\theta}_1. \tag{9}$$

Based on (8) and (9), one gets

$$\begin{aligned}\dot{V}_1\leq&-\rho_1(s_1^{(0\sim p_1-1)})^Ts_1^{(0\sim p_1-1)}-\frac{1}{2}l_1\tilde{\theta}_1^2+a_1^2+\bar{\delta}_1^2+\frac{1}{2}l_1\theta_1^2\\&+2P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}g_1(x_1^{(0\sim p_1-1)})P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}.\end{aligned} \tag{10}$$

Step 2: Based on the notations, one has

$$P_{3L}^T(A_3^{0\sim p_3-1})s_3^{(0\sim p_3-1)}=x_3-\alpha_2, \tag{11}$$

From (1) and (11), the time derivative of s_2 is

$$\begin{aligned}s_2^{(p_2)}=&Q_{2F}(A_2^{0\sim p_2-1})(\dot{x}_2-\dot{\alpha}_1)+Q_{2M}(A_2^{0\sim p_2-1})\dot{x}_2^{(1\sim p_2-2)}+Q_{2L}(A_2^{0\sim p_2-1})f_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})\\&+Q_{2L}(A_2^{0\sim p_2-1})g_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})P_{3L}(A_3^{(0\sim p_3-1)})s_3^{(0\sim p_3-1)}+Q_{2L}(A_2^{0\sim p_2-1})g_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})\alpha_2.\end{aligned} \tag{12}$$

Choose virtual controller α_2 as

$$\begin{aligned}\alpha_2=&-\frac{1}{Q_{2L}(A_2^{0\sim p_2-1})g_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})}(A_2^{(0\sim p_2-1)}s_2^{(0\sim p_2-1)}\\&+\frac{1}{2a_2^2}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}\hat{\theta}_2S_2^TS_2+\frac{1}{2}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}),\end{aligned} \tag{13}$$

and (12) can be rewritten as state-space form

$$\dot{s}_2^{(0\sim p_2-1)}=\Phi(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}+\begin{bmatrix}0\\H_2\end{bmatrix}$$

where $H_2=Q_{2F}(A_2^{0\sim p_2-1})(\dot{x}_2-\dot{\alpha}_1)-\frac{1}{2a_2^2}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}\hat{\theta}_2S_2^TS_2+Q_{2M}(A_2^{0\sim p_2-1})\dot{x}_2^{(1\sim p_2-2)}+Q_{2L}(A_2^{0\sim p_2-1})$ $\times f_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})-\frac{1}{2}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)})+Q_{2L}(A_2^{0\sim p_2-1})g_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})P_{3L}(A_3^{(0\sim p_3-1)})s_3^{(0\sim p_3-1)}$.

The Lyapunov function candidate V_2 is presented as

$$V_2=V_1+(s_2^{(0\sim p_2-1)})^TP_2(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}+\frac{1}{2}\tilde{\theta}_2^2. \tag{14}$$

And similar to the (8), one gets

$$\begin{aligned}&P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}F_2(X_2) = P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}(W_2^T S_2(X_2)+\delta_2)\\ \le&\frac{(P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)})^2\theta_2 S_2^T(X_2)S_2(X_2)}{2a_2^2}+\frac{1}{2}a_2^2+\frac{1}{2}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}+\frac{1}{2}\bar{\delta}_2^2\end{aligned} \tag{15}$$

where $F_2(X_2) = Q_{2F}(A_2^{0\sim p_2-1})(\dot{x}_2-\dot{\alpha}_1)+Q_{2M}(A_2^{0\sim p_2-1})\dot{x}_2^{(1\sim p_2-2)}+Q_{2L}(A_2^{0\sim p_2-1})f_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})$ $+P_{1L}^T(A_1^{0\sim p_1-1})s_1^{(0\sim p_1-1)}$, $X_2=[x_1^{(0\sim p_1-1)},x_2^{(0\sim p_2-1)},\hat{\theta}_1]$.

The adaptive update law $\hat{\theta}_2$ is designed as

$$\dot{\hat{\theta}}_2 = \frac{1}{a_2^2}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}S_2^T S_2 - l_2\hat{\theta}_2. \tag{16}$$

Replacing (15) and (16) into (14), one gives

$$\begin{aligned}\dot{V}_2\le&-\sum_{j=1}^{2}\tau_j(s_j^{(0\sim p_j-1)}|_{j=1\sim 2})^T(s_j^{(0\sim p_j-1)}|_{j=1\sim 2})+\sum_{j=1}^{2}(a_j^2+\bar{\delta}_j^2+\frac{1}{2}l_j\theta_j^2)-\sum_{j=1}^{2}\frac{l_j}{2}\tilde{\theta}_j^2\\&+2P_{2L}^T(A_2^{0\sim p_2-1})s_2^{(0\sim p_2-1)}Q_{2L}(A_2^{0\sim p_2-1})g_2(x_i^{(0\sim p_i-1)}|_{i=1\sim 2})P_{3L}(A_3^{(0\sim p_3-1)})s_3^{(0\sim p_3-1)}\end{aligned} \tag{17}$$

Step k:($3\le k\le n-1$) Based on the notations, one has

$$P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)} = x_k-\alpha_{k-1}, \tag{18}$$

Choose virtual controller α_k as

$$\begin{aligned}\alpha_k=&-\frac{1}{Q_{kL}(A_k^{0\sim p_k-1})g_k(x_i^{(0\sim p_i-1)}|_{i=1\sim k})}(A_k^{(0\sim p_k-1)}s_k^{(0\sim p_k-1)}\\&+\frac{1}{2a_k^2}P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}\hat{\theta}_k S_k^T S_k+\frac{1}{2}P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}).\end{aligned} \tag{19}$$

The Lyapunov function candidate V_k is presented as

$$V_k = V_{k-1}+(s_k^{(0\sim p_k-1)})^T P_k(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}+\frac{1}{2}\tilde{\theta}_k^2. \tag{20}$$

And similar to the (8), one gets

$$\begin{aligned}&P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}F_k(X_k) = P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}(W_k^T S_k(X_k)+\delta_k)\\ \le&\frac{(P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)})^2\theta_k S_k^T(X_k)S_k(X_k)}{2a_k^2}+\frac{1}{2}a_k^2+\frac{1}{2}P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}+\frac{1}{2}\bar{\delta}_k^2\end{aligned} \tag{21}$$

where $F_k(X_k) = Q_{kF}(A_k^{0\sim p_k-1})(\dot{x}_k-\dot{\alpha}_{k-1})+Q_{kM}(A_k^{0\sim p_k-1})\dot{x}_k^{(1\sim p_k-2)}+Q_{kL}(A_k^{0\sim p_k-1})f_k(x_i^{(0\sim p_i-1)}|_{i=1\sim k})$ $+P_{kL}^T(A_k^{0\sim p_k-1})s_k^{(0\sim p_k-1)}Q_{kL}(A_k^{0\sim p_k-1})g_k(x_i^{(0\sim p_i-1)}|_{i=1\sim k})$, $X_k = [x_1^{(0\sim p_1-1)},\cdots,x_k^{(0\sim p_k-1)},\hat{\theta}_1,\cdots,\hat{\theta}_k]$ and a_k is a constant.

The adaptive law $\hat{\theta}_k$ is designed as

$$\dot{\hat{\theta}}_k = \frac{1}{a_k^2} P_{kL}^T (A_k^{0\sim p_k-1}) s_k^{(0\sim p_k-1)} P_{kL}^T (A_k^{0\sim p_k-1}) s_k^{(0\sim p_k-1)} S_k^T S_k - l_k \hat{\theta}_k. \tag{22}$$

Based on (21) and (22), one gives

$$\begin{aligned}\dot{V}_k \leq & - \sum_{j=1}^{k} \tau_j (s_j^{(0\sim p_j-1)}|_{j=1\sim k})^T (s_j^{(0\sim p_j-1)}|_{j=1\sim k}) + \sum_{j=1}^{k} (a_j^2 + \bar{\delta}_j^2 + \frac{1}{2} l_j \theta_j^2) - \sum_{j=1}^{k} \frac{l_j}{2} \tilde{\theta}_j^2 \\ & + 2P_{kL}^T (A_k^{0\sim p_k-1}) s_k^{(0\sim p_k-1)} g_k (x_i^{(0\sim p_i-1)}|_{i=1\sim k}) Q_{kL} (A_k^{0\sim p_k-1}) P_{(k+1)L} (A_{k+1}^{(0\sim p_{(k+1)}-1)}) s_{k+1}^{(0\sim p_{k+1}-1)}\end{aligned} \tag{23}$$

Step n: In this part, the adaptive HOFA event-triggered controller of the system is constant as

$$v(t) = -(1+\gamma)(\alpha_n \tanh(\frac{s_n^{(0\sim p_n-1)} P_{nL} g_n \alpha_n Q_{nL}}{\rho}) + \bar{b} \tanh(\frac{s_n^{(0\sim p_n-1)} P_{nL} g_n \bar{b} Q_{nL}}{\rho}), \tag{24}$$

$$u(t) = v(t_k), \qquad t_k \leq t < t_{k+1}, \tag{25}$$

$$t_{k+1} = \inf\{t \geq 0 || \psi(t)| \geq \gamma |u(t)| + o\}, \tag{26}$$

where $\psi(t) = u(t) - v(t)$, $o > 0$, $\rho > 0$, $0 < \gamma < 1$ and $\bar{b} > \frac{d}{1-\gamma}$ are design parameters.

From (1), the time derivative of s_n is

$$\begin{aligned} s_n^{(p_n)} = & Q_{nF} (A_n^{0\sim p_n-1}) (\dot{x}_n - \dot{\alpha}_{n-1}) + Q_{nM} (A_n^{0\sim p_n-1}) \dot{x}_n^{(1\sim p_n-1)} + Q_{nL} (A_n^{0\sim p_n-1}) f_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n}) \\ & + Q_{nL} (A_n^{0\sim p_n-1}) g_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n}) u + Q_{nL} (A_n^{0\sim p_n-1}) g_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n}) \alpha_n \\ & - Q_{nL} (A_n^{0\sim p_n-1}) g_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n}) \alpha_n \end{aligned} \tag{27}$$

Choose virtual controller α_n as

$$\begin{aligned} \alpha_n = & - \frac{1}{Q_{nL} (A_n^{0\sim p_n-1}) g_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n})} (A_n^{(0\sim p_n-1)} s_n^{(0\sim m_n-1)} \\ & + \frac{1}{2a_n^2} P_{nL}^T (A_n^{0\sim p_n-1}) s_n^{(0\sim m_n-1)} \hat{\theta}_n S_n^T S_n + \frac{1}{2} P_{nL}^T (A_n^{0\sim p_n-1}) s_n^{(0\sim p_n-1)}), \end{aligned} \tag{28}$$

and (27) can be rewritten as state-space form

$$\dot{s}_n^{(0\sim p_n-1)} = \Phi (A_n^{0\sim p_n-1}) s_n^{(0\sim p_n-1)} + \begin{bmatrix} 0 \\ H_n \end{bmatrix}$$

where $H_n = Q_{nF} (A_n^{0\sim p_n-1}) (\dot{x}_n - \dot{\alpha}_{n-1}) + Q_{nM} (A_n^{0\sim p_n-1}) \dot{x}_n^{(1\sim p_n-2)} + Q_{nL} (A_n^{0\sim p_n-1}) f_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n})$ $- \frac{1}{2a_n^2} P_{nL}^T (A_n^{0\sim p_n-1}) s_n^{(0\sim p_n-1)} \hat{\theta}_n S_n^T S_n - \frac{1}{2} P_{nL}^T (A_n^{0\sim p_n-1}) s_n^{(0\sim p_n-1)} - Q_{nL} (A_n^{0\sim p_n-1}) g_n (x_i^{(0\sim p_i-1)}|_{i=1\sim n}) \alpha_n.$

The Lyapunov function candidate V_n is presented as

$$V_n = V_{n-1} + (s_n^{(0\sim p_n-1)})^T P_n (A_n^{0\sim p_n-1}) s_n^{(0\sim p_n-1)} + \frac{1}{2} \tilde{\theta}_n^2. \tag{29}$$

The FLS is used to approximate nonlinear dynamics and adaptive law are same as (21, 22). And from (24, 25, 26), we have $v(t) = \lambda_2(t) o + (1 + \gamma \lambda_1(t)) u(t)$, $\forall t \in [t_k, t_{k+1})$, where $\lambda_1(t) \in [-1, 1]$, $\lambda_2(t) \in [-1, 1]$. Then, we can get

$$u(t) = \frac{v(t)}{1 + \gamma \lambda_1(t)} - \frac{\lambda_2(t) o}{1 + \gamma \lambda_1(t)}. \tag{30}$$

According to $\frac{P_{nL}^T s_n^{(0\sim p_n-1)} Q_{nL} g_n}{1+\gamma\lambda_1(t)} \le \frac{P_{nL}^T s_n^{(0\sim p_n-1)} Q_{nL} g_n}{1+\gamma}$, $P_{nL}^T s_n^{(0\sim p_n-1)} Q_{nL} g_n | \frac{\lambda_2 o}{1+\gamma\lambda_1(t)} | \le P_{nL}^T s_n^{(0\sim p_n-1)} Q_{nL} g_n \frac{o}{1-\lambda_1}$, $\bar{b} > \frac{o}{1-\gamma}$, it yields

$$\dot{V}_n \le -\sum_{j=1}^{n} \rho_j (s_j^{(0\sim p_j-1)}|_{j=1\sim n})^T (s_j^{(0\sim p_j-1)}|_{j=1\sim n}) + \sum_{j=1}^{n} (a_j^2 + \bar{\delta}_j^2 + \frac{1}{2} l_j \theta_j^2) - \sum_{j=1}^{n} \frac{l_j}{2} \tilde{\theta}_j^2 + 0.2785\rho \tag{31}$$

3.2. Stability analysis

Theorem 1: For the high-order fully actuated nonlinear system (1) under the Assumption 1, the virtual controller (6), (13), (19), (28), the actual controller (24), the adaptive law (9), (16), (22), and the event-triggered mechanism (24,25,26) are designed. Then, the following statements hold:

1) All signals in the closed-loop system are bounded.

2) There is a positive constant ϖ which satisfies $t_{k+1} - t_k \ge \varpi$. In other words, the event-triggered condition is Zeno-free.

Proof: 1) Let $V = V_n$. Then we can get

$$\begin{aligned}\dot{V}_n \le& -\sum_{j=1}^{n} \rho_j (s_j^{(0\sim p_j-1)}|_{j=1\sim n})^T (s_j^{(0\sim p_j-1)}|_{j=1\sim n}) + \sum_{j=1}^{n} (a_j^2 + \bar{\delta}_j^2 + \frac{1}{2} l_j \theta_j^2) - \sum_{j=1}^{n} \frac{l_j}{2} \tilde{\theta}_j^2 + 0.2785\epsilon \\ \le& -\varrho_1 V_n(t) + \varrho_2,\end{aligned} \tag{32}$$

where $\varrho_1 = \min\{\frac{\rho_i}{\lambda_{\min}(P_i)}, l_i, i = 1, \cdots, n\}$, $\varrho_2 = \sum_{j=1}^{n} (a_j^2 + \bar{\delta}_j^2 + \frac{1}{2} l_j \theta_j^2)$. According to (40), one has

$$0 \le V(t) \le \frac{\varrho_2}{\varrho_1} + (V(0) - \frac{\varrho_2}{\varrho_1}) e^{-\varrho_1 t}, \tag{33}$$

which means that all signals are bounded.

2) From $\psi(t) = u(t) - v(t)$, $\forall t \in [t_k, t_{k+1})$, we have

$$\tfrac{d}{dt}|\psi| = \tfrac{d}{dt}(\psi \times \psi)^{\frac{1}{2}} = sign(\psi)\dot{\psi} \le \bar{\psi}.$$

where $\bar{\psi}$ is a constant. Since $\psi(t_k) = 0$ and $\lim_{t \to t_{k+1}} \psi(t) = (\gamma|u(t)| + o)$ thus $t_{k+1} - t_k \ge (\gamma|u(t)| + o)/\bar{\psi} > 0$.

4. SIMULATION

In this section, to demonstrate the effectiveness of the designed HOFA event-triggered mechanism, a single-link robot arm simulation is carried out.

Example 1: Consider a single-link robot system whose manipulators with an elastic revolute joint are actuated by a brushed direct current motor that can be given by

$$\begin{cases} M\ddot{a}_1 + mgl\sin(a_1) - K(a_2 - a_1) = 0, \\ J\ddot{a}_2 + B\dot{a}_2 - K(a_1 - a_2) - K_T I = 0, \\ L\dot{I} + RI + K_B\dot{a}_2 - u = 0, \end{cases}$$

where a_1 and a_2 are the angular positions on the link and motor sides, respectively. M and m represent the load and link masses, respectively. B is the coefficient of viscous friction, g is the gravitational acceleration, K

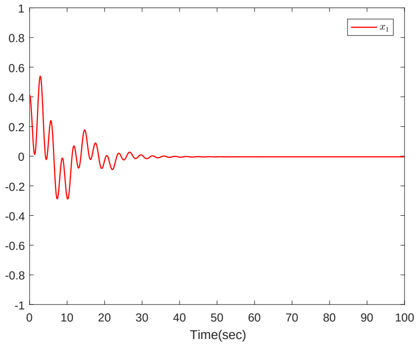


Figure 1. Trajectories of x_1 in Example 1.

is stiffness coefficient of the torsional spring, J is the rotor inertia, l is the link length. K_T and K_B are torque constants of the direct current motor and back-emf coefficient, respectively. L, I, and R are the armature inductance, current, and resistance, respectively. u is the torque input.

Obviously, the above system is a second-order system, and the proposed high-order ETC backstepping can be handled directly without transforming it into a first-order state space form. Let $x_1 = a_1, x_2 = a_2, x_3 = I$.

In simulations, the robot system factors are designed as follows: $R = 25\omega$, $M = 1kg$, $K_T = 1Nm/A$, $mgl = 1Nm$, $K_0 = 2Nm/rad$, $J = 1kgm_2$, $B = 0.9Nms/rad$, $L = 0.125H$, and $K_B = 1Nm/A$.

The design parameters are chosen as $a_1 = 15$, $\rho_1 = 0.16$, $l_1 = 100$, $a_2 = 16$, $\rho_2 = 0.16$, $l_2 = 80$, $a_3 = 15$, $\rho_3 = 112$, $l_3 = 60$, $\bar{m} = 1.1$, $d = 0.5$, $\gamma = 0.5$, and $\epsilon = 6$. The robot system of initial conditions are chosen as $x_1(0) = 0.41$, $\dot{x}_1(0) = 0.02$, $x_2(0) = 0.02$, $\dot{x}_2(0) = 0.22$, $x_3(0) = 0.65$, $\hat{\theta}_1(0) = 0.24$, $\hat{\theta}_2(0) = 0.35$, and $\hat{\theta}_3(0) = 0.41$. In order to satisfy Lyapunov Theorem, some matrices are designed as follows.

$$P_1(A_1^{(0\sim1)}) = \begin{bmatrix} 341/5 & 2/5 \\ 2/5 & 84/25 \end{bmatrix}; \quad A_1 = \begin{bmatrix} 20 & 0.4 \end{bmatrix};$$

$$P_2(A_2^{(0\sim1)}) = \begin{bmatrix} 1630/7733 & 2/101 \\ 2/101 & 121/3031 \end{bmatrix}; A_2 = \begin{bmatrix} 4.04 & 0.4 \end{bmatrix};$$

$$P_3(A_3) = \begin{bmatrix} 7 \end{bmatrix}; \qquad A_3 = \begin{bmatrix} 8 \end{bmatrix}.$$

The simulation results are given as follows. Figure 1 represents the response of the state x_1. The response of the state x_2 is shown by Figure 2. Figure 3 shows the trajectories of the state x_3. The trajectory of the state $\dot{x}_1$ is plotted in Figure 4. Figure 5 portrays the response of the state $\dot{x}_2$. Figure 6 shows the trajectory of the input u. The trigger time intervals are illustrated in Figure 7. The trajectories of adaptive laws are given in Figure 8, Figure 9, and Figure 10 .

5. CONCLUSIONS

In this article, a novel adaptive high-order event-triggered control scheme is proposed for uncertain HOFA nonlinear systems. This scheme not only does not require prior knowledge of the nonlinear function of the system but also saves communication resources by designing the event-triggered scheme. Moreover, the practicality of the control scheme is verified. The future of work will be concerned with the prescribed performance control problem and network attack problem of high-order fully activated nonlinear systems.

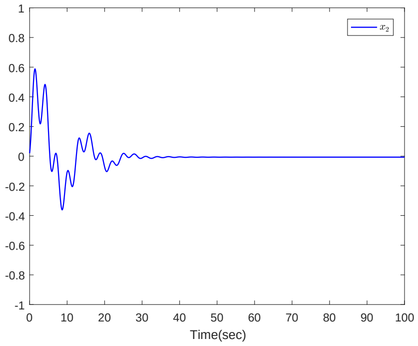


Figure 2. Trajectories of x_2 in Example 1.

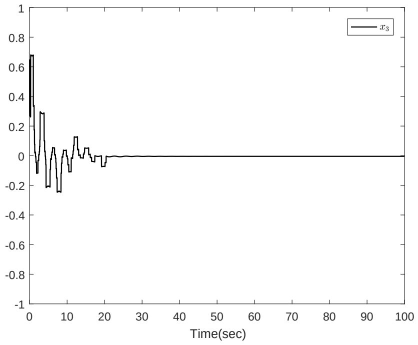


Figure 3. Trajectories of x_3 in Example 1.

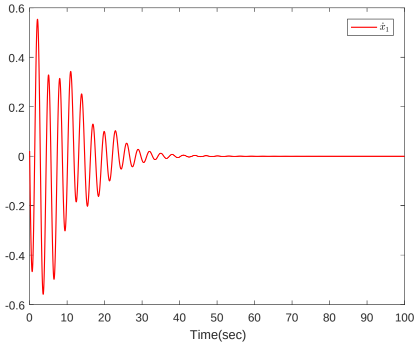


Figure 4. Trajectories of $\dot{x}_1$ in Example 1.

DECLARATIONS

Authors' contributions

Made substantial contributions to the conception and design of the study and performed data analysis and interpretation: Yan C, Xia J, Liu X, Yue H

Performed data acquisition and provided administrative, technical, and material support: Xia J, Li C

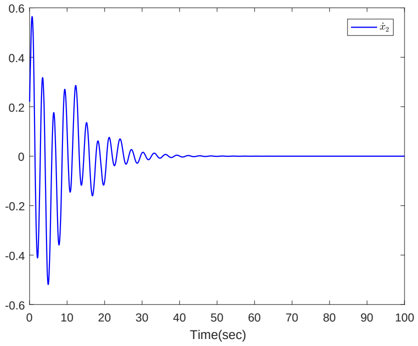


Figure 5. Trajectories of $\dot{x}_2$ in Example1.

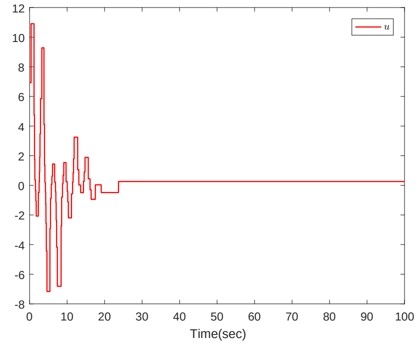


Figure 6. Trajectories of u in Example 1.

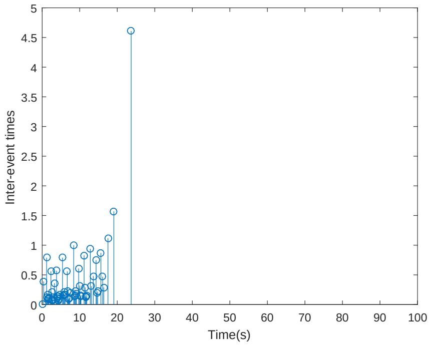


Figure 7. Inter-event times in Example 1.

Financial support and sponsorship

This work was supported by the National Natural Science Foundation of China under Grants 61973148 and Discipline with Strong Characteristics of Liaocheng University: Intelligent Science and Technology under Grant 319462208.

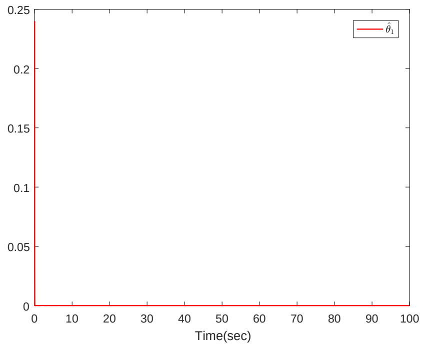


Figure 8. Trajectories of adaptive laws $\hat{\theta}_1$ in Example 1.

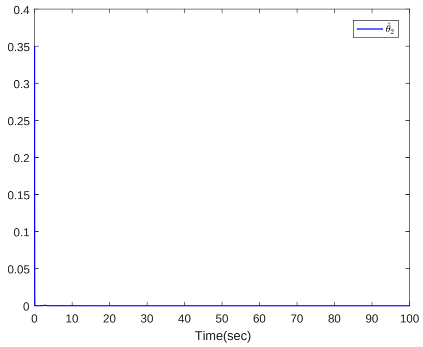


Figure 9. Trajectories of adaptive laws $\hat{\theta}_2$ in Example 1.

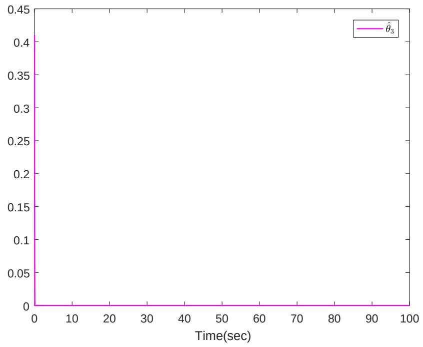


Figure 10. Trajectories of adaptive laws $\hat{\theta}_3$ in Example 1.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Yuan S, Lv M, Baldi M, Zhang L. Lyapunov-Equation-Based stability analysis for switched linear systems and its application to switched adaptive control. *IEEE Trans Automat Contr* 2021;66:2250-6. DOI
2. Shi Y, Sun X. Bumpless transfer control for switched linear systems and its application to Aero-Engines. *IEEE Trans Circuits Syst I* 2021;68:2171-82. DOI
3. Zhang K, Zhou B, Jiang H, Liu G, Duan G. Practical prescribed-time sampled-data control of linear systems with applications to the air-bearing testbed. *IEEE Trans Ind Electron* 2022;69:6152-61. DOI
4. Zhang J, Shao X, Zhang W. Cooperative enclosing control with modified guaranteed performance and aperiodic communication for unmanned vehicles: a path-following solution. *IEEE Trans Ind* 2023;1-10. DOI
5. Zuo Z, Liu C, Han Q, Song J. Unmanned aerial vehicles: control methods and future challenges. *IEEE/CAA J Autom Sinica* 2022;9:601-14. DOI
6. Liu Y, Zhan W, Xing M, Wu Y, Xu R, Wu X. Boundary control of a rotating and length-varying flexible robotic manipulator system. *IEEE Trans Syst Man Cybern Syst* 2022;52:377-86. DOI
7. Hou Z, Xiong S. On model-free adaptive control and its stability analysis. *IEEE Trans Automat Contr* 2019;64:4555-69. DOI
8. Wen G, Chen C, Ge S. Simplified optimized backstepping control for a class of nonlinear strict-feedback systems with unknown dynamic functions. *IEEE Trans Cybern* 2021;51:4567-80. DOI
9. Fei J, Wang H, Fang Y. Novel neural network fractional-order sliding-mode control with application to active power filter. *IEEE Trans Syst Man Cybern, Syst* 2022;52:3508-18. DOI
10. Li X, He J, Wen C, Liu X. Backstepping-Based adaptive control of a class of uncertain incommensurate fractional-order nonlinear systems with external disturbance. *IEEE Trans Ind Electron* 2022;69:4087-95. DOI
11. Li Y, Hou Z, Che W, Wu Z. Event-Based design of finite-time adaptive control of uncertain nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2022;33:3804-13. DOI
12. Ma J, Park J, Xu S. Command-Filter-Based finite-time adaptive control for nonlinear systems with quantized input. *IEEE Trans Automat Contr* 2021;66:2339-44. DOI
13. Huang J, Wang W, Zhou J. Adaptive control design for underactuated cranes with guaranteed transient performance: theoretical design and experimental verification. *IEEE Trans Ind Electron* 2022;69:2822-32. DOI
14. Lai G, Tao G, Zhang Y, Liu Z. Adaptive control of noncanonical neural-network nonlinear systems with unknown input dead-zone characteristics. *IEEE Trans Neural Netw Learn Syst* 2020;31:3346-60. DOI
15. Duan G. High-order fully actuated system approaches: Part I. Models and basic procedure. *Int J Syst Sci* 2021;52:422-35. DOI
16. Duan G. High-order fully actuated system approaches: Part II. Generalized strict-feedback systems. *Int J Syst Sci* 2021;52:437-54. DOI
17. Duan G. High-order fully actuated system approaches: Part III. Robust control and high-order backstepping. *Int J Syst Sci* 2021;52:952-71. DOI
18. Duan G. High-order fully actuated system approaches: Part IV. Adaptive control and high-order backstepping. *Int J Syst Sci* 2021;52:972-89. DOI
19. Duan G. High-order fully actuated system approaches: Part V. Robust adaptive control. *Int J Syst Sci* 2021;52:2129-43. DOI
20. Liu W, Duan G, Hou M. Concurrent learning adaptive command filtered backstepping control for high-order strict-feedback systems. *IEEE Trans Circuits Syst I* 2023;70:1696-709. DOI
21. Lu S, Tsakalis K, Chen Y. Development and application of a novel high-order fully actuated system approach-part I: 3-DOF quadrotor control. *IEEE Control Syst Lett* 2023;7:1177-82. DOI
22. Zhang D, Liu G, Cao L. Constrained cooperative control for high-order fully actuated multiagent systems with application to air-bearing spacecraft simulators. *IEEE/ASME Trans Mechatron* DOI
23. Zhang D, Liu G, Cao L. Proportional integral predictive control of high-order fully actuated networked multiagent systems with communication delays. *IEEE Trans Syst Man Cybern Syst* 2023;53:801-12. DOI
24. Lian Z, Shi P, Lim C, Yuan X. Fuzzy-Model-Based lateral control for networked autonomous vehicle systems under hybrid cyber-attacks. *IEEE Trans Cybern* 2023;53:2600-9. DOI
25. Pan Y, Wu Y, Lam H. Security-Based Fuzzy control for nonlinear networked control systems with dos attacks via a resilient event-triggered scheme. . *IEEE Trans Fuzzy Syst* 2022;30:4359-68. DOI
26. Zhang X, Han Q, Ge X, et al. Networked control systems: a survey of trends and techniques. *IEEE/CAA J Autom Sinica* 2020;7:1-17. DOI
27. Jiang B, Karimi H, Kao Y, Gao C. Takagi-Sugeno model based event-triggered fuzzy sliding-mode control of networked control systems with semi-markovian switchings. *IEEE Trans Fuzzy Syst* 2020;28:673-83. DOI
28. Peng C, Sun H. Switching-Like event-triggered control for networked control systems under malicious denial of service attacks. *IEEE Trans Automat Contr* 2020;65:3943-9. DOI
29. Wang J, Yang C, Xia J, Wu Z, Shen H. Observer-Based sliding mode control for networked fuzzy singularly perturbed systems under weighted try-once-discard protocol. *IEEE Trans Fuzzy Syst* 2022;30:1889-99. DOI
30. Chen G, Sun J, Chen J. Passivity-Based robust sampled-data control for markovian jump systems. *IEEE Trans Syst Man Cybern Syst* 2020;50:2671-84. DOI
31. Chen G, Xia J, Park J, Shen H, Zhuang G. Sampled-Data synchronization of stochastic markovian jump neural networks with time-varying delay. *IEEE Trans Neural Netw Learn Syst* 2022;33:3829-41. DOI
32. Xia J, Chen G, Park J, Shen H, Zhuang G. Dissipativity-Based sampled-data control for fuzzy switched markovian jump systems. *IEEE*

Trans Fuzzy Syst 2021;29:1325-39. DOI

33. Zou Y, Zuo Z, Xia K, Basin M. Saturated Sampled-data distributed control for interval consensus of multi-agent systems. *IEEE Trans on Signal and Inf Process over Networks* 2022;8:1024-36. DOI
34. Huang J, Wang W, Wen C, Li G. Adaptive event-triggered control of nonlinear systems with controller and parameter estimator triggering. *IEEE Trans Automat Contr* 2020;65:318-24. DOI
35. Zhang J, Li S, Xiang Z. Adaptive fuzzy output feedback event-triggered control for a class of switched nonlinear systems with sensor failures. *IEEE Trans Circuits Syst I* 2020;67:5336-46. DOI
36. Zhang J, Li S, Xiang Z. Adaptive event-triggered control of uncertain nonlinear systems using intermittent output only. *IEEE Trans Automat Contr* 2022;67:4218-25. DOI
37. Wang X, Xia J, Park J, Xie X, Chen G. Intelligent control of performance constrained switched nonlinear systems with random noises and its application: an event-driven approach. *IEEE Trans Circuits Syst I* 2022;69:3736-47. DOI
38. Wang X, Xia J, Park J, Xie X, Chen G. Event-Triggered adaptive tracking with guaranteed transient performance for switched nonlinear systems under asynchronous switching. *IEEE Trans Cybern* 2023. DOI
39. Meng R, Hua C, Li K, Ning P. A multifilters approach to adaptive event-triggered control of uncertain nonlinear systems with global output constraint. *IEEE Trans Cybern* 2023. DOI
40. Meng R, Hua C, Li K, Ning P. Adaptive event-triggered control for uncertain high-order fully actuated system. *IEEE Trans Circuits Syst II* 2022;69:4438-42. DOI
41. Yang D, Zong G, Su S. H∞ Tracking control of uncertain markovian hybrid switching systems: a fuzzy switching dynamic adaptive control approach. *IEEE Trans Cybern* 2020;52:3111-22. DOI
42. Chen H, Zong G, Gao F, Shi Y. Probabilistic event-triggered policy for extended dissipative finite-time control of mjss under cyber-attacks and actuator failures. *IEEE Trans Automat Contr* 2023. DOI
43. Chen H, Zong G, Zhao X, Gao F, Shi K. Secure filter design of fuzzy switched cpss with mismatched modes and application: a multidomain event-triggered strategy. *IEEE Trans Ind Inf* 2022. DOI

Intelligence & Robotics

Review

Open Access

Intelligent flood forecasting and warning: a survey

Yue Zhang[1], Daiwei Pan[1], Jesse Van Griensven[1,2], Simon X. Yang[1], Bahram Gharabaghi[1]

[1]School of Engineering, University of Guelph, Guelph ON N1G 2W1, Canada.
[2]Mech and Robotics Engineering, University of Waterloo, Waterloo ON N2L 3G1, Canada.

Correspondence to: Prof. Bahram Gharabaghi and Simon X. Yang, School of Engineering, University of Guelph, Guelph ON N1G 2W1, Canada. E-mails: bgharaba@uoguelph.ca; syang@uoguelph.ca

How to cite this article: Zhang Y, Pan D, Van Griensven J, Yang SX, Gharabaghi B. Intelligent flood forecasting and warning: a survey. *Intell Robot* 2023;3(2):190-212. http://dx.doi.org/10.20517/ir.2023.12

Received: 15 Mar 2023 **First Decision:** 21 Apr 2023 **Revised:** 17 May 2023 **Accepted:** 30 May 2023 **Published:** 28 Jun 2023

Academic Editor: Chen Peng **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

Accurately predicting the magnitude and timing of floods is an extremely challenging problem for watershed management, as it aims to provide early warning and save lives. Artificial intelligence for forecasting has become an emerging research field over the past two decades, as computer technology and related areas have been developed in depth. In this paper, three typical machine learning algorithms for flood forecasting are reviewed: supervised learning, unsupervised learning, and semi-supervised learning. Special attention is given to deep learning approaches due to their better performance in various prediction tasks. Deep learning networks can represent flood behavior as powerful and beneficial tools. In addition, a detailed comparison and analysis of the multidimensional performance of different prediction models for flood prediction are presented. Deep learning has extensively promoted the development of real-time accurate flood forecasting techniques for early warning systems. Furthermore, the paper discusses the current challenges and future prospects for intelligent flood forecasting.

Keywords: flood forecasting, intelligent prediction, supervised learning, unsupervised learning, semi-supervised learning, deep learning

1. INTRODUCTION

Flooding, especially in developing countries, is one major cause of fatalities among both humans and animals. In addition to the loss of life, flooding damages property and destroys crops. Floods can be dangerous to humans and animals, and timely flood forecasting can help prevent or mitigate disasters. Therefore, it is essential to develop prediction models for accurate flood forecasting.

Floods can be defined as the result of water overflowing and submerging land that is normally dry. Flooding is the phenomenon of a rapid increase in the flow of water due to a sudden surge in the volume of water in rivers, lakes, and seas. The main natural factors leading to changes in water volume and levels include heavy rainfall, rapid melt of ice and snow, and storm surges, among others[1]. Generally speaking, the direct economic damage caused by floods to human beings is comparable to that of other natural disasters such as earthquakes and hurricanes. Floods are characterized by a wide range of impacts, extreme destructiveness, and rapidity of disaster[2]. Floods can be categorized into different types based on their causes and severity, including river floods, flash floods, coastal floods, and urban floods. In addition to these categories, floods can be classified based on their severity or frequency. Depending on meteorological conditions and geographical factors, floods are also characterized by elevated seasonal frequency and regional vulnerability.

The global assessment report of the Intergovernmental Panel on Climate Change (IPCC) on natural disasters shows that 313 natural disasters occurred worldwide in 2020, affecting 123 countries and regions[3]. Among them, flood disasters had the highest frequency, occurring 193 times, accounting for 61.66% of the total, and directly affecting a population of 33,215,600[4]. The worldwide direct economic losses caused by natural disasters amounted to US$173 billion, with storms, floods, wildfires, and earthquakes accounting for 99% of the losses. For example, the floods that occurred in Henan, China, in 2020 caused a direct economic loss of US$17 billion, the costliest natural disaster worldwide that year, causing losses to the urban economy, agricultural development, etc.[5]. In some cases, floods can act as triggers for numerous indirect disasters, including geological disasters such as mudslides and landslides, health disasters such as plagues and viruses, and negative impacts on the environment and climate[6]. Such secondary disasters generally have long recovery cycles and high recovery costs, and the hazard of some disasters is irreversible.

Real-time prediction of floods is essential to lay the foundation for mitigating damage to human property and planning defenses[7]. Datasets and machine learning algorithms are two of the most important factors that influence the accuracy of the forecasting result. Firstly, selecting the raw data for flood forecasting involves a careful and systematic process of identifying relevant variables, gathering historical data, preprocessing the data, selecting relevant features, integrating the data, and splitting it into training and testing sets. Then high-quality datasets are critical for building accurate and effective machine learning models. These datasets should be large, diverse, well-labeled, balanced, clean, representative of the problem, and ethically collected. Real-time flood forecasting uses historical data and information from past events to predict flooding. This is because historical data are easier to collect than real-time data. Once processed, historical data (high-quality datasets) are then used to create models for predicting where and when flooding may occur in the future. Therefore, the warnings can be issued to the public so that they can prepare for floods[8,9]. Artificial intelligence (AI) technology has been employed increasingly by governments to create automated flood forecasting systems[10]. At first, supervised learning technology is used for forecasting because the labeled inputs are easier to process and achieve good results. As the need for forecasting increases, there is a growing demand to use unlabeled data, leading to the study of unsupervised learning problems. In some cases, forecasting inputs contain both labeled and unlabeled data, for which semi-supervised learning is of particular interest in order to reach better forecasting. It has been observed that in most situations, conventional supervised learning, unsupervised learning, and semi-supervised learning approaches cannot maintain good performance as the forecasting factors involved become complicated, and the size of forecasting features increases. To overcome this challenge, deep learning approaches have received enough attention recently in the area of forecasting. While machine learning methods have made a great contribution to forecasting, it remains to be quite challenging when applying such methods to natural disaster prediction, such as floods, which could be much more complicated. In the past few years, some primary results have been reported in flood prediction by means of a variety of common machine learning methods. For example, Spatio-Temporal Attention Long Short Term Memory (STA-LSTM) model has great performance in basic flood forecasting, but its robustness and generalization have been improved.[11] Nonetheless, with social development and climate change, there has been increasing natural and human factors affecting flood prediction significantly. Such impacts mainly result from rising temperatures, snow melt, ice melt, rainstorms, and soil conditions, and the main human factors include high urbanization, population explosion, and overexploitation of trees. In addition, simple machine learning cannot be suitable the complex spatio-temporal datasets because natural and human factors are evolving. Furthermore, each flood prediction algorithm has its own characteristics, such as robustness, generalization, calculating speed, gradient problems, weight problems, and fitting problems. Therefore, the purpose of this paper is to provide a comprehensive understanding of the current state of machine learning in flood prediction and to encourage further research in this area.

Based on the above observations, we believe that it is timely and necessary to conduct a review of the current application of machine learning in flood prediction, particularly an abundant literature survey in real-time flood forecasting. While machine learning techniques have been widely applied in various prediction tasks, it remains challenging to obtain stable prediction performance in the area of flood forecasting, especially due to the diversity in the spatio-temporal datasets[12]. Given the large amount of literature available, we specifically analyze and compare the performance of mainstream algorithms currently used in real-time flood prediction. Of particular interest is how these algorithms perform when faced with huge spatio-temporal datasets; it is of great importance to understand whether or not they can still maintain high levels of accuracy and robustness. This aspect has not been shown in previous studies. The contributions of this paper are as follows: (1) Several practical problems and popular machine learning methods used for flood prediction are presented, and their advantages and disadvantages with respect to the computational cost, gradient problem, and robustness and accurateness are discussed in depth; (2) The difficulties in the development of a real-time flood prediction are exhaustively analyzed in terms of the characteristic of flood prediction, spatio-temporal data, and noise. Based on this, a complete literature assessment of recent achievements in dealing with these difficulties is done; (3) The results of this study are summarized, and a number of interesting research trajectories are identified that may help to further advancement of this field.

The rest of the paper is organized as follows. Section 2 introduces the machine learning techniques related to flood prediction. Section 3 addresses several common technical issues that arise in machine learning algorithms designed for flood forecasting. Section 4 includes the comparisons of flood prediction models and some valuable challenges and future work. Finally, Section 5 is the conclusion of this paper.

2. MACHINE LEARNING FOR FLOOD FORECASTING

The performance of a flood forecasting model is heavily influenced by the prediction algorithm used in the model. This section highlights three common challenges in machine learning and deep learning models and presents a model that is well-suited for handling large amounts of data.

2.1. Supervised learning

Supervised learning is a form of AI that can support flood prediction[13,14]. Supervised learning algorithms are used to identify flood levels and give early warning. In the forecasting process, these algorithms are a boon in determining how to use historical data most quickly and accurately to successfully predict future disasters[15–17]. Supervised learning requires input from the user - meaning that they need to distinctly state what they are looking for before they can effectively identify patterns or make predictions about user input data[18]. There are several supervised learning models that can be used to predict flood events: decision trees, k-nearest neighbors, support vector machines (SVM), and neural networks[19]. Each model has its advantages and disadvantages, while SVM is the most commonly used method for flood forecasting[20]. They are easy

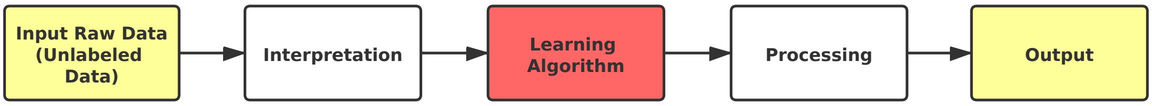


Figure 1. Unsupervised learning block diagram.

to implement and have a high accuracy rate compared to other methods. In addition, they have optimal performance in high-dimensional spaces.

The training dataset for supervised learning has two attributes (x_i,y_i), where "x" stands for the system supervision input and "y" for the system output. Meanwhile, "i" is the training sample indicator for "x" and "y", respectively. In supervised learning, a training input "x_i" is provided to the learning system, which then produces an output "$\widetilde{y}_i$". An arbitrator then calculates the difference between the two outputs for comparison with the ground truth marker "y_i". The discrepancy is referred to as the error signal, which is then transmitted to the learning system to modify the settings of learners. The learning process aims to generate ideal learning system parameters with high accuracy by reducing the difference between "$\widetilde{y}_i$" and "y_i" for all "i"[21,22]. A set of discrete values or a vector space is represented by the input or output. The arbitrator is not subject to any unique limitation under the learning paradigm. "y_i" - "$\widetilde{y}_i$" is typically used to calculate the error signal when "y_i" is taken from a continuous space. The arbitrator typically generates the error signal based on the equality between y_i and "$\widetilde{y}_i$" if "y_i" is one of a set of discrete values. The arbitrator assumes output to be 0 for same "y_i" and "$\widetilde{y}_i$" and 1 for the different "y_i" and "$\widetilde{y}_i$"[22,23].

Flood prediction models rely on historical data gathered from diverse locations worldwide, including Canada, Pakistan, China, India, and Bangladesh[24]. These models are trained with the collected data to forecast future flood events, enabling preparedness measures in the face of such events. This allows for preparedness when a flood event occurs. The supervised learning algorithm is an effective way of accurately predicting future events based on historical datasets[25]. It is especially useful for protecting lives during dangerous floods and other natural disasters.

2.2. Unsupervised learning

Unlike supervised learning, there is no supervision to be given in unsupervised learning; for example, it is usually used to address the datasets that are not labeled. The aim of unsupervised learning is to infer the underlying structure of the datasets provided, in which some evident groups can be identified[26,27].

In unsupervised learning, a computer program learns without any labeled input, as shown in Figure 1. Models for real-time flood prediction rarely incorporate unsupervised learning. This is due to the subjectivity and absence of specific analytical goals, such as response prediction, that characterize unsupervised learning. Evaluation of the results of unsupervised learning techniques is also hard because there is no commonly accepted method for cross-validation or testing the results across several datasets.

2.3. Semi-supervised learning

Semi-supervised learning is a type of machine learning based on supervised learning with semi-supervised data. It is an extension of supervised learning and unsupervised learning. In semi-supervised learning, the algorithm learns the model using labeled data and unlabeled data, as shown in Figure 2. Therefore, semi-supervised learning is a useful tool in situations with a shortage of labeled data[28,29]. As such, unlabeled data can help create better classifiers if there are enough data available and certain distributional assumptions about the data are accurate[30,31]. Moreover, since many machine learning applications exist, developers always try to improve the algorithms by developing new ideas. One of these ideas is semi-supervised learning, which makes it easy to train models for different tasks without having access to all the training data.

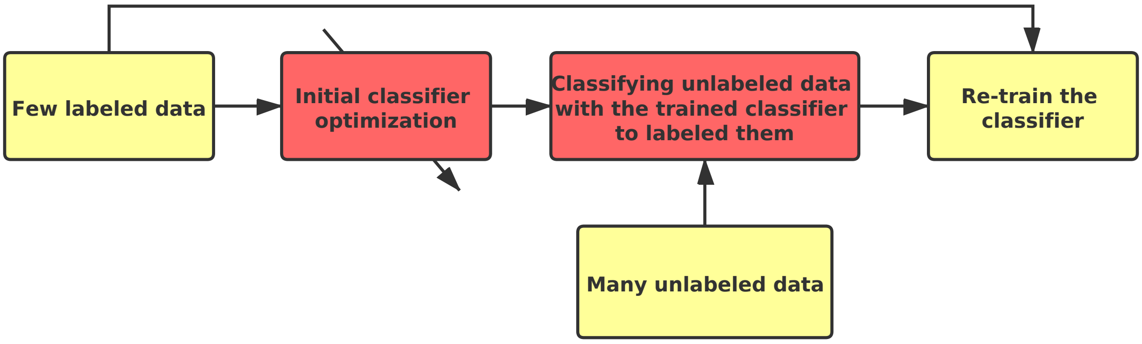

Few labeled data
Initial classifier optimization
Classifying unlabeled data with the trained classifier to labeled them
Re-train the classifier
Many unlabeled data

Figure 2. Semi-supervised learning block diagram.

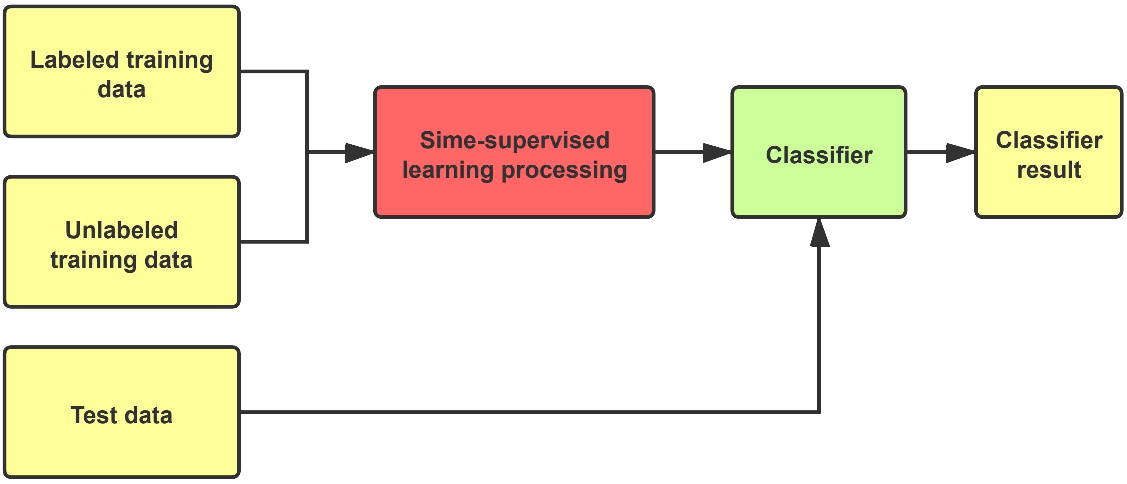


Figure 3. Inductive semi-supervised learning diagram.

According to the statistical learning theory, it can be further divided into inductive semi-supervised learning, as shown in Figure 3 and transductive semi-supervised learning, as shown in Figure 4[32]. The full data collection contains two different sorts of sample sets. Let $D_{Labeled} = \{X_{train}, Y_{train}\}$ denote the labeled sample set and $D_{Unlabeled} = \{X_{unknown}, X_{test}\}$ the unlabeled sample set, and suppose $C_{D_{Unlabeled}} \gg C_{D_{Labeled}}$. For inductive semi-supervised learning, denote $D_{train} = \{X_{train}, Y_{train}, X_{unknown}\}$ the training set, and $X_{unknown}$ and X_{test} are both unlabeled sets and $X_{unknown} \neq X_{test}$. For transductive semi-supervised learning, $D_{train} = \{X_{train}, Y_{train}, X_{unknown}\}$ is the training set, and $X_{unknown}$ is unlabeled and $X_{unknown} = X_{test}$. In other words, the purpose of training the model is just to use the model to classify $X_{unknown}$. In general, the main difference between inductive semi-supervised learning and transductive semi-supervised learning is whether the prediction samples encountered during training are the same as the samples to be classified.

Based on the learning scenario, semi-supervised learning methods can also be classified in accordance with the problems, e.g., classification, regression, clustering, and dimensionality reduction[33,34]. Most studies on semi-supervised learning have focused on the classification, as most machine learning research does. In particular, in this case, there are typically four types of problems: discriminant learning, generative learning, disagreement-based learning, and semi-supervised graph learning[35]. The two typical problems are stated as follows:

1. Generative Learning is an early type of semi-supervised learning that is involved a cyclic process and a

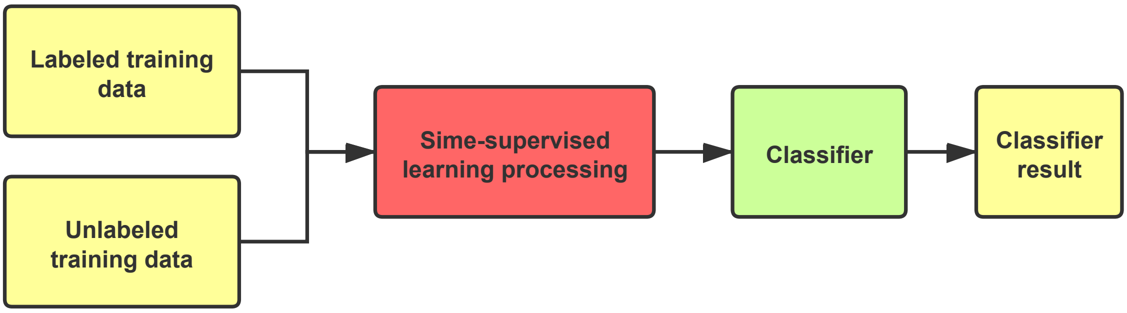


Figure 4. Transductive semi-supervised learning diagram.

maximum likelihood parameter estimation procedure. The two main steps are given as follows:
Step 1: Compute the posterior probability of unlabeled data $P_\theta = (C_1|x^u)$, Initialization is:

$$\theta = \{P(C_1), P(C_2), \mu^1, \mu^2, \sum\} \tag{1}$$

Step 2: Update the algorithm

$$P(C_1) = \frac{N_1 + \sum_{x^u} P(C_1|x^u)}{N} \tag{2}$$

$$\mu^1 = \frac{1}{N_1} \sum_{x^r \in C_1} x^r + \frac{1}{\sum_{x^u} P(C_1|x^u)} \sum_{x^u} P(C_1|x^u)x^u ... \tag{3}$$

Then, back to step 1.
In equation (1), θ means initialization, and the posterior probability of unlabeled data depends on model θ. The N means the total number of examples, and the N_1 means the number of examples belonging to C_1.

2. Low-density Separation (also known as Self-training) works as follows:
   In Figure 5, given the $X_{labeled} = \{(x^r, \hat{y}^r)\}_{r=1}^{R}$ and the $X_{unlabeled} = \{x^u\}_{u=1}^{U}$, we could get the $X_{pseudo} = \{(x^u, y^u)\}_{u=1}^{U}$ after apply the train model to $X_{unlabeled}$ as step 2. Steps 1–4 can be continued until the anticipated class labels from step 2 no longer meet a specified probability threshold or until there are no more unlabeled data.

Due to the wide variety of semi-supervised learning methods, each method has a corresponding dataset due to different principles. To evaluate each method fully, there are many evaluation indicators in semi-supervised learning, including accuracy, true positive rate (TPR), false positive rate (FPR), and Receiver Operating Characteristic (ROC)[37], etc. The accuracy is a very common evaluation indicator, so the TPR, FPR, and ROC are detailed below:

1. True positive rate (TPR): is described as the proportion of true positive outcomes among all positive samples.

   $$TPR = \frac{TP}{TP + FN} = \frac{Ture\ Positive}{All\ Positive\ Case} \tag{4}$$

   A point is said to be True Positive (TP) if this point lies above the Upper Control Limit (UCL) after the damage, which is defined according to a certain damage condition. Conversely, a point is said to be False Negative (FN) if this point lies under the UCL after the damage. Then the UCL signifies the degree of confidence in the training period, as shown in Figure 6.
2. False positive rate (FPR): is described as the proportion of false positive outcomes among all negative samples.

   $$FPR = \frac{FP}{FP + TN} = \frac{False\ Positive}{All\ Negative\ Case} \tag{5}$$

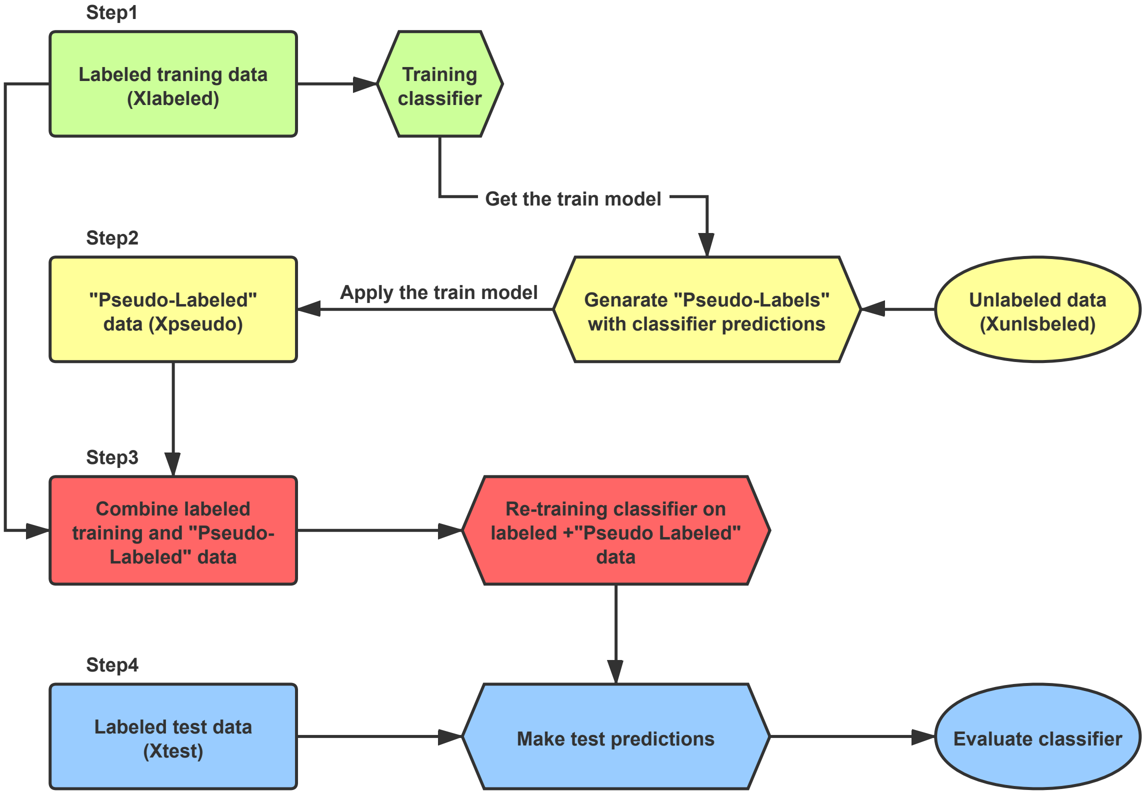

Step1
Labeled traning data (Xlabeled)
Training classifier
Get the train model
Step2
"Pseudo-Labeled" data (Xpseudo)
Apply the train model
Genarate "Pseudo-Labels" with classifier predictions
Unlabeled data (Xunlsbeled)
Step3
Combine labeled training and "Pseudo-Labeled" data
Re-training classifier on labeled +"Pseudo Labeled" data
Step4
Labeled test data (Xtest)
Make test predictions
Evaluate classifier

Figure 5. Self-training flowchart [36].

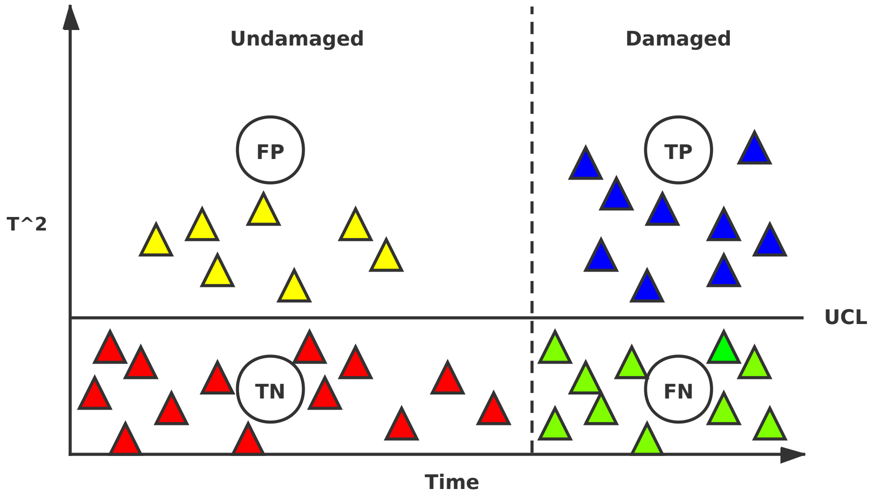


Figure 6. True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

A point is defined to be False Positive (FP) if this point lies above the UCL before the damage, while a point is said to be True Negative (TN) if it lies under the UCL.

3. Receiver Operating Characteristic (ROC): a ROC curve is plotted by the TPR against FPR at various threshold settings. When the ROC curve is closer to the upper-left-hand corner of the ROC space, the model is

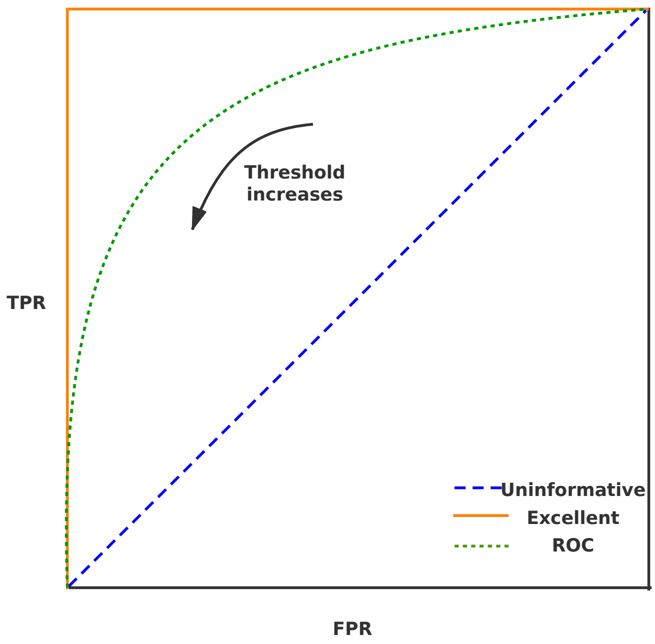


Figure 7. Receiver Operating Characteristic (ROC) curve.

more accurate. On the contrary, when the ROC curve is closer to the uninformative line, the model is less accurate as Figure 7.

2.4. Deep learning

Even though the flood is a common phenomenon because it is a natural disaster that occurs in many countries every year, predicting it is rather difficult. Simple machine learning methods and models are not effective in solving complex flood forecasting problems. As a result, deep learning approaches have been centered by both scientists and governments to solve the flood prediction conundrum[38]. These approaches help them make accurate predictions and save lives. Roughly speaking, deep learning can be viewed as a subset of machine learning, which is usually employed to process a large amount of data. As such, it makes it possible to train powerful computer programs so as to perform specific tasks, as shown in Figure 8. One of the applications of deep learning can be seen in flood prediction, where a deep learning model can be trained to predict the occurrence of floods using the data collected over many years[39]. For this purpose, the National Weather Service (NWS) collects weather data from thousands of local stations across the country. They also collect data from satellite photos aiming to measure the extent of the floods and the damage levels. Using this information, deep models can be created to predict where, how much, and what type of flood conditions will happen subsequently. The results of the prediction are then forwarded to the local governments for pre-emptive planning purposes. This allows authorities to respond quickly to unexpected floods by shutting down streets or opening up new paths through flooded areas.

Although humans are constantly faced with a large amount of perceptual data, they can always obtain important information that is worth noting in a deft way. Imitating the efficiency of the human brain in accurately representing information has long been a core challenge in AI research. Research on the mammalian brain was based on anatomical knowledge: the time when sensory signals traveled from the retina to the prefrontal cortex, which then transferred to the motor nerves, was used to determine how the brain represented information. Inferences were made regarding the cerebral cortex not directly displaying the data but allowing the received stimulus signals to pass through a lamellar network model and then obtaining the observed rules.

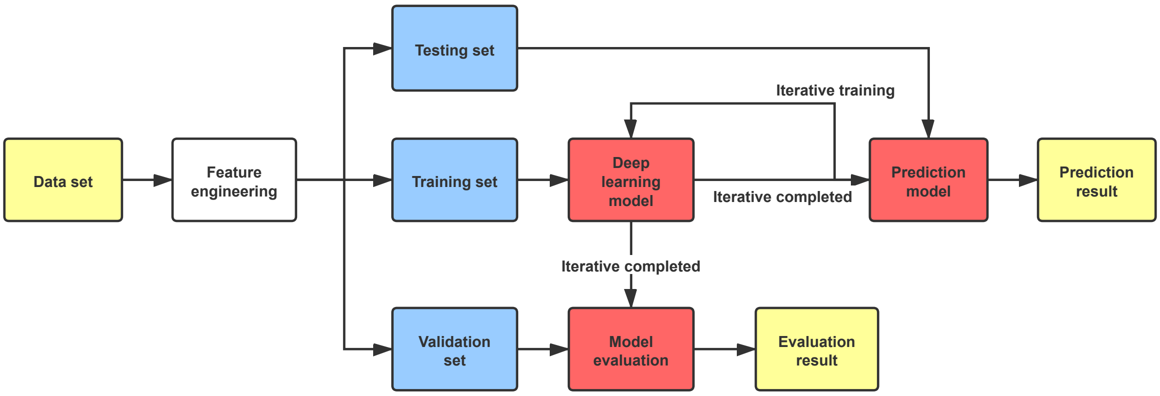


Figure 8. Deep learning block diagram.

Thus, the human brain does not project images directly onto the retina but rather processes information by aggregating and decomposing it to recognize objects. In order for the visual cortex to function properly, it must reproduce the image on the retina and feature the perceptual signal[40]. By retaining useful structural information about objects and reducing the amount of data processed by the visual system, this hierarchy of human perception reduces the amount of data that the visual system processes. In addition to capturing essential features in structurally rich data, deep learning can also capture potentially complex structural rules, such as natural images, videos, and voice[41]. In the field of machine learning, deep learning is a promising research direction that is aimed at getting machine learning closer to AI. As part of deep learning, the internal rules and representation levels of sample data are learned[42], and the information gained from this process can be used to interpret text, image, and sound data. Deep learning is an algorithm that has achieved far more in speech and image recognition than previous methods related to machine learning. The ultimate goal is for machine learning to be able to recognize words, images, and sounds in a manner similar to humans.

Models for deep learning include convolutional neural networks (CNN), deep reinforcement learning[43], and stacked autoencoder networks[44,45]. In the neural and cognitive machine of Fukushima, the first computational model of CNN is proposed, inspired by the visual system[46,47]. To obtain a translation invariant neural network structure form, neurons with the same parameters are applied to different positions of the previous layer based on the local connections between neurons and hierarchical image transformation. As a result of this concept, Le Cun *et al*. developed CNN with error gradients[48], which improved pattern recognition performance. Historically, CNN has shown extraordinary performance, such as the handwritten character recognition[49].

The Deep Belief Networks (DBNs) are a type of Bayesian probabilistic generation model[50,51], which can be trained to generate inputs in a probabilistic way. The resulting trained layers can even be used as a feature detector, which is able to be further trained to solve classification problems. For DBNs, a stack of Restricted Boltzmann machines (RBMs) is used, which sometimes consists of two layers of the recurrent neural network (RNN), i.e., the visible layer and hidden layer. Such layers are undirected; thus, an efficient, fast training capability can be obtained when performing the unsupervised procedure. It is observed that due to the network structure of the DBNs, such models can be trained by a greedy learning strategy layer-by-layer. As a result, the DBNs have been found to have a broad range of applications in practice[52,53].

Similar to DBNs, stack autocoding networks consist of several layers of structural units[54]. However, the structural units in the autocoding model are auto-encoders instead of RBMs[55]. In the autocoding model, there are two layers, i.e., an encoding layer and a decoding layer[56]. An effective method for building multi-layer neural networks based on unsupervised data was proposed by Hinton in 2006[57]. A single layer of neurons is constructed in two steps: first, a single layer is built layer by layer, which means a single layer of the network is trained every time; then, the wake-sleep algorithm is used to tune all layers. A bidirectional weighting of all layers except the topmost is used, so the topmost layer remains a single-layer neural network[58], while the other layers become graph models. Use the wake-sleep algorithm to adjust all weights, with the upward weight for ”cognition” and the downward weight for ”generation”. The topmost representation generated must be capable of restoring the underlying node as accurately as possible to ensure cognition and generation agree[59]. If a node represents a face at the top, then all face images should activate this node, resulting in an image that can be used to represent a general image of a face.

Currently, to improve the performance of real-time flood forecasting, we could use the Fully Convolutional Network (FCN), Long Short Term Memory model (LSTM), Gated Recurrent Unit Network (GRU), Graph Convolutional Network (GCN), Generative Adversarial Networks (GAN), CNN-Long Short Term Memory Network (CNN-LSTM), and STA-LSTM. Each model or algorithm has a different focus on hydrological prediction and, therefore, has a different effect when we focus on different kinds of flooding.

1. **Fully Convolutional Network (FCN):** A FCN can mainly solve the image segmentation technology from the semantic level. After sampling the feature map of the convolutional layer, the FCN can restore it to the same size as the input image to make pixel predictions. FCN and Multioutput FCN are quite useful in solving problems related to spatial data but do not encode the position and orientation of applied objects. The image-based model of FCN can capture the basic structure of the problem, but it has higher requirements on time and space complexity[60].
2. **Long Short Term Memory (LSTM)/Gated Recurrent Unit Network (GRU):** LSTM networks are designed to have a large short-term memory, allowing for more efficient and resource-rich training on datasets of sequential samples. As a consequence of a series of activation functions and processes, LSTM neurons produce two different values, as compared to a convolution operation generating one output and holding it to urgent receptors in both the next and same layer[61]. While both outcomes are kept within the LSTM layer to keep a record of what was learned in the preceding part of the sequential manner, one is transmitted to the next layer[62]. GRU and LSTM have comparable performance, and the GRU is the variant of LSTM. However, the parameter of GRU is less, and the training speed is faster than LSTM because the input gate and the forget gate of LSTM are combined with the update gate and the cell state, and the hidden state of LSTM is combined in GRU. The GRU has advantages over LSTM in forecasting when the dataset is huge.
3. **Graph Convolutional Network (GCN):** Graph Neural Network (GNN) refers to the application of neural networks on graphs. According to the classification of propagation mode, GNN can be divided into graph convolutional neural network (GCN) and graph attention network (GAT). GCN can efficiently extract features from non-euclidean structure data, which is prevalent in a large amount of data types lacking regular structure.
4. **Generative Adversarial Networks (GAN):** The GAN introduces the concept of adversarial learning[63]. The concept is introduced between the generator and the discriminator. GAN, as presented by Goodfellow *et al.* (2014)[64], presents two separate CNNs that work in unison but are seemingly separate at the same time. It resembles that they are competing against one another in a min-max game. Here, one of the CNN will aim to generate fake examples from the given dataset. Another CNN acts as the discriminator. The role of the discriminator is to test whether the dataset is real or not. Since both the CNNs are working against one another and the training is better, it makes the CNN get better over time. Moreover, Markov models usually compute at a slow speed and might have a high degree of inaccuracy; the GAN can work with high-volume complex data in a faster manner[65].
5. **Convolutional Neural Network LSTM (CNN-LSTM):** The main feature of CNN is the convolution operators. CNN is suitable for processing spatial data because it consists of a convolutional layer and a pooling layer. The convolution layer can maintain the spatial continuity of the image and extract the local features

of the image. The pooling layer can use max-pooling or mean-pooling, reducing the dimension of the middle-hidden layer and the amount of computation. Moreover, LSTM is suitable for processing temporal data. Therefore, CNN-LSTM has better performance than LSTM for spatio-temporal prediction problems.

6. **Spatio-temporal attention LSTM (STA-LSTM):** STA-LSTM includes the main LSTM network and Spatial Attention and Temporal Attention. The main LSTM network is used for feature extraction, spatial-temporal correlation utilization, and final prediction. We can adjust the attention weights dynamically, and the performance of LSTM cells can be improved due to Spatial Attention and Temporal Attention.

Then relevant experimental verification should be conducted appropriately for flood forecasting models. Experimental verification is an essential step in evaluating the performance of a flood forecasting model and ensuring that it is accurate and reliable. The previously unused data (test dataset) can be used to validate the accuracy of the model and use different evaluation metrics to assess its performance. In general, flood forecasting belongs to a regression task, for which evaluation indexes are usually important. Thus, a few commonly used indexes are listed, including Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), R Squared, and Nash-Sutcliffe Efficiency (NSE). Relevant experimental verification can and should be conducted appropriately for flood forecasting models. Experimental verification is an essential step in evaluating the performance of a flood forecasting model and ensuring that it is accurate and reliable[66,67].

1. **Mean Squared Error (MSE):** MSE is the expectation of squared error, that is, the average squared deviation between the prediction values and the observation values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2 \tag{6}$$

In equation (6), O_i means the observation value, P_i means the prediction value, $\bar{O}_i$ means the average observation value and n means the number of observations. Usually, the MSE could be led to the dimension issues.

2. **Root Mean Square Error (RMSE):** RMSE shows the degree of dispersion of the samples and assesses how well the predicted value matches the observed value. RMSE can avoid the dimension issues compared with MSE. If the machine learning problem is sensitive to the dimension issues, the RMSE could be used as an evaluation index of model performance. Since RMSE is the sum of squared errors and then takes the square root, it would be possible to magnify the gap for the larger errors.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \tag{7}$$

3. **Mean Absolute Error (MAE):** MAE is the expected value of the absolute error loss, that is, the average deviation between the predicted value and actual value. Similar to RMSE, the value of the MAE should be optimized as low as possible too. Since the MAS directly takes the absolute value of the error, the MAS reflects the real error.

$$MAE = \frac{\sum_{i=1}^{n}|O_i - P_i|}{n} \tag{8}$$

4. **R Squared (R^2):** R^2 is called the coefficient of determination. The performance of the model cannot be accurately judged when the evaluation index does not have the upper and lower limits, while R^2 is able to take this into account, confining the index value into a certain bound.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O}_i)^2} \tag{9}$$

5. **Nash-Sutcliffe Efficiency (NSE):** A normalized statistic, called the NSE, can evaluate the performance of the model to measure variables and can display the percentage of the initial variance of the model.

$$NSE = (1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O}_i)^2}) \times 100 \tag{10}$$

Note that when models deal with regression problems, if the models have higher performance, the values of the MSE, RMSE, and MAE are as low as possible, but the R^2 and NSE are both nearly 1.

3. PROBLEMS AND CHALLENGES OF DIFFERENT ALGORITHMS FOR FLOOD PREDICTION

After analyzing the machine learning working mechanism, this section aims to understand the distinctions and problems between different machine learning methods and propose corresponding models to solve the problems. The researchers must comprehend the characteristics and performance of every machine learning approach while selecting models. Understanding the drawbacks of each algorithm, in particular, can help us better pick models for flood prediction.

3.1. Problems and challenges in supervised learning

When an algorithm or model is proposed, the following aspects need to be considered in order to obtain a good learning performance:

1. **The trade-off between bias and variance:** To generate more precise prediction results, We make a trade-off between bias and variance[68]. Generally, bias and variance are a trade-off in a learning algorithm, so a lower bias learning algorithm needs to be flexible so that it can match the data well. However, it is likely that if the learning algorithm is too flexible, it will match every training dataset, which results in a high variance. Therefore, most supervised learning methods offer a bias/variety parameter that the user can adjust so that they can adjust this trade-off between bias and variance. For flood forecasting, the bias (accuracy) would be sacrificed for lower variance (stability) because lower variance means better robustness and generalization ability.
2. **The complexity of the function and the amount of training data:** The second issue is the amount of complexity of the training data relative to the ”real” function (classification or regression)[69]. With a small amount of data, an ”inflexible” learning algorithm can be used to learn a simple function with a high bias and low variance[70]. The function, however, will only be trained from a very large number of data, and the use of ”flexible” learning algorithms has a low bias and variance if it involves many different input elements in a complex interaction and it has behavior in various parts of the input space[71]. Based on data availability and perceived complexity of a function, good learning algorithms automatically adjust bias/variance trade-offs.
3. **The dimension of the input space:** There is also the problem of the input space dimension[72]. Even if the true function only depends on a small number of input features, the learning problem is difficult if the input feature vectors have high dimensionality. As a result, high input dimensionality usually requires adjustments to classifiers with low variance and high bias. This is because many of the ”extra” sizes can be confounded by the learning algorithm and make it have a high variance. Engineers can improve the accuracy of this learning function by manually removing irrelevant features from the input data. Further, many algorithms are available for choosing features that are relevant and ignoring those that are not.
4. **Output value in noise:** The fourth issue is the level of noise at the desired output values (monitoring the target variables)[73]. If the desired output value is usually incorrect (because of human or sensor errors), the learning algorithm should not try to find a training example that exactly matches those values. Trying to fit the data too cautiously leads to overfitting[74]. When there is no measurement error (random noise), if you are struggling to learn the function, it is your learning pattern itself that is too complex, potentially leading to overfitting.

3.2. Problems and challenges in unsupervised learning

Unsupervised learning is not commonly employed in real-time flood prediction models[75] due to its susceptibility to subjectivity and lack of clear analytical objectives, such as response prediction. Furthermore, there is no widely accepted approach for performing cross-validation or validating the outcomes of unsupervised learning techniques on different datasets, making it difficult to evaluate their effectiveness[76,77].

However, Chen *et al.* (2022) suggested the Flood Domain Adaptation Network (FloodDAN) strategy, which combined adversarial domain adaptation and extensive pretraining to create a model for unsupervised flood forecasting[78]. They undertook adversarial domain adaptation between the two datasets after pre-training the source model on large-scale datasets. The final model was built using the source prediction head and target encoder produced in the first and second stages, respectively. The findings of the experiment demonstrated that FloodDAN could make flood predictions using simple rainfall data. In addition, Chen *et al.* (2022) compared the performances of fully supervised learning and unsupervised learning in flood forecasting. In fully supervised learning, researchers used the entire training set for supervised training and compared the evaluation findings to determine the optimum model structure for flood prediction. They computed the lower bound of the unsupervised learning method using the historical runoff input as the model output. The findings demonstrated a typical issue in supervised learning, which was when the amount of data diminished, the performance of the model degraded. However, the FloodDAN unsupervised learning model may perform at the same level as supervised learning which took 450–500 hours, which is very significant and valuable for flood forecasting in regions without hydrological data.

3.3. Problems and challenges in semi-supervised learning

While semi-supervised learning has found extensive use in flood prediction, there remain certain challenges in its implementation process. Specifically, two key aspects warrant consideration:

1. **Sample division:** Semi-supervised learning divides samples into unlabeled and labeled samples, which is important for building models[79]. In the process of model training by predicting labeled samples and then building a new model, the process is cyclic iteration; and in the process of prediction, it is necessary to consider that the sample prediction is wrong, which will greatly affect its generalization performance[80] and even lead to performance degradation. Building a model that can guarantee prediction accuracy without over-labeling is an important challenge in semi-supervised learning[81].
2. **Selection of semi-supervised learning methods:** Semi-supervised learning is not easy to build better learning programs from unlabeled data. As mentioned earlier, unlabeled data are useful if and only if they contain information useful for predicting labels, which do not contain data that cannot be extracted or are difficult to extract in labeled data. Semi-supervised learning methods can be applied in practice to provide effective information for model building, which further leads practitioners and researchers to the question of when such situations occur. It is currently difficult to precisely define the conditions for any particular semi-supervised learning job, and it is not easy to assess the extent to which these conditions are met by the methods that can be used. This question has so far been left unanswered. Research on this problem stops at inferring the applicability of different learning methods to various types of problems[82], such as graph-based methods[83] that apply local similarity measures to construct graphs of all data points[84].

3.4. Problems and challenges in deep learning

While deep learning methods have decent fitting ability and robustness, there are still some problems that cannot be overlooked.

1. **Expensive computation and low portability:** Deep learning is expensive since it takes a lot of data and processing power. Additionally, a lot of programs are still incompatible with mobile devices. Many businesses and teams are now working on creating semiconductors for portable devices[85].
2. **Strict hardware specifications:** Deep learning demands a lot of computational power, which is something that standard CPUs can no longer handle[86]. GPU and TPU are the primary components used in mainstream computing; therefore, both the cost and hardware requirements are relatively high.
3. **Intricate model design:** Deep learning model creation is quite complicated, taking a lot of time, labor, and material resources to create new algorithms and models. Most individuals are limited to using pre-made models. For example, the STA-LSTM and CNN-LSTM can somewhat resolve the robustness and gradient issues, and GRU is able to perform calculations more quickly than LSTM[87,88].

Table 1. Comparison of machine learning algorithms in the flood forecasting

Algorithms	Computational cost	Other benefits	Disadvantages
SVM	High on large dataset [91]	Effective in handling high-dimensional data; robust to noise; accurate in nonlinear problems; good for small datasets	The input data needs to be normalized. Sensitivity to kernel choice; parameter tuning difficultly
ANN	High on large dataset	Capture complex nonlinear relationships; better robustness; the ability to learn and adapt to changes in the input data; better generalization	Overfitting easily; parameter tuning difficultly; poor interpretability; sensitivity to input data
CNN	High on large dataset	Spatial and temporal feature extraction ability; No pressure for high-dimensional data processing; Robustness to noise;	Limited applicability; need to adjust parameters and a large and high-quality sample size
FCN	Higher than CNN	handle high-resolution data; learn complex spatial patterns: handle missing data and noise; computationally efficient	Need a large amount and high-quality of labeled data for training; overfitting; not explicitly model temporal dynamics [60].
GCN	Higher than ANN and CNN	Deal with non-Euclidean data; learn the spatial dependencies between nodes; deal with missing data and noise in the input data; computationally efficient; handle large graphs [92]	Loss of spatial information; requires a well-defined graph structure; not explicitly model temporal dynamics
LSTM	Higher than CNN and ANN	Solved Gradient problem; model temporal dependencies and capture long-term dependencies in time-series data; handle missing data and noise; computationally efficient; handle large datasets [61]	With the increase of time, the modeling ability of LSTM begins to decline due to the gradient problem; overfitting; struggle with handling sudden changes in the data distribution; not explicitly model spatial dependencies [93]
GRU	Lower than LSTM	Alleviate the gradient problem; high memory efficiency; higher computationally efficient; capturing temporal dependencies; good generalization ability [94].	Gradient problem as LSTM; training difficultly; require careful tuning of hyperparameters; sensitivity to initialization; overfitting; limited interpretability
CNN-LSTM	Higher than LSTM	Spatial and temporal feature extraction; improved accuracy; better robustness; better generalization	Overfitting; gradient problem; computational complexity
STA-LSTM	Higher than CNN-LSTM	The high robustness and generalization of hydrological prediction, and the prediction accuracy better than most forecasting models [95]	lower computational efficiency than CNN-LSTM; carefully tune each attention weight; the identification of important spatio -temporal features [11]

4. COMPARISONS AND FURTHER DIRECTION FOR FLOOD FORECASTING

As a result of our determination of the requirements for choosing the right machine learning algorithm, we have listed them below in descending order, starting with the most important and going down to the least important in the prediction of a flood as the following: accuracy, robustness, generalization, low configuration, low preprocessing of data, and insights into factors influencing prediction.

4.1. Characteristics of machine learning algorithms for flood prediction

As many machine learning models have been used for flood prediction, we select some representative models to compare. Table 1 shows the comparison between various machine learning algorithms used in flood prediction. Firstly, SVM has good performance on both low-dimensional and high-dimensional datasets and has certain robustness[89]. Nonetheless, due to the high computational complexity, SVM is usually not efficient for the case when the training datasets are rather large. The input datasets of the flood forecasting model not only have a very large size but also have a relatively high dimension[90]. Therefore, SVM seems to be inadequate to deal with the flood forecasting problem. Moreover, the flood forecasting problem is a nonlinear regression problem; an overfitting issue would more likely occur when the SVM deals with nonlinear problems.

In Artificial Neural Networks (ANN), 3-layer neural network models are employed as the function estimators in order to forecast the flood. Unlike conventional physical models described by certain differential equations, ANN models are able to be trained from the historical data, after which the future trend of the flood can be predicted through the obtained models[96,97]. In this procedure, the actual hydrological datasets are utilized for training the ANN models; the connecting weights of each neuron are adjusted accordingly in order to fit well the relationship between the occurrence of the flood and related impact factors[98]. Clearly, the quality of the provided dataset is one of the most influential features for the ANN performance[99]. Due to the fact that the real-time datasets of flood forecasts are sometimes subject to the large noise, data preprocessing is quite necessary in order to improve the ANN models. The data cleaning, normalization, and transformation are good preprocessing techniques that are commonly employed to enhance the quality of the dataset[90,100]. Besides that, overfitting is another common issue in ANN model training, particularly when the models used are overly complex or the size of the dataset is relatively small. To this end, the dropout regularization and weight decay are frequently utilized to prevent overfitting, thus improving the generalization ability and robustness[101]. Notice that basic ANN architecture merely includes three layers: input layer, hidden layer, and output layer, and usually, such a 3-layer structure cannot achieve the flood forecasting with high accuracy.

The CNN is a kind of ANN architecture trained with deep-learning algorithms[102]. CNNs are well suited for solving prediction problems with spatial datasets. This is because CNNs are designed to handle the spatial structure of data and can learn to extract features hierarchically from the input data. The accuracy of a CNN depends on several aspects, such as the size and complexity of the model architecture, the quality of the training data, and the choice of hyperparameters[103]. CNNs can be robust in flood forecasting, particularly when trained on a diverse range of data that captures variations in environmental conditions and flood events. This can help the model to identify features that are associated with flooding and to generalize well to new and unseen data. Therefore, CNNs can be made less sensitive to changes in the input data, such as changes in weather conditions or changes in the landscape, which may possibly affect the accuracy of the model. As with most machine learning methods, overfitting may occur as well when the model employed is too complex or the training data is too limited, leading to poor generalization performance on new datasets. Similar to ANN, overfitting would be prevented by the appropriate regularization techniques. Furthermore, the CNN models can be combined with other models to improve the performance of flood forecasting.

The FCN, as a specific type of CNN architecture, has been designed for image segmentation tasks. FCNs and CNNs can be adopted for different types of tasks in flood forecasting due to their different characteristics. The accuracy, robustness, and generalization of FCNs are much similar to CNNs[104]. FCNs may sometimes require more computational resources than CNNs owing to their architecture, which could be a limitation for real-time flood forecasting applications. Techniques such as the use of a smaller model architecture or transfer learning can be employed to reduce the computational cost of training an FCN. Mu (2022) trained the models of FCNs, multi-output FCNs, and their RNN variants in watersheds with frequent rainfall and performed both quantitative and qualitative analysis for specific rainfall events[60]. In cases where the predicted water depth exceeded 50 cm, the multi-output FCNs had a very significant advantage. Under the recurrent effect, the accuracy of long-term flood prediction of FCNs will be greatly improved. In the process of this test, the three predicted NSE values of the FCN model were 81.0%, 79.95%, and 78.16%, respectively. From this result, this model had a certain accuracy for predicting floods, but the values were not particularly close to 1. The single FCN model here still has a lot of room for optimization. FCNs and CNNs are both useful in flood forecasting, but their specific strengths and weaknesses depend closely on the type of data and task at hand.

GCNs are a class of deep learning models that can effectively handle graph-structured data, making them promising tools for flood forecasting. GCNs have good robustness. For example, they can deal with noisy data and missing values by learning from the local connections in the graph, allowing the model to smooth out noise in the data. GCNs can also effectively adapt to changes in the data by dynamically updating the weights of the connections in the graph, enabling the model to learn from new data without requiring a full retraining[105]. Notice that GCNs are mainly designed to handle graph-structured data, which makes them well-suited to handle complex spatial relationships in flood forecasting data. GCNs may struggle to generalize to new and unseen data outside of the training set, especially when the graph structure is significantly different from the training data. This can lead to overfitting and reduce performance in real-world flood forecasting applications. GCNs may be computationally expensive, especially for large graphs with many nodes and edges. This can make it challenging to scale up GCN-based flood forecasting models in order to cover larger geographic areas. Then Graph convolutional RNN (GCRNN) is a kind of neural network based on GCN. The GCN is used to capture image data to represent spatial relationships, while the RNN is used to capture temporal data[87]. The combination of these two structures gives this model an advantage over time and space. GCRNN can be used to predict the time series of water quantity in a geographic area or water supply area. GCRNN is an image-based model that captures water volume in both time and space. Zanfei *et al.* (2022) have tested this prediction model in the presence of sensor failures[92]. The test results showed that GCRNN can accurately predict floods when considering the spatial criteria of time series. Especially in fault testing, GCRNN performed much better than LSTM. Since GCRNN has relatively high complexity and the demand for computing time is also high, there are still some deficiencies on the whole. While GCRNN is able to predict the whole flood cycle reliably and stably, the modeling ability begins to decline with the increase of time. From this point of view, GCRNN still has higher advantages. In the model test of the three time periods in the Chongqing section of the Yangtze River Basin, the GCRNN model has significantly higher accuracy than the FCN model since the NSE results of the GCRNN in these three tests are 93.10%, 89.33%, and 83.56%. These three values are larger and closer to 1 than the respective values of FCNs.

LSTM is a type of RNN, and the gradient problem of RNNs can be partially resolved by LSTM models. LSTM models have been shown with high accuracy in flood forecasting compared to traditional statistical models. This is because LSTM models are able to capture the complex temporal dependencies and patterns for the time series data[106]. LSTM models have also been shown with the ability to effectively model the nonlinear and non-stationary behavior of hydrological variables, which is important in flood forecasting[107,108]. However, the accuracy of LSTM models can be affected by several factors, such as overfitting, underfitting, and the presence of outliers or anomalies in the data. The quality and quantity of input data can significantly impact the robustness of an LSTM model. The architecture of an LSTM model, including the number of layers, hidden units, and input/output dimensions, can impact its robustness too. Moreover, the choice of training parameters, e.g., learning rate, batch size, and optimization algorithms, may also impact the robustness of an LSTM model. The use of appropriate training parameters can improve the ability of the model to generalize to new data. LSTMs performed very well for one-day, two-day, and three-day forecasts. The NSE for these three predictions were 95%, 93%, and 88%, respectively, which are better than FCNs and GCNs.

GRU and LSTM have comparable performance, whereas GRU has a lower computational cost than LSTM. The NSE values of GRU in the three predictions are nearly 95%, 93%, and 88%, respectively. Some researchers combined GRU and CNN models to create the Convolutional GRU (CONV-GRU), aiming to maximize the benefits of both models[88]. The combination is accomplished by connecting the output of the CNN model with the input of the GRU model. This model was used to forecast the water level and flood phenomena of a Taiwanese river. This model was thought to be an extension of the GRU model. When compared to other neural network models, the CONV-GRU model was found to be superior to other neural network methods in predicting water level characteristics, and it was very useful in detecting local flood characteristics. The CONV-GRU model can detect both normal and abnormal time series behavior. Furthermore, the error between the predicted and actual value of this model is relatively small, and the CONV-GRU model is superior to the four models of LSTM, CNN, ANN, and Seq2seq (Sequence to sequence).

Although the applications of machine learning techniques in flood prediction have been increasing, most of them are based on a collection of one-dimensional data. The CNN-LSTM model uses the two-dimensional radar map to calculate the runoff and then achieves the purpose of flood prediction[109]. The two-dimensional processing of the precipitation radar map by CNN and the one-dimensional processing of the sequence by LSTM makes it possible to figure out the upstream and downstream flow series[86]. The database in three

Table 2. Measures of machine learning algorithms

Measures	Algorithms
High accuracy	STA-LSTM [11,93]
Good robustness	SVM, STA-LSTM
Good generalization	SVM, STA-LSTM, GRU
Low computational cost	SVM, ANN, CNN
Gradient problems	CNN, LSTM, GRU
Fitting problems	SVM, ANN, CNN, FCN, GCN
High dataset quality	SVM, CNN, Hybrid LSTM, GRU

different years was taken as the study period. After forecasting at three different water level periods, the NSE values were 93.51%, 94.25%, and 95.18%, respectively. Comparing this set of results with the NSE values of the LSTM, it was clear that the CNN-LSTM predictions were more accurate. After using the NSE and evaluating the performance of the model, it is found that the better prediction performance of CNN-LSTM highly depends on the optimized input dataset[110]. It means while models can be useful for estimating water levels and issuing flood warnings, high-quality datasets are fundamental to their value.

It is observed that most prevailing deep learning structures, such as LSTM networks, are unlikely to model the spatial correlation of hydrological data and, thus, fail to produce satisfactory prediction results[95]. It is clear that flood is uncertain and highly nonlinear, which leads to the low robustness of hydrological prediction[111]. The physical interpretability of the models is easily overlooked by general machine learning methods. Some scholars proposed the STA-LSTM model based on the original LSTM, which contains an explainable attention mechanism. Experiments showed that this model outperformed FCN, CNN, GCN, and the conventional LSTM in most cases. The rationality of this model is mainly reflected by the visual and interpretable weight of spatial and temporal attention. When carrying out modeling experiments, Lyu *et al.* (2021) found that in the STA-LSTM model, spatio-temporal attention and time have similar weights, and they all move forward slowly. In terms of the overall weight, its change trend is not obvious and relatively slow[112,113]. However, in the case of actual flood input, with the extension of the forecast time, the time attention weight also keeps moving forward, and a certain time point will have a synchronous forward speed with the predicted speed. This implies that the time attention weight is very similar to the whole fusion process[114]. Moreover, if the weight of time attention deviates, it is highly likely to cause the deviation of the prediction results. From this perspective, although the STA-LSTM model has better performance than other neural networks in terms of datasets and performance, it is greatly affected by the weight of time attention[115]. The three values of NSE in this evaluation were 97.03%, 96.73%, and 95.10%, respectively, which shows STA-LSTM had the highest accuracy in predicting floods.

All in all, the high-quality datasets, the great quantity of data, concise model structure, suitable hyperparameters, and data preprocessing are important for the accuracy, robustness, and generalization of flood forecasting models. As a result, the selection of a machine learning algorithm for flood forecasting should be based on the unique requirements of the application and the properties of the available data.

4.2. Future direction for real-time flood forecasting

Although the challenges of flood forecasting have been partially addressed by current machine learning methods, there are still challenges not completely being solved in flood forecasting, as shown in Table 2. The vanishing gradient problem is a common challenge in deep learning-based flood forecasting models[116]. LSTM models can relieve the vanishing gradient problem of RNN and CNN models, but it also occurs when the gradient of the loss function becomes very small or very large, making it difficult for the optimizer to update the model parameters effectively during training. Various techniques, such as weight initialization, batch normalization, gradient clipping, and data augmentation, can be used to address the gradient problem and improve the performance of these models[117].

An ideal flood forecasting model should exhibit low bias and low variance to accurately predict true values while not being overly sensitive to minor input data changes[118]. However, striking this balance can be challenging, especially when dealing with complex flood systems that are influenced by a multitude of factors and variables. The variance impacts the robustness and generalization of the flood forecasting model, while the bias affects its accuracy[119]. Therefore, the balance between variance and bias needs to be adjusted according to the specific forecasting requirements.

A lot of flood forecasting models are sensitive to the input data changing. Because flood events can be affected by various factors, such as weather patterns and land-use changes, future work could focus on developing machine learning models that can adapt to changing conditions and provide accurate predictions under different scenarios. In other words, the flood forecasting models of high generalization stability have to be given more attention. For example, the hybrid CNN models could be created since CNN models are hyposensitive to the input data changing[120,121].

Notwithstanding the fact that machine learning models have flexibility, speed, and simplicity with physical models, their generalization ability is restricted[122]. The physical models are based on the underlying physical laws and can generalize well to different conditions and locations. Therefore, hybrid models that combine the strengths of machine learning and physical models can improve flood prediction accuracy and generalization.

The quality and quantity of the dataset will have a large impact on the accuracy of the flood forecasting models. In some regions, data scarcity can limit the effectiveness of deep learning models. Developing methods to address data scarcity is imminent, such as transfer learning and data augmentation. The concept of GAN-LSTM is to generate synthetic flood data that can be used to augment the training data of an LSTM model. GANs can generate realistic and diverse samples from a given distribution, while the LSTM learns to predict the next flood event based on the input data and the synthetic data[123]. This approach has the potential to improve the accuracy and generalization of the LSTM models, as it can learn to predict floods based on a wider range of data. Researchers could try to explore the use of advanced deep learning algorithms, such as GANs, to improve the accuracy and robustness of flood forecasting.

5. CONCLUSIONS

The purpose of this manuscript is to provide a survey of the current state of machine learning applications in flood prediction. Due to the abundance of literature on the topic, the focus of this review is to analyze and compare the performance of mainstream algorithms that are currently being used in real-time flood prediction.

Accurately detecting the timing and magnitude of major floods is a challenge for watershed managers, as it is critical to provide timely early warnings to those at risk and save lives. Recent advancements in remote sensing technology and the installation of real-time flood water level detection sensors, in conjunction with advanced machine learning techniques, have made it possible to provide more accurate and longer forecast windows for predicting the timing and magnitude of future flooding events. This improved prediction capability enables better countermeasures, evacuation efforts, and mobilization of emergency management teams.

By utilizing advanced remote sensing techniques and real-time flood water level monitoring sensors, a large volume of data can be collected, quickly analyzed, and processed by machine learning algorithms to forecast floods. These algorithms can identify complex patterns in real-time data to accurately predict flood water levels in complex river networks and urban sewer sheds. The data collected can be used to create flood inundation maps and issue warnings to impacted residents through mobile applications on their cell phones, enabling emergency response teams to mobilize and coordinate evacuation efforts more effectively.

In conclusion, this review summarizes the combination of remote sensing, real-time monitoring, and machine learning technology, offering a promising solution to the challenge of accurately forecasting floods and reducing their impact on at-risk communities.

DECLARATIONS

Authors' contributions

Made substantial contributions to the research and investigation process, reviewed and summarized the literature, and wrote and edited the original draft: Yue Zhang
Conducted the research activity and execution, collaborated on writing the article: Zhang Y, Pan D
Oversight, leadership responsibility, commentary, and critical review: Yang SX, Gharabaghi B, Van Griensven J

Availability of data and materials

Not applicable.

Financial support and sponsorship

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Alliance Grant 401643. Grant co-funded by Lakes Environmental Research Inc.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Moftakhari HR, AghaKouchak A, Sanders BF, Allaire M, Matthew RA. What is nuisance flooding? Defining and monitoring an emerging challenge. *Water Resour Res* 2018;54:4218-27. DOI
2. Zhang M, Conti F, Le Sourne H, et al. A method for the direct assessment of ship collision damage and flooding risk in real conditions. *Ocean Eng* 2021;237:109605. DOI
3. Luiz-Silva W, Oscar-Júnior AC. Climate extremes related with rainfall in the State of Rio de Janeiro, Brazil: a review of climatological characteristics and recorded trends. *Nat Hazards* 2022;114:713-32. DOI
4. Bozorg O. Review on IPCC reports. *Climate Change in Sustainable Water Resources Management* 2022:123. DOI
5. Guo Y, Wu Y, Wen B, et al. Floods in China, COVID-19, and climate change. *The Lancet Planet Health* 2020;4:e443-44. DOI
6. Yamamoto H, Naka T. Quantitative analysis of the impact of floods on firms' financial conditions. Bank of Japan; 2021.
7. Romero M, Finke J, Rocha C. A top-down supervised learning approach to hierarchical multi-label classification in networks. *Appl Netw Sci* 2022;7:1-17. DOI
8. Henriksen HJ, Roberts MJ, van der Keur P, et al. Participatory early warning and monitoring systems: a nordic framework for web-based flood risk management. *Int J Disast Risk Re* 2018;31:1295-306. DOI
9. Ferrans P, Torres MN, Temprano J, Sánchez JPR. Sustainable Urban Drainage System (SUDS) modeling supporting decision-making: a systematic quantitative review. *Sci Total Environ* 2022;806:150447. DOI
10. Brunner MI, Slater L, Tallaksen LM, Clark M. Challenges in modeling and predicting floods and droughts: a review. *Wiley Interdiscip Rev: Water* 2021;8:e1520. DOI
11. Ding Y, Zhu Y, Wu Y, Jun F, Cheng Z. Spatio-temporal attention LSTM model for flood forecasting. In: 2019 International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE; 2019. pp. 458-65. DOI

12. De La Cruz R, Olfindo Jr N, Felicen M, et al. Near-realtime Flood Detection From Multi-temporal Sentinel Radar Images Using Artificial Intelligence. *ISPRS* 2020;43. DOI
13. Belabid N, Zhao F, Brocca L, Huang Y, Tan Y. Near-real-time flood forecasting based on satellite precipitation products. *Remote Sens* 2019;11:252. DOI
14. Munawar HS, Hammad AW, Waller ST. A review on flood management technologies related to image processing and machine learning. *Autom. Constr* 2021;132:103916. DOI
15. Bronfman NC, Cisternas PC, Repetto PB, Castañeda JV. Natural disaster preparedness in a multi-hazard environment: characterizing the sociodemographic profile of those better (worse) prepared. *PloS one* 2019;14:e0214249. DOI
16. Chhajer P, Shah M, Kshirsagar A. The applications of artificial neural networks, support vector machines, and long-short term memory for stock market prediction. *Decision Analytics Journal* 2022;2:100015. DOI
17. Zhang J, Bargal SA, Lin Z, et al. Top-down neural attention by excitation backprop. *Int J Comput Vision* 2018;126:1084-102. DOI
18. Greener JG, Kandathil SM, Moffat L, Jones DT. A guide to machine learning for biologists. *Nat Rev Mol Cell Bio* 2022;23:40-55. DOI
19. Rustam F, Reshi AA, Mehmood A, et al. COVID-19 future forecasting using supervised machine learning models. *IEEE access* 2020;8:101489-99. DOI
20. El Boujnouni M. A study and identification of COVID-19 viruses using N-grams with Naïve Bayes, K-nearest neighbors, artificial neural networks, decision tree and support vector machine. In: 2022 International Conference on Intelligent Systems and Computer Vision (ISCV). IEEE; 2022. pp. 1-7. DOI
21. Cunningham P, Cord M, Delany SJ. Supervised learning machine learning techniques for multimedia. Springer; 2008. DOI
22. Seel NM. Encyclopedia of the sciences of learning. Springer Science & Business Media; 2011. DOI
23. Zhou ZH. A brief introduction to weakly supervised learning. *Natl Sci Rev* 2018;5:44-53. DOI
24. de Bruijn JA, de Moel H, Jongman B, et al. A global database of historic and real-time flood events based on social media. *Sci data* 2019;6:311. DOI
25. Khan W, Ghazanfar MA, Azam MA, et al. Stock market prediction using machine learning classifiers and social media, news. *J Amb Intel Hum Comp* 2020:1-24. DOI
26. Van Engelen JE, Hoos HH. A survey on semi-supervised learning. *Mach Learn* 2020;109:373-440. DOI
27. Ghahramani Z. Unsupervised learning. Advanced Lectures on Machine Learning. LNAI 3176. Springer-Verlag; 2004. DOI
28. Zhou ZH, Zhou ZH. Semi-supervised learning. *Mach Learn* 2021:315-41. DOI
29. Sammut C, Webb GI. Encyclopedia of machine learning. Springer Science & Business Media; 2011. DOI
30. Jukes E. Encyclopedia of machine learning and data mining. *Reference Reviews* 2018;32:3-4. DOI
31. Zhu XJ. *Semi-supervised learning literature survey* 2005.
32. Mey A, Loog M. Improved generalization in semi-supervised learning: a survey of theoretical results. *IEEE T Pattern Anal* 2022. DOI
33. Xu W, Tang J, Xia H. A review of semi-supervised learning for industrial process regression modeling. In: 2021 40th Chinese Control Conference (CCC). IEEE; 2021. pp. 1359-64. DOI
34. Yang X, Song Z, King I, Xu Z. A survey on deep semi-supervised learning. *IEEE T Knowl Data En* 2022:1-20. DOI
35. Poldrack RA, Huckins G, Varoquaux G. Establishment of best practices for evidence for prediction: a review. *JAMA Psychiat* 2020;77:534-40. DOI
36. Gull T, Khurana S, Kumar M. Semi-supervised labeling: a proposed methodology for labeling the twitter datasets. *Multimed Tools Appl* 2022 03;81. DOI
37. Giglioni V, García-Macías E, Venanzi I, Ierimonti L, Ubertini F. The use of receiver operating characteristic curves and precision-versus-recall curves as performance metrics in unsupervised structural damage classification under changing environment. *Eng Struct* 2021;246:113029. DOI
38. Opella JMA, Hernandez AA. Developing a flood risk assessment using support vector machine and convolutional neural network: a conceptual framework. In: 2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA). IEEE; 2019. pp. 260-65. DOI
39. Sankaranarayanan S, Prabhakar M, Satish S, et al. Flood prediction based on weather parameters using deep learning. *J Water Clim Change* 2020;11:1766-83. DOI
40. Arzoumanian Z, Baker PT, Blumer H, et al. The NANOGrav 12.5 yr data set: search for an isotropic stochastic gravitational-wave background. *The Astrophysical journal letters* 2020;905:L34. DOI
41. Benetos E, Dixon S, Duan Z, Ewert S. Automatic music transcription: an overview. *IEEE Signal Proc Mag* 2018;36:20-30. DOI
42. Zhou T, Thung KH, Zhu X, Shen D. Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis. *Hum Brain Mapp* 2019;40:1001-16. DOI
43. Lin J, Li J, Chen J. An analysis of English classroom behavior by intelligent image recognition in IoT. *Int J Syst Assur Eng* 2021:1-9. DOI
44. Chen S, Yu J, Wang S. One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes. *J Process Contr* 2020;87:54-67. DOI
45. Masarczyk W, Głomb P, Grabowski B, Ostaszewski M. Effective training of deep convolutional neural networks for hyperspectral image classification through artificial labeling. *Remote Sens* 2020;12:2653. DOI
46. Lindsay GW. Convolutional neural networks as a model of the visual system: Past, present, and future. *J Cognitive Neurosci* 2021;33:2017-31. DOI
47. Zeman AA, Ritchie JB, Bracci S, Op de Beeck H. Orthogonal representations of object shape and category in deep convolutional neural networks and human visual cortex. *Sci Rep-Uk* 2020;10:2453. DOI

48. Alzubaidi L, Fadhel MA, Oleiwi SR, Al-Shamma O, Zhang J. DFU_QUTNet: diabetic foot ulcer classification using novel deep convolutional neural network. *Multimed Tools Appl* 2020;79:15655-77. DOI
49. Li Z, Wu Q, Xiao Y, Jin M, Lu H. Deep matching network for handwritten Chinese character recognition. *Pattern Recogn* 2020;107:107471. DOI
50. Devaraj J, Madurai Elavarasan R, Shafiullah G, Jamal T, Khan I. A holistic review on energy forecasting using big data and deep learning models. *Int J Energ Res* 2021;45:13489-530. DOI
51. Larochelle H, Erhan D, Courville A, Bergstra J, Bengio Y. An empirical evaluation of deep architectures on problems with many factors of variation. In: Proceedings of the 24th international conference on Machine learning; 2007. pp. 473-80. DOI
52. Imamverdiyev Y, Abdullayeva F. Deep learning method for denial of service attack detection based on restricted boltzmann machine. *Big data* 2018;6:159-69. DOI
53. Satarzadeh E, Sarraf A, Hajikandi H, Sadeghian MS. Flood hazard mapping in western Iran: assessment of deep learning vis-à-vis machine learning models. *Nat Hazards* 2022:1-19. DOI
54. Ying C, Li Q, Liu J. A Brief Investigation for Techniques of Deep Learning Model in Smart Grid. In: 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM). IEEE; 2021. pp. 173-81. DOI
55. Raza K, Singh NK. A tour of unsupervised deep learning for medical image analysis. *Curr Med Imaging* 2021;17:1059-77. DOI
56. Amin SU, Alsulaiman M, Muhammad G, Mekhtiche MA, Hossain MS. Deep learning for EEG motor imagery classification based on multi-layer CNNs feature fusion. *Future Gener Comp Sy* 2019;101:542-54. DOI
57. Zhang Y, Wu J, Cai Z, Du B, Philip SY. An unsupervised parameter learning model for RVFL neural network. *Neural Networks* 2019;112:85-97. DOI
58. Mittal S, Lamb A, Goyal A, et al. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In: International Conference on Machine Learning. PMLR; 2020. pp. 6972-86.
59. Forbus KD, Ferguson RW, Lovett A, Gentner D. Extending SME to handle large-scale cognitive modeling. *Cognitive Sci* 2017;41:1152-201. DOI
60. Mu Y. An evaluation of deep learning models for urban floods forecasting; 2022.
61. Sit M, Demiray BZ, Xiang Z, et al. A comprehensive review of deep learning applications in hydrology and water resources. *Water Sci Technol* 2020;82:2635-70. DOI
62. Nevo S, Morin E, Gerzi Rosenthal A, et al. Flood forecasting with machine learning models in an operational framework. *Hydrol Earth Syst Sc* 2022;26:4013-32. DOI
63. Saxena D, Cao J. Generative adversarial networks (GANs) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)* 2021;54:1-42. DOI
64. Goodfellow IJ. On distinguishability criteria for estimating generative models. *arXiv preprint arXiv:14126515* 2014. DOI
65. Salazar A, Vergara L, Safont G. Generative adversarial networks and markoveandom fields for oversampling very small training sets. *Expert Syst Appl* 2021;163:113819. DOI
66. Le XH, Ho HV, Lee G, Jung S. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water* 2019;11:1387. DOI
67. Ren Q, Li M, Song L, Liu H. An optimized combination prediction model for concrete dam deformation considering quantitative evaluation and hysteresis correction. *Adv Eng Inform* 2020;46:101154. DOI
68. Yang Z, Yu Y, You C, Steinhardt J, Ma Y. Rethinking bias-variance trade-off for generalization of neural networks. In: International Conference on Machine Learning. PMLR; 2020. pp. 10767-77.
69. Wang Q, Ma Y, Zhao K, Tian Y. A comprehensive survey of loss functions in machine learning. *Annals of Data Science* 2020:1-26. DOI
70. Anisha P, Polati A. A bird eye view on the usage of artificial intelligence. In: Communication Software and Networks: Proceedings of INDIA 2019. Springer; 2021. pp. 61–77. DOI
71. Belkin M, Hsu D, Ma S, Mandal S. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *P Natl Acad Sci* 2019;116:15849-54. DOI
72. Heinlein A, Klawonn A, Lanser M, Weber J. Combining machine learning and adaptive coarse spaces—a hybrid approach for robust FETI-DP methods in three dimensions. *SIAM J Sci Comput* 2021;43:S816-38. DOI
73. Jiang Y, Yin S, Dong J, Kaynak O. A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sens J* 2020;21:12868-81. DOI
74. Arnott R, Harvey CR, Markowitz H. A backtesting protocol in the era of machine learning. *The Journal of Financial Data Science* 2019;1:64-74. DOI
75. Sulaiman J, Wahab SH. Heavy rainfall forecasting model using artificial neural network for flood prone area. In: IT Convergence and Security 2017: Volume 1. Springer; 2018. pp. 68-76. DOI
76. Chen D, Liu F, Zhang Z, Lu X, Li Z. Significant wave height prediction based on wavelet graph neural network. In: 2021 IEEE 4th International Conference on Big Data and Artificial Intelligence (BDAI). IEEE; 2021. pp. 80-85. DOI
77. MacKinnon DP. Introduction to statistical mediation analysis. Routledge; 2012. DOI
78. Chen D, Zhou R, Pan Y, Liu F. A simple baseline for adversarial domain adaptation-based unsupervised flood forecasting. *arXiv preprint arXiv:220608105* 2022. DOI
79. Li J, Socher R, Hoi SC. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:200207394* 2020. DOI
80. Scheinost D, Noble S, Horien C, et al. Ten simple rules for predictive modeling of individual differences in neuroimaging. *NeuroImage* 2019;193:35-45. DOI

81. Lowrance CJ, Lauf AP. An active and incremental learning framework for the online prediction of link quality in robot networks. *Eng Appl Artif Intel* 2019;77:197-211. DOI
82. Cai S, Wang Z, Lu L, Zaki TA, Karniadakis GE. DeepM&Mnet: inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *J Comput Phys* 2021;436:110296. DOI
83. Guo Q, Zhuang F, Qin C, et al. A survey on knowledge graph-based recommender systems. *IEEE T Knowl Data En* 2020;34:3549–68. DOI
84. Kang Z, Pan H, Hoi SC, Xu Z. Robust graph learning from noisy data. *IEEE T Cybernetics* 2019;50:1833-43. DOI
85. Lyu L, Fang M, Wang N, Wu J. Water level prediction model based on GCN and LSTM. In: 2021 7th International Conference on Computer and Communications (ICCC). IEEE; 2021. pp. 1600-1605. DOI
86. Yang W, Chen L, Chen X, Chen H. Sub-daily precipitation-streamflow modelling of the karst-dominated basin using an improved grid-based distributed Xinanjiang hydrological model. *J Hydrol-Reg Stud* 2022;42:101125. DOI
87. Feng J, Wang Z, Wu Y, Xi Y. Spatial and temporal aware graph convolutional network for flood forecasting. In: 2021 International Joint Conference on Neural Networks (IJCNN). IEEE; 2021. pp. 1-8. DOI
88. Miau S, Hung WH. River flooding forecasting and anomaly detection based on deep learning. *IEEE Access* 2020;8:198384-402. DOI
89. Taşar B, Kaya YZ, Varçin H, Üneş F, Demirci M. Forecasting of suspended sediment in rivers using artificial neural networks approach. *International Journal of Advanced Engineering Research and Science* 2017;4:237333. DOI
90. Sahoo A, Samantaray S, Ghose DK. Prediction of flood in Barak River using hybrid machine learning approaches: a case study. *J Geol Soc India* 2021;97:186-98. DOI
91. Shilton A, Palaniswami M, Ralph D, Tsoi AC. Incremental training of support vector machines. *IEEE T Neural Networ* 2005;16:114-31. DOI
92. Zanfei A, Brentan BM, Menapace A, Righetti M, Herrera M. Graph convolutional recurrent neural networks for water demand forecasting. *Water Resour Res* 2022;58:e2022WR032299. DOI
93. Zhang Y, Gu Z, Thé JVG, Yang SX, Gharabaghi B. The discharge forecasting of multiple monitoring station for Humber River by hybrid LSTM models. *Water* 2022;14:1794. DOI
94. Cho M, Kim C, Jung K, Jung H. Water level prediction model applying a long short-term memory (lstm)–gated recurrent unit (gru) method for flood prediction. *Water* 2022;14:2221. DOI
95. Ding Y, Zhu Y, Feng J, Zhang P, Cheng Z. Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing* 2020;403:348-59. DOI
96. Dtissibe FY, Ari AAA, Titouna C, Thiare O, Gueroui AM. Flood forecasting based on an artificial neural network scheme. *Nat Hazards* 2020;104:1211-37. DOI
97. Ahmad M, Al Mehedi MA, Yazdan MMS, Kumar R. Development of machine learning flood model using Artificial Neural Network (ANN) at Var River. *Liquids* 2022;2:147-60. DOI
98. Hassanpour Kashani M, Montaseri M, Lotfollahi Yaghin MA. Flood estimation at ungauged sites using a new hybrid model. *J Appl Sci* 2008;8:1744-49. DOI
99. Tabbussum R, Dar AQ. Comparative analysis of neural network training algorithms for the flood forecast modelling of an alluvial Himalayan river. *J Flood Risk Manag* 2020;13:e12656. DOI
100. Jabbari A, Bae DH. Application of Artificial Neural Networks for accuracy enhancements of real-Time flood forecasting in the Imjin Basin. *Water* 2018;10. DOI
101. Dong P, Liao X, Chen Z, Chu H. An improved method for predicting CO_2 minimum miscibility pressure based on artificial neural network. *Advances in Geo-Energy Research* 2019;3:355-64. DOI
102. Kimura N, Yoshinaga I, Sekijima K, Azechi I, Baba D. Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. *Water* 2019;12:96. DOI
103. Zhang L, Huang Z, Liu W, Guo Z, Zhang Z. Weather radar echo prediction method based on convolution neural network and long short-term memory networks for sustainable e-agriculture. *J Clean Prod* 2021;298:126776. DOI
104. Sun W, Wang R. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with DSM. *IEEE Geosci Remote S* 2018;15:474-78. DOI
105. Yuan F, Xu Y, Li Q, Mostafavi A. Spatio-temporal graph convolutional networks for road network inundation status prediction during urban flooding. *Comput Environ Urban* 2022;97:101870. DOI
106. Mehedi MAA, Khosravi M, Yazdan MMS, Shabanian H. Exploring temporal dynamics of river discharge using univariate Long Short-Term Memory (LSTM) Recurrent Neural Network at east branch of Delaware River. *Hydrology* 2022;9:202. DOI
107. Liu M, Huang Y, Li Z, et al. The applicability of LSTM-KNN model for real-time flood forecasting in different climate zones in China. *Water* 2020;12:440. DOI
108. Song T, Ding W, Wu J, et al. Flash flood forecasting based on long short-term memory networks. *Water* 2019;12:109. DOI
109. Li X, Xu W, Ren M, Jiang Y, Fu G. Hybrid CNN-LSTM models for river flow prediction. *Water Supply* 2022;22:4902–19. DOI
110. Li P, Zhang J, Krebs P. Prediction of flow based on a CNN-LSTM combined deep learning approach. *Water* 2022;14:993. DOI
111. Kasiviswanathan KS, He J, Sudheer K, Tay JH. Potential application of wavelet neural network ensemble to forecast streamflow for flood management. *J Hydrol* 2016;536:161-73. DOI
112. Lin L, Li W, Bi H, Qin L. Vehicle trajectory prediction using LSTMs with spatial-temporal attention mechanisms. *IEEE Intel Transp Sy* 2021;14:197-208. DOI
113. Noor F, Haq S, Rakib M, et al. Water level forecasting using spatiotemporal attention-based Long Short-Term Memory Network. *Water*

2022;14:612. DOI

114. Wang Y, Huang Y, Xiao M, et al. Medium-long-term prediction of water level based on an improved spatio-temporal attention mechanism for long short-term memory networks. *J Hydrol* 2023;618:129163. DOI
115. Chen C, Luan D, Zhao S, et al. Flood discharge prediction based on remote-sensed spatiotemporal features fusion and graph attention. *Remote Sens* 2021;13:5023. DOI
116. Liu M, Chen L, Du X, Jin L, Shang M. Activated gradients for deep neural networks. *IEEE T Neur Net Lear* 2021. DOI
117. Luo Y, Huang Z, Zhang Z, Wang Z, Li J, et al. Curiosity-driven reinforcement learning for diverse visual paragraph generation. In: Proceedings of the 27th ACM International Conference on Multimedia; 2019. pp. 2341-50. DOI
118. Chang FJ, Hsu K, Chang LC. Flood forecasting using machine learning methods. MDPI; 2019. DOI
119. Tran DA, Tsujimura M, Ha NT, et al. Evaluating the predictive power of different machine learning algorithms for groundwater salinity prediction of multi-layer coastal aquifers in the Mekong Delta, Vietnam. *Ecol Indic* 2021;127:107790. DOI
120. Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. pp. 427-36. DOI
121. Hendrycks D, Gimpel K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:161002136* 2016. DOI
122. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM* 2021;64:107-15. DOI
123. Cheng M, Fang F, Navon I, Pain C. A real-time flow forecasting with deep convolutional generative adversarial network: Application to flooding event in Denmark. *Phys Fluids* 2021;33:056602. DOI

Li *et al*. *Intell Robot* 2023;3(2):213-21
DOI: 10.20517/ir.2023.13

Intelligence & Robotics

Research Article

Open Access

Check for updates

The cooperatability of the first-order multi-agent systems consisting of a leader and a follower with multiplicative noises under Markov switching topologies

Dianqiang Li, Tao Li

School of Mathematical Sciences, East China Normal University, Shanghai 200241, China.

Correspondence to: Dr. Dianqiang Li, School of Mathematical Sciences, East China Normal University, No.5005, South Lianhua Road, Shanghai 200241, China. E-mail: ldq2015ahdx@126.com.

How to cite this article: Li D, Li T. The cooperatability of the first-order multi-agent systems consisting of a leader and a follower with multiplicative noises under Markov switching topologies. *Intell Robot* 2023;3(2):213-21. http://dx.doi.org/10.20517/ir.2023.13

Received: 24 Mar 2023 **First Decision:** 17 Apr 2023 **Revised:** 29 May 2023 **Accepted:** 6 Jun 2023 **Published:** 28 Jun 2023

Academic Editor: Simon X. Yang, Hongtian Chen **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

We investigate the cooperatability of the first-order leader-following multi-agent systems consisting of a leader and a follower with multiplicative noises under Markov switching topologies. Each agent exhibits first-order linear dynamics, and there are multiplicative noises along with information exchange among the agents. What is more, the communication topologies are Markov switching topologies. By utilizing the stability theory of the stochastic differential equations with Markovian switching and the Markov chain theory, we establish the necessary and sufficient conditions for the cooperatability of the leader-following multi-agent systems. The conditions are outlined below: (i) The product of the system parameter and the square of multiplicative noise intensities should be less than 1/2; (ii) The transition rate from the unconnected graph to the connected graph should be twice the system parameter; (iii) The transition rate from the connected graph to the unconnected graph should be less than a constant that is related to the system parameter, the intensities of multiplicative noises, and the transition rate from the unconnected graph to the connected graph. Finally, the effectiveness of our control strategy is demonstrated by the population growth systems.

Keywords: Leader-following, Markov switching topologies, multiplicative noises, cooperatability

1. INTRODUCTION

In the past few decades, distributed cooperative control of multi-agent systems under fixed topologies has attracted much attention from the system and control community[1–3]. However, in practical systems, the communication networks connecting the agents often experience sudden interruptions and restoration. These mutations lead to the changes in the structures or parameters of the system. Here, we describe this changing topology by the Markovian switching topology. For such systems, we usually use the Markov switching systems to describe them. In recent years, the stability of linear Markov switching systems has been widely studied[4–8]. By Kronecker product and Lyapunov exponent, Mariton *et al.*[4] gave the necessary and sufficient conditions for the moment stability and the almost sure stability of the system, respectively. Feng *et al.*[5] studied the stochastic stability of the system and revealed the relationship between the moment stability and the almost sure stability. Feng *et al.*[6] studied the stabilization problem. The literature[7,8] investigated the robust stability problems and gave sufficient and necessary conditions in the form of linear matrix inequalities for the mean square stability.

In many real-world systems, it is inevitable for systems to be subjected to random noises[9]. These noises may change the trajectory of the system and even affect its stability. Therefore, an increasing number of researchers have focused on studying the stability of the Markov switching stochastic systems. The stability of linear Markov switching systems with stochastic noises was studied in previous literature[10–12]. Fragoso *et al.*[10] studied the Markov switching systems with additive noises and provided the necessary and sufficient conditions for the mean square stability of the system. On the other hand, the literature[11,12] explored the Markov switching systems with multiplicative noises. By employing the operator theory, Dragan *et al.*[11] derived the necessary and sufficient conditions in the form of linear matrix inequalities for the mean square stability. Similarly, Sheng *et al.*[12], also using the operator theory, presented a new necessary and sufficient condition for the mean square stability. Using the Lyapunov method, Mao *et al.*[13] established a sufficient condition for the pth moment exponential stability of the nonlinear Markov switching system and revealed the relationship between the pth moment exponential stability and the almost sure exponential stability of the system. In the context of nonlinear Markov switching systems, Deng *et al.*[14] addressed the problem of mean square stabilization.

The stability theory of Markov switching systems with noises has numerous practical applications[15–17]. Previous studies[18–22] have focused on the distributed control problem of multi-agent systems with random noises under Markov switching topologies. The literature[18,19] studied the distributed control problem of discrete-time multi-agent systems. By the state space decomposition method, Huang *et al.*[18] gave a sufficient condition for almost sure consensus and mean square consensus, respectively. Zhang *et al.*[19] studied the mean square consensus problem. The literature[20–22] considers the distributed control problem of continuous-time multi-agent systems. Zhang *et al.*[20] studies the distributed control problem of multi-agent systems with first-order integrator dynamics. Li *et al.*[21] studied the containment control problem. Wang *et al.*[22] studied mean square consensus and almost sure consensus of higher-order multi-agent systems.

Compared with additive noises, multiplicative noises play a stabilizing role in the almost sure stability of systems[23]. Many scholars have studied the distributed control problem of multi-agent systems with multiplicative noises[24–28]. However, as the state of the system is related to the Markov chain, we cannot write the expectation of the product of the state variable and the indicative function in the form of the expected product. This leads to the fact that the distributed control problem of multi-agent systems with multiplicative noises under the Markov switching topology has not yet been solved. As a preliminary study, we study the cooperatability of the first-order leader-following multi-agent systems consisting of a leader and a follower with multiplicative noises under Markov switching topologies. Each agent has first-order linear dynamics, and there are multiplicative noises along with information exchange among agents. What is more, the communication topologies are Markov switching topologies. Compared with existing literature[24–28], we have revealed the influence of multiplicative noises and switching rates on the cooperatability of the system. To analyze this influence, we delve into the stability theory of Markov switching systems with noises. Therefore, we introduced a new lemma to address this issue. We establish the necessary and sufficient conditions for the cooperatability of the leader-following multi-agent systems by combining the stability theory of the stochastic differential equation with Markovian switching and the Markov chain theory. These conditions are outlined below: (i) The product of the system parameter and the square of multiplicative noise intensities should be less than 1/2; (ii) The transition rate from the unconnected graph to the connected graph should be twice the value of the system parameter; (iii) The transition rate from the connected graph to the unconnected graph should be lower than a constant, which is related to the system parameter, the intensity of multiplicative noises, and the transition rate from the unconnected graph to the connected graph.

The remaining sections of this paper are structured as follows: Section 2 formulates the problem. Section 3 presents the admissible cooperative distributed control strategy. Section 4 provides the main result. Section 5 includes a numerical simulation to demonstrate the effectiveness of our control laws. Section 6 concludes the paper.

Notation: The symbols $\mathbb{R}$ and $\mathbb{R}_+$ denote real and non-negative numbers, respectively. I_n denotes the $n \times n$ dimensional identity matrix. The symbol diag$\{A_1, \ldots, A_N\}$ represents the block diagonal matrix with entries being $A_1, \ldots, A_N$. For a given vector or matrix X, X^{T} denotes its transpose, and $|X|$ represents the determinant of X. For two matrices C and D, $C \otimes D$ denotes their Kronecker product, and $C \oplus D = C \otimes I + I \otimes D$ represents the Kronecker sum. Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geqslant t_0}, \mathbb{P})$ be a complete probability space with a filtration $\{\mathcal{F}_t\}_{t \geqslant t_0}$ that satisfies the usual conditions, namely, it is right continuous and increasing while $\mathcal{F}_0$ contains all $\mathbb{P}$-null sets; $w(t) = (w_1(t), \ldots, w_m(t))^T$ denotes a m-dimensional standard Brownian motion defined in $\left(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geqslant t_0}, \mathbb{P}\right)$. For a given random variable X, the mathematical expectation of X is denoted by $\mathbb{E}[X]$.

2. PROBLEM FORMULATIONS

Consider a leader-following multi-agent system consisting of a leader and a follower, where the leader and the follower are indexed by 0 and 1, respectively. The dynamics of the leader is given by

$$\dot{x}_0(t) = a x_0(t), \tag{1}$$

where $x_0(t) \in \mathbb{R}$ is the state, and $a \in \mathbb{R}^+$ is a known constant.

The dynamics of the follower is given by

$$\dot{x}_1(t) = a x_1(t) + b u(t), \tag{2}$$

where $x_1(t) \in \mathbb{R}$ is the state, $u(t) \in \mathbb{R}$ is the input, and $a \in \mathbb{R}^+$ and $b \in \mathbb{R}/0$ are known constants.

In this section, we assume that the topology graph is a Markovian switching topology. Let the switching signal $\theta(t)$ be defined in the probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geqslant 0}, \mathbb{P})$. The signal $\theta(t)$ is a right continuous homogeneous Markov chain and has a finite state space $\mathbb{S} = \{1, 2\}$. The matrix $Q = [q_{ij}]_{1 \leqslant i,j \leqslant 2}$ is the transfer rate matrix of the Markov chain $\theta(t)$ and satisfies

$$P(\theta(t+\triangle) = j | \theta(t) = i) = \begin{cases} q_{ij} \triangle + o(\triangle), & i \neq j, \\ 1 + q_{ij} \triangle + o(\triangle), & i = j, \end{cases}$$

where if $i \neq j$, q_{ij} is the transition rate of the Markov chain from the state i to the state j with $q_{ij} \geqslant 0$; if $i = j$, $q_{ii} = -\sum\limits_{j \neq i}^{2} q_{ij}$; $\triangle > 0$ and $\lim\limits_{t \to \infty} \frac{o(\triangle)}{\triangle} = 0$. We use $\mathcal{G}_{(\theta(t))} = (\mathcal{V}, \mathcal{E}(\theta(t)), \mathcal{A}(\theta(t)))$ to represent a weighted graph formed by the leader and the follower, where the set of nodes $\mathcal{V} = \{0, 1\}$ and the set of edges $\mathcal{E}(\theta(t)) \subseteq \mathcal{V}\times\mathcal{V}$. Denote the neighbors of the ith agent by $\mathcal{N}_i(\theta(t))$. The adjacency matrix $\mathcal{A}(\theta(t)) = \begin{bmatrix} 0 & 0 \\ a_{10}(\theta(t)) & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, where if $0 \in \mathcal{N}_1(\theta(t))$, then $a_{10}(\theta(t)) = 1$, otherwise $a_{10}(\theta(t)) = 0$. The Laplacian matrix of $\mathcal{G}_{(\theta(t))}$ is given by $\mathcal{L}(\theta(t)) = \mathcal{D}(\theta(t)) - \mathcal{A}(\theta(t))$, where $\mathcal{D}(\theta(t)) = \mathrm{diag}(0, a_{10}(\theta(t)))$. Without losing generality, we assume that the transition rate matrix of the Markov chain $\theta(t)$ is the matrix $Q = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$, where α represents the transition rate from the unconnected graph to the connected graph; β represents the transition rate from the connected graph to the unconnected graph.

3. ADMISSIBLE DISTRIBUTED COOPERATIVE CONTROL STRATEGY

In the real network, the relative state measurement information obtained by the follower from the leader is often affected by noises. Therefore, for the leader−following multi-agent system (1)−(2), we assume that the relative state measurement information has the following form

$$y_{10}(t) = x_1(t) - x_0(t) + \sigma_{10}\left(x_1(t) - x_0(t)\right)\xi_{10}(t), \tag{3}$$

where $\xi_{10}(t)$ represents the multiplicative measurement noise, and σ_{10} represents the intensity of multiplicative measurement noise.

We consider the following set of admissible distributed cooperative control strategies based on (3) and the randomness of the communication topology

$$\mathcal{U} = \left\{U = \{u(t) = ka_{10}(\theta(t))y_{10}(t), t > 0\}, k \in \mathbb{R}\right\}. \tag{4}$$

This paper primarily focuses on investigating the necessary and sufficient conditions for the cooperatability of the first-order leader-following multi-agent systems. These systems are composed of a leader and a follower and are subjected to multiplicative noises under Markov switching topologies.

The assumption and lemma required in this section are given below.

Assumption 1 The noise process $\xi_{10}(t)$ satisfies $\int_0^t \xi_{10}(s)\mathrm{d}s = w_{10}(t), t \geqslant 0$, where $w_{10}(t)$ is a one-dimensional standard Brownian motion.

Lemma 1 [12] The solution of the Markov switching stochastic differential equations

$$\mathrm{d}x(t) = A(\theta(t))x(t)\mathrm{d}t + C(\theta(t))x(t)\mathrm{d}w(t) \tag{5}$$

is mean square stable if and only if $F = \mathrm{diag}(A(1) \oplus A(1), \dots, A(S) \oplus A(S)) + \mathrm{diag}(C(1) \oplus C(1), \dots, C(S) \oplus C(S)) + Q^{\mathrm{T}} \otimes I_{n^2}$ is a Hurwitz matrix, where $Q = [q_{ij}]_{1\leqslant i,j\leqslant S}$ is the transition rate matrix of the Markov chain $\theta(t)$. If $\theta(t) = i$, we denote $A(\theta(t)) = A(i)$, $C(\theta(t)) = C(i)$, and $i = 1, \dots, S$.

4. MAIN RESULTS

By leveraging the stability theory of stochastic differential equations with Markovian switching and the Markov chain theory, we provide the necessary and sufficient conditions for the cooperatability of the leader-following multi-agent systems.

Theorem 1 Suppose Assumption 1 is satisfied. In that case, there exists an admissible cooperative control strategy denoted by $U \in \mathcal{U}$, which ensures that the follower can track the leader for any initial value under the distributed control law U. This holds if and only if the following conditions are met: $2a\sigma_{10}^2 < 1$, $\alpha > 2a$, $0 \leqslant \beta < \frac{(\alpha-2a)(1-2a\sigma_{10}^2)}{2a\sigma_{10}^2}$.

Proof: Denote $\delta(t) = x_1(t) - x_0(t)$. By Assumptions 1 and (1)–(4), we get

$$\mathrm{d}\delta(t) = [a + bka_{10}(\theta(t))]\delta_1(t)\mathrm{d}t + bka_{10}(\theta(t))\sigma_{10}\delta_1(t)\mathrm{d}w_{10}(t), \tag{6}$$

where if $a_{10}(\theta(t)) = 0$, then we denote $A(1) = a$ and $C(1) = 0$; if $a_{10}(\theta(t)) = 1$, then we denote $A(2) = a + bk$ and $C(2) = bk\sigma_{10}$.

Denote $F = \mathrm{diag}(A(1) \oplus A(1), A(2) \oplus A(2)) + \mathrm{diag}(C(1) \oplus C(1), C(2) \oplus C(2)) + Q^{\mathrm{T}} \otimes I_1$. By the definition of F, we have

$$\begin{aligned} F &= \begin{bmatrix} 2a & 0 \\ 0 & 2(a+bk) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b^2k^2\sigma_{10}^2 \end{bmatrix} + \begin{bmatrix} -\alpha & \beta \\ \alpha & -\beta \end{bmatrix} \\ &= \begin{bmatrix} 2a-\alpha & \beta \\ \alpha & 2(a+bk) + b^2k^2\sigma_{10}^2 - \beta \end{bmatrix}. \end{aligned} \tag{7}$$

Necessity: If there exists an admissible cooperative control strategy denoted by $U \in \mathcal{U}$, such that for any initial value, the follower can track the leader under the distributed control law U, it implies that the system (6) is mean square stable. According to Lemma 1, it can be inferred that all eigenvalues of F have negative real parts.

Noting that $|\lambda I - F| = \begin{vmatrix} \lambda - 2a + \alpha & -\beta \\ -\alpha & \lambda - 2(a+bk) - b^2k^2\sigma_{10}^2 + \beta \end{vmatrix}$, we have $|\lambda I - F| = (\lambda - 2a)^2 + (\beta + \alpha - 2bk - b^2k^2\sigma_{10}^2)(\lambda - 2a) - 2bk\alpha - \alpha b^2k^2\sigma_{10}^2$.

Denote $m = \lambda - 2a$ and $f(m) = m^2 + (\beta + \alpha - 2bk - b^2k^2\sigma_{10}^2)m - 2bk\alpha - \alpha b^2k^2\sigma_{10}^2$. As all eigenvalues of F have negative real parts, we know that the real parts of the zero point of $f(m)$ are less than $-2a$. As the real parts of the zero point of $f(m)$ are less than $-2a$, by considering the image of the function $f(m)$, the following two conditions can be inferred.

Condition $(\mathbf{C_1})$: $f(-2a) = 4a^2 - 2a(\beta + \alpha - 2bk - b^2k^2\sigma_{10}^2) - 2bk\alpha - \alpha b^2k^2\sigma_{10}^2 > 0$.

Condition $(\mathbf{C_2})$: $\frac{2bk-\alpha-\beta+b^2k^2\sigma_{10}^2}{2} + 2a < 0$.
By Condition $(\mathbf{C_1})$, we obtain

$$(2a - \alpha)(2a + 2bk + b^2k^2\sigma_{10}^2) > 2a\beta. \tag{8}$$

In the following, we discuss the Conditions $(\mathbf{C_1})$ and $(\mathbf{C_2})$.

(1) If $2a + 2bk + b^2k^2\sigma_{10}^2 = 0$ holds, then we have $(2a - \alpha)(2a + 2bk + b^2k^2\sigma_{10}^2) = 0$. This contradicts the inequality (8). Therefore, this situation does not hold.

(2) If $2a + 2bk + b^2k^2\sigma_{10}^2 > 0$ holds, by (8), we get $0 \leq \alpha < 2a$.
By $\frac{2bk-\alpha-\beta+b^2k^2\sigma_{10}^2}{2} + 2a < 0$, we have $\beta + \alpha - 2bk - b^2k^2\sigma_{10}^2 > 4a$. By $\beta + \alpha - 2bk - b^2k^2\sigma_{10}^2 > 4a$, $2a + 2bk + b^2k^2\sigma_{10}^2 > 0$ and $0 \leq \alpha < 2a$, we obtain $f(-2a) = 4a^2 - 2a(\beta + \alpha - 2bk - b^2k^2\sigma_{10}^2) - 2bk\alpha - \alpha b^2k^2\sigma_{10}^2 < -4a^2 - 2bk\alpha - \alpha b^2k^2\sigma_{10}^2 = -4a^2 - \alpha(2bk + b^2k^2\sigma_{10}^2) < -4a^2 + 2\alpha a = 2a(\alpha - 2a) < 0$. This contradicts Condition $(\mathbf{C_1})$. Therefore, this situation also is not valid.

(3) If $2a + 2bk + b^2k^2\sigma_{10}^2 < 0$, then by $\beta \geqslant 0$, $a > 0$ and (8), we have $\alpha > 2a$.

By $\alpha > 2a$ and (8), we get

$$b^2k^2\sigma_{10}^2 + 2bk < \frac{2a\beta}{2a-\alpha} - 2a. \tag{9}$$

By Condition ($\mathbf{C_2}$), we obtain

$$b^2k^2\sigma_{10}^2 + 2bk < \alpha + \beta - 4a. \tag{10}$$

By $\beta \geqslant 0$, $a > 0$ and $\alpha > 2a$, we get

$$\alpha + \beta - 4a - (\frac{2a\beta}{2a-\alpha} - 2a) = \alpha - 2a + \beta + \frac{2a\beta}{\alpha - 2a} > 0, \tag{11}$$

which implies $\alpha + \beta - 4a > \frac{2a\beta}{2a-\alpha} - 2a$.

By (9), (10), and (11), we have

$$b^2k^2\sigma_{10}^2 + 2bk < \frac{2a\beta}{2a-\alpha} - 2a. \tag{12}$$

Denote $g(t) = \sigma_{10}^2 t^2 + 2t - \frac{2a\beta}{2a-\alpha} + 2a$ and $t = bk$. By (12), we know that $g(t) < 0$ has a solution for variable t. By $g(t) < 0$, we have $\Delta = 4 - 4\sigma_{10}^2(-\frac{2a\beta}{2a-\alpha} + 2a) > 0$. By $\Delta > 0$, we get $\beta < \frac{(\alpha-2a)(1-2a\sigma_{10}^2)}{2a\sigma_{10}^2}$. Combining $0 \leqslant \beta < \frac{(\alpha-2a)(1-2a\sigma_{10}^2)}{2a\sigma_{10}^2}$ and $\alpha > 2a$, we have $2a\sigma_{10}^2 < 1$. In summary, we obtain $2a\sigma_{10}^2 < 1$, $\alpha > 2a$, $0 \leqslant \beta < \frac{(\alpha-2a)(1-2a\sigma_{10}^2)}{2a\sigma_{10}^2}$.

Sufficiency: By $2a\sigma_{10}^2 < 1$, $\alpha > 2a$ and $0 \leqslant \beta < \frac{(\alpha-2a)(1-2a\sigma_{10}^2)}{2a\sigma_{10}^2}$, we get (12). By (12), we have $bk \in \left(\frac{-2(\alpha-2a)-\sqrt{\rho}}{2(\alpha-2a)\sigma_{10}^2}, \frac{-2(\alpha-2a)+\sqrt{\rho}}{2(\alpha-2a)\sigma_{10}^2}\right)$, where $\rho = 4(\alpha-2a)^2 - 4(\alpha-2a)\sigma_{10}^2[2a(\alpha-2a) + 2a\beta]$. From the value range of bk, it can be seen that Condition ($\mathbf{C_1}$) and Condition ($\mathbf{C_2}$) hold. Therefore, since the real parts of the zero point of $f(m)$ are less than $-2a$, it can be concluded that all eigenvalues of F have negative real parts. Lemma 1 implies that the system (6) is mean square stable. Therefore, there exists an admissible cooperative control strategy $U \in \mathcal{U}$, such that for any initial value, the follower can track the leader under the distributed control law U.

Remark 1 The conditions $\alpha > 2a$, $2a\sigma_{10}^2 < 1$ and $0 \leqslant \beta < \frac{(\alpha-2a)(1-2a\sigma_{10}^2)}{2a\sigma_{10}^2}$ stated in Theorem 1 highlight the influence of multiplicative noises and both the transition rates α and β on the cooperatability of the system. It is shown that smaller multiplicative noises, lower transition rate β, and higher transition rate α are all favorable for the cooperatability of the system. Moreover, the transition rates α and β have lower and upper bounds, respectively. What is more, the noises and the system parameters satisfy the corresponding inequality.

We have the following corollary for the case without measurement noises.

Corollary 1 Suppose Assumption 1 and $\sigma_{10} = 0$ hold. In that case, there exists an admissible cooperative control strategy denoted by $U \in \mathcal{U}$, such that for any initial value, the follower can track the leader under the distributed control law U, if and only if $\alpha > 2a$.

5. NUMERICAL SIMULATION

In this section, we will use a numerical example to demonstrate the effectiveness of our control laws.

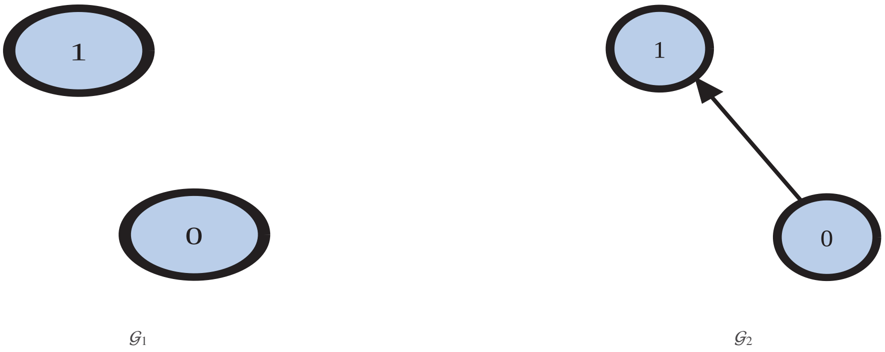


Figure 1. The communication topology graphs.

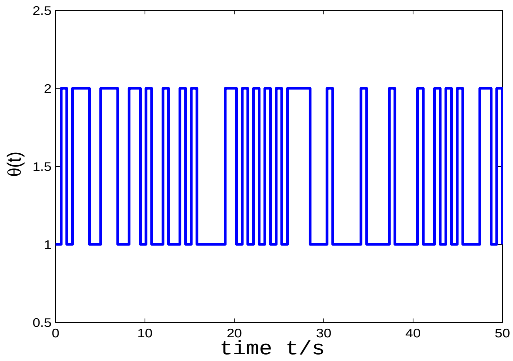


Figure 2. Markov chain $\theta(t)$.

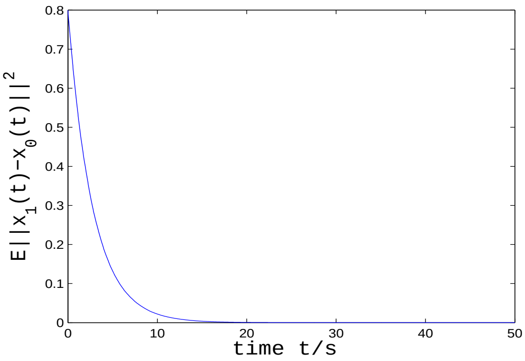


Figure 3. Mean square tracking errors.

Referring to the literature[29], the population growth system is given by (1). Consider the leader-following population growth systems (1)-(2), where $a = 0.01$ and $b = 0.2$, we will verify that the population of the follower can track the population of the leader under the distributed control law U.

The communication topology graphs are shown in Figure 1, and the trajectory of the Markov chain $\theta(t)$ is shown in Figure 2. The intensity of multiplicative measurement noise in (3) is given by $\sigma_{10} = 0.4$. The transition rate matrix is given by $Q = \begin{bmatrix} -2 & 2 \\ 1 & -1 \end{bmatrix}$. The initial states of agents are given by $x_0(0) = 0.2$ and $x_1(0) = 0.4$.

If we choose $k = -2$, then under the control law U, the mean square error of the population between the follower and the leader is shown in Figure 3. From Figure 3, we can see that the mean square error of the population tends to zero, which implies that the follower can achieve mean square tracking under the control law of U.

6. CONCLUSION

In this paper, we have studied the cooperatability of the first-order leader-following multi-agent systems that consist of a leader and a follower. The systems are subjected to multiplicative noises under Markov switching topologies. Each agent in this system follows first-order linear dynamics, and there are multiplicative noises along with information exchange among agents. Additionally, the communication topologies are characterized by Markov switching. By employing the stability theory of the stochastic differential equation with Markovian switching and the Markov chain theory, we have established the necessary and sufficient conditions for achieving the cooperatability in the leader-following multi-agent systems. Furthermore, there are several other interesting topics that can be explored in future research. For instance, it would be valuable to investigate the cooperatability of the leader-following multi-agent systems with both multiplicative noises and delays under Markov switching topologies

DECLARATIONS

Authors' contributions

Made substantial contributions to the research and investigation process, reviewed and summarized the literature, and wrote and edited the original draft: Li D
Performed oversight and leadership responsibility for the research activity planning and execution and performed critical review, commentary, and revision: Li T

Availability of data and materials

Not applicable.

Financial support and sponsorship

Not applicable.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Mao J, Huang S, Xiang Z, Wang Y, Zheng D. Practical finite-time sampled-data output consensus for a class of nonlinear multiagent systems via output feedback. *Int J Robust Nonlinear Control* 2021;31:920-49. DOI
2. Mao J, Yan T, Huang S, Li S, Jiao J. Sampled-data output feedback leader-following consensus for a class of nonlinear multi-agent systems with input unmodeled dynamics. *Int J Robust Nonlinear Control* 2021;31:4203-26. DOI
3. Shang Y, Ye Y. Leader-follower fixed-time group consensus control of multiagent systems under directed topology. *Complexity* 2017;2017:1-9. DOI
4. Mariton M. Almost sure and moments stability of jump linear systems. *Syst Control Lett* 1988;11:393-7. DOI
5. Feng X, Loparo K, Ji Y, Chizeck H. Stochastic stability properties of jump linear systems. *IEEE Trans Automat Contr* 1992;37:38-53. DOI
6. Fang Y, Loparo K. Stabilization of continuous-time jump linear systems. *IEEE Trans Automat Contr* 2002;47:1590-603. DOI
7. El Ghaoui L, Rami MA. Robust state feedback stabilization of jump linear systems via LMIs. *Int J Robust Nonlin Contr* 1996;6:1015-22. DOI
8. Costa OLV, Boukas EK. Necessary and sufficient condition for robust stability and stabilizability of continuous-time linear systems with markovian jumps. *J Optim Theory Appl* 1998;99:359-79. DOI
9. Somarakis C, Motee N. Aggregate fluctuations in networks with drift-diffusion models driven by stable non-gaussian disturbances. *IEEE Trans Control Netw Syst* 2020;7:1248-58. DOI
10. Fragoso MD, Costa OL. A unified approach for mean square stability of continuous-time markovian jumping linear systems with additive disturbances. In: Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, 12−15 December, 2000, pp. 2361−2366. DOI
11. Dragan V, Morozan T, Stoica AM. Mathematical methods in robust control of linear stochastic systems. 2006, New York, USA: Springer.

12. Sheng L, Gao M, Zhang W. Spectral characterisation for stability and stabilisation of linear stochastic systems with Markovian switching and its applications. *IET Control* 2013;7:730-7. DOI
13. Mao X. Stability of stochastic differential equations with Markovian switching. *Stoch* 1999;79:45-67. DOI
14. Deng F, Luo Q, Mao X. Stochastic stabilization of hybrid differential equations. *Automatica* 2012;48:2321-8. DOI
15. Shang Y. Consensus seeking over Markovian switching networks with time-varying delays and uncertain topologies. *Appl Math Comput* 2016;273:1234-45. DOI
16. Shang Y. Couple-group consensus of continuous-time multi-agent systems under Markovian switching topologies. *J Franklin Inst* 2015;352:4826-44. DOI
17. Shang Y. Consensus of noisy multiagent systems with markovian switching topologies and time-varying delays. *Math Probl Eng* 2015;2015:1-13. DOI
18. Huang M, Dey S, Nair GN, Manton JH. Stochastic consensus over noisy networks with Markovian and arbitrary switches. *Automatica* 2010;46:1571-83. DOI
19. Zhang Y, Tian Y. Consentability and protocol design of multi-agent systems with stochastic switching topology.*Automatica* 2009;45:1195-201. DOI
20. Zhang Q, Zhang JF. Zhang Q, Zhang JF. Distributed consensus of continuous-time multi-agent systems with Markovian switching topologies and stochastic communication noises. *Int J Math Syst Sci* 2011;31:1097-110. DOI
21. Li W, Xie L, Zhang J. Containment control of leader-following multi-agent systems with Markovian switching network topologies and measurement noises. *Automatica* 2015;51:263-7. DOI
22. Wang Y, Cheng L, Ren W, Hou Z, Tan M. Seeking consensus in networks of linear agents: communication noises and markovian switching topologies. *IEEE Trans Automat Contr* 2015;60:1374-9. DOI
23. Li T, Wu F, Zhang J. Multi-agent consensus with relative-state-dependent measurement noises. *IEEE Trans Automat Contr* 2014;59:2463-8. DOI
24. Long Y, Liu S, Xie L. Distributed consensus of discrete-time multi-agent systems with multiplicative noises: consensus of discrete-time multi-agent systems. *Int J Robust Nonlin Contr* 2015;25:3113-31. DOI
25. Zong X, Li T, Zhang J. Consensus control of discrete-time multi-agent systems with time-delays and multiplicative measurement noises. *Scientia Sinica Mathematica* 2016;46:1617–36. DOI
26. Ni Y, Li X. Consensus seeking in multi-agent systems with multiplicative measurement noises. *Syst Control Lett* 2013;62:430-7. DOI
27. Djaidja S, Wu Q. Leader-following consensus for single-integrator multi-agent systems with multiplicative noises in directed topologies. *Int J Syst Sci* 2015;46:2788-98. DOI
28. Zong X, Li T, Zhang J. Consensus conditions of continuous-time multi-agent systems with time-delays and measurement noises. *Automatica* 2019;99:412-9. DOI
29. Malthus TR. An Essay on the Principle of Population. Fifth edition, Volume III, 1817, London: John Murray.

Short *et al*. *Intell Robot* 2023;3(2):222-41
DOI: 10.20517/ir.2023.14

Intelligence & Robotics

Research Article

Open Access

A bio-inspired algorithm in image-based path planning and localization using visual features and maps

Daniel Short[1,#], Tingjun Lei[1,#], Chaomin Luo[1], Daniel W. Carruth[2], Zhuming Bi[3]

[1]Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762, USA.
[2]Center for Advanced Vehicular Systems, Mississippi State University, Mississippi State, MS 39762, USA.
[3]Department of Civil and Mechanical Engineering, Purdue University Fort Wayne, Fort Wayne, IN 46805, USA.
[#]The authors contributed equally.

Correspondence to: Prof. Chaomin Luo, Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762, USA. E-mail: Chaomin.Luo@ece.msstate.edu

How to cite this article: Short D, Lei T, Luo C, Carruth DW, Bi Z. A bio-inspired algorithm in image-based path planning and local-ization using visual features and maps. *Intell Robot* 2023;3(2):222-41. http://dx.doi.org/10.20517/ir.2023.14

Received: 19 Feb 2023 **First Decision:** 13 Apr 2023 **Revised:** 26 May 2023 **Accepted:** 5 Jun 2023 **Published:** 28 Jun 2023

Academic Editors: Simon X. Yang, Jiankun Wang **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

With the growing applications of autonomous robots and vehicles in unknown environments, studies on image-based localization and navigation have attracted a great deal of attention. This study is significantly motivated by the observation that relatively little research has been published on the integration of cutting-edge path planning algorithms for robust, reliable, and effective image-based navigation. To address this gap, a *biologically inspired* Bat Algorithm (BA) is introduced and adopted for image-based path planning in this paper. The proposed algorithm utilizes visual features as the reference in generating a path for an autonomous vehicle, and these features are extracted from the obtained images by convolutional neural networks (CNNs). The paper proceeds as follows: first, the requirements for image-based localization and navigation are described. Second, the principles of the BA are explained in order to expound on the justifications for its successful incorporation in image-based navigation. Third, in the proposed image-based navigation system, the BA is developed and implemented as a path planning tool for global path planning. Finally, the performance of the BA is analyzed and verified through simulation and comparison studies to demonstrate its effectiveness.

Keywords: Image-based navigation, Bat Algorithm, path planning, localization, autonomous vehicles, Mapping

1. INTRODUCTION

Autonomous vehicle path planning plays a crucial role in Simultaneous Localization and Mapping algorithms (SLAM) and navigation modules. Its purpose is to construct a safe and collision-free trajectory, including map-based path planning and image-based path planning[1–9]. Map-based navigation utilizes sensor readings performed in Cartesian space with terrain costs in maps[10–15]. Image-space of an on-board camera is used to feed raw images into a cost-image to form image-based path planning[16–18]. The constructed maps have various fundamental representations, such as metric maps, geometric maps, feature maps, and hybrid models. These representations essentially are an integration of multiple methods. As the outdoor landscape tends to be myopic, the limited range of LIDAR sensors poses challenges in building up maps with long-distance perception. Consequently, this limitation results in inefficient path planning in sensor-based navigation systems[19–22]. In map-based path planning, obstacles can often present aliasing issues when converting from sensor readings to the Cartesian space. For instance, in sensor-based navigation, solid areas may have gaps or slender corridors may disappear completely. However, when we consider an image feature-based navigation, researchers can represent such an image-based map by defining a graph, with the set of vertices representing images and the set of edges representing paired relationships between these images. Maps may be constructed in light of the sequence of images acquired to find paths from the initial to the destination pose. Image-based localization, with feature representations for image retrieval and image processing, contributes to mapping and navigation.

1.1. Related work

Navigation in an unknown and unstructured environment is extremely challenging and is involved in multiple, well-known, and well-studied disciplines[3,23–26]. Within the research community, there is a body of knowledge relating to neural network (NN) architectures that apply machine learning concepts to extract features and image descriptors from scenes, thereby facilitating various tasks related to autonomous vehicles[27,28]. This is typically accomplished through the implementation of a pooling layer inspired by the Vector of Locally Aggregated Descriptors (VLAD), which pools extracted descriptors into a fixed image representation, allowing its NN parameters to be learned by back-propagation NN. VLAD descriptors accomplish visual location recognition by directly using the intermediate layer outputs of pre-trained CNNs[29].

Visual place recognition methods can be based on directed acyclic graph matching, where feature maps extracted from deep CNNs are used to form probability distributions on image representation[30]. Another approach is based on extreme K-means and effective extreme learning machines (EELMs), where image decomposition with curvelet transformation is used to reduce dimensionality and generates a set of distinctive features[31] for use in future processing phases. As an example, Muthusamy *et al.*[32] applied a Gaussian mixture model (GMM) in various pattern recognition applications with EELM with excellent results.

The developments in feature extraction through NNs and CNNs naturally lend themselves to the performance of tasks associated with autonomous vehicles, such as SLAM[33]. One particular type of SLAM, Learning SLAM, can obtain camera pose and create a 3D map. However, it needs to have the prior dataset to train the network, and its performance often depends on that dataset. As a result, learning SLAM has poor generalization and is less versatile than some of its other implementations, such as geometric SLAM. This is generally owing to its 3D map being less accurate than one created via geometric SLAM. Visual SLAM, on the other hand, is a more dynamic system that typically consists of three components: (1) a visual odometry algorithm that provides an initial state estimate; (2) a place recognition system that is able to relate the currently observed scene to previously observed scenes; and (3) an optimization back-end which consistently integrates the place matches from the place recognition system with the full stream of state estimates[34]. While visual SLAM has proven to be highly valuable in ideal environments, there are certain disadvantages to this methodology that researchers are actively addressing in current research. The fundamental issue lies in the challenges posed by environmental changes, which induce errors and make feature identification difficult. These challenges can occur due to changes in time, environment, camera posture, and other factors[35].

Various forms of deep NNs (DNNs) and artificial intelligence (AI) techniques are utilized to make a system more robust by autonomously tuning parameters, thereby enhancing overall performance. Learned features, such as those derived from massive amounts of deep learning (DL) meta-sensor data, are aggregated into coherent large-scale maps and then classified features extracted from those DNNs, such as "lake", "road", "field", "obstacles", or "traversable terrain", assigned with corresponding cost values in a Cartesian space[36]. Graphs are then constructed based on these maps to enable path planning. Training data can be constructed either directly from image-space features or by projecting pixel data into Cartesian space, depending on the methods employed. To create an optimal trajectory, DL-based methods are used to construct a path that takes into account the geometry of the explored areas, taking into account the starting and ending positions. Outputs of image processing techniques, such as edge detection and region segmentation, are used to determine the explorable regions of the map[17]. Path planning inputs are derived from standard commercially available NNs, which are used to generate topological maps. Image-sensor input data are represented as a bipartite cost-graph, with disjoint nodes representing image-feature distances and their corresponding geographic distances[37].

In a parallel research area, biologically inspired algorithms have become popular for performing optimizations and solving problems involving n-dimensional search spaces. Evolutionary Algorithms (EAs) are widely used as global searching methods for optimization, and hybrids of EAs and analytical methods are providing a promising atmosphere for NN and CNN training applications[38]. Ongoing research in this domain provides fuel for the integration of subsets of EA-like swarm methodologies for parameter estimation and optimization. The swarm intelligence methodology aims to optimize the conditioning of EELM, and some researchers have already proposed an effective particle swarm optimization (PSO) algorithm known as the Multitask Beetle Antennae Swarm (MBAS) Algorithm. This algorithm was inspired by the structures of the artificial bee colony (ABS) algorithm and the Beetle Antennae Search (BAS) algorithm[39]. In fact, some research efforts even claim that the approach is so straightforward and effective that it may be possible to completely replace traditional NN training paradigms, including back-propagation[40].

EELM is not the only area benefiting from swarm methodologies such as PSO. The controller of autonomous vehicles has seen significant improvements as well, particularly in the case of four-wheel steer four-wheel drive (4WS4WD) vehicles. The improved PSO-based controller takes into account all the slip forces acting on the vehicle, leading to a notable enhancement in its robustness[41–43]. Population-based algorithms have been used in a more generalized way to solve optimization problems that are typically challenging for traditional deterministic algorithms, such as the issue of the local optima. One novel population based on an optimization strategy involves leveraging the different behaviors of mosquitoes during the foraging process[44]. Another biomimetic swarm intelligence optimization algorithm, pigeon-inspired optimization (PIO), has also been introduced as a novel approach in this field. PIO simulates the homing behavior of pigeons using magnetic fields and landmarks. PIO has also been further improved to enhance its effectiveness in solving many-objective optimization problems (MaOPs)[45]. The list of EA research is extensive. Some of the optimization algorithms currently published include bacterial colony chemotaxis (BCC) optimization[46], the "swarm of bees" or BSO algorithm[47], and the discrete artificial bee colony (DABC) algorithm[48].

One important bio-inspired optimization algorithm is the Bat Algorithm (BA). The BA is the focus of our current research and has been the subject of many research papers. Various variations to the BA have been proposed, demonstrating its versatility and effectiveness in solving a wide range of optimization problems across multiple fields. This study showed that the proposed BA algorithm clearly outperformed competing methods, such as the Pareto concept, in terms of performance and solution quality. Furthermore, due to the similarities between the BA and the PSO, BA has also been applied to solve the Wireless Sensor Network (WSN) coverage problem. In the context of wireless sensor node development research, the sensor node itself is regarded as a

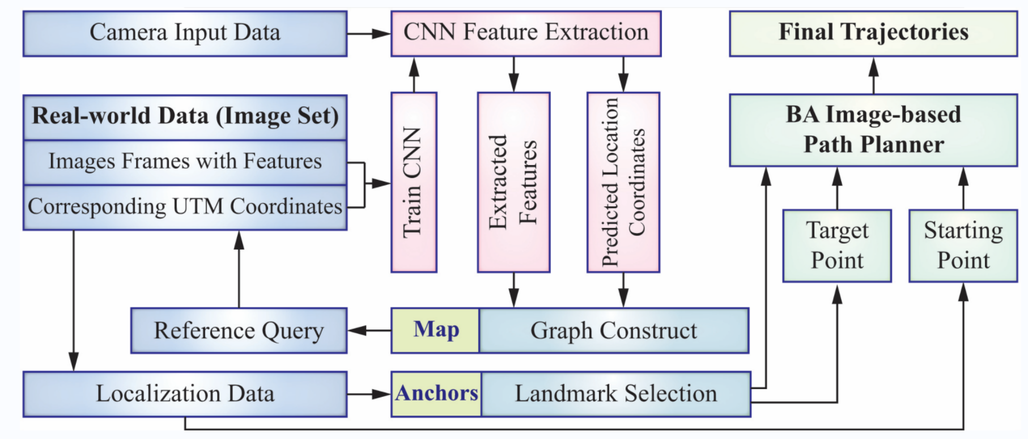


Figure 1. Proposed framework for Bat Algorithm in image-based path planning and localization using visual features and maps.

bat, and its position is regarded as the position of the bat. By implementing the BA, significant improvements in the accuracy of node positioning in practical and complicated environments have been demonstrated[49]. Additionally, an improved version of the BA combined with Differential Evolution (DE), known as IBA, has been proposed to optimize the three-dimensional path planning problem for Uninhabited Combat Air Vehicles (UCAV)[50]. This application of BA in UCAV path planning is notable because it was a novel approach at the time. The knowledge of this fact serves as the primary motivator for our current research. If BA can be used to plan and optimize 3D paths in combat aircraft, it is reasonable to believe that it could yield similar benefits for autonomous ground vehicles[51].

In the related work reviewed, *it is found* that no **BA** (or any other biologically inspired optimization) has been applied to the SLAM path planning problem. Since SLAM and seqSLAM utilize visual features and CNNs to produce a global path through a sequence of images, the implementation of a BA would be a novel and useful application of the optimization capabilities for optimizing the shortest path. This would involve computing a local path once the global trajectory has already been computed.

1.2. Proposed framework and original contributions

Studies on image-based navigation have become increasingly popular due to the increasing interest in their numerous applications in the field of autonomous vehicles[52,53]. In particular, several frameworks have been proposed and prototyped for image-based navigation. For example, Thoma *et al.*[37] developed a framework for image-based mapping, localization, and navigation of autonomous vehicles, and the algorithms for map constructions and self-localizations were discussed in detail. However, the algorithms required for path planning play an irreplaceable role in making an integrated imaged-based navigation system successful. Although limited work has been done on integrating path planning algorithms into learning systems, it has been noted that a reliable, biologically inspired, or nature-inspired optimization algorithm is missing from the image-feature-based navigation path planning models[54]. Therefore, this paper aims to develop a more robust path planner and evaluate the improved navigation results in comparison with Thoma *et al.*[37]. The BA is proposed and integrated as the state-of-the-art path planner into the program flow, and it is verified by measuring the performance of the outcomes based on the same datasets in the reference[55]. The flowchart of the proposed framework is presented in Figure 1.

The main contributions of this paper are summarized as follows:

- A biologically inspired image-based path planning and localization framework is proposed for robust, reliable, and effective image-based navigation.
- An image-based navigation is developed to use visual features and mapping that satisfy the require-

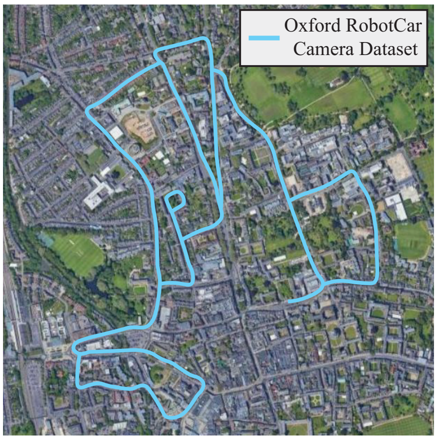


Figure 2. Oxford RobotCar map.

ments for image-based navigation and self-localization.

- The BA for image-based path planning is proposed to navigate the autonomous vehicle through the landmark identification on image-based reference query sequences.

The rest of this article is organized as follows. In Section 2, the image feature-based localization and map creation for image-based navigation are introduced. Section 3 presents the proposed BA for image-based navigation. In Section 4, the results of test cases, comparison studies, and simulations are presented, and performance characterization is provided for various maps, challenge datasets, and environments. Several important properties of the presented framework are summarized in Section 5.

2. IMAGE-BASED NAVIGATION

Image-based navigation is presented to describe our proposed nature inspired path planning algorithm. Image-based navigation is a navigation system that uses visual features and mapping to satisfy the requirements for image-based navigation and self-localization. It focuses on using reference images as a set of landmarks within a video sequence[37]. The publicly available dataset is utilized in this paper for evaluation. Figure 2 shows an aerial view of the outdoor dataset and outlines the route taken by the Oxford RobotCar from the acquired camera datasets, which will be adopted for our simulation studies.

Two sub-tasks handle compact map construction and accurate self-localization in this research:

(a) Finding of landmarks in a sub-sequence of the Oxford RobotCar run captured (see Figure 3)[37], and
(b) Matching to a short reference query sequence.

Precalculated distances are used based on features extracted from a VGG-16 + NetVLAD + whitening network. An Off-the-shelf Pitts30k model is utilized, which is freely available on the NetVLAD[29]. This solution is implemented using the NetVLAD TensorFlow. The modified implementation of their model produces the following output figures:

- Accuracy vs. distance plot of the final matching.
- Selected landmarks.
- Scatter plot of the original reference and query sequences shown in Figure 4.
- Topology of the reference sequence used for finding landmarks with network flow.

2.1. Image feature-based localization

Recently, there have been advances in camera technologies that allow them to be used in robotic applications as navigational sensors[30]. Cameras are used to provide the basic sensor input for localization, using robot pose

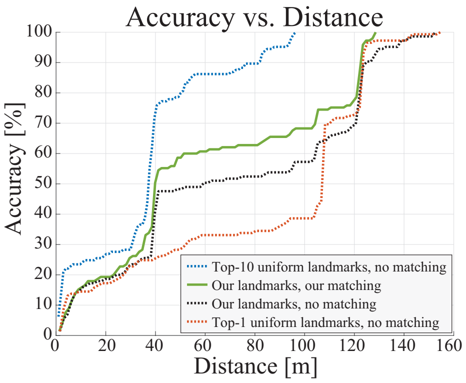


Figure 3. Accuracy vs. distance plot of the final selected landmark matching.

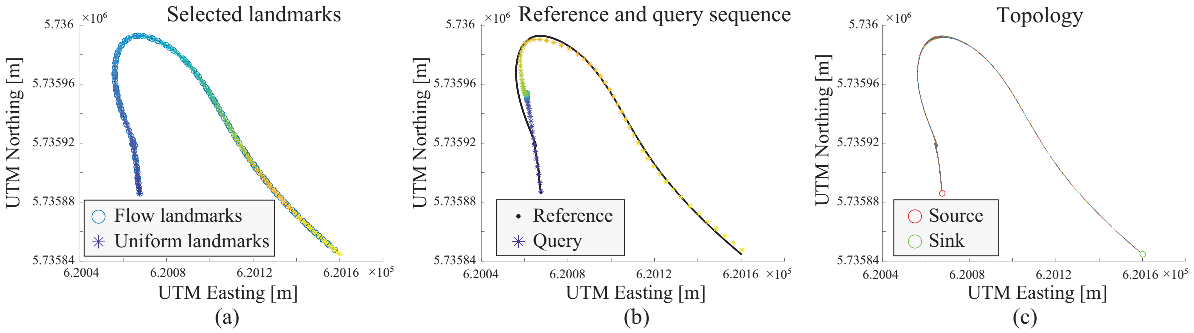


Figure 4. The scatter plot of the original reference, query sequences, and topology of reference sequence used for finding landmarks with network flow. UTM is Universal Transverse Mercator. (a) Selected landmarks. (b) Reference query. (c) Reference topology.

in each world environment[37]. Cameras deliver high quality image streams that enable feature identification, which may be employed to determine the location of the robot within the defined world environments. Images, for example, can still contain embedded information in the form of features such as points, lines, conics, spheres, and angles, which might be valuable for image-based navigation.

When images are coupled with the CNNs and appropriate filters are used to extract these features, localization can be possible. Image-based localization is a large topic that includes two main types of "worlds": those environments that are known a *priori* and others that are unknown. The method using a known environment consists of online and real-time mapping, and the latter is commonly known as SLAM[34]. SLAM methods can incorporate geometric, learned, topographical, and marker identification techniques[30]. The method considered in this study involves using a *known* environment of learned features, which will be integrated with the BA.

The method is a form of learned mapping that examine features that have been identified using two types of sources[30]. The first source is based on the VGG16 CNN coupled with off-the-shelf NetVLAD weights[29], while the second source is simply the fully connected output from the last VGG16 FC3 layer[37]. These features serve as the basis for lookup queries against the landmarks that have been established by a reference dataset of known location and geometry in this paper.

The geometric distances in the test data are calculated algorithmically and defined in the paper. The results of the query are then used to establish anchors and sensitivity values, which are used to identify the best waypoints between the identified landmarks. This network of points is best represented by a graph, representing the features where the Network Flow algorithm is then used to find the path with the lowest cost from source to destination.

The graph is represented by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where utilizing cost values c_{ij} and capacity values ω_{ij} to represent the edge $e_{ij} \in \mathcal{E}$ between images $\mathcal{I}_i$ and $\mathcal{I}_j$.

$$\omega_{ij} = \eta_{\mathcal{X}} \mathbf{d}\left(\mathcal{X}_i, \mathcal{X}_j\right) \text{ and } c_{ij} = \eta_{\mathcal{F}} / \mathbf{d}\left(\mathcal{F}_i, \mathcal{F}_j\right) \tag{1}$$

where $\eta_{\mathcal{X}}$ and $\eta_{\mathcal{F}}$ are weights of geometric and visual measures, respectively. Thus, $\mathbf{d}(\mathcal{X}_i, \mathcal{X}_j)$ and $\mathbf{d}(\mathcal{F}_i, \mathcal{F}_j)$ denote the geometric and visual distances between images $\mathcal{I}_i$ and $\mathcal{I}_j$, respectively.

To generate visually comparable matches, the visual distance between landmarks $\mathcal{V}' = \{v'_i\}_{i=1}^{n}$ and a sequence of images $\mathcal{I}_\rho$ is utilized to calculate the flow cost rate at corresponding locations $\{x_i\}_{i=1}^{n}$. The flows from the source to landmarks and from query pictures to the target incur no additional cost. In addition, we offer a robust loss for feature matching with associated cost values c_{ij} and capacity values ω_{ij} defined as:

$$c_{ij} = \mathcal{L}(\boldsymbol{d}(\mathcal{F}_i, \mathcal{F}_j)), \forall e_{ij} \in \mathcal{E}_\pi; c_{ij} = 0, \forall e_{ij} \in \mathcal{E} \setminus \mathcal{E}_\pi \tag{2}$$

where $\mathcal{L}(.)$ is the Huber loss function. Images are matched to (at most) a single landmark, so the maximum absolute flow at each query image is limited to a value of one, thus creating the capacity constraint:

$$\omega_{ij} = q, \forall e_{ij} \in \mathcal{E}_{v'}; \omega_{ij} = 1, \forall e_{ij} \in \mathcal{E} \setminus \mathcal{E}_{v'} \tag{3}$$

with $\mathcal{E}_{v'}$ and $\mathcal{E}_\pi$ representing the directed edges spanning the partition of a bipartite graph between vertices of landmarks and image sequences, respectively. Given a radius of navigation r, and source and target vertices $\{s, t\}$, the flows $\{y_{ij}\}$ are solved for by application of Second Order Cone Programming (SOCP) to the resulting convex optimization problem:

$$\begin{aligned} min_{y_{ij}} \sum_{e_{ij} \in \mathcal{E}} c_{ij} y_{ij}, \\ 0 \le y_{ij} \le \omega_{ij}, \quad \forall_{e_{ij}} \in \mathcal{E}, \\ y_s = q, y_i = 0, y_T = -q, \forall v_i \in \mathcal{V} \setminus (s \cup t), \\ \| \sum_{v_i \in \mathcal{V}'} x_i y_{i(l+1)} - \sum_{v_i \in \mathcal{V}'} x_i y_{il} \| \le r, \forall \pi_l, \pi_{l+1} \in \Pi \end{aligned} \tag{4}$$

The solution of Equation (4) is used as the location, x_l, of the query image $\mathcal{I}_l$ indicated by the vertex $\pi_l \in \Pi$.

2.2. Map creation from neural network features

The output from the NN is a feature map matrix of nodes, edges, and geographic distances between objects that correspond to actual physical locations from a small reference query (see Figure 5a). Therefore, in our developed model, a graph is created to represent the map of free space that is to be explored/traversed by our bats (Figure 6). All nodes in this graph correspond to geographic coordinates of features extracted from along the global path taken by the Oxford RobotCar (Figure 5b). These coordinates are also provided as a matrix of Northing/Easting values used in most GPS navigation systems. Section 3 provides further details on the use of the Bat Algorithm for image-based navigation in this research.

3. BAT ALGORITHM FOR IMAGE-BASED NAVIGATION

In this section, the BA for image-based path planning is proposed in consideration of the image maps obtained in Section 2.

3.1. Bat algorithm

The BA is a fresh technology inspired by the social behaviors of bats and their use of echolocation for distance sensing. It pertains to the swarm intelligence family of optimization algorithms. The BA is based on the premise that certain echolocation qualities are idealized, as specified in the following specific rules.

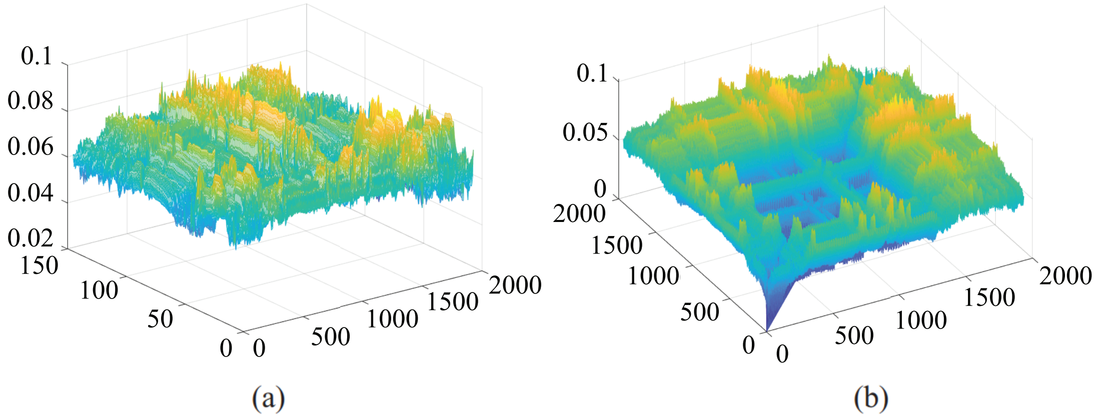


Figure 5. Neural network image feature space. (a) Matched reference sequence feature space. (b) Features space extracted from the RobotCar dataset.

- Bats apply echolocation to sense distance and always have knowledge of their surrounding environment.
- Bats fly randomly with a velocity v_i and a fixed frequency f_{min} at a position x_i, varying the wavelength λ, and loudness A_0 as they hunt for prey. They are able to instantly sense the wavelength of their emitted pulses and adjust the rate of the pulse emission $r \in [0, 1]$ based on the proximity of their target.
- The loudness of the emission varies from a minimum constant (positive) A_{min} to a large A_0.

In the BA, each bat is defined by its position x_i^t, velocity v_i^t, frequency f_i, loudness A_i^t, and the emission pulse rate r_i^t in a d-dimensional search space. The new solutions x_i^t and velocities v_i^t at time step t are given by

$$\begin{aligned} f_i &= f_{min} + (f_{max} - f_{min})\beta, \\ \boldsymbol{v}_i^t &= \boldsymbol{v}_i^{t-1} + (\boldsymbol{x}_i^t - \boldsymbol{x}_*) f_i, \\ \boldsymbol{x}_i^t &= \boldsymbol{x}_i^{t-1} + \boldsymbol{v}_i^t, \end{aligned} \tag{5}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. $\boldsymbol{x}_*$ is the current global optimized solution from among all n bats. In general, the values that are typically assigned to the frequency range of the BA fall between 0 and 100, such that $f_{min} = 0$ and $f_{max} = 100$, and initially, each bat in the population is given a frequency that is drawn uniformly from $[f_{min}, f_{max}]$.

During the search portion of the algorithm, the current best solution is used to calculate a new solution for each bat using a random walk.

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{old} + \epsilon A^t, \tag{6}$$

where $\epsilon \in [-1, 1]$ is a random number scaling factor and $A^t = \langle A_i^t \rangle$ is the average loudness of all bats at some time step t. Velocities and positions are updated in a similar manner to the standard updates in a PSO, where f_i essentially controls the pace and range of movement of bats.

Indeed, the BA can be considered a balanced combination of standard PSO and an intensive local search controlled by loudness and pulse rate. The loudness and rate of the pulse are updated as follows:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t, \\ r_i^{t+1} &= r_i^0 [1 - e^{(-\gamma t)}], \end{aligned} \tag{7}$$

where α and γ are constants set to $\alpha = \gamma = 0.9$ for the purposes of this paper.

3.2. Image-based bat algorithm path planning

Now that a primer to the BA and the computer-vision-based navigational topics have been presented, the integration method with the BA used to realize the combined algorithm is discussed in this section.

It is desirable to build an implementation that can integrate the essential features of the straightforward BA with the image-based navigation optimization technique. Instead of using external optimization libraries, such as the MOS optimization library, for selecting landmarks using visual features, our version of the BA is enhanced to include Dijkstra's Algorithm (DA) with the shortest path finding. The DA is incorporated into the BA to provide solutions for the routes created by individual bats.

Algorithm 1: Bat Algorithm for Image-based Navigation

```
Initialize the generation counter $t = 1$;
Randomly initialize population of $N_P$ bats and each bat $P$ corresponding to a potential solution to the
  given problem;
Initialize loudness $A_i$, pulse frequency $Q_i$, pulse rate $r_i$, and initial velocities $v_i (i = 1, 2, ..., N_P)$;
while $t < MaxGeneration$ do
    Generate new solutions by adjusting frequency, and updating velocities and locations/solutions per
      Equation (5);
    if $rand > r_i$ then
        Select best solution;
        Generate local solution from best;
    end
    Generate a new solution by flying randomly;
    if $rand < A_i$ and $f(x_i) < f(x_*)$ then
        Accept the new solutions;
        Increase $r_i$ and reduce $A_i$;
    end
    Rank bats & find current best $\boldsymbol{x}_*$;
    $t = t + 1$;
end
Post-processing results and visualization
```

In algorithm 1, the BA is shown with two principal procedures defined. The first procedure is used to initialize all of the algorithm parameters required during the iterations of the computational steps, and the second procedure describes the actual computational flow. In the compute solution procedure, random adjustments to frequencies and velocities generate and select new solutions from the individual bats at any given time step. The best solution is carried forward to subsequent time steps until either the maximum iterations are reached or the optimal solution is found. The Equation (5)-Equation (7) are used for each of the operations referred to by the psuedocode presented below. Figure 6 illustrates the normalized feature coordinates in the Northing/Easting plane as nodes, and the length of the interconnecting line segments correspond to the distances between each feature illustrated in Figure 6. Thus, finding the shortest path through this network will also be the shortest path through the real geometry that is represented by the elements of the graph in this paper. By using both the graph and the coordinate matrices in this paper, a reference topological map and its coordinates can be created and then provided as arguments to the BA.

3.3. Algorithm complexity

The proposed image-based navigation involves the following steps:

(1) Capturing images of the environment;

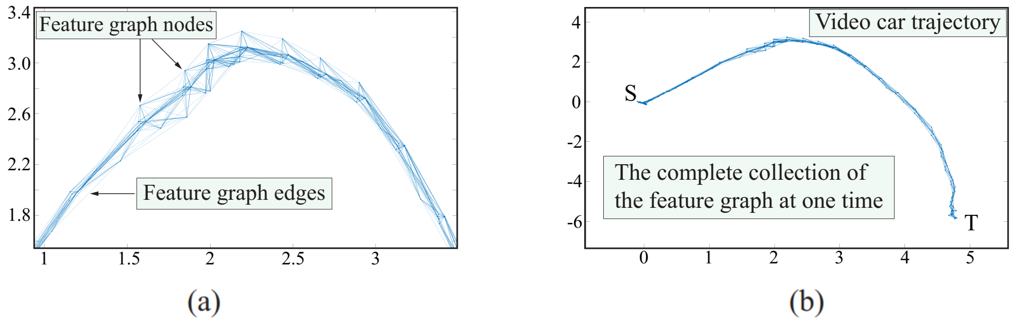


Figure 6. Feature graph nodes (spatial coordinates) and edges (geographic distance between features). (a) Closeup of feature graph. (b) Relative feature graph.

(2) Processing and extracting visual features from the images;
(3) Identifying the location of the vehicle using the extracted features;
(4) Generating a path for the vehicle based on the location information.

The time complexity of image processing and feature extraction depends on the complexity of the algorithms used, the size and resolution of the images, and the number of features extracted. For example, CNNs used for feature extraction can have a time complexity of $O(n^2)$, where n is the number of pixels in the image. However, various optimization techniques, such as down sampling and parallel processing, can be used to reduce the computational complexity.

The time complexity of BA used for image-based navigation in this paper depends on the complexity of the BA and the size of the environment. The worst-case time complexity of the BA is $O(\mathcal{N} * \mathcal{I} * \mathcal{C})$, where $\mathcal{N}$ is the number of bats, $\mathcal{I}$ is the maximum number of iterations, and $\mathcal{C}$ is the time complexity of the objective function evaluation. However, in practice, the number of iterations required is much smaller, making the algorithm computationally efficient. The time complexity of the BA is dependent on the size and complexity of the input image datasets (landmarks) and the convergence rate of the algorithm. Besides, the BA is known for its efficiency and effectiveness in solving complex optimization problems.

The time complexity of the proposed image-based navigation is dependent on the properties of the environment being explored. The BA, as an efficient algorithm and optimization technique, is used to reduce the computational complexity and improve the performance of image-based navigation.

4. SIMULATION AND COMPARISON STUDIES

In this section, simulation and comparison studies are undertaken to demonstrate the effectiveness, feasibility, and robustness of our proposed BA path planning method. In the first experiment, simulations are conducted by utilizing different maps, and the results are compared with other state-of-the-art path planning algorithms. In the second experiment, the proposed BA method is applied to the map of real-world scenarios with random simulated image-based landmarks. In the third experiment, we apply this algorithm to image-based navigation in real environments (The image-based datasets are taken by the Oxford RobotCar).

4.1. Simulation and comparison studies in various environments

To evaluate our proposed BA model, we first use it within a grid map and compare it with the PSO algorithm, Fuzzy NN (FNN), and Hybrid PSO and FNN Algorithm (HPFA) from[38] in Scenario 1. Two grid-based maps are selected in Figure 7 and Figure 8. The size of the map is 30 × 30. The population size of BA is set to 50. The parameter settings and experimental results of other algorithms are in[38]. The comparative results of a variety of path planning methods are illustrated in Figure 7, which demonstrates that all four algorithms

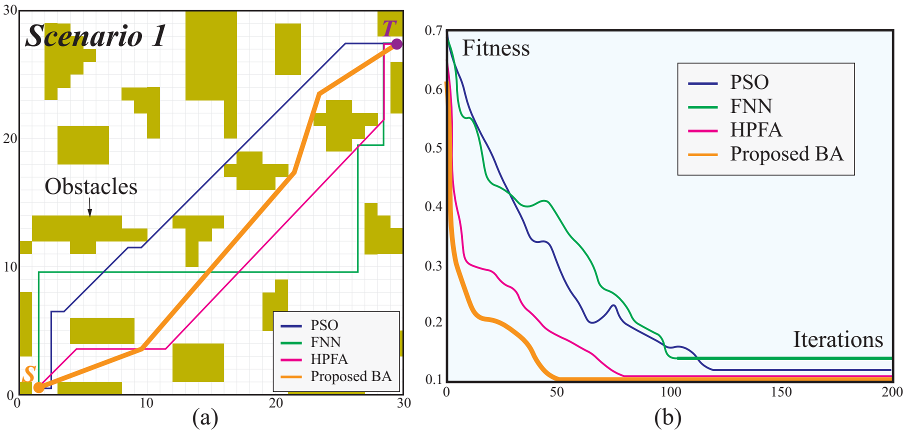


Figure 7. Path planning results in Scenario 1 from [38]. (a) Generated trajectories of PSO, FNN, HPFA, and proposed BA. (b) Convergence of PSO, FNN, HPFA, and proposed BA.

Table 1. Comparison of maximum path length, minimum path length, and average processing time of PSO, FNN, HPFA, and proposed BA

Algorithm	Max length	Min length	Average time (ms)
PSO	85.71	46.97	1639
FNN	102.89	48.28	1921
HPFA	74.76	40.76	*1426*
Proposed BA	**42.56**	**33.62**	**1597**

can effectively plan the way from the start to the target points in the grid map environment. However, the trajectory planned by the proposed BA is much shorter than the paths generated by the other three techniques. Table 1 summarizes the qualitative comparison between the features of our algorithm and selected algorithms, including maximum path length, minimum path length, and average processing time.

The simulation results in terms of the convergence of various methods are shown in Figure 7(b). The fitness function converges to a pretty high value after around 100 iterations of the PSO algorithm, as shown by the findings, which indicate that the method first converges more quickly and subsequently slows down significantly. The FNN method is distinct in that its convergence speed increases, and it converges after around 120 iterations; nevertheless, the fitness function converges to a higher value than the PSO algorithm. The HPFA method has a relatively better convergence result, while its convergence speed is slow. After around 50 iterations, the proposed BA method converges to the lowest fitness function among the four methods, and thus the convergence curve turbulence is minor.

In order to further verify the optimization and path planning capabilities of the proposed BA, we conducted another grid-based environment comparison study with Ant Colony Optimization (ACO), Genetic Algorithm (GA), A* Algorithm Optimization (AAO), Kth Shortest Path Algorithm (KSPA), and HPFA in Scenario 2. The simulation results provided in Figure 8(a) reveal that the path length of the proposed BA algorithm is substantially smoother and shorter than the other five methods. Figure 8(b) summarizes the length findings of the trajectories. When compared to previous grid map path planning algorithms, our developed BA approach has superior convergence and optimal solution. This might highlight the viability of the proposed BA in path planning.

To validate the adaptability and efficiency of our algorithm in various environments, one map with resolution 1500 × 1500 from Massachusetts Roads Dataset[56] is then selected for simulation and comparative studies. The initial and multiple target positions of the autonomous vehicle are randomly set. The initial position is

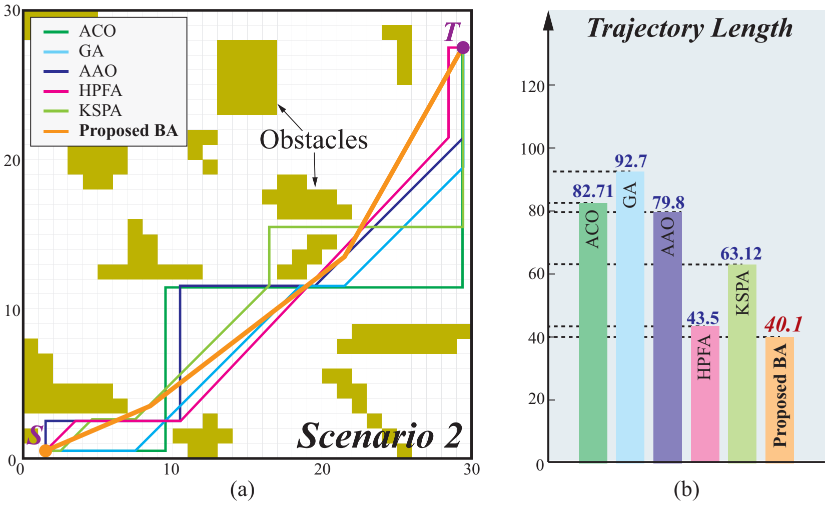


Figure 8. Path planning results in Scenario 2 from[38]. (a) Generated trajectories of ACO, GA, AAO, HPFA, KSPA, and proposed BA. (b) Trajectories length of ACO, GA, AAO, HPFA, KSPA, and proposed BA.

(1108, 1433), target positions are $T_1 = (189, 1062)$, $T_2 = (381, 163)$, $T_3 = (1431, 640)$, and $T_4 = (728, 611)$, as shown in Figure 9. Two cutting-edge path planning algorithms, Probabilistic Roadmap (PRM) and Rapidly Exploring Random Tree* (RRT*), are utilized for comparative studies. PRM is a well-known and effective path planning method based on sampling. This approach can seek a solution utilizing a limited number of random sampling points and takes minimal processing time. It generates random sampling points in the free space of a given workspace to create a route network graph. Then, use Dijkstra to search the created route network graph for suitable routes. The number of PRM sampling points in this study is 1500. RRT* is also a sampling-based path planning algorithm similar to PRM. From the initial location, it randomly generates the sampling point to the spanning tree in the workspace and links it to the nearest obstacle-free point on the path tree to the sampling point. In all situations, the maximum RRT* iteration time is set to 500000, and the maximum connection distance is set to 4. We repeatedly execute 40 times in the selected scenario. Table 2 provides a qualitative assessment of the path length, smoothness rate, and execution time of the algorithm in comparison to other algorithms. The path smoothness rate ψ is based on the sum of the angles differences between adjacent path segments calculated by Equation (8).

$$\psi = \sum_{i=2}^{n-1} \text{abs}\left(\theta_{i+1} - \theta_i\right) \tag{8}$$

$$\theta_{i+1} = \text{atan}\left[\left(y_{i+1} - y_i\right) / \left(x_{i+1} - x_i\right)\right] \tag{9}$$

$$\theta_i = \text{atan}\left[\left(y_i - y_{i-1}\right) / \left(x_i - x_{i-1}\right)\right] \tag{10}$$

The results indicate that path planning methods, such as PRM and RRT*, are insufficient when the sample points are inadequate or the distribution of the map is inappropriate. Although the RRT* method has a lower execution time, the proposed BA approach has a significant advantage in terms of path length and smoothness.

4.2. Image-based navigation with simulated datasets

Image-based navigation utilizes reference pictures as a collection of landmarks in a video sequence to meet image-based navigation and self-localization navigation techniques. However, due to insufficient datasets, the

Table 2. Comparison of path length, path smoothness rate, algorithm execution time, and success rate of PRM, RRT*, and the proposed Bat Algorithm. These values are reported as the result of 40 executions

$S \Longrightarrow T$	Model name	Minimum path length	Average path length	Smoothness rate (rad)	Average execution time (sec)	Success rate ($\%$)
$S \Longrightarrow T_1$	PRM	1251.3	1690.5	1.0099	598.07	67.5
	RRT*	1384.8	144.88	0.6701	**30.84**	95
	BA	**1073.5**	**1246.2**	**0.2133**	329.65	**100**
$T_1 \Longrightarrow T_2$	PRM	1287.4	1403.8	1.1263	482.10	77.5
	RRT*	1505.3	1589.0	0.9239	**18.63**	**100**
	BA	**993.44**	**1079.4**	**0.1357**	134.72	**100**
$T_2 \Longrightarrow T_3$	PRM	1507.9	1540.1	0.7502	500.02	72.5
	RRT*	1402.0	1448.8	0.6137	**27.13**	92.5
	BA	**1252.4**	**1246.2**	**0.2133**	329.65	**100**
$T_3 \Longrightarrow T_4$	PRM	1292.3	1318.6	0.2328	494.10	62.5
	RRT*	825.94	884.39	0.5306	**35.66**	95
	BA	**735.48**	**801.27**	**0.2133**	329.65	**100**

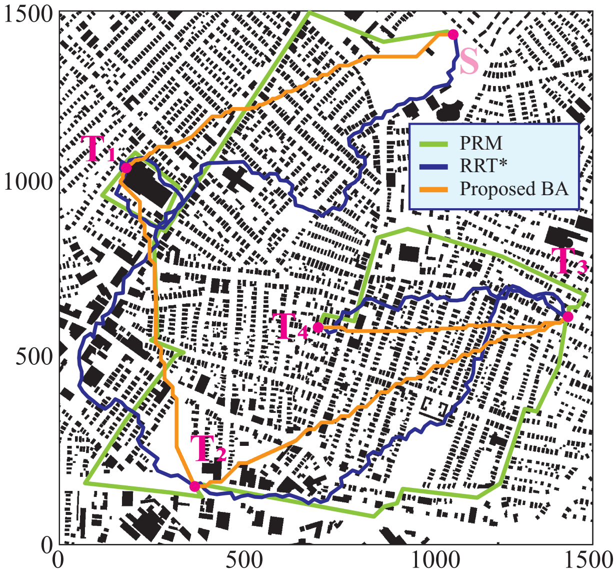


Figure 9. Illustration of the real-world map from Massachusetts Roads Dataset, composed of 1500×1500 nodes. The pink circle marks are starting and target points. The green line, blue line, and orange line denote the trajectory of PRM, RRT*, and our proposed BA, respectively.

navigation experiment is conducted by utilizing simulated images as landmarks in this paper. It should be noted that since the reference image or the captured video is dependent extremely on the road segment passed by the previous video car, its generated landmarks are limited and random in the environment. The image-based landmarks we simulated have the same limitations, being only within a random portion of the map [Figure 10]. Therefore, to evaluate our proposed BA model, the image-based landmarks that are simulated have the same limitations, being only within a random portion of the map (Figure 10). Two algorithms, Dijkstra's algorithm and Slime Mound Algorithm (SMA)[57], are selected for comparison. The test workspace is a real-world map, as New York City map from the benchmark[58]. 100 landmarks and 150 landmark data are simulated in Figure 10(a) and Figure 10(b), respectively. Among them, in Figure 10(a) and Figure 10(b), the green and yellow lines indicate the feasibility of connecting between landmarks (spatial coordinates) in the topological map, which are geographic distances between features. Purple, pink, and orange lines in Figure 10(a) and Figure 10(b) represent the final trajectories of Dijkstra's algorithm, SMA, and the proposed BA, respectively. Table 3 outlines the length of the trajectory planned by the tested algorithms. The results demonstrate that in the complex connection topological map, our proposed BA algorithm can obtain shorter paths than the Dijkstra's algorithm and SMA.

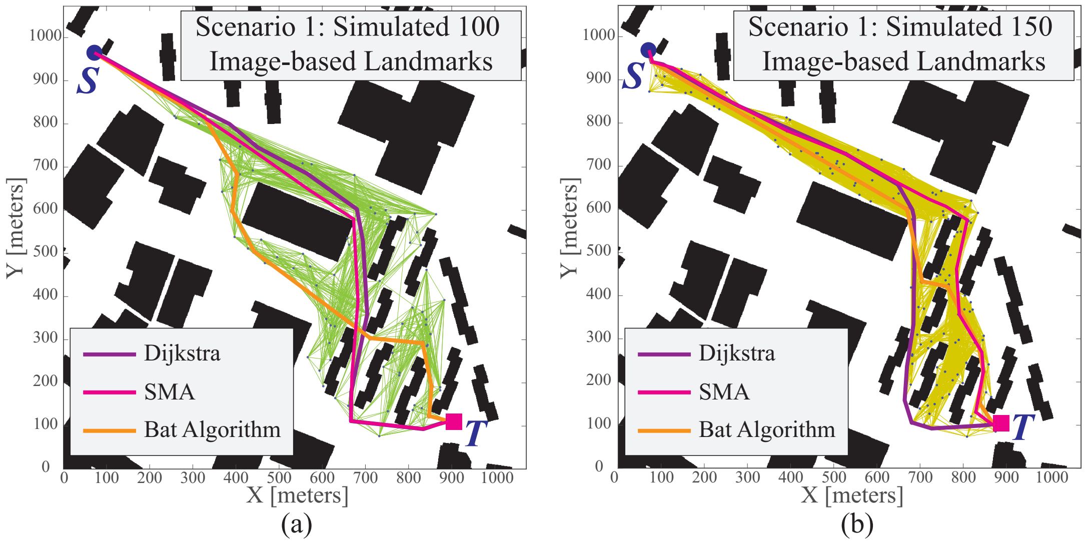


Figure 10. Comparison of the trajectory by the proposed Bat Algorithm image-based navigation with basic Dijkstra's algorithm. The image-based landmark datasets are randomly generated in the New York benchmark map[58].

Table 3. Comparison of the Dijkstra's algorithm and SMA with proposed BA in simulated image-based landmarks datasets

Scenario	Simulated image-based landmarks	Algorithm	Path length
		Dijkstra	775.1
Figure 10(a)	100	SMA	768.7
		Proposed BA	**710.9**
		Dijkstra	771.2
Figure 10(b)	150	SMA	743.9
		Proposed BA	**700.2**

4.3. Image-based navigation with Oxford RobotCar Datasets

The simulation and comparison studies in this section aim to validate the proposed image-based BA navigation in real-world datasets. The image-based datasets are taken by the Oxford RobotCar. The proposed BA path planner initially obtains the solid green trajectory with the real world maps, as illustrated in Figure 11. While the updated dataset finds that the road segment is interrupted by an obstacle (as a blue polygon-shaped obstacle in Figure 11), BA can replan the trajectory in light of the existing dataset. The dashed purple and pink trajectories reveal the proposed image-based BA navigation with the replanned procedure. The detail of the proposed image-based BA navigation is accomplished by modifying and building the BA algorithm with an integrated Dijkstra's algorithm. The input to the BA algorithm is the graph representation of the reference query sequence shown in Figure 6 and a matrix of indices that corresponds to the feature coordinates.

Once the BA algorithm is functional, it is converted and integrated at the correct location to process the image-feature graph. Output compares with the Thoma's output[37]. The BA generated a series of graphs that are superimposed on the images presented in Section 3. Figure 12 also displays the best-fit values of each BA iteration to illustrate how swiftly the BA converges to its optimal solution. Several simulations were carried out utilizing our integrated technique, and the findings shown here are illustrative of typical worst-case convergence times. Each simulation result is accompanied by a brief caption and a zoomed-in view, allowing the reader to gain a clearer sense of the volume and scale of each of the features being processed in the graph. The intention is to demonstrate how the BA output compares to the Thoma's output[37].

As can be seen in Figure 12 above, the BA was able to converge upon an optimal solution in 33 iterations. This represents the worst-case results observed during the experimentation phase of this project. On average, convergence occurs in approximately 5-10 iterations in our studies. In the results shown in Figure 13 and Figure 14, the BA output is overlayed against the original simulation outputs from the[37]. It is clear from these

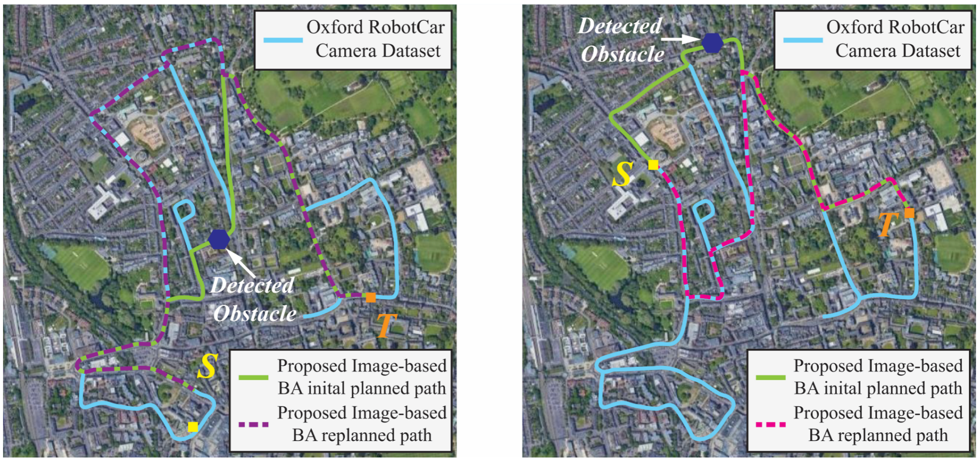


Figure 11. Illustration of the overall trajectory by the proposed image-based Bat Algorithm navigation. The image-based datasets are taken by the Oxford RobotCar.

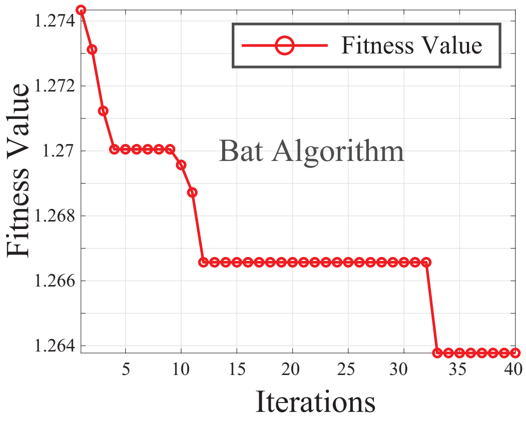


Figure 12. Number of iterations until Bat Algorithm convergence.

plots that the BA finds the optimal path through the given topography and does not hinder or otherwise alter the effectiveness of the global navigation strategy.

4.4. Analysis and discussions

Several simulations using our developed hybrid model were performed during the course of testing this implementation, with the results of the BA proving to be adept at traversal of the known topography produced by the image feature-based recognition. This is an encouraging result since new feature identification strategies are the topic of much research today and are likely to achieve even better results over the next few years. The true test of whether or not the algorithm can be used in real world environments will come once a dynamic obstacle avoidance method can be added to the algorithm.

We made many attempts to integrate the BA algorithm at various locations in the processing pipeline, but it became clear that the geographic feature distances were not *calculated in* the[37] algorithm but were instead *processed by* the algorithm. This makes the identification of the best place to perform the task of integration very difficult and may point to the need to have optimization algorithms performed up front in the process, potentially eliminating the need for other proposed methods entirely (future work). Nevertheless, the best implementation of BA would take place where the path-optimizer and the graph-traversal libraries can be switched out with their equivalent optimization biologically inspired implementation.

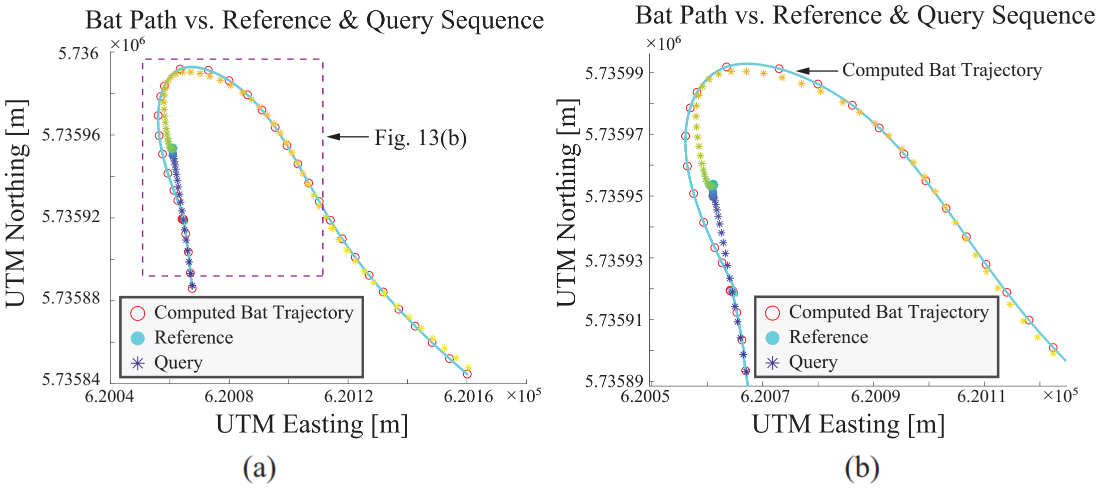


Figure 13. Reference query with Bat Algorithm solution superimposed. UTM is Universal Transverse Mercator. (a) Bat Algorithm solution vs. reference query. (b) Zoomed solution vs. reference query.

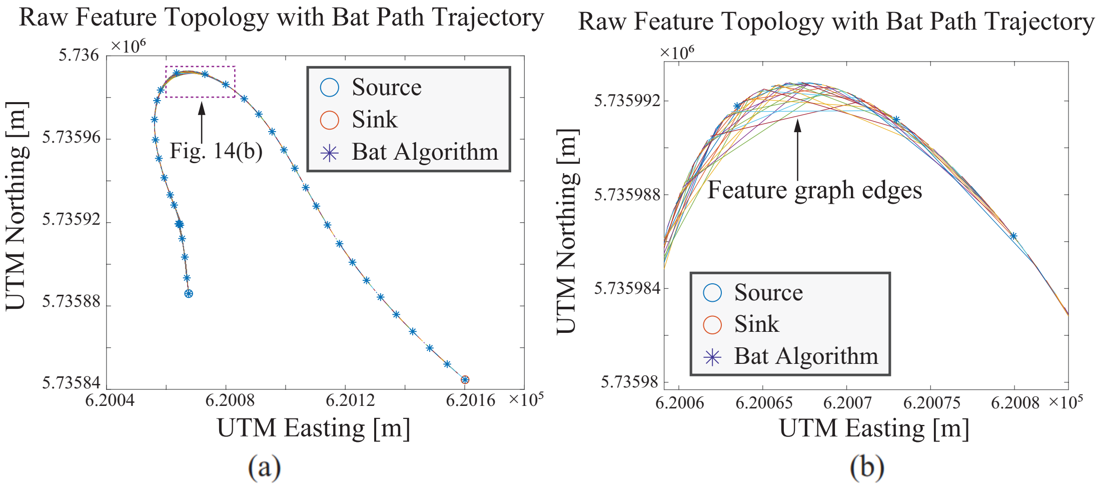


Figure 14. Raw feature topology with Bat Algorithm solution superimposed. (a) Bat Algorithm solution vs. raw feature topology. (b) Zoomed Bat Algorithm solution vs. raw feature topology.

The full image feature extraction process was explored here to process the raw image features into the format that is best suited to a dynamic real-time path planner with inherent obstacle avoidance such as the D*-Lite algorithm[59]. Thoma *et al.*[37] integrated a localization method against a reference map, but to be fully integrated with a path planning algorithm, it needs to be more functional in a real-world sense. Other researchers have focused on competing technologies, such as using PSO methodology to train FNNs[38] focusing on the optimization of parameters during the front-end NN feature extraction. Yet, others have used reinforcement learning techniques to select an optimal path in ultra-dynamic emergency environments through hybrid NNs and A* algorithms[54]. Their reasoning is to focus entirely on the back-end processing to help intelligent vehicles plan dynamic routes around traffic emergency conditions, including limitations on vehicle height, width, weight restrictions, or accidents and traffic jams.

These approaches differ from the current research in that they focus on optimizing both before and long after map building occurs. The need for a simultaneous SLAM-like solution remains. When compared with other biologically inspired algorithms for path planning in image-based systems, the BA has shown that it is capable of the challenge, but it will also need to prove that it can be deployed in a mature and accurate localizing methodology (which has been demonstrated on a known dataset, but not on a live video camera feed). For these reasons, it would make the most sense to integrate a parallel architecture with the path planner that has access to those same raw features as the SLAM method given in the article. The local map that is built by the raw camera feed is used to generate a current coordinate location for the autonomous vehicle body within the region of the visible and identified map. The path planning algorithm, however, would be capable of resolving a path to the goal location from the current localized coordinates up to the extreme edge of the map, which could presumably be incomplete or inaccurate due to the sensor errors. A heuristic distance to the end goal coordinate could be used as a guideline to navigate in the local region, avoiding and replanning based on dynamic obstacle avoidance, and as new information becomes available in the dataset, continue to work toward that goal location.

In future research, an experiment and design to implement the BA into a Field Programmable Gate Array (FPGA) would also help to keep the burden of constant re-calculation down and help to distribute the task of feature extraction to a specific piece of parallel modular hardware. An interface to the input camera feed can be sent to the SLAM module and, simultaneously, to the embedded hardware housing the FPGA.

5. CONCLUSION

Despite the potentiality of image-based navigation for the usage of an autonomous vehicle in an unknown environment, current studies on system development focused largely on the algorithms for mapping, feature retrievals, and localization. However, these studies in this paper concluded that selecting a robust, reliable, and efficient path planning algorithm was at least as important to the success of an image-based navigation system.

The goal of the presented work was to develop an original Bat path planning algorithm. By introducing the BA at a point where the graph of features is calculated, we were able to show how a biologically inspired algorithm can be used to enhance the SLAM performance with a local path planner. The results presented here show that the BA can produce path planning results that are not only optimized but also maintain the integrity of the image-based feature recognition and self-localization upon which it has been added.

It was our finding that during the overall integration of the BA into the image-based system framework, it became apparent that there is an increased need to gain direct access to the raw image features as they are extracted from the NNs themselves. Although it was the original intent of this paper to work within the given framework, it became clear that the computational expense of this methodology is high. Hence, for our future exploration, it should be prudent to modify the research parameters to implement a more parallel approach to the BA that aided in the path planning by simultaneously sharing information between the two systems. This type of solution would also benefit from having FPGA hardware integrated that could perform the tasks of DNNs in real time and assist in distributing those features to the BA and the SLAM algorithm in parallel.

DECLARATIONS

Acknowledgments

The authors would like to thank the editor-in-chief, the associate editor, and the anonymous reviewers for their valuable comments.

Authors' contributions

Made substantial contributions to the research, idea generation, algorithm design, and simulation and wrote and edited the original draft: Short D, Lei T, Carruth D, and Luo C

Performed critical review, commentary, and revision and provided administrative, technical, and material support: Carruth D, Bi Z

Financial support and sponsorship

This research was supported by the Mississippi Space Grant Consortium under the NASA EPSCoR RID grant.

Availability of data and materials

Not applicable.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Wang L, Luo C, Li M, Cai J. Trajectory planning of an autonomous mobile robot by evolving ant colony system. *Int J Robot Autom* 2017;32:406–13. DOI
2. Lei T, Chintam P, Carruth DW, Jan GE, Luo C. Human-autonomy teaming-based robot informative path planning and mapping algorithms with tree search mechanism. In: 2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS). IEEE; 2022. pp. 1–6. DOI
3. Zhao W, Lun R, Gordon C, et al. A privacy-aware Kinect-based system for healthcare professionals. In: IEEE International Conference on Electro Information Technology (EIT); 2016. pp. 0205–10. DOI
4. Jayaraman E, Lei T, Rahimi S, Cheng S, Luo C. Immune system algorithms to environmental exploration of robot navigation and mapping. In: Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II 12. Springer; 2021. pp. 73–84. DOI
5. Lei T, Chintam P, Luo C, Rahimi S. Multi-robot directed coverage path planning in row-based environments. In: 2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). IEEE; 2022. pp. 114–21. DOI
6. Zhu D, Yan T, Yang SX. Motion planning and tracking control of unmanned underwater vehicles: technologies, challenges and prospects. *IR* 2022;2:200–22. DOI
7. Li J, Xu Z, Zhu D, et al. Bio-inspired intelligence with applications to robotics: a survey. *IR* 2021;1:58–83. DOI
8. Lei T, Luo C, Ball JE, Rahimi S. A graph-based ant-like approach to optimal path planning. In: IEEE Congress on Evolutionary Computation (CEC); 2020. pp. 1–6. DOI
9. Chen J, Luo C, Krishnan M, Paulik M, Tang Y. An enhanced dynamic Delaunay triangulation-based path planning algorithm for autonomous mobile robot navigation. In: Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques. vol. 7539. SPIE; 2010. pp. 253–64. DOI
10. Sellers T, Lei T, Luo C, Jan GE, Ma J. A node selection algorithm to graph-based multi-waypoint optimization navigation and mapping. *IR* 2022;2:333–54. DOI
11. Lei T, Sellers T, Luo C, Zhang L. A bio-inspired neural network approach to robot navigation and mapping with nature-inspired algorithms. In: International Conference on Swarm Intelligence. Springer; 2022. pp. 3–16. DOI
12. Luo C, Yang SX, Meng MQH. Neurodynamics based complete coverage navigation with real-time map building in unknown environments. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2006. pp. 4228–33. DOI
13. Sellers T, Lei T, Jan GE, Wang Y, Luo C. Multi-objective optimization robot navigation through a graph-driven PSO mechanism. In: Advances in Swarm Intelligence: 13th International Conference, ICSI 2022, Xi'an, China, July 15–19, 2022, Proceedings, Part II. Springer; 2022. pp. 66–77. DOI
14. Jan GE, Luo C, Hung LP, Shih ST. A computationally efficient complete area coverage algorithm for intelligent mobile robot navigation. In: 2014 International Joint Conference on Neural Networks (IJCNN). IEEE; 2014. pp. 961–66. DOI
15. Lei T, Luo C, Ball JE, Bi Z. A hybrid fireworks algorithm to navigation and mapping. In: Handbook of Research on Fireworks Algorithms and Swarm Intelligence. IGI Global; 2020. pp. 213–32. DOI
16. Li X, Li X, Khyam MO, Luo C, Tan Y. Visual navigation method for indoor mobile robot based on extended BoW model. CAAI Transactions on Intelligence Technology 2017;2:142–47. DOI
17. Lei T, Luo C, Jan GE, Bi Z. Deep learning-based complete coverage path planning with re-joint and obstacle fusion paradigm. *Front Robot AI* 2022;9. DOI

18. Wang D, Yang SX. Intelligent feature extraction, data fusion and detection of concrete bridge cracks: current development and challenges. *IR* 2022;2:391–406. DOI
19. Lei T, Li G, Luo C, et al. An informative planning-based multi-layer robot navigation system as applied in a poultry barn. *IR* 2022;2:313–32. DOI
20. Ortiz S, Yu W. Autonomous navigation in unknown environment using sliding mode SLAM and genetic algorithm. *IR* 2021;1:131–50. DOI
21. Lei T, Sellers T, Rahimi S, Cheng S, Luo C. A nature-inspired algorithm to adaptively safe navigation of a Covid-19 disinfection robot. In: International Conference on Intelligent Robotics and Applications. Springer; 2021. pp. 123–34. DOI
22. Luo C, Gao J, Murphey YL, Jan GE. A computationally efficient neural dynamics approach to trajectory planning of an intelligent vehicle. In: 2014 International Joint Conference on Neural Networks (IJCNN). IEEE; 2014. pp. 934–39. DOI
23. Liu L, Luo C, Shen F. Multi-agent formation control with target tracking and navigation. In: IEEE International Conference on Information and Automation (ICIA); 2017. pp. 98–103. DOI
24. Zhao W, Lun R, Gordon C, et al. Liftingdoneright: a privacy-aware human motion tracking system for healthcare professionals. *IJHCR* 2016;7:1–15. DOI
25. Lei T, Luo C, Jan GE, Fung K. Variable speed robot navigation by an ACO approach. In: International Conference on Swarm Intelligence. Springer; 2019. pp. 232–42. DOI
26. Luo C, Yang SX, Mo H, Li X. Safety aware robot coverage motion planning with virtual-obstacle-based navigation. In: 2015 IEEE International Conference on Information and Automation. IEEE; 2015. pp. 2110–15. DOI
27. Luo C, Yang SX, Krishnan M, Paulik M. An effective vector-driven biologically-motivated neural network algorithm to real-time autonomous robot navigation. In: IEEE International Conference on Robotics and Automation (ICRA); 2014. pp. 4094–99. DOI
28. Zhu D, Tian C, Jiang X, Luo C. Multi-AUVs cooperative complete coverage path planning based on GBNN algorithm. In: 29th Chinese Control and Decision Conference (CCDC); 2017. pp. 6761–66. DOI
29. Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp. 5297–307. DOI
30. Maltar J, Marković I, Petrović I. Visual place recognition using directed acyclic graph association measures and mutual information-based feature selection. *Rob Auton Syst* 2020;132:103598. DOI
31. Cao F, Liu B, Park DS. Image classification based on effective extreme learning machine. *Neurocomputing* 2013;102:90–97. DOI
32. Muthusamy H, Polat K, Yaacob S. Improved emotion recognition using gaussian mixture model and extreme learning machine in speech and glottal signals. *Math Probl Eng* 2015;2015. DOI
33. Yang Y, Deng Q, Shen F, Zhao J, Luo C. A shapelet learning method for time series classification. In: IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI); 2016. pp. 423–30. DOI
34. Cieslewski T, Choudhary S, Scaramuzza D. Data-efficient decentralized visual SLAM. In: 2018 IEEE International Conference on Robotics and Automation (ICRA); 2018. pp. 2466–73. DOI
35. Oishi S, Inoue Y, Miura J, Tanaka S. SeqSLAM++: View-based robot localization and navigation. *Rob Auton Syst* 2019;112:13–21. DOI
36. Otte M. A survey of machine learning approaches to robotic path-planning. University of Colorado at Boulder; 2008. PhD Preliminary Exam.
37. Thoma J, Paudel DP, Chhatkuli A, Probst T, Gool LV. Mapping, localization and path planning for image-based navigation using visual features and map. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 7383–91. DOI
38. Liu X, Zhang D, Zhang J, Zhang T, Zhu H. A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. *Cluster Comput* 2021;24:1901–15. DOI
39. Zhang X, Yang Z, Cao F, et al. Conditioning optimization of extreme learning machine by multitask beetle antennae swarm algorithm. *Memet ComputMemet Comput* 2020;12:151–64. DOI
40. Chu Z, Wang F, Lei T, Luo C. Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance. *IEEE Trans Intell Veh* 2023;8:108–20. DOI
41. Dai P, Taghia J, Lam S, Katupitiya J. Integration of sliding mode based steering control and PSO based drive force control for a 4WS4WD vehicle. *Auton Robot* 2018;42:553–68. DOI
42. Teng F, Zhang H, Luo C, Shan Q. Delay tolerant containment control for second-order multi-agent systems based on communication topology design. *Neurocomputing* 2020;380:11–19. DOI
43. Chu Z, Sun B, Zhu D, Zhang M, Luo C. Motion control of unmanned underwater vehicles via deep imitation reinforcement learning algorithm. *IEEE trans Intell Transp Syst* 2020;14:764–74. DOI
44. Shishika D, Paley DA. Mosquito-inspired distributed swarming and pursuit for cooperative defense against fast intruders. *Auton Robot* 2019;43:1781–99. DOI
45. Lei T, Luo C, Sellers T, Rahimi S. A bat-pigeon algorithm to crack detection-enabled autonomous vehicle navigation and mapping. *Intelligent Systems with Applications* 2021;12:200053. DOI
46. Zhao QS, Hu YL. Multidimensional scaling localisation algorithm based on bacterial colony chemotaxis optimisation. *IEEE Trans Mob Comput* 2016;11:151. DOI
47. Pham DT, Castellani M. A comparative study of the bees algorithm as a tool for function optimisation. *Cogent Eng* 2015;2:1091540. DOI
48. Li X, Xiao S, Wang C, Yi J. Mathematical modeling and a discrete artificial bee colony algorithm for the welding shop scheduling problem. *Memet Comput* 2019;11:371–89. DOI
49. Tu D, Wang E, Zhang F. An intelligent wireless sensor positioning strategy based on improved bat algorithm. In: 2019 International

Conference on Intelligent Transportation, Big Data & Smart City (ICITBS); 2019. pp. 174–77. DOI

50. Wang GG, Chu HE, Mirjalili S. Three-dimensional path planning for UCAV using an improved bat algorithm. *Aerosp Sci Technol* 2016;49:231–38. DOI
51. Lei T, Luo C, Sellers T, Wang Y, Liu L. Multitask allocation framework with spatial dislocation collision avoidance for multiple aerial robots. *IEEE Trans Aerosp Electron Syst* 2022;58:5129–40. DOI
52. Chu Z, Xiang X, Zhu D, Luo C, Xie D. Adaptive trajectory tracking control for remotely operated vehicles considering thruster dynamics and saturation constraints. *ISA Trans* 2020;100:28–37. DOI
53. Zhang Q, Luo R, Zhao D, Luo C, Qian D. Model-free reinforcement learning based lateral control for lane keeping. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE; 2019. pp. 1–7. DOI
54. Liu X, Zhang D, Zhang T, et al. Novel best path selection approach based on hybrid improved A* algorithm and reinforcement learning. *Appl Intell* 2021;51:9015–29. DOI
55. Yang J, Chai T, Luo C, Yu W. Intelligent demand forecasting of smelting process using data-driven and mechanism model. *IEEE Trans Ind Electron* 2018;66:9745–55. DOI
56. Mnih V. Machine learning for aerial image labeling. University of Toronto; 2013.
57. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. *FGCS* 2020;111:300–323. DOI
58. Sturtevant NR. Benchmarks for grid-based pathfinding. *IEEE T COMP INTEL AI* 2012;4:144–48. DOI
59. Koenig S, Likhachev M. Fast replanning for navigation in unknown terrain. *IEEE Trans Robot* 2005;21:354–63. DOI

AUTHOR INSTRUCTIONS

1. Submission Overview

Before you decide to publish with *Intelligence & Robotics* (*IR*), please read the following items carefully and make sure that you are well aware of Editorial Policies and the following requirements.

1.1 Topic Suitability

The topic of the manuscript must fit the scope of the journal. Please refer to Aims and Scope for more information.

1.2 Open Access and Copyright

The journal adopts Gold Open Access publishing model and distributes content under the Creative Commons Attribution 4.0 International License. Copyright is retained by authors. Please make sure that you are well aware of these policies.

1.3 Publication Fees

IR is an open access journal. When a paper is accepted for publication, authors are required to pay Article Processing Charges (APCs) to cover its editorial and production costs. The APC for each submission is $1200. There are no additional charges based on color, length, figures, or other elements. For more details, please refer to OAE Publication Fees.

1.4 Language Editing

All submissions are required to be presented clearly and cohesively in good English. Authors whose first language is not English are advised to have their manuscripts checked or edited by a native English speaker before submission to ensure the high quality of expression. A well-organized manuscript in good English would make the peer review even the whole Editorial handling more smoothly and efficiently.

If needed, authors are recommended to consider the language editing services provided by OAE to ensure that the manuscript is written in correct scientific English before submission. An extra charge is required to enjoy this service. Please visit https://www.oaepublish.com/index/author_services or contact English-Editing@oaepublish.com for more details.

1.5 Work Funded by the National Institutes of Health

If an accepted manuscript was funded by National Institutes of Health (NIH), the author may inform editors of the NIH funding number. The editors are able to deposit the paper to the NIH Manuscript Submission System on behalf of the author.

2. Submission Preparation

2.1 Cover Letter

A cover letter is required to be submitted accompanying each manuscript. Here is a guideline of a cover letter for authors' consideration:
List the highlights of the current manuscript and no more than 5 short sentences;
All authors have read the final manuscript, have approved the submission to the journal, and have accepted full responsibilities pertaining to the manuscript's delivery and contents;
Clearly state that the manuscript is an original work on its own merit, that it has not been previously published in whole or in part, and that it is not being considered for publication elsewhere;
No materials are reproduced from another source (if there is material in your manuscript that has been reproduced from another source, please state whether you have obtained permission from the copyright holder to use them);
Conflicts of interest statement;
If the manuscript is contributed to a Special Issue, please also mention it in the cover letter;
If the manuscript was presented partly or entirely in a conference, the author should clearly state the background information of the event, including the conference name, time, and place in the cover letter.

2.2 Types of Manuscripts

There is no restriction on the length of manuscripts, number of figures, tables and references, provided that the manuscript is concise and comprehensive. The journal publishes Research Article, Review, Technical Note, etc. For more details about paper type, please refer to the following table.

Manuscript Type	Definition	Word Limit	Abstract	Keywords	Main Text Structure

Research Article	A Research Article is a seminal and insightful research study and showcases that often involves modern techniques or methodologies. Authors should justify that their work is of novel findings.	8000 max	The abstract should state briefly the purpose of the research, the principal results and major conclusions. No more than 250 words.	3-8 keywords	The main content should include four sections: Introduction, Methods, Results and Discussion.
Review	A Review should be an authoritative, well balanced, and critical survey of recent progress in an attractive or a fundamental research field.	5000 max /10000 max	Unstructured abstract. No more than 250 words.	3-8 keywords	The main text may consist of several sections with unfixed section titles. We suggest that the author include an "Introduction" section at the beginning, several sections with unfixed titles in the middle part, and a "Conclusions" section at the end.
Technical Note	A Technical Note is a short article giving a brief description of a specific development, technique, or procedure, or it may describe a modification of an existing technique, procedure or device applied in research.	3500 max	Unstructured abstract. No more than 250 words.	3-8 keywords	/
Editorial	An Editorial is a short article describing news about the journal or opinions of senior Editors or the publisher.	1000 max	None required	None required	/
Commentary	A Commentary is to provide comments on a newly published article or an alternative viewpoint on a certain topic.	2500 max	Unstructured abstract. No more than 250 words.	3-8 keywords	/
Perspective	A Perspective provides personal points of view on the state-of-the-art of a specific area of knowledge and its future prospects.	2000 max	Unstructured abstract. No more than 250 words.	3-8 keywords	/

2.3 Manuscript Structure

2.3.1 Front Matter

2.3.1.1 Title

The title of the manuscript should be concise, specific and relevant, with no more than 16 words if possible.

2.3.1.2 Authors and Affiliations

Authors' full names should be listed. The initials of middle names can be provided. The affiliations and email addresses for all authors should be listed. At least one author should be designated as the corresponding author. In addition, corresponding authors are suggested to provide their Open Researcher and Contributor ID upon submission. Please note that any change to authorship is not allowed after manuscript acceptance. The authors' affiliations should be provided in this format: department, institution, city, postcode, country.

2.3.1.3 Abstract

The abstract should be a single paragraph with word limitation and specific structure requirements (for more details please refer to Types of Manuscripts). It usually describes the main objective(s) of the study, explains how the study was done, including any model organisms used, without methodological detail, and summarizes the most important results and their significance. The abstract must be an objective representation of the study: it is not allowed to contain results that are not presented and substantiated in the manuscript, or exaggerate the main conclusions. Citations should not be included in the abstract.

2.3.1.4 Graphical Abstract

The graphical abstract is essential as this can catch first view of your publication by readers. We recommend you submit an eye-catching figure. It should summarize the content of the article in a concise graphical form. It is recommended to use it because this can make online articles get more attention.

The graphical abstract should be submitted as a separate document in the online submission system. Please provide an image with a minimum of 531 × 1328 pixels (h × w) or proportionally more. The image should be readable at a size of 5 cm × 13 cm using a regular screen resolution of 96 dpi. Preferred file types: TIFF, PSD, AI, JPEG, and EPS files.

2.3.1.5 Keywords

Three to eight keywords should be provided, which are specific to the article, yet reasonably common within the subject discipline.

Sections 2.3.1.1 and 2.3.1.2 should appear in all manuscript types.

2.3.2 Main Text

Manuscripts of different types are structured with different sections of content. Please refer to Types of Manuscripts to make sure which sections should be included in the manuscripts.

2.3.2.1 Introduction

The introduction should contain background that puts the manuscript into context, allow readers to understand why the study is important, include a brief review of key literature, and conclude with a brief statement of the overall aim of the work and a comment about whether that aim was achieved. Relevant controversies or disagreements in the field should be introduced as well.

2.3.2.2 Methods

The methods should contain sufficient details to allow others to fully replicate the study. New methods and protocols should be described in detail while well-established methods can be briefly described or appropriately cited. Statistical terms, abbreviations, and all symbols used should be defined clearly. Protocol documents for clinical trials, observational studies, and other non-laboratory investigations may be uploaded as supplementary materials.

2.3.2.3 Results

This section contains the findings of the study. Results of statistical analysis should also be included either as text or as tables or figures if appropriate. Authors should emphasize and summarize only the most important observations. Data on all primary and secondary outcomes identified in the section Methods should also be provided. Extra or supplementary materials and technical details can be placed in supplementary documents.

2.3.2.4 Discussion

This section should discuss the implications of the findings in context of existing research and highlight limitations of the study. Future research directions may also be mentioned.

2.3.2.5 Conclusion

It should state clearly the main conclusions and include the explanation of their relevance or importance to the field.

2.3.3 Back Matter

The following sections should appear in all manuscript types.

2.3.3.1 Acknowledgments

Anyone who contributed towards the article but does not meet the criteria for authorship, including those who provided professional writing services or materials, should be acknowledged. Authors should obtain permission to acknowledge from all those mentioned in the Acknowledgments section. This section is not added if the author does not have anyone to acknowledge.

2.3.3.2 Authors' Contributions

Each author is expected to have made substantial contributions to the conception or design of the work, or the acquisition, analysis, or interpretation of data, or the creation of new software used in the work, or have drafted the work or substantively revised it.

Please use Surname and Initial of Forename to refer to an author's contribution. For example: made substantial contributions to conception and design of the study and performed data analysis and interpretation: Salas H, Castaneda WV; performed data acquisition, as well as providing administrative, technical, and material support: Castillo N, Young V.

If an article is single-authored, please include "The author contributed solely to the article." in this section.

2.3.3.3 Availability of Data and Materials

In order to maintain the integrity, transparency and reproducibility of research records, authors should include this section in their manuscripts, detailing where the data supporting their findings can be found. Data can be deposited into data repositories or published as supplementary information in the journal. Authors who cannot share their data should state that the data will not be shared and explain it. If a manuscript does not involve such issues, please state "Not applicable." in this section.

2.3.3.4 Financial Support and Sponsorship

All sources of funding for the study reported should be declared. The role of the funding body in the experiment design, collection, analysis and interpretation of data, and writing of the manuscript should be declared. Any relevant grant numbers and the link of funder's website should be provided if any. If the study is not involved with this issue, state "None." in this section.

2.3.3.5 Conflicts of Interest

Authors must declare any potential conflicts of interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. If there are no conflicts of interest, please state "All authors declared that there are no conflicts of interest." in this section. Some authors may be bound by confidentiality agreements. In such cases, in place of itemized disclosures, we will require authors to state "All authors declared that they are bound by confidentiality agreements that prevent them from disclosing their conflicts of interest in this work.". If authors are unsure whether conflicts of interest exist, please refer to the "Conflicts of Interest" of *IR* Editorial Policies for a full explanation.

2.3.3.6 Consent for Publication

Manuscripts containing individual details, images or videos, must obtain consent for publication from that person, or in the case of children, their parents or legal guardians. If the person has died, consent for publication must be obtained from the next of kin of the participant. Manuscripts must include a statement that written informed consent for publication was obtained. Authors do not have to submit such content accompanying the manuscript. However, these documents must be available if requested. If the manuscript does not involve this issue, state "Not applicable." in this section.

2.3.3.7 Copyright

Authors retain copyright of their works through a Creative Commons Attribution 4.0 International License that clearly states how readers can copy, distribute, and use their attributed research, free of charge. A declaration "© The Author(s) 2023." will be added to each article. Authors are required to sign License to Publish before formal publication.

2.3.3.8 References

References should be numbered in order of appearance at the end of manuscripts. In the text, reference numbers should be placed in square brackets and the corresponding references are cited thereafter. List all authors when the number of authors is less than or equal to six, if there are more than six authors, only the first three authors' names should be listed, other authors' names should be omitted and replaced with "et al.". The journal's name should be required to be italicized and the journal references should have corresponding DOI numbers. Information from manuscripts accepted but not published should be cited in the text as "Unpublished material" with written permission from the source. Journal names should be abbreviated according to the List of Title Word Abbreviations.

References should be described as follows, depending on the types of works:

Types	Examples

Journal articles by individual authors	Cao MS, Pan LX, Gao YF, et al. Neural network ensemble-based parameter sensitivity analysis in civil engineering systems. *Neural Comput Applic* 2017;28:1583-90. [DOI: 10.1007/s00521-015-2132-4]
Organization as author	Diabetes Prevention Program Research Group. Hypertension, insulin, and proinsulin in participants with impaired glucose tolerance. *Hypertension* 2002;40:679-86. [DOI: 10.1161/01.HYP.0000035706.28494.09]
Both personal authors and organization as author	Vallancien G, Emberton M, Harving N, van Moorselaar RJ; Alf-One Study Group. Sexual dysfunction in 1,274 European men suffering from lower urinary tract symptoms. *J Urol* 2003;169:2257-61. [PMID: 12771764 DOI: 10.1097/01.ju.0000067940.76090.73]
Journal articles not in English	Mao X, Ding YK. Sentiment feature analysis and harmonic sense evaluation of images. *J Electronic* 2001;29:23-7. (in Chinese)
Journal articles ahead of print	Cong Y, Gu CJ, Zhang T, Gao YJ. Underwater Robot Sensing Technology: A Survey. *Fundamental Res* 2021; Epub ahead of print [DOI: 10.1016/j.fmre.2021.03.002]
Books	Gaydon AG, Wolfhard HG. Flames. 2nd ed. London: Chapman and Hall Ltd.; 1960. pp. 10-20.
Book chapters	Goel AK, Fitzgerald T, Parashar P. Analogy and metareasoning: Cognitive strategies for robot learning. In: Lawless WF, Mittu R, Sofge DA, Editors. Human-Machine Shared Contexts. Academic Press; 2020. pp. 23-44.
Online resource	Intel Technology Journal. Developing smart toys - from idea to product. Available from: https://www.intel.com/content/dam/www/public/us/en/documents/research/2001-vol05-iss-4-intel-technology-journal.pdf. [Last accessed on 20 Feb 2021]
Conference proceedings	Harnden P, Joffe JK, Jones WG, Editors. Germ cell tumours V. Proceedings of the 5th Germ Cell Tumour Conference; 2001 Sep 13-15; Leeds, UK. New York: Springer; 2002.
Conference paper	Christensen S, Oppacher F. An analysis of Koza's computational effort statistic for genetic programming. In: Foster JA, Lutton E, Miller J, Ryan C, Tettamanzi AG, editors. Genetic programming. EuroGP 2002: Proceedings of the 5th European Conference on Genetic Programming; 2002 Apr 3-5; Kinsdale, Ireland. Berlin: Springer; 2002. pp. 182-91.
Unpublished material	Tian D, Araki H, Stahl E, Bergelson J, Kreitman M. Signature of balancing selection in Arabidopsis. *Proc Natl Acad Sci U S A*. Forthcoming 2002.

The journal also recommends that authors prepare references with a bibliography software package, such as EndNote to avoid typing mistakes and duplicated references.

2.3.3.9 Supplementary Materials

Additional data and information can be uploaded as Supplementary Materials to accompany the manuscripts. The supplementary materials will also be available to the referees as part of the peer-review process. Any file format is acceptable, such as data sheet (word, excel, csv, cdx, fasta, pdf or zip files), presentation (powerpoint, pdf or zip files), image (cdx, eps, jpeg, pdf, png or tiff), table (word, excel, csv or pdf), audio (mp3, wav or wma) or video (avi, divx, flv, mov, mp4, mpeg, mpg or wmv). All information should be clearly presented. Supplementary materials should be cited in the main text in numeric order (e.g., Supplementary Figure 1, Supplementary Figure 2, Supplementary Table 1, Supplementary Table 2, etc.). The style of supplementary figures or tables complies with the same requirements on figures or tables in main text. Videos and audios should be prepared in English, and limited to a size of 500 MB.

2.4 Manuscript Format

2.4.1 File Format

Manuscript files can be in DOC and DOCX formats and should not be locked or protected.

Manuscript prepared in LaTex must be collated into one ZIP folder (including all source files and images, so that the Editorial Office can recompile the submitted PDF).

When preparing manuscripts in different file formats, please use the corresponding Manuscript Templates.

2.4.2 Length

The word limit is specified in the item “Types of Manuscripts”. There are no restrictions on number of figures or number of supporting documents. Authors are encouraged to present and discuss their findings concisely.

2.4.3 Language

Manuscripts must be written in English.

2.4.4 Multimedia Files

The journal supports manuscripts with multimedia files. The requirements are listed as follows:

Video or audio files are only acceptable in English. The presentation and introduction should be easy to understand. The frames should be clear, and the speech speed should be moderate;

A brief overview of the video or audio files should be given in the manuscript text;

The video or audio files should be limited to a size of up to 500 MB;

Please use professional software to produce high-quality video files, to facilitate acceptance and publication along with the submitted article. Upload the videos in mp4, wmv, or rm format (preferably mp4) and audio files in mp3 or wav format.

2.4.5 Figures

Figures should be cited in numeric order (e.g., Figure 1, Figure 2) and placed after the paragraph where it is first cited;

Figures can be submitted in format of TIFF, PSD, AI, EPS or JPEG, with resolution of 300-600 dpi;

Figure caption is placed under the Figure;

Diagrams with describing words (including, flow chart, coordinate diagram, bar chart, line chart, and scatter diagram, *etc*.) should be editable in word, excel or powerpoint format. Non-English information should be avoided;

Labels, numbers, letters, arrows, and symbols in figure should be clear, of uniform size, and contrast with the background;

Symbols, arrows, numbers, or letters used to identify parts of the illustrations must be identified and explained in the legend;

Internal scale (magnification) should be explained and the staining method in photomicrographs should be identified;

All non-standard abbreviations should be explained in the legend;

Permission for use of copyrighted materials from other sources, including re-published, adapted, modified, or partial figures and images from the internet, must be obtained. It is authors' responsibility to acquire the licenses, to follow any citation instruction requested by third-party rights holders, and cover any supplementary charges.

2.4.6 Tables

Tables should be cited in numeric order and placed after the paragraph where it is first cited;

The table caption should be placed above the table and labeled sequentially (e.g., Table 1, Table 2);

Tables should be provided in editable form like DOC or DOCX format (picture is not allowed);

Abbreviations and symbols used in table should be explained in footnote;

Explanatory matter should also be placed in footnotes;

Permission for use of copyrighted materials from other sources, including re-published, adapted, modified, or partial tables from the internet, must be obtained. It is authors' responsibility to acquire the licenses, to follow any citation instruction requested by third-party rights holders, and cover any supplementary charges.

2.4.7 Abbreviations

Abbreviations should be defined upon first appearance in the abstract, main text, and in figure or table captions and used consistently thereafter. Non-standard abbreviations are not allowed unless they appear at least three times in the text. Commonly-used abbreviations, such as DNA, RNA, ATP, *etc*., can be used directly without definition. Abbreviations in titles and keywords should be avoided, except for the ones which are widely used.

2.4.8 Italics

General italic words like *vs*., *et al*., *etc*., *in vivo*, *in vitro*; *t* test, *F* test, *U* test; related coefficient as *r*, sample number as *n*, and probability as *P*; names of genes; names of bacteria and biology species in Latin.

2.4.9 Units

SI Units should be used. Imperial, US customary and other units should be converted to SI units whenever possible. There is a space between the number and the unit (i.e., 23 mL). Hour, minute, second should be written as h, min, s.

2.4.10 Numbers

Numbers appearing at the beginning of sentences should be expressed in English. When there are two or more numbers in a paragraph, they should be expressed as Arabic numerals; when there is only one number in a paragraph, number < 10 should be expressed in English and number > 10 should be expressed as Arabic numerals. 12345678 should be written as 12,345,678.

2.4.11 Equations

Equations should be editable and not appear in a picture format. Authors are advised to use either the Microsoft Equation Editor or the MathType for display and inline equations.

Display equations should be numbered consecutively, using Arabic numbers in parentheses;

Inline equations should not be numbered, with the same/similar size font used for the main text.

2.4.12 Headings

In the main body of the paper, three different levels of headings may be used.

Level one headings: they should be in bold, and numbered using Arabic numbers, such as **1. INTRODUCTION**, and **2. METHODS**, with all letters capitalized;

Level two headings: they should be in bold and numbered after the level one heading, such as **2.1 Statistical analyses**, **2.2** ..., **2.3**..., *etc*., with the first letter capitalized;

Level three headings: they should be italicized, and numbered after the level two heading, such as *2.1.1 Data distributions*,and *2.1.2 outliers and linear regression*, with the first letter capitalized.

2.4.13 Text Layout

As the electronic submission will provide the basic material for typesetting, it is important to prepare papers in the general editorial style of the journal.
The font is Times New Roman;
The font size is 12pt;
Single column, 1.5× line spacing;
Insert one line break (one Return) before the heading and paragraph, if the heading and paragraph are adjacent, insert a line break before the heading only;
No special indentation;
Alignment is left end;
Insert consecutive line numbers;
For other details please refer to the Manuscript Templates.

2.5 Submission Link

Submit an article via https://oaemesas.com/login?JournalId=ir.

3. Publication Ethics Statement

OAE is a member of the Committee on Publication Ethics (COPE). We fully adhere to its Code of Conduct and to its Best Practice Guidelines.

The Editors of this journal enforce a rigorous peer-review process together with strict ethical policies and standards to guarantee to add high-quality scientific works to the field of scholarly publication. Unfortunately, cases of plagiarism, data falsification, image manipulation, inappropriate authorship credit, and the like, do arise. The Editors of *IR* take such publishing ethics issues very seriously and are trained to proceed in such cases with zero tolerance policy.

Authors wishing to publish their papers in *IR* must abide by the following:

The author(s) must disclose any possibility of a conflict of interest in the paper prior to submission;
The authors should declare that there is no academic misconduct in their manuscript in the cover letter;
Authors should accurately present their research findings and include an objective discussion of the significance of their findings;
Data and methods used in the research need to be presented in sufficient detail in the manuscript so that other researchers can replicate the work;
Authors should provide raw data if referees and the Editors of the journal request;
Simultaneous submission of manuscripts to more than one journal is not tolerated;
Republishing content that is not novel is not tolerated (for example, an English translation of a paper that is already published in another language will not be accepted);
The manuscript should not contain any information that has already been published. If you include already published figures or images, please get the necessary permission from the copyright holder to publish under the CC-BY license;
Plagiarism, data fabrication and image manipulation are not tolerated;
Plagiarism is not acceptable in OAE journals.

Plagiarism involves the inclusion of large sections of unaltered or minimally altered text from an existing source without appropriate and unambiguous attribution, and/or an attempt to misattribute original authorship regarding ideas or results, and copying text, images, or data from another source, even from your own publications, without giving credit to the source.

As to reusing the text that is copied from another source, it must be between quotation marks and the source must be cited. If a study's design or the manuscript's structure or language has been inspired by previous studies, these studies must be cited explicitly.

If plagiarism is detected during the peer-review process, the manuscript may be rejected. If plagiarism is detected after publication, we may publish a Correction or retract the paper.

Falsification is manipulating research materials, equipment, or processes, or changing or omitting data or results so that the findings are not accurately represented in the research record.

Image files must not be manipulated or adjusted in any way that could lead to misinterpretation of the information provided by the original image.

Irregular manipulation includes: introduction, enhancement, moving, or removing features from the original image; the grouping of images that should be presented separately, or modifying the contrast, brightness, or color balance to obscure, eliminate, or enhance some information.

If irregular image manipulation is identified and confirmed during the peer-review process, we may reject the manuscript. If irregular image manipulation is identified and confirmed after publication, we may publish a Correction or retract the paper.

OAE reserves the right to contact the authors' institution(s) to investigate possible publication misconduct if the Editors find conclusive evidence of misconduct before or after publication. OAE has a partnership with iThenticate, which is the most trusted similarity checker. It is used to analyze received manuscripts to avoid plagiarism to the greatest extent possible. When plagiarism becomes evident after publication, we will retract the original publication or require modifications, depending on the degree of plagiarism, context within the published article, and its impact on the overall integrity of the published study. Journal Editors will act under the relevant COPE guidelines.

4. Authorship

Authorship credit of *IR* should be solely based on substantial contributions to a published study, as specified in the following four criteria:
1. Substantial contributions to the conception or design of the work, or the acquisition, analysis, or interpretation of data for the work;
2. Drafting the work or revising it critically for important intellectual content;
3. Final approval of the version to be published;
4. Agreement to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

All those who meet these criteria should be identified as authors. Authors must specify their contributions in the section Authors' Contributions of their manuscripts. Contributors who do not meet all the four criteria (like only involved in acquisition of funding, general supervision of a research group, general administrative support, writing assistance, technical editing, language editing, proofreading, *etc*.) should be acknowledged in the section of Acknowledgement in the manuscript rather than being listed as authors.

If a large multiple-author group has conducted the work, the group ideally should decide who will be authors before the work starts and confirm authors before submission. All authors of the group named as authors must meet all the four criteria for authorship.
AI and AI-assisted technologies should not be listed as an author or co-author.

5. Reviewers Exclusions

You are welcome to exclude a limited number of researchers as potential Editors or reviewers of your manuscript. To ensure a fair and rigorous peer review process, we ask that you keep your exclusions to a maximum of three people. If you wish to exclude additional referees, please explain or justify your concerns—this information will be helpful for Editors when deciding whether to honor your request.

6. Editors and Journal Staff as Authors

Editorial independence is extremely important and OAE does not interfere with Editorial decisions. Editorial staff or Editors shall not be involved in processing their own academic work. Submissions authored by Editorial staff/Editors will be assigned to at least two independent outside reviewers. Decisions will be made by the Editor-in-Chief, including Special Issue papers. Journal staff are not involved in the processing of their own work submitted to any OAE journals.

7. Policy of the Use of AI and AI-assisted Technologies in Scientific Writing

Generative AI and AI-assisted technologies (e.g., large language models) are expected to be increasingly used to create content. In the writing process of manuscripts, using AI and AI-assisted technologies to complete key researcher work, such as producing scientific insights, analyzing and interpreting data or drawing scientific conclusions, is not allowed, and they should only be used to improve the readability and language of manuscripts.
AI and AI-assisted technologies should be used under human control and supervision as they may generate incorrect or prejudiced output, and they should not be listed as an author or co-author, nor cited as an author.
The use of AI and AI-assisted technologies should be disclosed by authors in their manuscripts, and a statement will be required in the final publication.
OAE will keep monitoring the development and adjust the policy when necessary.

8. Conflict of Interests

OAE journals require authors to declare any possible financial and/or non-financial conflicts of interest at the end of their manuscript and in the cover letter, as well as confirm this point when submitting their manuscript in the submission system. If no conflicts of interest exist, authors need to state "All authors declared that there are no conflicts of interest". We also recognize that some authors may be bound by confidentiality agreements, in which cases authors need to state "All authors declared that they are bound by confidentiality agreements that prevent them from disclosing their competing interests in this work".

9. Editorial Process

9.1. Pre-Check

New submissions are initially checked by the Managing Editor from the perspectives of originality, suitability, structure and formatting, conflicts of interest, background of authors, *etc*. Poorly prepared manuscripts may be rejected at this stage. If your manuscript does not meet one or more of these requirements, we will return it for further revisions.

Once your manuscript has passed the initial check, it will be assigned to the Assistant Editor, and then the Editor-in-Chief, or an Associate Editor in the case of a conflict of interest, will be notified of the submission and invited to review. Regarding Special Issue paper, after passing the initial check, the manuscript will be successively assigned to the Assistant Editor, and then to the Editor-in-Chief, or an Associate Editor in the case of conflict of interest for the Editor-in-Chief to review. The Editor-in-Chief, or the Associate Editor may reject manuscripts that they deem highly unlikely to pass peer review without further consultation. Once your manuscript has passed the Editorial assessment, the Associate Editor will start to organize peer-review.

All manuscripts submitted to *IR* are screened using CrossCheck powered by iThenticate to identify any plagiarized content. Your study must also meet all ethical requirements as outlined in our Editorial Policies. If the manuscript does not pass any of these checks, we may return it to you for further revisions or decline to consider your study for publication.

9.2. Peer Review

IR operates a single-blind review process, which means that reviewers know the names of authors, but the names of the reviewers are hidden from the authors. The scientific quality of the research described in the manuscript is assessed by a minimum of two independent expert reviewers. The Editor-in-Chief is responsible for the final decision regarding acceptance or rejection of the manuscript.

All information contained in your manuscript and acquired during the review process will be held in the strictest confidence.

9.3. Decisions

Your research will be judged on scientific soundness only, not on its perceived impact as judged by Editors or referees. There are three possible decisions: Accept (your study satisfies all publication criteria), Invitation to Revise (more work is required to satisfy all criteria), and Reject (your study fails to satisfy key criteria and it is highly unlikely that further work can address its shortcomings). All of the following publication criteria must be fulfilled to enable your manuscript to be accepted for publication:

Originality
The study reports original research and conclusions.
Data availability
All data to support the conclusions either have been provided or are otherwise publicly available.
Statistics
All data have been analyzed through appropriate statistical tests and these are clearly defined.
Methods
The methods are described in sufficient detail to be replicated.
Citations
Previous work has been appropriately acknowledged.
Interpretation
The conclusions are a reasonable extension of the results.
Ethics
The study design, data presentation, and writing style comply with our Editorial Policies.

9.4. Revisions

Authors are required to submit the revised manuscript within one week if minor revision is recommended while two point-to-point response to all comments of reviewers and the Editor-in-Chief or the Associate Editor should be supplied along with the revised manuscript to allow quick assessment of your revised manuscript. This document should outline in detail how each of the comments was addressed in the revised manuscript or should provide a rebuttal to the criticism. Manuscripts may or may not be sent to reviewers after revision, dependent on whether the reviewer requested to see the revised version. Apart from in exceptional circumstances, *IR* only supports a round of major revision per manuscript.

10. Contact Us

Journal Contact

Intelligence & Robotics Editorial Office
Suite 1504, Plaza A, Xi'an National Digital Publishing Base,
No. 996 Tiangu 7th Road, Gaoxin District, Xi'an 710077, Shaanxi, China.

Managing Editor

Wen Zhang
editorial@intellrobot.com

Last updated on 6 June, 2023