Supporting Human-Machine Interaction in Ship Collision Avoidance Systems

Huang, Yamin

DOI
10.4233/uuid:18c6ac5f-fed3-470d-a222-fa95ce423037

Publication date
2019

Document Version
Final published version

Citation (APA)
Huang, Y. (2019). *Supporting Human-Machine Interaction in Ship Collision Avoidance Systems* (1 ed.). [Dissertation (TU Delft), Delft University of Technology]. Delft University Publishers. https://doi.org/10.4233/uuid:18c6ac5f-fed3-470d-a222-fa95ce423037

Supporting Human-Machine Interaction in Ship Collision Avoidance Systems

Yamin HUANG

Delft University of Technology

Supporting Human-Machine Interaction in Ship Collision Avoidance Systems

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 18 november 2019 om 12:30 uur

door

Yamin HUANG
Master of Science in Traffic Information Engineering and Control,
Wuhan University of Technology, China,
geboren te Xianyou, Fujian, China.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus,	voorzitter
Prof. dr. ir. P.H.A.J.M. van Gelder	Technische Universiteit Delft, promotor

Onafhankelijke leden:

Prof.dr.ir. M. van Koningsveld	Technische Universiteit Delft
Prof.dr.ir. H.A.P. Blom	Technische Universiteit Delft
Dr. F. Goerlandt	Dalhousie University, Canada
Dr. R. Szłapczyński	Gdańsk University of Technology, Poland
Dr. E. Papadimitriou	Technische Universiteit Delft

Overig lid:

Prof.dr. R.R. Negenborn	Technische Universiteit Delft

This research has been funded by Delft University of Technology (TU Delft) and China Scholarship Council (CSC) under Grant: 201406950010.

Cover illustration by Linying Chen
Published and distributed by: Yamin Huang
E-mail: y.m.huang@outlook.com
ISBN: 978-94-6323-937-0

Printed in the Netherlands

Dedicated to
My fiancée Linying Chen

Preface

I still remember the date on 28th Jan. 2014, three days before Chinese New Year, when I had a skype meeting with Prof. P.H.A.J.M. van Gelder and Mr. Cees Timmers, which completely changes the trajectory of my life. A flight took me to the Netherlands, and I started the fantastic journey in the Netherlands as a Ph.D. student. At the end of this journey, I would like to acknowledge everyone that helped, encouraged, and accompanied me.

Firstly, I would like to thank my promotor Prof. Pieter van Gelder for his consistent patience and professional supervision. Dear Pieter, I enjoyed the freedom I got in setting my research topic and also each long discussion with you. You always encourage me to explore different ideas, which indeed stimulate me to think as an independent researcher. Many sparks are generated during/after our conversations. Your supports are indispensable in this research and my academic career.

I would like to thank Prof. Rudy R. Negenborn, Prof. Henk A.P. Blom, Prof. Mark van Koningsveld, Dr. Floris Goerlandt, Dr. Rafał Szłapczyński, and Dr. Eleonora Papadimitriou for taking place in my Ph.D. committee. Your comments have been very valuable and helpful for improving the manuscript. Dear Rudy, thank you for organizing the RAS event and inviting me to present my work. Dear Henk, thank you for the insightful discussion that inspires my future work. Dear Rafał, thanks for addressing the constructive suggestions. Dear Floris, I am really grateful to know you at the early age of my academic life as a friend and as an advisor. Special thanks to Dr. Max Mendel for working together to develop “risk watch” that inspired me a lot. Special thanks to Prof. Genserik L.L. Reniers for helping me as a qualified journal reviewer.

I am lucky to have a wonderful group of colleagues and friends from the Safety and Security Science Group in TU Delft. Although we have very different backgrounds, knowledge, and research domains, we have a common interest in “safety”. The interesting discussions and debates with you indeed motivated me to reflect my grounds, and help me frame my research

questions. Now, I would like to thank all of you for your creation of this pleasing and culturally inclusive working environment. These smart and valued colleagues are Paul, Genserik, Zahra, Saba, Peter, Yuling, Laobing, Pengfei, Chao, Xin, Jie, Shokoufeh, Paolo, Federica, Behi Nia, Nima, Frank, Simone, Peihui, Eelco, Dick, Eleonora, Karolien, Osiris, Behi Nia, Shenae Lee, Francesca, Valeria, Jos, Maju, Santiago, Rongshui, Weitao, Fuqiang, Dan, Wei, Yunfeng, Kunru, Jianfeng, Yaping, Xuchao, Jerry, and Hang. Moreover, there are so many unforgettable social events with you. We visited many Dutch Museums with a learned guide (Paul Swuste), organized (Chinese-Iranian-Dutch-Brazilian-Indian-Greek) dinners, participated twice Secret Santa (coordinated by Saba, Behi Nia, Shenae, and myself), etc. Additionally, thanks to the secretariat of this group, particularly Astrid and Monique for their support in practicalities.

I also want to thank my friends in Delft. You are not only friends but also reliable comrades. Yang Zhou, Wen Zhang, and Pengfei Chen, I would like to thank you for your accompany these years. It is you make me feel at home in Delft. Yuling and Laobing, thank you for your help during the moment that I needed in these years. Thanks to Zhaolong, Xiao, Jiapeng, and Zhi, we spend so much time to explore the Netherlands, which turns out be my valuable memory of this land. Haiyang, Fanchao, Yuxin, Wenhua, Selma, Dahta, Xavia, and Yaqing thank you for your time for discussion on research and life. It is enjoyable to have talks with you.

Thanks to the China Scholarship Council that funds my research and supports me to meet such great and friendly friends.

I also would like to thank Dr. Tsz Leung Yip for hosting me during my visit to the Hong Kong Polytechnic University. Our cooperation yielded a nice paper.

Thanks to the colleagues from Wuhan University of Technology where is the starting point of my research dream. Thanks to Prof. Yuanqiao Wen who supervised my Master thesis and offered me an enjoyable study environment. Thanks to Prof. Junmin Mou who introduced TU Delft to my life. Thanks to Lichuan Wu, a friend as well as a mentor of mine in WUT, Man Zhu, Lei Du, Yongqing Cai, Zhe Du, and Yimeng Zhang, your research passions continued to infect me.

另外，我想感谢远在祖国的家人对我的关爱和支持。你们无限的理解和支持是我前进的动力。感谢父亲母亲多年的培养与包容，感谢我的姐姐和姐夫对家庭的付出，感谢我所有的亲戚在最艰难的时刻提供无私的帮助。

Thanks to my fiancée Linying Chen. It is my fortune of having one more glance on you twelve years ago. Life is long and boring without your accompanying. You told me you are not good at doing research, while you received the Ph.D. degree a few hours earlier than me. It's time to sail our next waypoint as a vessel train.

Yamin Huang

Delft, October 2019

Contents

List of Figures

List of Tables

List of Symbols

a_u	Weigh of changing surge speed in cost function J_{UO}
a_ψ	Weigh of changing heading in cost function J_{UO}
B	B matrix
CH($\cdot$)	Convex-hull function outputs a convex hull regarding inputted points
$\mathbf{C}(\mathbf{v})$	Coriolis–centripetal matrix
ConfP	Conflict Position
d_{thes}	Threshold distance (or safety distance) setting off alarms
$\vec{d}_{ij}$	Relative distance between ship i and ship j
diag($\cdot$)	A diagonal function outputs a diagonal matrix regarding the inputted vector
D_{cpa}	Distance at CPA
D^*_{cpa}	Distance at CPA considering non-linear trajectory of the ship
$\mathbf{D}(\mathbf{v})$	Damping matrix.
f_1	Coefficient function of system states
f_2	Coefficient function of system inputs
$\mathbf{f}_i$	Dynamics formulation of ship i
$\mathbf{I}^{m\times n}$	Identity n-by-m matrix
J_{UO}	Cost function of choosing collision-free solution in ΔU space
$k(t)$	Time-varying projection factor
K_d	non-negative feedback gain for the derivative term in PID controller
K_p	non-negative feedback gain for the proportional term in PID controller

L_i	Length of ship i
$\mathbf{M}$	Ship's inertia matrix (including added mass)
$n_{\text{collision}}(t)$	Number of velocities leading to collisions at time t
$n(t)$	Number of reachable velocities before collisions at time t
N()	The size of a set.
T_{cpa}	Time to CPA
T^{*}_{cpa}	Time to CPA considering non-linear trajectory of the ship
$TCR(t)$	Time-varying collision risk level at time t
O	Origin
P	Position
$p(\text{collision}\mid\vec{v}_i)$	Probability of collision given $\vec{v}_i$;
$p(\vec{v}_i)$	Probability of choosing velocity $\vec{v}_i$
$p(\text{VO}_k)$	Probability the appearance of the k^{th} VO set
r	Radius
$\boldsymbol{r}_i$	Reference velocity vector
R	Radius of *ConfP*
$\mathbf{R}(\psi)$	Rotation matrix
S_c	Intersection of VO set and RV set
$\bar{S}_c$	Complement of set S_c
$S_{l,r}$	Two sub-sets in RV set
$S_{1,2,3,\ and\ 4}$	Four sub-sets in *V*-space when VO set is introduced
sVO(t)	Sub-VO set at time t
sUO(t)	Sub-UO set at time t
t	Continuous time
t_0	Starting time

t'	Collision-warning time
$T(\bullet)$	Transition function
u	Surge velocity
$u_{min/max}$	Maximal/minimal surge speed
u_{eco}	Economical speed of the ship
$\mathbf{u}^0$	Initial desired velocity
$\mathbf{u}^*$	Desired velocity vector contains u and ψ
U_{ij}^{fea}	Feasible solution set in ΔU space
U_i^{bound}	Kinematic bound of ship i
$\mathrm{UO}(t)$	UO set at time t
UO_{ij}	UO set when ship i encounters with ship j
$\overline{\mathrm{UO}}_{ij}$	The complement of UO set
$\vec{v}$	Velocity vector
VO(t)	VO set at time t
V	Observe matrix
wp	Waypoint
$\hat{\mathrm{w}}$	Repulsive term
$\mathbf{w}$	Nearest point to $\mathbf{u}^0$ on UO set
$\mathbf{x}_i$	State vector of ship i consists of $\boldsymbol{\eta}$ and ψ
$\dot{\mathbf{x}}_i$	Derivative of system-state vector
x	Position in x-axis
y	Position in y-axis
α	Scale of length
ε	Buffer adding to the collision-free solution

$\boldsymbol{\eta}$	Coordinates and heading of the ship
v	Sway speed of the ship
$\mathbf{v}$	Velocity vector of the ship consists of surge speed, sway speed, and yaw rate
τ_v	Sway force of the ship
τ_u	Surge force of the ship
τ_r	Yaw moment of the ship
$\boldsymbol{\tau}$	Force inputs vector contains τ_u, τ_v, and τ_r
$\boldsymbol{\tau}_{\mathbf{lb}}$	Lower bound of forces vector
$\boldsymbol{\tau}_{\mathbf{ub}}$	Upper bound of forces vector
ψ	Heading of the ship
$\psi_{\max}$	Maximal heading
ψ_{ref}	Relative bearing of the waypoints
$\mathbf{0}^{m\times n}$	Zero m-by-n matrix

List of Acronyms and Abbreviations

AI	Artificial Intelligence
APF	Artificial Potential Field
ARPA	Automatic Radar Plotting Aid
ASV	Autonomous Surface Vehicle
BFS	Brute-Force Search
CCA	Cooperative Collision Avoidance
CA	Collision Avoidance
CAS	Collision Avoidance System
CDS	Collision Danger Section
COLREGs	Convention on the International Regulations for Preventing Collisions at Sea
CPA	Closest Point of Approach
DCPA	Distance at CPA
TCPA	Time to CPA
CRI	Collision Risk Index
CTPA	Collision Threat Parameter Area
C-Space	Configuration Space
DD	Decision Disc
DIO	Discrete Inputs Optimization
DR	Danger Region
DR-Vspace	DR in Velocity-space
DR-Wspace	DR in Work-space

DW	Dynamic Window
EMSA	European Maritime Safety Agency
ENC	Electronic Nautical Charts
FCDD	Fuzzy Collision Danger Domain
FMM	Fast Marching Method
GMDSS	Global Maritime Distress and Safety System
GNC	Guidance Navigation Control
HMI	Human-Machine Interaction
HMI-CAS	Human-Machine-Interaction oriented Collision Avoidance System
ICT	Information and Communication Technology
IMO	International Maritime Organization
INS	Integrated Navigation System
KF	Kalman Filter
LCM	Limited Cycle Method
LBS	Lattice-Based Search
MASS	Maritime Autonomous Surface Ship
MMG	Mathematic Model Groups
MPC-CA	Model Predictive Control-CA
NM	Nautical Mile
OOW	Officer On Watch
ORCA	Optimal Reciprocal Collision Avoidance
OS	Own Ship
OZT	Obstacle Zone by Target
PAD	Predicted Area of Dangers
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
POA	Projected Obstacle Area

RI	Risk Indicators
SCR	Spatial Collision Risk
SOLARS	International Convention for the Safety Of Life At Sea
SQ	Sub-Question
TCR	Time-varying Collision Risk
TS	Target Ship
TTC	Time To Collision
VC	Vision Cone
VTS	Vessel Traffic Service
VTSOs	VTS Operators
VO	Velocity Obstacle
EBVO	Ellipse-Based VO
GVO	Generalized VO
LVO	Linear VO
NLVO	Nonlinear VO
PVO	Probabilistic VO
RVO	Reciprocal VO
WR-SD	Warning Rings by Ship Domain

Chapter 1 Introduction

Ship collision is one type of major accidents at sea, which receives numerous concerns from our society. Due to that, preventing collision has been the main task for maritime practitioners and academics. Moreover, developments on Maritime Autonomous Surface Ships (MASS) in recent years stimulate another wave of interest in studying on collision avoidance for unmanned ships. With this background, this dissertation focuses on the theme of proactive collision preventions, aiming at developing collision avoidance systems across manned and unmanned platforms. In this chapter, the background of this research is addressed, followed by motivations, research questions, contributions, and the outline of this dissertation.

1.1 Background

The shipping industry plays a crucial role in our economy, though it usually operates in the shadow of public attention. Over 80% of the world's trade is carried by the shipping industry, making it an integral part of the global economy [1]. Moreover, driven by growing economic activities, the growth of shipping volume is continuously increasing, which is anticipated to reach 3.2% per year [2]. With the development of the shipping industry, it also faces new expectations from the public. Specifically, the public pays more attention to the environmental protection, occupational safety, and transportation efficiency. A safer, smarter, and greener shipping industry becomes a general consensus of all maritime practitioners and researchers.

Various ship casualties, however, are the challenges to these values. Ship collision is one type of accidents which received numerous concerns due to its high frequency and negative impacts. The casualties and incidents data collected by the European Maritime Safety Agency (EMSA)[1]

---

[1] The ship accidents related with EU Member States are collected, specifically, the involved ships flying a flag of one of the EU Member States; the accident occurs within EU Member States' territorial; or involve other substantial interests of EU Member States.

show that "collision" and "contact" (collision with a floating object, fixed object, etc.) contribute to nearly 40% of accidents at sea, see Figure 1.1 [3]. In fact, from 2011 to 2015, the collision (together with contact) is the most frequent accident at sea, according to EMSA's reports [4]. Nevertheless, a collision accident usually accompanies with serious damages to the ships and enormous financial losses. Additionally, it might trigger much severe consequence, such as the sinking of the ship, oil spill, explosion, fatalities, etc. Some accidents that happened in the last three years are briefly demonstrated as examples:

- In Nov. 2016, FLINTERSTAR sunk near the Belgian coast after a collision with LNG carrier AL ORAIQ [5]. The public was concerned about the leakage of gas and oil spill at sea. Fortunately, the LNG carrier is safe and little oil spilled.
- In 2017, two navy ships, USS FITZGERALD and USS JOHN S. MCCAIN, collided with merchant ships and resulted in 17 deaths in total [6, 7].
- In Jan. 2018, the oil tanker SANCHI got fire and explosion after collision with a cargo ship CF CRYSTAL in the East China Sea, which caused the death of 32 crew and the worst oil spill accident in the past 35 years. The ship sunk in the end with a full cargo of 136,000 tonnes [8].
- In Oct. 2018, the ferry ship COTUNAV ULYSSE collided with a containership CSL VIRGINIA, which directly led to 600 tonnes of an oil spill near Cap Corse, France[2].

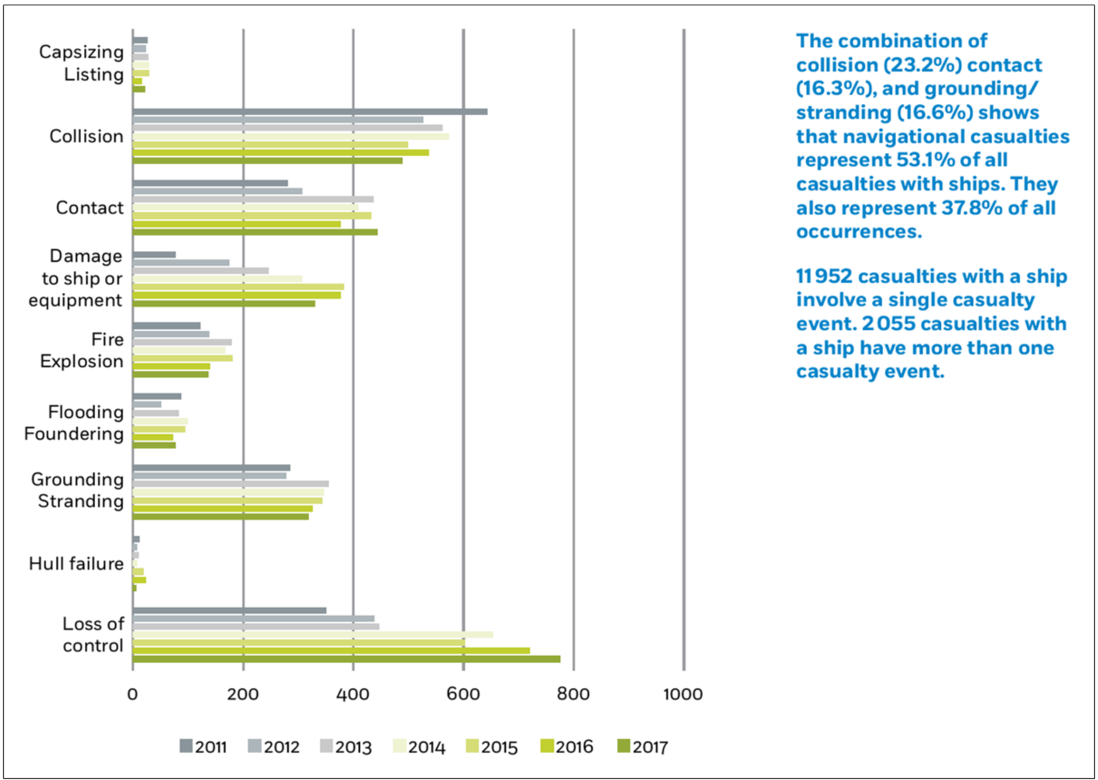


Figure 1.1 The distribution of marine casualties[3] (from [3]).

[2] Information comes from: https://www.thelocal.it/20181008/fuel-cleanup-begins-after-cargo-ships-collide-off-corsica

[3] Definition of marine casualties is from "Resolution MSC.255(84): Casualty Investigation Code". "A marine casualty means an event, or a sequence of events, that has resulted in any of the following which has occurred directly in connection with the operations of a ship: the death of, or serious injury to, a person; the loss of a person from a ship; the loss, presumed loss or abandonment of a ship; material damage to a ship; … …, or the potential for severe damage to the environment, brought about by the damage of a ship or ships. However, a marine casualty does not include a deliberate act or omission, with the intention to cause harm to the safety of a ship, an individual or the environment."

Ship collision-related topics have been a continuing hotspot in the maritime research domain [9]. With numerous studies and investigations, researchers reach a common conclusion that "human error" is the main cause of ship collision [10]. According to the annual report of EMSA, human errors represent 58% of the accidental events. According to Allianz's report, 75% of 15,000 marine liability insurance claims (during 2011-2016) are attributed to human errors as primary factors [11]. Therefore, researchers are keen to find solutions to reduce or to eliminate the human errors.

Automation is one of promising solutions. Especially, autonomous vehicles in other domain approach to reality, and the revolution in automation achieves huge efficiency improvement. People believe that autonomous systems for ships are not impossible and the systems can benefit the safety and efficiency of the shipping industry. Therefore, increasing number of researchers are engaged upon the development of the autonomous systems, aiming at replacing the role of the human on board. A ship with such an autonomous system is named as unmanned surface vehicle or Autonomous Surface Vehicle (ASV).

Figure 1.2 shows the word cloud of the literature in the near 10 years on the topic of ship collision avoidance, which indicates the change of research interests. Specifically, the research supporting human operators (e.g., navigators) is a dominant group of the literature at the early age (i.e., the top-left corner of the figure), while the research of the unmanned ship becomes popular in recent years (i.e., the right part of the figure).

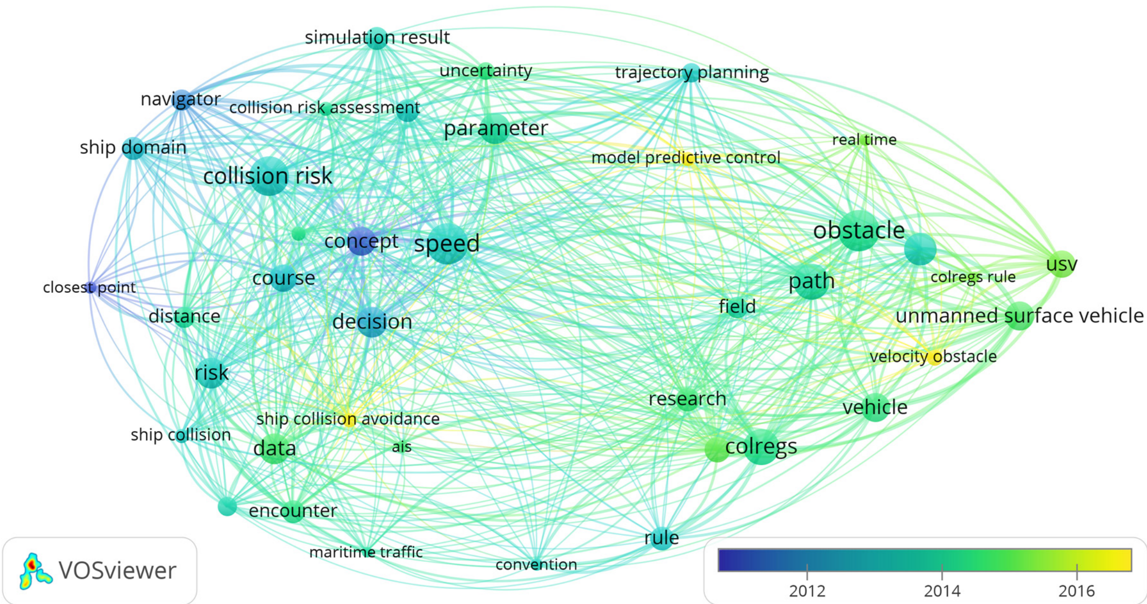


Figure 1.2 Cloud map of words in titles and abstracts of the literature in recent years[4].

1.2 Motivations

With the blooming of new techniques, such as Information and Communication Technology (ICT), Artificial Intelligence (AI), developments of a fully autonomous ship have gained a remarkable amount of discussions. In 2018, the International Maritime Organization (IMO)

[4] Using VOS viewer 1.6.10; the database is presented in Chapter 2; the occurrence of each word is larger than 6 times.

announced four degrees of autonomy of the ship to facilitate the progress of the regulatory scoping exercise. These degrees of autonomy are organized as follows:

Level 1: ship with automated processes and decision support,
Level 2: remotely controlled ship with seafarers on board,
Level 3: remotely controlled ship without seafarers on board, and
Level 4: fully autonomous ship.

From the IMO's categorization, human still plays an important role from Level 1 to Level 3. Consequently, there are still strong demands on navigational assistance to reduce human errors in collision avoidance, specifically, identifying collision dangers, sharing the situational awareness with experts, etc. Moreover, on the road to the autonomy shipping, challenges exist.

At Level 2 and Level 3 stages, the human needs to cooperate with autonomous systems, e.g., monitoring and supervising the systems, taking over control of the systems if necessary, etc. Then, if the opinion of the human and the machine are divergent, how can the human take over control of the machine safely without making situations even worse is one challenge. Moreover, even if autonomous techniques become mature enough in the future, many concerns still exist around the humans' willingness to adopt the techniques. Trust is one of these concerns [12]. Proper designed module of human-machine interaction might help to build the trust in the autonomous systems. Then, design and integration of the human-machine interaction in autonomous systems, that help the human build the trust in the autonomous systems, become another challenge.

At Level 4 stage, it is still impossible that fully autonomous ships replace all conventional ships overnight. Therefore, the human operators working in the conventional ships have to interact with autonomous ships. Under this circumstance, supporting the human operators to cooperate with the autonomous ships during collision avoidance is challenging.

These challenges motivate the research on developing a system that supports situational awareness incorporating human-machine interactions during collision avoidance. Conventional studies either focus on supporting human operators to detect dangers or on automatic collision avoidance. Differently, this dissertation pays more attention to developing the tool supporting various interactions between the human and the machine during collision avoidance. The subsequent influence of the tool on human cognition and social-technical systems are out of the scope of this dissertation.

1.3 Research questions

The research objectives of this study are to develop the collision avoidance system for both manned and unmanned ships and to facilitate human-machine interactions during collision avoidance. To achieve the objectives, the main research question of this dissertation is as follows:

> *How can a Collision Avoidance System (CAS) be designed for both manned and unmanned ships considering Human-Machine Interactions (HMIs)?*

To answer this main research question, the following sub-questions are addressed.

Question on the state-of-the-art:

1. *What techniques have been developed for collision avoidance at sea, and what research gaps can be explored in a generic framework?*

Numerous studies are working on ship collision avoidance, which have developed various types of CASs. However, there is a lack of overviews that present and compare the achievements/gaps of the studies in a generic framework. The answers to these questions show the performance of the existing CASs in the generic framework, revealing limitations and tendencies of the state-of-the-art prevention techniques.

Question on the framework of Human-Machine-Interactions oriented CAS (HMI-CAS):

2. *How can the framework of HMI-CAS be designed to support various modes of human-machine interactions during collision prevention?*

Most of existing CASs are specifically developed for manned ships or unmanned ships, neglecting HMIs in the design of the CASs. In particular, the CAS developed for manned ships only deliveries collision alerts to human operators, while the CAS for unmanned ships directly implements collision-free solutions. The demands on HMIs are not properly considered in these CASs. However, human-machine interactions are crucial for supporting collision avoidance in the different autonomy levels addressed in Section 1.2. The answers to this question address the demands on HMIs and the framework of HMI-CAS.

Question on methodology:

3. *What are the proper methods that support human-machine interactions in conflict detection and conflict resolution of HMI-CAS?*

Supporting human-machine interactions is not simply adding a human-machine interface to the conventional CAS that only delivers information to a human, but requests a new design of the CAS that allows human operators and machines to share knowledge and to cooperate in conflict detection and conflict resolution. Although there are numerous prevention techniques proposed in literature, not all of them can support the various interactions. The answer to this question is to find a group of methods suitable for supporting HMIs during collision avoidance.

Question on improving the conflict resolution module:

4. *How can ship dynamics and navigational regulations be incorporated in HMI-CAS for conflict resolution?*

Many CASs neglect two factors in conflict resolution, namely ship dynamics and regulations [9, 13]. As a result, the solutions suggested by the CAS might not be achievable or not be rule-compliant. In extreme situations, the solutions offered by the CAS could be unsafe for users. To solve these issues, the conflict resolution module of the HMI-CAS needs to be improved. The answer to this question is looking for the method that not only considers the two factors but also can be integrated to the proposed framework of HMI-CAS.

Question on improving the conflict detection module:

5. *How can the conflict detection module in HMI-CAS be improved that supports the human operators take evasive action in time?*

Measurements of risk are foundations of various CASs, which trigger collision avoidance. Most measurement tools assess collision risk in pairwise encounters [14], and the measurement is independent of conflict resolution. Measuring risk in these ways may lead to some problems: firstly, a ship usually encounters with more than one ship, and the influence of other ships on the ship cannot be ignored; secondly, when the measured risk is independent of conflict resolution, the collision-risk level is also independent of the development of collisions, e.g., the collision is avoidable or not. The answer to this question is to introduce the new measure of collision risk that can be applied to the proposed HMI-CAS and supports the human operators to take evasive actions in time.

1.4 Contribution

The contributions of this dissertation are listed below:

(1) Contribution on the state-of-the-art: this dissertation overviews the achievements of existing collision-avoidance studies in a generic framework, which facilitates research peers gaining knowledge of the existing methods and identifies research gaps in improving various autonomy levels of CASs.

(2) Contribution on the development of CASs: this dissertation proposes the HMI-CAS that fulfills the collision avoidance for both manned and unmanned ships and supports human-machine interactions that allow operators to take over the control of unmanned ships safely.

(3) Contribution on the measure of collision risk: this dissertation introduces the novel risk measure that is capable of assessing collision risk in multiple ships scenarios. The proposed measure also can reflect the urgency of the upcoming collision, which are important to facilitate ships avoiding collisions in time.

1.5 Outline of the dissertation

The structure of this dissertation is shown in Figure 1.3.

Chapter 2 presents a generic framework of ship collision avoidance in manned and unmanned ships, where three key modules are focused, namely “motion prediction”, “conflict detection”, and “conflict resolution”. Then, the achievements in each module are collected and discussed in detail. In the end, the research gaps in collision avoidance are highlighted.

In Chapter 3, the demands on HMIs during collision avoidance in different autonomy levels of ships are identified; the framework of HMI-CAS is introduced, which is based on the typical Guidance, Navigation, and Control (GNC) system and integrates a human-machine interface.

In Chapter 4, a family of Velocity Obstacle (VO) algorithms is introduced and applied into the HMI-CAS, which satisfies the requirements of HMIs. Comparisons of the VO algorithms and traditional collision prevention techniques are conducted to show the advantages of applying VO algorithms at sea.

Chapter 5 aims at improving the fundamental HMI-CAS so that the dynamics of ships are considered. Since the horizontal movements of ships have three degrees, i.e., two degrees of linear velocity and one degree of rotation, the interface of the HMI-CAS developed in Chapter

4 is improved. Simulation experiments are carried out to show the performance of the proposed HMI-CAS in various encounter scenarios at sea.

Chapter 6 focuses on improving the conflict detection module of the HMI-CAS. The core of this chapter is constructing the measure of collision risk that incorporates conflict resolution and handles multiple-encounter scenarios. Tests with various encounters in simulators are presented to demonstrate how the proposed measure works.

Chapter 7 summarizes the conclusions of this dissertation, answering the research questions formulated in Section 1.3. Recommendations for future research are provided in the end.

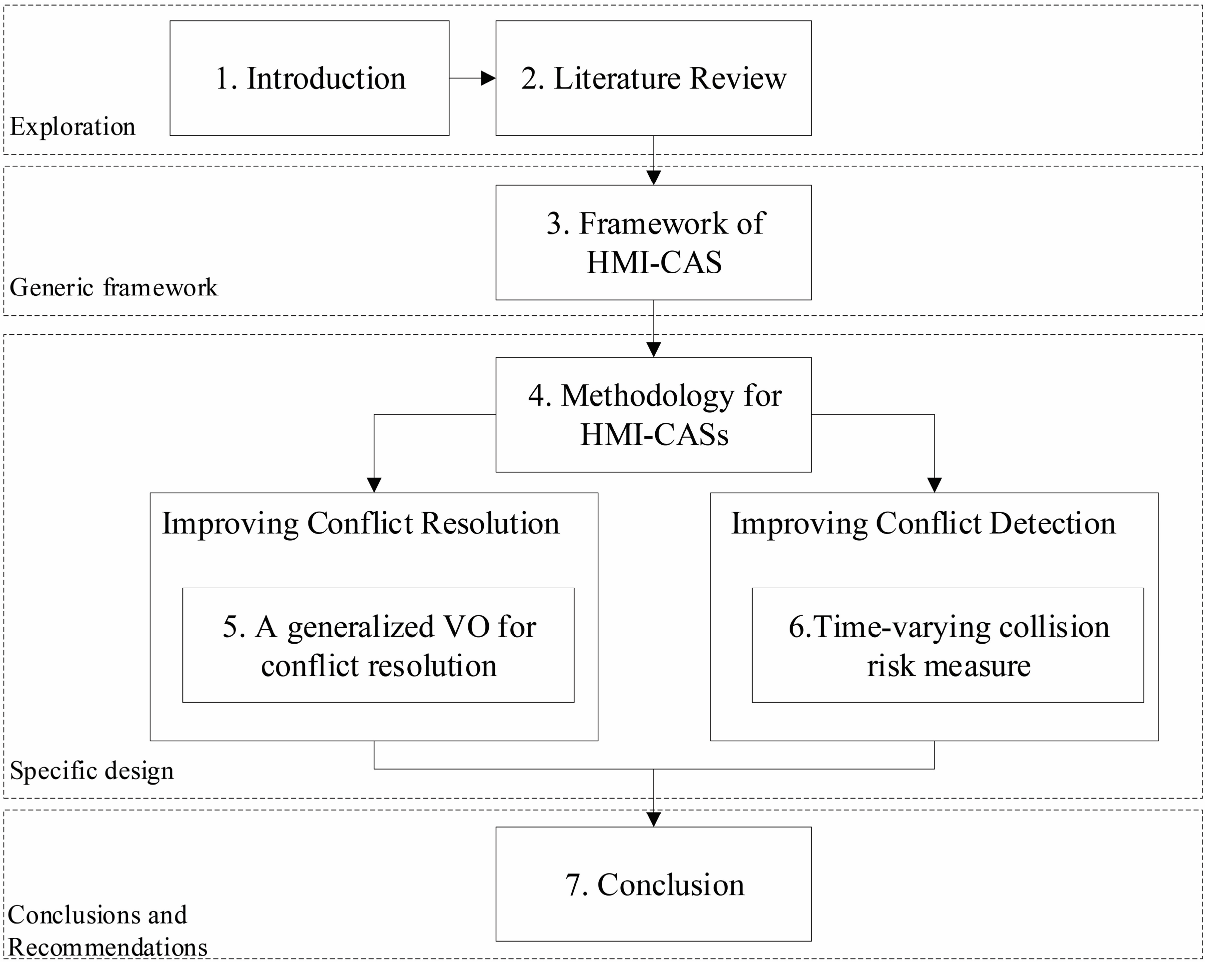


Figure 1.3 Overview of dissertation structure.

Chapter 2 State-of-the-art of Collision Avoidance

Various collision avoidance techniques have been proposed in the literature, either for supporting navigational assistance or automatic collision avoidance. However, there is a lack of a generic framework, which hinders the exchange of knowledge among researchers with different research backgrounds. In this chapter, a generic framework of ship collision avoidance is presented, regardless of manned ships or unmanned ships, and the achievements in each module of the framework are collected and discussed. In the end, a road of transition towards autonomous ship regarding collision avoidance and the role of human-machine interaction on this road are discussed.

This chapter is organized as follows: Section 2.2 introduces the generic framework of various collision prevention methods; Section 2.3, 2.4, and 2.5 conduct comprehensive surveys of motion prediction, conflict detection, and conflict resolution, respectively. Section 2.6 discusses the developments of existing techniques for collision avoidance and the essential role of human-machine interaction in the steps from the manned ship to the unmanned ship. Finally, conclusions are drawn in Section 2.7.

Acknowledgment The content of this chapter is based on the published paper:

Huang, Y., Chen, L., Chen, P., Negenborn, R.R., & van Gelder, P. H. A. J. M. (2020). Ship collision avoidance methods: State-of-the-art. *Safety Science*, 121, 451-473.

Huang, Y., van Gelder, P. H. A. J. M.. A measure of collision risk for triggering evasive actions. (Submitted).

2.1 Introduction

The human factor is believed to be the main cause of ship collision accidents[10], which motivates the developments of techniques handling its negative effect. One group of researchers aims at assisting the human on board to avoid collisions, which is a classical research subject in maritime research from the 1950s [9]. Many methods and navigational equipment were developed and applied in practice, such as Closest Point of Approach (CPA), Automatic Radar Plotting Aid (ARPA), Global Maritime Distress and Safety System (GMDSS), etc., which have contributed to the significantly declining trends of accident frequency [15]. However, the frequency of serious accidents is not decreasing but increasing by 30% after 2000 [16]. That drives the other group of researchers to develop the unmanned ship (or ASV) that avoids collisions automatically. The unmanned ship is expected to remove the human on board and to limit effect of human factor on accident occurrence. In recent years, related studies have gained a remarkable amount of attention. Many companies, organizations, and institutes have announced their plans or prototypes of unmanned ships, e.g., Rolls-Royce, Kongsberg Maritime, etc. Detailed information of prototypes of the unmanned ship is presented in [13, 17].

Studies on manned ships and unmanned ships are developing in parallel with different focuses. The navigational assistance studies focus on supporting situational awareness of OOWs that triggers collision alert in time for them; the other one is more interested in finding a way automatically avoiding collisions. Many scholars believe the studies in two domains may benefit each other [18]. However, there is a lack of a literature review that presents a generic framework of collision avoidance across these two research domains, discusses the achievements and tendency of existing collision avoidance studies, and shows the gaps in the transition towards autonomous era.

Many review articles collected various techniques for collision avoidance, such as [9, 13, 19-21]. However, the state-of-the-art regarding this topic is motivated by three main reasons:

Firstly, these reviews have not pointed out the links between the state-of-the-art methods for the manned and unmanned ship. The reviews are either for supporting the human in collision avoidance [9, 19] or for developing ASVs [13, 20, 21]. The discussion across these two groups of studies is still missing.

Secondly, these studies usually have a wide scope, which only mentions a few techniques specifically related to collision avoidance. For instance, paper [20] and [13] addressed the developments of the ASV in general, while conflict detection and obstacle avoidance were of less focus. Paper [21] focused on the techniques used in path planning and only included a few studies related to reacting collision avoidance for unmanned ships. Paper [19] described the collision risk assessment, but it neglected the techniques for conflict resolution.

Thirdly, as the quantity of related literature increasing dramatically, an update is needed for the peer-researchers' convenience. The review paper [9] concluded the shortcomings of ship collision avoidance in early age, in particular from the 1950s to early 2000s. The concluded limitations are widely accepted and used in recent articles and literature review. These limitations are list as follow:

(1) Environmental factors are ignored; [9, 21]
(2) Regulations are usually out of consideration, e.g., International Regulations for Preventing Collisions at Sea (COLREGs); [9, 13, 20, 21]

(3) The moving obstacle is assumed to be static or semi-dynamic[5] [9, 13];
(4) The ship is assumed to have a highly ideal maneuverability [9, 21];
(5) The discussion on the balance of efficiency and effectiveness is missing [13].

As new methods and techniques are emerging, some limitations have been improved and changed, which are rarely discussed in existing review articles.

This chapter aims at collecting developments of techniques used for ship collision prevention either for manned ships or unmanned ships in a generic framework, providing a comparative evaluation and overview on these techniques, and highlighting the demands of considering human-machine interaction in the transition from conventional shipping towards autonomous shipping. Compared with existing reviews, the main contributions of this chapter are:

(1) The knowledge of ship collision avoidance techniques is updated with detailed comparisons of the strengths and weakness of methods in a generic framework containing three processes, namely motion prediction, conflict detection, and conflict resolution.
(2) A potential road of transition from the manned ship towards the unmanned ship is presented based on the generic collision avoidance framework, and the role of human-machine interactions serving for the transition is shown.

2.2 Structure of the state-of-the-art

2.2.1 Scope of review

Collision avoidance has many different interpretations from different perspectives. In some studies, the collision avoidance refers to find a collision-free path/trajectory that connects the origin and the destination in a given the map, such as [22], [23], etc., which are also noted as route planning problem or path planning problem. In other studies, the meaning of collision avoidance is slightly different, which refers to the ship departing from its current path for avoiding the approaching dangers, e.g., [24, 25]. This kind of collision avoidance is also named as reactive collision avoidance. To eliminate the ambiguity, the definition of collision avoidance is re-defined for both manned and unmanned ships in this dissertation as follows:

> **Definition:** Collision Avoidance (CA) is a process that one ship (no matter manned or unmanned) deviates from its current route to avoid a potential undesired physical contact at a certain time in the future.

The ship under our control is called Own-Ship (OS). Obstacles include stationary obstacles and moving obstacles (or Target-Ship, TS).

The scope of this review narrows down to reactive collision avoidance for both manned and unmanned ships. Specifically, two types of research are collected: 1) prevention techniques for manned ships, which support the OOW on board, e.g., collision warning and searching evasive

[5] Semi-dynamic obstacles refer to the obstacle moves without changes on headings and speed.

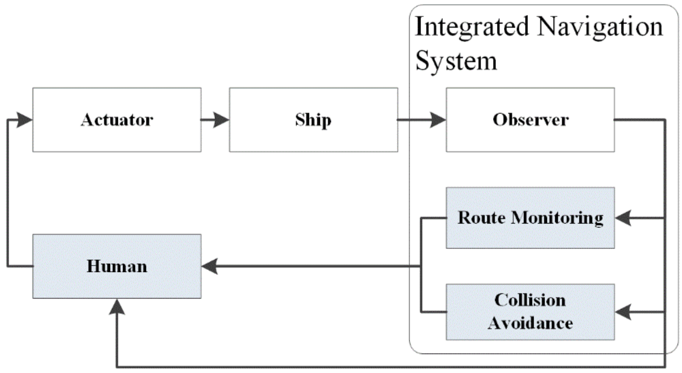
actions; and 2) methods applied in ASVs that drive the vehicle to deviate from the predefined path for collision avoidance.

2.2.2 Generic framework of ship collision avoidance

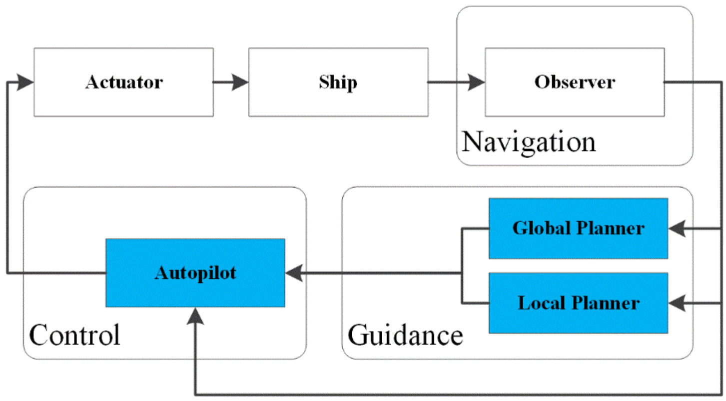
According to this definition, the collision prevention problem contains two sub-problems: "*conflict detection*" and "*conflict resolution*". Solving the "conflict detection" problem is to determine whether the ship is in danger and when to take evasive actions. Solving the "conflict resolution" problem is to answer the question of what actions should be taken to prevent collision [26].

In manned ships, modern bridge systems, such as Integrated Navigation Systems (INS), are designed to support collision avoidance mainly during conflict detection stage, which consist of various observers (i.e., sensors) and instruments [27]. Its main function is to offer information to navigators and to send an alarm if necessary. Human, who decides whether to take actions, plays a major role in conflict resolution.

In ASVs, a GNC system takes the whole responsibility for collision prevention, which consists of "Guidance" system, "Navigation" system, and "Control" system [28]. The "Guidance" system is engaged to detect and to solve the conflict at the same time, which decides *When* and *How* to take evasive actions. The other two sub-systems offer information to support the guidance system and implement the planned actions. The data/information flows in a manned ship and an unmanned ship during collision avoidance are separately presented in Figure 2.1.



(1) The decision process in a manned ship.



(2) The decision process in an unmanned ship.

Figure 2.1 Representations of navigation systems in manned and unmanned ships.

Based on Figure 2.1, one can see that either for the navigation system in manned ships or in unmanned ships, some essential modules are needed to reach a collision-free solution for both manned and unmanned ships. When the ship observes the positions of obstacles (such as TSs) at present, it estimates the possible positions of these obstacles in the future and their corresponding collision risks. Based on these estimations, the OS (the OOW or the ASV) decides to keep its current route or to find a new collision-free solution.

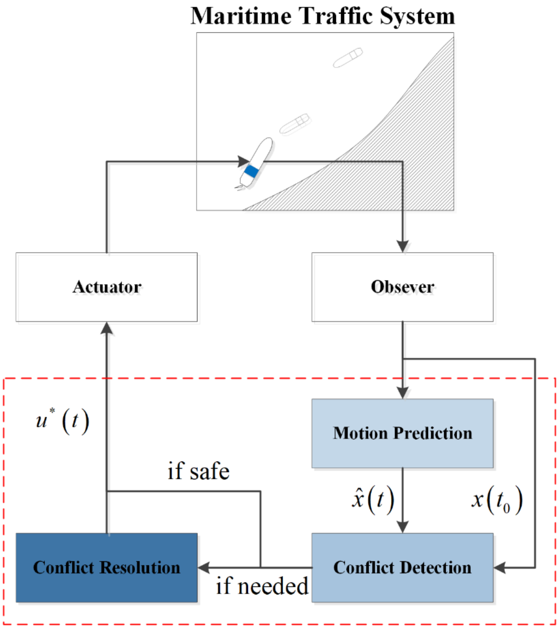


Figure 2.2 Generic representation of ship collision avoidance process.

The information flows of collision prevention in the manned ship and the unmanned ship can be abstracted in Figure 2.2. Five basic components are included: (1) “Observer”, which contains various sensors offering data to support other modules; (2) “Motion Prediction” module, which estimates the future trajectories of the OS and the obstacles; (3) “Conflict Detection” module, which checks collision risk and launches collision warning if necessary; (4) “Conflict Resolution” module, which determines the evasive solutions and then, (5) “Actuator”, which implements the solutions via rudder and propeller.

The “Motion Prediction”, “Conflict Detection” and “Conflict Resolution” are the main focuses of this chapter, which are investigated in Section 2.3-2.5. In particular, the following questions are discussed:

(1) What methods are used to predict the trajectory of ships and obstacles? (Section 2.3)
(2) How is the collision risk assessed and served for early alarm? and (Section 2.4)
(3) What algorithms are used to determine evasive actions? (Section 2.5)

Other modules, such as “Observer” and “Actuator”, are also necessary for collision prevention, but they are not included in the scope of this review. In this review, it presumes that the observer can offer accurate information about the states of the system; the actuator can execute the collision-free solutions. For readers who are interested in the developments on observers and actuators can read more in [13].

Therefore, the articles collected in this chapter mainly contains the topics associated with the three sub-modules: motion prediction, conflict detection, and conflict resolution. Although each module covers numerous studies, the studies applied to avoid collisions are found with three steps:

Firstly, databases of “Web of Knowledge” and “Scopus” are used to collect journal and conference papers with the following keywords in title, keywords, and abstract: “ship”, “vessel”, “unmanned surface vehicle”, “USV”, “autonomous surface vehicle”, and “ASV”, “collision avoidance”, “collision prevention”, “avoid collision”, “prevent collision”, “navigation safety”. The research with a series of keywords which indicate that it is out of our scope is excluded, such as “underwater”, “aircraft”, “car”, “collision protection”, “estimation of collision damage”, “ship-bridge collision”, “ship-iceberg”, etc. The searching result is narrowed down by limiting the language to “English”, and research domain to “engineering”. At this step, 304 pieces of record are obtained until Mar. 1st, 2019.

A further literature filtering is performed to identify the studies that are not completely fitting the scopes. According to the scope described in Section 2.1, some records are removed, e.g., the studies relating with sharing navigation experience, the studies only considering path planning or formation control, the studies focusing on the construction of ship domain, the studies that do not consider moving obstacles. In the end, 90 pieces of records are obtained.

After reading the selected papers, some papers are added as the complement to the database. Three types of studies are added: the papers that are cited in the 90 papers but not included in the database; the studies which were published before 2000 but are classical and are sources of some methods; the papers published in 2019 but have not appeared in the database.

2.3 Motion prediction

Motion prediction is a fundamental module for collision avoidance, which contains a process that predicts the trajectories of the OS and obstacles. When the OS encounters with potential dangers, the predicted trajectories are used to determine the collision risk for conflict detection. Moreover, when the OS determines a resolution, the predicted trajectories are also needed for checking collisions.

In this section, some popular motion models used in trajectory prediction are presented, followed by existing techniques used in trajectory predictions. A summary of the developments and challenges are discussed in the end.

2.3.1 Ship motion models in prediction

The predictions usually rely on the mathematical expression of the system, i.e., motion models of the ship. Since the ship moves in a horizontal plane, the workspace of the ship in collision avoidance studies is also the horizontal space, i.e., $W = \mathrm{R}^2$. The configuration space (C-space) of the ship consists of position and orientation, i.e., $C = \mathrm{R}^2 \times \mathbb{S}^1$.

According to the constraints used in modeling, the motion models are categorized as holonomic models (constraints on configurations only) and non-holonomic models. Non-holonomic models contain kinematic models, dynamics models, and simplified dynamics models. Brief information of these models is shown in summary form in Table 2.1.

Table 2.1 Overview of ship motion models

	Holonomic model	**Kinematic model**	**Dynamics model**	**Simplified dynamics model**
Eq.	$\begin{cases}\dot{x}=u_x\\ \dot{y}=u_y\end{cases}$ or $\begin{cases}\dot{x}(t)=u\cos\psi\\ \dot{y}(t)=u\sin\psi\end{cases}$	$\begin{cases}\dot{x}=u\cdot\cos\psi\\ \dot{y}=u\cdot\sin\psi\\ \dot{u}=a_t\\ \dot{\psi}=a_n/u\end{cases}$	$\dot{\eta}=R(\psi)v$ and one of the following models: (1) Vectorial representation: $M\dot{v}+C(v)\dot{v}+D(v)v+g(\eta)=\tau+w(t)$ (2) MMG: $\left.\begin{array}{r}(m+m_x)-(m+m_y)v_m r-x_g m r^2=X\\ (m+m_y)\dot{v}+(m+m_x)ur+x_g m\dot{r}=Y\\ (I_{zG}+x_g^2 m+J_z)\dot{r}+x_g m(\dot{v}+ur)=N\end{array}\right\}$	$1^{st}/2^{nd}$ order response model (KT Eq.) Successively linearization Meta model
Pro	Simple	Simple Reflect some feature of ship motion	Accurate (based on existing techniques)	Relatively simple and accurate
Con	Unrealistic	Inaccurate	Complicate Uncertainty on parameters Form (1) ignores high order variables Form (2) requires a better understanding of ship hull, rudder, and propeller	Less accurate (when initial conditions are not satisfying)
Exa.	[29-35]	[36-38]	Form (1): [39-43]; Form (2): [44-46].	Meth. 1: [47, 48]; Meth. 2: [49, 50]; Meth. 3: [51, 52].

Note: "Eq.": Equation; "Meth.": Method. "Exa.": example.

Holonomic model is the simplest way to describe the ship's motion, which is based on the assumption that the ship is a holonomic vehicle which moves freely in a horizontal plane. In the trajectory prediction of the TSs, these equations (shown in Table 2.1) are also called "constant velocity" model [53] that are widely used.

Kinematic model is proposed to overcome the limitations of the holonomic model, which ignores the force that causes movements. A standard form of kinematic models is shown in table 2.1, which comprises various kinematic models. Two popular kinematic models used in the maritime are "*dubins car*" model (e.g., [36, 37]) and "*simple car*" model ([38]).

Dynamics model is introduced to handle the impact of the ship's mass on ship motion that is usually neglected in the above models. Specifically, researchers incorporated the relation between the applied forces and the resulting movements to increase the accuracy of prediction. Two forms have been widely used in the literature. **Form 1:** *Vectorial representation for marine vehicle.* The dynamics model is described in a compact vectorial setting, which contains two formulations: one describes the kinematic relations; the other shows the kinetic equations using forces as inputs [54], see table 2.1. This form is widely used in designing controllers and observers in ASVs [13]. **Form 2:** *Mathematic Model Groups (MMG),* which is widely used in maneuverability prediction. The MMG employs rudder angle and propeller revolutions as the inputs and considers the specifications of rudders and propellers. This model is usually used in the theoretical analysis of ship maneuverability. Details of MMG refers to [55].

Simplified dynamics models are popular among researchers even though they are less precise than preview dynamics models [53]. Three simplification methods can be found in the literature. **Method 1:** one simplification technique is to ignore some less important terms in the aforementioned models (more details in [13]). A popular way is using $1^{st}/2^{nd}$-order response equation to describe the dynamics of rotations and assumes the surge speed is constant and a zero sway speed, e.g., [47, 48]. **Method 2:** another frequently used simplification technique is called successive linearization, which is based on Taylor expansion. Researchers linearized the ship motion model around an estimated trajectory. As a result, the motion model has a relatively simple form (a linear form), and the predicted trajectory approximates the real trajectory. As the real input deviates from the initial setting, the errors of prediction might increase. **Method 3:** Instead of simplifying ship dynamics based on formulations, the other simplifications are based on simulation/experiment data. Specifically, researchers either use a simulator to generate the responses of the ship with different inputs or collect the experimental data of the ship's response with different inputs. Then they use regression methods to find equations that fit the data best.

2.3.2 Prediction of trajectory

Prediction of the OS's trajectory

In an ideal case, i.e., the control inputs and motion models of the OS are known, the prediction of the OS's trajectory turns to be solving the ordinary differential equations in Section 2.3.1. This idea is popular in the studies for ASV. The simplest way is to assume that the ship is a holonomic vehicle, which is popular in many collision prevention studies, e.g., [29, 34, 35]. However, the errors between the predicted trajectory and the real trajectory are huge due to this unrealistic assumption. Thus, many researchers employ either the dynamics model [49] or simplified dynamics model [47] in trajectory prediction. Due to the complicated form of the equations, the analytical solutions are usually infeasible, and a numerical method is usually needed, e.g., Runge-Kutta methods [56], etc.

In other cases, researchers are faced with more practical problems, such as uncertainties on motion models and parameters. Then, some parameter identifications techniques [57] are needed to obtain the motion model. Moreover, a challenging issue is considering noise and errors in predictions. In this case, studies usually apply Kalman Filter and its variations (e.g., extended KF) in trajectory prediction with a relatively simple model, e.g., "*simple car*" model [38].

Prediction of the TS's trajectory

Since the information of the TS is insufficient for the OS, e.g., parameters of motion model, inputs to the system, etc., the prediction of the TS is more challenging than that of the OS. Due to these uncertainties, researchers usually prefer to use simple models, such as the holonomic model and kinematic model [53]. The simplest way to predict the trajectory of the TS is based on assumptions that the TS keeps its velocity and environmental disturbance is neglect. It is widely used but less accurate for collision avoidance. A more reasonable approach is considering the uncertainties of models, controls, and disturbance. The methods to predict the trajectory of the TS can be categorized into three modes according to the knowledge of the TS.

Mode 1: *Physics-based methods* predict the motion of the ship only depending on the laws of physics, while the existing studies either ignore the control inputs or treat the maneuvers as white noise. Kalman Filter (KF) is a preferred technique used to consider these noises and give the best guess of the ship's trajectory in many studies. Together with the KF, holonomic models [58] or kinematic models (e.g., "*simple car*" model [59]) are employed. To handle the nonlinearities and uncertainties of these motion models, the variations of the KF are used, e.g., extended KF [38], Particle Filter, Interacting-Multiple-Model Kalman filter, probabilistic filter [25], etc. Although these methods can predict the trajectory of the ship in a short period, they cannot predict the changes in trajectory due to the changes of maneuvers [60].

Mode 2: *Manoeuvre-based methods* take the maneuvers of the ship into account, i.e., navigational intention, which is learned/estimated from historical traffic data or by the protocols for ship encounter situations, e.g., navigational regulations. Algorithms learn the behavioral patterns of ships in a certain area from massive traffic data and then use these patterns to support the prediction [61]. Some popular learning models are neural network [62], Gaussian process [63], Hidden Markov Model [64], etc. More details of these models are addressed in [19].

Mode 3: *Interaction-aware methods* consider the interactions between ships in prediction. Specifically, communications between ships are usually included. . The OS would broadcast [47], exchange, or negotiate their maneuver intentions (e.g., intended course [65-67]) or the planned trajectory [68, 69] with the TS. When the ship shares its planned trajectory that is estimated by the ship itself, the predicted trajectory would be more accurate than that predicted by other ships since the ship has a better knowledge about its dynamics and intentions. When the ship only shares its intention instead of its trajectory, the other ship still can reduce the uncertainties in prediction. In return, the predicted trajectories would still be better than those predicted by other methods.

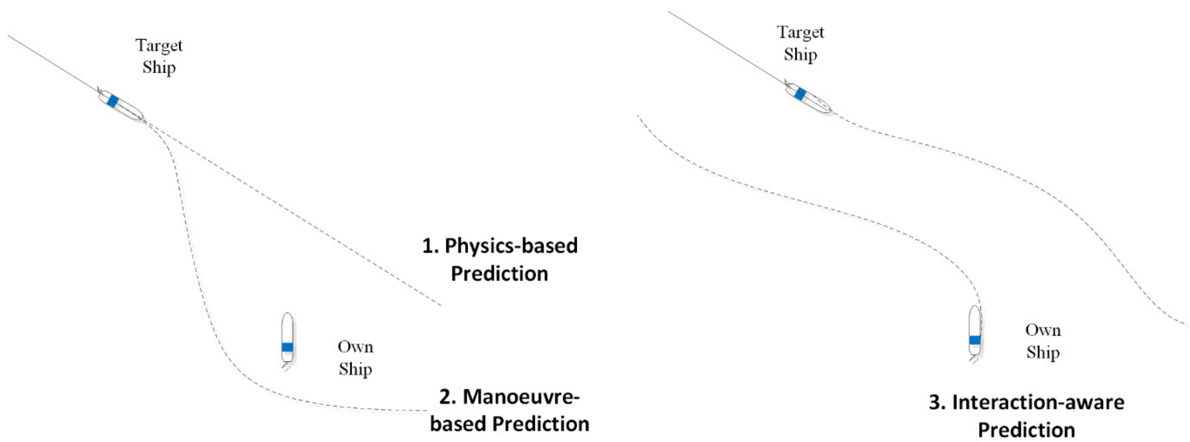


Figure 2.3 Illustration of different predictions modes.

In these three modes, the asymmetry of information is eliminated gradually. An illustration of different prediction methods is shown in Figure 2.3. The physics-based prediction provides the trajectory according to its historical data only. Since the TS is sailing to the southeast with constant speed and course, the physics-based method suggests that the TS would continue this movement in the future. The manoeuver-based method, however, first recognizes the pattern of the TS, e.g., the give-way intention, and predicts that the TS would perform a starboard turn according to navigational regulations, i.e., COLREGs. Then, the trajectory is predicted based on the recognized pattern. Different from previous methods, the Interaction-aware method requires the exchange of information among ships, e.g., the OS and the TS broadcast their predicted trajectories. Thus, the predicted trajectory is provided by each ship itself.

2.3.3 Summary of motion prediction techniques

In previous review papers, researchers concluded that the studies of collision avoidance are suffering from ignoring dynamics of the OS and the TS, semi-dynamic obstacles moving with constant speed and course, and no environmental disturbance [9, 13]. The physical-based method is widely used in those studies. In recent years, many researchers engaged in overcoming these drawbacks. Some achievements are listed as follows:

Firstly, researchers have applied various non-holonomic models in the prediction of the OS's trajectory, such as kinematic models, dynamics models, and simplified dynamics models. Given control inputs, the trajectory of the OS is usually calculated by the Runge-Kutta method. In some studies, environmental disturbances are considered, where the distribution of disturbance is assumed to be known, e.g., [50]. Then, the predicted trajectory of the OS is bounded in a tube or a funnel.

Secondly, the developments of the prediction of the TS are currently towards the usages of the maneuver-based methods and interaction-aware methods that are capable to incorporate more information in the prediction. Manoeuver-based methods estimate the steering intentions first and then predict the trajectory, e.g., [70]. However, the errors of the estimated intention could not be completely eliminated by existing methods, and collision avoidance is sensitive to these errors. For instance, when two ships encounter in a close range, any misestimation would result in a collision. Therefore, some researchers preferred to use the interaction-aware method to predict the TS's trajectory, which allows cooperation among ships. A simple way is exchanging their intentions among ships, such as intended course [65] or turning points [47]. Alternatively, each ship can broadcast its own predictive trajectory, such as [49, 56], which is more accurate than the above methods since the ship has better knowledge about its own dynamics. However, in this way, communication burdens are increased.

By these new changes, many collision avoidance techniques do not hold the assumption that the TS sails with constant speed and heading, see more in Section 2.5. However, some problems remain, e.g., the uncertainty of the motion model, limited knowledge about environmental disturbance acting on the ship, performance of communications, etc. These uncertainties should be analyzed and bounded for the subsequent collision avoidance, where more studies are needed.

2.4 Conflict detection

Conflict detection involves following sub-problems: *who* have conflicts with the OS, *whether* and *when* evasive actions should be taken by human or machine. These questions are strongly related to collision risk. For instance, a target ship which has a high likelihood to strike the OS is a potential danger. When the risk is higher than a predefined threshold, a collision alert is launched for reminding the OS for precaution. Before the risk reaches its upper bound that the collision becomes unavoidable, the evasive action should be taken.

Measuring the true collision risk during an encounter, however, is a challenging work due to uncertainties. These uncertainties include but are not limited to: the intentions of the TSs, the performance of machines (the OS and the TSs), the environmental conditions, etc. Since there are lack of knowledge of all these parameters during encounters, estimating the collision risk considering all the uncertainties is difficult.

To deal with the uncertainties in risk measures, two lines of thinking are widely accepted, namely "**expert-based methods**" and "**model-based methods**". One group of studies directly utilizes experienced experts to assess collision risk who work with those uncertainties in practice. The measured risk reflects the belief of experts about a possible collision event. The second group of methods assesses the probability of a collision event based on a simplified model that describes the physical process of collision. Consequently, the measured risk is a conditional probability of collision.

The measurements of collision risk serve for conflict detection in following ways: identify potential collisions for the OOWs [71, 72]; launch an alarm for taking evasive actions [51]; trigger the autonomous system to find evasive actions [73]; evaluate the risk of alternative paths or evasive actions [59, 74].

In the following sections, the existing measures of collision risk are divided into two main categories, namely expert-based methods and model-based methods, followed by an overview of these two categories. Besides, the issues how do existing methods serve for conflict detection and how do they perform in manned ships and unmanned ship are discussed. In the end, a summary is given.

2.4.1 Expert-based methods

Collision Risk Index (CRI)

One category of methods set off a collision alarm based on a numerical value called Collision Risk Index (CRI). A popular way is combining several indicators into one number, i.e., CRI. The selected indicators, models, and thresholds are usually depending on experts' knowledge. A collision alarm is launched when the CRI violates a pre-set threshold. The construction of this index is strongly tied to experts, e.g., captains, pilots, etc. Thus, the meaning of collision alarm usually can be interpreted as *the situation when most of the experts believe the ship is in danger now*.

Table 2.2 Overview of Collision Risk Index methods

Formulations	**Brief descriptions**	**Ref.**
$\mathrm{CRI}_1 = w_1 f\left(DCPA\right) + w_2 f\left(TCPA\right)$	DCPA and TCPA are combined in a linear equation.	[75-77]
$\mathrm{CRI}_2 = \sum_i w_i f_i\left(RI_i\right) \Big/ \sum_i w_i$	DCPA, TCPA, and other factors are combined in a linear equation.	[18, 71, 78-83]
$\mathrm{CRI}_3 = \left[w_1 \left(\frac{DCPA}{d_s} \right)^2 + w_2 \left(\frac{TCPA}{T_s} \right)^2 + w_3 \left(\frac{d_{ij}}{d_s} \right)^2 \right]^{-\frac{1}{2}}$	DCPA, TCPA, and relative distance are used. The Euclidean distance as a measure of collision risk.	[84, 85]
$\mathrm{CRI}_4 = r_{basic} e^{-TCPA/10} e^{-\lvert DCPA \rvert} \cdot \mathrm{F}_{angle}$	DCPA, TCPA, angles, and frequency of collision event are combined nonlinearly.	[85, 86]
$\mathrm{CRI}_5 = \cos\left(v_{ij}, d_{ij}\right) = \frac{v_{ij} d_{ij}}{\lVert d_{ij} \rVert \lVert v_{ij} \rVert}$	The angle between v_{ij} and d_{ij} is used to describe the trend of relative movement.	[87, 88]
$\mathrm{CRI}_6 = k d_{ij}^{-1} v_{ij} \left(m \sin\left(\theta_{ij}\right) + n \sin\left(2\theta_{ij}\right) \right)$	v_{ij}, d_{ij}, and encounter angle are combined using supervised learning methods.	[89, 90]
Other forms (risk table)	Using a risk table to determine CRI.	[91-93]

A number of CRI methods have been proposed and a brief summary is presented in Table 2.2. In these risk measures, two popular indicators are Distance at Closest Point of Approach (DCPA) and Time to CPA (TCPA).

One group of studies synthesized two CPA indicators into a CRI with the help of fuzzy theory [76] or Probit model [18], noted as CRI_1. Some studies considered more Risk Indicators (RIs), e.g., relative distance [78], relative bearing [79], a variation of relative bearing [80], the ratio of speeds [81], ship domain [82], etc. Correspondingly, more techniques are employed to gain knowledge from experts, e.g., multilayer perceptron [82], evidential reasoning [79], support vector machine [81], Dempster-Shafer evidence theory [78], etc. A standard form is shown in table 2.2 and noted as CRI_2.

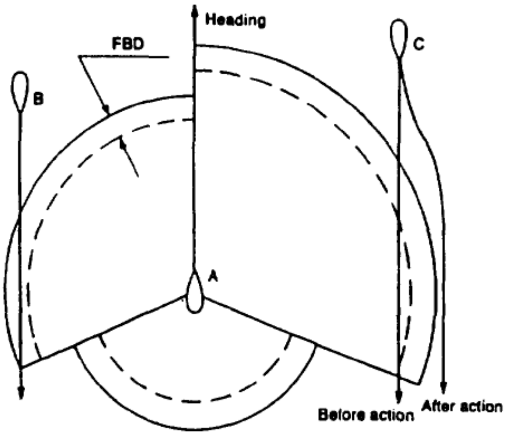
Instead of using linear formulations, some researchers proposed to use a nonlinear equation to calculate the risk. Some of them used a Euclidean norm to describe the collision risk, which is CRI_3 in table 2.2 ([84, 85]); some researchers incorporated the collision rate that is the historical occurance of collisions in certain waters in the risk measure (see CRI_4 in Table 2.2, [85]). Rather than using DCPA and TCPA as RIs, some researchers assigned the relative distance and relative velocity as fundamental indicators, e.g., CRI_5 and CRI_6. CRI_5 uses the tendency of relative motion to assess the risk (e.g., [87, 88]); CRI_6 selects three key indicators first and determine the coefficients by parameter identification methods [89, 90].

There is another group of methods which does not use a formula to calculate the risk level but a pre-set risk table. Given DCPA and TCPA levels, the CRI is determined by searching in the table that is directly determined by experts, e.g., [91-93].

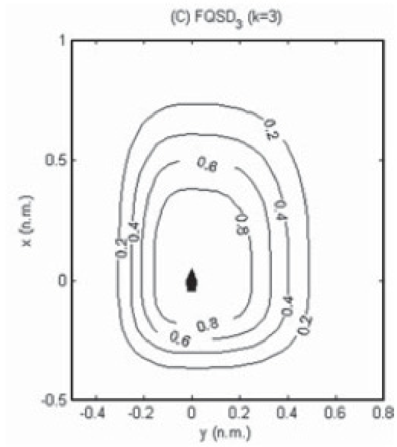
Warning Rings by Ship Domain (WR-SD)

Another category of methods usually visualizes collision risk by a set of warning rings surrounding the OS. When a TS enters/will enter the warning ring of the OS or the OS enters/will enter that of the TS, an alarm is triggered. The warning ring is usually related to a concept called "ship domain" that is a region surrounding one ship that the OOW prefers to keep it clear [94]. Ship domain reflects experts' belief about the minimum safety region. More details about ship domain can be found in [95].

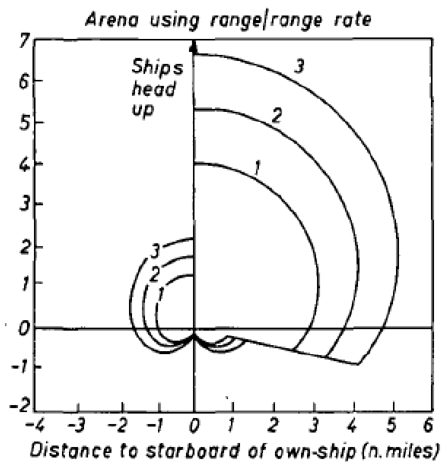
Most studies do not directly use ship domain to detect dangers. One reason is that if one ship used ship domain as the ring of warning, the alarm might be too late for the ship to keep its ship domain clear [96]. Thus, researchers usually adopt the following three ways to detect dangers based on ship domain, see Figure 2.4.



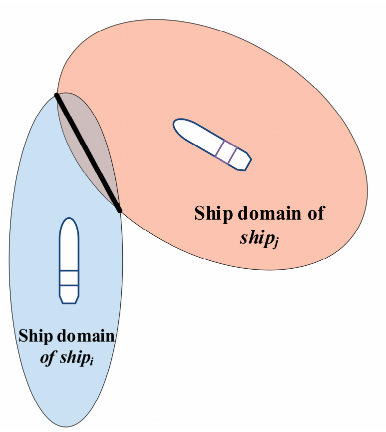
(1) WR-SD1: Fuzzy ship domain [97]



(2) WR-SD2: Spatial collision risk [98]



(3) WR-SD2: Ship arena [99]



(4) WR-SD3: domains Overlapping [72]

Figure 2.4 Illustration of warning rings by ship domain (WR-SD) methods.

Type I: Trajectory of the ship w.r.t. ship domain (**WR-SD1**). Collision danger is determined by comparing ship domain and predicted trajectory. When the predicted trajectory of the TS crosses the OS's domain or the predicted trajectory of the OS violates the TS's domain, a collision alarm is triggered. When the boundary of the domain is determined and clear, risk is a binary number (0/1). When the fuzzy boundary is accepted, such as [97, 100], risk is estimated by the degree of the violation of the domain.

Type II: Position of the ship w.r.t. ship domain (**WR-SD2**). Collision alarm is triggered by comparing the position of the TS with the expanded ship domain of the OS, see [101]. The expanded domain contains various contour lines showing various risk levels. Collision risk of the TS is determined by the position of the TS in the contour map.

Type III: Overlapping of ship domain (**WR-SD3**). Collision alarm occurs when the ship domain of the OS and the TS are overlapping [102] or/and satisfying certain conditions [72]. The risk level is estimated by the overlapping of the ship domains.

The similarities of the three types of studies are: (1) these studies are all relying on the determination of ship domain; (2) they are visualized in a map for the OOW to understand the risk level; (3) the boundary of ship domain is modified to represent different risk levels, where experts' knowledge is usually employed.

2.4.2 Model-based methods

Instead of using experts, the other category assesses collision risk based on a given scenario in which the uncertainties are known or eliminated. In return, the risk measure can address some truths of given scenarios to the OOW/ASV. These truths refer to collision happen/not happen or the conditional probability of collision in given scenarios. The violation of risk threshold means "*provided the scenario is true, the probability of collision is unacceptably high*".

Binary collision criteria

Binary collision criteria usually offer a deterministic result to users about collision event, i.e., happen/not happen, based on a given scenario. CPA is a popular criterion, which is based on the scenario that the TS keeps its velocity constantly, and the shape of the ships is seen as a circle. In this scenario, if the Distance at CPA is smaller than the sum of two ships' radius and Time to CPA is positive, the collision is going to happen; otherwise, it not. Although this scenario is highly unrealistic, it is widely used for manned- and unmanned ships [59]. Besides, it is also an important RI in CRI measures [91] (see Section 2.4.1).

Probability of collision (P_{coll})

Since the process of collision is not deterministic, it is natural to use probability to describe the collision risk. In particular, many uncertainties might influence the result of dangerous encounters, e.g., sensors errors, environmental disturbances, reactions of the TS, errors in prediction, etc. However, considering all of these uncertainties is challenging, especially because some of them are difficult to estimate. Alternatively, researchers only consider one/part of uncertainties of which the probability distributions are known. Hence, the probability of collision with given uncertainties can be calculated. In [59], Monte-Carlo simulation is employed to assess the probability. In [103], a concept of probability flow is employed to calculate the probability of collision which can save some computational time, which is fast and approaches to the result obtained by Monte-Carlo simulations, see Figure 2.5. The uncertainties considered in these studies mainly come from sensors and environment disturbance [59, 103]. However, how to set a threshold to launch collision alarm is not mentioned.

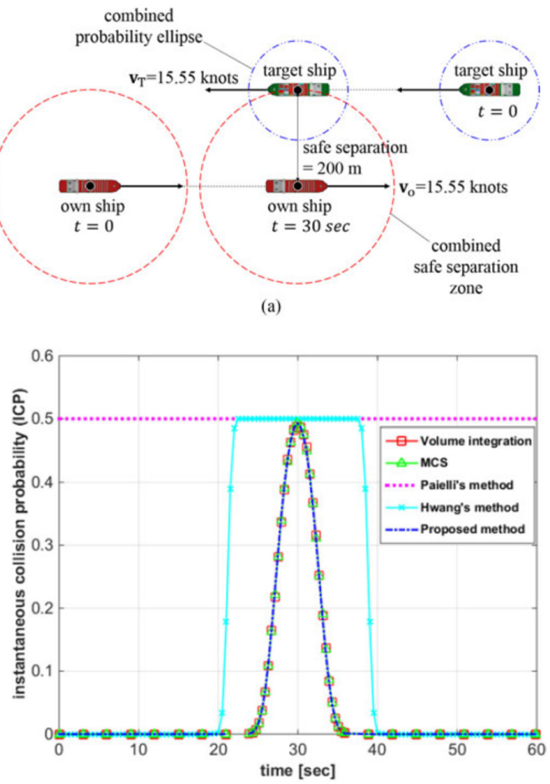


Figure 2.5 Illustration of collision probability (P_{coll}) method (from [103]).

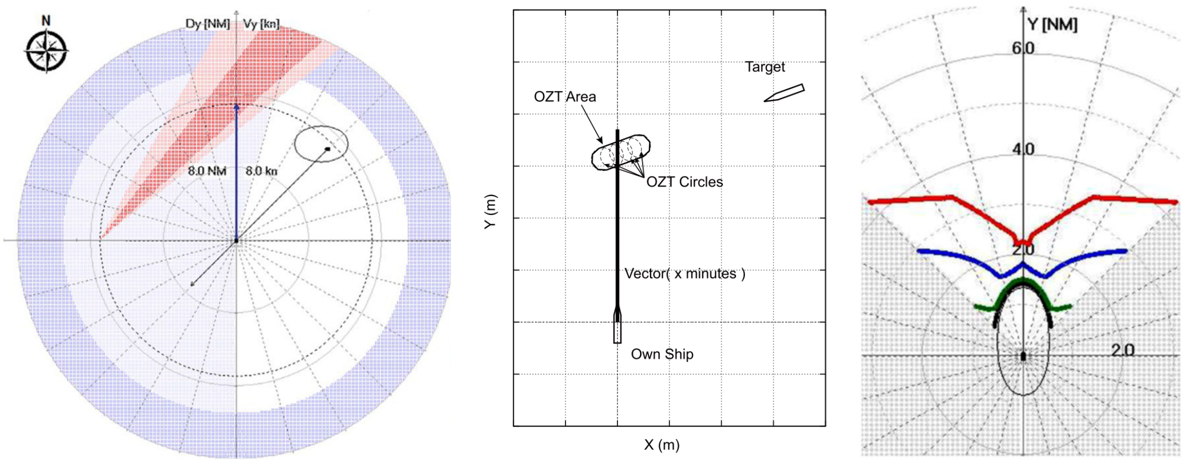
Dangerous Region (DR)

One group of methods aims at collecting a set of the OS's speed or course leading to collisions with the TS and presenting this set in the map to the OOW. A collision alarm is triggered when the current velocity of the OS is inside of this set. Since the set is presented in velocity space, this group of studies is named as DR in Velocity-space (DR-Vspace). Researchers have given various names to this kind of techniques, e.g., Collision Threat Parameter Area (CTPA) [31], Collision Danger Sector (CDS) [32], Velocity Obstacle (VO) set [104], etc. In early age, to construct the CTPA/CDS set, the researchers adopt some strict assumptions, e.g., the TS keeps a constant speed and course, and the TS and the OS are shaped as circles, etc. [31, 32]. Recently, researchers released some of these assumptions and expanded its applications in maritime practice. For instance, in [33], ship domain is introduced. An illustration is shown in Figure 2.6(1).

The other group of studies directly presents a dangerous area leading to the OS strike one TS in the electronic navigational char which is also named as the workspace of the ship. The dangerous area is usually placed at the closest point (e.g., CPA). A collision alert is launched when the current velocity of the OS might lead to OS violating this area. Since these methods use workspace instead of velocity space, this group of studies is named as DR in work-space (DR-Wspace). One representative method is called Predicted Area of Dangers (PAD) [105] or Projected Obstacle Area (POA) [106]. Other used concepts include Obstacle Zone by Target (OZT) whose size and position are determined by a joint probability distribution [107, 108], Fuzzy Collision Danger Domain (FCDD) that considers multiple factors to determine its size [109], etc. See Figure 2.6 (2).

Action Lines (ActLines)

Another group of studies focuses on identifying an action line surrounding the OS in the workspace. This line indicates the last chance for the OS to avoid collision by a fixed evasive action, e.g., a hard-port turn, etc. This concept is similar to the above-mentioned arena in Section 2.4.1. However, the determination of the action line depends on simulations rather than the experts' judgment. These studies presume that the TS keeps its initial speed and the OS takes a fixed evasive action, e.g., a hard-port turn. By multiple simulations, a set of initial positions of the TS that the OS cannot avoid collision with via the fixed action is found, i.e., action line [51] or critical distance [110]. A similar idea is presented in [111], in which the line is named as the last line of defense. Recently, researchers also presume the OS is the stand-on ship and investigated the action line. In [112], the maneuvers of both ships are considered. By repeating the simulations with different fixed evasive actions, a series of action lines are obtained, which can be used for collision alarm. An illustration is shown in Figure 2.6 (3).



(1) DR-Vspace [34] (2) DR-Wspace [107] (3) Action lines[6] [51]

Figure 2.6 Illustration of dangerous region (DR) methods.

2.4.3 Overview of existing measures of collision risk

In this section, six groups of methods have been identified, which are CRI methods, WR-SD methods, CPA method, P_{coll} methods, DR-Vspace methods, and ActLines. These methods are categorized into two big categories, namely expert-based methods and model-based methods.

Expert-based methods consider uncertainties by employing experienced experts. The estimated risk presents a general belief of a group of experts, which reflects the probability of the collision. The outputs (risk levels) represent a general judgment of participant experts, that allows the OOWs or VTS Operators (VTSOs) obtain the experts' situational awareness [71]. In practice, this group of methods is usually served as a tool that helps the human to identify the potential dangers that are believed to be dangerous by experts, e.g., CRI methods; or reminds the human to pay special attention to dangers that are believed to be/will be too close to the OS, e.g., WR-SD methods. These measures can easily incorporate the regulations [85, 91] by following modes: considering a rule-compliant ship domain [82], assigning different weights for different encounter types [71], using different measures for different encounter types [113], etc.

[6] The red/blue/green/black curse is the action line that when the TS's speed is 16/12/8/4 knots, respectively.

However, these methods share two drawbacks: (1) elimination of biases from experts are challenging; (2) the risk only reflects the belief of experts rather than the physical process of the specific collision, e.g., collision is going to happen or not. In general, these methods are popular in academia, which transfers human's experience and judgments in the risk measure and meets demands from different users (e.g., the OOWs and the VTSOs).

Model-based methods eliminate the uncertainty by simplifying a real encounter into an ideal scenario considering no uncertainty or parts of uncertainties. The measured risk presents the collision risk of a simplified scenario, which is a conditional probability of collision. This type of research might not give the OOW a real probability of collision but some facts about the scenarios, which is also helpful in collision avoidance. One main advantage of these methods is delivering users a clear conclusion that is easy for them to use. Three common drawbacks also obvious: (1) a simplified scenario cannot reflect the real environment that the OS faces; (2) it is impossible to enumerate all scenarios; (3) it is difficult to meet various demands of different users, e.g., some people are more sensitive to dangers and require a lower threshold of risk, whereas, others are risk takers and need a higher threshold. Although these methods are ideal, they help the OOW to know some facts about encounters, which is also helpful in collision avoidance. Today, some of this type of methods have been widely used in practice, in the development of ASV [79], and in some expert-based methods [114], etc.

Two representations of collision risk are found, i.e., digital form and graphical form. The digital form of risk is convenient for users to compare the risk in different cases. Specifically, users can rank the approaching obstacles by their risk levels and identify the most dangerous one. These methods include: CRI methods, CPA method, and Pcoll. On the other hand, the graphical form only categorizes the TS into various groups, while the TSs in the same group are not comparable. The graphics-based form also has its advantage which is more intuitive for users [95], and the graphical risk is easily integrated into the maps to support users to be aware of the surrounding situation [34, 104]. These methods include: Warning Ring methods, DR methods, and Action Lines.

2.4.4 Discussion on collision risk measures

Discussion on gaps in existing risk measures for conflict detection

Conflict detection contains three key sub-questions:

SQ1: who (or which ship) will strike the OS in this situation? (Collision candidates)
SQ2: when does the OS need to pay attention to the dangers? (Collision alert)
SQ3: when does the OS need to (has to) take evasive actions? (Time for evasive actions)

Traditionally, the CPA method has been seen as the solutions to all these SQs, and it has been integrated into ARPA for supporting collision avoidance. The logic of using CPA methods in conflict detection is addressed as follows:

A1: the ship whose DCPA is smaller than safety distance is the TS would strike the OS;
A2: as long as the TS might strike the OS, the collision alert is triggered;
A3: when the alert is triggered, the OS needs to take evasive actions.

The traditional way, however, might lead to frequent alerts in dense waters and the OOWs usually turn it off [107, 115]. In the dense region, the ship easily has a conflict with the OS and not all the conflicts require the OOW to pay special attention or immediately take evasive actions. Many factors need to be considered, e.g., time to collision, the obligations of the regulation (COLREGs), the OS's maneuverability, etc.

The expert-based methods are proposed to handle these issues and focus on SQ2 that triggers collision alert according to experts expectations. Since the DCPA and TCPA have been widely accepted to the judgment of potential conflict, they subsequently become the most popular indicators in CRI methods [114]. At the meantime, the development of ship domain also inspired a group of researchers to present the collision risk as warning ring instead of a digital number, which is more intuitive for the users.

The experts-based methods, however, might not be suitable for SQ3. SQ3 is related to solving collision by the OS, while most of the experts-based methods do not consider the ability of the ship solving the conflict, i.e., independent of conflict resolution. As a result, the violation of thresholds of CRI methods or WR-SD methods does not indicate whether the collision is avoidable or not.

Table 2.3 Overview of collision risk measures in existing studies

Name	Type	Form	Method	Meaning of Violations	Main Purpose			Multi-encounter
					SQ1	SQ2	SQ3	
CRI	Expert Based	Dig. R^+	Synthetic indicators	Experts believe this TS is dangerous for the OS and needs special attention.	√	√	-	P
Warning Ring	Expert Based	Gra. $[R^+]$	Expanding of ship domain	**FSD:** experts believe this TS will get too close to the OS, thus this TS is dangerous for the OS.	√	√	-	P
				SDR/Arena: experts believe this TS is too close to the OS and needs special attention.	√	√	-	P
CPA	Model Based	Dig. 0/1	Closest position given scenario	This TS will strike the OS if two ships keep their velocity.	√	-	-	P
P_{coll}	Model Based	Dig. 0~1	Probability with known distribution	This TS probably will strike the OS given uncertainties.	√	-	-	P
DR	Model Based	Gra. 0/1	Visualizing dangers present a risk map in terms of velocity. (equivalent to DCPA)	The OS will strike with one of TSs if the OS keeps its velocity and all the TSs keep their velocity.	√	-	-	M
Action Lines	Model Based	Gra. 0/1	Simulation of encounter with pre-set actions of the OS	This TS will strike the OS if the OS takes a certain action or the OS cannot avoid collision with this TS by the certain action.	-	-	√	P

Notes CRI: Collision Risk Index; DR: Dangerous Region; P_{coll}: Collision Probability. Dig.: digital form; Gra.: graphical form. R^+: positive real number; $[R^+]$: positive real number in a range, i.e., 0~1; "0/1": 0 or 1; "0~1": 0 to 1. FSD: Fuzzy Ship Domain; SDR: Spatial Danger Region. "√" means the method is suitable for solving the sub-question; "-" means the method is not suitable for the sub-question. P: Pairwise-based method; M: Multiple-based method.

In the above methods, two model-based methods consider evasive actions in risk measures, namely DR methods and ActLines methods. However, DR methods are usually used to identify the collision candidates (SQ1) and solve the conflict, which is not used to estimate the collision risk for taking evasive actions; Actions Lines are developed to find the last moment in time to take actions, but they are limited to two ships encounter case.

Additionally, most risk measures presented in this section are based on a pairwise encounter scenario. They treat multiple encounters as multiple pair-wise encounters [71] (see Table 2.3). Each measurement of collision risk indicates the threats from one TS to the OS and temporarily ignores the influence of other TSs. The entire risk level of the OS encountering all TSs together is usually out of consideration, while it is crucial for solving SQ3.

SQ3 requires a risk indicator to show the safety level of the ship, which needs to be related to conflict resolution and to handle the multiple-ship encounter. However, the existing measures cannot fulfill these demands.

Discussion on conflict detection in the manned-ship and unmanned-ship studies

In many ASV-related studies, CPA method is widely used in conflict detection module due to its simple form and high acceptance in practice, e.g., [59, 106]. However, the CPA method might not sufficient for developing a reliable conflict detection module. Firstly, the CPA method is unreliable due to its limitations which mainly due to its strict assumption on the TS's movement (see, [71, 104]). Secondly, using CPA alone in conflict detection is not enough for detecting dangers in various scenarios and multiple indicators are needed. Some developments in expert-based methods can offer a better alternative for the ASVs. Thirdly, to incorporate the navigational regulations in conflict detection, experts' knowledge is necessary to be introduced. The regulations are general and designed for the human. Thus, teaching the ASV to assess collision danger incorporating with regulation, the role of human cannot be ignored. From this perspective, experts' knowledge is necessary to be introduced but still needs more studies in the future [113]. Many expert-based methods considering the COLREGs [91] might also be applied in the unmanned ship for developing a rule-compliant ASV.

In the manned ships, conflict detection is mainly servicing as reminders for the OOWs, which needs to be adapted to human's performance. For example, the navigators who are risk taker might accept some risky scenarios, while others might see these scenarios as dangers. Thus, the expert-based methods are widely presented in studies for supporting the manned ship and Fuzzy methods are popular in this type of studies. However, this type of collision criteria cannot express more details about the collision process. It cannot give an explicit judgment about collision happening or not to users. Thus, model-based methods are still popular. Particularly, the model-based methods trying to find the last moment for preventing collisions become a new trend in recent years. Nevertheless, it is worthy to be aware that many model-based methods assuming deterministic system simplified the problem. The real system is a stochastic dynamic system, which contains various stochastic processes (such as weather conditions, etc.) [116]. Thus, it is necessary to incorporate the stochastic process in model-based methods in the future.

To summarize, expert-based methods can offer customized services for different navigators which can meet users' preference, and model-based methods usually addresses some relative objective fact which shows the hard boundary of safety and danger. These two types of methods can be combined to offer a better service to manned and unmanned ships.

2.4.5 Summary of conflict detection techniques

Conflict detection module mainly answers three sub-questions: SQ1, who will strike the OS; SQ2, when the OOW needs to pay much attention; and SQ3, when it is the time for applying evasive action. The techniques for conflict detection are collected in two main categories with six groups. CRI methods and WR-SD methods belong to expert-based methods; CPA, P_{coll}, DR-Vspace, DR-Wspace, and ActLines are model-based methods.

These methods serve conflict detections with different focuses. The CPA methods are popular in finding the potential dangers both in manned and unmanned ships, but it is not good at solving SQ2 and SQ3, especially, it easily results in frequent alarms. The expert-based methods, thus, are introduced to handle this problem, which is suitable for SQ1 and SQ2, but not for SQ3. In particular, the violation of a risk threshold provides limited information about whether the collision is avoidable since the ability of the OS avoiding the collision is neglect in the most expert-based methods. Two methods incorporate the OS's evasive actions. However, they are not suitable for triggering evasive action in multiple-encounter scenarios, which is a common shortcoming of the existing risk measures. In brief, the determination of time for taking evasive action is still an open question, which not only needs to incorporate the evasive actions of the OS but also suitable for multi-encounter scenarios.

For improving conflict detection in manned ships, a combination of expert-based methods and model-based methods would be suggested. The expert-based can offer customized services for different navigators which can meet users' preference, whereas the model-based method usually addresses some relative objective fact that shows the hard boundary of safety. These two methods can be combined to offer a better service to manned and unmanned ships.

Conflict detection for unmanned ships usually employs CPA based methods, but many manned-ship studies revealed the limitations of using CPA in conflict detection. To handle the shortcoming of using CPA methods and to consider COLREGs in conflict detection, some navigational assistance methods from manned-ship studies can offer some cues, specifically incorporating expert-based methods in the conflict detection module.

2.5 Conflict resolution

Conflict resolution is a core of collision prevention, which determines collision-free solutions. Many methods have been develop. However, most of them are similar, even though these methods appear to be quite disparate [117]. In this section, similar methods are collected in the same groups, and six groups of ship collision avoidance techniques are identified. They are

(1) Rule-based method uses "If-then" rules to guide the collision avoidance process;
(2) Virtual vector method generates a virtual vector field to guide the ships;
(3) Discretization of controls with collision check method searches a collision-free solution or an optimal solution in discrete control space.
(4) Continuous controls with collision constraints method find the optimal solution in continuous space with collision constraints;
(5) Re-planning method formulates the collision avoidance as a path planning problem and searches collision-free path in free configuration space;
(6) Hybrid method combines some of the previous methods in collision avoidance.

The details of these algorithms and their applications are presented, followed by an overview of these methods with their advantages and disadvantages. Then, the developments of collision avoidance algorithms and their performance in manned and unmanned ships are discussed, and the findings of these discussions are concluded at the end.

2.5.1 Main algorithms

Some terminologies used in the description of these algorithms are addressed here. The workspace of the ship is the horizontal plane consisting of positions. Configuration space (C-space) contains all the possible configurations of the ship. In this book, the ship's configurations refer to the position and the orientation. Additionally, the C-space of the OS consists of collision-free configurations (C_{free}) and obstacle configurations (C_{obs}) that results in collisions.

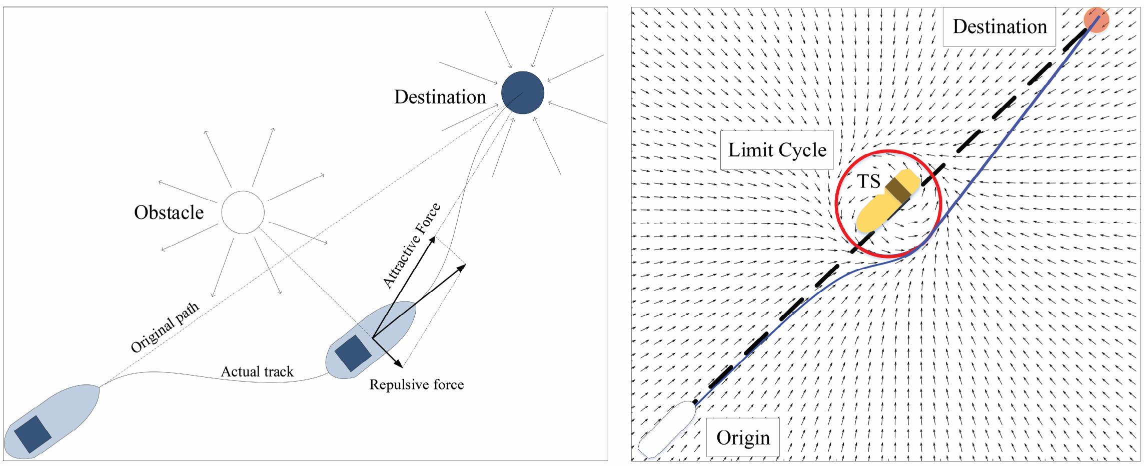
Rule-based Method

Rule-based methods use a set of pre-set rules to guide collision avoidance. For example, when a ship encounters with other ships, the ship will turn 75 degrees (or 30 degrees) to the starboard side (right-hand side) [118, 119] or enlarge rudder angle until the trajectory is collision-free [48].

Obviously, a single rule cannot handle all kinds of encounters in a dynamic environment. Thus, multiple rules are considered. A widely used way is incorporating "International Regulations for Preventing Collisions at Sea" (COLREGs) and good seamanship in a rule system that is also known as experts' system. This system is expected to suggest rule-compliant actions for the OS in various scenarios, which is usually based on Neural networks [120], Fuzzy logic, or Bayesian network [121]. Since the enumeration of rules for all scenarios is impossible, this method does not guarantee a collision-free solution. If a case is not studied in advance, this method might not find out a proper solution, which is one limitation of these methods.

Virtual vector field Method

Virtual vector method generates a virtual field to guide the OS's motion. Two specific algorithms are found: Artificial Potential Field (**APF**) and Limited Cycle Method (**LCM**).



(1) Artificial Potential Field (APF) (2) Limited Cycle Method (LCM)

Figure 2.7 Illustration of virtual vector field Methods (1) APF and (2) LCM.

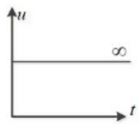
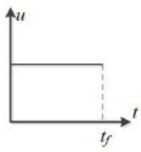
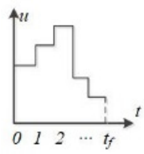
APF [122] (or virtual force field [123]) generates a repulsive potential field around the obstacles and an attractive potential at the destination. The sum of these potential fields determines the resultant virtual force to guide the motion of the vehicle, see Figure 2.7 (1). This algorithm does not directly provide a collision-free path but a direction of motion, which was not designed for a dynamic environment at the beginning. To avoid collisions in a dynamic environment, researchers improved this basic algorithm by considering two factors: the velocity of obstacles and the maximal deceleration of the vehicle [124]. Specifically, the repulsive potential of the obstacle is enlarged by considering these factors. This technique was also applied in ship collision avoidance, see [125, 126]. One main disadvantage of this method is that the ship might be trapped in a local minimum and stop at the local minimum. The conditions leading to local minima are concluded in [127]. Additionally, the dynamics of ships are not fully taken into account, where the ship is assumed to be holonomic.

LCM uses a stable limit cycle for motion planning. The stable limit cycle has a feature that all the neighboring points are attracted to the cycle. Researchers place the center of the limit cycle at the center of the obstacle, which encompasses the whole obstacle. A series of trajectories converging to the cycle is generated. Then, the OS follows one of these trajectories until the destination is clear, see Figure 2.7 (2). The ellipse and circle-shaped cycles are used in [42, 128] and its application of ship motion planning in rough sea conditions is presented in[129]. However, this algorithm usually requires that the obstacles are relatively static, or that the speed of the OS is much higher than the obstacles; and it only avoids one obstacle at a time. These limitations might lead to problems when the OS encounters multiple obstacles, or an obstacle's speed is higher than or equal to the OS.

Resolution search in discretizing control-space with collision check

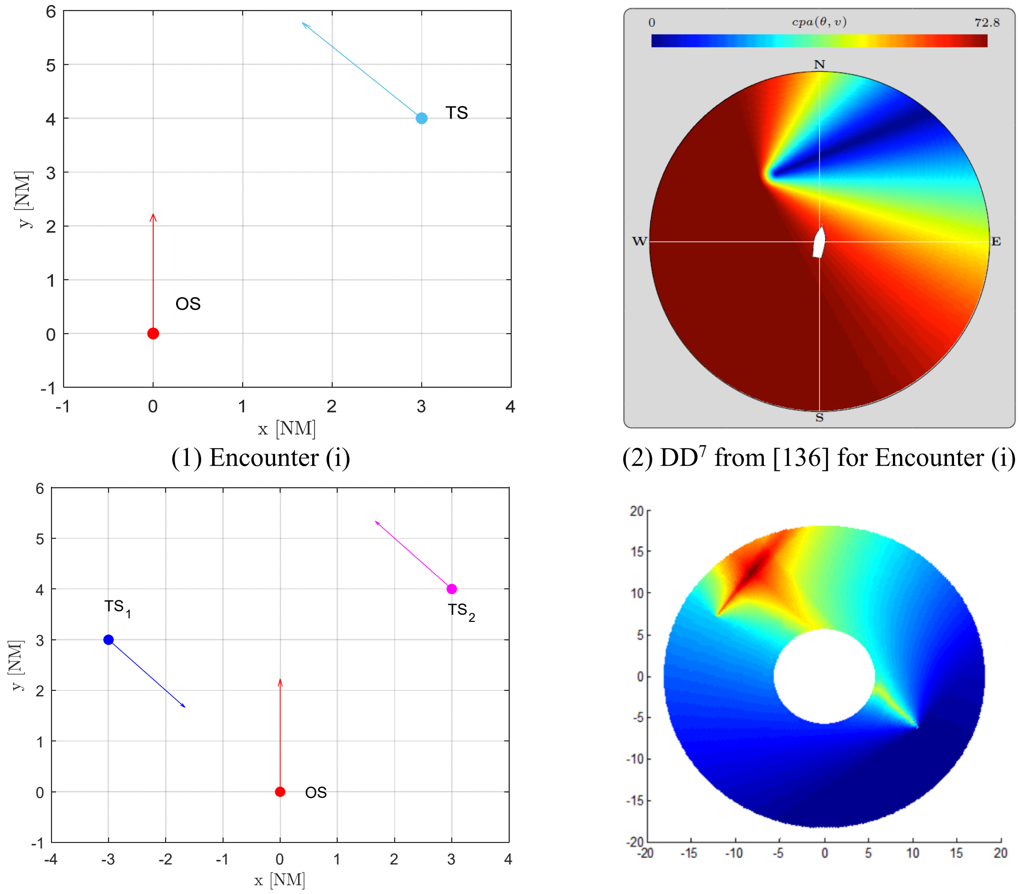
Another group of methods discretize the control space of the ship and eliminate the dangerous controls by a collision check. Then, a collision-free solution is chosen from the rest of the control space. These studies assume that the motion model of the OS is known, and the environmental disturbance is temporarily neglected. Then, given one control and its duration time, the trajectory of the OS is deterministic, and the collision check is possible. These types of methods discretize the control space first, and then different algorithms differ in duration time of the control, see Table 2.4.

Table 2.4 Overview of three groups of methods using discretization of control space

Name of methods	Controls	Duration	Illustration	Ref.
Decision Disc (DD)	(ψ,u)	$[0,\infty)$		[29-32, 73, 130, 131]
Dynamic Window (DW)	(u,r)	$[0,t_f]$		[132, 133]
Discrete inputs Optimization (DIO)	(ψ,u) or δ	$[0,t_f]$		[74, 134]
Lattice-based Search (LBS)	(ψ,u)	$\{0,1,...,t_f\}$		[59, 135]
Brute-force search (BFS)	ψ	$\{0,1,...,t_f\}$		[47]

(1) Each control keeps constant in the future

Decision Disc (**DD**) approach chooses course and speed as controls to the ship and presents the control space as a disc. The control input is assumed to be invariant in the future. Thus, each control has a unique trajectory. If this trajectory is collision-free, the control is reserved; otherwise, the control is rejected. Then, the collision-free controls are directly presented to the officers [30], or an optimal solution is chosen from those collision-free controls by optimization [29]. In this process, DCPA is usually employed for the collision check. This method has different names in different studies, e.g., Collision Threat Parameter Area (CTPA), Collision Danger Sector (CDS), etc. [31, 32]. When being used in practice, this method is also incorporated with ship domain [131], restricted waterways, worse environmental scenarios [130], etc. A similar technique is also applied in an ASV [73], where the irregular shape of the obstacles, sensing errors, and the COLREGs are considered. The main disadvantage of these studies is neglecting the kinematic and dynamic constraints of the OS, which might lead the method to fail to avoid collisions in close range encounters.



(1) Encounter (i) (2) DD[7] from [136] for Encounter (i)

(3) Encounter (ii) (4) DD[8] from [137] for Encounter (ii)

Figure 2.8 Illustration of Decision Discs (DD) from different studies.

[7] Color in this disc refers to DCPA. Thus, the smaller DCPA (in blue) is danger and the larger DCPA (in red) is safer.
[8] Color represents the danger level that is related with smallest distance between ships. Red is danger and blue is safety.

(2) Each control keeps constant in a given time window

Dynamic Window approach (**DW**) [138] chooses velocity tuples (u,r) as control inputs (*u* is linear velocity, and *r* is angular velocity) and presumes the chosen control is fixed in a given time step. The construction of the dynamic window includes two steps: firstly, all the tuples that the OS can reach in given time step are selected as an initial dynamic window (V_d in Figure 2.9), in which velocity and acceleration constraints are considered; secondly, the initial dynamic window is reduced by keeping those tuples that ensure the vehicle can stop before hitting with obstacles (i.e., V_a). The remaining tuples contain all admissible velocities, which constitutes the dynamic window (i.e., V_r). The optimal collision-free velocity tuple is searched in this dynamic window. The limitations of original DW include: susceptible to local minimum, assumption on circular arcs path, frozen environment during decision time step, etc. [139] Some improved algorithms solve parts of these problems: [140] incorporated connectivity of free space to avoid local minimum; [141] extended DW to deal with moving obstacles; [132] demonstrated DW with non-circular paths and applied the method in ASVs; [133] modified the original DW considering the dynamic of ASVs. However, either the original DW or the variations hold an assumption that the static state is always a safe state for the vehicle, which might not be held in practice, especially when a ship is crossing a busy traffic intersection.

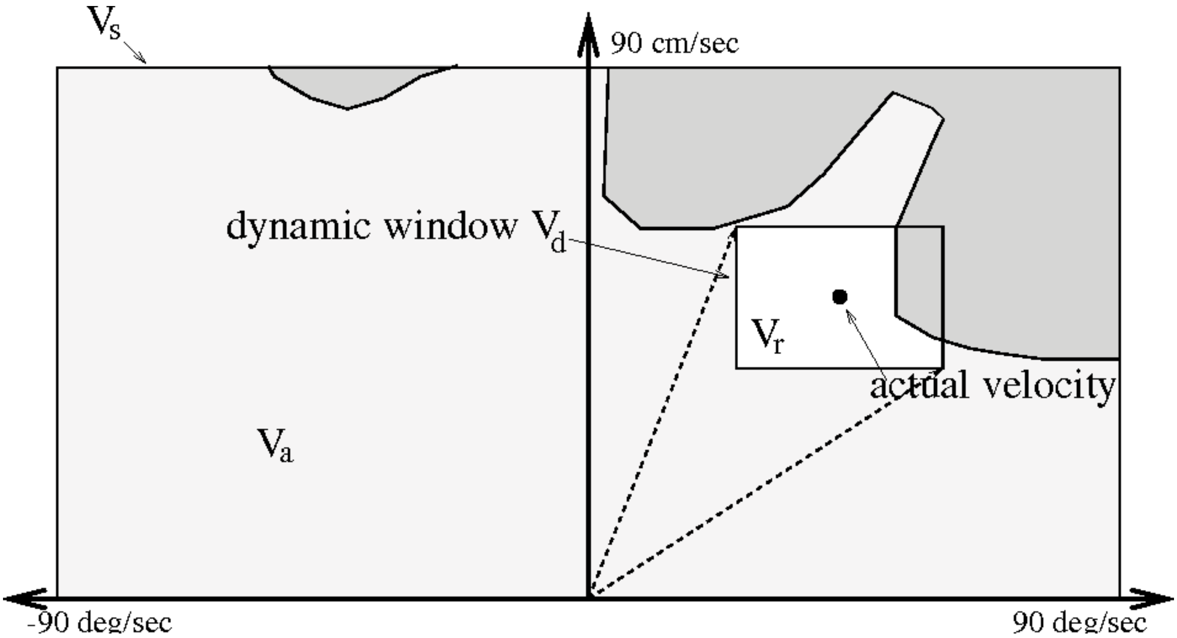


Figure 2.9 Illustration of Dynamic Window (DW) method (from [138]).

Discrete-Inputs Optimization **(DIO)** uses a group of control candidates as representatives of the whole control space and find a collision-free solution by optimization, which cuts down the computational burden. To avoid the obstacles, the cost function assigns a high cost to the trajectory close to the obstacle. The ship obtains an optimal control and the corresponding trajectory by minimizing the cost function [142]. An example of this method is shown in [74] where Model Predictive Control (MPC) with soft collision avoidance constraints in the objective function is used. In some article, this group of methods is called sampling-based MPC [143]. Instead of applying this optimal trajectory, MPC applies the optimal control at the first stage and update the optimization to obtain a new optimal solution/trajectory. More examples see [134, 144]. For these methods, how to balance the optimality and computational time is a challenging topic.

(3) Control changes at each time step

Lattice-Based Search **(LGS)** method allows control input to change at each time step [145, 146]. Since this type of algorithms is time-consuming, researchers usually search some representative candidates rather than the whole control space in each step. In a time step, these control candidates will generate a graph that consists of the trajectories of the OS with different controls. This graph can be reused to generate all the paths of the OS in the next time step, which saves the computational time. In return, a lattice-based graph is obtained, see Figure 2.10. By searching this graph, a collision-free trajectory and the corresponding control in each time step are obtained. Paper [59] and [135] applied this method in ship collision avoidance. The control input is defined as a vector (u_d, ψ_d) where u_d and ψ_d are desired surge speed and course of the ship. Ship dynamics are considered to generate the graph in their research. The main disadvantage is that searching all the branches in the graph are computationally expensive, which might not be used for real-time collision avoidance. To reduce the computation, [59] considered a small discrete control-space with 30 controls and employed risk assessment to assign a priority search direction in the graph.

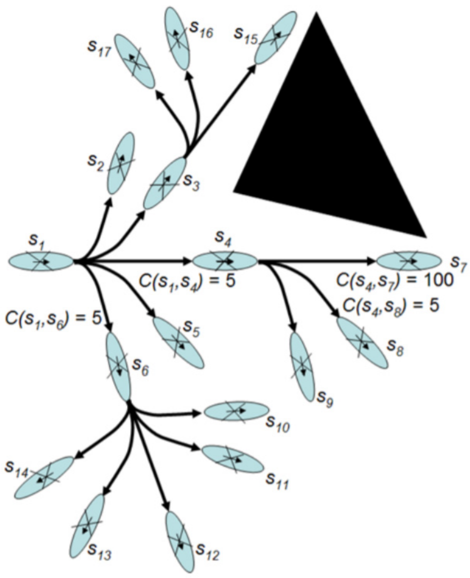


Figure 2.10 Illustration of lattice-based graph (from [146]).

Brute-Force Search **(BFS)** method is also one of the options. Controls-space and time horizon are discretized into cells. Each cell contains a combination of controls and its duration time. Correspondingly, a path of the vehicle can be generated, in which constraints on kinematic and dynamic can be considered. A collision check is used to abandon or keep the cell. Since the computational complexity is a big issue, especially for online decision making, some modifications are needed. For example, in [47], only one input is considered (i.e., desired course) and the course and time horizon are searched in the range [30°, 60°] and [180 sec, 900 sec]. The search starts in the time direction and stops when the first collision-free cell is found. The main drawbacks are that it has high computational complexity, and the solution is not optimal.

Resolution search in continuous control-space with collision constraints

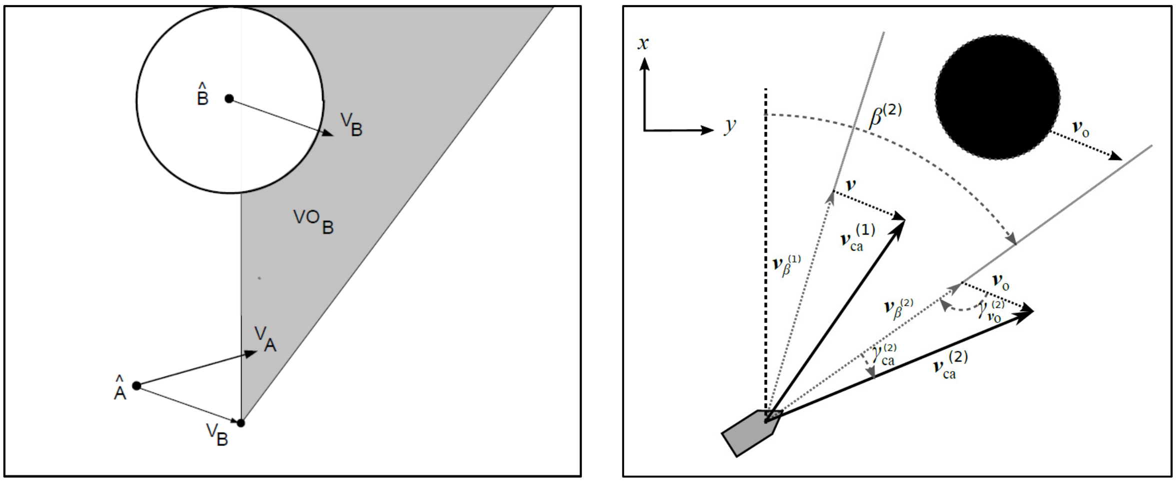
Some methods are not relying on the discretization of the control space and find the collision-free solutions in continuous control space. Two groups of methods are found according to the order of collision check.

(1) Identify collision-free control space first

This group of methods usually conduct a collision check first and then find an optimal solution in a collision-free space. Instead of checking each control one by one, these methods use a polygon/circle to represent an obstacle and then conduct additional operations to formulate a set of controls leading to collisions. Then, an optimal solution is obtained according to a cost function in collision-free controls.

Velocity Obstacle (**VO**) algorithm is one popular algorithm proposed in [147]. A circle is used to represent an obstacle, and researchers assume the obstacle moves with constant speed and course. In return, they formulated a set of velocities that lead to the relative velocity point to the obstacles as a VO set which shapes like a cone, see Figure 2.11 (1). Due to this simple shape, each VO set can be formulated by three linear constraints [148]. The idea of VO algorithm has been adopted in maritime research, e.g., [149]. Three key disadvantages of the original VO algorithm that pointed out by researchers are: (1) the algorithm holds the assumption that the velocity of obstacle has to be constant in the future; (2) the dynamic constraints on the OS and the TS are usually out of consideration; (3) the shape of obstacle is assumed to be regular and convex, e.g., circle, ellipse, etc.

Vision Cone (**VC**) method is an enlarged collision cone [150] with a buffer angle [151], see Figure 2.11 (2). The courses are chosen as control inputs to the ship, and two collision-free courses are identified by the vision cone, which are the boundaries of the cone. The optimal collision-free course is chosen from two collision-free courses, i.e. $v_{ca}^{(1)}$ and $v_{ca}^{(2)}$ in Figure 2.11 (2). This algorithm can deal with moving obstacles whose velocity is constant. In [152] and [153], this idea has been applied in ASVs. In [41] and [154], the authors expanded the algorithm for the under-actuated unmanned ship. The algorithm guarantees safety, while it only deals with one obstacle in each time and requires the speed of the obstacle to be smaller than the speed of the OS.



(1) from [147] (2) from [154]

Figure 2.11 Illustration of (1) VO method and (2) VC method.

(2) Optimization with Model Predictive Control in discretizing and finite time horizon

Another group of methods usually conduct a collision check and optimization together. Specifically, the collision check is one hard constraint in the optimization, e.g., $x(t) \in C_{free}$. A general form of this optimization is formulated as:

$$\begin{aligned} &\min_{u \in U} J(u, x) \\ \text{s.t.: } &x(k+1) = f\left(x(k), u(k), k\right); \\ &x(k) \in C_{free}, \forall k \in \{0, 1, 2, \ldots t_f\}; \\ &x(0) = x_0; \end{aligned}$$

where J is a cost function which depends on the control inputs and state of the system; $f(x,u,t)$ is the dynamics of the ship; C_{free} is collision-free configuration; x_0 is the initial state of the system (the ship).

In this optimal control problem, time is discretized, the dynamics model of the OS is discrete, and control input at each time step is variable that can be used to optimize the cost function [49, 68]. The constraints of this optimal control problem include kinematic constraints, the dynamics of the vehicle, collision-free conditions, etc. The solution of this optimization problem is collision-free solutions for the OS, which is executable and stratifies collision-check. In the literature, this method is usually combined with a receding horizon scheme. Thus, this group of methods is named as MPC-based Collision Avoidance **(MPC-CA)**. Paper [49] use a linearized dynamics model of ASV and a linear function (infinity norm) in collision check, which can solve the optimization problem efficiently. Some researchers consider the non-linear dynamics model and nonlinear function of collision-check. Paper [40] used two circles to represent ships and solve the optimization problem via a direct multiple shooting method. Paper [155] used circles and ellipses to present the OS and obstacles, respectively, and solve the optimization problem via a commercial solver. Though these studies show that this problem can be solved in a reasonable time, the solution is a local minimum due to the nonlinearity and non-convexity of the problem. Additionally, as time is discretized, a collision-check between time steps is needed, which is noted as safety verification.

Re-planning method

Instead of searching the collision-free solution in control space, **re-planning method** searches solutions in workspace directly. The re-planning is triggered when the collision criteria reach pre-set thresholds [156]. At each time step, the graph searching method is used to find an optimal collision-free path, which handles collision avoidance as a path planning problem in each time step [58]. Various algorithms based on this method have proposed. Two main differences among the algorithms are the assignment of cost in each cell/node and the approaches used to search the optimal solution. Some researchers assign a high cost to the cells surrounding the moving obstacles considering the speed of the obstacle, e.g., [35, 157, 158], while the other assigns an area surrounding the Projected Obstacle Area (POA) with a high cost, e.g., [106]. Based on the map of costs, researchers use searching algorithms to find an optimal path with lower cost, such as Fast Marching Method (FMM) [157], evolutionary methods [159], ant colony algorithm [160], genetic algorithms [161], and particle swarm optimization [162].

Hybrid algorithms

In previous sub-sections, the algorithms in the literature are addressed in detail. In fact, in maritime practice, researchers usually combine those algorithms to perform collision avoidance. For instance, the rule-based method is usually combined with other algorithms, such as VO, DW, DD, etc., to make sure the behavior of the ship compliant with regulations. In [73], the author used VO algorithm and COLREGs to exclude the velocity resulting collisions and the velocity violating regulations. In [163], VO algorithm was combined with APF which services as a global planner. In [125], the rules-based method incorporates with APF for guiding the unmanned ships.

2.5.2 Overview of conflict resolutions

The details of the comparison between the aforementioned methods are listed in Table 2.5. This section mainly addresses the general feature of each group of methods.

In general, rule-based methods are simple and easy to conduct. However, these methods cannot enumerate all the scenarios, especially when encountering with multiple obstacles. For example, the navigational rules only address the obligations of ships in a two-ship encounter scenario, while the ship might encounter more than two ships and more complex environmental conditions.

Virtual vector methods contain APF and LCM, which usually ignore the ship's dynamics during motion planning. The APF method might get trapped in a local minimum, which is addressed in many studies. LCM method only considers one obstacle or one group of obstacles in each time, and the obstacles need to be stationary or have a relatively low speed. Nevertheless, researchers found that the LCM method performs well with stationary obstacles in rough seas [129]. Both APF and LCM only show one solution to the ship, and this solution might not be optimal.

Discretization inputs with collision check is a big group of methods. These methods could take ship dynamics into account. Specifically, researchers assume the dynamics of the ship is known and predict the trajectory of the OS with different inputs. However, the calculation of each control is time-consuming. To solve this issue, some simplifications are necessary. Some methods only consider the change of input at the first time step, see DW, DD, etc. These methods not only offer one optimal collision-free solution but also present a set of unsafe solutions at the first time step. Others only select several representative inputs and consider changes in more time steps, such as LGS and BFS. These methods offer one collision-free solution to the ship from the selected inputs. A common challenge of these algorithms is the balance between efficiency and effectiveness. A small grid and short time step would benefit a better solution, but it is time-consuming; A big grid and long step save computational time, but it might skip the optimal solution. Moreover, the quality of prediction is quite important for collision check, while only a few of these studies discuss the impacts of uncertainties on these techniques.

Table 2.5 Overview of different collision avoidance algorithms

	Name	Motion Model	Number of TSs	Solution form	Limitations	Ref.
Rule-based	Single -rule	NA	Single	One solution	Simple, but only works in limited cases, e.g., open sea and few obstacles.	[48, 118]
	Multiple-rule	NA	Single	One solution	Simple, but might not figure out all the possible scenarios.	[121]
Virtual Vector	APF	NA	Multiple	One solution (course)	Simple, but easy goes to a local minimum, not for a dynamic environment, and ignore dynamics.	[125, 126]
	LCM	NA	Single	One solution (trajectory)	Simple, but only works with one high-speed obstacle and ignore dynamics.	[128, 129]
Discrete Inputs	DW	kinematic /dynamic	Multiple	One optimal solution (u,r) and dangerous solutions	Popular in robotic, but it is developed for a static environment, besides stop is not always a collision-free solution.	[132, 133]
	DD	holonomic	Multiple	One optimal solution (u,psi) and dangerous solutions	The assumption on holonomic motion model leads it difficult to generalize.	[29-32, 73, 130]
	DIO	dynamic	Multiple	One optimal solution	The balance between efficiency and effectiveness is challenging.	[74, 134]
	LBS	dynamic	Multiple	One optimal solution	Considering traffic congestion to reduce the search burden, while computing burden is still a challenging issue.	[59, 135]
	BFS	dynamic	Multiple	One solution (course, t)	The process is time-consuming, and the solution is not optimal.	[47]
Continuous Inputs	VO	holonomic	Multiple	One optimal solution (u, psi) and dangerous solutions	Simple and efficient, but strong assumptions.	[149]
	VC	dynamic	Single	One solution (course)	Simple and effective, but each time only can avoid one obstacle with low speed.	[41, 152, 154]
	MPC-CA	dynamic	Multiple	One optimal solution (forces)	Local minima and computing time is dependent on solvers.	[40, 49, 155]
Re-planning	FMM	NA	Multiple	One optimal solution (path)	Strongly depend on fresh frequency and velocity of an obstacle. Might lead to an inevitable collision state.	[157]
	POA	NA	Multiple	One solution (path)	Strongly depend on fresh frequency and velocity of an obstacle. Might lead to an inevitable collision state.	[105, 106]

NA: means ship dynamics is not directly considered in finding a collision-free solution, but during the implementation of the solution, the ship dynamics is considered.

Continuous inputs method is not relying on the discretization of the control inputs. One group of algorithms collects all the solutions leading to collisions in a set and then conducts optimization, e.g., VO, VC, etc. The other considers the collision check as one constraint in optimization and directly solve the optimal solution by solvers, e.g., MPC-CA. The difference between these algorithms is the form of the solutions. VO algorithm and its variations can visualize and present the unsafe solutions to users, which can help them to understand the choice of the optimal solution. Moreover, the ship is required to keep one solution in the whole prediction horizon. On the contrary, MPC methods directly offer one optimal solution which allows the ship to change course and speed in the prediction horizon. However, the optimal solution is calculated by solvers, which is not transparent to users.

Re-planning method follows the idea of path planning, which constructs a cost map firstly and then searches an optimal path on the map. The method, however, cannot avoid the situation that a ship violates an inevitable collision state around the dynamic obstacle. An inevitable collision state is a state (positions) that one ship cannot avoid collision with others no matter what actions the ship performs. This method usually offers one optimal path for users, but the dynamics of the ship is ignored.

2.5.3 Discussion on conflict resolutions

Discussion on developments of conflict resolutions in recent years

In recent years, many techniques have been proposed and used to develop ASVs, which enriches the tools for solving the collision avoidance problem. Correspondingly, some highly unrealistic and strong assumptions, which were criticized by researchers, have been released. Some important factors, like environmental disturbance and regulations, are also discussed in some studies, e.g., [50, 130].

In the beginning, the dynamics of the OS is usually ignored, and the OS is seen as a holonomic vehicle; besides, the moving obstacles are assumed to be semi-dynamic. Recently, many algorithms are capable of considering the dynamic constraints, e.g., MPC-CA, DW, etc. In those methods, the motion model of the OS is known or identified in advance. With the improvements in prediction techniques (in Section 2.3), many algorithms do not request the moving obstacle to keep its initial speed and course.

In the previous review articles, researchers argued that environmental disturbance is a critical factor in collision avoidance, which was less discussed in relevant studies. In recent years, many studies try to take this factor into account. Two lines of thinking have been followed to handle the disturbances. One assumes a perfect knowledge about the bound environmental disturbance and the possible trajectories of the OS can be calculated or bounded in a tube. Then, the solution that leads to all these possible trajectories/tubes away from the obstacles is the collision-free solution under environmental disturbance, see [50, 74]. The other one concentrates on the changes of control space caused by environmental disturbances, e.g., some feasible solution might not reachable in a harsh environment. Researchers try to figure out all the unsafe/infeasible solutions due to the harsh conditions, e.g., [130]. Some guidelines of navigation in adverse weather and sea conditions are used to identify and block dangerous solutions.

Another pitfall which has been highlighted in previous research was the lack of consideration of navigation regulations, e.g., COLREGs. In recent year, researchers introduce part of navigation rules in collision avoidance, e.g., [73, 74, 121, 126], etc. Some popular rules are frequently used in finding a rule-compliant collision-free solution, i.e., Rule 6, 8, 13-19 from COLREGs. These rules address the obligations of ships in two-ship scenarios. It seems that building a complete regulation-compliant ASV is close. However, the regulation is written for the human, which makes rules open to some interpretation and difficulty to "translate" in machine language [113]. Thus, incorporating all the rules from COLREGs and good seamanship in an autonomous system is still an open question. Quantifying the entire regulations and good seamanship for the ASV still need more efforts in the future [45, 113]. On the other hand, modifications of rules and regulations regarding different levels of ASVs need further investigations, as well.

Discussion on conflict resolutions in manned and unmanned ships

Most of the conflict resolutions are developed for automatic collision avoidance. However, they have great potential to support the human on board. Some algorithms, such as BFS, LGS, MPC-CA, etc., can offer a feasible/optimal collision-free solution to the OOWs directly. Other algorithms collect dangerous solutions and present them to users, such as DW, VO, DD, etc. However, these algorithms might not directly apply to manned situations. The reasons are as follows:

Firstly, collision-free solutions that some algorithms find are not friendly for the OOW. For instance, MPC-CA offers a series of forces acting on the ship for collision prevention, while the navigators might not know the effects of these forces and how to steer the ship to generate such forces.

Secondly, some algorithms might offer a readable solution for human, but they usually use ideal motion models, e.g., VO, DD, etc. Consequently, collision-free solutions may become unsafe or unreachable in certain situations. Specifically, when the relative distance is not large enough to ignore the errors between real dynamics and ideal dynamics, the solutions these methods find may still lead to collisions [56].

2.5.4 Summary of conflict resolution techniques

Many techniques proposed in recent years enrich the tools for preventing collisions, which are categorized into six categories. In this section, 14 collision avoidance algorithms have been introduced in detail.

In many collision avoidance methods, some limitations addressed in previous studies have been overcome. Firstly, many collision avoidance techniques incorporated the dynamics of the OS in finding a collision-free solution, which makes the collision-free solutions more feasible in practice. Secondly, many collision avoidance techniques can handle the TS whose movement is not constant. Thirdly, the environmental disturbance has been considered in two ways, e.g., introducing a tube or eliminating unsafe actions. Fourthly, incorporating some COLREGs rule for simple encounters (two ship encounter) has been widely seen in many studies.

Most of these emerging techniques are developed for collision avoidance in ASVs, which also have the potential for collision avoidance in the manned ship, in particular, finding an optimal collision-free path, checking collision-free of human's solutions, etc. However, these algorithms might not directly apply to manned situations. Firstly, collision-free solutions that some algorithms find are not friendly for the OOW. Secondly, some algorithms which can offer a readable solution for human ignore ship dynamics.

2.6 Discussion

2.6.1 Developments of collision avoidance in maritime research

In recent years, many efforts have been put to fill those gaps addressed in Section 2.1.1 and have led to some changes. This section provides a summary:

- Environmental factors are taken into account in some studies. Specifically, some researchers incorporate prediction errors in collision avoidance with an assumption that the uncertainties are bounded; alternatively, some researchers exclude the maneuvers that are unsafe in harsh environments.
- Rules from COLREGs have been considered in some simple encounter scenarios. However, more efforts are needed to apply the whole COLREGs rules and good seamanship in ship collision avoidance, especially in multiple-ship encounters;
- Many algorithms can handle collision avoidance with dynamic obstacles, but the trajectories of the TSs have to be known or bounded;
- The dynamics of ships have been taken into consideration in many studies. Nevertheless, these models are usually known and deterministic;
- Efficiency and effectiveness have been considered, while less discussion on the trade-offs between them in different scenarios.

Together with these changes, the following limitations and challenges in the development of a new collision avoidance system for ships are highlighted:

(1) Uncertainties of ship motion models are usually ignored.

Most existing studies use deterministic dynamics models in collision avoidance. However, uncertainties are inevitable. Specifically, these uncertainties mainly come from unmodeled dynamics, changes of parameters due to different working conditions, uncertainties of external disturbances, etc. It is observed that many existing studies had discussed the environmental disturbances, but few of them consider the uncertainties by parameters and unmodeled dynamics. However, these factors cannot be ignored since the parameters and dynamics would change when a ship has different working conditions. How to handle these uncertainties is challenging but essential for developing automatic collision avoidance systems on board.

(2) Developing fully rule-compliant navigation systems is still an open question.

Although many studies have implemented several rules when deciding collision avoidance actions, the research on the development of a completely rule-compliant system is still blank.

Firstly, many complicated scenarios in practice might activate multiple rules in COLREGs. Choosing the most suitable rule is difficult for collision avoidance systems. For instance, in a multiple-ship scenario, the ship might have two conflict obligations, e.g., “give way” and “stand on”; the ship might encounter with multiple types of ships (e.g., fishing ship, sailing ships) which apply different rules.

Secondly, the navigation rules, e.g., COLREGs, is written for the OOWs in human’s language [113], which is a guideline without quantifying information for the machine. For instance, the ship is asked to keep at a safe speed, while the value of safe speed is not addressed in the rules.

In fact, the implementation of COLREGs strongly relies on experts’ knowledge that is also noted as good seamanship [45]. A few studies focus on incorporating the seamanship in the navigation system, e.g., [29, 45, 113]. However, more efforts are needed before the navigation system can completely interpret the whole COLREGs in various scenarios.

(3) Risk measures for taking evasive action in multiple-encounter scenarios are missing.

A measurement of collision risk plays an important role in triggering evasive actions in collision avoidance, which determines the (last) timing for the ship to prevent a collision. However, few of the existing risk measures are fully suitable for this purpose.

Firstly, these methods are not applicable for the multiple-encounter scenario that is common in dense waters, and the measured risk only reflects the danger level of each approaching ship rather than the danger/safety level of the OS. Thus, in the multiple-encounter scenario, the existing methods cannot address the risk of collision and the changes on risk when the number of the TS is changed. Secondly, the existing risk measures neglect the ability of the OS to avoid collisions, and the risk level is relatively independent of the collision prevention event. That means the violation of the risk threshold does not indicate that the collision is avoidable or not.

From many investigation reports, “action too late” is a major cause of collisions and a late action often attribute to a late detection [164]. Proper risk measurment tool is needed, which not only identifies the potential dangers but also can trigger an alarm before the ship cannot solve the conflict. Thus, there is a need for incorporating conflict resolutions in risk measures that also handle multiple-encounter cases.

(4) Discussion on working conditions of collision avoidance methods is lacking.

The existing collision avoidance methods, more or less, are suffering from various problems. Thus, each method has its working conditions. If the working conditions are matched, the method can offer collision-free solutions; otherwise, the collision-free solution is not guaranteed. However, only a few studies discuss their working conditions. The working conditions include but not limited to the initial distance between ships that one algorithm can find a collision-free solution, the maximum of computation time that find a collision-free solution, the errors in solutions, the maximum of tolerance to environmental disturbance, etc. Further studies are needed to offer a unique criteria system to judge the working conditions of various algorithms, which is helpful for their application in practice.

(5) Safety verification is often overlooked when proposing a new method.

Many collision avoidance methods are working properly in designed simulations and case-based testing, but they lack systematic verifications. For instance, some collision prevention techniques utilized linearization techniques to simplify the problem, but this inevitably includes errors. Some algorithms use the discrete trajectories of the TS to check collision at each time, which does not guarantee the safety of the ship between time slots, e.g., MPC-CA, etc.

Instead of checking a finite set of scenarios in simulators, a framework that provides analytical proofs of safety is needed [165]. Some formal verification methods developed for autonomous car/aircraft can be introduced in the future, such as reachability analysis [166], funnel libraries [167], etc.

(6) The balance between effectiveness and efficiency of the methods should be considered.

The balance between efficiency and effectiveness is still less focused in the existing studies. These two benchmarks sometimes are conflicting, and researchers usually sacrifice the effectiveness to increase efficiency. For instance, some methods only check several controls instead of searching the whole control space, e.g., LGS, DIO, etc.; some researchers assume the controls will last for the whole prediction horizontal., e.g., DD, VO, DW, etc. These studies simplify the method for efficiency, but they do not provide a discussion of the influence of simplification on effectiveness.

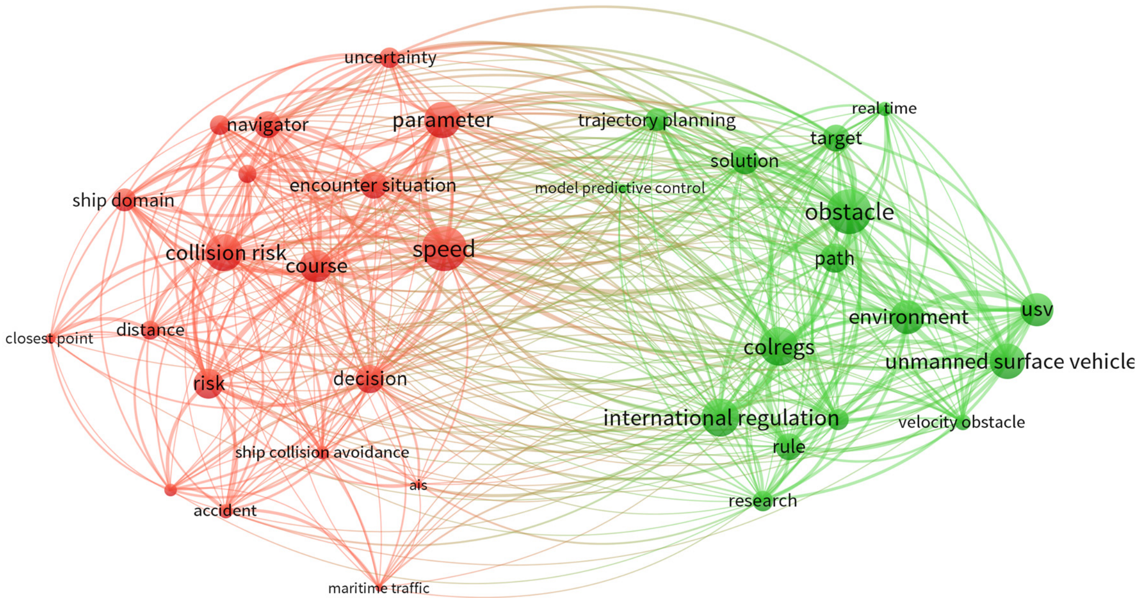
2.6.2 The word clouds of existing studies

Section 2.4 and Section 2.5 show that the focuses of the manned-ship study and the unmanned-ship study are slightly different. Most of the studies supporting manned ships were more interested in constructing a collision risk measure for conflict detection, and these studies usually left conflict resolution to the human operators on board, e.g., OOWs. On the other hand, the unmanned-ship study employed relatively simple methods to detect collision dangers (e.g., CPA, safety distance, etc.) and paid more attention to developing algorithms to find collision-free solutions.

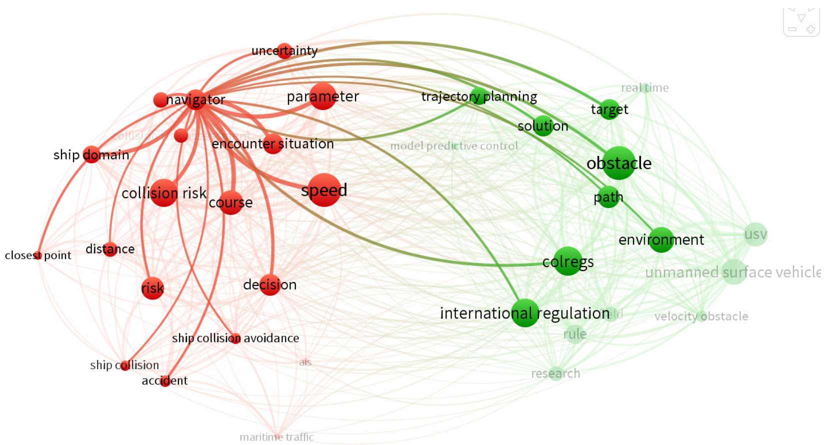
A more intuitive demonstration of these difference has been presented in Figure 2.12, where the occurrence of keywords is presented. In Figure 2.12, panel (1) shows the entire word clouds of collected articles; panel (2) shows the cloud associating to supporting navigators, i.e., the manned-ship study; panel (3) shows the cloud represents the unmanned-ship study.

In Figure 2.12 (1) and (2), it is observed that the manned-ship study (on the left-hand side of the figure) are more related to collision risk assessment instead of finding a collision-free path. The keyword “navigator” is strongly connected to risk-related terms, such as “risk”, “collision risk”, “ship domain”, “closest point”, etc., that includes some popular methods for conflict detection. Nevertheless, this could of words is less related with “model predictive control”, “USV”, etc.

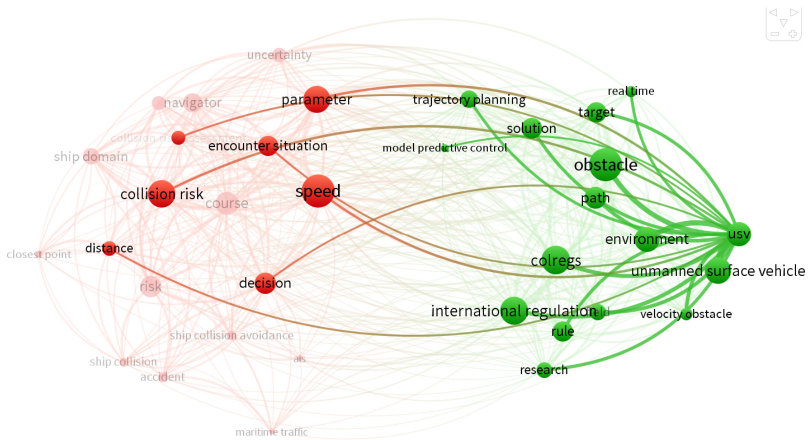
In Figure 2.12 (1) and (3), it is observed that the word cloud of the unmanned-ship study (i.e. the right-hand side of the figures) is apart from the word cloud of collision risk assessment, which is strongly connected to some words associating with collision resolution, e.g. “model predictive control”, “velocity obstacle”, etc. It implies only that a few unmanned-ship studies focused on collision risk and most of them are interested in finding a collision-free path.



(1) The entire word clouds[9]



(2) A word cluster serving for navigators in manned-ship studies



(3) A word cluster serving for unmanned-ship studies

Figure 2.12 Word clouds of literature working on ship collision avoidance.

[9] These word maps are generated via VOS viewer version 1.6.10.

In brief, it is observed that the focuses of the manned-ship research and the unmanned-ship research are different, one is on the conflict detection, and the other is on the conflict resolution. Based on the generic framework found in Section 2.2, these two groups of studies have great potential to be complementary.

2.6.3 The road to the autonomous shipping

As it showed in the last sub-section, the studies of the manned ship and the unmanned ship may complement one another, while awareness is still needed about some remaining gaps in the road to the autonomous shipping.

There are many ways to categorize control levels of the ship from the manned mode to the autonomous mode that shows the road to autonomous ship. A notable one is shown in the [168] that contains six levels of control, which ranges from the fully manual mode to fully autonomous mode. Inspired by this categorization (from the [168] and [169]) and the categorization of MASS (presented in Section 1.2), a similar six levels of control is presented, which breaks down the task of collision avoidance in different levels, see table 2.6.

Level 0 refers to a situation that no machine is involved in collision avoidance. Since most of the ships are requested to equip certain navigational assistance system on board, e.g., Integrated Navigation System, it is considered that this level has been passed.

Level 1 is the main scope of existing studies for the manned ship. These studies offered various supporting tools for conflict detection, i.e., ARPA, collision alerts system, etc. The aim of these studies is training a navigational assistance system which supporting the OOWs to identify collision dangers and to be aware of approaching dangers in time.

Level 2 expands the function of the assistance systems with conflict resolution. A few researchers have presented some prototypes on this theme, e.g., early versions of CTPA, CDS, etc. However, these prototypes usually ignore the ship's dynamics and used the holonomic model in collision prevention. As a result, the prototypes do not work well in close range. Here, the studies for ASV might provide some good references, e.g., the design of two-level controllers.

Level 3 introduces a deeper interaction between human and machine, which is seldom discussed in the existing literature. Many collision prevention algorithms are proposed to find one solution to the ship, regardless of the interaction between human and machine. For instance, how to present the solution in a way that human can easily understand and implement; how to support human to modify the collision actions without leading to a worse situation; how to validate the safety of human's choices, etc. This level is a critical step for improving the autonomy of the ships, and it is also a strong reason to convince human to trust the machines.

Level 4 requires the machine to be aware of emergencies in which the machine might not guarantee the safety of the ship. To achieve this, it requires researchers to test the extreme conditions of collision prevention algorithms, i.e., safety verification. These studies are not yet included in most collision prevention studies in maritime research.

Level 5 is the fully autonomous ships. It has received numerous attention from researchers and highlighted by societies. However, to achieve a fully autonomous system needs long-term developments. The challenges are the uncertainties of model and parameters. Moreover, how to fully comply with the various regulations in complicate scenarios is also an open question. Some studies in the manned ships might offer a line of thinking to help the ASVs to be rule-compliant, i.e., to incorporate experts' judgments in collision prevention.

From Level 1 to Level 5, the interactions between human operators and the collision avoidance system are gradually increasing. Level 2, 3, and 4 ask the machine contains more functions than conflict detection, specifically, supporting the OOW find one collision-free solution, checking the safety of the chosen solution by the OOW, eliminating unsafe solutions, supporting the human to understand the solution found by the machine, etc. These improvements are essential for testing the reliability of autonomous systems, increasing the trust between the human and the machine, and reducing the workload of the human. Thus, Level 2-4 would be the key steps to the autonomous era, which are less focused nor discussed.

In brief, propelling the autonomous shipping is not only continuing the existing studies on manned ships and unmanned ships (Level 1 and Level 5), but also filling the gaps between them (Level 2-4), specifically, making the unmanned ships user-friendly for human operators, exploring more functions in the existing manned ships, etc.

Table 2.6 Six levels of controls towards the unmanned ship w.r.t. collision avoidance

Level	**Implications in Collision Avoidance (CA)**	**MASS Types**	
Level 0	No machine is involved and the human fully takes responsibility to detect dangers and take evasive actions.	-	-
Level 1	The human directly controls the ship and machines offer certain service in conflict detection.	I	Human on board.
Level 2	The human directly controls the ship and machines offer supports both in detection and resolution, i.e., available solutions and validate chosen solutions.	I/II	Human in the offshore center & on board.
Level 3	Machines operate the ship under the monitoring of a human, which support the human to understand the choice of the solutions. The human can indirectly control the ship via machines or directly controls the ship via the on-board operators.	II/III	Humans in the offshore center or on board
Level 4	Machines can control the ship independently while it informs the human and sends an alarm when it in an emergency issue. Then, the human can indirectly control the ship via machines.	III	Humans in the offshore center
Level 5	Machines control the ship autonomously and humans cannot direct or indirect control the ship during each voyage.	IV	Humans in the offshore center

2.7 Conclusions

This chapter provides a comprehensive overview of the techniques used for ship collision avoidance both for manned ships and unmanned ships. The limitations addressed by existing review paper have been updated, some new challenges are highlighted, and the research tendencies are discussed. Additionally, another main task of this chapter is answering the research question addressed in Section 1.3: *What techniques have been developed for collision* *avoidance at sea and what research gaps can be explored in a generic framework?* The answers are concluded as follows:

- To answer these questions, a generic framework of collision avoidance, which contains five modules: “Observer”, “Motion Prediction”, “Conflict Detection”, “Conflict Resolution”, and “Controller”. Then, the developments of methods used in three key modules are analyzed. They are “Motion prediction”, “Conflict Detection”, and “Conflict Resolution”. (Section 2.2)

- Motion prediction relies on the formulation of ship dynamics. To predict the trajectory of the own-ship, researchers assumed the ship’s dynamics is known and the prediction is simplified as solving an ODE equation. In more practical cases, the dynamics of the own-ship or target-ship contain large uncertainties. Thus, researchers have to use a simplified model and the observed trajectory for prediction. Recently, maneuvers estimation using historical traffic data is introduced to improve the prediction quality. Additionally, some studies even accepted communications between ships to improve the quality of prediction. (Section 2.3)

- Conflict detection is the focus of conventional maritime studies, which is mainly interested in supporting human on board. A core of conflict detection is a collision risk measurement that is charged for three tasks: identifying dangers; setting off warnings for caution; setting off warnings for evasions. It is found that most risk measures cannot be used for the last task, which is discussed in Section 2.6.1. Firstly, many risk measures are independent of conflict resolution. Secondly, many risk measures mainly work in a pairwise encounter scenario. (Section 2.4)

- Conflict resolution gains a remarkable improvement in recent years due to the development of studies on autonomous ships. Six groups of collision prevention methods have been identified from the literature. The advantages and disadvantages of these methods are concluded in Section 2.5.2. Two main challenge includes ignorance of the ship dynamics and regulations are two common limitations. (Section 2.5)

- The studies for the manned ship and unmanned ship might complementary one another, where one focuses on conflict detection, and the other focuses on conflict resolution. However, developing various levels of autonomous ships require not only better conflict detection and conflict resolution modules but also a user-friendly design that facilitates various interactions between human operators and the autonomous system, which is less discussed in the literature. (Section 2.6)

This dissertation pays special attention to developing a Collision Avoidance System incorporating human-machine interaction that fills some of the gap addressed in above. It aims at handling the following challenges: firstly, the development of a novel risk measure which is suitable for triggering evasive actions in multiple-encounter scenarios; secondly, the development of a collision-avoidance system which is suitable for supporting manned ships and the unmanned ships in achieving collision avoidance. These challenges are addressed in the following chapters.

Chapter 3 Framework of Human-Machine Interaction oriented Collision Avoidance Systems

The review in the previous chapter reveals that the interactions between human operators and machines in collision avoidance are essential for developing autonomous systems at sea, but they are rarely discussed. Moreover, human intelligence and machine intelligence are complementary, where the first one is good at interpreting navigational regulations, while the other has the advantage of computing powers. Thus, it is necessary and promising to connect human operators and machines. This chapter is devoted to developing a framework of Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS). The framework is to help human operators and machines share their intelligence and powers in preventing collisions.

The structure of this chapter is assigned as follows: Section 3.1 introduces the role of Human-Machine Interactions (HMIs) in various control modes of autonomous ships; Section 3.2 explains the motivations of supporting HMIs in collision avoidances; Section 3.3 addresses HMIs in three key modules of CASs; a framework of the HMI-CAS is proposed in Section 3.4., followed by conclusions in Section 3.5.

Acknowledgment The content of this chapter is based on the submitted paper to the journal:

Y. Huang, L. Chen, R.R. Negenborn, P.H.A.J.M. van Gelder. “A HMI-CAS for human-machine cooperation during collision avoidance”. (Submitted)

3.1 Introduction

HMIs play important roles in the transition towards the autonomous era. In particular, the interactions between humans and machines become more and more intensive as the autonomy level increases. Even for a fully autonomous ship, the human-machine interactions are remaining since the machine still performs the task assigned by human operators. A categorization of control modes is presented to help readers build a clear picture of the role of HMIs in different autonomy levels.

There are many ways and standards to categorize autonomous ships and autonomy levels. The International Maritime Organization (IMO) proposed the Maritime Autonomous Surface Ship (MASS) categorization that divides the autonomous ships into four types based on the answers of the following two questions: *where* the control happens and by *whom*. For instance, the ship is controlled by a human on board (Type I), remotely controlled with/without a human on board (Type II/III), or automatically controlled by a machine (Type IV). Another categorization presented in Chapter 2 is based on *how* the control is made, which consists of six levels [168]. In early stage, Sheridan and Parasuraman [169] proposed eight degrees of automation that assign more levels for different interaction modes.

Based on the interpretation of these categorizations, the categorizations of control modes and autonomous ships are presented in Table 3.1 and Table 3.2. In the tables, four types of autonomous ships defined by the IMO are used, together with the six control modes that are similar to the six levels of control presented in [168].

First of all, the three agents that are frequently quoted need clear clarifications:

- "**Human**" refers to captains, seafarers, human operators on board or in offshore control centers.
- "**Machine**" means automation systems on board. The system is named as Integrated Navigation System (INS) or Integrated Bridge System (IBS) in manned ships and Guidance, Navigation and Control (GNC) system in unmanned ships.
- "**Ship**" refers to on-board hardware including ship hull, rudders, propellers, servo motors, etc.

The first three control modes (Control Mode 0 to Control Mode 2) are presented in Table 3.1, in which the control of the ship is still on human's side. The differences between these modes are how human operators observe the surrounding information:

In Control Mode 0, the observation relies on human operators, only;
In Control Mode 1, the observation is supported by machines;
In Control Mode 2, the observation mainly relies on machines.

During collision avoidance, Control Mode 0 is not widely found because most ships are required to install various navigational equipment by international conventions, e.g., the International Convention for the Safety Of Life At Sea (SOLARS) [170]. Therefore, human operators are, more or less, supported by the machine in conventional ships, i.e., Control Mode 1. Correspondingly, the conventional ship is considered to be a Type I MASS. In Control Mode 2, human operators rely on machines to detect the environment, and the machines offer more services to the human operators compared to Control Mode 1, e.g., suggesting actions.

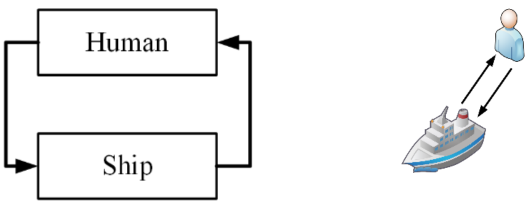
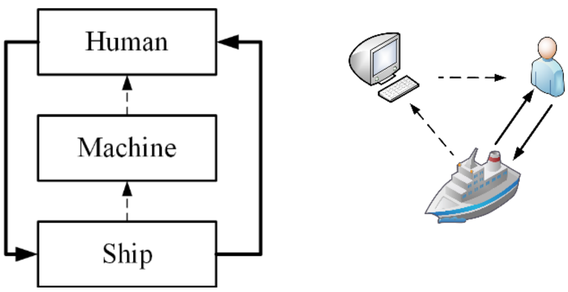
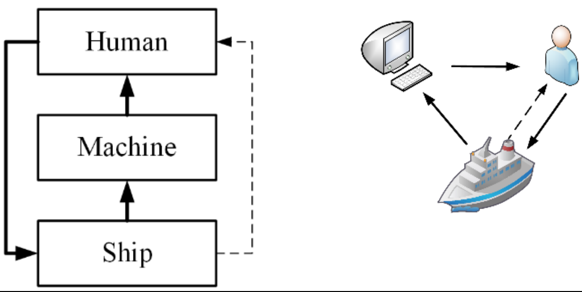
Nevertheless, observations by the human operators are still optional. This mode might appear in two types of MASS, i.e. the MASS Type I or Type II.

The subsequent modes (Mode 3 to Mode 5) are presented in Table 3.2, where the control of the ship is switched to machines. The difference is how the human operators influence the ship:

In Control Mode 3, human operators authorize machines to take actions;
In Control Mode 4, human operators supervise machines to take actions;
In Control Mode 5, operators cannot intervene but be informed during the voyage.

In Control Mode 3, human operators approve the collision-free solution selected by machines instead of directly making decisions on preventing collisions. However, in an emergency case, the human operators still can directly control the ship by changing rudder angles and propeller revolutions. This mode could be used in the MASS Type II or Type III. In Control Mode 4, machines can automatically avoid collisions if no interventions from human operators, and the operators have to use the machines to control the ship. This control mode might appear in a Type III MASS. Control Mode 5 is the final goal, which appears in a Type IV MASS. Machines fully control the ship, and human operators only receive information reported by the machines.

Table 3.1 Six modes of control and four types of MASS (the first 3 modes)

Mode	Human-Machine-Ship	Characteristics	MASS Types	
0		Control: Human Observe: Human	-	Human on board.
1		Control: Human Observe: Human or Machine (optional)	I	Human on board.
2		Control: Human (remote) Observe: Machine or Human(optional)	I/II	Human in offshore center & on board.

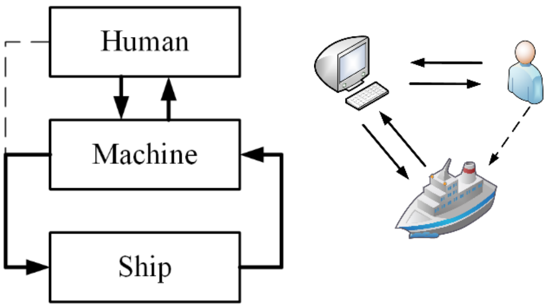
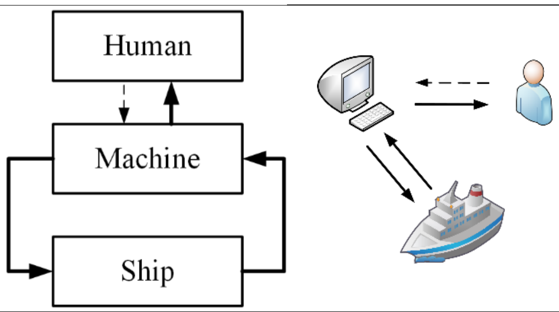
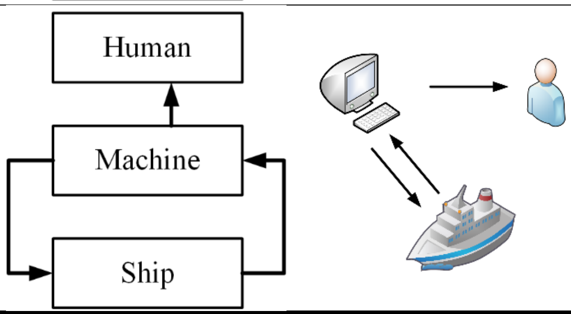
* Note: the dotted line means this data/information flow is optional; the solid one refers to the required flow.

From Table 3.1-3.2, it is clear that interactions between human operators and machines are indispensable in every control mode. For instance, in Control Mode 1 and Mode 2, the machines support the human operators by providing navigational information; in Control Mode 3 and Mode 4, the machines become irreplaceable, and the human operators have to interact with the machines to control the ships; in Control Mode 5, the interactions are reduced but still exist, in particular, the machines report the states of the ship and executes the voyage inputted by the human operators.

Moreover, good human-machine interactions are the key to promoting autonomous ships. Even if the fully autonomous ship (e.g., a Type IV MASS) is ready for use, the willingness of humans to adopt the technology is still questionable. The lack of public trust of automation systems is a main barrier, according to studies on driverless cars [12, 171]. In fact, a good human-machine interaction has a positive influence on building trust. Research shows that deeper interactions, such as providing an opportunity for human to observe how the automation works, benefit in developing greater trust [172].

In brief, studies on supporting HMIs for ship collision avoidance are necessary and essential for the transition of the conventional ships toward the autonomous ships. In the following sub-sections, HMIs during collision avoidance are focused, and a framework of HMI-CAS is proposed.

Table 3.2 Six modes of control and four types of MASS (the subsequent 3 modes)

Mode	Human-Machine-Ship	Characteristics	MASS Types	
3		Control: Machine or Human (optional) Observe: Machine or Human The machine is authorized by the human;	II/III	Humans in the offshore center or on board.
4		Control: Machine Observe: Machine The machine is supervised by the human	III	Humans in the offshore center.
5		Control: Machine Observe: Machine The machine informs the human	IV	Humans in the offshore center.

* Note: the dotted line means this data/information flow is optional; the solid one refers to the required flow.

3.2 Motivations of improving HMIs during collision avoidance

3.2.1 HMIs in different control modes

Generally, the contents of HMIs cover all forms of relationships between humans and machines, including three types: (1) the machines perform the humans' command or task; (2) the machines provide information to the humans; (3) the humans turn off/on the machines [169]. Guided by this categorization and the generic collision avoidance framework presented in Figure 2.2, seven types of HMIs during collision avoidance are identified: "turning off/on the system", "data collection", "trajectory prediction", "early warning", "solution advice", "automated solution execution", and "cooperation support". The meanings of these HMIs are presented in Table 3.3.

Each Control Mode in Table 3.1-3.2 might contain more than one type of HMIs. An overview of the HMIs in each control mode is shown in Table 3.4.

Control Mode 1 is a popular mode in existing manned ships, which allows human operators to turn off/on machines since the human operators are still on board. The machines in this mode can help to collect data, predict trajectories, and remind the operators about dangers based on predictions. In Control Mode 2, all functions in Control Mode 1 are kept, while advising a collision-free solution becomes necessary. In these modes, the machines lend computing power to the human operators and help them in detecting and avoiding dangers.

Table 3.3 Contents of HMIs during collision avoidance

HMIs	Content
Turning off/on the system	humans can turn on/off the systems during one voyage;
Data collection	systems can sense data and present to humans;
Trajectory prediction	systems can foresee the position of ships for humans;
Early warning	systems can find the potential dangers and send alarm;
Solution advice	systems can advise a conflict resolution to humans;
Automated solution execution	systems can automatically execute the solution;
Cooperation support*	systems can cooperate with humans in finding a collision-free solution.

*Note: "cooperation support" refers to the "machine" and "human" work together in finding a collision-free solution. The cooperation is mutual, in particular, the human can intervene the machine, and the machine validates the human's solutions.

Table 3.4 Overview of HMIs in different control modes

HMIs	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Turning off/on the system	●	●	◐	-	-
Data collection	●	●	●	●	-
Trajectory prediction	●	●	●	●	●
Early warning	●	●	●	●	-
Solution advice	◐	●	●	●	-
Automated solution execution	-	-	●	●	●
Cooperation support	-	-	◐	●	-

Note: The compulsory interaction is marked as "●", the optional one is marked as "◐", and "-" means that the interaction is not applicable.

In Control Mode 3 and Mode 4, machines can execute collision-free solutions automatically. Interactions between human operators and the machines become more frequent and more content-rich than in other modes. Subsequently, requirements on the machines increase since that the action selected by the machines needs to be authorized or supervised by the human operators. Firstly, the machines not only provide the information like that in Control Mode 1 and Mode 2 but also offer extra information for the human operators to understand the choices of the machines. Secondly, the machines need to contain the design that allows the human operators to influence the machines' decisions. For instance, if a human operator disagrees with a machine's solution, he/she is enabled to modify this solution. The difference between Control Mode 3 and Mode 4 is concluded as follows. In Control Mode 4, machines cannot be shut down since the ship is controlled via the machines only, while in Mode 3, "turning off/on the system" is optional since humans might work on board, such as a Type II MASS.

In Control Mode 5, interactions between humans and machines are reduced. Specifically, the machines only inform the operators and execute collision avoidance automatically.

3.2.2 HMI-related studies in the existing literature

In the literature, the seven types of HMIs received different attentions. According to Chapter 2, the existing studies are either focusing on Control Mode 1 that supports human in identifying dangers (i.e., "early warning") or Control Mode 5 that avoids collisions automatically (i.e., "Automated solution execution"). Other Control Modes and the relevant types of HMIs are less focused, e.g. "solution advice" and "cooperation support". An overview of the focuses on the existing studies on manned and unmanned ships is presented in Table 3.5.

Manned-ship studies offered many techniques for supporting operators to detect collision dangers, i.e., "early warning". In the studies, machines take the responsibility of calculating risk and offering alerts, the operators can modify thresholds of the alerts and turn off/on the machines. These studies, of course, are fundamental for supporting Control Mode 1, while other types of HMIs and other modes of Control Modes are out of discussions.

Unmanned-ship studies, on the other hand, concentrate on solving collision problem automatically by the machines only, which is considered to achieve "Automated execution of solution". Besides, some of the studies are enable to output the predicted collision-free trajectory that facilitates the human operators to understand the machine's decision. However, cooperation between humans and machines is usually out of their scopes.

Table 3.5 Overview of HMIs in manned-ship and unmanned-ship studies

HMIs	Studies on manned ships	Studies on unmanned ships
Turning off/on the system	-	-
Data collection	●	●
Trajectory prediction	●	●
Early warning	●	-
Solution advice	-	-
Automated solution execution	-	●
Cooperation support	-	-

Note: "-" means the interaction is not widely discussed in the group; "●" means relevant interaction is developed;

In summary, the existing studies on ship collision avoidance are more focusing on "early warning", "automatic executions of resolution" and "trajectory prediction", while "resolution advice" and "human-machine cooperation support" are rarely discussed.

3.2.3 A bridge between human operators and automation systems

The existing studies on manned ships or unmanned ships do not incorporate the cooperation between humans and machines in finding collision-free solutions. The human operators or the machines work independently, i.e., the decisions are either made by the human alone or the machine alone. Knowledge between the human and the machine is not shared.

In Figure 3.1, the system integrating two navigation modes is presented: one mode with humans on board (manual) and the other without humans (the unmanned loop), which simulates the case that the GNC system is directly embed into a conventional ship. One switch is employed to change the control between two loops.

In the unmanned loop, the Guidance system is in charge of finding a collision-free solution, and the Control system is designed to track collision-free solutions. Since humans are not considered in this loop, the outputs of the Guidance system or the Control system are usually difficult for the humans to read, to intervene, and to implement. For instance, if the Guidance system outputs desired forces in each direction, how to allocate these forces to steers is challenging for human operators. Alternatively, the system might offer a collision-free trajectory or a set of steering commands that are readable for human. However, steering the ship to each waypoint or following each planned steer sticking to a planned schedule is also challenging for the human operators.

If a human operator wants to intervene in this automatic collision avoidance process, the operator needs to switch to the manual loop. Then, the operator has to analyze the encounters again, to find a possible collision-free solution, and to implement the solution by himself/herself. In this hybrid system, the interaction between humans and machines is little, and there is a lack of information exchange.

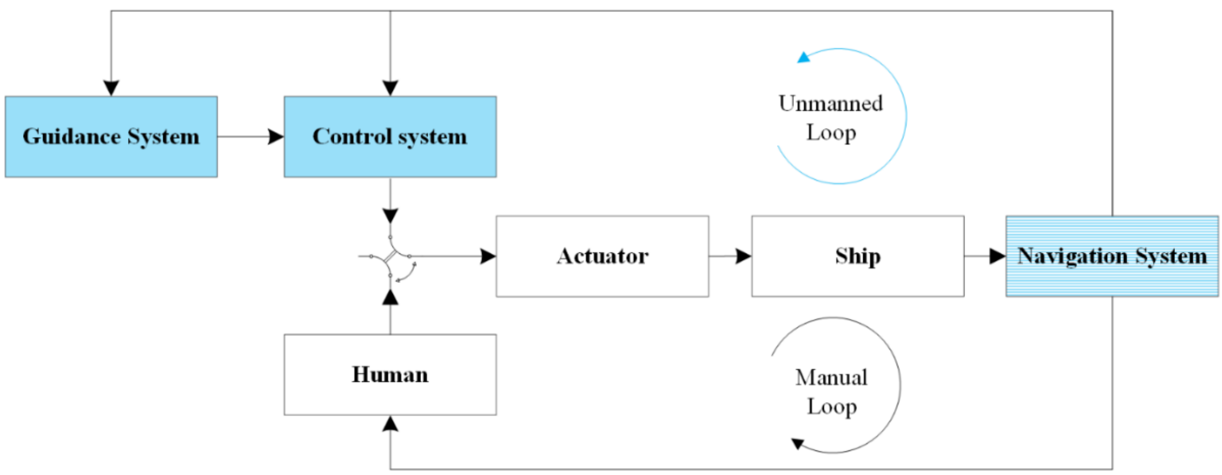


Figure 3.1 Representation of an integration of the manned and unmanned control loops.

Neither of these loops is perfect. If the manual loop is reliable, there would not be room for developing autonomous systems. Specifically, from numerous accident reports, people found that “too late actions” or “no actions” are two main causes of collision accidents. In fact, traditional navigational assistance systems lack the function of suggesting collision-free solutions [9], which is a strength of existing automation systems. One the other hand, the unmanned loop is also not perfect. Teaching the automation system to understand regulations is still an open question [20], where human knowledge is indispensable. Therefore, a connection between these loops would have a “win-win” situation, where an HMI system is needed.

In summary, the existing studies on manned or unmanned ships do not consider the cooperation between humans and machines in conflict resolution. Moreover, neither the humans or the machines are reliable enough to perform collision avoidance in various situations. Additionally, advantages of manual and automation systems are highly complementary. Thus, there is a huge potential of connecting manual and automation systems. The connection not only helps the automation systems gain the knowledge from human operators to achieve rule-compliant actions, but also helps the human operators take timely correct operations. From this perspective, HMIs are performed as the bridge combining the advantages of the human and the machine in preventing ship collisions.

3.3 Incorporating HMIs in the Collision Avoidance System (CAS)

Figure 3.2 illustrates the information flow in the designed CAS incorporating the seven types of HMIs, with the help of the generic CAS framework presented in Section 2.2.2.

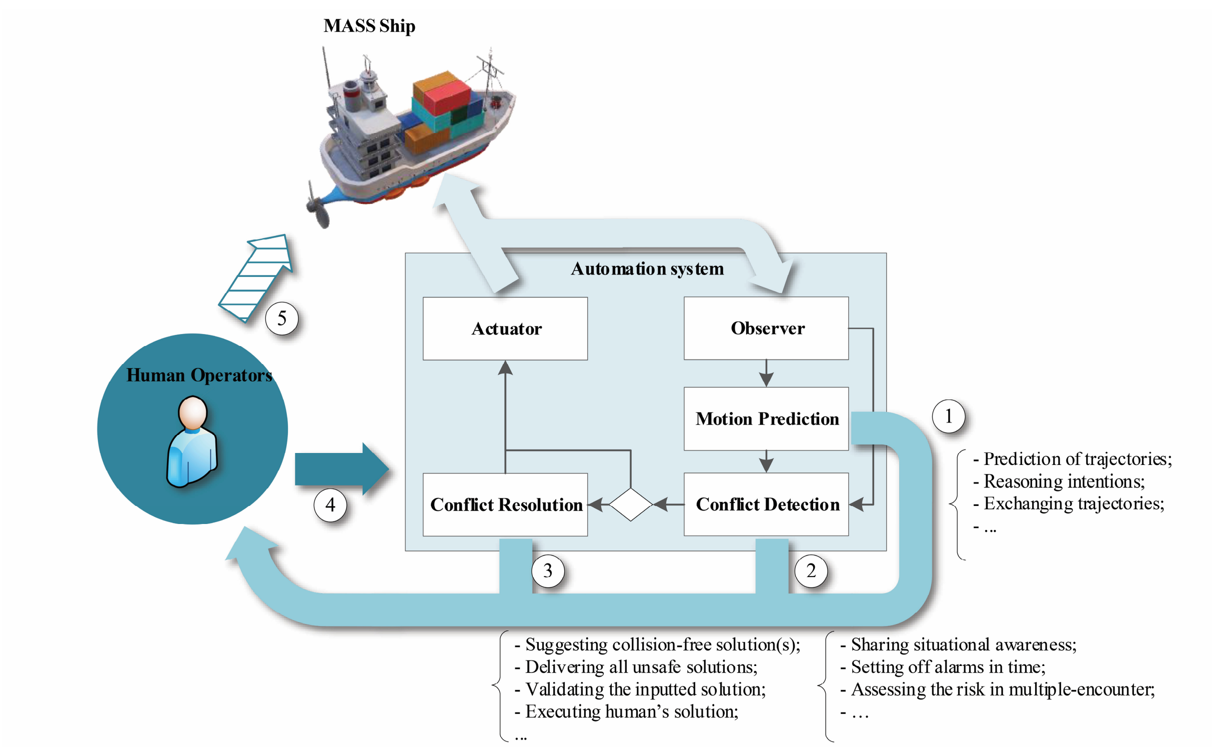


Figure 3.2 Illustration of human-machine interactions during collision avoidance.

The information flow among three key modules, labeled ①, ②, and ③ in Figure 3.2, is used for supporting three types of the HMIs, i.e., "trajectory prediction", "early warning", "solution advice". Moreover, a human operator can approve the machine's suggestion and let the machine to execute the solution automatically, or the human can intervene in the automated collision avoidance by feedbacks to "Conflict Resolution" module (labeled ④), which help to achieve "Automated solution execution" and "cooperation support". Finally, the human operator even can turn off the system and directly maneuver the ship according to the received information from the machine (labeled ⑤).

In the following sub-sections, the HMIs in each module are explained in detail. The information delivered to human operators in "Motion Prediction" and "Conflict Detection" is addressed in Section 3.3.1, and the HMIs in "Conflict Resolution" are explained in Section 3.3.2.

3.3.1 One-way information flow delivering to human operators

The cooperation of human operators and machines mainly takes place in the "conflict resolution" module, while information supports from previous modules are necessary for the cooperation. "Prediction" is in charge of obtaining trajectories of the ships (the OS and the TS), which is a foundation for automation in various control modes. The following items are the possible functions that the machines support the operators, which are marked as ① in Figure 3.2:

- Predicting the trajectory of the OS with a bounded range that considers the dynamics of the OS and uncertainties on parameters and on environmental disturbances;
- Predicting the trajectory of the TS with a bounded confidence range that considers uncertainties on its maneuvers, dynamics, environmental disturbance, etc.;
- Reasoning the intention of the TSs according to their trajectories, traffic patterns, navigational regulations, etc.;
- Exchanging predicted trajectories with other ships;
- …

"Conflict detection" plays an important role in early alarms and triggering evasive actions, which serves the operations in Control Model 1 to 4. Many current alert systems are based on CPA methods, which usually fall in two types of failures: alarms given too late and nuisance alarms. Moreover, the CPA methods cannot handle multiple-encounters. Thus, new functions are expected (see ② in Figure 3.2), including, but is not limited to:

- Setting off an alarm that reflects experts' knowledge for sharing situational awareness between experts and human operators ("soft boundary");
- Setting off an alarm that links with the conflict resolution for the OS, where the alarm can set a warning before the collision becomes inevitable ("hard boundary");
- Setting off an alarm properly in a multiple-encounter case;
- …

The first of alarm, serving as a "soft-boundary" between safety and collisions, is widely discussed in the existing conflict detection studies, see [71], [80], etc. The violation of this "soft-boundary" is considered to be unacceptable to most experts, but it does not mean a collision is unavoidable. The "hard boundary", however, shows a real boundary between safety and collisions. In particular, the violation means unavoidable collisions.

3.3.2 Bi-directional information flow in conflict resolution

"Conflict resolution" is delegated to find a collision-free solution. Interactions between human operators and the machines in this module are the keys to achieve Control Mode 2-4. The information delivered to the human operators includes, but is not limited to (see ③ in Figure 3.2):

(1) One collision-free solution, noted as u;
(2) One best solution that minimizes the utility function, noted as u^*;
(3) A finite number of safe solutions noted as $\mathbb{U}=\{u_1,u_1,\dots,u_n\}$;
(4) A closed region of collision-free solutions noted as $\mathcal{U}=\{u \mid f(u)\leq 0\}$;
(5) All closed regions of unsafe solutions noted as $\mathrm{K}=\{\bigcup_i \bar{\mathcal{U}}_i\}$.

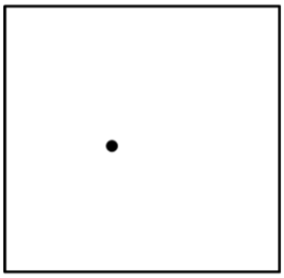
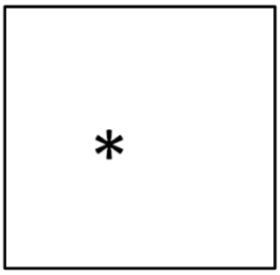
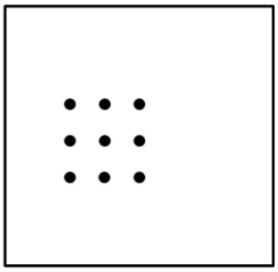
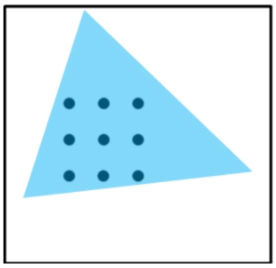
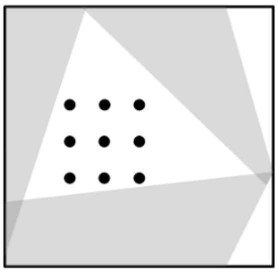
Illustrations of these types of information are shown in Table 3.6, where the solution space of a ship is presented (each point in this space represents one maneuver for the ship). The black dot in the first row represents the collision-free solution that is found by the machine and delivered to human operators. In the second row, the "*" mark means that the machine offers one optimal collision-free solution. In the third row, the dots are the feasible collision-free solutions that the machine can find. In the fourth row, a sub-space of the solution space containing collision-free solutions is colored in blue. In the fifth row, instead of finding a collision-free set, the machine identifies all the solution sub-spaces leading to the collision, colored in grey.

Human operators then can utilize this information for analyzing the encounter situation and do the following operations (i.e., ④ in Figure 3.2):

a. Switching to manual mode when human operators prefer to steer the ship by themselves or the solution found by the machine is not fully satisfying.
b. Accepting the solution when operators believe the solution is proper;
c. Changing the solution by modifying the utility function when the operator does not satisfy the selected "best" solution by machine;
d. Picking up one solution in given finite solutions $\mathbb{U}$ when the machine does not pick up any solution to human operators or the chosen one is not satisfied. (The machine shows several solution candidates);
e. Picking up one solution in given collision-free set $\mathcal{U}$ when the machine does not pick up any solution to human operators or the chosen one is not satisfied. (The machine shows a sub-space of feasible solutions);
f. Validating the safety of inputted solution by human operators when the machine does not pick up any solution or the chosen one is not satisfied. (The machine shows unsafe solutions)

The solution forms and their matching functions for supporting the human operators are presented in Table 3.6. One solution form might fulfill multiple functions. In many cases, a combination of different solution forms are needed to achieve more HMI types. For instance, a combination of form (2) and (3) provides an optimal solution and alternative solutions to operators at the same time.

Table 3.6 Overview of different forms of solution and relevant interactions

Service from machine		Operations that human can do
(1) A feasible solution: u.		a. Switch to manual mode; b. Accept the solution;
(2) An optimal solution: u^*.		a. Switch to manual mode; b. Accept the optimal solution; c. Modify utility function
(3) Finite feasible solutions: $\mathbb{U}$.		a. Switch to manual mode; d. Pick up one solution in $\mathbb{U}$.
(4) A closed region of feasible solutions: $\boldsymbol{\mathcal{U}}$.		a. Switch to manual mode; e. Freely choose one solution in $\boldsymbol{\mathcal{U}}$
(5) All closed regions of dangerous solutions: $\mathbf{K}$.		a. Switch to manual mode; e. Freely choose one solution in $\boldsymbol{\mathcal{U}}$; f. Validate arbitrary solution inputted by human using K.

* These figures demonstrate the solution space of the ship. Each dot in this space represents one control inputs to the ship. The region in blue is a collision-free sub-space, and the region in grey is a sub-space that leads to collisions.

3.4 Human-Machine Interaction oriented CAS (HMI-CAS)

3.4.1 Assumptions and focuses

As mentioned in Section 3.3, the information from the machine to human operators is coming from the "Motion Prediction" module, the "Conflict Detection" module, and "the Conflict Resolution" module. However, the "Prediction" module is out of our scope because the HMIs in this module is unidirectional. More contents of trajectory prediction can be found in [63]. Thus, this dissertation accepts the following assumption.

Assumption: Each ship will broadcast its predicted trajectory to neighboring ships or each ship can predict the trajectories of other ships precisely.

In brief, this dissertation is more interested in the interactions labeled ②, ③, and ④ in Figure 3.2, which specifically supports cooperation between humans and machines in collision avoidance. Two types of services are focused on:

Focus 1: Information for "Conflict Detection", e.g., collision candidates, collision risk, time for evasive actions;

Focus 2: Cooperation for "Conflict Resolution", e.g., selected/suggested a collision-free solution, the unsafe/safe set of maneuvers, the safety of inputted solution by humans, intervention of automated collision avoidance, etc.

Different from the hybrid system presented in Figure 3.1, this dissertation aims at adding an interaction between the automation system and human, specifically between the Guidance system and human operators, i.e., the red arrows shown in Figure 3.3. The Guidance system provides information to human and the human can send feedback to the Guidance system, e.g., authorization, intervention, etc.

To incarnate the connection between human operators and the Guidance system, an interface is introduced, that supports information exchanges. An illustration of this idea is shown in Figure 3.4. The blue arrow shows the information flow in the CAS, which delivers information to human operators. The green arrows represent the inputted information by humans, that feeds the humans' decisions back to the machine and influences the behavior of the ship.

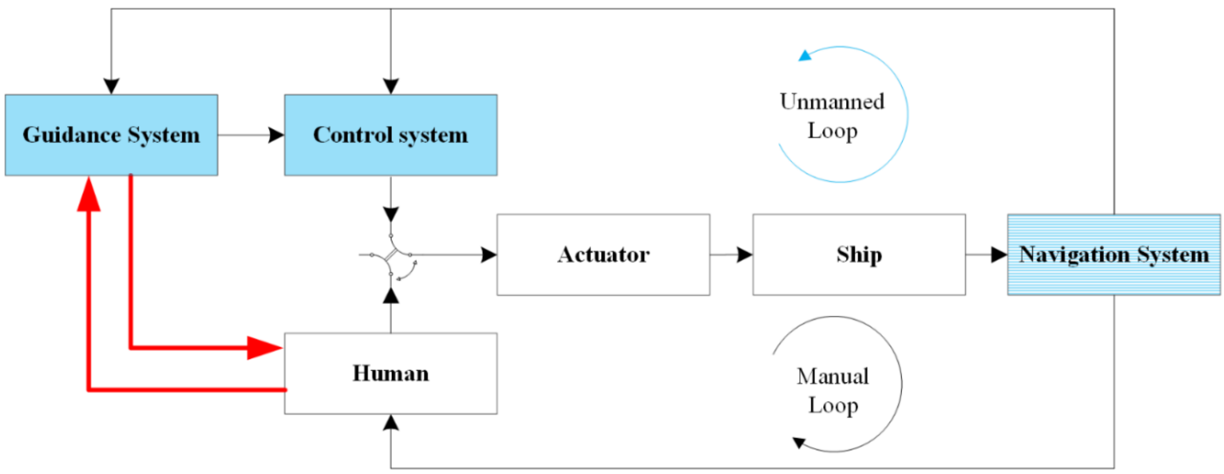


Figure 3.3 Representation of control loop within the proposed HMI-CAS.

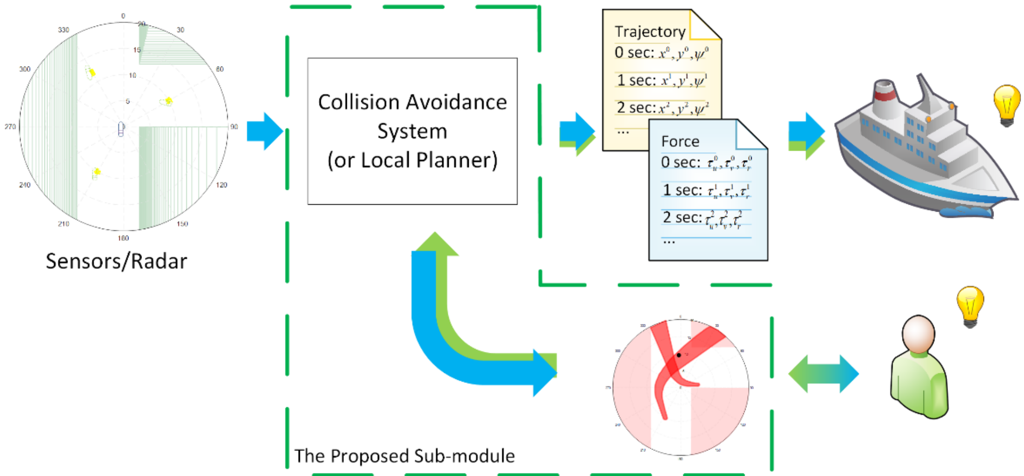


Figure 3.4 Illustration of the proposed HMI-CAS.

The CAS and interface in Figure 3.4 compose the collision avoidance module in the Guidance system. The entire GNC system with the designed CAS and interface is named as Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS).

The HMI-CAS can be applied to different types of MASS where human is involved. Although the manual mode is reserved in the HMI-CAS, it is expected that the human operator can get used to controlling the ship via the designed HMI-CAS. As time goes on, the operator builds trust with this new system. The operator, then, do not need to stay on board for directly controlling the ship, i.e., the human operator can intervene/interact with the ship in offshore control centers.

3.4.2 Framework of the HMI-CAS

An abstract representation of HMI-CAS is shown in Figure 3.5. The proposed HMI-CAS has two modes: manual mode and autonomous mode.

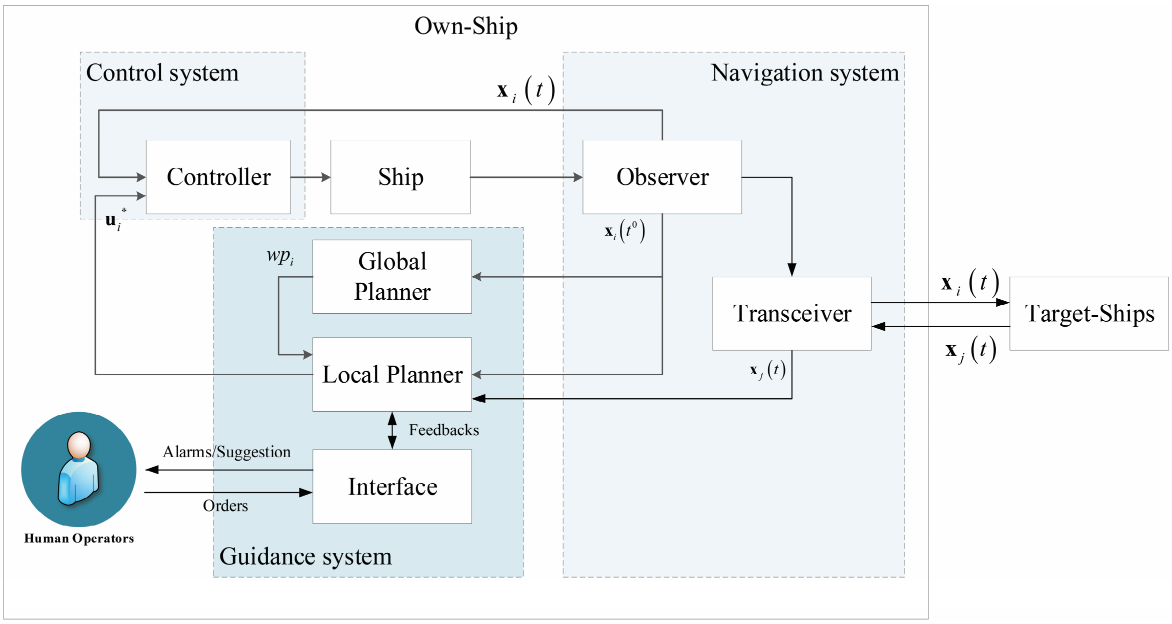


Figure 3.5 Abstract representation of the proposed HMI-CAS.

Note: sub-script *i* represents the states related to the own-ship; sub-script *j* is related to the target-ship.

When there is no human intervention, the OS follows a set of waypoints and takes evasive actions automatically, i.e., the autonomous mode. Thus, the framework of this system is based on the GNC system. Three basic systems are included: Guidance system, Navigation system, and Control system. However, to support the interactions between human operators and the GNC system, an interface is introduced.

The Navigation system contains sensors and transceivers that allow the ship to sense the surrounding environment and to communicate with other ships. The obtained information is sent to the Guidance system.

The Guidance system is performing as a “brain” that makes decisions according to the received information. Moreover, this electronic “brain” is allowed to interact with humans via the designed interface.

The Control system contains a controller which implements the inputted solutions from the Guidance system and outputs relevant commands to actuators, i.e., propellers and rudders.

3.4.3 Key modules in the Guidance system

Three main modules are designed in the Guidance system, i.e., a Global Planner, a Local Motion Planner, and an Interface.

Global Planner generates a planned waypoint which guides the ship to the destination, noted as *wp*. This waypoint is generated by path planning algorithms or human operators.

Local Planner generates a collision-free solution considering the observed information, planned waypoints, and the feedbacks from human operators, etc., noted as $\mathbf{u}^*$.

Interface has two main functions. One is to present information for human operators, e.g., a set of solutions leading to collisions, a set of collision-free solutions, and a selected solution to human operators. The other is to collect and send the orders from operators to the local motion planner. The orders could be an alternative solution, a command to stop/continue the existing mode, etc.

The structures of the modules are shown in Figures 3.6-3.8.

(1) Global Planner

Global Planner contains a path planning module in which various deliberate algorithms can be used, e.g., A* [22], Theta* [173], Dijkstra, Fast marching method [174], Potential Field [44], etc. An overview of these algorithms is shown in [20]. A common ground of these methods is a known roadmap. Some algorithms also need predefined cost functions in finding optimal paths. Alternatively, instead of using path planning algorithms, human operators also can directly input the desired path that usually is indicated by waypoints. Given the inputted/optimized path, this module also needs to find out which waypoint is activated, i.e., the waypoint that the ship is heading to.

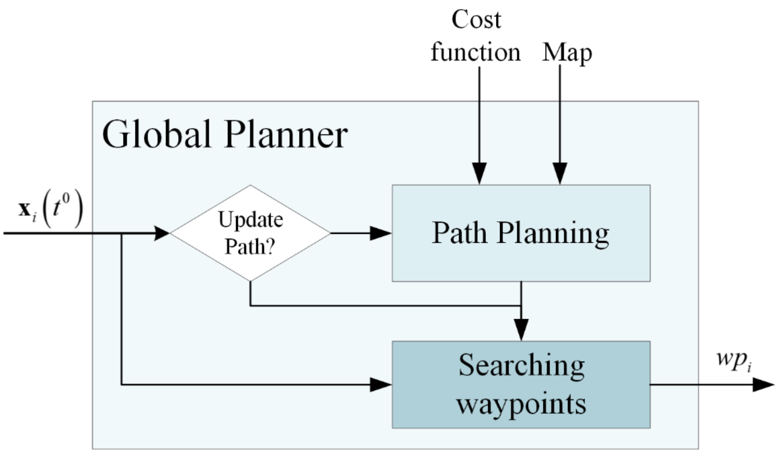


Figure 3.6 Representation of Global Planner Module.

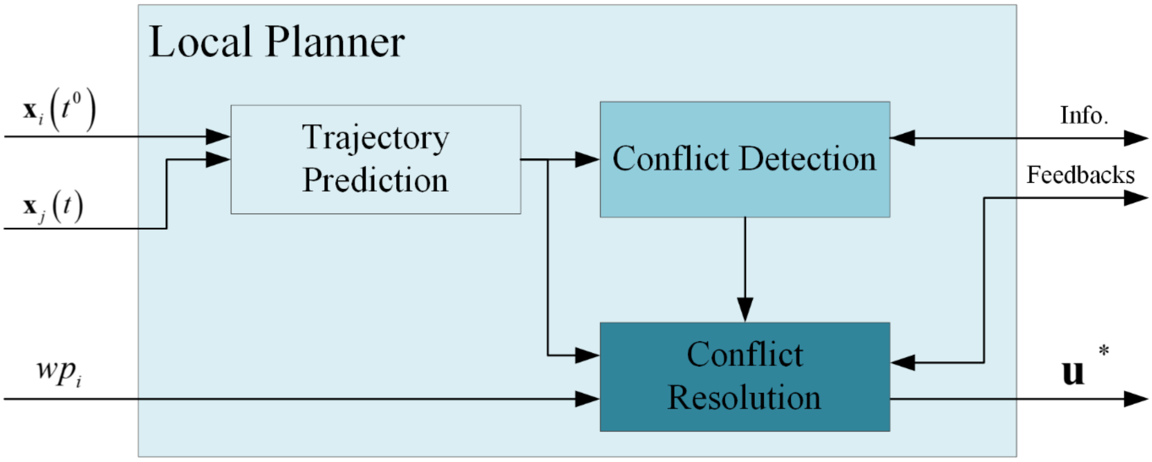


Figure 3.7 Representation of Local Planner Module.

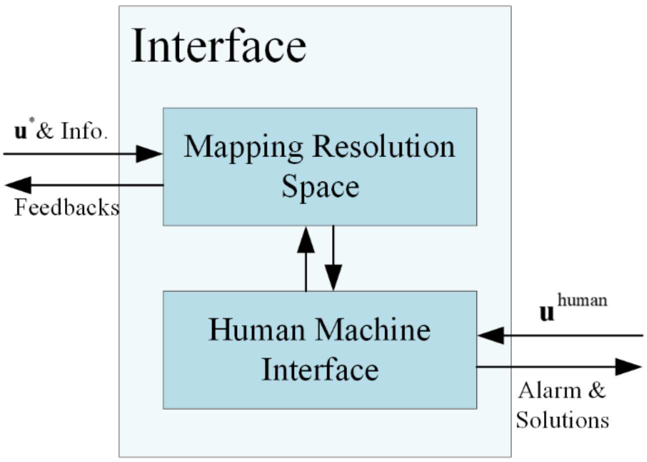


Figure 3.8 Representation of Interface Module.

(1) Local Planner

Local Planner is the focus of the Guidance system, which is in charge of finding a collision-free solution. It consists of three key processes, i.e., a "trajectory prediction", a "conflict detection", and a "conflict resolution". The "trajectory prediction" process outputs the predicted trajectories of the OS and the TS, i.e., $\mathbf{x}_i(t)$ and $\mathbf{x}_j(t)$. Based on these outcomes, the "conflict detection" calculates the relevant collision risk and triggers an alarm and "conflict resolution" if necessary. Lastly, the "conflict resolution" finds collision-free solutions according to the input information from "trajectory prediction", "conflict detection", and the feedbacks from human operators. The outputted information is delivered to the "Interface" module and the "Control" system. If no feedbacks are sent from the "Interface", the optimal solution will be executed by "Control" system.

(2) Interface

"Interface" is designed for connecting the human operators and the machines. It provides the operators with the supporting information from the Local Planner Module, and collects the feedbacks form the human operators.

The information from the machines to the human operators includes collision-free solutions, best solutions, dangerous solutions, etc. Since the inputted information from conflict resolution might not be directly readable by human operators, a mapping sub-module is designed to present the resolution spaces. The map, together with a solution selected by the machine, is projected in the human-machine interface. Human operators then can read the map, understand the selected solution of the machine, authorize the system to continue, intervene in the system, and find a new solution if necessary.

The information from human operators to the machine is the new solution selected by the operators, noted as $\mathbf{u}^{\text{human}}$. The new solution is sent back to the mapping sub-module, and then it is translated to the machine's language.

3.4.4 Requirements for achieving HMIs

It is expected that the HMI-CAS system supports human operators in the following aspects:

(1) to identify who have conflicts with the own-ship;
(2) to notice whether it is necessary to take actions;
(3) to show how the CAS system avoids the dangers;
(4) to be aware of what kind of operations are dangerous/safe;
(5) to inform whether the chosen solution by the human is safe/unsafe;
(6) to intervene in automated collision avoidance if necessary.

To meet these demands, the "Interface" in the HMI-CAS needs to be user-friendly. In particular, the interface facilitates the users to read the information, to understand the selected collision-free solutions by the machine, and to intervene in the conflict resolution. (supports ③ in Figure 3.2);

Moreover, since a ship might encounter with more than one ship in busy waters, the "Local Planner" in the HMI-CAS needs to handle multiple-encounter scenarios. The HMI-CAS needs not only to find a collision-free solution in the multiple-encounter situation but also to support conflict detection in this situations. (supports ② in Figure 3.2)

Lastly, this system needs to support various types of cooperation in conflict resolution. Thus, the "Conflict Resolution" in "Local Planner" module needs to offer various types of solutions for users. For instance, one collision-free solution, one best solution, a set of safe/unsafe solutions, etc. (supports ③ and ④ in Figure 3.2).

3.5 Conclusions

In this chapter, categories of autonomous ships and autonomy control levels are analyzed, which reveals huge demands on human-machine interactions (HMIs) in conventional ships, autonomous ships, and the transition towards autonomous ships. In particular, the interactions are indispensable in many Control Modes and different types of Maritime Autonomous Surface Ships (MASS). Additionally, in the transition towards autonomous ships, a good design of the system supporting HMIs benefits in building the trust of the system. These reasons motivate this study to work on a collision-avoidance system supporting HMIs.

This chapter answers the research question listed in Section 1.3, specifically *How can a framework of HMI-CAS be designed to support various modes of human-machine interactions during collision prevention?* The answers are concluded as follows:

- Seven types of HMIs during collision avoidance have been identified, i.e. , "early warning", "**solution advice**", "automated solution execution", "**cooperation support**", "trajectory prediction", "data collection", and "turning off/on the system".

- Two types of interactions, "solution advice" and "cooperation support", are less discussed in the literature, but they are important for developing Control Mode 2 to Mode 4 in autonomous ships. In fact, the existing studies do not incorporate the cooperation between human and machine in finding a collision-free solution, while the advantages of the human and the machine are complementary. In particular, the machine can help the human operator find a solution, and human can help machine find a rule-compliant solution. These two types of HMIs are the bridge that helps operators and machines share their knowledge and computing power.

- To support autonomous functions, the framework of the HMI-CAS remains the same as the Guidance, Navigation, and Control (GNC) system. Nevertheless, to facilitate information exchanges, an interface is needed. This interface is placed in the "Guidance" system, and it is used for information exchanging between the operators and the "Local Planner" module.

- The "Local Planner" module in the "Guidance" system contains three key sub-modules that are crucial for collision avoidance, namely "Prediction", "Conflict Detection", and "Conflict Resolution". In the "Prediction" module and "Conflict Detection" module, the machine offers information that supports the human operators to estimate the future states of the ships, potential dangers, and the soft/hard boundary of dangers. In the "Conflict Resolution" module, the machine can not only deliver a collision-free solution to human operators but also support the operator to take over control of the machine, e.g., offering finite collision-free solutions, identifying all unsafe solutions, etc.

In brief, the framework of the proposed HMI-CAS is on the basis of the GNC system with a new design of the "Guidance" system. The improved "Guidance" system contains a HMI interface that supports all forms of interactions presented in Table 3.3. The key of the interaction modes are "cooperation support", which requires the HMI-CAS to offer various forms of solutions and to be user-friendly. For instance, the HMI-CAS needs to allow the operators to participate in the decision making processes of the automation systems. The algorithms that are capable to meet the cooperation requirements and, therefore, be used in the HMI-CAS system, are presented in the next chapter.

Chapter 4 Methodology for Developing HMI-CASs

The framework of Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS) has been presented in the last chapter, consisting of a GNC system and an interface. However, not all the GNC system embedded an interface can support all forms of interactions addressed in Section 3.3. This chapter conducts a comparison among various prevention algorithms (collected in Chapter 2), and a group of algorithms supporting all forms of interactions is found, named as Velocity Obstacle (VO) algorithm. Moreover, by applying the variations of VO algorithm, some limitations of traditional ship prevention algorithm have been overcome. Specifically, the movement of the TS is no needed to be invariant and deterministic. Moreover, it is proofed in this chapter that the Linear VO algorithm (LVO) is identical to some classical methods from maritime studies, i.e., Collision Threat Parameter Area (CTPA).

This chapter is organized as follows: the comparison of prevention algorithms is presented in Section 4.1 and a family of VO algorithms is introduced in Section 4.2. In Section 4.3, the detail of using VO algorithms in HMI-CAS is presented, followed by a comparison of VO algorithms with traditional methods in Section 4.4. The relevant demonstrations and discussions are shown in Section 4.5 and Section 4.6, respectively. In the end, conclusion is addressed in Section 4.7.

Acknowledgment Part of the content in this chapter is based on the following papers:

Huang, Y., van Gelder, P. H. A. J. M., & Wen, Y. Q. (2018). Velocity obstacle algorithms for collision prevention at sea. Ocean Engineering, 151, 308-321.

Y. Huang, L. Chen, R.R. Negenborn, P.H.A.J.M. van Gelder. "A HMI-CAS for human-machine cooperation during collision avoidance". (Submitted)

Huang, Y. & van Gelder, P. H. A. J. M. (2017). Non-linear velocity obstacles with applications to the maritime domain. In: SOARES, G. & TEIXEIRA (eds.) the 17th International Congress on Maritime Transportation and Harvesting of Sea Resources. Lisbon, Portugal: Taylor & Francis Group.

4.1 Introduction

In the framework of the HMI-CAS, “Conflict Detection” and “Conflict Resolution” are two key modules. In this chapter, the algorithms that are suitable for Human-Machine Interactions (HMIs) in these two modules are shown.

4.1.1 Existing methods for conflict detection

In existing conflict detection methods, “Closest Point of Approach” (CPA) is a well-known concept, which shows the closest point that the Target-Ship (TS) can approach the Own-Ship (OS). Two indicators based on this concept are Distance at CPA (DCPA) and Time to CPA (TCPA). DCPA shows the distance between the TS and the OS when the TS is located at CPA. TCPA indicates the time remaining for the TS reaching the CPA. A collision warning is triggered when DCPA and TCPA are smaller than thresholds. This type of conflict detection method is named as the CPA method. To date, this method has been integrated into Automatic Radar Plotting Aid (ARPA) in conventional ships and has been widely used in practice.

Due to the high acceptance of the CPA method, many detection methods are developed based on the CPA method, aiming at improving the functionalities of collision alert systems on board.

Firstly, the CPA method neglects the influence of operational environments and the regulations, which easy results in frequent alarms in dense waters [115]. To handle this issue, researchers developed Collision Risk Index (CRI) methods that incorporate experts’ knowledge and multiple factors in setting warning criteria. Details about CRI are presented in Section 2.4.1. The integrated factors include sailing regulations, ships’ dimension, ship’s states, etc. [91]. These methods are expected to raise a collision alarm in compliant with sailing regulations and good seamanship.

Secondly, the shape of a ship is simplified as a point in the CPA method, which may result in an overestimation of feasible time for steering. It implies that a collision can happen before the ship reaches the CPA [175]. In order to consider the shape of ships in conflict detection and collision avoidance, “ship domain” [175, 176] is introduced as an alternative, named as WR-SD in Section 2.4.2.

Thirdly, for the convenience of searching an evasive action, Collision Threat Parameters Area (CTPA) is introduced, which is derived from the CPA method. CTPA collects a set of velocities leading to the DCPA be smaller than the safety distance [30, 31]. When the velocity of the OS belongs to this set, a collision alarm is raised. Correspondingly, all the velocities outside of this set are collision-free.

Although the CPA method and its variations are dominant in conflict detection, this group of methods has a strict assumption that presumes the TS sails with a constant speed and course. This assumption restricts the application of CPA methods in more general cases. Firstly, the motion of a ship is hard to be invariant, especially when the ship is affected by external disturbances or is taking evasive actions. Thus, the estimated position of CPA could by varying, which may easily lead to false alarms. Secondly, even if the OS obtains the trajectory of the TS by predictions or communication, two CPA indicators cannot truly reflect the closest distance between ships and the relevant time for steering.

In this chapter, the Velocity Obstacle (VO) algorithm and its variations from robotics are adopted that not only can perform as the same as some traditional detection methods but also suitable in more generic maritime environments. In particular, the strict assumption on the TS's movement is released.

4.1.2 Existing methods for conflict resolution

For supporting HMIs in conflict resolution, it is expected that the HMI-CAS enables to offer various forms of solutions to the human operators, specifically, all form of solutions addressed in Table 3.6.

a. one collision-free solution, noted as u;
b. one optimal solution regarding the cost function, noted as u^*;
c. a set of finite collision-free solutions that are checked by the algorithm, noted as $\mathbb{U}$;
d. a closed region of collision-free solutions, which is a continuous sub-space, noted as $\mathcal{U}$;
e. all closed regions of unsafe solutions collecting all unsafe solutions, noted as K.

These forms of solution help the human operators in various ways, e.g., find one solution (a-b), choose another solution in candidates (c-e), validate the safety of arbitrary solutions (e), and even understand how does the machine find a solution (b, e). A comparison of the existing algorithms is conducted to find a proper algorithm, in which 14 representative algorithms from 5 groups are analyzed. The results of the comparison are shown in Table 4.1.

Table 4.1 Overview of solution forms of collision avoidance algorithms

	Algorithm	Encounter types	Output solutions u	u^*	$\mathbb{U}$	$\mathcal{U}$	K	The physical meaning of the solution
Rule-based	Single-rule	Single-	◐	-	-	-	-	turning course [118] or pattern [48]
	Multiple-rule	Single-	◐	-	-	-	-	heading & speed [121]
Virtual vector	Article Potential Field	Multiple-	●	-	-	-	-	course/velocity [125, 126]
	Limited Cycle Method	Single-	●	-	-	-	-	trajectory[42, 128, 129]
Discrete inputs	Dynamic Window	Multiple-	●	●	●	◐	◐	speed & yaw rate [132, 133]
	Decision Disc	Multiple-	●	●	●	◐	◐	velocity [29-32, 73, 130]
	Discrete-Input Optimization	Multiple-	●	●	-	-	-	Velocity [74] / rudder angle [134]
	Lattice-Based Search	Multiple-	●	●	-	-	-	velocity [59, 135]
	Brute-Force Search	Multiple-	●	-	-	-	-	rudder angle & operation time [47]
Continuous inputs	Velocity Obstacle	Multiple-	●	●	●	●	●	Velocities [149]
	Vision Cone	Single-	●	-	-	-	-	Course [41, 152, 154]
	MPC-Collision Avoidance	Multiple-	●	●	-	-	-	trajectory /desired forces [40, 49, 155]
Re-planning	Fast Marching Method	Multiple-	●	●	-	-	-	Path [157]
	Projected Obstacle Area	Multiple-	●	●	-	-	-	Path [105, 106]

* Note: "●" means this algorithm matches the description; "◐" means this algorithm is not fully matching the description; "-" means this algorithm is not matching.

“Rule-based” methods usually offer one feasible solution to operators/ASVs. The feasible solution could be a course [118], a speed [121], or a pattern (enlarge rudder angle until it is collision-free) [48], etc. Nevertheless, since the collision check is not inclusive, some algorithms might not guarantee that the solution is collision-free.

“Virtual vector” methods offer a solution that might not be optimal but collision-free. The solution could be a collision-free course by Artificial Potential Field (APF) or a collision-free trajectory generated by Limited Cycle Method (LCM).

“Discrete inputs” methods follow a common idea that discretizes the control space and then finds solutions. Brute-force search checks the solutions from the discrete control space and returns one collision-free solution. Dynamic Window (DW) and Decision Disc (DD) are able to offer both alternative solutions and an optimal solution. These algorithms check each solution from the discrete solution space, and the optimal solution is found in these solutions. However, since the control space is discretized, these algorithms cannot offer a continuous solution space to users. Besides, the DW implies stopping the ship is always a safe option and DD assumes the target-ship sailing with constant speed and course, which are not always the case at sea. Discrete-Inputs Optimization and Lattice-based search contain an optimization finding one collision-free solution. Thus, they directly offer an optimal solution to the users.

“Continuous inputs” methods search collision-free solutions in a continuous space. Both Visual Cone (VC) and MPC based Collision Avoidance (MPC-CA) can find one collision-free solution to users, while MPC-CA could offer a solution that minimizes the cost function. The VO algorithm firstly identify the unsafe region in the solution space (i.e., K) and then find an optimal solution in the complement (i.e., $\overline{K}$).

Two representative algorithms from “Re-planning” are Fast Marching Method (FMM) and Projected Obstacle Area (POA). FMM assigns a cost map first and then find an optimal path; POA assigns a prohibited region around the predicted position of an obstacle and finds one path to avoid the prohibited region.

Though comparison, it is found that the VO algorithm meets the requirements of supporting all forms of HMIs in the proposed HMI-CA. The main reason is that the VO algorithm presents a set of dangerous solutions (i.e., K) that allows the human operators to validate their solutions (safe or not) and the optimization in the complementary of K set is possible. Moreover, the form of a solution provided by the VO algorithm is friendly for human operators to read and to implement.

To conclude, a family of VO algorithms is adopted for developing the proposed HMI-CAS. This family of algorithms releases the assumption on the movement of the TS and can support human operators with various solution forms. Some fundamental VO algorithms are introduced in the subsequent section, which are Linear-VO, Non-Linear VO, and Probabilistic VO algorithms. Using VO algorithms for Conflict Resolution and Conflict Detection are addressed in Chapter 5 and Chapter 6, respectively.

4.2 Velocity obstacle algorithms

Velocity obstacle algorithm is an algorithm developed in robotics for supporting collision avoidance in dynamic environments. The main idea of this algorithm is collecting the velocity leading to collisions in a set in velocity space, called VO set, and then avoids the collision by choosing a velocity out of the set. The name of this algorithm is given in [147] in the late 90s. Thanks to its simple form, this algorithm has been accepted for different platforms and been developed for different working environments. In the literature, this algorithm has been applied to the unmanned aerial vehicle [177], wheeled robot [178], unmanned surface vehicle [73], etc.

Based on the original VO algorithm, a series of velocity obstacle algorithms have been developed. The original VO algorithm requires that the moving obstacle's velocity is constant. Thus, the original VO is also called Linear VO (LVO) in this dissertation. To overcome the constraints on the constant velocity, a Non-Linear VO (NLVO) algorithm is proposed [179]. Since then, a family of VO algorithms has been developed for preventing collisions in different scenarios. The fundamental VO algorithms, however, easily lead to oscillatory behavior in collision avoidance due to the poor coordination between ships. To eliminate the oscillatory, a Reciprocal VO (RVO) is proposed, which assumes the vehicles could share the duty of maneuvers, which performs successfully in a crowd simulation with 1000 agents [180]. One remark of this algorithm is that the agents cooperate without communications. Based on this algorithm, the authors proposed an Optimal Reciprocal Collision Avoidance (ORCA) that incorporates the optimization in searching the best collision-free velocity [181]. Based on these developments, Cooperative Collision Avoidance (CCA) is proposed, which offers a solution to handle non-holonomic constraint for the vehicle [148]. In [178], a Generalized VO (GVO) algorithm is proposed, which accepts a relatively simple non-holonomic constraint on the vehicle. Bareiss and van den Berg [182], later on, developed a general framework of GVO algorithm considering different dynamics for unmanned vehicles, e.g., the wheel-based robot, quadrotors, [182, 183], etc. These mentioned VO algorithms usually assume the trajectory of an obstacle is known or predictable by a known and deterministic motion model, which is strict in practice. Probabilistic VO (PVO) algorithm, then, is proposed to consider the uncertainty of obstacles' movement, and it has successfully applied in an experimental car to avoid collisions with pedestrians [184]. Ellipse-Based VO (EBVO) [185], on the other hand, developed a VO algorithm deal with ellipse-shaped agents. This technique was applied to the human-shaped robot in a narrow passage, where the robot has to go through the passage by walking sideways. From the above, it is clear that the family of VO algorithms expands its potential in numerous applications.

In this section, the basic principle of VO algorithm and some of its variations are explained via a ship to ship encountering scenario. The terminologies used in this section are declared as follows. Two ships are encountering each other, say ship i and ship j. Ship i is under our control, which is also named as own-ship (OS). Ship j is so-called target-ship (TS). The position and the velocity of the ship are denoted as P and $\vec{v}$. The subscript i and j indicate that the states belong to ship i (the OS) and ship j (the TS), respectively. For instance, $P_i(t)$ refers to the position of the OS at time t. Additionally, since the workspace of the ship is two-dimensional, $P_i(t)$ is a vector containing two states.

4.2.1 Representation of the obstacle

In this section, the ship is represented by a geometric object. The collision between the OS and the TS is represented by an overlap of two geometric objects where one represents the TS and the other represents the OS.

Figure 4.1 Illustration of two representations of *ConfP* (circular and elliptical).

If the geometric object of the TS is set at the origin, all the configurations of the geometric object of the OS resulting in the overlap of two objects are named as "conflict position", noted as *ConfP*. An illustration is presented in Figure 4.1. If the position of the OS belongs to this set, the OS is striking the TS located at the origin. The shape and scale of *ConfP* strongly depend on the shape, the size, and the state of ships. If the OS and the TS are seen as two circles (with radius r_i and r_j, respectively), the *ConfP* can be formulated as:

$$ConfP(O,R) = \left\{ P \mid \left\| P - O \right\| \le r_i + r_j \right\},$$

where P is denoted as a position in workspace and O is the origin of the workspace. If the ship is modelled as an ellipse, the shape of *ConfP* is adjusted and presented in the lower panel of Figure 4.1. When the ship is modelled as an ellipse, the course of the ship also influences the shape of the *ConfP*. For simplicity, a circular representation of the ship is accepted in this dissertation. Thus, *ConfP* shapes like a circle.

To formulate the position that leads to a collision with the TS at any position, a Minkowski addition is employed, noted as $\oplus$, specifically:

$$P_j \oplus ConfP(O,R) \triangleq \left\{ P \mid P = P_j + p, \forall p \in ConfP(O,R) \right\}, \tag{4.1}$$

here p is any position in *ConfP*, P_j is the position of the TS, and P is an arbitrary position in the workspace.

4.2.2 Basic VO algorithm (Linear VO algorithm)

The basic VO algorithm is also noted as the LVO algorithm that assumes the movement of the ship j is known, and the ship keeps its velocity. The task of the LVO algorithm is to identify the velocity that leads to collision when the OS keeps such velocity in a period of time.

Interpretation I

Step 1: formulate a necessary and sufficient condition of the ship collision at time t_f.

Equation (4.1) shows the condition that the OS collides with the TS. If the OS takes the relevant position in this set, the collision between the OS and the TS happens. Thus, the OS will collide with the TS at time t_f, if and only if the position of Ship i is in $P_j(t_f) \oplus ConfP(O)$, i.e.,

$$P_i(t_f) \in P_j(t_f) \oplus ConfP(O,R). \tag{4.2}$$

Step 2: find a set of the velocity of the OS leading to a collision at time t_f.

Expand the $P_i(t_f)$ by $P_i(t_0)+\vec{v}_i(t_f-t_0)$, which means the OS keeps a chosen speed in the future, then:

$$P_i(t_0)+\vec{v}_i(t_f-t_0) \in P_j(t_f) \oplus ConfP(O,R). \tag{4.3}$$

Solve this equation regarding $\vec{v}_i$, Equation (4.3) turns to be

$$\vec{v}_i \in \frac{P_j(t_f)-P_i(t_0)}{t_f-t_0} \oplus \frac{ConfP(O,R)}{t_f-t_0} = sVO(t_f). \tag{4.4}$$

A velocity of the OS satisfying Equation (4.4) means if the OS keeps this velocity from t_0 to t_f, the collision will happen at time t_f. $sVO(t_f)$ denotes a set of the velocity of the OS that satisfies Equation (4.4).

Step 3: construct a set of the velocity leading to a collision at any time slice in the future.

Equation (4.4) collects a set of the velocity leading to a collision at time t_f. Thus, a set of velocities leading a collision at any time is a union of $sVO(t)$, where $t \in (t_0, \infty)$, i.e.,:

$$VO = \bigcup_t^{\infty} sVO(t) = \bigcup_t^{\infty} \left(\frac{P_j(t)-P_i(t_0)}{(t-t_0)} \oplus \frac{ConfP(O,R)}{t-t_0} \right). \tag{4.5}$$

The velocity in this set means if the OS keeps this velocity forever, a collision is guaranteed to happen in the future ($t_0 \to \infty$).

Step 4: formulate Linear VO with an assumption on the TS's movement

The VO set identified by linear VO algorithm is denoted as LVO set. The term "linear" refers to an assumption of the motion of the TS: the TS sails in straight-line with a constant speed and course. In other words, $P_j(t)=P_j(t_0)+\vec{v}_j(t-t_0)$. Substituting this condition in Equation (4.5), the mathematical expression of LVO that is formulated as:

$$LVO = \bigcup_t^{\infty} \left[\left(\frac{\vec{d}_{ij}(t_0)}{t-\mathrm{t}_0} + \vec{v}_j \right) \oplus \frac{ConfP(O,R)}{t-t_0} \right] \tag{4.6}$$

where, $\vec{d}_{ij}(t_0)$ is the relative distance between ships at time t_0, i.e., $P_j(t_0)-P_i(t_0)$.

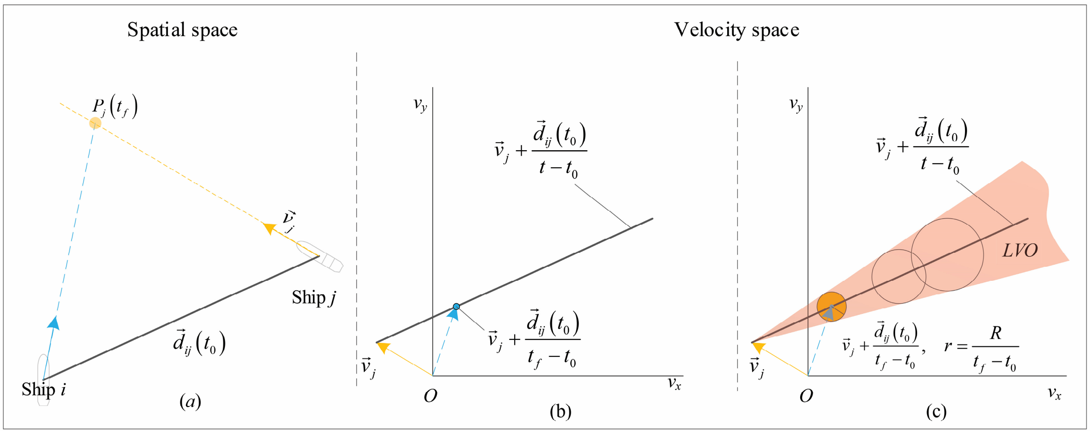


Figure 4.2 Illustration of Linear VO set following Interpretation I.
(note: (a) is a geographical display; (b) is the LVO set neglecting the dimension of ship; (c) LVO considering the dimension of the ship)

The first part of Equation (4.6) formulates the VO set ignoring ship's dimensions, i.e., a straight line in Figure 4.2 (b). The point on this line can be re-formulated as: $k(t)\cdot\vec{d}_{ij}(t_0)+\vec{v}_j$, where $k(t)=(t-t_0)^{-1}$ and t is the collision time. The second part of the equation considers the dimension and *ConfP* is rescaled by $k(t)$ and located at a point on the line $k(t)\cdot\vec{d}_{ij}(t_0)+\vec{v}_j$.

Combining these two parts, a family of prohibited zones in the velocity space is obtained, and the envelope of these zones is the LVO set, which is shaped like a cone. A demonstration is shown in Figure 4.2 (c).

Interpretation II

Step 1: collision between ship *i* and a stationary ship *j*.

As long as ship *i* do not head to the *ConfP* surrounding the stationary ship *j*, the collision is not going to happen. In other word, when the course of the ship *i* keeps pointing to the *ConfP*, the collision will happen in the future.

The region of velocities pointing to the *ConfP*, i.e., the velocities leading to a collision, is collected in Figure 4.3 (b), which is shaped like a cone. This cone is bounded by two rays tangent to the *ConfP*. When the velocity of the ship *i* is in this cone, the collision will happen in the future.

Step 2: collision between ship i and a ship j with a constant speed.

To reuse the finding in the previous step, i.e., (1), the ship *j* is treated as a stationary obstacle and the ship *i* moves with the relative velocity, i.e., $\vec{v}_{ji}=\vec{v}_i-\vec{v}_j$. Thus, the relative velocity leading to a collision is falling in the cone shown in Figure 4.3 (b). As long as the relative velocity of the ship *i* in this cone, the ship *i* will head to the *ConfP* and a collision will happen. Figure 4.3 (b) shows the relative velocity that leads to a collision. To show the velocity of the ship *i* leading to a collision, this region needed to be moved by $\vec{v}_j$. Then, the LVO set is obtained, see Figure 4.3 (c).

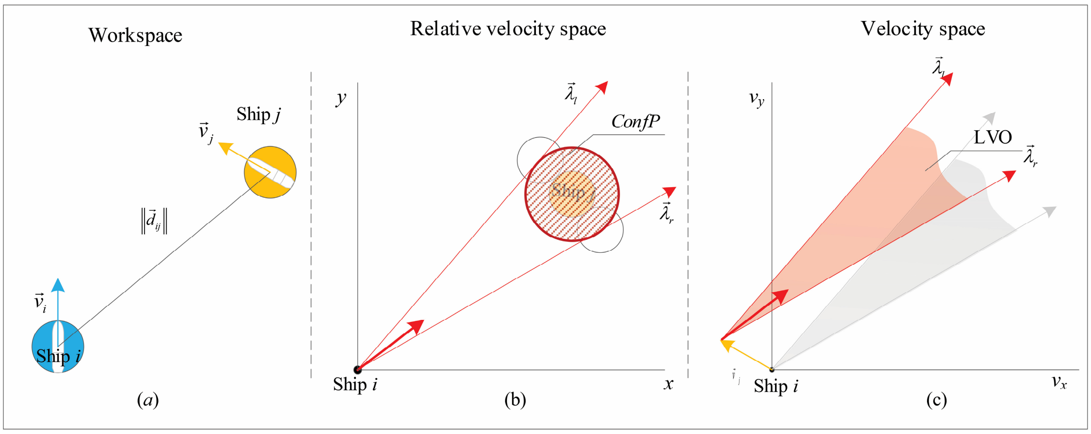


Figure 4.3 Illustration of Linear VO set following Interpretation II.
(note: (a) is a geographical display; (b) is the velocity resulting in the relative velocity heading to the obstacle; (c) LVO of ship *i*)

4.2.3 Non-linear VO algorithm

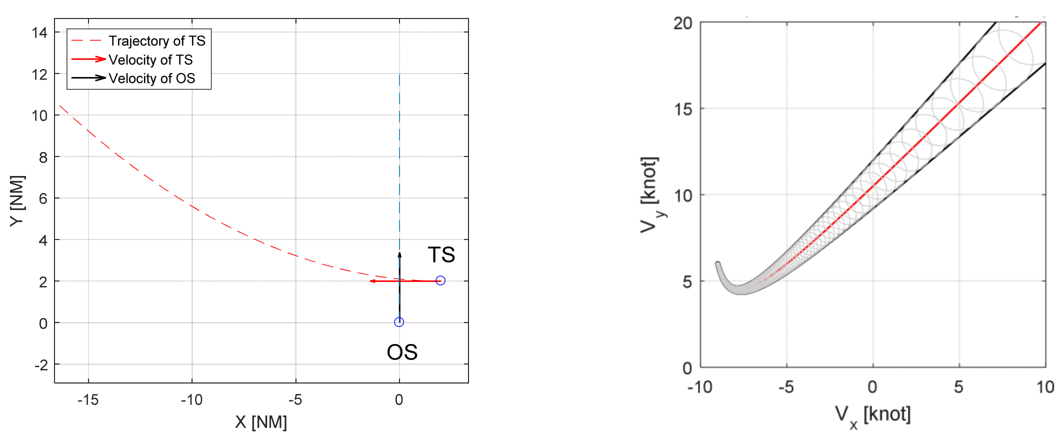
The non-linear VO algorithm is an expansion of the LVO algorithm, which is proposed by Large, Sekhavat [186]. This algorithm allows the TS to change its velocity during the collision avoidance process, but these changes are known by the OS. The relevant VO set is noted as NLVO set. A demonstration of NLVO is shown in Figure 4.4.

If the position of the OS is set at the origin, i.e., $P_i(t_0)=[0,0]^{\mathrm{T}}$, Equation (4.4) and (4.5) are simplified to:

$$\vec{v}_i \in \frac{P_j(t)}{(t-t_0)} \oplus \frac{ConfP(O,R)}{t-t_0} \quad \text{and} \tag{4.7}$$

$$\text{NLVO} = \bigcup_{t}^{\infty}\left(\frac{P_j(t)}{(t-t_0)} \oplus \frac{ConfP(O,R)}{t-t_0}\right). \tag{4.8}$$

In Equation (4.8), $\bigcup_{t}^{\infty}\left(P_j(t) \oplus ConfP(O,R)\right)$ is the trajectory of the TS in geographical space. Then, NLVO is seen as the projection of this trajectory from geographical space into velocity space via a projection function $(t-t_0)^{-1}$.



(1) Display of the encounter scenario (2) Relevant NLVO set
Figure 4.4 Illustration of encounter scenario and its NLVO set.

4.2.4 Probabilistic VO algorithm

The VO algorithms in Section 4.2.2 and 4.2.3 claim that the OS will strike the TS when the OS selects and keeps a velocity in a VO set in a certain time. This argument strongly relies on the quality of trajectory prediction of the TS, and it presumes the TS will follow the predicted trajectory. However, when two ships are not cooperating, the prediction of the TS's trajectory would inevitably include uncertainties. Thus, the prediction of trajectory might contain various possible trajectories instead of one unique trajectory.

To handle the non-unique trajectory prediction, Probabilistic VO (PVO) algorithm is proposed. It assumes that the possible trajectories of the TS and their probabilities are known. The sum of the probabilities is equal to 1. The VO set identified by the PVO algorithm is noted as PVO set. A demonstration of the PVO set is in Figure 4.5 where the TS has three equal likely trajectories.

Given an arbitrary trajectory of the TS, say k-th trajectory, the relevant VO set is identified via Equation (4.8), specifically:

$$VO_k = \bigcup_{t_f}^{\infty} \left(\frac{P_j^k\left(t_f\right)}{\left(t_f - t_0\right)} \oplus \frac{ConfP\left(O,R\right)}{t_f - t_0} \right), \tag{4.9}$$

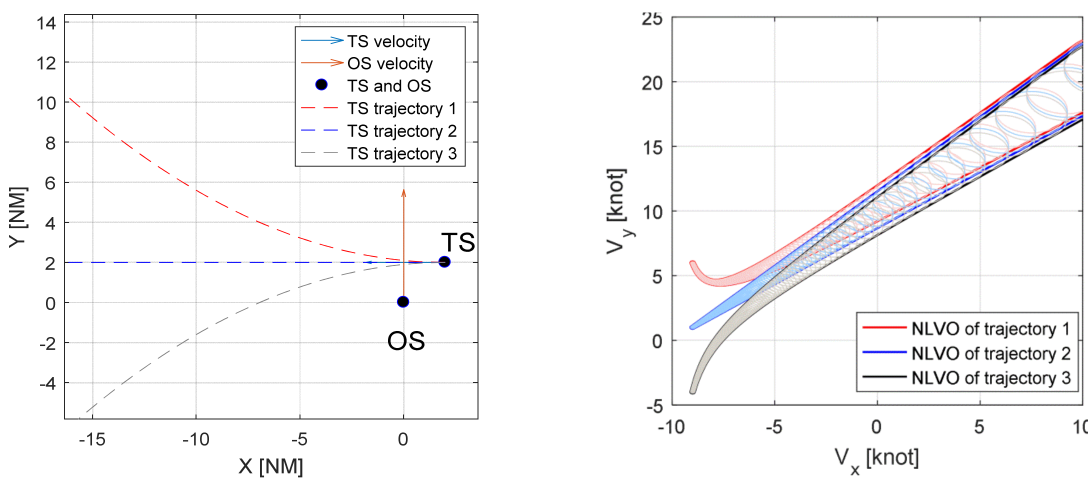
where $\bigcup_{t_f}^{\infty} P_j^k\left(t_f\right)$ is the k-th possible trajectory of the TS. Thereby, the whole PVO set is a collection of all the VO sets with different possible trajectories, which is formulated as:

$$PVO = \bigcup_{k=1}^{n} VO_k \,, \tag{4.10}$$

where n is the total number of possible trajectories of the TS. The probability of the TS choosing the k^{th} trajectory is noted as $p\left(\text{VO}_k\right)$ with:

$$\sum_{k=1}^{n} p\left(\text{VO}_k\right) = 1. \tag{4.11}$$

The probability of a velocity leading to a collision in this space has the following rules:



(1) Three possible trajectories of the TS (2) PVO set with 3 possible NLVO sets

Figure 4.5 Illustration of encounter scenario and its PVO set.

(1) When the velocity is out of PVO set, the collision will not happen. Thus, $p\left(\text{collision}|\vec{v}_i \notin \text{PVO}\right)=0$.

(2) When the velocity is in PVO set, the probability of collision is equal to the probability of the predicted trajectory. This probability is denoted by $p\left(\text{collision}|\vec{v}_i \in \text{PVO}\right)$ and formulated as follows:

$$\begin{aligned} p\left(\text{collision}|\vec{v}_i \in \text{PVO}\right) &= \sum_{k=1}^{n} p\left(\text{collision} \mid \vec{v}_i \in \text{VO}_k, \text{VO}_k\right) p\left(\vec{v}_i \in \text{VO}_k \mid \text{VO}_k\right) p\left(\text{VO}_k\right) \\ &= \sum_{k=1}^{n} p\left(\vec{v}_i \in \text{VO}_k \mid \text{VO}_k\right) p\left(\text{VO}_k\right) \end{aligned}, \tag{4.12}$$

where, $p\left(\text{collision} \mid \vec{v}_i \in \text{VO}_k, \text{VO}_k\right)=1$, $p\left(\vec{v}_i \in \text{VO}_k \mid \text{VO}_k\right)=\begin{cases} 1 & \text{if } \vec{v}_i \in \text{VO}_k \\ 0 & \text{otherwise} \end{cases}$.

4.2.5 Remarks

Remark 4.1

The apex (tip where the curves meet to form a sharp angle) of LVO set is determined by $\vec{v}_j$, and the direction of the cone is determined by $\vec{d}_{ij}\left(t_0\right)$, see Figure 4.3.

Remark 4.2

As the OS chooses its velocity $\vec{v}_i$ close to the apex of the LVO set, i.e., a, the collision time will be postponed.

Proof of Remark 4.2: A $\vec{v}_i$ in an LVO set satisfies $\vec{v}_i = k \cdot \vec{d}_{ij}\left(t_0\right) + \vec{v}_j$. When $\vec{v}_i \to \vec{v}_j$, we have $k \cdot \vec{d}_{ij}\left(t_0\right) \to 0$. Since $\vec{d}_{ij}\left(t_0\right) \neq 0$, we have $k = \left(t_f - t_0\right)^{-1} \to 0$. Thus, t_f needs to be infinitely large. That means the collision will happen, but in a far future. □

Remark 4.3

A set of velocities of the OS leading to collisions at t_f is a projection of the prohibited region at t_f from the workspace into the velocity space by a transition function $T(X)$ that is defined as $T\left(X\left(t\right)\right)=\left(t-t_0\right)^{-1} X\left(t\right)$.

By introducing $T\left(\bullet\right)$, Equation (4.4) can be rewritten as

$$\vec{v}_i \in sVO\left(t_f\right) = T\left(\underbrace{\left(P_j\left(t_f\right) - P_i\left(t_0\right)\right) \oplus ConfP\left(O,R\right)}_{X\left(t_f\right)}\right). \tag{4.13}$$

X is the prohibited region around the TS at the time t_f. By projecting this prohibited region from geographical space into velocity space via function $T\left(X\right)$, a set of velocities leading to collisions at t_f is obtained. Correspondingly, the VO set is

$$VO = \bigcup_{t_f}^{\infty}\left\{T\left[\left(P_j\left(t_f\right) - P_i\left(t_0\right)\right) \oplus ConfP\left(O,R\right)\right]\right\}. \tag{4.14}$$

Thus, a VO set can be seen as a projection of a series of prohibited regions from the geographical space into the velocity space via a time-dependent transition function: $T(\bullet)$.

From Remark 4.3, it is observed that the transition function is a monotonically decreasing function. It means that if the collision time t_f becomes infinitely large, $\left(t_f - t_0\right)^{-1}$ approaches to 0 and $T\left(ConfP\left(O,R\right)\right)$ goes to 0. In other words, the prohibited region is shrunk to a point; otherwise, the prohibited zone is enlarged. This phenomenon is concluded in Remark 4.4:

Remark 4.4

As the OS chooses its velocity close to the apex of the VO sets, i.e., the small size of the VO set, the collision time between two ships will be postponed.

4.3 Using VO algorithms in the HMI-CAS

To apply the aforementioned VO algorithms in the proposed HMI-CAS, several assumptions are made:

1. The dynamics of the ship is ignored, and the ship is seen as a holonomic vehicle;
2. The trajectory of the TS is known or probabilistically predictable;
3. The ship is shaped like a circle.

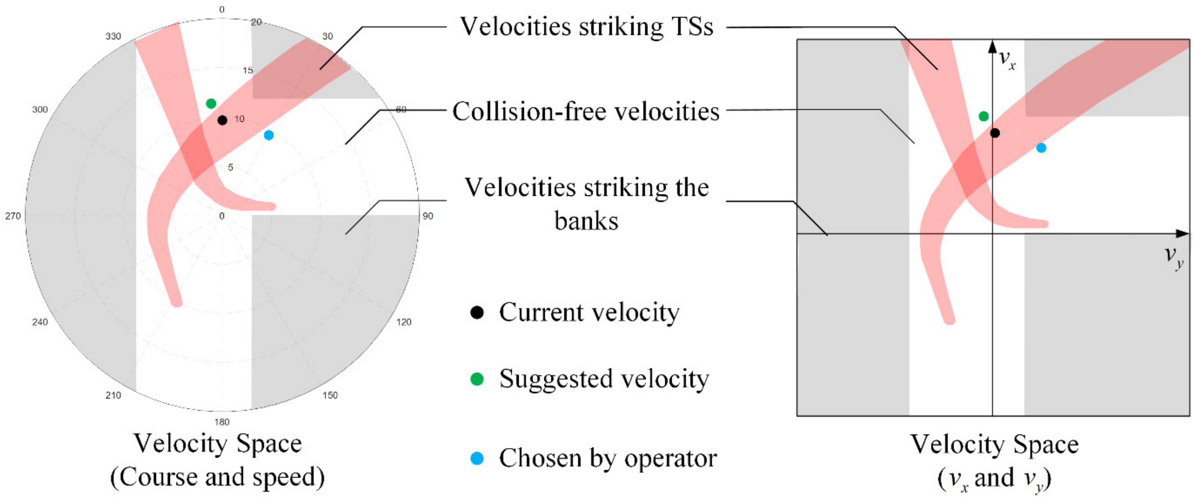
Following these assumptions, the VO algorithms can help the HMI-CAS to identify collision dangers and find collision-free solutions. Details are introduced in following sub-sections.

4.3.1 Design of interface

The LVO, NLVO, and PVO algorithms offer multiple solutions to the proposed HMI-CAS. When the TS has confirmed its sailing intention that keeps its course and speed, the LVO algorithm can identify the dangerous velocities. When the TS would like to take actions, and its trajectory has been communicating with the OS, the NLVO set can collect the velocity leading to collisions. When the OS has predicted the trajectory of the TSs with a certain confidence, the PVO set can collect all the velocity that results in a collision with an unacceptably high probability. All these sets are presented to human operators via an interface of HMI-CAS.

The interface of HMI-CAS using VO algorithms has two modes that are shown in Figure 4.6. In the left panel, a radar-mode interface is adopted. The rings in this mode show the speed of the own-ship and the orientations are the course of the own-ship. In the right panel, a Cartesian-mode is adopted, where the coordinates represent the speed in x-axis and y-axis respectively. Both modes are identical, but using different presentation ways.

In these interfaces, the velocities resulting in a collision with TSs are collected in red regions, and the one leading to a collision with banks is in grey regions. The current velocity of the ship is presented in a black dot. Since this dot is located in the red region, the HMI-CAS suggests a collision-free solution, i.e., the green dot. When the operators disagree with the HMI-CAS's solutions, they are allowed to choose another solution via the interface, e.g., the blue dot.



(1) Radar mode (2) Cartesian mode

Figure 4.6 Illustration of two forms of HMI-CAS interface.

4.3.2 Design of conflict detection

When the current velocity of the OS falls in VO sets generated by TSs, e.g., the presented interfaces in Figure 4.6, the HMI-CAS would trigger the conflict resolution module to find a solution. In the meantime, an early warning is triggered for reminding the human operators.

In principle, all the velocities falling in a VO set are unsafe for the OS, regardless of the position of the velocity inside the VO set. However, according to Remark 4.4, as the velocity approaches the apex of the VO set, the collision will be postponed to the distant future. In practice, if a TS will collide with the OS in the distant future, the TS is not seen as a threat since the OS has sufficient time. Hence, the velocity which is in the VO set but postpones the collision beyond a certain time threshold can also be seen as a collision-free solution for the OS. This time threshold is called collision-warning time (i.e.: t').

Given an encounter scenario and collision-warning time, a VO set can be reconstructed as a collection of the sub-VO set where the collision time smaller than t', i.e.,

$$VO = \bigcup_{t \le t'} sVO(t). \tag{4.15}$$

This collision-warning time can be set by the human operator. An example is shown in Figure 4.7, where the red region is the VO set with collision-warning time t'.

A VO set, then, divides the velocity space into several sub-spaces, namely S_1, S_2, S_3, and S_4. Velocities in different sections lead to different outcomes, such as passing safely, postponing the collisions and immediate collisions.

The OS's velocities in S_1 and S_4 segment will lead to collisions in the future, but the time of collision is different. The velocities in S_1 segment will result in collisions before time t', while the velocities in S_4 segment lead to collisions after time t'. If the collision beyond time t' is not a threat to OS, the velocity in S_4 can be temporary a collision-free solution.

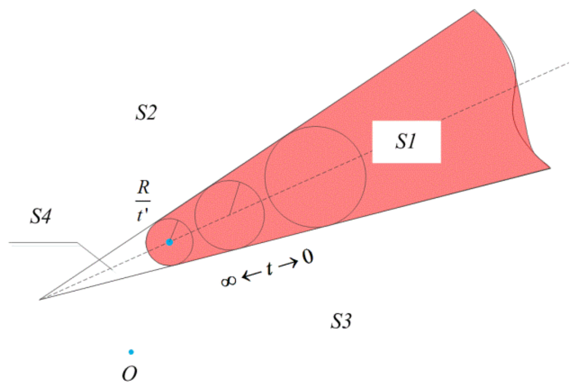


Figure 4.7 Illustration of the OS's velocity space divided by a VO set.

The velocities in S_2 and S_3 are all permanent collision-free velocities for the OS. The velocities in these two segments will make the relative distance always larger than the safety distance. In particular, the velocities in S_2 leads the TS to pass the OS by its stem, and the velocities in S_3 leads the TS to pass the OS by its stern.

In summary, the velocities in S_1 lead to collisions within time t', which is dangerous for the OS, the velocities in S_4 postpone the collision beyond t', and the velocities in S_2 and S_3 are collision-free velocities for the OS. This setting, in fact, increases the solutions of the HMI-CAS when the OS works in dense waters.

4.3.3 Design of conflict resolution

The design of the conflict resolution module using VO algorithms is shown in Figure 4.8. Three sub-modules are included: "Reference velocity" sub-module generates the reference velocity; "VO algorithms" sub-module produces various VO sets, which has been introduced in Section 4.2; "New Resolution" sub-module find a new collision-free solution using VO sets.

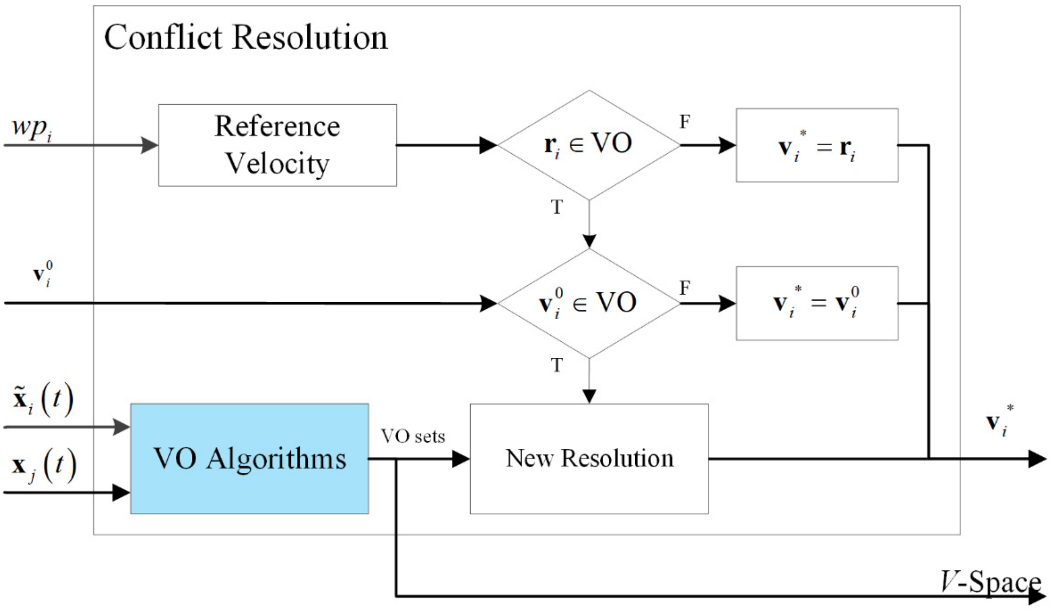


Figure 4.8 Representation of conflict resolution module using VO algorithms.

4.3.3.1 Reference velocity

To determine the time for the own-ship returns to its route the waypoints, a reference velocity is needed. To calculate the reference velocity ($\mathbf{r}_i$), a constant speed is introduced that is the economical speed of the OS, i.e., u_{eco}.

$$\mathbf{r}_i = \frac{wp_i - \mathbf{x}_i}{\left\| wp_i - \mathbf{x}_i \right\|} u_{eco}, \tag{4.16}$$

where wp_i is the waypoint the ship prefers to reach and $\mathbf{x}_i$ is the position of the ship at present.

4.3.3.2 New Resolution

When applying the VO algorithms, the VO sets are used to find a collision-free solution. The rules for finding a collision-free solution are concluded as follows:

Rule 1: The OS is expected to choose its reference velocity, if $\mathbf{r} \notin \mathrm{VO}$;

Rule 2: The OS prefers to continue with its initial desired velocity $\mathbf{v}^0$, when $\mathbf{v}^0 \notin \mathrm{VO}$;

Rule 3: A new $\mathbf{v}^*$ is chosen, which is close to its current velocity and satisfies $\mathbf{v}^* \notin \mathrm{VO}$.

Following these rules, the system firstly would check whether the reference velocity is inside any VO sets. If not, Rule 1 is applied and the OS steers to the reference velocity. If yes, the reference velocity is unsafe and the system will check whether the current velocity is safe. When the current velocity is safe, Rule 2 is activated. Otherwise, Rule 3 is activated and a new collision-free velocity is needed.

For Control Mode 1-2 in MASS type I and II, the captains and human operators can appoint one solution outside of VO sets which is collision-free. For Control Mode 3-5 in MASS type II-IV, the HMI-CAS either can find an arbitrary solution (velocity) outside of VO sets.

When the solution is selected, the "Control" system from the GNC system will track the selected solution or the human operator can directly steer to the selected velocity.

4.3.3.3 Additional auxiliary for human operators

Theoretically, an arbitrary velocity outside of VO sets is a collision-free solution. However, in practice, a specific one velocity from numerous collision-free solutions needs to be found. In this chapter, the HMI-CAS would offer sufficient information to human operators and let them determine the final actions.

A combination of VO algorithms and CPA indicators is proposed that is also a familiar way for mariners on board. However, the calculation of the CPA needs to adapt to a non-linear motion case. To distinguish from the traditional CPA, CPA* is denoted as the closest point of approach when a TS is sailing non-linearly. The DCPA* is the minimal distance between the OS and the TS and the TCPA* is the time to reach that scenario, which is formulated as:

$$\begin{cases} D_{cpa}^* = \min\limits_t \left\{ \left\| \vec{v}_i \cdot t - P_j(t) \right\| \right\} \\ T_{cpa}^* = \arg\min\limits_t \left\{ \left\| \vec{v}_i \cdot t - P_j(t) \right\| \right\} \end{cases}, \tag{4.17}$$

here $\vec{v}_i$ is a potential solution selected by operators and $P_j(t)$ is the position of the ship at *t*.

DCPA* and TCPA* reflect the effects of the selected velocity. If a velocity has a bigger DCPA*, this implies it can keep a relatively larger distance with ships in the future. If a velocity has a small TCPA*, this indicates the encountering process will speed up and the OS will pass the TS soon. In this way, human operators can know about the effect of the chosen velocity, compare the different velocity options, and make proper decisions.

4.4 Comparison of VO algorithms with traditional methods

In this section, VO algorithms are compared with two traditional methods in maritime studies, namely CTPA and CPA approaches. The CTPA from maritime studies, that is proposed in the early 1980s [30, 31], is identical to one type of VO algorithms, i.e., LVO. Comparing with VO algorithms, the CPA method might suffer from three limitations in terms of conflict detection and conflict resolution.

4.4.1 Comparison with CTPA

CTPA and VO algorithms are both used to identify the velocity of the OS leading to a collision with TSs. However, they have different assumptions about the motion of TS. CTPA presumes the TS to sail linearly, while VO algorithms accept the non-linear motion of the TS, e.g., NLVO.

CTPA is developed from the CPA approach. Thus, they share the same assumption about TS's motion. In particular, the CTPA method identifies a set of velocities which will result in the DCPA being smaller than a certain threshold [31], i.e., $d_{CPA} \le d_{thres}$. The CTPA is shaped like a cone in the velocity space, and its apex is numerically equal to the velocity of the TS.

VO algorithms, on the other hand, are not in replying to the calculation of CPA, and they can be used in more generalized scenarios. In fact, the CTPA algorithm is identical to one special case of VO algorithms, i.e., the LVO algorithm.

A velocity in LVO set, which satisfies Equation (4.2), leads to the OS violating the prohibited region of a TS at a specific time in the future. It implies that the relative distance between the OS and the TS will be smaller than the threshold ($\exists t \ge t_0$, which satisfies $\left\|\vec{d}_{ij}(t)\right\| \le d_{thres}$). In other words the smallest relative distance between ships is smaller than the threshold. This smallest relative distance is also noted as DCPA. Therefore, the velocity belongs to LVO set also belongs to DCPA smaller than the threshold, i.e., velocity belongs to CTPA. Thus, LVO is identical to CTPA. The proof is addressed as follow.

Proof:

$$\vec{v} \in LVO \Leftrightarrow \exists\left\|\vec{d}_{ij}(t)\right\| \le d_{thres} \Leftrightarrow \min\left\{\left\|\vec{d}_{ij}(t)\right\|\right\} \le d_{thres} \Leftrightarrow d_{CPA} = \min\left\{\left\|\vec{d}_{ij}(t)\right\|\right\} \le d_{thres} \Leftrightarrow \vec{v} \in CTPA\,. \square$$

4.4.2 Comparison with the CPA approach

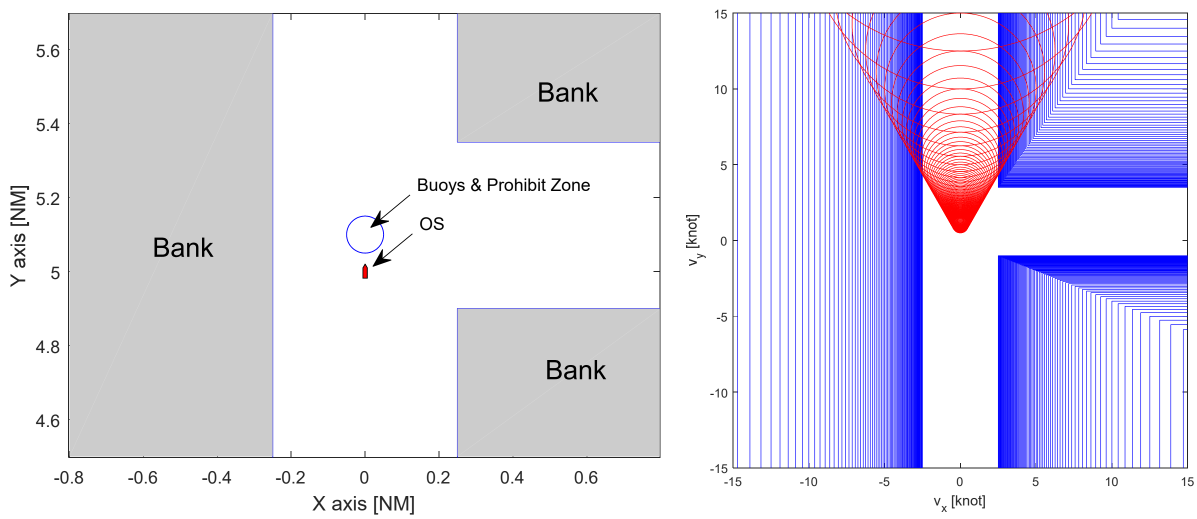
DCPA and TCPA are conventionally employed as criteria to detect collisions. A TS whose DCPA and TCPA are smaller than thresholds will trigger a collision alarm. Although this method has been widely used, it has three shortcomings.

Firstly, the CPA approach easily falls into two types of failures [71]: causing a nuisance alarm during normal navigation, and; not raising the alarm until the collision is unavoidable. The main reason is that the calculation of CPA is based on the observed velocities of the TSs at each time slice. If a TS keeps its velocity, the position of CPA is stationary, and the value of indicators (like DCPA and TCPA) is steady. In this case, the CPA criterion is accurate and consistent. However, if the motion of the TS is non-linear, the CPA is always changing, and the value of indicators is unsteady. As a result, the criterion will be unstable and inconsistent. In Section 4.5, two examples are presented that the CPA-based approach causes a false alarm.

Secondly, to find collision-free actions by the CPA approach is inconvenient. Mariners have to try several virtual maneuvers before they can find a collision-free action. If the OS encounters with multiple TSs, the action to avoid collision with one TS might lead to new collision risks with others. Besides, any changes in the OS's motion will influence the value of DCPA and TCPA from different TSs, which increases the difficulties to find collision-free actions in limited time.

Last but not least, the CPA approach neglects the shapes of the obstacles. The TSs and the OS are seen as points. The CPA is calculated as the closest point that the corresponding TS can reach. However, a collision can happen before the TS reaches the CPA, due to the dimensions of the ships. Additionally, ships not only would encounter these small and regular obstacles but also some huge and irregular obstacles, e.g., islands and coastal line. Since it is impractical to regard these obstacles as points, CPA becomes ineffective for those obstacles.

VO algorithms can fill some of these gaps. Firstly, various types of VO algorithms are developed to consider all kinds of motion assumptions on TS, e.g., linear, non-linear, probabilistic. Secondly, the evasive velocities for the OS are visualized. All the velocities outside of the VO set are collision-free velocities. Hence, the OS that is controlled by humans or machines can avoid collision by choosing a velocity outside of all the VO sets. Moreover, VO algorithms can deal with different obstacles. The coastal-line and the bank can be mapped into the OS's velocity space via Remark 4.3.



(1) Illustration of navigating the environment (2) Relevant VO sets

Figure 4.9 Illustration of VO sets in a restricted water area.

Figure 4.9 demonstrates an example to project the coastal-line into the velocity space. The coastal-line can be obtained from Electronic Nautical Charts (ENC). The left panel shows that the OS placed at [0,5] NM sailing in an intersection connecting three channels and containing one buoy. The buoy is located at [0, 5.1] NM with radius 0.025 NM. The right panel is the velocity space of the OS. The blue region is the VO sets generated by the banks, and the red region comes from the prohibited zone of the buoy. Collision-warning time is set as 10 min in this case. Thus, the OS's velocity in the blue or red region means collision will happen in 10 min. Any of the velocities outside of the VO sets can keep the OS safe in the following 10 min.

4.5 Case studies

4.5.1 Scenario 4-I: comparing with the CPA approach in a single encounter

(1) Settings

A target-ship with the non-linear trajectory is introduced to demonstrate the performance of the NL-VO in the proposed HMI-CAS. Two ships are involved, namely the OS and the TS. The configurations of these ships are presented in Table 4.2 and Figure 4.10. The safety distance is R = 0.16 NM.

Table 4.2 Scenario 4-I: settings of scenario

	Origin [NM]	**Destination [NM]**	**Speed [knots]**	**Heading [°]**	**Ship length [m]**
OS	[0, 0]	[-3, 6]	7.1	334	150
TS	[1, 1]	[-9, 1]	8	270	150

The TS shares her trajectory with the OS (see the blue line in the left panel of Figure 4.10). Then, the relative distance between ships is shown in the right panel, which implies two ships will have a collision in the future.

(1) Employing NLVO algorithm in HMI-CAS

Figure 4.11 shows the velocity space of the OS, where the right panel is the enlarged figure of the left panel. In these panels, the velocity of the OS leading to a collision with the TS have been collected in the blue regions, i.e., NL-VO set. The initial velocity of the OS marked in red is inside the NL-VO set, which indicates the OS will collide with TS. Moreover, the NL-VO set is allowed us to find the solution for OS. The velocity outside of the NL-VO set is potential collision-free solutions. In the right panel, the yellow point that represents [-3.6,7.2] knots is selected as one collision-free solution.

Figure 4.12 shows the situation when the OS accepts her initial velocity to the yellow point. The new trajectory of the OS is presented in yellow in Figure 4.12. The new trajectory is close to the original path, while the relative distance between OS and TS is always larger than the safety distance. That means, the OS can pass the TS safely.

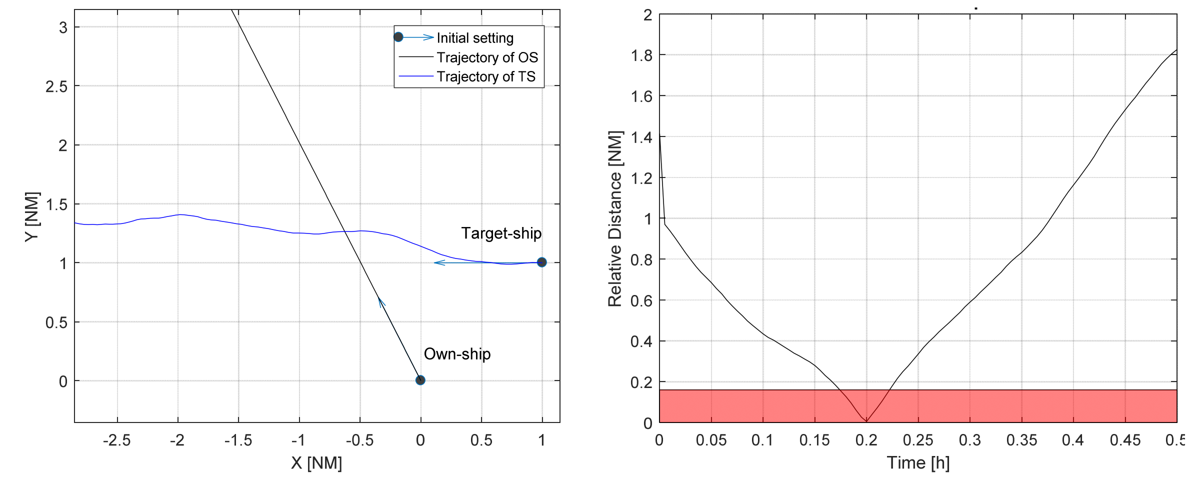


Figure 4.10 Scenario 4-I: encounter scenario and relative distance without evasive actions.

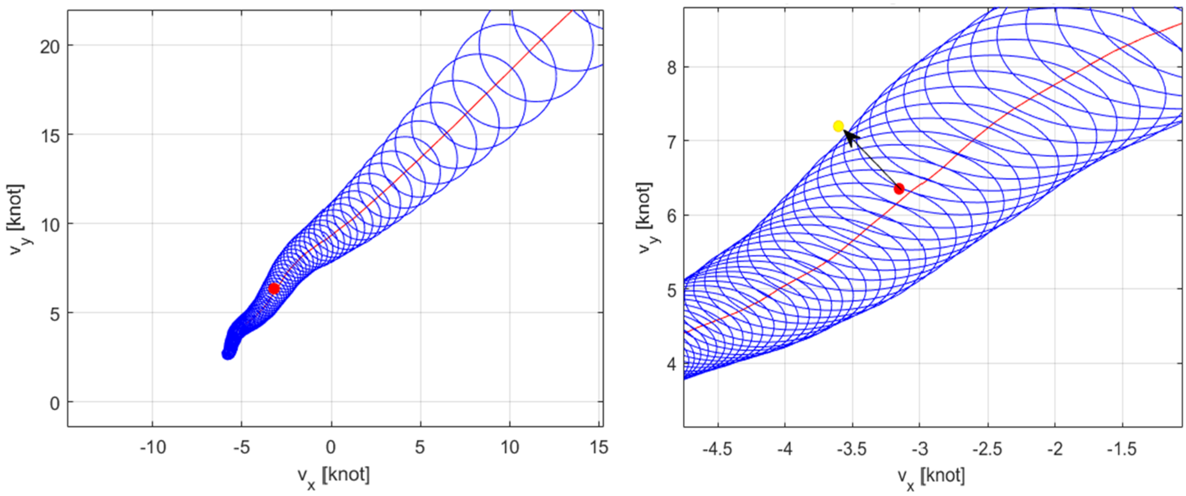


Figure 4.11 Scenario 4-I: VO set in the OS's interface at 0 [min].

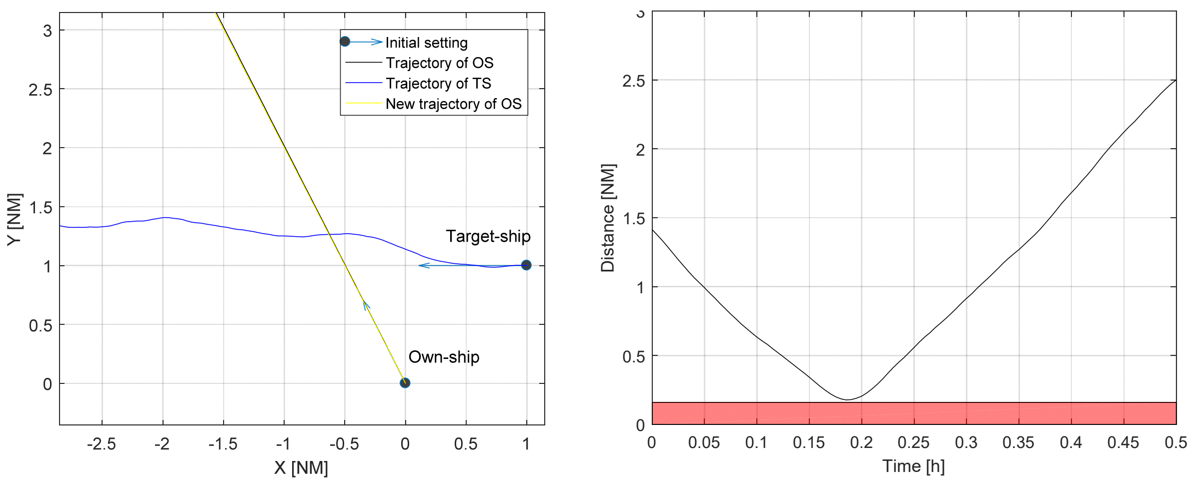
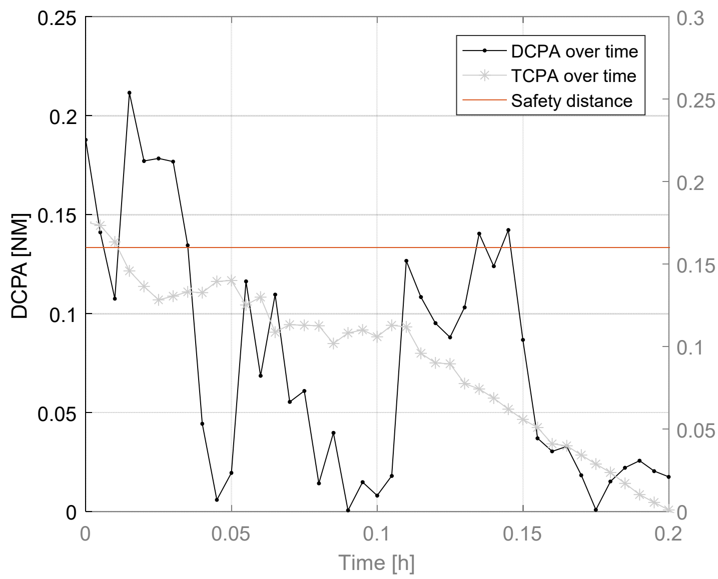


Figure 4.12 Scenario 4-I: trajectories and the relative distance with a new velocity.

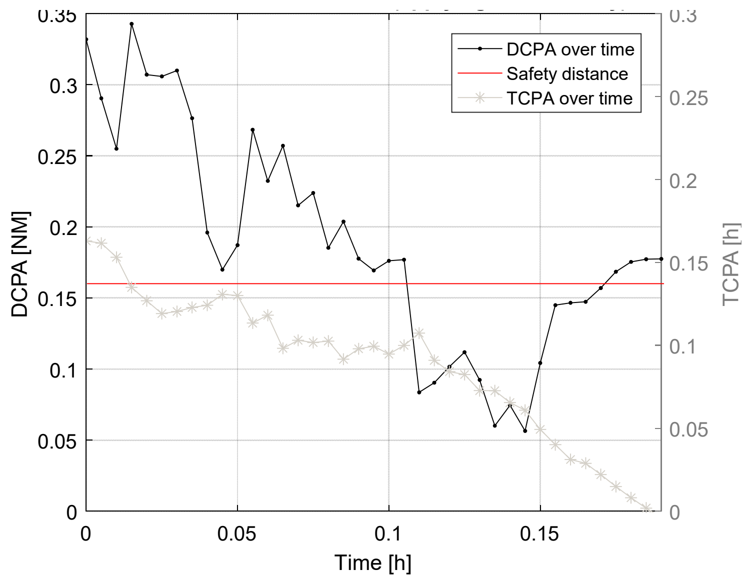
(2) Employing CPA methods in the HMI-CAS

Figure 4.13 shows the evolution of DCPA and TCPA. The left panel shows the DCPA/TCPA value when the OS keeps its initial velocity v_i = [-3.15,6.35] knots. From the panel, it is found that, when TCPA reaches 0, the DCPA is less than 0.16 NM, which means a collision happens. Before the collision, it is hard to obtain feasible information by using DCPA and TCPA alone. In particular, the value of DCPA and TCPA are not stable. The trend of TCPA is clear, which is decreasing. While the value of DCPA is changed dramatically. At 0.05 h (3 minutes before the collision), the value of DCPA is even larger than the safety distance, which may confuse the mariners.

The lower panel in Figure 4.13 shows the DCPA/TCPA when the OS applies v_i = [-3.6,7.2] knots. As shown in Figure 4.12, this solution is safe for OS. While, in the figure, the DCPA was below the safety distance before TCPA reaches 0. It also could lead to a false alarm for the captain.



(1) Applying the initial velocity



(2) Applying a new velocity

Figure 4.13 Scenario 4-I: evolution of DCPA & TCPA with/without a new velocity.

4.5.2 Scenario 4-II: comparing with CTPA in multiple-ship scenario

(1) Settings

The settings of the ships are shown in Table 4.3. TS1 sails with a constant course but accelerates from 10 knots to 12 knots. TS2 sails from the East to the North with a speed of 9 knots and turns to the North around [0.5, 2.5] NM. The dashed line in Figure 4.14 (1) is the planned trajectories of the TSs. The velocities of the TSs at each moment are shown in Figure 4.14 (2).

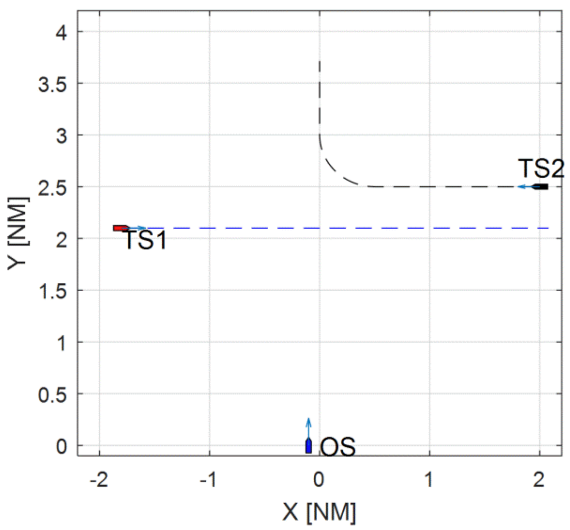
The length of the ships is 150 m. Thus, the prohibited zone is set to be a circular area with a radius of 0.16 NM. In the simulation, the collision-warning time is set to be 30 min, i.e., an obstacle which might collide the OS beyond 30 minutes is not treated as a threat to the OS.

(2) Using NLVO versus CTPA (i.e., LVO) in HMI-CAS

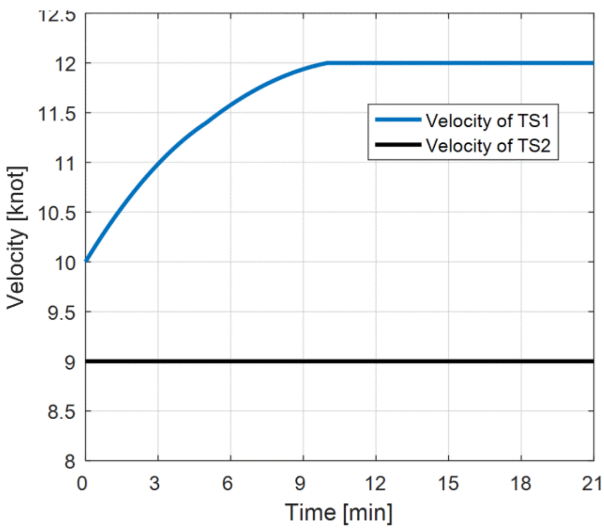
Figure 4.15 shows the VO sets from the OS's perspective. A point in this figure represents an alternative velocity for the OS. In Figure 4.15 (1), two NLVO sets are identified: the red one is a set of velocities leading the OS to collide with TS1 in 30 minutes; the blue one is a set of velocities resulting in a collision with TS2. Since the OS's velocity is in TS2's NLVO set and outside of TS1's NLVO set, OS will pass TS1 safely, but collide with TS2. Specifically, TS1 will pass the OS from the OS's stem safely, while the OS will collide with TS2 at the OS's starboard side.

Table 4.3 Scenario 4-II: simulation settings

	Origin [NM]	Destination [NM]	Speed [knots]	Course [°]	Ship length [m]
OS	[-0.1, 0]	[-0.1, 10]	12	000-360	150
TS1	[-1.8, 2.1]	[2, 2.1]	10 to 12	90	150
TS2	[2, 2.5]	[0, 4]	9	270 at the beginning and turns to the North around [0.5 2.5]	150



(1)



(2)

Figure 4.14 Scenario 4-II: multiple-encounter scenario and evolution of speed of TSs.

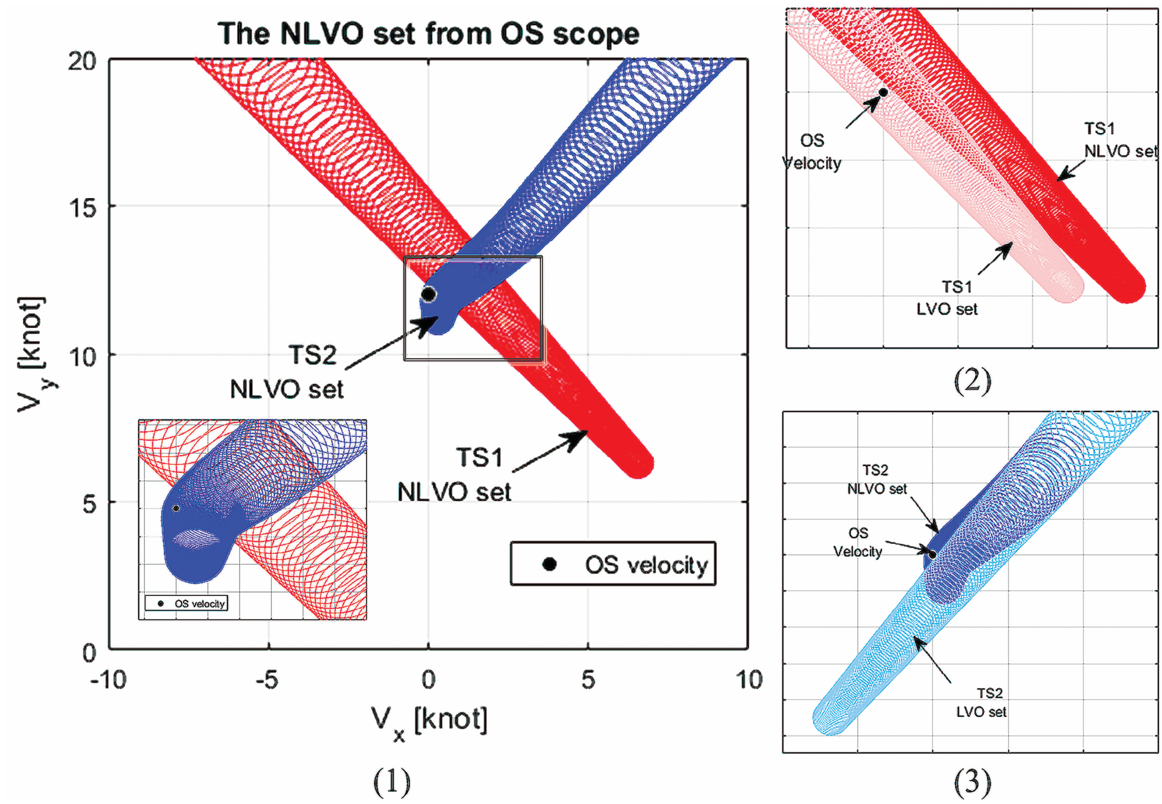


Figure 4.15 Scenario 4-II: VO sets of Multiple-ship encounter scenario at 0 [min].

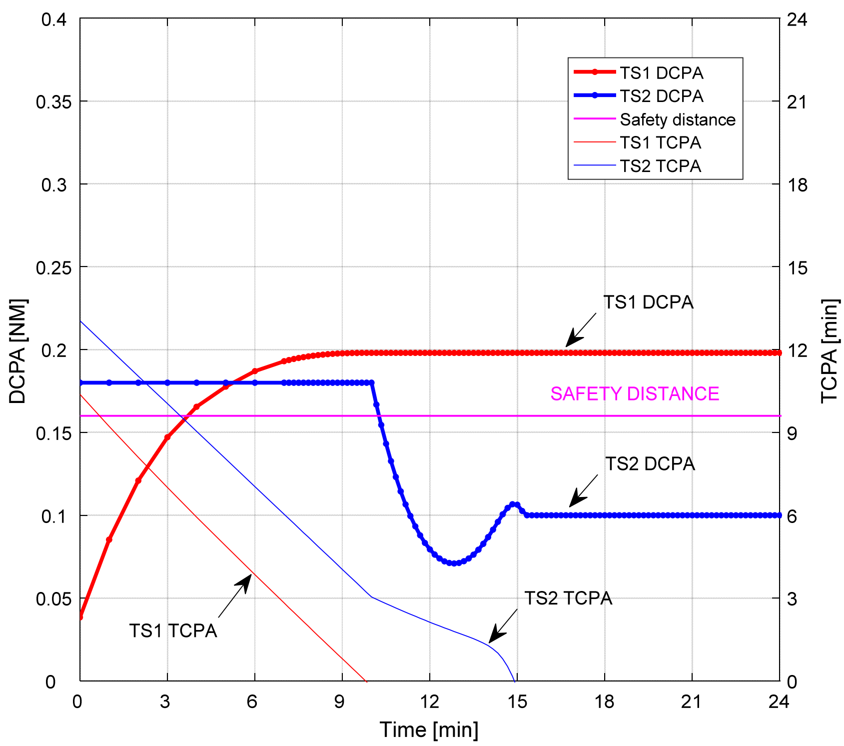


Figure 4.16 Scenario 4-II: evolution of DCPA and TCPA.

However, when the CTPA technique or CPA approach is applied, the opposite results occur. In Figure 4.15 (2) and (3), the CTPA (identical to LVO) sets caused by TS1 and TS2 are shown respectively. The OS's velocity falls in the TS1's CTPA set (in the light red region in Figure 4.15 (2)), but outside of TS2's CTPA set (the light blue region in Figure 4.15 (3)). It indicates that the OS will collide with TS1 and pass TS2 safely. Similarly, the DCPA and TCPA for TS1 and TS2 at time 0 implies the same result: the initial DCPA of TS1 is below the threshold (0.16 NM), while the initial DCPA of TS2 exceeds the safety distance (see Figure 4.16).

Figure 4.17 shows the relative distance between the OS and the TSs if the OS does not change its speed. The minimal distance between the OS and TS1 is larger than the safety distance, while the minimal distance between the OS and TS2 is below the safety distance at time 15 min. That is to say, if the OS keeps current velocity, the OS will pass TS1 safely but collide with TS2. Obviously, the argument made by CTPA and CPA at the beginning is wrong.

In fact, the CTPA and CPA approaches can identify collision dangers at the end, but the detected time is much later than the NLVO algorithm. Figure 4.16 presents the evolution of TSs' DCPA and TCPA. From the figure, the TS1 is seen as a threat at the beginning, but this mistake is corrected when its DCPA exceeds the safety threshold (around 4 min); On the other hand, the real threat (TS2) is seen to be safe at the beginning and is confirmed to be a danger at 15 min (4 min before the collision). In other words, in this scenario, the CPA based approach cannot reject the false alarm until 4-5 min before TS1 passes the CPA, and the real thread TS2 cannot be detected until 4 min before the collision.

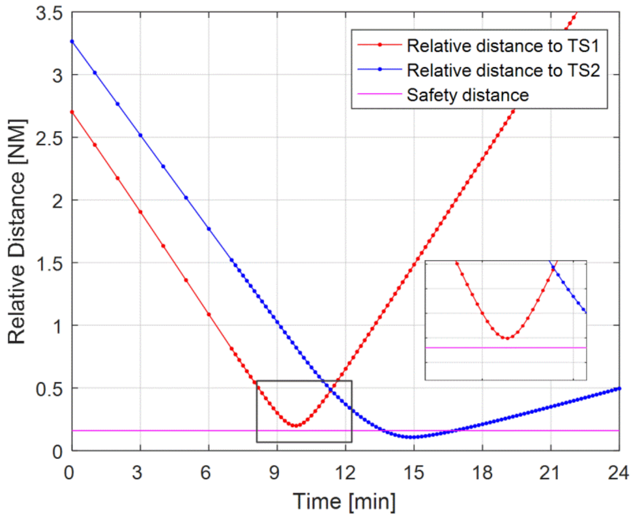


Figure 4.17 Scenario 4-II: relative distance between ships.

(3) Supporting collision avoidance in HMI-CAS

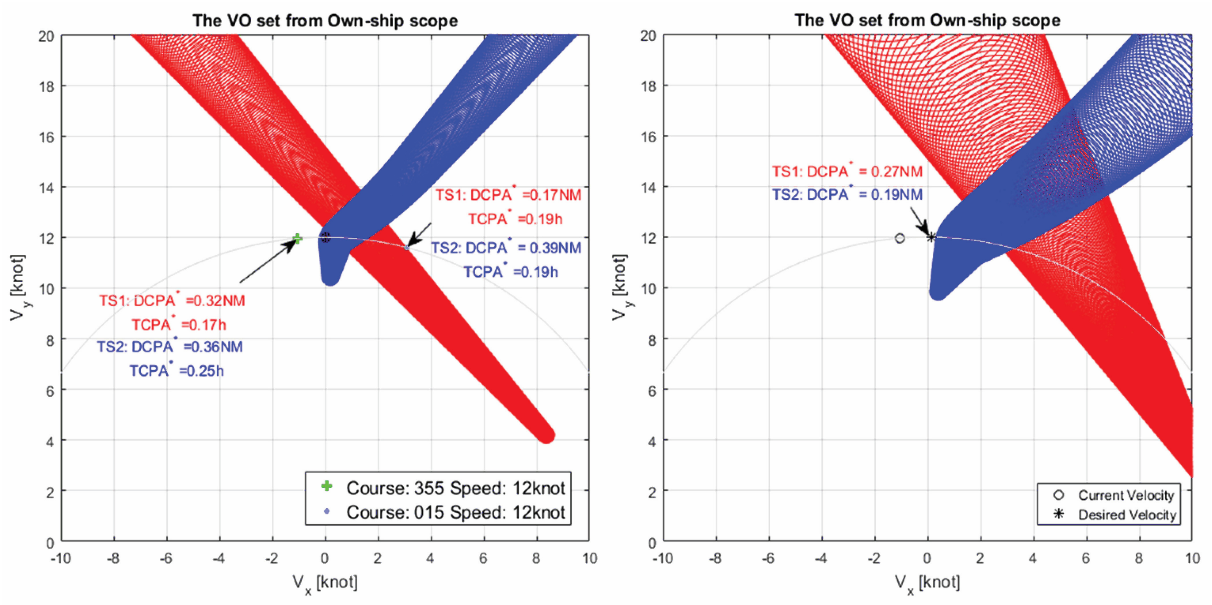
The VO algorithms can not only detect the dangers earlier than the CPA approach but also provide possible collision-free solutions to the OS. In this scenario, if the process in Section 4.3 is followed, the actions to avoid the collision with TS1 and TS2 can be determined. The reference velocity heading to the destination can be calculated by Equation (4.16). The steps include:

1. Danger detection. In the beginning, since the reference velocity and existing velocity are identical, and both fall in NLVO sets. Thus, the OS will have a collision with TS in the 30 min, and it needs to seek a collision-free velocity.

2. Choosing an evasive action: velocities outside the VO sets are the solutions. Two alternative velocities are chosen as examples, namely, port-side solution and starboard-side solution. As shown in Figure 4.18 (1), both velocities are the collision-free solutions for the OS. They have the same speed of 12 knots, the port-side one needs a port-side turn to 355, while the starboard-side solution needs the ship head to 015. The solution with less effort is chosen, i.e., the port-side one.

3. Return action: when the port-side turn is applied until 6.4 min, the desired velocity is outside of the NLVO sets, as shown in Figure 4.18 (2). That means the desired velocity will keep the relative distance larger than the set threshold (0.16 NM). Thus, the OS turns back to its desired velocity until it reaches the destination.

Figure 4.19 shows the evolution of relative distance between OS and TSs when the proposed evasive actions are applied. The minimum relative distance is larger than 0.16 NM. Therefore, the OS can pass the TSs safely, and the proposed velocity is a valid collision-free option.



(1) Interface at time 0 [min] (2) Interface at time 6.4 [min]

Figure 4.18 Scenario 4-III: selection of a collision-free velocity using interfaces.

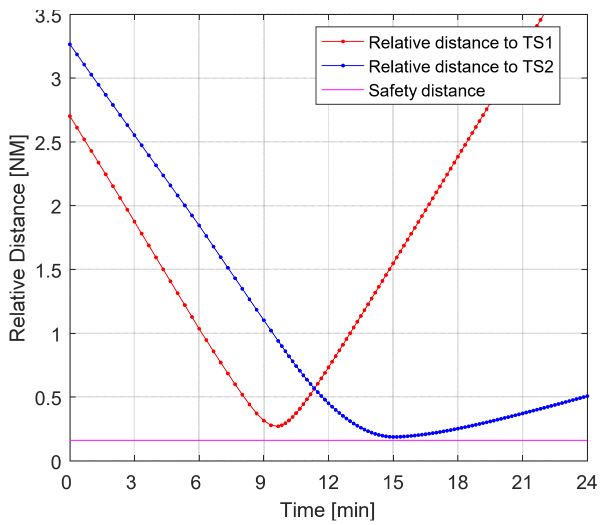


Figure 4.19 Scenario 4-III: relative distance between ships with evasive actions.

4.5.3 Scenario 4-III: encountering with a TS in channel intersection

In this section, a scenario is discussed in which stationary obstacles and the uncertainty of the TS's trajectories are introduced.

(1) Settings

The layout of this scenario is shown in the left panel of Figure 4.20. Two ships are involved, and three channels are connected in an intersection. One buoy is set at this intersection. The width of the channels is 2 NM, and the prohibited zone of the buoy is a circle with radius 0.25NM. The detail settings of ships are shown in Table 4.4. Besides, collision-warning time is set as 30 minutes for a moving obstacle and 10 minutes for a stationary obstacle, i.e., bank and buoy.

Table 4.4 Scenario 4-III: simulation settings

	Origin	Destination	Initial speed [knots]	Ship length [m]
OS	[0.5, -4]	[0.5, 6]	8.5	150
TS	[-0.5,3]	Unknown at the beginning	10	150

The trajectory of TS is assumed to be unknown at the beginning. However, the crew on the OS can predict several possible trajectories. In this simulation, four possible trajectories of the TS are employed. These trajectories are shown in different colors in Figure 4.20. At the beginning of the simulation, each trajectory has an equal probability of being chosen as the real trajectory. With updating observations, such as the position and velocity of the TS, the probabilities will be updated.

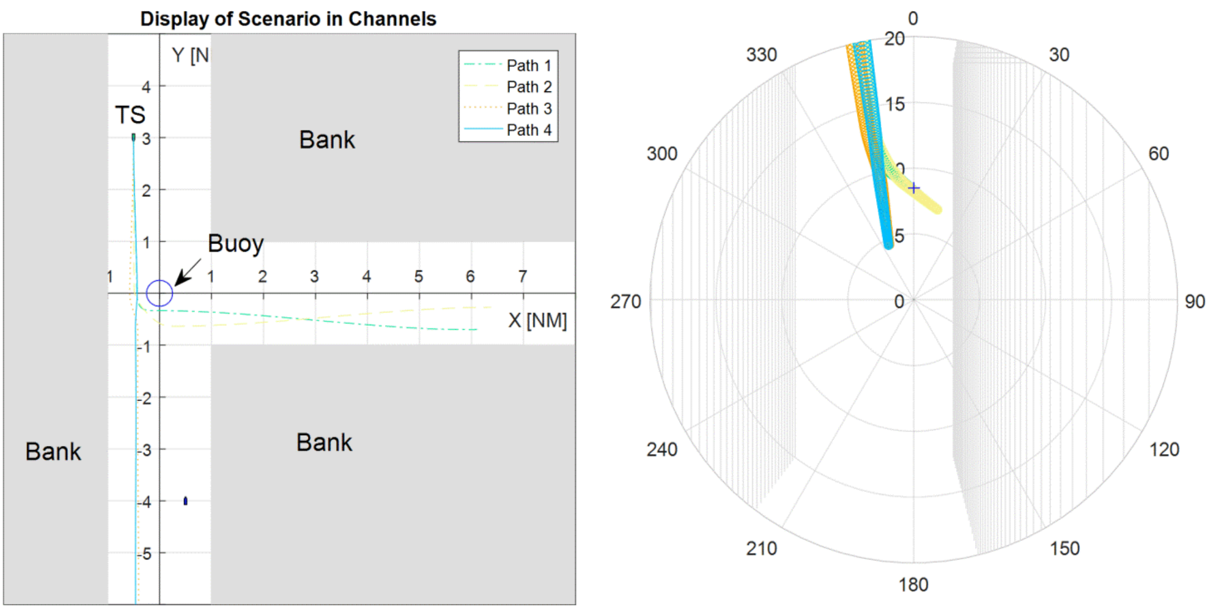


Figure 4.20 Scenario 4-III: encounter scenario and velocity space at 0 [min].

(2) Supporting collision avoidance in HMI-CAS using PVO sets

In this case, the PVO algorithm is used to support collision avoidance in HMI-CAS. When the OS's velocity falling in a PVO set, the OS might have two strategies:

1. Choosing a velocity outside of all PVO sets immediately which guarantees no conflict with the TS, i.e. $p\left(collision \mid \vec{v}_i \notin PVO\right) = 0$, or;

2. Taking the risk and maintaining current velocity, until the predictions are relatively accurate. To help mariners make a trade-off between these strategies, a probability threshold is needed. If the probability of collision is less than or equal to this threshold, the TS would not be seen as a thread temporarily, e.g., the threshold can be set as 50%. That means, the OS can keep existing velocity until $p\left(collision \mid \vec{v}_i \in PVO\right)$ exceeds 50%, and then, it will choose a velocity outside of all the PVO sets.

In this scenario, Strategy 2 is demonstrated in detail.

(3) Simulation results

The right panel in Figure 4.20 is enlarged and presented in Figure 4.21. The colorful sets in these figures represent the velocity might lead to a collision with TSs in 30 minutes, and; the gray regions are composed of velocities causing a collision with stationary obstacles in 10 min. The PVO set in Figure 4.21 is roughly divided into three parts: Zone 1, Zone 2, and Zone 3. A velocity in Zone 1 will result in a collision in 30 minutes, no matter how the TS chooses a trajectory; a velocity in Zone 2 will lead to a collision if the TS sails to the South; a velocity in Zone 3 might trigger a collision with the eastbound the TS.

At this moment (t = 0 min), the current velocity of the OS is labeled as "+" and falls in Zone 3. Additionally, this velocity is at the intersection of green and yellow NLVO sets, which implies that if the OS keeps this velocity, the collision with the eastbound TS will occur in 30 min. Since each trajectory has an equal chance at this stage, the $p\left(collision \mid \vec{v}_i \in PVO\right) = 50\%$. Thus, the OS can keep its existing velocity.

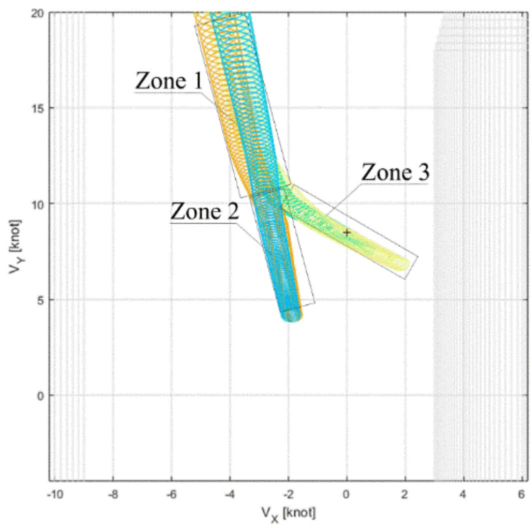


Figure 4.21 Scenario 4-III: enlarged V-space of the OS at time 0 [min].

At t = 20 min (Figure 4.22), the TS is turning to the East, which implies the probability of the TS sails to South drops to 0, and the probabilities of the eastern route are increased to 100%. Correspondingly, the probability of collision $p\left(collision \mid \vec{v}_i \in VO_{Path1}\right)$ reaches 100%, and evasive actions are needed. A velocity outside of the PVO set is an alternative for the OS, e.g., course 015 and speed of 8.5 knots is one feasible solution, which only requires a starboard turn. If this velocity at this stage is chosen, the change of the PVO set is shown in the left panel of Figure 4.23. The OS will keep this velocity until its desired velocity is outside of the PVO set. The desired velocity is calculated by equation (17) and labeled as "*" in Figure 4.23.

Five minutes later (at 25 min), the desired velocity that is [-0.17, 8.49], is outside of the PVO set. That means, the OS can choose its desired velocity, and the velocity will not lead to any collision. Moreover, the OS passes the TS from the TS's stem.

Figure 4.24 provides the relative distance between OS and TS with and without the proposed evasive actions. As it was shown, without any evasive actions, a collision occurs, and the proposed evasive actions are collision-free solutions for the OS.

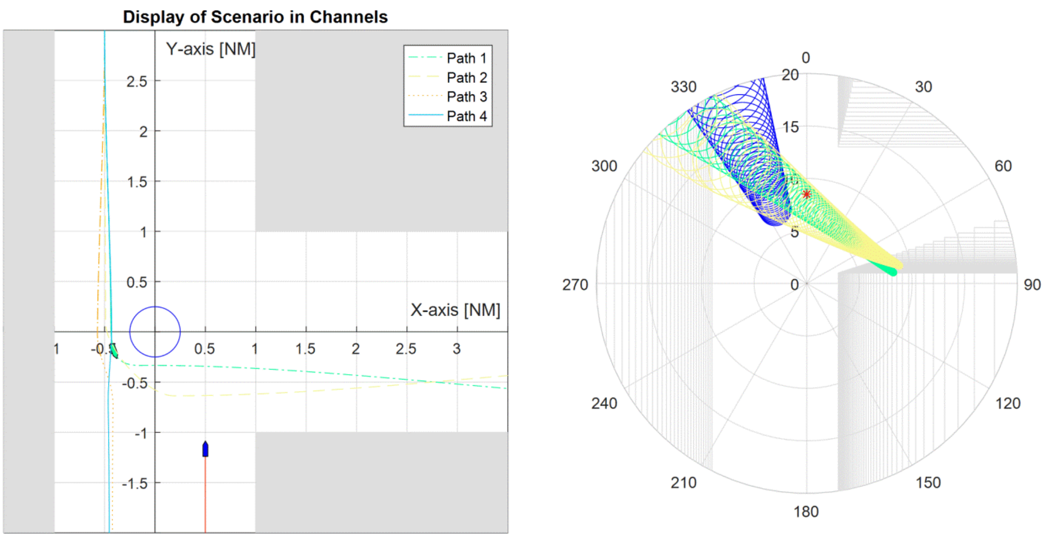
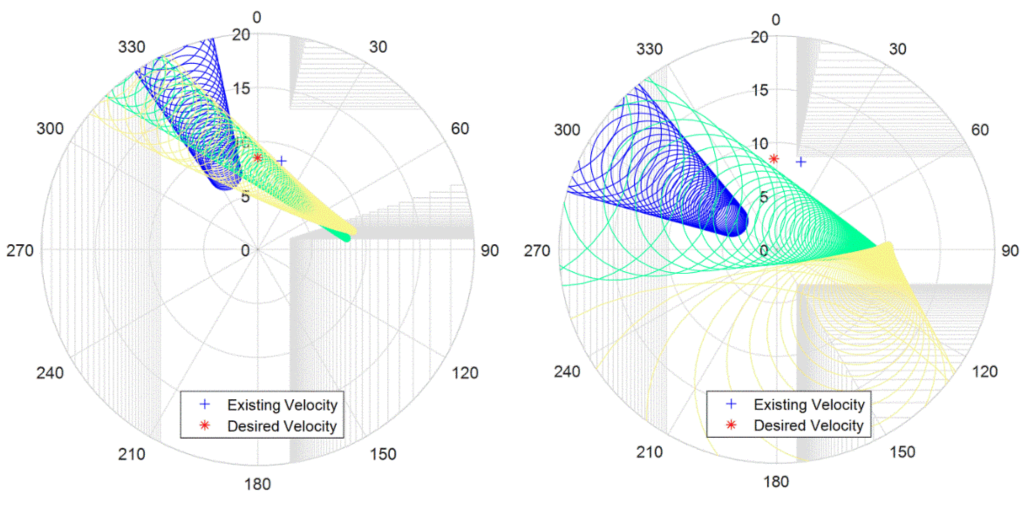


Figure 4.22 Scenario 4-III: encounter scenario at 20 [min] with V-space.



Time = 20 min *Time = 25.2 min*

Figure 4.23 Scenario 4-III: interface of V-space at 20 [min] and 25.2 [min].

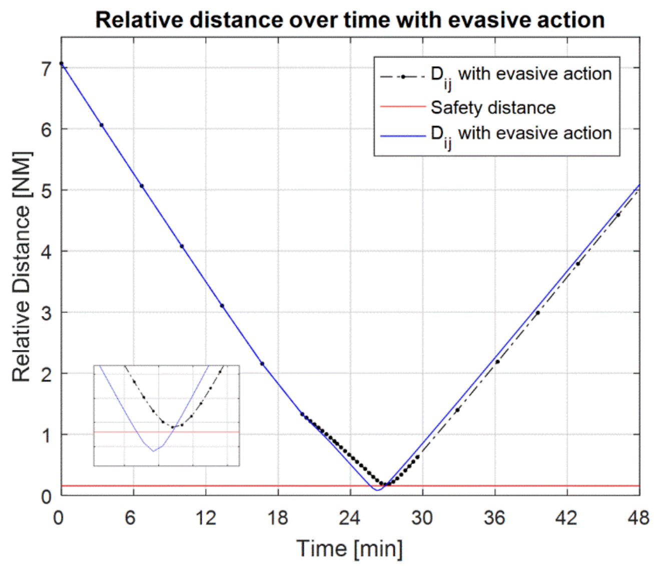


Figure 4.24 Scenario 4-III: relative distance between ships with/without evasive actions.

4.5.4 Discussion on the result of scenarios

In the previous sub-sections, Scenario 4-I~III show the scenarios when the OS encounters with a single ship or multiple ships whose trajectory are non-linear but known or predictable. Three techniques have been applied in these scenarios for comparison, namely VO algorithms, CPA and CTPA.

The results of simulations show that the traditional methods have two problems: firstly, they could not identify the real threats until 4-5 min before the collision; secondly, the indicator DCPA rises and falls several times, which would confuse mariners (see Figure 4.16 and Figure 4.13). On the other hand, VO algorithms can overcome these problems. They can detect collision risk earlier and show the potential collision-free solutions intuitively to mariners.

In summary, VO algorithms can be used to support collision avoidance at sea. Compared with CPA and CTPA, VO algorithms perform better in some aspects. Especially, if the motion of TS is changing, the results from CTPA and CPA are both unreliable even the motion is known or probabilistically predictable, while the VO algorithms can detect collision dangers early and reliably. Moreover, it is shown that the VO algorithm and its variations can help the captains to identify the dangerous solutions and facilitate them to find a collision-free solution. Therefore, it is believed that these VO algorithms have a wide scope of application in collision prevention at sea.

4.6 Discussion

Applying the VO algorithms for collision prevention in practice still faces several challenges, since the maritime environment is more complex than the simulator. Two main challenges and relevant improvements are highlighted in this section.

Firstly, the ships that the OS encounters might be cooperative or non-cooperative. For the non-cooperative ships, some of them have probabilistic trajectories. They are taken into account in the Probabilistic VO algorithm. Some non-cooperative ships' trajectories are fully uncertain, and the PVO set might be not reliable. The applied VO algorithms can only eliminate the velocity leading a collision with those cooperative/probabilistically predictable TSs and let the OS focus on preventing collision with TSs have fully uncertain trajectories. To solve this problem, integrating prediction techniques with the PVO algorithm is necessary. For instance, in [187], the Bayesian filter is integrated into the PVO algorithm to support collision avoidance with the cars with uncertain trajectories. This combination would be considered in future work.

Secondly, VO algorithms figure out collision-free velocities, but these velocities are indifferent. Specifically, the VO algorithms cannot tell the mariner which solution is safer or more fuel-efficient. A possible solution is combining the VO algorithms with optimization theory. The collision prevention problem then can be seen as an optimization problem. The cost function could be the number of course alternations, the consumption of oil, total travel time, etc. The VO set can be seen as a constraint to bound the feasible solutions. In this way, more practical situations can be considered. For instance, the ship maneuverability and collision regulations can be considered as additional constraints. By minimizing the cost function subject to constraints, the human operators can find the best solutions which can achieve their objectives while guaranteeing the safety. The relevant developments are shown in Chapter 5.

Lastly, the VO algorithms presented in this chapter ignore the dynamics of the ship, which might result in system failure when two ships get too close. To handle this issue, incorporating the dynamics model in VO algorithm is needed. The relevant developments are shown in Chapter 5.

4.7 Conclusions

Chapter 4 is drawn to find out proper algorithms that stratify the demands on human-machine interactions that are addressed in Chapter 3. Specifically, this chapter is to answer the research question: *What are the proper methods that support human-machine interactions in conflict detection and conflict resolution of HMI-CAS?*

The answers are concluded as follows:

- By overviewing conflict detection methods, it is found that many methods for conflict detection are based on the concept of Closest Point of Approach (CPA) that is widely used. However, the CPA method has a strict assumption that the TS is semi-dynamic, i.e., keeps speed and course. In return, the methods relying on CPA easily fail when the TS changes its movements.

- By reviewing conflict resolution methods, VO algorithms have been focused. In existing conflict resolution algorithms, only a few algorithms can facilitate human-machine interactions addressed in Chapter 3. They are Dynamic Window (DW), Decision Disc (DD), and VO algorithm. DW might not be suitable for a dynamic environment since it assumes that a stop of the vessel is always a safe solution. Moreover, DW and DD methods are relying on the discretization of the solution-space. Thus, the solution between the grids is not checked by these methods. It means that the operators cannot validate the safety of an arbitrary solution, but choose the one from the grids.

- Velocity Obstacle (VO) algorithm and its variations are the ones suitable for supporting human-machine interactions. Firstly, many variations of VO algorithms enable to detect the collision dangers when the movement of the target-ship is changing; Secondly, it not only offers one solution to users but a velocity space that facilitate human operators.

In this chapter, Linear VO (LVO) algorithm, Non-Linear VO (NLVO) algorithm, and Probabilistic VO (PVO) algorithm have been introduced and applied in the proposed HMI-CAS. By showing the velocity space to users, the HMI-CAS enables to support the operators to detect collision dangers, to find proper evasive solutions, and to cooperate with the automation system.

By comparing the introduced VO algorithms with other existing techniques, some features of VO algorithms are highlighted: (1) CTPA method from traditional maritime studies is identical to the LVO algorithm, while VO algorithms have wider applications due to their variations; (2) VO algorithms can detect collision dangers, no matter the trajectory of the target ship is linear or non-linear, deterministic or probabilistic; (3) VO algorithms perform better to find collision-free velocities in multiple-ships scenarios compared to CPA methods; (4) VO algorithms can easily handle large and irregular size of obstacles that are challenging for CPA methods.

Three encounter scenarios are simulated to compare VO algorithms with traditional methods and demonstrate the usage of VO algorithms in HMI-CAS. The results show that VO algorithms are suitable to avoid collision with non-linear and predictable TSs. They can help mariners detect collision earlier and easier. Moreover, they can offer collision-free solutions for mariners.

Applying the HMI-CAS using VO algorithms in practice still faces some challenges. Firstly, the dynamics of the ship is neglected, which is unrealistic in practice. In some extreme situations, the system might go to failure. Secondly, the HMI-CAS can identify collision-free solutions, but it cannot determine which one is optimal. These two issues would be improved in Chapter 5. Thirdly, if the human operators decide to tolerate the risk in HMI-CAS, what is the bottom line, i.e., how can the operators be reminded to take actions in time. This question is related to research question 5, which is answered in Chapter 6.

Chapter 5 Generalized Velocity Obstacle Algorithm for Conflict Resolution in HMI-CAS

As many CASs, the HMI-CAS developed in Chapter 4 ignore the dynamics of the own-ship and ignore the navigational regulations in conflict resolution. However, in close range the ignorance of ship maneuverability might lead to failure of the CAS. To handle these issues, this chapter proposed to employ the modified Generalized Velocity Obstacle (GVO) algorithm that incorporates the ship's dynamics and is suitable for the proposed HMI-CAS.

The structure of Chapter 5 is organized as follows. The backgrounds and motivation are presented in Section 5.1, followed by the introduction of ship dynamics and controller design in Section 5.2. The details about the modified GVO algorithm are shown in Section 5.3, and the application of GVO algorithm in HMI-CAS is presented in Section 5.4, followed by simulation experiments in Section 5.5. Section 5.6 discusses the major findings, and Section 5.7 concludes this chapter.

Acknowledgment Part of the content in this chapter is based on the following papers:

Y. Huang, L. Chen & P. H. A. J. M. van Gelder. 2019. Generalized velocity obstacle algorithm for preventing ship collisions at sea. Ocean Engineering, 173, 142-156.

L. Chen, Y. Huang, H. Zheng, J.J. Hopman & R. R Negenborn. 2019. Cooperative multi-vessel systems in urban waterway networks. IEEE Transactions on Intelligent Transportation Systems, 99, 1-14.

Y. Huang, L. Chen, R.R. Negenborn, P.H.A.J.M. van Gelder. "A HMI-CAS for human-machine cooperation during collision avoidance". (Submitted)

5.1 Introduction

Many techniques have been developed for preventing collisions at sea. In the body of literature, most research has many assumptions and are applied to specific scenarios, such as relatively simple dynamics of the ship (e.g., holonomic vehicle), conflicts involving only a pair of ships, and constant-velocity of target-ships. Details can be found in Chapter 2. Those assumptions may make the methods not suitable for collision prevention in more general collision avoidance cases. Moreover, most methods only provide one solution, see Table 4.1. If those methods function as navigation assistance for human-operated ships, the operators have no information about the decision processes. Consequently, they do not have choices but to accept the proposed solution.

A family of Velocity Obstacle (VO) algorithms presented in Chapter 4 bridges some of these gaps: VO algorithms enable to resolve conflicts with multiples moving obstacles (static and/or dynamic); the algorithms collect all the velocities that result in collisions and present a set of collision-free velocities for human/machine, which facilitates the search for the best option. Today, these algorithms have been widely used for collisions prevention of various vehicles, e.g., car-like robots [188], airplanes [189], unmanned underwater vehicles [190], etc.

5.1.1 VO studies on maritime studies

The advantages of the VO algorithms have been noticed by researchers in maritime engineering, while researchers used different names, such as Collision Threat Parameter Area (CTPA) in [31], Collision Danger Sector in [32], and methods proposed in [30] and [29]. The early version of the VO algorithm at sea is noted as CTPA that was proposed in the 1980s [30, 31]. These methods share the same characteristics with LVO algorithm [104]: firstly, they all collect velocities which satisfy certain conditions (leading to collisions), i.e., DCPA is smaller than a threshold; secondly, the moving obstacles keep constant velocity; thirdly, the own-ship can adopt new course and speed simultaneously.

The LVO algorithm is popular in the maritime domain, due to its advantages over other methods. For one reason, the LVO algorithm can provide identical conflict detection results as DCPA/TCPA method. Moreover, it also provides the operators the conflict resolution. In particular, the LVO set can be integrated into a Radar system, which could make collision prevention more intuitive for the Officer On Watch (OOW). Pedersen, Inoue [32] showed that this method could provide better support for the OOW in collision prevention comparing with traditional Automatic Radar Plotting Aid (ARPA). Later on, a series of studies proposed to use VO/CTPA algorithms for collision avoidance in various scenarios, e.g., restricted waters [34], multiple-ship [131], incorporating with regulations [79], and unmanned ship [73]. However, the application is still restricted due to its assumptions on the motion of obstacles. To overcome this issue, the NLVO and PVO have been introduced in Chapter 4, which expands its applications in the maritime field. Moreover, researchers found that NLVO also facilitates the detection of collision candidates from historical data for risk analysis, see reference [191].

VO algorithms also attract much attention in the studies on ASVs. Many researchers have used the VO algorithms and tested them with ASV prototypes. Benjamin, Leonard [29] used this method to eliminate dangerous solutions and search for an optimal collision-free velocity for ASVs. In [73], the VO algorithm incorporated with COLREGs is proposed to obtain rule-compliant solutions. Additionally, Zhao, Li [79] combined the VO algorithm with an experts' judgment system to meet regulations and seamanship.

5.1.2 Motivation

The main challenge of using VO algorithms in ship collision prevention is the violation of the holonomic assumption in ship dynamics, i.e., the ship can change speed and course simultaneously. This assumption might not influence the performance of the VO algorithm when the distances between ships are large enough. However, in close range scenarios, this assumption might fail collision prevention (as shown in Section 5.5.2). Thus, this chapter aims at finding a suitable VO algorithm which discards the holonomic assumption and is capable of handling the ship's dynamics. A Generalized Velocity Obstacle (GVO) algorithm based on [182] is used and modified for collision prevention in HMI-CAS. The main contributions of this chapter are as follows:

(1) A modified GVO algorithm considering the dynamics of ships is used for collision prevention in the maritime environment, which can be used to support collision prevention with multiple dynamic obstacles;

(2) An optimization process is included to support the HMI-CAS to choose an optimal collision-free solution considering a predefined cost function.

5.2 Ship dynamics and controller design

5.2.1 Vectorial representation of ship dynamics

The vectorial representation of ship dynamics introduced by Fossen [54] is used in this chapter, which is formulated as:

$$\begin{cases} \dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}\dot{\mathbf{v}} = -\mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} + \boldsymbol{\tau} \end{cases}, \tag{5.1}$$

where the system state is $\mathbf{x} = \begin{bmatrix} \boldsymbol{\eta}^{\mathrm{T}} & \mathbf{v}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, and system input is $\boldsymbol{\tau}$.

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}, \mathbf{v} = \begin{bmatrix} u \\ v \\ r \end{bmatrix}, \ \boldsymbol{\tau} = \begin{bmatrix} \tau_u \\ \tau_v \\ \tau_r \end{bmatrix}.$$

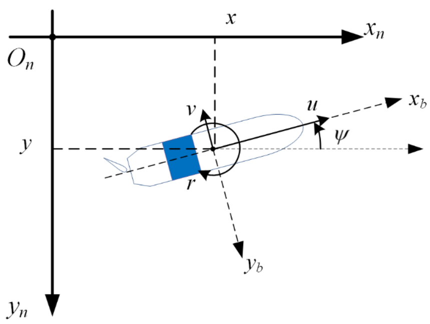


Figure 5.1 Illustration of inertial frame $\{n\}$ and the body frame $\{b\}$ for a ship

$\boldsymbol{\eta}$ contains coordinates and heading; $\mathbf{v}$ is a velocity vector consisting of linear velocities (surge, sway) and angular velocity (yaw). The control input vector is composed of surge force, sway force and yaw force. **M** is a system inertia matrix (including added mass). **C**(**v**) is the Coriolis–centripetal matrix and **D**(**v**) is a damping matrix. $\mathbf{R}(\psi)$ is a rotation matrix which is formulated as:

$$\mathbf{R}(\psi)=\begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}.$$

Equation (5.1) also can be written as:

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}\dot{\mathbf{x}}=\begin{bmatrix}\mathbf{R}(\psi)\mathbf{v}\\ -\mathbf{C}(\mathbf{v})\mathbf{v}-\mathbf{D}(\mathbf{v})\mathbf{v}\end{bmatrix}+\mathrm{B}\boldsymbol{\tau}, \tag{5.2}$$

where, $\mathrm{B}=\begin{bmatrix}\mathbf{0}^{3\times3} & \mathbf{I}^{3\times3}\end{bmatrix}^{\mathrm{T}}$, $\mathbf{I}^{3\times3}$ is a 3-by-3 identical matrix and $\mathbf{0}^{3\times3}$ is a 3-by-3 zero matrix.

5.2.2 Design of controller

A controller is designed to control the state of the system with respect to reference velocity. Well known controllers include Proportional-Integral-Derivative (PID) controllers, robust H_∞ controllers, fuzzy controllers, etc. [192]. A comparison using P, PD, PID controller in HMI-CAS is presented in Figure 5.2, where the system tracks an inputted velocity (surge speed: 0.1 m/s, sway speed: 0 m/s, and heading 058).

From these figures, it is found that the P controller might lead to an oscillating behavior, e.g., see panel (1) in Figure 5.2. PD and PID controllers help to eliminate the oscillatory. However, the PID controller suffers from an integrator wind-up problem, especially when the system is nonlinear (our system is nonlinear), see (3) in Figure 5.2. When the ship is turning, the sway-speed increases and departs from 0. The integral term in the PID controller, then, accumulates these errors and leads to an increasing overshoot. As a result, it takes a longer time for the PID controller to let the sway-speed converge to 0. To avoid this problem, a further careful tuning the PID controller is needed. On the other hand, the PD controller is relatively simple and converges quickly. It provides a satisfying performance in velocity tracking, which satisfies our demand for this research. Thus, a PD controller is used in this HMI-CAS.

With a PD controller, the input of Equation (5.2) is formulated as:

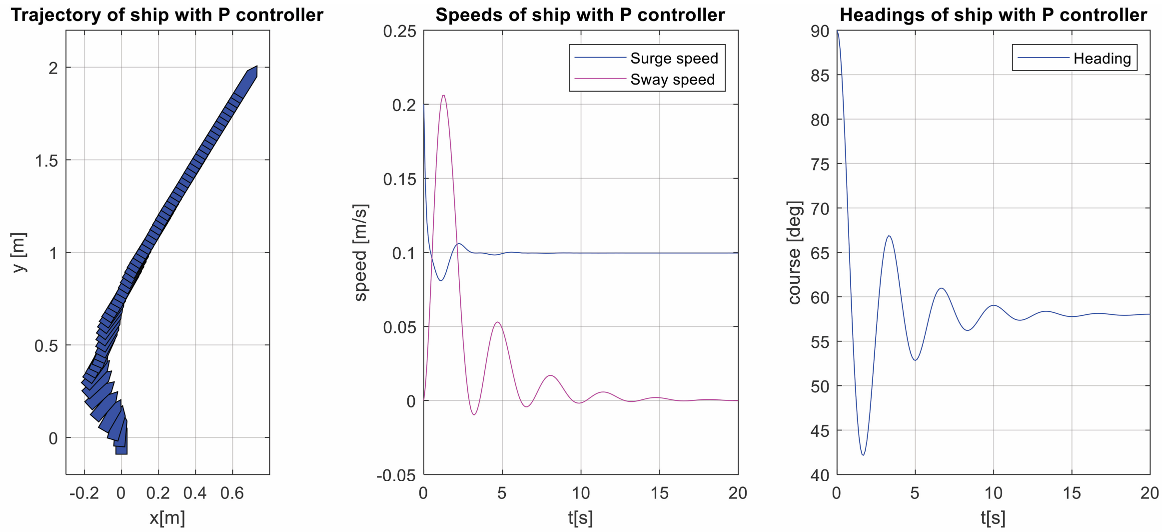
$$\boldsymbol{\tau}=\mathrm{K_p}\left(\mathbf{u}^*-\mathrm{V}\mathbf{x}\right)-\mathrm{K_d}\mathrm{V}\dot{\mathbf{x}}, \tag{5.3}$$

where K_p and K_d are non-negative feedback gains; V is the observe matrix and $\mathbf{u}^*$ is the desired velocity:

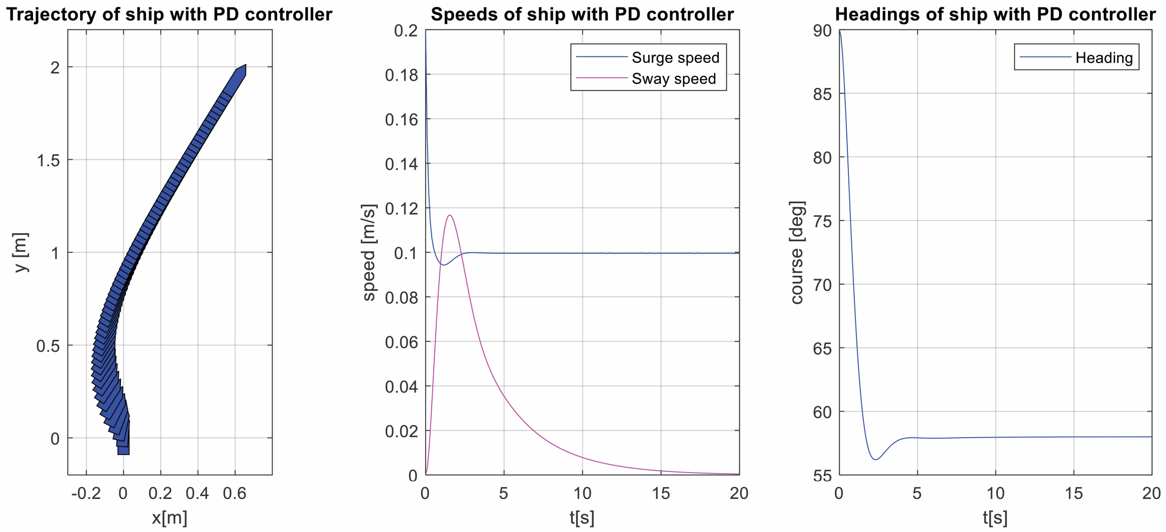
$$\mathbf{u}^*=\begin{bmatrix}u^*\\ v^*\\ \psi^*\end{bmatrix},\ \mathrm{V}=\begin{bmatrix}0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&1&0&0&0\end{bmatrix}.$$

By substituting Equation (5.3) into Equation (5.2), the control input is changed from the force $\boldsymbol{\tau}$ to the desired velocity $\mathbf{u}^*$:

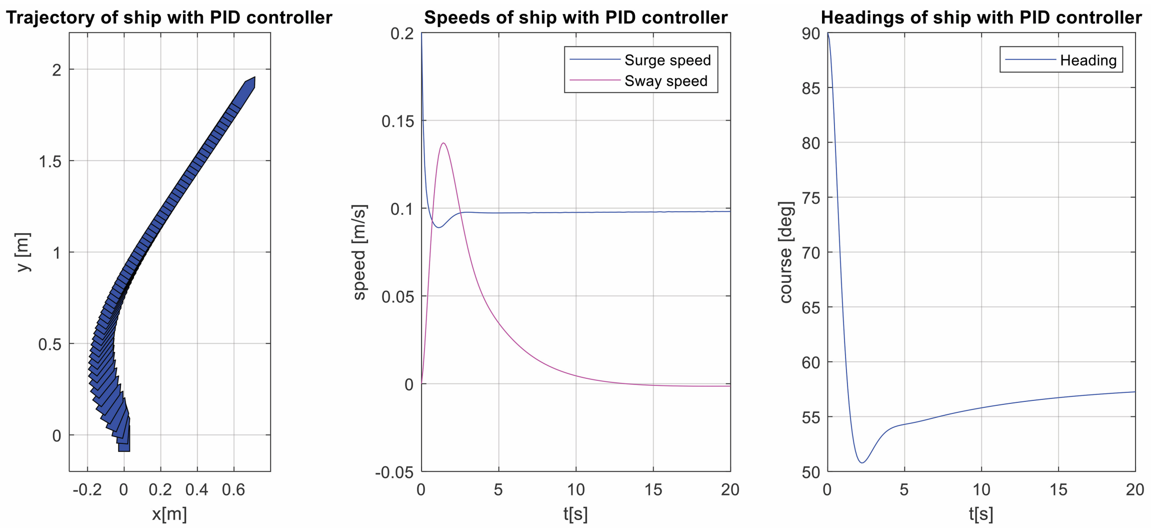
$$\left(\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}+\mathrm{BK_d V}\right)\dot{\mathbf{x}}=\begin{bmatrix}\mathbf{R}(\psi)\mathbf{v}\\ -\mathbf{C}(\mathbf{v})\mathbf{v}-\mathbf{D}(\mathbf{v})\mathbf{v}-\mathrm{K_p}\mathrm{V}\mathbf{x}\end{bmatrix}+\mathrm{BK_p}\mathbf{u}^*. \tag{5.4}$$



(1) P controller with $K_p = \mathrm{diag}([200, 10, 10])$



(2) PD controller with $K_p = \mathrm{diag}([200, 10, 10])$ and $K_d = \mathrm{diag}([5, 5, 5])$



(3) PID controller with $K_p = \mathrm{diag}([200, 10, 10])$, $K_I = \mathrm{diag}([1, 1, 1])$, and $K_d = \mathrm{diag}([5, 5, 5])$

Figure 5.2 Comparison of using P, PD, or PID controller for tracking the desired velocity.

5.3 Generalized velocity obstacle algorithm

In this section, the basic VO algorithm and GVO algorithm are introduced, respectively. Notations are presented as follows: Sub-index $\cdot_i$ indicates the own-ship (OS) and Sub-index $\cdot_j$ refers to the target-ship (TS). The state of the system is denoted as a vector **x** which is in bold. The position and velocity are denoted as vectors **P** and **v**. The superscript index $\cdot^k$ indicates the value at time *k*. For example, $\mathbf{x}_i^k$ is the state of the OS at time *k*.

5.3.1 Foundation of velocity obstacle algorithms

A collision occurs at time *t*, if and only if $\mathbf{P}_i(t)$ is inside $\mathbf{P}_j(t) \oplus ConfP$. An illustration is shown in Figure 5.3. If the OS keeps its existing velocity, a collision will happen at time *t*, i.e., Equation (4.2) is satisfied. Since there is still time before the collision, the OS can apply a new $\mathbf{v}_i$ to change its position at time *t* and to prevent the collision. The relation of the new $\mathbf{v}_i$ and new position $\mathbf{P}_i(t)$ is described in Equation (5.5). The idea of VO algorithm is using this relation to collect all the $\mathbf{v}_i$ leading to the position of the OS falling in $\mathbf{P}_j(t) \oplus ConfP$, and the rest of velocities are safe for the OS at time *t*.

If the OS is a holonomic vehicle, the relation between position $\mathbf{P}_i(t)$ and input $\mathbf{v}_i$ is:

$$\mathbf{P}_i(t) = \mathbf{P}_i(0) + \mathbf{v}_i \cdot t \,. \tag{5.5}$$

Those $\mathbf{v}_i$ leading to a collision at time *t* is collected in a sub-VO set that is formulated by substituting Equation (5.5) to Equation (4.2), i.e., Equation (4.4).

Then, a set of velocities leading to a collision at any time in the future is a union of all the sub-VO sets in the future, i.e., Equation (4.5). If the OS keeps a velocity in this VO set, a collision will happen in the future.

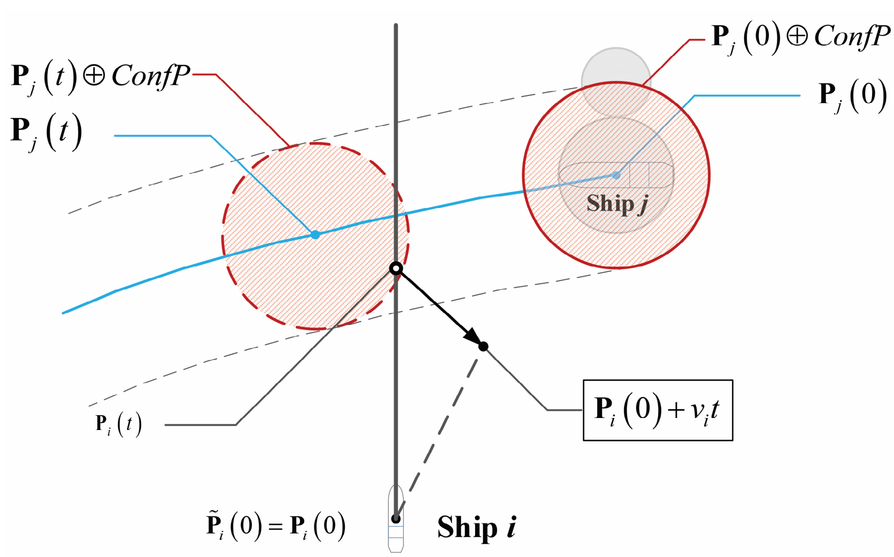


Figure 5.3 Schematic sketch of the VO algorithm.

5.3.2 Generalized velocity obstacle algorithm

The GVO algorithm is one of the VO algorithms, which is proposed to consider vehicle dynamics and generalize the basic VO algorithm. The GVO algorithm presented in this section is based on the original GVO from the reference [182], the trajectory of other agents are assumed to be known (Section 3.4.1). The difference between the original GVO and the modified GVO in this chapter can be found in Section 5.6.2.

Instead of collecting velocities leading to collisions, the GVO algorithm collects a set of controls leading to collisions in the future that is named as UO set. Although the holonomic assumption is no longer held, the construction of UO set follows the same line of thinking as that of VO set presented in Section 5.3.1. The algorithm firstly formulates the **position of the OS** at time *t* with respect to **control inputs** and the dynamics of ships. This step is the most challenging part due to the nonlinearity of ship dynamics. Subsequently, a set of controls leading to a collision at time *t* is found, named as a sub-UO set, i.e., sUO set. Then, the union of sUO sets is the UO set, i.e., a set collecting all the controls of the OS resulting in collisions.

Let the dynamics of the OS be described as a nonlinear ordinary differential equation (ODE):

$$\dot{\mathbf{x}}_i = \mathbf{f}_i\left(\mathbf{x}_i, \mathbf{u}_i\right), \tag{5.6}$$

where $\mathbf{u}_i$ is control input, $\mathbf{f}_i$ is a continuous-time equation of motion and $\mathbf{x}$ the system state containing ship position **P**, velocity, etc. Thus, the position of the ship is obtained from the system state via:

$$\mathbf{P}_i\left(t\right) = C \cdot \mathbf{x}_i\left(t\right), \tag{5.7}$$

where $C = [\mathbf{I}^{2\times2}, \mathbf{0}^{2\times4}]$ contains a 2-by-2 identical matrix and a 2-by-4 zero matrix.

The GVO algorithm consists of the following steps:

Step 1: the mathematical expression of the system $\mathbf{x}_i(t)$ in terms of control input $\mathbf{u}_i$.

Due to the nonlinearity of the system, a linearization of this system about an initial trajectory is needed. The initial trajectory is a trajectory of the system without additional inputs, noted as $\tilde{\mathbf{x}}_i$, which is calculated via Runge-Kutta Integration with known initial state $\mathbf{x}^0$ and input $\mathbf{u}^0$. Figure 5.4 shows the initial trajectory of the OS ($\tilde{P}_i$) in the solid green line. In this way, the state of the system can be formulated as:

$$\mathbf{x}_i\left(t\right) \approx \int_0^t \mathbf{f}_i\left(\mathbf{x}^0, \mathbf{u}^0\right) d\tau + \int_0^t \Delta\dot{\mathbf{x}}_i\left(\tau\right) d\tau = \tilde{\mathbf{x}}_i\left(t\right) + G\left(t\right)\Delta\mathbf{u}_i, \tag{5.8}$$

where $\Delta\mathbf{u}_i$ is the difference between $\mathbf{u}^0$ and real input $\mathbf{u}_i$, i.e. $\Delta\mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_i\left(0\right)$, and $G\left(t\right) = \int_0^t e^{A(t-\tau)} B d\tau$, with $A = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^0,\mathbf{u}^0}$ and $B = \left.\frac{\partial \mathbf{f}}{\partial u}\right|_{\mathbf{x}^0,\mathbf{u}^0}$. When the initial input $\mathbf{u}^0$ is kept, i.e., $\Delta\mathbf{u}_i = \mathbf{0}$, the trajectory of the OS is equal to the initial trajectory ($\tilde{P}_i$ in Figure 5.4). When $\Delta\mathbf{u}_i \neq \mathbf{0}$, the position of the OS is pushed away from $\tilde{P}_i$. The new position is calculated via Equation (5.7) and Equation (5.8), i.e., $\tilde{P}_i + CG\left(t\right)\Delta\mathbf{u}_i$.

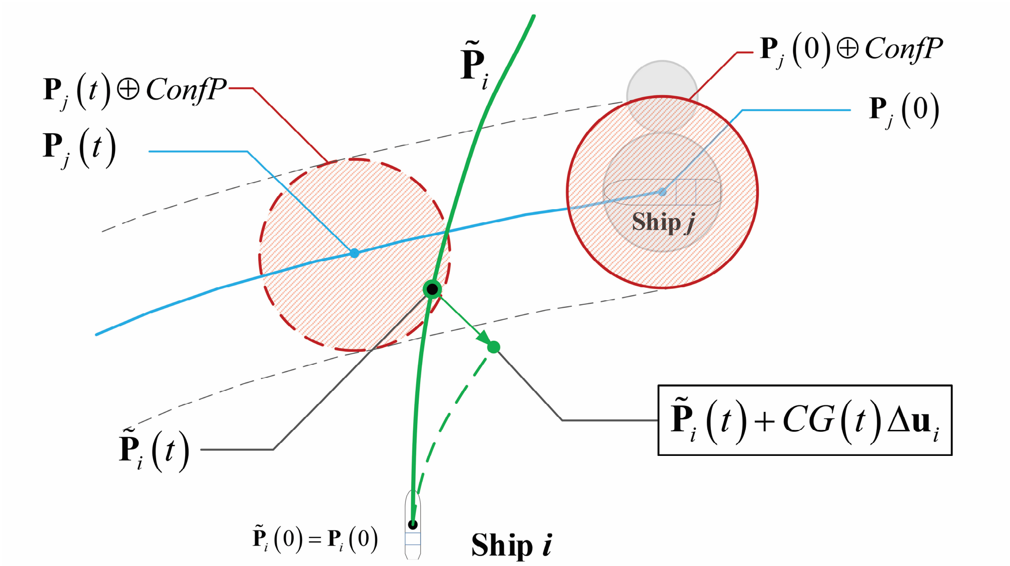


Figure 5.4 Schematic sketch of the GVO algorithm.

Step 2: Formulation of the sUO set.

A sUO(t) set collects all the $\Delta\mathbf{u}_i$ resulting in the position of the OS at time t inside the $\mathbf{P}_j(t) \oplus ConfP$, which can be formulated as:

$$C\tilde{\mathbf{x}}_i(t)+CG(t)\Delta\mathbf{u}_i \in \mathbf{P}_j(t) \oplus ConfP\,. \tag{5.9}$$

Solving this equation, regarding $\Delta\mathbf{u}_i$, leads to:

$$\Delta\mathbf{u}_i \in \left(CG(t)\right)^{-1} \cdot \left[-\left(\tilde{\mathbf{P}}_i(t)-\mathbf{P}_j(t)\right) \oplus ConfP\right] = \mathrm{sUO}(t)\,. \tag{5.10}$$

This equation collects all $\Delta\mathbf{u}_i$ leading to a collision at time t. It means that a sufficient and necessary collision condition will be satisfied (i.e., Equation (4.2)) if $\Delta\mathbf{u}_i$ falls in this set and the ship keeps the control $\left(\mathbf{u}_i^0+\Delta\mathbf{u}_i\right)$ till time t.

Step 3: Construction of UO set.

The UO set is a generalized VO set, which contains the changes of control leading to collisions. Therefore, any $\Delta\mathbf{u}_i$ outside this set is collision-free. By combining all these sUO sets, the UO set is obtained:

$$\mathrm{UO} = \bigcup_t^{\infty} \mathrm{sUO}(t)\,. \tag{5.11}$$

5.3.3 From GVO algorithm to VO algorithms

GVO enables to consider a non-holonomic system and holonomic system. When the vehicle is holonomic, the outputs from GVO algorithm is identical to NLVO algorithm in Section 4.2.3. For instance, when the vehicle is a holonomic vehicle, i.e., $\dot{x}_i(t) = f(x_i, v_i) = v_i$.

Step 1: $x_i(t) = v_i^0 t + \Delta v_i t = \tilde{\mathbf{P}}_i(t) + \Delta v_i t$;

Step 2: $\mathrm{sUO} = \frac{1}{t}\left[-\left(v_i^0 t - \mathbf{P}_j(t)\right) \oplus ConfP\right] = -v_i^0 + \frac{1}{t}\left[\mathbf{P}_j(t) \oplus ConfP\right]$;

Step 3: $\mathrm{UO} = \bigcup_t^\infty \mathrm{sUO}(t) = -v_i^0 + \bigcup_t^\infty \frac{1}{t}\left[\mathbf{P}_j(t) \oplus ConfP\right]$.

Then, the sufficient and necessary condition of collision is:

$$\Delta v_i \in \mathrm{UO} \Leftrightarrow \Delta v_i \in -v_i^0 + \bigcup_t^\infty \frac{1}{t}\left[\mathbf{P}_j(t) \oplus ConfP\right] \Leftrightarrow v_i \in \bigcup_t^\infty \frac{1}{t}\left[\mathbf{P}_j(t) \oplus ConfP\right] \Leftrightarrow v_i \in \mathrm{NLVO}.$$

That means the result of GVO is identical to NLVO when the vehicle is considered as a holonomic vehicle. Moreover, according to Section 4.1.1, the NLVO is identical to LVO when the trajectory of the obstacle is invariant. Thus, GVO is considered to be a generalized expression of a family of VO algorithms.

5.4 Using GVO algorithms in HMI-CAS

Different from VO algorithms, GVO algorithm incorporates the dynamics of the ship into conflict resolution, which enables the system to offer a solution that is feasible and collision-free.

In the proposed HMI-CAS system, the dynamics of the ship is assumed to be known, and the PD controller is employed as a low-level controller tracking the desired velocity. The ship with the low-level controller forms a new system whose dynamics is described in Equation (5.4). The GVO algorithm is assigned as a high-level controller, which finds the desired velocity tracked by the low-level controller.

To find the desired velocity leading to the ship collisions, Equation (5.4) is substituted in Equation (5.6) and the steps from Equation (5.6) to (5.11) are followed. Then, a set of changes in the desired velocity, i.e., $\Delta\mathbf{u}^*$, leading to collisions can be identified as a UO set. Since the VO set and UO set are different, some modules in the HMI-CAS need different designs.

5.4.1 Design of interface

An illustration of the interface using UO sets is presented in Figure 5.5 (1), which shows whether the ship is in danger and how the HMI-CAS assists users to avoid collisions. The point in this space is the changes of control input that refers to the difference between the chosen desired velocity and the initial desired velocity $\mathbf{u}^0$, i.e., $\Delta\mathbf{u}^* = \mathbf{u}^* - \mathbf{u}^0$. Thus, this space is also called ΔU space. The red regions are the UO sets, and the controls outside of these sets are collision-free. However, two modifications are made to facilitate human users.

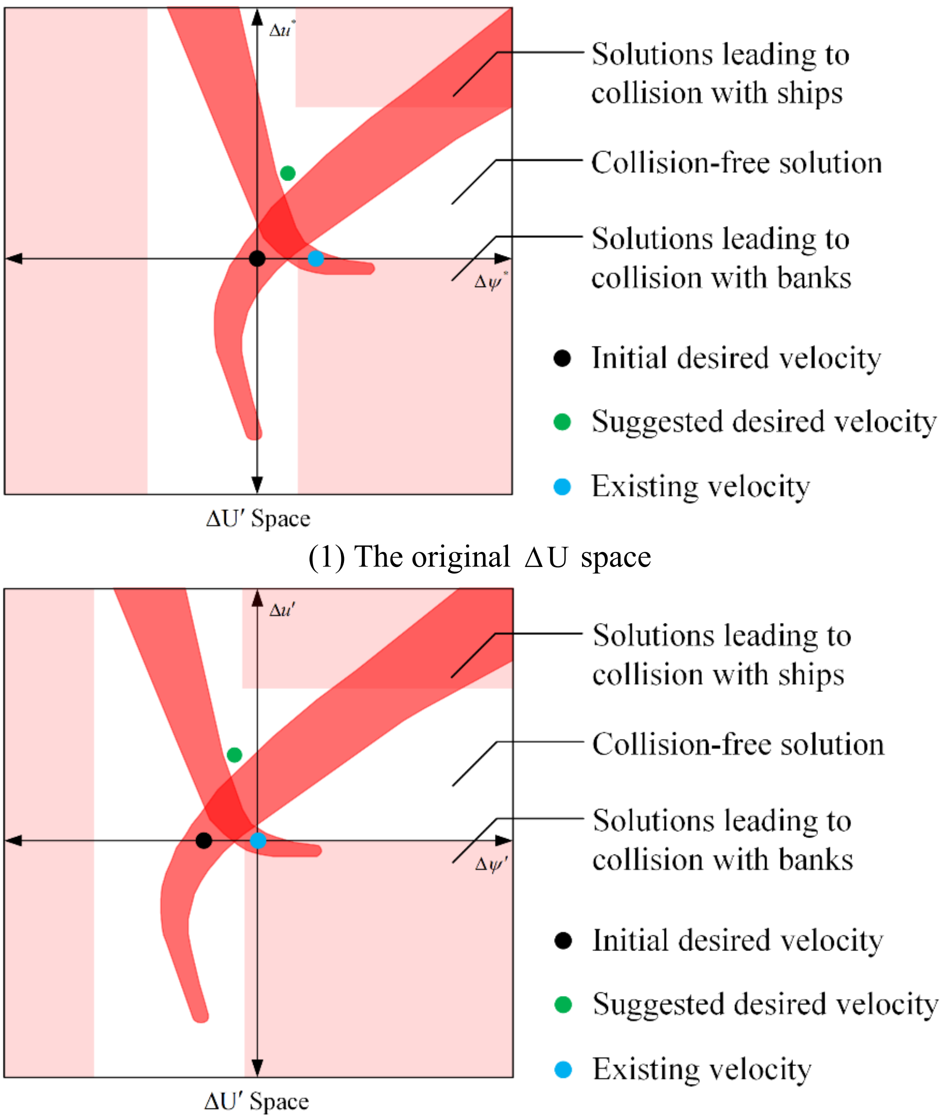
(1) The desired velocity $\mathbf{u}^*$ is reduced to two degrees of freedom.

The desired velocity $\mathbf{u}^*$ has three degrees of freedom, which contains three variables, namely, surge speed u^*, sway speed v^*, and heading ψ^*. Thus, the UO set and the collision-free solution are all containing three dimensions. For two main reasons, the degrees of freedom are reduced, in particular, the desired sway speed is always set to be 0, i.e., $v^* = 0$.

Firstly, the ship hull is not designed to sail by its sway direction. In principle, it is free for GVO algorithm to choose a solution in surge speed u^* or sway speed v^*. Thus, it is possible that the machine finds a sway speed solution and the ship moves in its sway side only to avoid a collision, but that is not common in practice. Secondly, a two-dimensional map is more popular on board, e.g., radar figures, nautical charts, etc. So, the human users can easily read the UO set like the nautical chart and radar figure.

By reducing the degrees of freedom, the ΔU space can be displayed in the two-dimensional Cartesian coordinate system (e.g., Figure 5.5 (1)). Moreover, the horizontal axis is assigned to represent the changes in headings and the vertical axis to represent the changes in speed. These settings are expected to be friendly for human users.

Remark 5.1: The desired sway speed v^* is always set to be 0, and the desired velocity $\mathbf{u}^*$ remains with two degrees of freedom: u^* and ψ^*.



(1) The original ΔU space

(2) The modified ΔU space

Figure 5.5 Illustrations of an interface of the HMI-CAS using GVO algorithm.

(2) The origin of ΔU space is moved to the existing velocity of the ship

The origin of this ΔU space is the desired velocity outputted from the last control loop, i.e., the initial desired velocity $\mathbf{u}^0$. The horizontal axis of ΔU space is the difference in the heading ($\Delta\psi^* = \psi^* - \psi^0$), and the vertical axis is the difference in surge speed ($\Delta u^* = u^* - u^0$).

Since the initial desired velocity is not identical to the existing velocity of the ship, some outputs might look strange for the users. For instance, three velocities are marked in Figure 5.5 (1), which are an initial desired velocity (in black), suggested desired velocity (in green), and existing velocity (in blue). The suggested velocity (by machine) advices the OS to accelerate and adjust the desired velocity to its starboard side. If the users choose this solution, they observe the ship still turns port side because the suggested velocity is still located at the port side of the existing velocity.

To prevent the outputs incompliant with the feelings of human users, the origin of ΔU space from the initial desired velocity should be moved to the real velocity of the ship. As a result, the negative half-plane always means the port-side turns regarding the real velocity. Additionally, any point in this space represents the difference between the alternative desired velocity and the real velocity at the present time, i.e., $\Delta\mathbf{u}^* = \mathbf{u}^* - V\mathbf{x}$. An illustration of the modified ΔU space is shown in Figure 5.5 (2).

Remark 5.2: The origin of the space that UO set is presented is moved from the initial desired velocity $\mathbf{u}^0$ to the real velocity at present, i.e., $V\mathbf{x}$.

By introducing Remark 5.1 and Remark 5.2, the ΔU space is modified to be more friendly for human users to read, to interact, and to control the ship.

When the HMI-CAS delivers a solution to human, e.g., the green dot in Figure 5.5 (2), the human operators can have two options to interact with the machine:

(1) If human operators agree with the solution offered by the system, they can authorize the system. Then, the PD controller in HMI-CAS will track this solution as the desired velocity, and the interface will be updated again;

(2) If human operators disagree with the solution by the system, e.g., rules violation, etc., they can find an arbitrary solution in ΔU space and input to the system, say $\Delta\mathbf{u}^{\mathrm{human}}$. Then, the machine would track a new desired velocity: $\mathbf{u}^* = \Delta\mathbf{u}^{\mathrm{human}} + V\mathbf{x}$.

5.4.2 Design of conflict detection

The proposed HMI-CAS also contains a conflict detection module to trigger the conflict resolution module. The conflict resolution module is triggered if and only if the desired velocity is located in any UO sets. For instance, if the existing velocity is in the UO set, but the initial desired velocity is outside of UO set, the system would not find a new solution since the initial desired velocity is collision-free for the ship. This setting prevents the system to find a new solution that differs from the initial desired velocity, which is expected to reduce the oscillation of the system.

Since human operators have superiority over the HMI-CAS, the operators can terminate the implementation of the solutions provided by the machine, i.e., Mode 3 in Table 3.4. However, the conflict resolution is still triggered, which only offers suggestions to users. At the meantime, the collision risk calculated by the proposed Time-varying Collision Risk (TCR) measure is shown to the users in the interface, which is addressed in Chapter 6. The TCR reminds the users of the "room-for-maneuver" for the OS.

5.4.3 Design of conflict resolution

The design of the conflict resolution module is shown in Figure 5.6. Three sub-modules are included: "Reference velocity" sub-module generates the reference velocity that guides the ship to its waypoints; "GVO algorithm" sub-module produces UO sets, which is presented in Section 5.3; "New Resolution" sub-module find a new collision-free solution using UO sets.

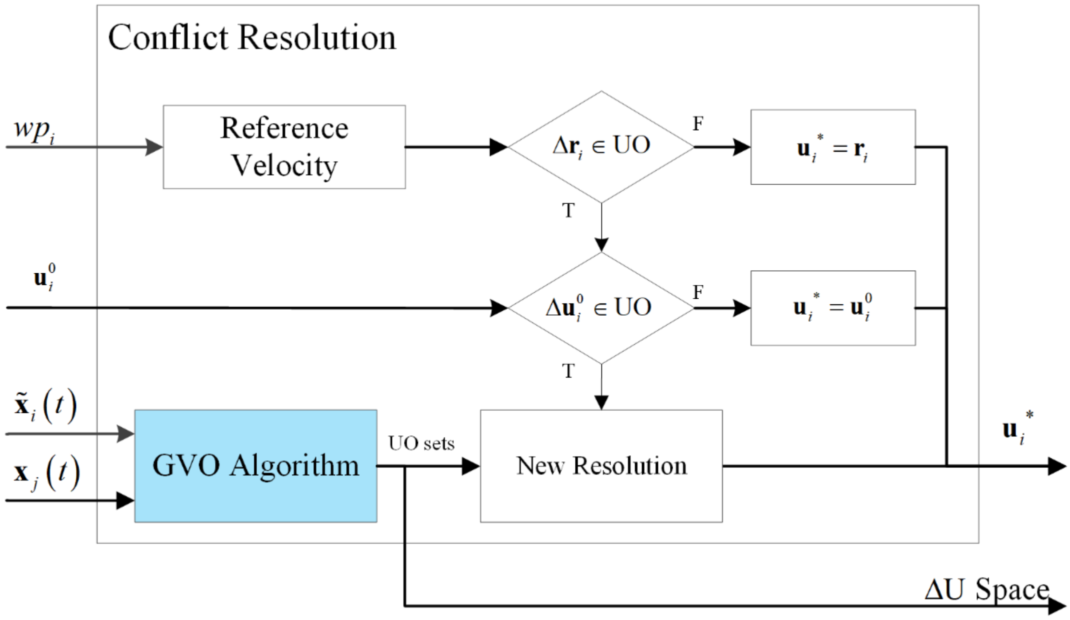


Figure 5.6 Representation of conflict resolution module using the GVO algorithm.

5.1.1.1 *Reference velocity*

"Reference Velocity" sub-module receives waypoints and produces a reference velocity. In this chapter, the reference velocity is defined as:

$$\mathbf{r}_i\left(t^0\right) = \left[u_{eco}, 0, \psi_{ref} + \psi\right]^{\mathrm{T}}, \tag{5.12}$$

where u_{eco} is a constant speed, which is the economical speed of the OS;

ψ_{ref} is the relative bearing of the waypoints.

5.1.1.2 *New Resolution*

Two modes of conflict resolution are developed in HMI-CAS: COLREGs-aware mode and optimal mode. One is considering navigational rules in finding a collision-free solution, which is named as COLREGs-aware mode; The other neglects the regulations and finds a collision-free solution via an optimization, which is named optimal mode.

(1) COLREGs-aware mode

COLREGs-aware mode enables to offer a rule-compliant solution in some simple scenarios, e.g., two-ship encounters. To explain the rules of conflict resolution, some vectors are defined in advance: the vector $\Delta\mathbf{r}$ is the reference velocity in ΔU space which is formulated as: $\Delta\mathbf{r} = \mathbf{r} - V\mathbf{x}$; the vector $\Delta\mathbf{u}^0$ is the initial desired velocity in ΔU space, i.e., $\Delta\mathbf{u}^0 = \mathbf{u}^0 - V\mathbf{x}$; the vector $\Delta\mathbf{u}$ is the alternative desired velocity in ΔU space, and the output desired velocity is calculated by $\mathbf{u}^* = \Delta\mathbf{u} + V\mathbf{x}$.

The following rules are employed to choose a new desired velocity, and they are presented in order of priority:

Rule 1: The OS is expected to choose reference velocity $\mathbf{r}$ if $\Delta\mathbf{r} \notin UO$;
Rule 2: The OS prefers to continue with its initial desired velocity $\mathbf{u}^0$ when $\Delta\mathbf{u}^0 \notin UO$;
Rule 3: A new $\mathbf{u}^*$ should keep its current speed, avoid port-side turn and satisfy $\Delta\mathbf{u} \notin UO$;
Rule 4: A new $\mathbf{u}^*$ is close to its current velocity and satisfies $\Delta\mathbf{u} \notin UO$.

Rule 1 means when the reference velocity is collision-free, the ship will choose the reference velocity. This rule is with the highest priority, which helps the ship to follow waypoints.

Rule 2 indicates that if the current desired velocity is collision-free and the reference velocity is not, the ship's controller will still implement the current desired velocity.

Rule 3 shows the principle of the OS to find a collision-free solution compliant with COLREGs. The expression "keep its current speed" follows Rule 8 in COLREGs, which encourages the ship to find a collision-free solution via changing the course. Besides, "avoid port-side turn" is introduced to comply with Rule 14, 15 and 17 in COLREGs (the relevant rules are listed in Appendix I), i.e., find a solution in right half-plane.

The nearest collision-free solution is usually located on the boundary of the UO set. Thus, HMI-CAS searches the intersection between the UO set and the positive side of the x-axis. The intersection is a collision-free candidate. If the intersection is out of the feasible region, HMI-CAS will search the nearest points on the boundary in the right half-plane. However, when the solution on the boundary is chosen, two ships might get infinitely close to each other. To avoid this situation, a small value ε is added to the candidate. An example is shown in Figure 5.7, Point A indicates the initial velocity, which is inside the UO set; Point B is the closest point to A on the UO set, and it is the suggested collision-free desired velocity; Point C is the final solution, which can be formulated as: $AC = (1+\varepsilon)AB$.

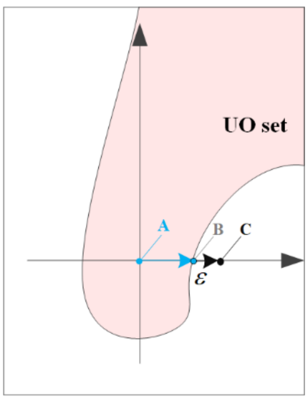


Figure 5.7 Illustration of finding an optimal solution using UO set with buffer ε.

Rule 4 is introduced when there is no COLREGs-compliant and collision-free solution within the feasible range. In this urgent case, a collision-free control is found which is closest to the current system's state. Details of finding such a collision-free solution are explained by Equation (5.13) to (5.15).

(2) Optimal mode

The proposed HMI-CAS also offers a mode which neglects the regulations, i.e., optimal mode. The task of finding rule-compliant actions is assigned to human operators who are experts in understanding rules. The HMI-CAS only offers an optimal solution to human operators.

The rules for finding a new desired velocity in HMI-CAS is changed as follows, and they are presented in order of priority.

Rule 1: The OS is expected to choose reference velocity $\mathbf{r}$ if $\Delta\mathbf{r} \notin \mathrm{UO}$;

Rule 2: The OS prefers to continue with its initial desired velocity $\mathbf{u}^0$ when $\Delta\mathbf{u}^0 \notin \mathrm{UO}$;

Rule 3: A new $\mathbf{u}^*$ is close to its current velocity and satisfies $\Delta\mathbf{u} \notin \mathrm{UO}$.

When the reference velocity and the initial desired velocity are both unsafe, finding a new solution is necessary. This problem can be modeled as a non-convex optimization problem since the solution space $\mathbb{R}^2 \setminus \mathrm{UO}_{ij}$ is non-convex.

The final control input for ship i not only needs to fall in $\mathbb{R}^2 \setminus \mathrm{UO}_{ij}$, but also satisfy the kinematic constraints, i.e., $u \in [u_{\min}, u_{\max}]$ and $\psi \in [-\psi_{\max}, \psi_{\max}]$, noted as U_i^{bound} . Thus, the feasible space for ship i to prevent collision with ship j is $U_{ij}^{fea} = \overline{\mathrm{UO}}_{ij} \cap U_i^{bound}$, see Figure 5.8 (1). Obviously, U_{ij}^{fea} is non-convex, and an approximation of feasible space is necessary.

i. Approximation of feasible space.

Firstly, a convex hull is employed to approximate the intersection of UO_{ij} and U_i^{bound}, noted as $\mathrm{CH}\left(\mathrm{UO}_{ij} \cap U_i^{bound}\right)$, see Figure 5.8 (2). Secondly, the closest point on the boundary of $\mathrm{CH}\left(\mathrm{UO}_{ij} \cap U_i^{bound}\right)$ to the initial desired velocity $\mathbf{u}^0$ is found, which is $\mathbf{w}$.

$$\mathbf{w} = \underset{\mathbf{u} \in \partial \mathrm{CH}\left(\mathrm{UO}_{ij} \cap U_i^{bound}\right)}{\arg\min} \left\| \mathbf{u} - \mathbf{u}^0 \right\|, \tag{5.13}$$

where ∂ refers to the boundary of a set. In [148], the authors reported when the target velocity is on the boundary of the VO set, two vehicles will approach each other infinitely close. Thus, in this system, a repulsive term $\hat{w}$ is adopted with $\mathbf{w} := \mathbf{w} + \mathbf{w} / \|\mathbf{w}\| \hat{w}$ and $\hat{w}=0.02$, as shown in Figure 5.8 (3). Lastly, U_{ij}^{fea} is approximated via a linear constraint that is formulated as:

$$\tilde{U}_{ij}^{fea} = \begin{cases} \left\{ \mathbf{u} \middle| \left(\mathbf{u} - \mathbf{w}\right) \cdot \left(\mathbf{w} - \mathbf{u}^0\right) \geq 0 \right\} & \text{if } \mathbf{u}^0 \in \mathrm{CH}\left(\mathrm{UO}_{ij} \cap U_i^{bound}\right) \\ \left\{ \mathbf{u} \middle| \left(\mathbf{u} - \mathbf{w}\right) \cdot \left(\mathbf{w} - \mathbf{u}^0\right) \leq 0 \right\} & \text{otherwise} \end{cases}. \tag{5.14}$$

In return, the feasible space becomes convex (see Figure 5.8 (3)).

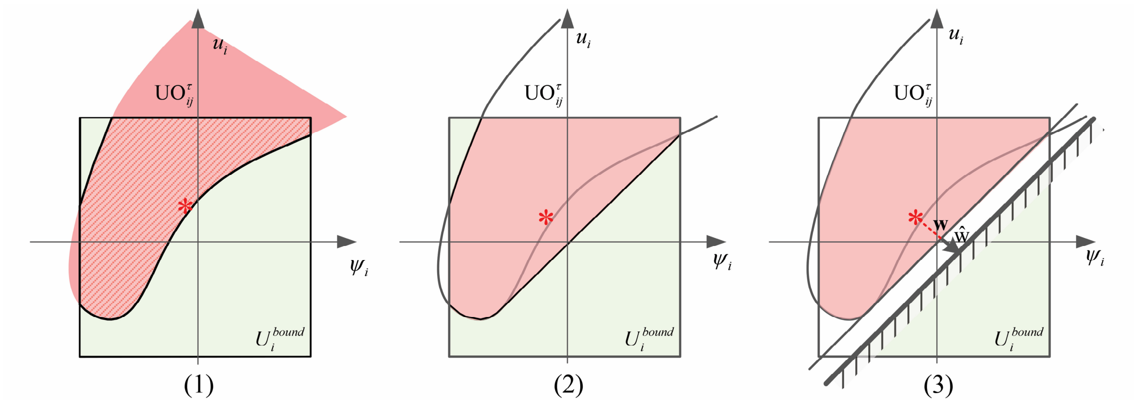


Figure 5.8 Illustration of approximation of feasible space U_{ij}^{fea}.

ii. Formulation of the optimization problem.

The collision problem can be formulated as a quadratic problem:

$$\text{Minimize} \quad J_{UO}(\mathbf{u}_i) = \left(\mathbf{u}_i - \mathbf{u}_i^0\right)^T \begin{bmatrix} a_u & 0 \\ 0 & a_\psi \end{bmatrix} \left(\mathbf{u}_i - \mathbf{u}_i^0\right) \tag{5.15}$$
$$\text{Subject to:} \quad \mathbf{u}_i \in \bigcap_{j \neq i} \tilde{U}_{ij}^{fea}, \ \mathbf{u}_i \in U_i^{bound}.$$

The matrix diagonal matrix is introduced to reflect the experience in ship navigation that course turning is more popular in collision avoidance at sea, i.e. $a_u >> a_\psi$.

5.5 Case studies

In this section, simulation experiments are carried out to demonstrate the proposed HMI-CAS using GVO algorithm in various encounter scenarios. The structures and settings of HMI-CAS following the contents addressed in Section 5.4.

5.5.1 Setup

In this section, a scale model of the ship, "CyberShip II" [193], is employed. Details of this model ship are attached in Appendix II. The ship is seen as a circle. Accordingly, a set of positions leading to collisions (*ConfP*) is shaped like a circle with radius R and is defined as:

$$ConfP(\mathrm{R}) = \left\{\mathbf{P} \middle| \|\mathbf{P} - \mathbf{O}\|_2 \leq \mathrm{R}\right\}, \tag{5.15}$$

where R is the sum of two ships' radius; $\mathbf{O}$ is the origin and $\mathbf{P}$ is a position vector.

Froude scaling law is used to link the scaled model and the real physical world. The scale factor is set to $\alpha = 1/70$. In the following sections, the units are all in scale system, otherwise indicate. The details of links between scaled model and the real world refers to Table II.2.

Feedback gains K_p and K_d in Equation (5.3) are set as: K_p = diag([200, 10, 10]) and K_d = diag([2, 2, 2]). The economical speed in Equation (5.12) is $u_{eco} = 0.5$ [m/s]. That means the ship prefers to sail 0.5 [m/s] in the scaled world, i.e., 8 [knot] in the real-size world. The small value ε used in Rule 3 and Rule 4 is set as 0.01.

Some kinematic constraints are considered to determine a feasible range. Specifically, the maximum turning is $\pm 90^\circ$, the minimal speed is 0 [m/s], and maximal speed is 1 [m/s].

A simulator, which is employed to simulate the encountering scenarios, is developed in Matlab 2016b platform. The frequency of the control sequence is set as 1 Hz. The prediction horizon is 80 [s]. Runge-Kutta integration uses 0.1 [s] time-step. The *ConfP* is approximated as a 15-sided polygon.

5.5.2 Scenario 5-I: VO algorithms versus GVO algorithm in heading scenarios

The performance of VO algorithm and GVO algorithm are compared in this section. Specifically, a series of heading scenarios is simplified in which the relative distances between OS and TS are different. These ships are “CyberShip II” with a length of 1.255 [m]. Thus, the radius of *ConfP* and safety distance are both 1.255 [m], i.e., $L_{i.} =$ 1.255 [m]. The OS is placed at origin heading to the North, and its waypoint is set as (0, 28) [m].

When applying the VO algorithm, the VO set is used to find a collision-free solution instead of UO set. Correspondingly, Rule 1~3 in Section 4.3.3.2 is employed.

The TSs in these scenarios have an identical motion feature but different initial positions. The sketch diagram is shown in Figure 5.9. The distance between the OS and TS increases from $5L_i$ to $16L_i$. In total, 12 scenarios are considered. The details are shown in Table 5.1.

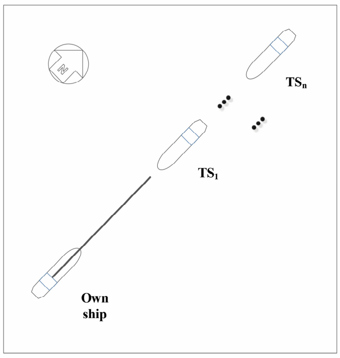


Figure 5.9 Scenario 5-I: illustration of a series of heading scenarios.

Table 5.1 Scenario 5-I: setting of heading scenarios

	Position [m]	Speed [m/s]	Course [30]
OS	[0,0]	0.5	000
TS$_k^*$	$\left[0,\left(k+4\right)L_i\right]$	0.5	180

Note *: k = 1, 2...12. L_i is the length of the OS.

The results are shown in Table 5.2. If the VO algorithm is applied in these scenarios, the safety constraint might not always be satisfied. Particularly, when the distance is smaller than $9L_i$, a collision cannot be avoided using the VO algorithm. On the contrary, the GVO algorithm performs properly in these scenarios.

Moreover, in the scenarios (Heading 9, 11, 12, 13, 15) in which the OS and the TS successfully avoid a collision, the VO algorithms require the OS to take more evasive actions than the GVO algorithm. The main reason is that the dynamics of the ship is neglected in the VO algorithm. When the VO algorithm offers one collision-free velocity ($\mathbf{u}^*$), the OS needs time to steer to the desired velocity, and this process needs time and space. After the OS reaches $\mathbf{u}^*$, the encountering situation has been changed. In the new situation, the previous $\mathbf{u}^*$ might become unsafe, and a new collision-free velocity is needed. However, the GVO algorithm considers the ship dynamics, i.e., the algorithm takes the action time and space into consideration. Following the solution provides by the GVO algorithm, the OS only needs one control to avoid a collision.

From these scenarios, it is seen that the GVO algorithm can perform better in close-range scenarios. Moreover, when applying the GVO algorithm, the succession of small actions can be avoided, which is advocated by regulations [194].

Table 5.2 Scenario 5-I: comparison of using VO/GVO algorithms to avoid collisions

	Target Ship	**Initial Dist. [L_i]**	**VO Algorithm**			**GVO Algorithm**		
			Minimal Dist. [m]	Successful Avoidance	No. of $\mathbf{u}^*$	Minimal Dist. [m]	Successful Avoidance	No. of $\mathbf{u}^*$
Heading 1	TS_1	5	0.992	×	5	1.730	√	1
Heading 2	TS_2	6	1.134	×	6	1.631	√	1
Heading 3	TS_3	7	1.154	×	8	1.434	√	1
Heading 4	TS_4	8	1.169	×	7	1.408	√	1
Heading 5	TS_5	9	1.276	√	6	1.429	√	1
Heading 6	TS_6	10	1.225	×	6	1.339	√	1
Heading 7	TS_7	11	1.265	√	4	1.327	√	1
Heading 8	TS_8	12	1.357	√	4	1.355	√	1
Heading 9	TS_9	13	1.325	√	4	1.349	√	1
Heading 10	TS_{10}	14	1.2547	×	2	1.294	√	1
Heading 11	TS_{11}	15	1.298	√	2	1.322	√	1
Heading 12	TS_{12}	16	1.395	√	1	1.389	√	1

5.5.3 Scenario 5-II: HMI-CAS using UO sets in a crossing scenario

Additionally, a crossing scenario is used to show how the proposed HMI-CAS using UO set to find a solution that complies with navigational regulations. The setting of this scenario is listed in Table 5.3, and the encounter scenario is shown in the left panel of Figure 5.10.

Figure 5.10 (1)-(3) show the interface of HMI-CAS at different time slices. In these figures, x-axis and y-axis represent the changes in heading and speed, respectively. The origin refers to no changes in heading or speed, i.e., $\Delta\mathbf{u} = (0,0)$, which implies the OS keeps its velocity at present. The “*” in red is the reference velocity shown in $\Delta \mathrm{U}$ space, which is calculated by the “Global Planner” module. The “o” in blue is the suggested control by the proposed system, i.e., new desired velocity $\mathbf{u}^*$.

Figure 5.10 (1) shows the $\Delta \mathrm{U}$ space at t = 1 [s]. At this moment, the OS is sailing at its reference velocity, and this velocity is its initial desired velocity. However, HMI-CAS detects that Δr (“*” in red) is inside the UO set. That implies that Rule 1-2 are not held, and there is a collision danger in the near future. Therefore, Rule 3 is activated, and the system searches a new solution in the positive direction of the x-axis. Consequently, a solution $\Delta u = (0, 0.3688)$ is found (“o” in blue): the OS is asked to keep its speed and turn 0.3688 [rad] (or 21 [$^\circ$]) to starboard. The desired velocity (u^*) then is the sum of velocity at present Vx and Δu .

Table 5.3 Scenario 5-II : settings of crossing scenario

	Position [m]	Speed [m/s]	Course [°]
OS	[0,0]	0.5	000
TS	$\left[5L_i, 5\left(1+\sqrt{2}\right)L_i\right]$	0.5	-145

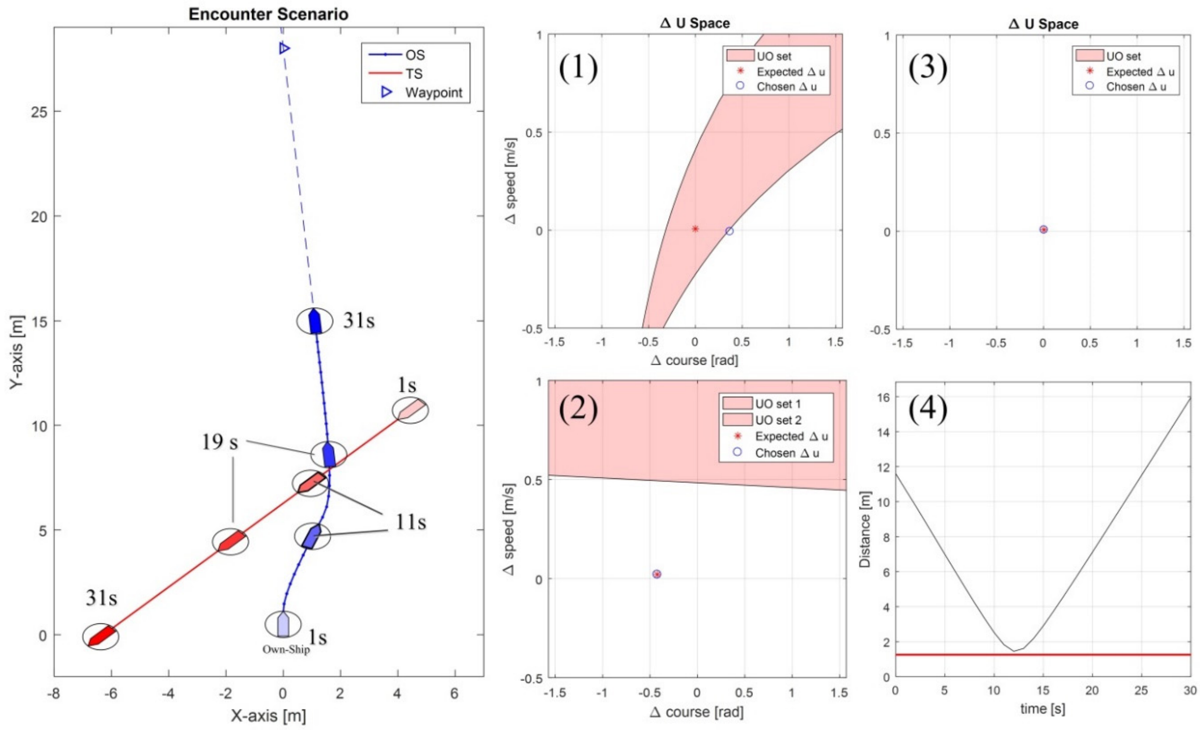


Figure 5.10 Scenario 5-II: encounter scenario, interface of HMI-CAS, and relative distance.
(Note: (1) UO set at 1 [s]; (2) UO set at 11 [s]; (3) UO set at 19 [s]; (4) Relative Distance)

Figure 5.10 (2) shows the ΔU space at t = 11 [s]. Before this moment, the OS keeps the desired velocity obtained at 1 [s]. However, then, HMI-CAS detects that Δr is outside of the UO set, which activates Rule 1 in Section 5.4.3.1. Though the TS is just located on the straight-line between the OS and its waypoint, the ship chooses to head to its waypoint. This solution requests the OS to turn port-side and keep a speed close to the current speed.

In the following time, if there are no new conflicts with other ships, the OS will keep its reference velocity r and heading to the waypoint. In Figure 5.10 (3), the ΔU space is presented at 19 [s]. Since there is no UO set and the desired velocity (also reference velocity) is collision-free, the OS is suggested to keep its desired velocity. The relative distance at each moment in Figure 5.10 (4) shows that the HMI-CAS works properly and prevents a collision in this case.

Since human operators have a superiority over the HMI-CAS, they can intervene in the system when the solution offered by the system is not satisfied. For instance, at t = 1 [s], the HMI-CAS offers a solution, and this solution is presented in the interface. If the human operators do not agree with this solution, they can choose another solution outside of the UO set and feed the new solution back to the HMI-CAS. At t = 11 [s], the system suggests returning to its initial route, while the operators can postpone this decision by choosing the origin, i.e., to continue the existing velocity.

5.5.4 Scenario 5-III: cooperative collision avoidance in a multiple-ship case

A multiple-ship scenario is designed to show the potential of the HMI-CAS in a more complicated case. Three “CyberShip II” ships are employed, and they all have HMI-CAS on board to determine their evasive action. The encounter situation is displayed in Figure 5.11. The safety distance is set as 1.255 [m]. Other parameters are shown in Table 5.4.

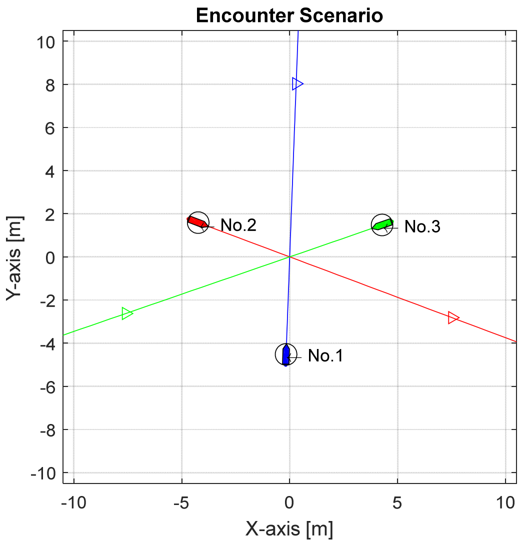


Figure 5.11 Scenario 5-III: encounter scenario at 0 [s].

Table 5.4 Scenario 5-III: settings of ships in multiple-ship scenario

	Position [m]	Speed [m/s]	Course [30]	Destination [m]
Ship 1	$[-0.19,-5.02]$	0.5	002	$[0.30,8.03]$
Ship 2	$[-4.70,1.76]$	0.5	200	$[7.52,-2.82]$
Ship 3	$[4.74,1.64]$	0.5	251	$[-7.59,-2.62]$

It is presumed that these ships are all installing HMI-CAS on board and they are capable of knowing the trajectories of other ships. Thus, in the simulation, ships broadcast their trajectories sequentially: when one ship chooses an evasive action, the updated trajectory will be broadcasted to the others; then, the next ship finds a collision-free solution according to the updated trajectory and broadcasts it; this process continues until all the ships find their solutions. Additionally, if one ship could not find an available solution, it will keep its current state, and other ships will try to find evasive actions. These settings result in a cooperative collision avoidance scenario. For more discussion in cooperative collision avoidance, readers can find more materials in [195].

Figure 5.12 shows the system states and the ΔU spaces of the ships at time 1 [s]. Figure 5.12 (1) shows the encounter scenario at 1 [s], and all the ships have found their collision-free solutions. Figure 5.12 (2)-(4) show how Ship 1, Ship 2, and Ship 3 choose a collision-free solution based on the UO sets, respectively. Ship 1 is the first ship to search its evasive action. As shown in panel (2), a starboard turn solution was found, and the ship updated its trajectory and broadcasted the changes to the others, see the blue line in panel (1). Then, Ship 2 determines its evasive action according to the initial trajectory of Ship 3 and the updated trajectory of Ship 1. In the end, Ship 3 search its evasive action based on the updated information from other ships.

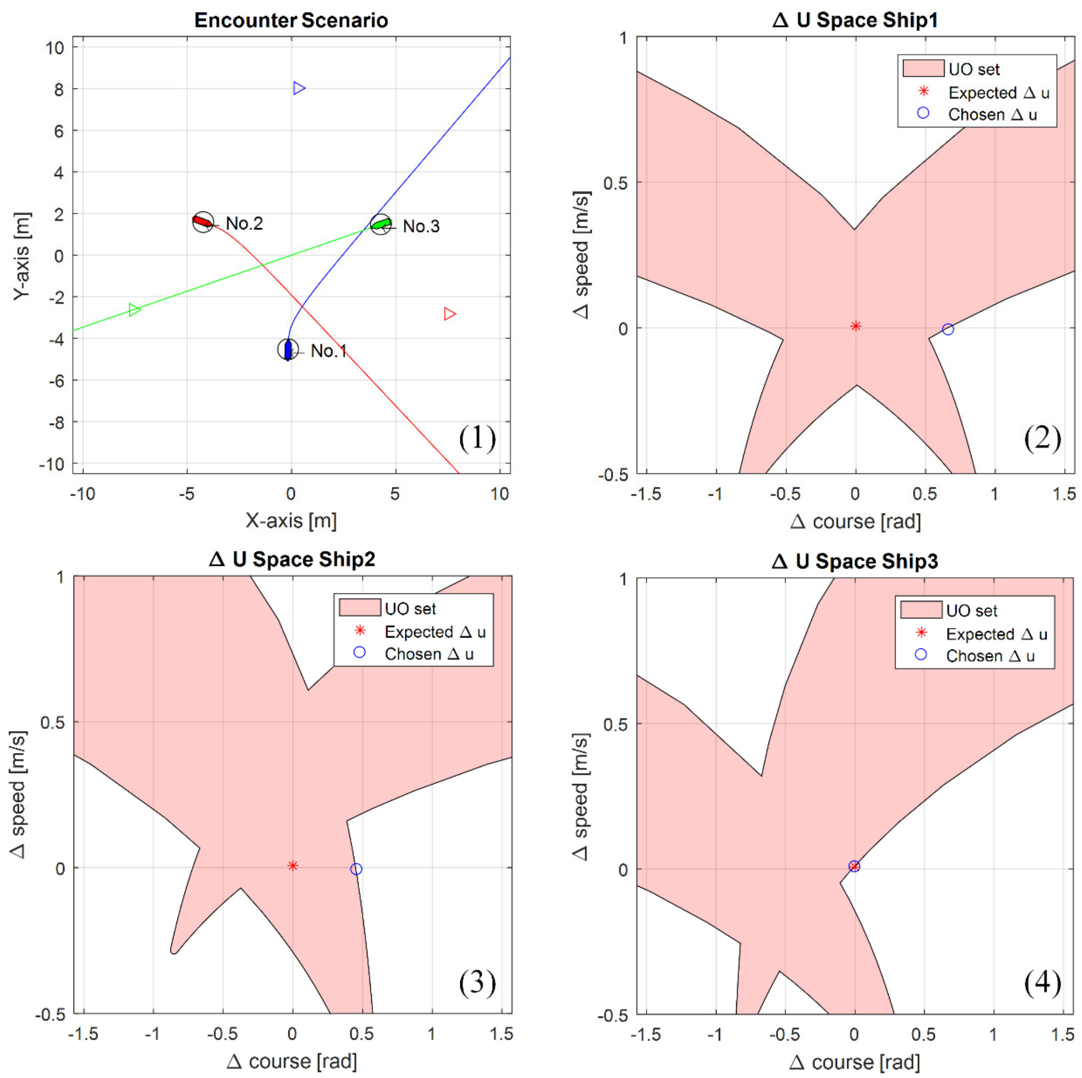


Figure 5.12 Scenario 5-III: encounter scenario and UO sets from each ship's view at 1 [s].

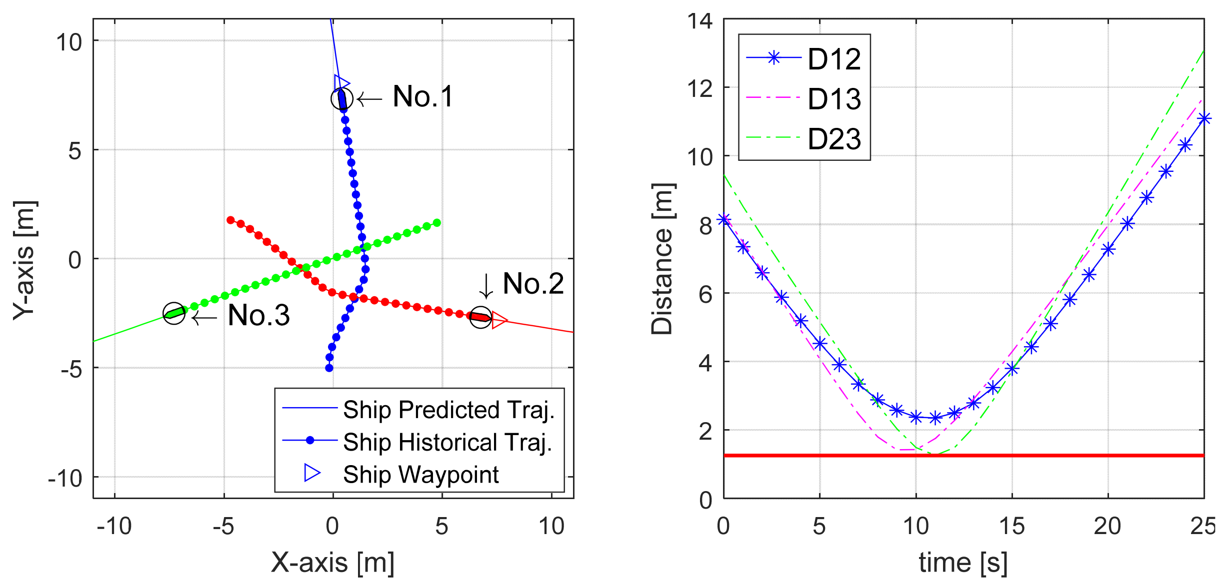


Figure 5.13 Scenario 5-III: encounter scenario at time 25 [s] and relative distance.

*Note: 'D12' means the distance between Ship 1 and Ship2, 'D13' is the distance between Ship 1 and 3, and 'D23' is the distance between Ship 2 and 3.

5.5.5 Discussion of case studies

Heading scenarios compare the performance of the VO and GVO algorithm. When two ships have sufficient initial relative distance, both the VO algorithm and the GVO algorithm can find collision-free solutions. However, when the initial distance decreases, the VO algorithm fails to find a collision-free solution. Moreover, the VO algorithm usually offers a series of small maneuvers, which is not encouraged by COLREGs (Rule 8(b)). From these perspectives, the GVO algorithm performs better in conflict resolution than the VO algorithm under the same condition.

The crossing scenario shows the capability of the proposed system in offering regulation-compliant solutions. Here, it is demonstrated how the HMI-CAS searches a collision-free solution automatically. Nevertheless, this method also can be used for supporting the human operators. Specifically, the operators can either choose the solution suggested by HMI-CAS or choose any collision-free solutions outside the UO set themselves.

The multiple-ship scenario shows the potential of the HMI-CAS in compliant encounter situations. Using the proposed system, the ships can cooperate and search for the collision-free solution automatically. In the simulation, the order of ships finding solutions are predetermined: Ship 1 finds its solution firstly; then Ship 2 starts searching; and, Ship 3 is the last one. This arrangement, in fact, gives Ship 3 a higher priority than Ship 2 and Ship 1. As shown in Figure 5.13, the routes of Ship 1 and Ship 2 are longer than the route of Ship 3. Ship 3 might even not need to take evasive action if the actions Ship 1 and Ship 2 took have solved the collision conflicts. As cooperation is not the focus of this chapter, this mode is introduced with priority as an example of cooperation. For the reader interested in this theme, more cooperation modes are found in [195].

5.6 Discussion

In this section, a comparison is carried out between the GVO algorithm and the VO algorithm in the maritime environment. The differences between the modified GVO algorithm in this work and other related works is also illustrated. Besides, the limitations of the GVO algorithm and regulation-compliant issue of HMI-CAS are also discussed.

5.6.1 Comparison between the GVO and the VO algorithms

This chapter shows that both the VO and the GVO algorithm can support collision prevention for the ship in many situations. However, the GVO algorithm has better performance in providing a more reliable solution, and fewer control actions are needed for collision avoidance than the VO algorithm.

From a theoretical perspective, the GVO algorithm is more suitable for ships in a dynamic environment than the VO algorithm. The derivations of the VO and GVO algorithms both start from the same collision condition, i.e., Equation (4.2): $\mathbf{P}_i(t) \in \mathbf{P}_j(t) \oplus ConfP$. The disparity starts from the calculation of the OS's trajectories $\mathbf{P}_i(t)$. The VO algorithm assumes that the ship can change its velocity immediately, while in the GVO algorithm, the calculation of $\mathbf{P}_i(t)$ depends on the ship's dynamics. Therefore, $\mathbf{P}_i(t)$ calculated by the second method is closer to the real ship's trajectory. Thus, the GVO algorithm is more reliable than the VO algorithm in conflict detection and resolution.

Since the GVO algorithm is a generalized version of the VO algorithm, some developments of the VO algorithms can be directly applied to the GVO algorithm. For example, in Section 5.3, the GVO algorithm is capable of dealing with non-linear trajectories of ships, which was learned from the non-linear VO algorithm. It is also possible to integrate other VO algorithms in GVO algorithm such as PVO algorithm [184], EBVO algorithm [185], RVO algorithm [181], etc., which can expand its applications at sea.

5.6.2 Comparison of GVO algorithm with related works

The GVO algorithm in this chapter is based on the original GVO in [182]. The original GVO algorithm is proposed for unmanned vehicles, such as wheel-based robots, drones, etc., in a lab environment. Due to the differences in operating environments and application purposes, some modifications have been made to accommodate to the maritime environment.

First of all, the existing GVO method assumes that each participant has complete information about itself and others in advance, e.g., the dynamics of ships. However, this is not realistic in the maritime environment. Therefore, in the current method, the dynamics of the ship and the controls are not necessary to be completely known or observable to each other. Instead, it is assumed that the ships can exchange their predicted trajectories, which is more realistic in the maritime practice.

Secondly, although the existing GVO method can find collision-free options, they only choose one optimal solution for unmanned vehicles. Readers also can see the reference [196, 197] for more cases. In this chapter, the GVO is applied as a decision-supporting tool for human operators. In HMI-CAS, the visualization of UO set and ΔU spaces make the decision process of GVO clear for human operators. This characteristic makes that the HMI-CAS not only can be used in the autonomous ship (MASS IV) but can also support decision making of human operators in manned ships (MASS I) and remoted ASVs (MASS II&III). The detail of MASS is addressed in Section 3.1.

Thirdly, the original GVO methods find an optimal solution via minimizing the changes in control, irrespective of the rule of the road. However, in the maritime environment, the evasive actions should be taken in compliance with navigation regulations. Thus, this chapter offers a conflict resolution mode that finds a collision-free solution incorporating COLREGs instead of only minimizing changes in controls.

5.6.3 Limitations of the proposed HMI-CAS

The HMI-CAS is based on the linearization of the ship's dynamics via Taylor expansion, see Section 5.3.2. As a result, there are errors between the predicted trajectory calculated by Equation (5.8) and the actual nonlinear trajectory of the ship. When the difference between alternative control and initial control is too big, the solution which the algorithm finds is not guaranteed to be collision-free.

To overcome this problem, four methods have been considered in this chapter, i.e., successive linearization, enlarged buffer, increasing sampling rate, and reducing fluctuation. Firstly, the successive linearization technique is used to reduce errors. Usually, the linearization of a system is around its initial state in the first stage. Hence, errors are accumulated. In this chapter, the system is always linearized around the estimated trajectory at each time step. Thus, the errors are less than the errors in the usual approach by linearizing only at the first stage. Secondly, the UO set is enlarged by a positive small factor ε and $\hat{w}$ when the HMI-CAS finds a collision-free solution, which can avoid the collision caused by linearization errors under most conditions. Thirdly, the sampling rate in the HMI-CAS is increased to ensure that a collision-free solution can be found in time. Lastly, Rule 4 in COLREGs-aware mode and Rule 3 in the optimal mode of HMI-CAS are introduced to prevent the fluctuations and dramatic changes of control inputs, e.g., from hard-port turn to starboard turn. Although these methods can work in most cases, a rigorous proof is needed in future research.

5.6.4 Compliance with regulations

Rules of the road in COLREGs are considered in the HMI-CAS as Rule 3 and Rule 4 in Section 5.4.3.1. The experiment results show that these rules can help the system to find a COLREGs-compliant solution. Moreover, though it was not the intension to let the system minimize the number of control actions, the system automatically found one solution, which avoids a succession of small changes in course and/or speed.

However, it needs to be remembered that these encounter situations are relatively simple. In the proposed HMI-CAS, a simple rule was embedded and asked the ship to prevent to take port-side turns. When the OS is the "give-way" ship, this behavior is rule-compliant, but the same behavior is not advocated by rules when the OS is the "stand-on" ship. In this dissertation, the question of finding a rule-compliant solution in various situations is left to human operators. The machine offers one collision-free solution and the relevant solution space to human operators, and human operators monitor and support the system to be rule-compliant.

Until today, developing a rule-compliant CAS is still an open question. To be rule-compliant, the system at least needs to contain three key functions: (1) identification of encounter scenarios; (2) selection of proper rules; (3) translation of ambiguous rules to specific actions. Combining HMI-CAS with artificial intelligence methods could offer one solution. For instance, HMI-CAS projects a dynamic encounter scenario as a static map into the solution space, and it can store the decision made by human operators that is a point in the map. With these data as inputs, a neural network can be trained with multiple layers to learn the choice of human operators. In the end, the system can make a choice close to human operators who follow the rules.

5.7 Conclusions

In this chapter, a generalized velocity obstacle (GVO) and its applications in the proposed HMI-CAS are introduced. Moreover, the relevant modifications of the HMI-CAS are explained in detail, which aims at facilitating human to use the system in collision avoidance.

Case studies of two-ship encounter scenarios and multiple-ship encounter scenarios show the performance of the GVO algorithm and the HMI-CAS. The two-ship encounter scenarios show that the GVO algorithm is more reliable than the Velocity Obstacle (VO) algorithm: it is capable of finding rule-compliant evasive actions for the ship, and fewer actions are needed to avoid collisions. In the multiple-ship encounter case, HMI-CAS is capable of supporting multiple ship collision avoidance problems.

After all, this chapter answers the research question in Section 1.3: How can ship dynamics and navigational regulations be incorporated in HMI-CAS for conflict resolution? The answers are drawn as follows:

- A modified generalized velocity obstacle (GVO) replaces VO algorithm in HMI-CAS, which can consider ship dynamics in conflict resolution. In particular, the PD controller and GVO algorithm are assigned as a low-level controller and a high-level controller, respectively. Then, the input to the whole system is velocity, and the changes on velocity leading to collisions are collected in UO sets;

- To facilitate human users, the interface of the new HMI-CAS is redesigned. Firstly, the solution space is reduced from three dimensions to two dimensions, which includes surge speed and heading; Secondly, the origin of solution space is moved to the existing velocity of the ship, so that the solutions are more intuitive for human users.

- Two modes of conflict resolution have been proposed in HMI-CAS, in which one mode incorporates some simple rules from COLREGs to find a collision-free solution, and the other employs optimization to find a collision-free solution. If no rule-aware solution is found, the optimization is triggered.

Similar to the HMI-CAS using VO algorithms, the updated HMI-CAS also supports collision avoidance in the manned and unmanned ships and supports human operators to intervene in the automated collision avoidance, e.g., modify the solution found by HMI-CAS, validate the manned solutions, postpone the execution of avoidance, etc. Then, one question comes up, which is how can the collision dangers be detect before it becomes too late. This question is answered in the next chapter.

Chapter 6 Time-varying Collision Risk (TCR) Measures for Conflict Detection in HMI-CAS

In previous chapters, the proposed Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS) enables to detect collision dangers, to find a collision-free solution, and to cooperate with human operators. However, if human operators decide to postpone evasive actions, how to prevent unavoidable situations is the main question, which is addressed in this chapter. In this chapter, a new collision risk measure is proposed. The collision risk is measured as the percentage of maneuvers in which the ship cannot avoid collisions, named as Time-varying Collision Risk (TCR), which is used to remind the human operators to take actions in time.

This chapter is organized as follows: Section 6.1 addressed the motivation for the new collision risk measure. Section 6.2 introduced the proposed TCR measure, followed by Section 6.3 that demonstrates the performance of TCR measure in various scenarios. Discussion and conclusions are made in Section 6.4 and Section 6.5, respectively.

Acknowledgment The main content of this chapter is based on two papers:

Huang, Y. & van Gelder, P.H.A.J.M. 2019. Time-Varying Risk Measurement for Ship Collision Prevention. Risk Analysis (Online).

Y. Huang, P.H.A.J.M. van Gelder. A measure of collision risk for triggering evasive actions. (Submitted).

6.1 Introduction

Various systems have been proposed to support conflict detection, which can be categorized into two groups: collision alert system and collision avoidance system. The proposed HMI-CAS is one of the collision avoidance system. Both types of systems can set off an alarm when the ship is in danger, while the difference is the meanings of those alarms. One is for attracting attention from the human operators (collision alert system), while the other is for triggering evasive actions (collision avoidance system). A common ground of these systems is taking measurements of collision risk. The alarm is triggered when the risk is exceeding a certain threshold.

Risk usually associates with the level of safety/danger of systems. In quantitative risk assessments, the risk is usually defined as a combination of probability and consequence of an event, which is also interpreted as expected losses. However, the collision risk for preventing a collision on board usually only focuses on the likelihood of a collision and the consequence of the collision is seen as unacceptably high, i.e., infinitely large. In return, the measure of collision risk focuses on estimating the probability of a collision event.

Two popular ways of assessing collision risk are concluded in Chapter 2. One group of studies relies on experts' knowledge. The other group of studies usually calculates the probability of collision based on a simplified scenario. Some representative methods are collected and presented in Section 2.4. These methods suffer from one or two following limitations.

(1) Independent of conflict resolutions

Many risk measures are independent of the conflict resolution. Then, the high risk does not mean the collision is unavoidable or not. In return, these methods might result in one type of failure that no alarm is given until the collision is unavoidable. The representatives of those methods are CRI methods, Warning Ring methods, CPA methods, Probability Methods, Dangers Region Methods, etc. Details of these methods are presented in Section 2.4.

It is a common consensus that good maneuverability contributes to ship safety [198], which is also a motivation of engineers to improve the maneuverability of the ship. Better maneuverability means the ship has more maneuver options than one with poor maneuverability. More options indicate the conflict is easier to solve, and the ship has a higher chance to avoid a collision and, therefore, a lower risk of collision. Unfortunately, few risk measures can distinguish the risk change due to the difference in the conflict resolution.

Furthermore, the "bow-tie" theory from safety management can also be used to understand the necessity of incorporating conflict resolution in collision risk assessment. In the "bow-tie" of collision process, the "threats" are the target-ships, "top event" is a collision accident, the "consequence" is ship damage, and the "barrier" is the maneuver from the Own-Ship (OS), i.e., conflict resolutions, see Figure 6.1. The traditional risk measures only appraisal the possibility of the "threat" reaches the "top-event" and ignores the function of "barrier", see Figure 6.1(1). However, the OS enables to prevent the accident by using "barriers", i.e., maneuvers or conflict resolutions, see Figure 6 (2). Then, the assessment of the probability of "top event" needs to be associated with "barrier", i.e., conflict resolutions.

Figure 6.1 Illustration of bow-tie model of collision event (1) with & (2)without barriers.

(2) Dependent on pairwise encounters

Most of the risk measures are based on the encounter scenario that involves a pair of ships and ignores the impacts of other ships in risk assessments. In return, the changes in risk when the OS encounters with one more ship cannot be reflected. That means these risk measures cannot tell users the entire risk of collision when the ship encounters with multiple ships.

From the technical perspective, there are no agreements on combining various risk value in each pairwise encounter into one number that represents the risk of the entire traffic. Some operations would be alternatives, such as "average", "sum", and "maximum", but they more or less have some limitations. The "average" will underestimate the risk of the most dangerous Target-Ship (TS); the "maximum" ignores the influence from non-conflicting TSs; the "sum" offers limited information about the collision event in each pair.

Furthermore, when the traffic is decoupled in several pairs of ships, some information about traffic is lost and some biases are introduced in collision risk assessment. The biases of risk are caused by two aspects: (1) the risk caused by a non-conflicting TS is ignored. Although a non-conflicting TS does not directly have a conflict with the OS, it might block some operations of the OS which might result in an inevitable collision between the OS and another TS; (2) the risk caused by traffic characteristics is ignored. For example, well-organized traffic seems to be safer than the traffic in chaos (read Section 6.3.2 for more details).

In this chapter, a new risk measure is proposed with the help of VO algorithms, which considers both “threats” and “barriers”, i.e., the danger level of the approaching ships and the OS’s solutions for avoiding dangers. Collision risk is defined as the probability of collisions in the future, given the uncertainty in the OS’s maneuvers. The risk is formulated as the percentage of the achievable maneuvers that lead to collisions. This risk measure reflects the OS’s *room for maneuver* to avoid collisions with ships. A higher collision risk indicates fewer feasible maneuvers to avoid collisions, and more confidence that the collision will happen. A lower risk means more collision-free maneuvers, which implies that there are more chances to avoid the collision. Moreover, this method is also suitable for assessing collision risk in multiple encounters.

6.2 Time-varying collision risk measure

6.2.1 Definitions

There is no agreement on risk definition [199], but the risk is usually tied to probability and consequences. Researchers developing collision avoidance/alert systems are more interested in the probability of an unwanted event for preventing accidents [71]. In this work, therefore, the definition of risk based on probability concepts [200] is accepted.

Definition (6.1): Risk is the probability of an unwanted event.

The unwanted event refers to a ship collision event in this article. In each collision event, at least two ships are involved, i.e., an OS and a TS. Since the states of ships are time-varying, the probability of collision is time-varying, which is named time-varying collision risk (TCR).

Definition (6.2): TCR is the time-dependent probability of the event that OS cannot avoid a collision with TSs.

6.2.2 Idea of TCR measure

In TCR measure, the OS is modeled as a dynamic system. The position of the OS, say x, is updated by its velocity $\vec{v}$, i.e., $\dot{x} = f(x, \vec{v})$. A collision is the overlap of the position of the OS and a TS. Let K be a set of the OS’s states that triggers collisions, i.e., the un-safe set. Then, a collision is described as an event that the trajectory of the system (the solid curve in Figure 6.2) crosses an un-safe set K (the red region in Figure 6.2), i.e., $\exists t \geq t_0, x(t) \in \chi$.

With different maneuvers, the trajectory of the system might cross or avoid the un-safe set, e.g., the dashed lines in Figure 6.2. Hence, the TCR at time t can be formulated as a probability of maneuvers leading the trajectory to cross the un-safe sets, i.e.,

$$\mathrm{TCR}(t) = \sum_{i=1}^{n} p\left(\text{collision} \middle| \vec{v}_i, x_i(t)\right) \cdot p\left(\vec{v}_i, x_i(t)\right), \tag{6.1}$$

where $p(\vec{v}_i, x_i(t))$ is the probability of choosing $\vec{v}_i$ given state of system (x_i) at time t;

$p\left(\text{collision} \middle| \vec{v}_i, x_i(t)\right)$ is the probability of collision given $\vec{v}_i$ is chosen;

n is the total number of reachable velocities.

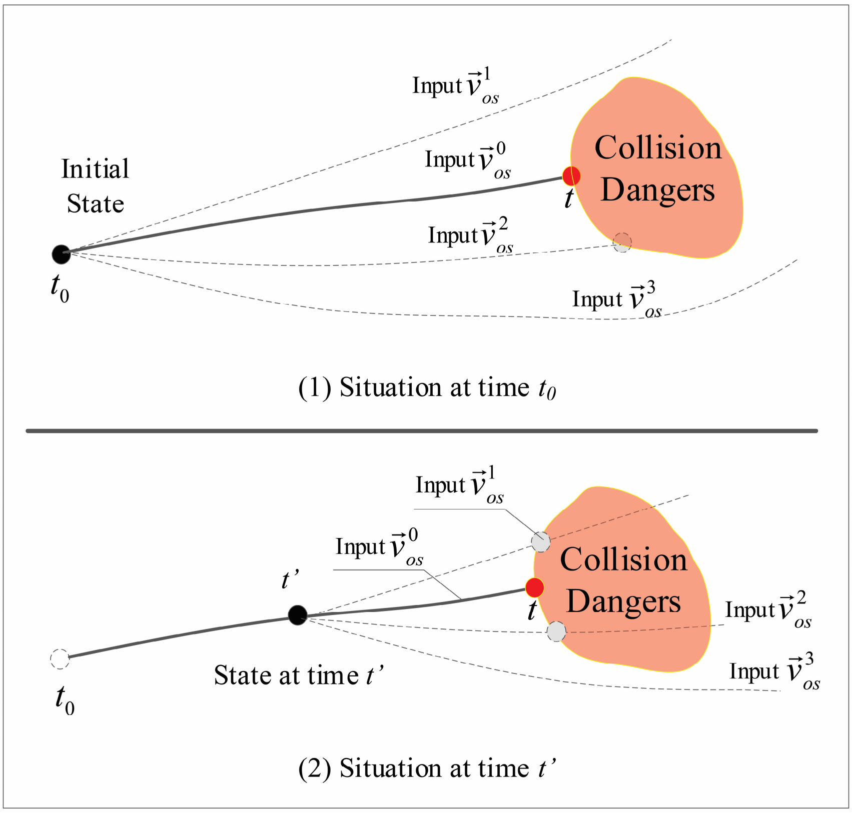


Figure 6.2 Illustration of a collision process in OS's state space.

Figure 6.2 illustrates the idea of TCR measure. A collision will happen at time t if no actions are taken. Any maneuvers before time t will change the trajectories of the OS. Different maneuvers could lead to different results, e.g., pass safely, near miss, and serious damage. Assume that the un-safe set (the red region in Figure 6.2) is known and stationary; the ship in Figure 6.2 has 4 feasible maneuvers at each time, and each maneuver has an equal chance. In Figure 6.2 (1), at time t_0, the system has two available solutions which can help the OS to avoid the collision. Thus, the TCR at time t_0 is 0.5. However, as time passes, the number of collision-free solutions decreases. In Figure 6.2 (2) at time t', only one maneuver is available for collision avoidance, say $\vec{v}_{OS}^{\,3}$. Therefore, the TCR at time t' increases to 0.75.

6.2.3 Implementation of the TCR measure

For the sake of convenient calculation, in this chapter, two assumptions are made to simplify the TCR measure in Equation (6.1):

Assumption (6.1): The probability of choosing maneuvers yield to a uniform distribution.

Assumption (6.2): The trajectories of the TSs are known.

From Assumption (1), the chance of choosing each solution in solution-space is set as equal. In return, the bias from users' preference is exclusive. Moreover, another distribution can also be assigned to replace the uniform distribution when the user determines one solution.

From Assumption (2), the motion of the TS is assumed to be known, which makes the un-safe sets certain and invariant. Subsequently, given a $\vec{v}_i$, there is only one result: the trajectory of the OS crosses the un-safe set, then $p\left(\text{collision} \mid \vec{v}_i, x_i\right) = 1$, or the trajectory avoids the un-safe set, which $p\left(\text{collision} \mid \vec{v}_i, x_i\right) = 0$.

Providing these settings, the probability of collision for the OS can be simplified as the proportion of maneuvers leading to collisions to its all feasible maneuvers before a collision:

$$\mathrm{TCR}(t) = \sum_{i=1}^{n} p\left(\text{collision} \middle| \vec{v}_i, x_i(t)\right) \cdot p\left(\vec{v}_i, x_i(t)\right) = \frac{n_{\text{collision}}(t)}{n(t)}, \tag{6.2}$$

where $n_{\text{collision}}(t)$ is the number of velocities leading to collisions at time t;

$n(t)$ is the number of reachable velocities before collisions at time t.

There are two keys to calculate the TCR in Equation (2): 1) identify all the velocities leading to collisions; 2) find all the reachable velocities before the collision. Thus, two sets are defined:

Definition (6.3): Velocity Obstacle set (VO set) is a set of velocities of the OS which could lead to collisions between the OS and the TSs.

Definition (6.4): Reachable Velocity set (RV set) is a set of velocities that OS can reach before the collisions.

In Figure 6.3, VO set and RV set are shown, where the VO set is depicted as a red cone, and the RV is represented as a green zone. The RV set is divided into several sub-areas by VO set, namely S_r, S_c, and S_l. The sub-area S_c is the intersection of VO set and RV set. It is a set of velocities that the OS can reach, but will lead to a collision:

$$S_c = \left\{ \vec{v}_i \mid \left(\vec{v}_i \in \mathrm{RV}\right) \cap \left(\vec{v}_i \in \mathrm{VO}\right) \right\}. \tag{6.3}$$

The remaining sub-areas belong to RV set are collectively noted as $\overline{S}_c$. The velocity in this set is reachable for the OS, and it helps the OS to avoid the collision:

$$\overline{S}_c = \left\{ \vec{v}_i \mid \left(\vec{v}_i \in \mathrm{RV}\right) \cap \left(\vec{v}_i \notin \mathrm{VO}\right) \right\}. \tag{6.4}$$

According to Equation (6.2), the TCR for the OS can be interpreted as the proportion of the overlap (S_c) to the RV set ($S_c + \overline{S}_c$):

$$\mathrm{TCR}(t) = \frac{\mathcal{N}\left(S_c(t)\right)}{\mathcal{N}\left(S_c(t) + \overline{S}_c(t)\right)}, \tag{6.5}$$

where $\mathcal{N}(\cdot)$ means the size of a set;

$S_c(t)$ and $\overline{S}_c(t)$ refer to corresponding sub-areas at time t.

A bigger sub-set $S_c(t)$ on RV set shows the fewer solutions for OS avoiding the collision, which indicates a higher risk (TCR(t)); a smaller $S_c(t)$ implies adequate collision-free solutions and means a lower risk (TCR(t)).

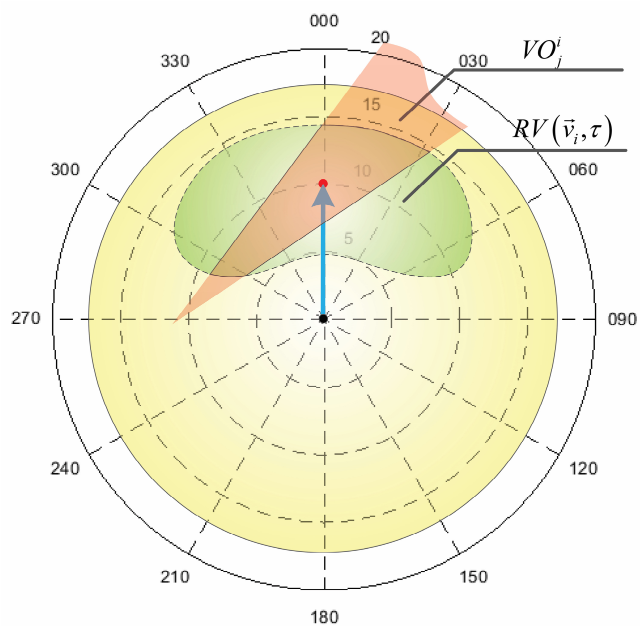


Figure 6.3 Illustration of VO set and RV set in velocity space of the OS.
(note: a point in this space represents a velocity of the OS: the axes indicate the course; the distance from the pole represents the speed)

6.2.4 VO set in velocity space

The velocity leading to a collision is collected in VO set. To simplify the problem, it is assumed that the ship is represented as a circle.

Assumption (6.3): The shape of ships is represented as circles in which the length of the ship is its diameter.

The Assumption (3) is introduced to make the collision cone independent of ships' shape, and the headings of ships would not impact on the shape of VO set.

According to Chapter 4 and Chapter 5, there are two ways to collect the velocity leading to collisions, i.e., VO algorithm and GVO algorithm.

6.2.5 RV set in velocity space

RV set is determined by the dynamics of the ship. The construction of RV set contains two cases.

(1) Using GVO algorithm

If GVO algorithm is used to find the velocity leading to collisions, i.e., UO set, the construction of RV set is equivalent to the problem that finds the boundaries of the desired velocity $\mathbf{u}^*$ given constraints on maximal input forces. Let say, the force in each direction is satisfying constraints:

$$\boldsymbol{\tau}_{\mathbf{lb}} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\mathbf{ub}}, \tag{6.6}$$

where $\boldsymbol{\tau}_{\mathbf{lb}}$ is the lower bound of input force and $\boldsymbol{\tau}_{\mathbf{ub}}$ is the upper bound.

Then, Equation (5.4) is substituted into Equation (5.3) to have an expression of forces with respect to the desired velocity $\mathbf{u}^*$, i.e.:

$$\boldsymbol{\tau} = \left(K_p - K_d V f_2\right)\mathbf{u}^* - \left(K_p V \mathbf{x} - K_d V f_1\right), \tag{6.7}$$

where, $f_1 = \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} + \mathrm{BK_d V}\right)^{-1} \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ -\mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathrm{K_p V}\mathbf{x} \end{bmatrix}$;

$$f_2 = \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} + \mathrm{BK_d V}\right)^{-1} \mathrm{BK_p}.$$

Combining Equation (6.7) and (6.6), the constraints on the desired velocity $\mathbf{u}^*$ are derived:

$$\left(K_p - K_d V f_2\right)^{-1}\left(\boldsymbol{\tau}_{\mathbf{lb}} + K_p V \mathbf{x} - K_d V f_1\right) \le \mathbf{u}^* \le \left(K_p - K_d V f_2\right)^{-1}\left(\boldsymbol{\tau}_{\mathbf{ub}} + K_p V \mathbf{x} - K_d V f_1\right). \tag{6.8}$$

Equation (6.8) is the reachable velocity set, satisfying the constraints on forces given a PD controller.

(2) Using VO algorithms

One simple way is simplifying the dynamics of the ship as a holonomic vehicle, and define the RV set as the maximal velocity one ship can achieve. Since backing speed is not common, the RV set can be simplified as the half of the whole V-space of the OS, i.e., velocity in the front direction accepts $v_x \in [0, v_{\max}]$, and velocity in the sideway direction is $v_y \in [-v_{\max}, v_{\max}]$. In this case, the ship has ideal maneuverability.

Another way is decoupling the construction of VO set and RV set. Specifically, the RV set is formulated as a problem of finding all the speeds one ship can achieve in a given time window. In this case, many velocities become unreachable since the collision might happen before the ship reaches such velocity. Thus, the area of RV set is shrunk compared with the one neglecting ship's maneuverability. The detail of constructing such RV set is shown in Appendix III.

6.3 Case studies

To demonstrate the performance of TCR measure in various scenarios, three groups of scenarios are simulated. The measurement of TCR in multiple-ship encounters is shown in Group 1 in Section 6.3.1, and that of TCR in different traffic modes are presented in Section 6.3.2. In Section 3.3, a demonstration of TCR measurement considering ship dynamics and constraints is shown.

In the following scenarios, the ship follows following assumptions: the chance of choosing each solution is equal; the trajectory of the TS is known; the shape of the ship is represented as a circle; the OS is a holonomic vehicle which can immediately change its velocity.

6.3.1 Scenario 6-I: performance of TCR measure in multiple-ship scenarios

Three encounter scenarios have been simulated to show the performance of the TCR measure. In each scenario, the own-ship is placed at the origin heading to the North with speed at 10 knots, while the number of target ships is increasing from one to three:

Table 6.1 Scenario 6-I: settings of scenario

	Position [NM]	Heading [30]	Speed [knot]	DCPA [NM]	TCPA [h]
OS	(0,0)	000	10.0	-	-
TS1	(0.65,-1.44)	358	16.0	0.5	0.25
TS2	(-2.45,1.80)	081	8.5	0.5	0.25
TS3	(3.38,3.02)	268	14.7	0.5	0.25

(1) Single-encounter case: the OS only encounters with one target-ship (TS1) whose DCPA is 0.5 [NM] and TCPA is 0.25 [h].

In a single-encounter scenario, the target ship (TS1) approaches the OS from its stern and TS1 blocks the starboard-turn options of the OS, as shown in Figure 6.4 (2). The blue area is the VO set which collects the velocity of the OS leading to a collision with the TS1. The rest of the area is collision-free for the OS, which also can be interpreted as the "room-for-maneuver". According to Section 6.2, the percentage of VO set shows the danger level of the OS, which is 0.749. That means, the ship still has 0.251 chance to avoid the collision.

(2) Two-ships-encounter case: One extra target-ship (TS2) is introduced whose the DCPA and TCPA remain the same as those of the TS1.

When one more ship (TS2)is introduced whose DCPA and TCPA are identical to TS1, the entire collision risk is undefined by traditional methods (e.g., CRI methods), especially the new ship has the same TCPA/DCPA with the TS1. It can be expected that more ships in the same area might increase the collision risk, but how to calculate the entire collision risk using the risk value in each pairwise encounter is unclear.

The TCR measure offers a solution to this problem. One more ship blocks some extra "room-for-maneuver", which leads to less chance to avoid collision dangers, as shown in Figure 6.4 (2). As a result, the encounter scenario would be more dangerous than the previous scenario. As indicated in the figure, the value of TCR, in this case, raises from 0.749 to 0.925, which means the number of solutions for the OS to avoid collision decreases and the situation that the OS faced becomes more dangerous than the previous case.

(3) Multiple-encounter case: the OS encounters with three target ships together, namely TS1, TS2, and TS3. The details of the settings are presented in Table 6.1.

When the OS encounters with three ships together, the area of "room-for-maneuver" is further shrunk, and TCR increases. However, the increasing range is less than the range when TS2 is introduced because most of the solutions leading to a collision with TS3 is inclusive in previous VO sets.

When the whole velocity space is occupied by the VO sets, i.e., the value of TCR reaches 1, the collision becomes inevitable by the OS alone. At the moment, the cooperation among ships are needed, i.e., the TSs also need to take evasive actions.

From these simulations, it is observed that the value of TCR would not reduce when one more target-ship is introduced. The introduced ship might block several velocities that lead to collisions, which increases TCR. In the extreme case, TCR would remain the same if the introduced VO set is a subset of existing VO sets or the VO set is outside of the reachable velocity set.

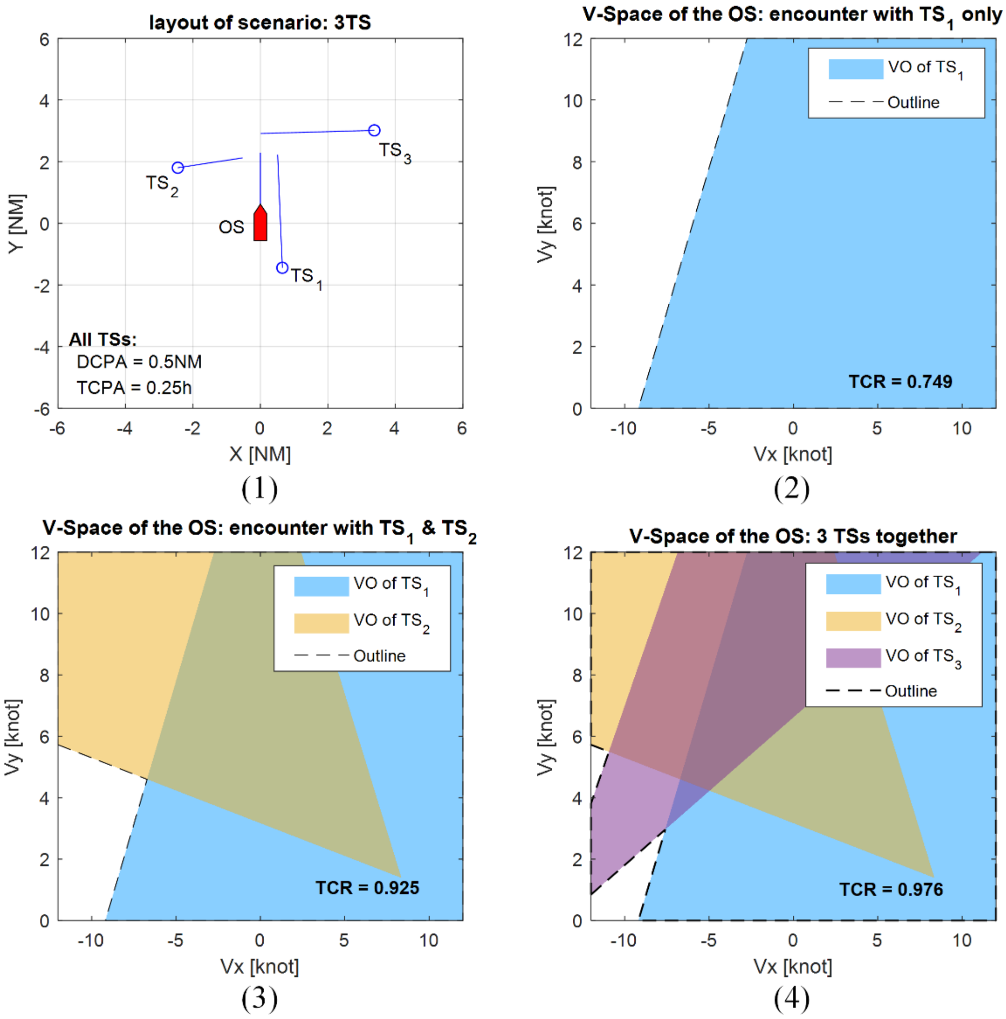


Figure 6.4 Scenario 6-I: (1) multiple-encounter scenario and *V*-Space of the OS when the ship encounters with (2) one TS, (3) two TSs and (4) three TSs.

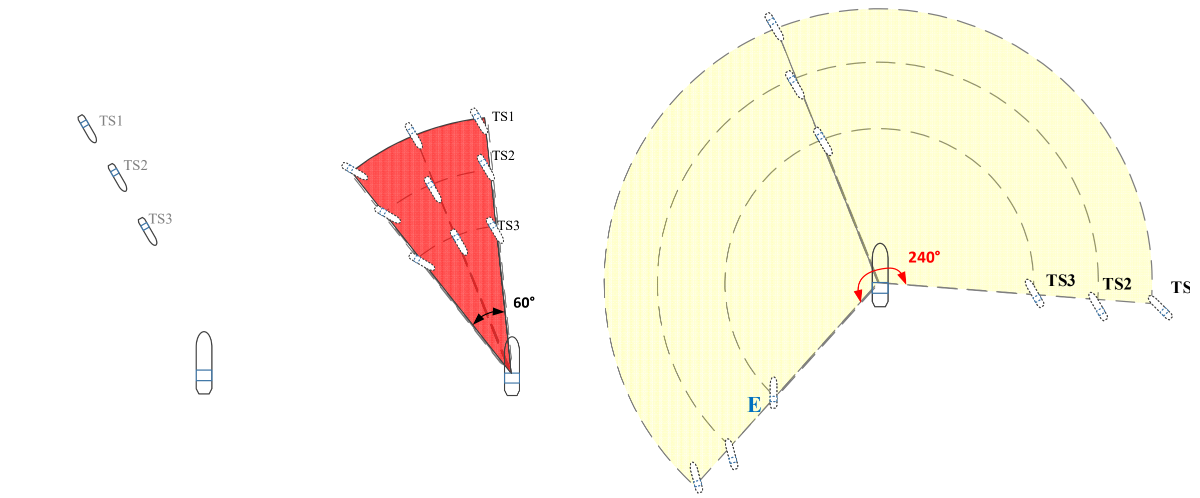
6.3.2 Scenario 6-II: well-organized traffic versus chaotic traffic

In this section, the influence of traffic modes on the measurement of TCR is shown. Three scenarios are simulated, in which three target ships are involved, namely TS1, TS2, and TS3. The same ship in different scenarios has the same DCPA, TCPA, and relative distance, while the position and velocity are slightly different. For example, the TS1 in each scenario has different positions and speeds, but the DCPA, TCPA, and relative distance remain the same.

(1) **Well-organized case**: three ships are grouped as a vessel train [49]. Specifically, these vessels have the same velocity and keep the formation.

(2) **Disorder case**: each target ship keeps its relative distance to the OS in the first scenario, but the bearings of each target ship are different, and the settings are shown in Table 6.2. (See Figure 6.5 (2));

(3) **Chaotic case**: the ship encounters with vessels from different bearings. An illustration is shown in Figure 6.5(3), and the settings are shown in Table 6.2.



(1) Well-organized case (2) Disorder case (3) Chaotic case

Figure 6.5 Scenario 6-II: illustration of three different traffic flows (1) well-organized, (2) disorder, and (3) chaotic case.

Table 6.2 Scenario 6-II: settings of scenarios

Case		**Ship**	**Position [NM]**	**Heading [deg]**	**Speed [knot]**	**DCPA [NM]**	**TCPA [h]**
		OS	(0,0)	000	10.0	-	-
Case (1)	Well-organized case	TS1	(-0.29,4.99)	174	10.0	0.05	0.25
		TS2	(-0.19,4.00)			0	0.20
		TS3	(-0.10,3.00)			0.05	0.15
Case (2)	Dis-order case	TS1	(-1.23,4.84)	153	10.6	0.05	0.25
		TS2	(0.14,4.00)	184	10.0	0	0.20
		TS3	(-0.46,2.97)	161	10.1	0.05	0.15
Case (3)	`Chaotic case	TS1	(4.14,2.81)	265	16.5	0.05	0.25
		TS2	(-2.56,3.07)	112	13.9	0	0.20
		TS3	(2.98,0.40)	070	21.0	0.05	0.15

Many traditional methods cannot indicate the entire collision risk of the OS in these cases. CRI methods and CPA methods are depending on the DCPA, TCPA, and relative distance, while the DCPA, TCPA, and relative distance of TSs in each case are the same. Thus, the collision risks between the OS with TSs are invariant. Hence, it might be concluded that the collision risk in each scenario is the same. However, the OS in the last scenario seems to be more dangerous than the others because the OS might not easy to avoid collisions.

The TCR measure can capture the difference in collision risk in these scenarios. Figure 6.6 (1)-(3) show the layouts of each scenario and (4)-(6) show the V-space of the OS in relevant scenarios. The value of TCR indicates that the collision risks in these scenarios are different, which are 0.61, 0.62, and 0.93. The three VO sets from three target-ships have been identified in each scenario.

In the first case, the traffic is well-organized, and all these VO sets are contained in one VO set generated by TS3. That means, if the OS can avoid collision with the TS3, the ship can avoid collision with TS2 and TS1, as well.

In the second case, some solution preventing collision with TS3 cannot help the OS avoid the collision with TS2, e.g., the OS turns port side by 90 degrees and reduces the speed to 2.5 knots, i.e., [-2.5, 0] knots. Thus, the value of TCR slightly increases in this case.

In the last case, each VO set blocks different groups of maneuvers. For instance, the collision-free solutions for avoiding TS3 (the bottom-left area) are blocked by TS1. That means, even if the solution can help the OS to avoid the TS3, it might not avoid collision with TS1. As a result, the collision danger is more difficult to avoid. Thus, the collision risk is high.

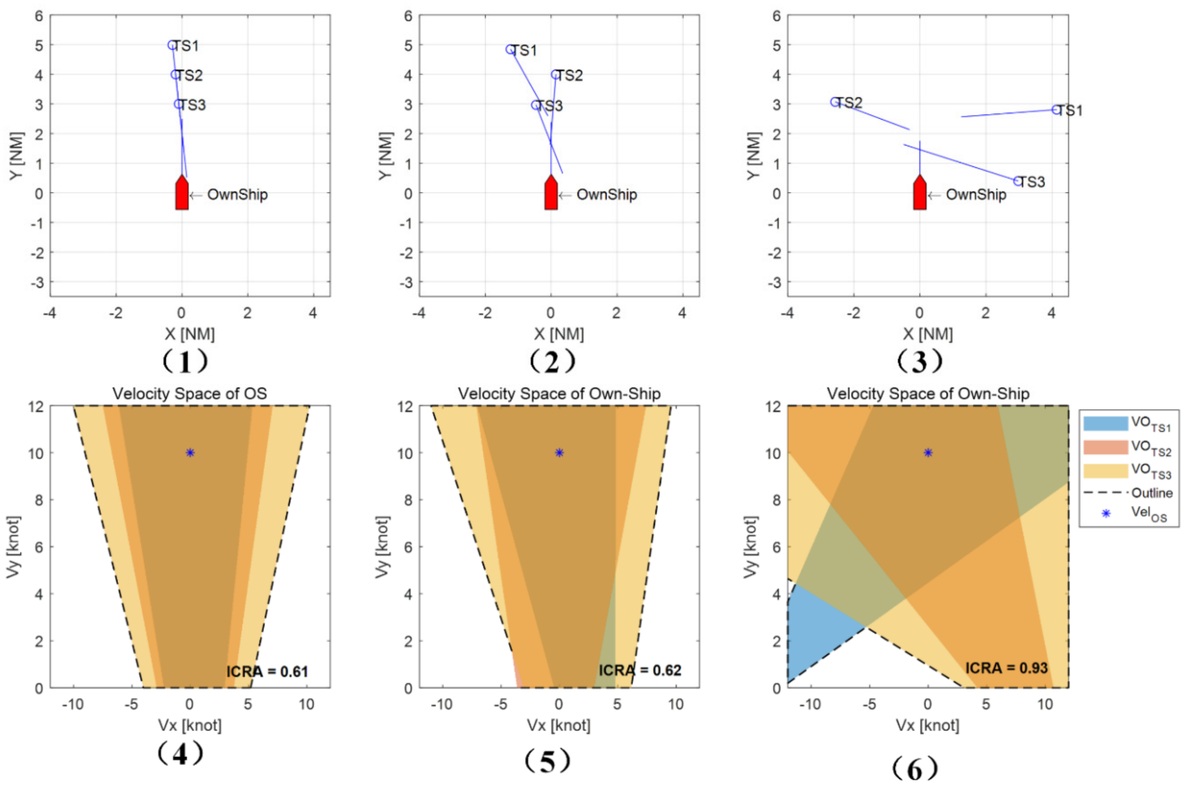


Figure 6.6 Scenario 6-II: TCRs and interfaces of the OS encountering with three traffic flow ((1)&(4) well-organized case, (2)&(5) disorder case, and (3)&(6) chaotic case).
(Note: the DCPA, TCPA and relative distance of one ship, e.g., TS1 in case 1, are the same as the ship in other cases.)

6.3.3 Scenario 6-III: good maneuverability versus poor maneuverability

In the previous scenarios, the maneuverability of the ship is out of consideration. In this scenario, the changes in TCR are investigated when the maneuverability of the ship is considered.

Since the decision has to be made before a collision happens, the ship does not have infinite time to change its motion. Given a time window, the ship with good maneuverability can achieve more velocities than the one with poor maneuverability. For instance, the holonomic ship can achieve arbitrary velocity immediately. However, when the ship has poor maneuverability, the size of the RV set is shrunk since some velocities are infeasible in a given time. In this scenario, the time window for taking actions is identical to Time To Collision (TTC). The details of construction of RV set considering maneuverability is displayed in Appendix III.2.

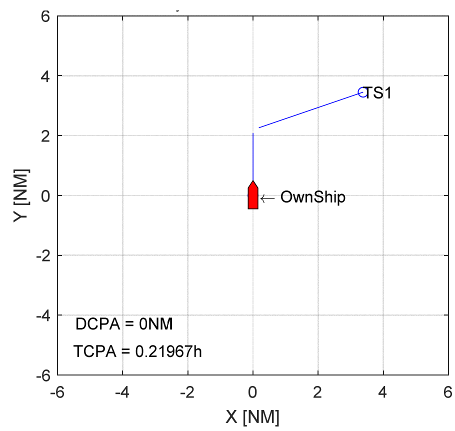
A two-ship encounter scenario is simulated, where the TS encounters with the OS from the starboard side of the OS, see Figure 6.7 (1). The detailed settings are shown in Table 6.3.

The *V*-space of the holonomic ship and the ship with maneuverability are shown in Figure 6.7 (2) and (4) where the existing velocity of the OS is marked as blue "*". The red regions in these panels are VO sets whose sizes are equivalent. However, the sizes of RV sets are different. When the ship is holonomic, the ship can obtain arbitrary speed smaller than the maximal speed, and the changes of heading are usually smaller than 90 degrees in both sides. Thus, the RV set is a semicircle, see Figure 6.7 (2). The RV set considering ship's maneuverability is shown in Figure 6.7 (4), where some velocities in the semicircle become infeasible since the time window is limited.

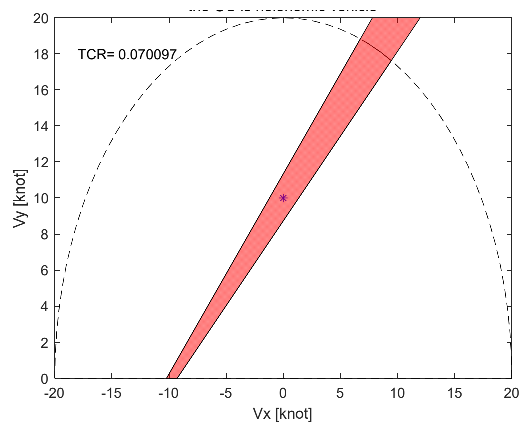
From these figures, it can be seen that the ship with good maneuverability (Figure 6.7 (2)) has a lower TCR than the one with poor maneuverability. This result is not surprising that the ship with poorer maneuverability has fewer reachable solutions, thus a higher TCR, than the one with good maneuverability, in the same condition.

Table 6.3 Scenario 6-III: settings of scenario

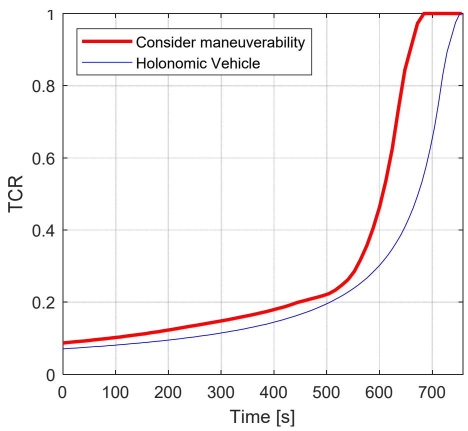
	Position [NM]	Heading [30]	Speed [knot]	DCPA [NM]	TCPA [h]	Radius [NM]
OS	(0,0)	000	10.0	-	-	0.10
TS1	(3.378,3.451)	250	16	0	0.22	0.10



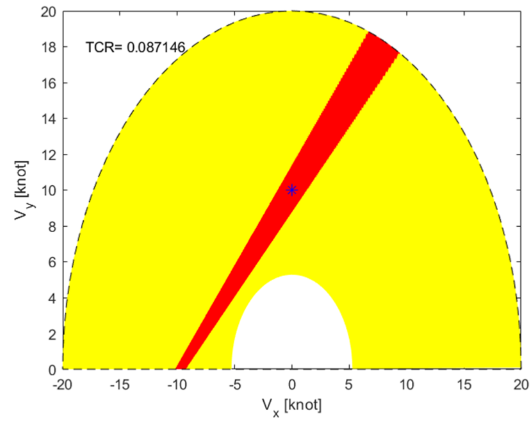
(1) Two-ship encounter scenario at 0 [s]



(2) *V*-space of the holonomic OS



(3) TCR before collision



(4) V-space considering maneuverability

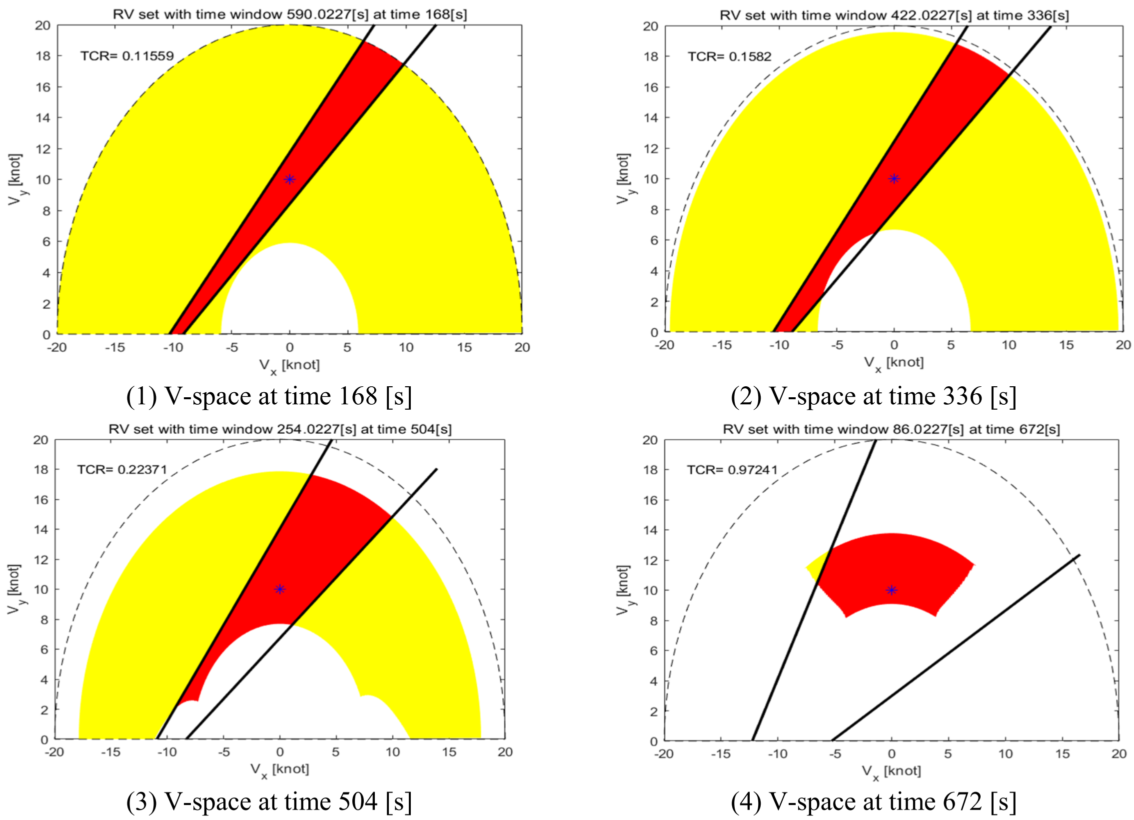
Figure 6.7 Scenario 6-III: comparison of collision risk with good and bad maneuverability.

In Figure 6.7 (3), the developments of the value of TCR are recorded, where the blue line shows the TCR level neglects the maneuverability of the OS and the one in red is incorporating the ship's maneuverability. As shown, the TCR level is underestimated when the maneuverability is neglected.

More details of RV sets at different time slices (168 [s], 336 [s], 504 [s], 672 [s]) are shown in Figure 6.8. With time increasing, two ship approach to each other and more and more velocities could lead to a collision. Thus, the VO set is enlarging. On the other hand, since the time to collision is decreasing, the number of reachable velocities is reducing, and the RV set is shrinking.

At time 504 [s], the time to collision is around 254 [s], and the ship still has time to apply evasive actions, and TCR rises to 0.22. It implies that the OS still has sufficient time and space to avoid the collision with the TS. At time 672 [s], the time to collision reduces to 86 [s], and TCR reaches 0.97. That means, the OS is in an urgent situation and only the port-turning solutions are feasible and collision-free for the ship.

When TCR reaches 1, it means there is no feasible solutions can help the ship avoid collisions. That means, although the collision has not happened yet, the OS cannot avoid collision by itself. Then, cooperation between ships would be needed.



(1) V-space at time 168 [s]
(2) V-space at time 336 [s]
(3) V-space at time 504 [s]
(4) V-space at time 672 [s]

Figure 6.8 Scenario 6-III: *V*-space of the OS during the conflict (before the collision).

6.4 Discussion

This chapter introduced a novel collision risk measure, called time-varying collision risk (TCR) measure. This measure is proposed to support the users to identify the moment in time to avoid collisions, which is incorporating the resolution of conflict and is suitable for multiple-encounter scenarios.

6.4.1 Using the TCR for collision prevention in HMI-CAS

The TCR measure is using the VO set to identify dangers. When the OS's velocity is falling a VO set of the TS, it indicates that the OS has a conflict with the TS. When the velocity is located in an overlap of multiple VO sets, the OS has conflicts with multiple TSs simultaneously.

When the velocity of the ship falling in any VO sets, the HMI-CAS is triggered to offer a collision-free solution to human operators. It means that the ship is suggested to take action. However, in practice, the human operators might not take evasive actions immediately for various reasons: firstly, the OS is a "stand-on" ship that is advocated to keep course and speed by COLREGs rules; secondly, the human operators are risk-takers. In these two cases, there is still a need to show the user the last moment to take evasive actions and value of TCR plays a role.

The value of TCR shows the chance of the ship avoiding collisions, which offers support to human operators using HMI-CAS. Specifically, when the operator decides to postpone the prevention process by the machine, the TCR shows the margin of safety. The HMI-CAS still can find an executable collision-free solution before the value of TCR reaches 1. However, if the TCR reaches 1, it would be no collision-free solution and the coordination between ships will be needed. Thus, the evasive action is suggested to be taken before the TCR reaches 1.

6.4.2 Features of the TCR measure comparing with other measures

Three main differences between the traditional risk measures presented in Section 2.4 and the proposed TCR measure are concluded as follows. Firstly, the TCR measure incorporates the chance of avoiding a collision. However, most of the risk measures ignore this part. Secondly, the TCR measure maps all the obstacles into resolution space (i.e., V-space) together and then assesses the collision risk, whereas the traditional measures decouple the traffic first and then assess the risk in each pair of ships. Thirdly, the construction of the TCR measure is relatively independent of the experts' judgment.

(1) The proposed measure incorporates the ability of the OS to avoid a collision.

The TCR measure considers the maneuverability of the OS and conflict resolution in the measurement of collision risk. In return, the measurement of TCR not only presents the danger levels of the threats (TSs) but also reflects the ability of the OS to avoid dangers. When the TCR exceeds the thresholds, it basically tells the officer on board that the coming threats (TSs) are not only dangerous but also difficult to find a solution to prevent the collisions.

(2) The proposed measure assesses the collision risk of traffic as a whole.

The TCR measure assesses the collision risk as a whole, which can prevent biases caused by decoupling the traffic into the pairwise. Specifically, the decoupling technique ignores the influence of other ships and loses some information about the traffic. An example is shown in section 6.3.2. A well-organized scenario is less dangerous than the traffic in chaos, even the risk index of each pair in these scenarios remains the same. Most of traditional methods cannot show these differences to the users, but the TCR measure could.

Moreover, traditional methods only show the risk in pairs and ignore the impacts of other ships. Thus, if one solution reducing the risk in one pair of ships is found, it cannot be guarantees that this solution can also reduce the risk in another pair. In some worse cases, this solution might create some new conflicts. Thus, the users need to try and test the solutions in each pair of ships, until they can find the one which reduces all the conflict in each pair of ships. Conversely, the TCR measure assesses the collision risk as a whole, and it can directly identify collision-free solutions to the users. The target-ship which is temporarily not in conflict is also considered in the TCR measure. As a result, the solutions identified by the TCR method can solve all the conflicts and would not create a new conflict.

(3) The proposed measure is independent of the experts' judgment.

The measurement and the meaning of TCR are independent of experts, while the construction of many traditional methods strongly relies on expert's knowledge. Moreover, there is a lack of general agreements on the parameters in traditional methods, e.g., CRI [71]. That means the same scenario might have different CRIs and different conclusions when the different experts are involved. On the other hand, the construction of the VO set relies on the obtained traffic data, and the RV set depends on the maneuverability of the ship, which are relatively independent of experts. Additionally, the meaning of TCR is also clear. When the value of TCR reaches 1, then the OS inevitably collides with obstacles, even if the collision has not happened yet. When TCR is 0.5, that means if the OS chooses the solutions randomly, the ship still has 50% to collide with other ships.

6.4.3 Remarks of collision risk for ship

Through three cases studies, some findings related to the changes in collision risk are concluded.

Remark 6.1: collision risk would not reduce when the ship encounters one more ship. (Section 6.3.1)

Remark 6.2: the collision risk is underestimated when the ship's maneuverability is ignored. (Section 6.3.2)

6.4.4 Potential applications

The TCR measure offers a new perspective to assess collision risk, which can enrich the tools for risk-informed decision making on board and in a VTS center. In [71], researchers proposed a framework for risk-informed collision alert, which helps share situational awareness between experts and OOWs. Some widely used indicators are listed, but few indicators reflect the ability of the ship to avoid a collision and consider the entire traffic. The measurement of TCR can be used as one indicator in this framework, which offers some information about the difficulty of the OS ship to avoid collision with the entire traffic.

The TCR measure is also helpful to find collision candidates from historical AIS data, which is more stable and less sensitive than the traditional methods. Estimation of collision candidates is an essential task of assessing risk for waterway safety management. The traditional methods are based on D/TCPA or ship domains, which easily overestimates the number of collision candidates. For instance, a pair of ships might have a conflict with each other in a period of time. The traditional methods would identify multiple collision candidates since the conflict at each time slice is seen as one collision candidate. However, the TCR measure could successfully identify this process as one conflict process with two collision candidates. For the readers interested in this topic, more details are presented in [191].

6.5 Conclusions

The measurement of collision risk is a vital step for supporting situational awareness of the human operators and for trigging collision avoidance in the HMI-CAS in time. This chapter is drawn to answer the research question associated with conflict detection in Section 1.3: *How can the conflict detection module in HMI-CAS be improved, supporting that the human operators take evasive actions in time?* The answers are concluded as follows:

- Firstly, the existing techniques for conflict detection are overviewed and it is found that these methods have two limitations on triggering evasive actions. (1) Most of the existing measures assess the risk in the pairwise encounter, while the ship might encounter with more than one ships. (2) The measured risk is usually independent of the conflict resolution. Thus, the risk level addressed limited information about whether the collision is avoidable.

- To overcome these issues, a collision risk measure, called Time-Varying Collision Risk (TCR) measure, is proposed. The TCR is measured as the percentage of maneuvering solutions leading to a collision in a reachable solution space. This percentage also reflects the difficulty of avoiding an upcoming collision.

- The measurement of TCR is serving as an auxiliary tool in the conflict detection module of the HMI-CAS. When the human operators plan to postpone the evasive actions suggested by the system, this indicator shows the "room-for-maneuver" of the ship and the operators need to decide before this indicator reaches 1.

The value of the TCR directly reflects the Officer On Watch (OOW)'s *room for maneuver* to avoid a collision, which helps their awareness of the safety level of approaching ships. Three features are highlighted: (1) it measures the collision risk considering conflict resolution; (2) it measures collision risk of the entire traffic instead of decoupling the traffic, which is more suitable in multiple-ship scenarios; (3) the measure is independent of experts' opinions.

Simulation experiments are carried out to show the potential of the TCR measure. The results show that the collision risk increases when the ship encounters with more ships, the encountering traffic is in chaos, and the ship has a poor maneuverability. The evolution of TCR shows the changes of risk that support the human operator to take evasive actions. The evasive actions from the OS should be applied before the TCR reaches 1, and when the TCR reaches 1 cooperation between ships is needed to prevent collisions.

The proposed TCR measure offers a new perspective on assessing collision risk, which not only enriches the choices in the developments of risk-informed collision alert systems but also supports the risk-based collision avoidance in multiple-ship scenarios.

Chapter 7 Conclusions and Future Research

This dissertation proposes a prototype of a collision avoidance system that supports Human-Machine Interactions, which is named as Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS). The proposed system enables cooperation and interactions between human operators and the automation system in collision avoidance, which can be used in both manned ships and unmanned ships.

The main findings and the answers to research questions are addressed in Section 7.1. Section 7.2 discusses the remaining limitations and points out directions for future research.

7.1 Answers to research questions

The research objective of this study is to develop a collision avoidance system across manned and unmanned platforms, which facilitates human-machine interactions (HMIs) during ship collision avoidance. The main research question addressed in this dissertation is:

> **Main Question:**
> *How can a Collision Avoidance System* (*CAS*) *be designed for both manned- and unmanned ships considering Human-Machine Interaction* (*HMI*)*?*

To develop such a system, the state-of-the-art of collision prevention techniques is investigated in Chapter 2, where a generic framework of collision avoidance is explored. Based on that, a CAS framework for supporting HMIs is proposed in Chapter 3, together with the requirements on prevention techniques. By comparing the requirements and existing techniques, a family of Velocity Obstacle (VO) algorithms are found and adopt for the development of the HMI-CAS (Chapter 4). An improvement of the HMI-CAS in terms of conflict resolution is presented in Chapter 5, which incorporates the ship dynamics and navigational regulations. Another improvement takes place in Chapter 6, where a novel time-varying collision risk measure is proposed to support conflict detection. The simulations in Chapter 4-6 demonstrate the effectiveness of the proposed HMI-CAS and the relevant methods.

Following the main research question, five sub-questions raised in Chapter 1 are answered.

Question on the state-of-the-art:

1. *What techniques have been developed for collision avoidance at sea and what research gaps can be explored in a generic framework?*

Answer to Question 1:
To answer these questions, a generic framework of collision avoidance is proposed, which consists of five modules: "Observer", "Motion Prediction", "Conflict Detection", "Conflict Resolution", and "Controller". The state-of-the-art techniques for "Conflict Detection" and "Conflict Resolution" are the focuses.

Conflict detection is the main focus of conventional maritime studies, which aims at supporting human on board in three aspects: finding approaching dangers, setting off warnings for caution, and setting off warnings for evasions. The core of conflict detection is a measurement of collision risk. Six categories of measures are found. However, most of the existing methods are suffering from two limitations: (1) decoupling the traffic into multiple pairwise encounters; (2) independent of conflict resolution. Thus, the existing measures are not suitable for triggering evasive actions.

Conflict resolution gains remarkable attention in recent years due to the rapid development of autonomous ships. Six groups of collision prevention methods are identified in the literature. The advantages and disadvantages of these methods are given in Section 2.5.2. Two of the main challenges are as follows: many methods ignore the ship dynamics in finding collision-free solutions; incorporating navigational regulations is still an open question.

Moreover, the focuses of the studies for manned ships and unmanned ships are slightly different and complementary. The research for manned ships concentrates more on conflict detection, while the research for unmanned ships focuses on conflict resolution. Thus, it is natural to learn the advantages of each other. However, developing unmanned ships with various autonomy levels not only requires improvements in conflict detection and conflict resolution but also incorporates a user-friendly design of autonomous systems. Specifically, the design of the system supports various forms of human-machine interactions. Unfortunately, this kind of design is less focused in the existing studies.

Question on the framework of HMI-CAS:

2. *How can the framework of HMI-CAS be designed to support various modes of human-machine interactions during collision prevention?*

Answer to Question 2:
Chapter 3 firstly analyzes the demand on HMIs in various types autonomous ships, and then concludes the essential modes of interactions during collision avoidance. The identified interactions include "early warning", "solution advice", "automated solution execution",

“cooperation support”, “trajectory prediction”, “data collection”, and “turning off/on the system”.

The proposed HMI-CAS system focuses on “cooperation support” and “solution advice”, which are fundamental for MASS Type I-III and control Mode 1-4. The “cooperation” means that the “machine” and “human” work together to find collision-free solutions. Five forms of supports provided by machines for human operators are distinguished: one solution, an optimal solution, finite feasible solutions, a set of feasible solutions, all sets of unsafe solutions.

To support autonomous functions, the framework of the HMI-CAS is similar to the typical Guidance, Navigation, and Control (GNC) system. Nevertheless, to deliver information between the human operators and the system, an interface is designed. This interface is integrated into the “Local Planner” in the “Guidance” module of the GNC system.

Question on methodology:

3. *What are the proper methods that support human-machine interactions in conflict detection and conflict resolution of HMI-CAS?*

Answer to Question 3:
A comparison of the existing collision prevention techniques is addressed in Chapter 4. Among the methods, a family of Velocity Obstacle (VO) algorithms are found to be suitable for the proposed HMI-CAS as they can collect the solutions leading to dangers and visualize the solution spaces to users.

Therefore, a group of VO algorithms are applied to the maritime environment, e.g., Linear VO (LVO) algorithm, Non-Linear VO (NLVO) algorithm, Probabilistic VO (PVO) algorithm. Additionally, the LVO is identical to conventional techniques from the maritime studies, such as Collision Threat Parameter Area (CTPA), Collision Danger Section (CDS), etc. However, the variations of VO algorithms enrich their applications in practice.

Comparing with traditional methods, the VO algorithms do not presume that the TSs move with constant speed and course. Thus, the HMI-CAS using the introduced VO algorithms is able to incorporate various types of predicted trajectories, e.g., non-linear, probabilistic, etc.

Question on improving the conflict resolution module:

4. *How can ship dynamics and navigational regulations be incorporated in HMI-CAS for conflict resolution?*

Answer to Question 4:
In Chapter 5, a modified Generalized Velocity Obstacle (GVO) algorithm is integrated into the HMI-CAS. The GVO algorithm takes the controllers and the dynamics of ship into consideration. Modifications are made when using the GVO algorithm in the HMI-CAS framework.

Firstly, the interface of the HMI-CAS using GVO algorithm is re-designed to be user-friendly. In particular, the solution space is reduced to two dimensions, so that the solution space can be presented as a 2D map like nautical charts and radar figures which are familiar to the operators on board. Additionally, the origin of solution space in moved to the existing state of the system, so that the chosen solution in solution space is compliant with the behaviors of the ship. For example, the solution in the left-hand side of solution space represents the portside turning.

Secondly, to find a solution compliant with navigational rules, certain rules from COLREGs are integrated to find collision-free solutions. The improved HMI-CAS enables to find rule-aware solutions in certain simple encounter scenarios. Moreover, if there are no rule-aware solutions, the HMI-CAS can also offer an optimal collision-free solution via an optimization process.

Question on improving the conflict detection module:

5. *How can the conflict detection module in HMI-CAS be improved, supporting that the human operators take evasive actions in time?*

Answer to Question 5:
In Chapter 2, the existing achievements and research gaps in conflict detection have been discussed. The review shows that the existing risk measures have two limitations on triggering evasive actions. In particular, most of the existing risk measures assess the risk in pairwise encounters, and they are independent of the conflict resolution.

To overcome these issues, Chapter 6 proposes a new method that assesses the collision risk via the solution space. Specifically, the collision risk is defined as the percentage of solutions leading to a collision in a reachable solution space. This ratio reflects the difficulty of avoiding the upcoming collisions, which incorporates ship maneuverability, and it is applicable to handle the multiple-encounter scenarios. Simulation results show that the proposed measure is compliant with common sense: collision risk would not decrease when the ship encounters with more ships and when the ship has poor maneuverability.

This risk indicator can serve as an auxiliary tool in the conflict detection module of the HMI-CAS. When the human operators plan to postpone the evasive actions suggested by the system, this indicator shows the “room-for-maneuver” and the operators need to decide before this indicator reaches 1.

In summary, a brief answer to the main research question is presented as follows:

The design of CAS needs to incorporate self-navigation of the unmanned platform and to support interactions between humans and machines. Thus, the framework of the proposed HMI-CAS adopts GNC as a foundation and adds an interface to support HMIs. However, not all the GNC systems adding an interface can satisfy various demands of HMIs.

The prevention algorithm in the HMI-CAS is the key. The algorithm should not only be able to avoid collision automatically but also can offer intuitive outputs for human operators to read, to understand, and to interact. Then, a family of velocity obstacle algorithms, which firstly visualizes the solution space and then finds an optimal solution, is found. Thus, a HMI-CAS using generalized velocity obstacle algorithm is developed in this thesis. The HMI-CAS has the functions of autonomous collision avoidance, visualization of the solution space, and allows the human operators to modify the selected solution via the designed human-machine interface.

7.2 Recommendations for future research

In this thesis, a prototype of the HMI-CAS for a ship in the environment with multiple TSs is proposed, which aims at supporting conventional ships or unmanned ships supervised /monitored by human operators to prevent collisions at sea. Potentials of the proposed system and challenging issues that require future research are listed below.

Recommendation on improving HMI-CAS for applying in practice.

In the proposed prototype, there are still some limitations that need to be overcome before applying the HMI-CAS in practice:

(1) The uncertainties of ship dynamics need to be taken into account, which are introduced due to environmental disturbances, simplification of models, etc.

(2) The development of the prediction module is necessary when TSs do not broadcast their trajectories. Then, prediction algorithms are needed, and the confidence levels of prediction are also required.

(3) A global planner would be incorporated in HMI-CAS in the future, which is needed to prevent deadlocks. The deadlocks are the situations in which the ships block each other in a way such that none of them can continue sailing without collisions.

(4) Incorporating the irregular shapes of ships in the HMI-CAS should be considered. In narrow channels or waters with traffic separate scheme, the circular assumption on the ship's shape is no longer applicable. It might result in no collision-free solution in dense waters and could lead to an overestimation of TCR.

(5) Under-actuated ships should be considered in future studies since most of the conventional ships are under-actuated.

Recommendation on installing HMI-CASs in conventional ships to join the heterogeneous traffic flow with unmanned ships.

This dissertation presents the applications of the HMI-CAS in one ship in detail. However, the HMI-CAS is also applicable to supporting collision avoidance in a homogenous waterborne traffic system. A demonstration is presented in [69], where all ASVs installing HMI-CASs can successfully avoid collisions in an waterway

The next step, which is more attractive, is using HMI-CAS for a heterogeneous traffic system in which autonomous ships and conventional ships are involved. This scenario would be inevitable and essential on the road to an autonomous era. Some researchers have proposed an elegant cooperative framework for traffic system with autonomous ships presented in [195], which uses Alternating Direction Method Of Multipliers (ADMM) to help ASVs make agreements on their future trajectories. However, the challenge is how to integrate the conventional ship in this framework.

The HMI-CAS can play a significant role in this process. The HMI-CAS can help the conventional ships to join the cooperative framework presented in [195]. A conventional ship installing the HMI-CAS can exchange its trajectory with other ships and updates the solution space accordingly. The operators, then, can choose one collision-free solution from the updated solution space. Then, the updated trajectory will be broadcasted. This process is repeated until all the ships achieve an agreement.

Recommendation on cooperation between ships with low/no burden on communication

Many existing CASs applied in a traffic system require a huge burden on communication, e.g., HMI-CAS, Vessel Train Fleet, etc. That means these systems rely strongly on the reliability of communication.

The GVO algorithm used in HMI-CAS enables the ships to cooperate with a low burden of communication. All the ships assign their priorities in advance and share their responsibilities to avoid collision according to the pre-set priorities, i.e., without needs for real-time communication systems. However, this function is not costless. The dynamics of the ships need to be known in advance. So that, one ship can calculate the total efforts need to spend for avoiding collisions and then share the responsibility with the other ship via pre-set priorities. This technique is also named as "generalized reciprocal collision avoidance", see [181, 182].

The generalized reciprocal collision avoidance works with a pre-condition that all ships will follow a protocol which determines the percentage of responsibility each ship should take in collision avoidance. When this technique is applied in the proposed HMI-CAS, the dangerous sets presented in the interface refer to the solution violating the protocol. If the other ship do not aware of the violation, the collision would happen. As long as the protocol is followed, the ship can avoid collisions with other ships. Based on that, the human operators are free to choose one solution from the collision-free solutions.

However, such a cooperative system requires perfect information of all ships in the traffic system (e.g., known dynamics, priorities, and protocol), and it cannot cooperate with a ship using other prevention techniques, i.e. a heterogeneous ship.

Recommendation on applying navigational rules in the automation system.

Incorporating the navigational regulations to CAS has been a dream of researchers working on this topic for decades. In a common effort, many CASs enable to suggest rule-compliant actions in certain simple encounter scenarios. However, teaching the machines to understand the whole regulations as captains is still challenging. To cope with this issue, three possible ways are recommended to pay attention to (1) explaining regulations for the machine, (2) learning rule-compliant actions from human, and (3) creating new regulations.

At the current stage, borrowing the knowledge from human operators via HMI-CAS is a realizable solution, i.e., the second way. In the future, though the interactions with human operators, the HMI-CAS can learn the experiences via artificial intelligent methods.

Generally, the regulations are not written for the machine. Thus, many terms are ambiguous. Thus, quantifying the regulations for the machine might not be efficient. One line of thinking is collecting the interaction data between human operators and HMI-CAS, especially the interventions from human operators. These data are seen as labeled data, and then the system is trained via the labeled data to learn the pattern of human operators. The HMI-CAS with learning function is expected to offer rule-compliant solutions that approach to the choices of human operators.

Recommendation on studying the impact of using HMI-CAS in social-technical systems

The HMI-CAS is proposed to support the human operators in preventing collisions. Some similar methodologies are tested in simulators, which shows the methods can facilitate the mariners to detect dangers and to solve the conflict. However, awareness is also needed that the possible negative consequence due to the misuse of the system.

In conventional navigational assistance systems, the solution space are not provided. Thus, the captains may incorporate the safety margin to avoid collisions. That means the operators would take safer solutions. However, when the solution space is presented, risk-takers may reduce their safety margin (e.g., choosing a solution close to the dangerous set) to pursue potential benefits (e.g., save time, save distance, etc.). From this perspective, whether the HMI-CAS would make the whole system safer is unknown.

Therefore, it is necessary to carry a careful investigation of the safety of such a social-technical system when a new technique is introduced. This step would be inevitable before this system can be spread and used on board. One possible way is using agent-based safety risk assessments [201, 202] to test the influences.

Recommendation on using the TCR measure in waterway safety management

In this dissertation, a family of VO algorithms is introduced for conflict detection and conflict resolution on board. The potentials of the VO algorithms are not limited to supporting the HMI-CAS. They can also be applied for the identification of collision candidates. Identification of collision candidates from historical AIS data is a fundamental step for assessing collision risk for waterway safety management. In [191], NLVO algorithm is applied to detecting collision candidates, which is more efficient and stable than traditional methods. However, as many other methods, the applied VO algorithm detects collision dangers based on pair-wise encounters, which departs from reality. It would be interesting to use the VO algorithms to identify collision candidates in multiple-encounter scenarios, which supports revealing the mechanism of collision in multiple ships scenarios. The output can be used to support the safety management in waterway intersections.

Appendix I Parts of COLREGs Regulations

A brief overview of the main operational requirements of COLREGs [194] related to our research is listed in this section:

Rule 8: Action to avoid collision.

(a) Any action taken to avoid collision shall be made in ample time. (b) Any alteration of course and/or speed shall be large enough and a succession of small alterations of course and/or speed should be avoided. (c) If there is sufficient sea room, alteration of course alone may be the most effective. (d) Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance. (e) If it is necessary to avoid a collision, a vessel shall slacken her speed or take all way off by stopping or reversing her means of propulsion.

Rule 13: Overtaking.

(a) A vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam. (b) Any subsequent alteration of the bearing between the two vessels shall not make the overtaking vessel relieve her of the duty of keeping clear of the overtaken vessel.

Rule 14: Head-on situation.

When two power-driven vessels are meeting on nearly reciprocal courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.

Rule 15: Crossing situation.

When two power-driven vessels are crossing, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

Rule 16: Action by give-way vessel.

Every vessel keeping out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.

Rule 17: Action by stand-on vessel.

The stand-on vessel shall keep her course and speed, except two cases: (i) the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules; (ii) the vessels are so close that collision cannot be avoided by the action of the give-way vessel alone. The vessel which takes action in a crossing situation in accordance with case (i) of this Rule to avoid collision with another vessel shall not alter course to the port side of a vessel on her own port side.

Appendix II Parameters of CyberShip II

CyberShip II is a scaled ship model with m = 23.8 kg and L_m = 1.255 m. The settings of these parameters are presented in Table II.1.

Table II.1 The parameters of CyberShip II [193]

m	23.800	Y_v	-0.88965	N_v	0.03130
I_z	1.760	$Y_{\dot{v}}$	-10.0	N_r	-1.900
x_g	0.046	Y_r	-7.250	$N_{\dot{v}}$	-0.0
X_u	-0.72253	$Y_{\dot{r}}$	-0.0	$N_{\dot{r}}$	-1.0
$X_{\dot{u}}$	-2.0	$Y_{\|v\|r}$	-0.845	$N_{\|v\|r}$	0.08
$X_{\|u\|u}$	-1.32742	$Y_{\|v\|v}$	-36.47287	$N_{\|r\|r}$	-0.750
X_{uuu}	-5.86643	$Y_{\|r\|v}$	-0.805	$N_{\|r\|v}$	0.130
		$Y_{\|r\|r}$	-3.450	$N_{\|v\|v}$	3.95645

Froude scaling law is applied in the simulation to link the scaled model and the real physical world, see Table II.2. The gravity keeps the same in scale model and the real world. The scale of length is $\alpha = 1/70$. The scale of velocity and time are $\sqrt{\alpha}$. Thus, the speed of the scale ship is 0.5 [m/s] is approximately equal to the full-scale ship speed 4.2 [m/s] (roughly 8 [knot]).

Table II.2 The scale relations between the model and the real world ($\alpha = 1/70$)

	Scale Model	**Real world**	**Relation**
Gravity	g_m [m/s^2]	g_s [m/s^2]	$g_m = g_s$
Length	L_m [m]	L_s [m]	$L_m = \alpha L_s$
Speed	V_m [m/s]	V_s [m/s]	$V_m = \sqrt{\alpha} V_s$
Time	T_m [s]	T_s [s]	$T_m = \sqrt{\alpha} T_s$

Appendix III Construction of Reachable Velocity Set

A RV set is used to figure out all the reachable velocities of the OS before collision. In this appendix, one way to construct RV set is shown that is decoupling with construction of VO set. This RV set is a function of time to collision and existing velocity of the OS. A RV set with a time window τ and the initial velocity $\vec{v}_i$ is noted as $RV(\vec{v}_i, \tau)$, see Figure III.1. The yellow zone is the RV set of the OS regardless of the TTC (Time To Collison). The green zone is the RV set considering the TTC (i.e., $RV(\vec{v}_i, \tau = \text{TTC})$).

The angular range of $RV(\vec{v}_i, \tau)$ reflects the turning ability of the OS, and the radius range of $RV(\vec{v}_i, \tau)$ shows the acceleration/deceleration ability. A bigger time window or better maneuverability contributes to a larger RV set.

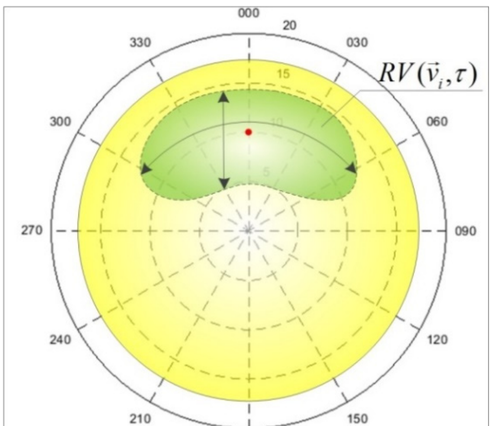


Figure III. 1 Illustration of reachable velocity set in velocity space of the OS.

The construction of the RV set is, in fact, the reachability analysis of a dynamic system (i.e., OS) considering the TTC and the ship maneuverability. Reachability analysis is a method to find a set of states which the system can reach under constraints. The reachability analysis has been applied in many fields: planning paths for robots [203]; detecting dangers for aircraft [204], verifying safety for autonomous cars [166], etc.

III.1 Problem statement

The main problem of the RV set construction is:

find all velocities which the OS can achieve, given an initial velocity ($\vec{v}_0$) *and time window* (τ).

An illustration of the RV set is shown in Figure III.2. The green ellipse is a RV set of OS, with respect to an initial velocity ($\vec{v}_0$) and a certain time window τ . All the velocities in this set are reachable for the OS, while the rest are not reachable in time τ . The point "A" represents the initial velocity, and the point "B" is an arbitrary velocity. The OS wants to modify its velocity from "A" to "B". Numerous controls can achieve this goal and let denote them as $U^{A\to B}$. $u_i \in U^{A\to B}$ is a control input which leads OS's velocity move from "A" to "B"; *traj_i* is the path how the velocity changes. t_{traj_i} is the time needed for the velocity move from "A" to "B" along *traj*$_i$. Within these controls, an optimal one (say u_k^{opt}) is found which minimizes the time:

$$t_k^{opt} = \min\left\{t_{traj_i}\left(u_i^{opt}\right) \mid u_i^{opt} \in U^{A\to B}\right\} \tag{III.1}$$

If $\tau \geq t_k^{opt}$, at least one control can complete this mission within τ , and "B" is contained in $RV\left(\vec{v}_0,\tau\right)$; if $\tau < t_k^{opt}$, no controls can change the velocity from "A" to "B" within τ, which means that "B" is out of $RV\left(\vec{v}_0,\tau\right)$; if $\tau = t_k^{opt}$, point "B" is located on the boundary of the RV set $RV\left(\vec{v}_0,\tau = t_k^{opt}\right)$.

Therefore, the main problem of RV set construction is equivalent to:

finding a minimum operation time from an initial velocity to all other velocities.

Thus, $RV\left(\vec{v}_0,\tau\right)$ is a collection of the velocity whose minimal operation time less than τ .

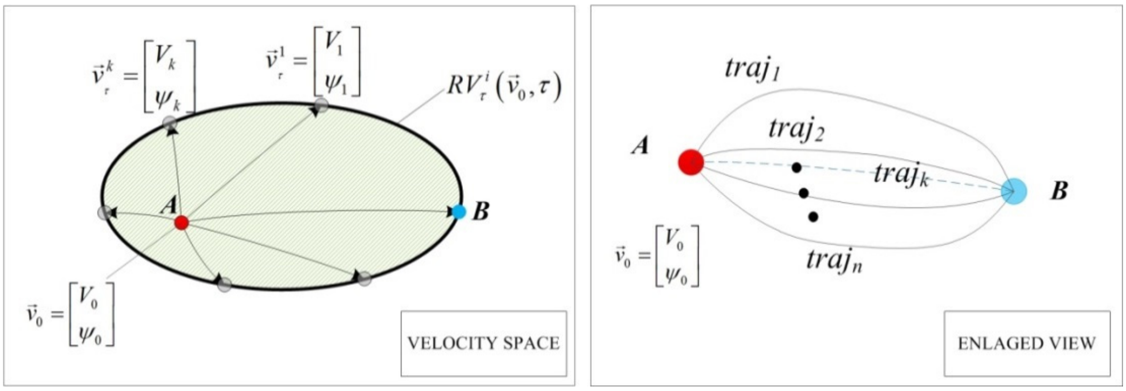


Figure III. 2 Illustration of a reachable velocity set of OS.

III.2 Steps of RV Set construction

To construct a RV set, firstly, we need to find the minimum operational time t^{opt} for each velocity in the velocity space. This velocity is on the boundary of the set $RV\left(\vec{v}_0, \tau = t_k^{opt}\right)$, which is also contained in the set $RV\left(\vec{v}_0, \tau \geq t_k^{opt}\right)$. The $RV\left(\vec{v}_0, \tau\right)$ is a collection of velocities whose $t_k^{opt} < \tau$. Therefore, the steps of RV set construction are as follows:

Step (1) Gridding: Grid the velocity space of OS, say: $\vec{v}_{alter.} : \mathbb{R}^m \times \mathbb{R}^n$ and $\vec{v}_{alter.} = \begin{bmatrix} v \in [0, v_{\max}] \\ \psi \in [-\pi, \pi] \end{bmatrix}$;

Step (2) Initialization: Initialize the OS's current velocity, say $\vec{v}_0 = \begin{bmatrix} V_0 \\ \psi_0 \end{bmatrix} = \begin{bmatrix} V_0 \\ 0 \end{bmatrix}$;

Step (3) Target velocity setting: Choose one vector in velocity space ($\vec{v}_{alter.}$) and set it as $\vec{v}_{\text{target}}$;

Step (4) Strategy searching: Find optimal control input u_k^{opt};

Step (5) Calculation: Calculate the time for the control t_k^{opt}, and store it in an m-by-n matrix T_{target};

Step (6) Loop: If $\vec{v}_{\text{target}}$ is not the last element in $\vec{v}_{alter.}$, go back to Step (3); otherwise, quit the loop.

Step (7) RV set determination: The RV set for any time window τ is a set of velocities whose operational time t_k^{opt} is not larger than τ.

In this method, only the operational time for the optimal control is needed; the RV sets for any time window τ are calculated at the same time. Given an initial velocity, it is unnecessary to recalculate the RV set for a different time window τ.

III.2.1 Ship motion model

Ship maneuvering ability is a significant factor that influences the size of a RV set. Generally, ship maneuvering ability refers to the turning ability, course-keeping ability, stopping ability, etc.[205]. In this article, we focus on the changes in the ship's course (turning ability) and speed (acceleration/deceleration ability). Moreover, we hold two assumptions on ship maneuvering ability:

Assumption (i): The speed-loss effect during the turning phase is neglected, i.e., the ship is capable of keeping its speed during turning;

Assumption (ii): The external disturbance from wind, current, etc., are excluded in this paper. Therefore, the ship's position can be predicted with its velocity only.

These assumptions aim at simplifying the ship motion model. The assumption (i) allows us to consider the acceleration process and turning process separately. The assumption (ii) is made to exclude the impacts of environmental disturbances on the ship's motion.

The calculation of operational time t_{traj_i} is strongly tied to the ship's motion model. Ship motion is related to acceleration, velocity, and yaw rate. According to [206], when the initial speed of the ship is larger than 0, the speed and the acceleration have a relationship similar to the quadratic relation. Moreover, the yaw rate has a positive and linear relationship with speed. Therefore, we can formulate our ship motion model by acceleration equation (Equation (III.2)) and turning equation (Equation (III.3)). These dynamic equations describe a simplified ship motion model in which the turning process and acceleration process are decoupled (Assumption (i)). Moreover, the ship can obtain a constant yaw rate when it decides to turn, which means this ship can turn in a circle whose radius is $v(t)/r(t)$.

$$a = \dot{v} = p_1 v^2 + p_2 v + p_3 \quad \text{(III.2)}$$

$$r = \dot{\psi} = k \cdot v \quad \text{(III.3)}$$

where a is the acceleration ([m/s^2]);

v is speed of the OS ([m/s]);

p_1, p_2, and p_3 are the ship coefficients;

r is the yaw rate ([°/s]);

ψ is the course of the OS ([°]);

k is the turning coefficient.

III.2.2 Control strategies

To reach the desired velocity, different control strategies should be applied. In this paper, we choose three control strategies:

Strategy 1. Changing the speed first and then turning the course;

Strategy 2. Turning first and then changing the speed;

Strategy 3. Changing the speed and turning at the same time.

Figure III.3 shows how these strategies can steer the velocity towards a target velocity in the velocity space. In this article, the strategy whose operation time is minimum is the optimal one. Denote T_1, T_2, and T_3 as the time for applying Strategy 1, 2 and 3. The calculation of T_1, T_2, and T_3 is explained in the following parts.

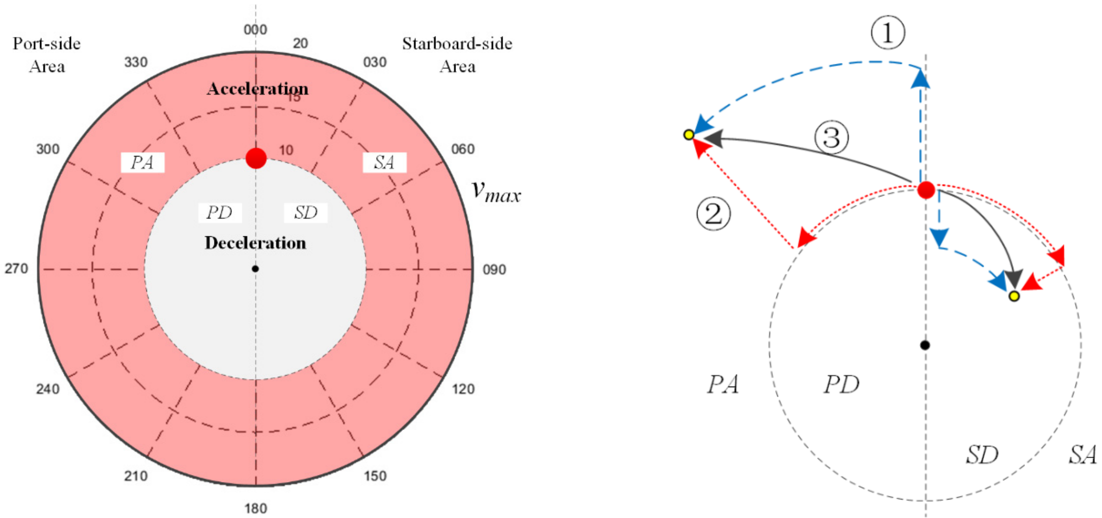


Figure III. 3 Illustration of different steering strategies in velocity space.

(notes: the red point is initial velocity: speed = 10 knots, course = 0 deg)

Given an initial velocity of the OS, the velocity space is divided into four parts (as shown in Figure III.3):

- Port-side Acceleration (PA): $\vec{v}_{alter.}$ in this part needs port-side turning and acceleration;
- Port-side Deceleration (PD): $\vec{v}_{alter.}$ in this part needs port-side turning and deceleration;
- Starboard-side Acceleration (SA): $\vec{v}_{alter.}$ in this part needs starboard-side turning and acceleration;
- Starboard-side Deceleration (SD): $\vec{v}_{alter.}$ in this part needs starboard-side turning and deceleration.

According to this deviation, two major situations can be identified. Giving an initial velocity is $\vec{v}_0 = \left[V_0, \psi_0\right]'$ and a target velocity is $\vec{v}_{\text{target}} = \left[V_1, \psi_1\right]'$,

- Acceleration: $\vec{v}_{\text{target}} = \vec{v}_{alter.} \in \left\{\mathrm{PA} \cup \mathrm{SA}\right\}$, $V_1 > V_0$;
- Deceleration: $\vec{v}_{\text{target}} = \vec{v}_{alter.} \in \left\{PD \cup SD\right\}$, $V_1 < V_0$.

For the general acceleration cases, the time for these strategies can be calculated as:

$$
\begin{aligned}
&T_1 = T_{\text{Accel1}} + T_{\text{Turn1}} = T_{\text{Accel}} + \frac{\Delta\psi}{kV_1};\\
&T_2 = T_{\text{Turn2}} + T_{\text{Accel2}} = \frac{\Delta\psi}{kV_0} + T_{\text{Accel}};\\
&T_3 = \begin{cases} T_{\text{Accel}} & , \quad \text{when } T_{\text{Accel}} \geq T_{\text{Turn}} \\ T_{\text{Accel}} + \dfrac{\Delta\psi - \Delta\psi_{\text{Accel}}}{kV_1}, & \quad \text{when } T_{\text{Accel}} < T_{\text{Turn}} \end{cases}.
\end{aligned}
\tag{III.4}
$$

where $\Delta\psi$ is the angular difference between ψ_0 and ψ_1;

$\Delta\psi_{Accel}$ is the turning course during the acceleration;

T_{Turn} is the time for the turning process, and it is related to the angular difference;

T_{Accel} is the time for accelerating, and it can be calculated with Equation (III.6).

$$T_{\text{Turn}} = \frac{\Delta\psi}{r} = \frac{\Delta\psi}{k \cdot V} \tag{III.5}$$

$$T_{\text{Accel}} = g\left(V_0, V_1\right) \tag{III.6}$$

According to Equation (III.5), turning with a lower speed needs a longer turning time. From Equation (III.4), we observe that the time for acceleration in each strategy is the same. Since, $V_1 > V_0$, we have $T_{\text{Turn1}} < T_{\text{Turn2}}$, and therefore, $T_1 < T_2$. In Equation (III.4), $\Delta\psi - \Delta\psi_{Accel}$ is the rest of course to the target course. Since $\Delta\psi - \Delta\psi_{Accel} < \Delta\psi$, it has $T_3 < T_2$. It implies that $T_3 < T_1 < T_2$. In other words, when the target speed V_1 is larger than V_0, Strategy 3 is the best strategy.

For the general deceleration cases, the operation time for the three strategies are

$$
\begin{aligned}
&T_1 = T_{\text{Decel1}} + T_{\text{Turn1}} = T_{\text{Decel}} + \frac{\Delta\psi}{kV_1};\\
&T_2 = T_{\text{Turn2}} + T_{\text{Decel2}} = \frac{\Delta\psi}{kV_0} + T_{\text{Decel}};\\
&T_3 = \begin{cases} T_{\text{Decel}} & , \quad \text{when } T_{\text{Decel}} \geq T_{\text{Turn}} \\ T_{\text{Decel}} + \dfrac{\Delta\psi - \Delta\psi_{\text{Decel}}}{kV_1}, & \quad \text{when } T_{\text{Decel}} < T_{\text{Turn}} \end{cases}.
\end{aligned}
\tag{III.7}
$$

where T_{Decel} is the time for decelerating, $T_{\text{Decel}} = h(V_0, V_1)$;

$\Delta\psi_{\text{Decel}}$ is the change of course during deceleration.

In the deceleration process, T_{Decel} is a common factor in each strategy. Since $V_1<V_0$, we have $T_{\text{Turn1}}>T_{\text{Turn2}}$, and therefore, $T_1>T_2$. In Strategy 3, the ship can reach the target course during deceleration (meaning $T_3 = T_{\text{Decel}}$), Strategy 3 would be the best strategy ($T_3<T_2<T_1$). If not, the extra time for steering the course to the desired course should be added and it can be formulated as: $\dfrac{\Delta\psi-\Delta\psi_{\text{Decel}}}{kV_1}$. In this case, $T_3<T_1$, since $\Delta\psi-\Delta\psi_{\text{Decel}}<\Delta\psi$. Therefore, the operational time in different strategies is sorted as $T_3<T_1$, $T_2<T_1$. That means, when the target speed V_1 is smaller than the initial speed V_0, Strategy 1 is not the best strategy.

In summary, to control the OS moving from an initial velocity to arbitrary target velocity, the optimal strategy is chosen as follows:

- In the acceleration case, applying Strategy 3 always can get the minimum time;
- In the deceleration case, the operation time by Strategy 2 and Strategy 3 should be calculated, and the smaller one is the optimal one.

III.2.3 Calculation of time for acceleration and turning

The calculation of T_{Decel}, T_{Accel}, and T_{Turn} are introduced underneath.

(1) Calculation of T_{Decel} and T_{Accel}

The time for acceleration or deceleration can be calculated by solving the ship motion model (Equation (III.2)) and finding the inverse function for the solution. Given the initial speed v_0, the solution of the differential Equation (III.2) is:

$$v(t)=\frac{1}{2p_1}\left\{\sqrt{D}\tan\left(\frac{\sqrt{D}}{2}(t+C)\right)-p_2\right\}, \tag{III.8}$$

where $D = 4p_1p_3 - p_2^2$;

C is a constant, $C=\dfrac{2}{\sqrt{D}}\text{artan}\left(\dfrac{2p_1v_0+p_2}{\sqrt{D}}\right)$.

The inverse function of (III.8) is the time for acceleration/deceleration:

$$T_{\text{accel/decel}}=\frac{2}{\sqrt{D}}\text{artan}\left(\frac{2p_1v_{\text{target}}+p_2}{\sqrt{D}}\right)-C, \tag{III.9}$$

where $v_{\text{target}}=\left\|\vec{v}_{\text{target}}\right\|$ is an alternative speed.

The Equation (III.9) can be used to calculate the time for acceleration and deceleration process using different groups of the coefficients (p_1,p_2,p_3) in Equation (III.9) for calculating acceleration and deceleration are different.

(2) Calculation of T_{Turn}

If the speed (say v^*) does not change during the turning process, e.g., Strategy 1 and 2, the time for turning is the inverse function of the solution of the differential Equation (III.3). The solution of Equation (III.3) is

$$\psi(t) = \int_0^t k \cdot v^* dt = k \cdot v^* \cdot t \,. \tag{III.10}$$

Therefore, the time for turning (T_{Turn}) is

$$T_{\text{Turn}} = \frac{\psi_{\text{target}} - \psi_0}{k \cdot v^*} = \frac{\psi_{\text{target}}}{k \cdot v^*} \,. \tag{III.11}$$

If the speed changes during a turning process, e.g., Strategy 3, we need another equation. Applying Equation (III.8) into Equation (III.3) and then we have: $\dot{\psi}(t) = k \cdot v(t)$. Theoretically, solving this equation, we have

$$\psi(t) = \frac{k}{2p_1}\left\{-2\left[\ln\cos\left(\frac{\sqrt{D}}{2}\cdot(t+C)\right) - \ln\cos\left(\frac{\sqrt{D}}{2}\cdot C\right)\right] - p_2 t\right\}. \tag{III.12}$$

where coefficients are dependent on the speed changing process, i.e., acceleration or deceleration.

To find the inverse function of Equation (III.12) is complicated. In this paper, a numerical method is applied: firstly, we simulate the turning and acceleration process; secondly, we record the evolution of course's response under the actions; then we make a regression of the time as a function of course. As a result, for each target course ψ , we can find a corresponding operation time for turning and acceleration. The operation time for turning and deceleration is also obtained in this way.

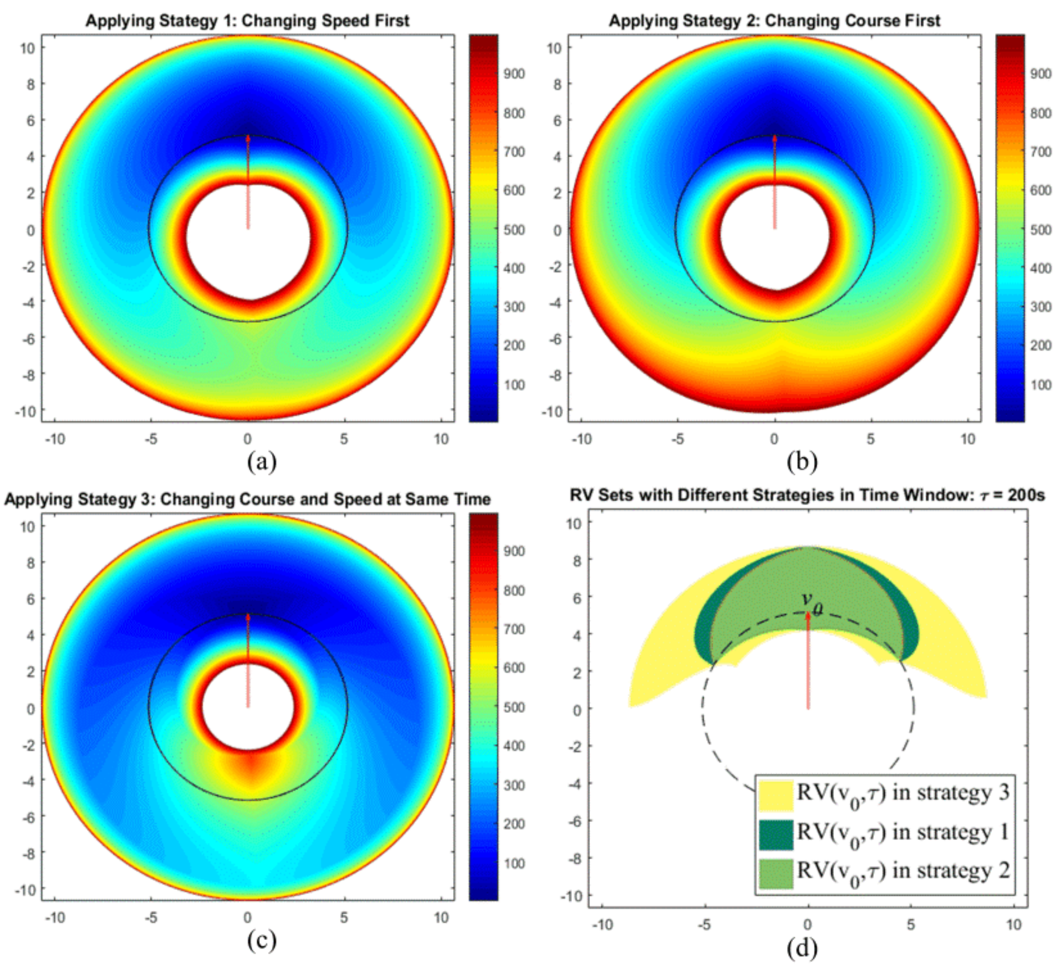


Figure III. 4 Illustration of reachable-velocity set under different steering strategies.
(operational time window: 0-1000 seconds; v_0 = 10 [knots])

For a better understanding of the constructions of the RV set with different strategies, we provide an example in Figure III.4. Figure III.4 (a) to (c) show the maps of operation time by applying different steering strategies (Strategy 1, 2, 3). Each point in the figure represents a potential target velocity ($\vec{v}_{\text{target}}$); the color reflects operation time (t_k) for steering the OS from $\vec{v}_0$ to $\vec{v}_{\text{target}}$. With v_0=10 [knots] and $\tau = 200$ [seconds], we can construct three sets of velocities for the three strategies, respectively, as shown in Figure III.4(d). The $RV(\vec{v}_0, \tau)$ with is the union of the three sets, i.e., the set for Strategy 3.

III.3 Time window of RV Set

Time To Collision (TTC) is the time left to avoid a collision. The calculation of the TTC is explained in Figure III.5 related to the scenario in which two ships encounter. Setting the TS as the reference, expanding the TS's scale by the OS's dimension (noted as Expanded Obstacle (EO)), the collision scenario turns out to be a particle which keeps moving with a relative velocity and colliding with a static EO at "Collision Point". Thus, the TTC can be formulated as:

$$TTC = \frac{D'_{ij}}{\left\|\vec{v}_{ji}\right\|}, \tag{III.13}$$

where $D'_{ij} = \left\|\vec{d}_{ij}\right\|\cos(\alpha) - \sqrt{R_{EO}{}^2 - D_{CPA}{}^2}$, $\vec{d}_{ij} = P_j - P_i$, $\cos(\alpha) = \frac{\vec{d}_{ij} \cdot \vec{v}_{ji}}{\left\|\vec{d}_{ij}\right\|\left\|\vec{v}_{ji}\right\|}$, and

$$D_{CPA} = \left\|\vec{d}_{ij}\right\|\sqrt{1 - \cos^2(\alpha)}.$$

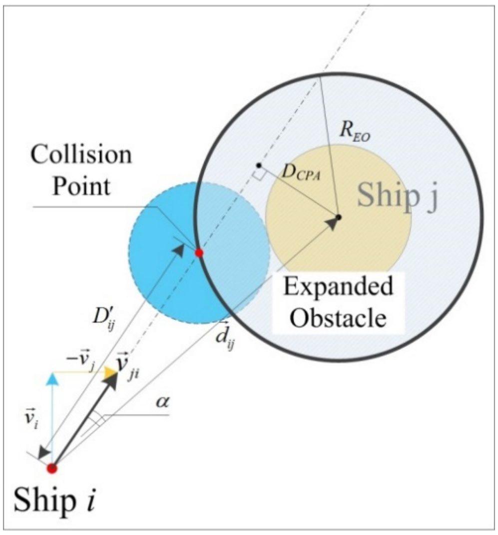


Figure III. 5 Illustration of TTC calculation.

References

1. WMU, *Transport 2040: Automation, Technology, Employment - The Future of Work*. 2019, World Maritime University (WMU): London, UK.
2. UNCTAD, *Review of Maritime Transport 2017*. 2018, United Nations: Geneva, Switzerland.
3. EMSA, *Annual Overview of Marine Casualties and Incidents 2018*. 2018, European Maritime Safety Agency: Lisboa, Portugal.
4. EMSA, *Summary Overview of Marine Casualties and Incidents: SUMMARY OVERVIEW 2011-2015*. 2017, European Maritime Safety Agency: Lisboa, Portugal.
5. Backer, P.D., *Report on the joint investigation of the collision between the LNG Carrier-mts AL ORAIQ and the mv FLINTERSTAR off the coast of Belgium on 6 October 2015 with the total loss of the mv FLINTERSTAR*. 2016, Staatssecretaris Bestrijding Sociale Fraude, Privacy en Noordzee: Belgium.
6. NAVY, U.S.A., *Report on the collision between USS FITZGERALD (DDG 62) and Motor Vessel ACX CRYSTAL*. 2017, Department of the Navy, the USA: Washington, DC.
7. NAVY, U.S.A., *Report on the Collision between USS JOHN S MCCAIN (DDG 56) and Motor Vessel ALNIC MC*. 2017, Department of the Navy, the USA: Washington, DC.
8. MSA, P.R.C., *Report on the Investigation of the Collision between M.T. SANCHI and M.V. CF CRYSTAL in East China Sea on 6 January 2018*. 2018, Maritime Safety Administration (MSA) of P.R. China.
9. Tam, C., R. Bucknall, and A. Greig, *Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters.* Journal of Navigation, 2009. **62**(3): p. 455-476.

10. Chauvin, C., S. Lardjane, G. Morel, J.P. Clostermann, and B. Langard, *Human and organisational factors in maritime accidents: analysis of collisions at sea using the HFACS.* Accid Anal Prev, 2013. **59**: p. 26-37.

11. Dobie, G., *Safety And Shipping Review 2018 - an annual review of trends and developments in shipping losses and safety*. 2019, Allianz Global Corporate & Specialty SE (AGCS).

12. Kaur, K. and G. Rampersad, *Trust in driverless cars: Investigating key factors influencing the adoption of driverless cars.* Journal of Engineering and Technology Management, 2018. **48**: p. 87-96.

13. Liu, Z.X., Y.M. Zhang, X. Yu, and C. Yuan, *Unmanned surface vehicles.: An overview of developments and challenges.* Annual Reviews in Control, 2016. **41**: p. 71-93.

14. Zhang, W., F. Goerlandt, P. Kujala, and Y. Wang, *An advanced method for detecting possible near miss ship collisions from AIS data.* Ocean Engineering, 2016. **124**: p. 141-156.

15. Eliopoulou, E. and A. Papanikolaou, *Casualty analysis of large tankers.* Journal of Marine Science and Technology, 2007. **12**(4): p. 240-250.

16. Eleftheria, E., P. Apostolos, and V. Markos, *Statistical analysis of ship accidents and review of safety level.* Safety Science, 2016. **85**: p. 282-292.

17. Schiaretti, M., L. Chen, and R.R. Negenborn, *Survey on Autonomous Surface Vessels: Part II - Categorization of 60 Prototypes and Future Applications*, in *Computational Logistics*. 2017. p. 234-252.

18. Lopez-Santander, A. and J. Lawry, *An Ordinal Model of Risk Based on Mariner's Judgement.* Journal of Navigation, 2016. **70**(02): p. 309-324.

19. Tu, E.M., G.H. Zhang, L. Rachmawati, E. Rajabally, and G.B. Huang, *Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology.* Ieee Transactions on Intelligent Transportation Systems, 2018. **19**(5): p. 1559-1582.

20. Campbell, S., W. Naeem, and G.W. Irwin, *A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres.* Annual Reviews in Control, 2012. **36**(2): p. 267-283.

21. Polvara, R., S. Sharma, J. Wan, A. Manning, and R. Sutton, *Obstacle Avoidance Approaches for Autonomous Navigation of Unmanned Surface Vehicles.* Journal of Navigation, 2017. **71**(01): p. 241-256.

22. Liu, C., Q. Mao, X. Chu, and S. Xie, *An Improved A-Star Algorithm Considering Water Current, Traffic Separation and Berthing for Vessel Path Planning.* Applied Sciences, 2019. **9**(6).

23. Greytak, M. and F. Hover, *Motion Planning with an Analytic Risk Cost for Holonomic Vehicles.* Proceedings of the 48th Ieee Conference on Decision and Control, 2009 Held Jointly with the 2009 28th Chinese Control Conference (Cdc/Ccc 2009), 2009: p. 5655-5660.

24. Wang, X., Z.J. Liu, and Y. Cai, *The ship maneuverability based collision avoidance dynamic support system in close-quarters situation.* Ocean Engineering, 2017. **146**: p. 486-497.

25. Eriksen, B.H., E.F. Wilthil, A.L. Flåten, E.F. Brekke, and M. Breivik. *Radar-based maritime collision avoidance using dynamic window*. in *2018 IEEE Aerospace Conference*. 2018.

26. Kuchar, J.K. and L.C. Yang, *A Review of Conflict Detection and Resolution Modeling Methods.* IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,, 2000. **1**(4): p. 179-189.

27. IMO, *Resolution MSC.64(67) Adoption of New and Amended Performance Standards*, in *Adoption of new and amended performance standards*. 1996, IMO: London.

28. Fossen, T.I., *Part II: motion control*, in *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011, John Wiley & Sons Ltd.: United Kingdom.

29. Benjamin, M.R., J.J. Leonard, J.A. Curcio, and P.M. Newman, *A method for protocol-based collision avoidance between autonomous marine surface craft.* Journal of Field Robotics, 2006. **23**(5): p. 333-346.

30. Degre, T. and X. Lefevre, *A Collision Avoidance System.* Journal of Navigation, 1981. **34**(2): p. 294-302.

31. Lenart, A.S., *Collision Threat Parameters for a New Radar Display and Plot Technique.* Journal of Navigation, 1983. **36**(3): p. 404-410.

32. Pedersen, E., K. Inoue, and M. Tsugane, *Simulator studies on a collision avoidance display that facilitates efficient and precise assessment of evasive manoeuvres in congested waterways.* Journal of Navigation, 2003. **56**(3): p. 411-427.

33. Szlapczynski, R. and J. Szlapczynska, *A Target Information Display for Visualising Collision Avoidance Manoeuvres in Various Visibility Conditions.* Journal of Navigation, 2015. **68**(6): p. 1041-1055.

34. Szlapczynski, R. and J. Szlapczynska, *A method of determining and visualizing safe motion parameters of a ship navigating in restricted waters.* Ocean Engineering, 2017. **129**: p. 363-373.

35. Lazarowska, A., *A new deterministic approach in a decision support system for ship's trajectory planning.* Expert Systems with Applications, 2017. **71**: p. 469-478.

36. Hvamb, K., *Motion Planning Algorithms for Marine Vehicles*, in *Department of Engineering Cybernetics*. 2015, NTNU: Throndheim.

37. Vincent, T.L., *Collision avoidance at sea*, in *Differential Games and Applications*, P. Hagedorn, H.W. Knobloch, and G.J. Olsder, Editors. 1977, Springer Berlin Heidelberg. p. 205-221.

38. Fossen, S., *Visualization of Ships in a Mixed-Reality Environment and Automated Situational Awareness using Live AIS Data*, in *Faculty of Information Technology and Electrical Engineering Department of ICT and Natural Sciences*. 2018, Norwegian University of Science and Technology.

39. Moe, S. and K.Y. Pettersen, *Set-Based Line-of-Sight (LOS) Path Following with Collision Avoidance for Underactuated Unmanned Surface Vessel*, in *24th Mediterranean Conference on Control and Automation* 2016, IEEE: Athens, Greece.

40. Abdelaal, M., M. Franzle, and A. Hahn, *Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances.* Ocean Engineering, 2018. **160**: p. 168-180.

41. Wiig, M.S., K.Y. Pettersen, and T.R. Krogstad, *A reactive collision avoidance algorithm for vehicles with underactuated dynamics*. 2017. p. 1452-1459.

42. Soltan, R.A., H. Ashrafiuon, and K.R. Muske, *ODE-based obstacle avoidance and trajectory planning for unmanned surface vessels.* Robotica, 2010. **29**(05): p. 691-703.

43. Eriksen, B.H., M. Breivik, K.Y. Pettersen, and M.S. Wiig. *A modified dynamic window algorithm for horizontal collision avoidance for AUVs*. in *2016 IEEE Conference on Control Applications (CCA)*. 2016.

44. Xue, Y., B.S. Lee, and D. Han, *Automatic collision avoidance of ships.* Proceedings of the Institution of Mechanical Engineers Part M-Journal of Engineering for the Maritime Environment, 2009. **223**(M1): p. 33-46.

45. He, Y.X., Y. Jin, L.W. Huang, Y. Xiong, P.F. Chen, and J.M. Mou, *Quantitative analysis of COLREG rules and seamanship for autonomous collision avoidance at open sea.* Ocean Engineering, 2017. **140**: p. 281-291.

46. Li, S.J., J.L. Liu, and R.R. Negenborn, *Distributed coordination for collision avoidance of multiple ships considering ship maneuverability.* Ocean Engineering, 2019. **181**: p. 212-226.

47. Zhang, J.F., D. Zhang, X.P. Yan, S. Haugen, and C.G. Soares, *A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs.* Ocean Engineering, 2015. **105**: p. 336-348.

48. Fang, M.-C., K.-Y. Tsai, and C.-C. Fang, *A Simplified Simulation Model of Ship Navigation for Safety and Collision Avoidance in Heavy Traffic Areas.* Journal of Navigation, 2017. **71**(04): p. 837-860.

49. Chen, L., H. Hopman, and R.R. Negenborn, *Distributed model predictive control for vessel train formations of cooperative multi-vessel systems.* Transportation Research Part C-Emerging Technologies, 2018. **92**: p. 101-118.

50. Zheng, H., *Coordination of Waterborne AGVs*. 2016, Delft University of Technology: Delft.

51. Szlapczynski, R., P. Krata, and J. Szlapczynska, *Ship domain applied to determining distances for collision avoidance manoeuvres in give-way situations.* Ocean Engineering, 2018. **165**: p. 43-54.

52. Miloh, T. and M. Pachter, *Ship Collision Avoidance and Pursuit Evasion Differential-Games with Speed-Loss in a Turn.* Computers & Mathematics with Applications, 1989. **18**(1-3): p. 77-100.

53. Li, X.R. and V.P. Jilkov, *Survey of maneuvering target tracking. Part I: Dynamic models.* Ieee Transactions on Aerospace and Electronic Systems, 2003. **39**(4): p. 1333-1364.

54. Fossen, T.I., *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles.* 2002, Trondheim, Norway: Marine Cybernetics.

55. Yasukawa, H. and Y. Yoshimura, *Introduction of MMG standard method for ship maneuvering predictions.* Journal of Marine Science and Technology, 2015. **20**(1): p. 37-52.

56. Huang, Y., L. Chen, and P.H.A.J.M. van Gelder, *Generalized velocity obstacle algorithm for preventing ship collisions at sea.* Ocean Engineering, 2019. **173**: p. 142-156.

57. Verhaegen, M. and V. Verdult, *Filtering and system identification: a least squares approach*. 2007: Cambridge University Press.

58. Candeloro, M., A.M. Lekkas, and A.J. Sørensen, *A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels.* Control Engineering Practice, 2017. **61**: p. 41-54.

59. Shah, B.C., P. Švec, I.R. Bertaska, A.J. Sinisterra, W. Klinger, K. von Ellenrieder, M. Dhanak, and S.K. Gupta, *Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic.* Autonomous Robots, 2015. **40**(7): p. 1139-1163.

60. Lefèvre, S., D. Vasquez, and C. Laugier, *A survey on motion prediction and risk assessment for intelligent vehicles.* ROBOMECH Journal, 2014. **1**(1): p. 1-14.

61. Scheepens, R., H. van de Wetering, and J.J. van Wijk, *Contour based visualization of vessel movement predictions.* International Journal of Geographical Information Science, 2014. **28**(5): p. 891-909.

62. Simsir, U., M.F. Amasyalı, M. Bal, U.B. Çelebi, and S. Ertugrul, *Decision support system for collision avoidance of vessels.* Applied Soft Computing, 2014. **25**: p. 369-378.

63. Rong, H., A.P. Teixeira, and C. Guedes Soares, *Ship trajectory uncertainty prediction based on a Gaussian Process model.* Ocean Engineering, 2019. **182**: p. 499-511.

64. Peel, D. and N.M. Good, *A hidden Markov model approach for determining vessel activity from vessel monitoring system data.* Canadian Journal of Fisheries and Aquatic Sciences, 2011. **68**(7): p. 1252-1264.

65. Kim, D., K. Hirayama, and T. Okimoto, *Distributed Stochastic Search Algorithm for Multi-ship Encounter Situations.* Journal of Navigation, 2017. **70**(4): p. 699-718.

66. Kim, D.G., K. Hirayama, and G.K. Park, *Collision Avoidance in Multiple-Ship Situations by Distributed Local Search.* Journal of Advanced Computational Intelligence and Intelligent Informatics, 2014. **18**(5): p. 839-848.

67. Hu, Q., C. Yang, H. Chen, and B. Xiao, *Planned Route Based Negotiation for Collision Avoidance Between Vessels.* International Journal on Marine Navigation and Safety of Sea Transportation, 2008. **2**(4).

68. Zheng, H.R., R.R. Negenborn, and G. Lodewijks, *Fast ADMM for Distributed Model Predictive Control of Cooperative Waterborne AGVs.* IEEE Transactions on Control Systems Technology, 2017. **25**(4): p. 1406-1413.

69. L. Chen, Y. Huang, H. Zheng, J.J. Hopman, and R.R. Negenborn, *Cooperative multi-vessel systems in urban waterway networks.* IEEE Transactions on Intelligent Transportation Systems, 2019.

70. Cho, Y., J. Han, and J. Kim, *Intent inference of ship maneuvering for automatic ship collision avoidance.* Ifac Papersonline, 2018. **51**(29): p. 384-388.

71. Goerlandt, F., J. Montewka, V. Kuzmin, and P. Kujala, *A risk-informed ship collision alert system: Framework and application.* Safety Science, 2015. **77**: p. 182-204.

72. Kao, S.-L., K.-T. Lee, K.-Y. Chang, and M.-D. Ko, *A Fuzzy Logic Method for Collision Avoidance in Vessel Traffic Service.* Journal of Navigation, 2006. **60**(1): p. 17-31.

73. Kuwata, Y., M.T. Wolf, D. Zarzhitsky, and T.L. Huntsberger, *Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles.* Ieee Journal of Oceanic Engineering, 2014. **39**(1): p. 110-119.

74. Johansen, T.A., T. Perez, and A. Cristofaro, *Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment.* Ieee Transactions on Intelligent Transportation Systems, 2016. **17**(12): p. 3407-3422.

75. Kearon, J., *Computer programs for collision avoidance and track keeping*, in *Mathematical Aspects of Marine Traffic*, S.H. Hollingdale, Editor. 1979, Academic Press INC. LTD.: London, UK.

76. Lee, H.J. and K.P. Rhee, *Development of collision avoidance system by using expert system and search algorithm.* Int. Shipbuild. Progr.,, 2001. **48**(3): p. 197-212.

77. Chin, H.C. and A.K. Debnath, *Modeling perceived collision risk in port water navigation.* Safety Science, 2009. **47**(10): p. 1410-1416.

78. Li, B. and F.-W. Pang, *An approach of vessel collision risk assessment based on the D–S evidence theory.* Ocean Engineering, 2013. **74**: p. 16-21.

79. Zhao, Y.X., W. Li, and P. Shi, *A real-time collision avoidance learning system for Unmanned Surface Vessels.* Neurocomputing, 2016. **182**: p. 255-266.

80. Bukhari, A.C., I. Tusseyeva, B.-G. lee, and Y.-G. Kim, *An intelligent real-time multi-vessel collision risk assessment system from VTS view point based on fuzzy inference system.* Expert Systems with Applications, 2013. **40**(4): p. 1220-1230.

81. Gang, L.H., Y.H. Wang, Y. Sun, L.P. Zhou, and M.G. Zhang, *Estimation of vessel collision risk index based on support vector machine.* Advances in Mechanical Engineering, 2016. **8**(11): p. 168781401667125.

82. Ahn, J.H., K.P. Rhee, and Y.J. You, *A study on the collision avoidance of a ship using neural networks and fuzzy logic.* Applied Ocean Research, 2012. **37**: p. 162-173.

83. Baldauf, M., K. Benedict, S. Fischer, F. Motz, and J.U. Schroder-Hinrichs, *Collision avoidance systems in air and maritime traffic.* Proceedings of the Institution of Mechanical Engineers Part O-Journal of Risk and Reliability, 2011. **225**(O3): p. 333-343.

84. Lisowski, J., *Game Control of Moving Objects.* IFAC Proceedings Volumes, 2002. **35**(1): p. 373-378.

85. Szlapczynski, R., *A unified measure of collision risk derived from the concept of a ship domain.* Journal of Navigation, 2006. **59**(3): p. 477-490.

86. Ren, Y., J. Mou, Q. Yan, and F. Zhang. *Study on assessing dynamic risk of ship collision*. in *ICTIS 2011: Multimodal Approach to Sustained Transportation System Development - Information, Technology, Implementation - Proceedings of the 1st Int. Conf. on Transportation Information and Safety*. 2011.

87. Perera, L.P. and C.G. Soares, *Collision risk detection and quantification in ship navigation with integrated bridge systems.* Ocean Engineering, 2015. **109**: p. 344-354.

88. Wen, Y.Q., Y.M. Huang, C.H. Zhou, J.L. Yang, C.S. Xiao, and X.C. Wu, *Modelling of marine traffic flow complexity.* Ocean Engineering, 2015. **104**: p. 500-510.

89. Zhang, W., J. Montewka, and F. Goerlandt, *Semi-qualitative method for ship collision risk assessment*, in *Safety and Reliability; Methodology and Applications*, Nowakowski and e. al., Editors. 2015, Taylor and Francis Group: London. p. 1563 - 1572.

90. Zhang, W.B., F. Goerlandt, J. Montewka, and P. Kujala, *A method for detecting possible near miss ship collisions from AIS data.* Ocean Engineering, 2015. **107**: p. 60-69.

91. Hilgert, H. and M. Baldauf, *A Common Risk Model for the Assessment of Encounter Situations on Board Ships* German Journal of Hydrography, 1997. **49**(4): p. 531-542.

92. Ożoga, B. and J. Montewka, *Towards a decision support system for maritime navigation on heavily trafficked basins.* Ocean Engineering, 2018. **159**: p. 88-97.

93. Perera, L.P., J.P. Carvalho, and C. Guedes Soares, *Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions.* Journal of Marine Science and Technology, 2011. **16**(1): p. 84-99.

94. Fujii, Y. and K. Tanaka, *Traffic Capacity.* Journal of the Institute of Navigation, 1971. **24**(4): p. 543-&.

95. Szlapczynski, R. and J. Szlapczynska, *Review of ship safety domains: Models and applications.* Ocean Engineering, 2017. **145**: p. 277-289.

96. Davis, P.V., M.J. Dove, and C.T. Stockel, *A Computer-Simulation of Marine Traffic Using Domains and Arenas.* Journal of Navigation, 1980. **33**(2): p. 215-222.

97. Zhao, J.S., Z.L. Wu, and F.C. Wang, *Comments on Ship Domains.* Journal of Navigation, 1993. **46**(3): p. 422-436.

98. Wang, N., *An Intelligent Spatial Collision Risk Based on the Quaternion Ship Domain.* Journal of Navigation, 2010. **63**(4): p. 733-749.

99. Colley, B.A., R.G. Curtis, and C.T. Stockel, *Maneuvering Times, Domains and Arenas.* Journal of Navigation, 1983. **36**(2): p. 324-328.

100. Zhao, J., M. Tan, W.G. Price, and P.A. Wilson, *Dcpa Simulation Model for Automatic Collision Avoidance Decision Making Systems Using Fuzzy Sets.* Oceans 94 - Oceans Engineering for Today's Technology and Tomorrow's Preservation, Proceedings, Vol Ii, 1994: p. B244-B249.

101. Tam, C. and R. Bucknall, *Collision risk assessment for ships.* Journal of Marine Science and Technology, 2010. **15**(3): p. 257-270.

102. Kijima, K. and Y. Furukawa, *Automatic collision avoidance system using the concept of blocking area.* IFAC Proceedings Volumes, 2003. **36**(21): p. 223-228.

103. Park, J. and J. Kim, *Predictive Evaluation of Ship Collision Risk Using the Concept of Probability Flow.* IEEE Journal of Oceanic Engineering, 2016(99): p. 1-10.

104. Huang, Y., P.H.A.J.M. van Gelder, and Y. Wen, *Velocity obstacle algorithms for collision prevention at sea.* Ocean Engineering, 2018. **151**: p. 308-321.

105. Zhao-lin, W., *Analysis of Radar PAD Information and a Suggestion to Reshape the PAD.* Journal of Navigation, 1988. **41**(01): p. 124-129.

106. Gerhart, G.R., J. Larson, C.M. Shoemaker, M. Bruch, J. Ebken, and D.W. Gage, *Autonomous navigation and obstacle avoidance for unmanned surface vehicles.* 2006. **6230**: p. 623007.

107. Fukuto, J. and H. Imazu, *New collision alarm algorithm using obstacle zone by target (OZT).* IFAC Proceedings Volumes, 2013. **46**(33): p. 91-96.

108. Kayano, J. and K. Kumagai, *Effectiveness of the OZT taking into account with the Other Ships' Waypoints Information*, in *9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*. 2017, IEEE: Otsu, Japan.

109. Su, C.M., K.Y. Chang, and C.Y. Cheng, *Fuzzy Decision on Optimal Collision Avoidance Measures for Ships in Vessel Traffic Service.* Journal of Marine Science and Technology-Taiwan, 2012. **20**(1): p. 38-48.

110. Krata, P. and J. Montewka, *Assessment of a critical area for a give-way ship in a collision encounter.* Archives of Transport, 2015. **34**(2): p. 51-60.

111. Baldauf, M., R. Mehdi, S. Fischer, and M. Gluch, *A perfect warning to avoid collisions at sea?* Scientific Journals of the Maritime University of Szczecin, 2017. **49**(121): p. 53-64.

112. Szlapczynski, R., P. Krata, and J. Szlapczynska, *A Ship Domain-Based Method of Determining Action Distances for Evasive Manoeuvres in Stand-On Situations* Journal of Advanced Transportation, 2018. **2018**: p. 1-19.

113. Woerner, K., M.R. Benjamin, M. Novitzky, and J.J. Leonard, *Quantifying protocol evaluation for autonomous collision avoidance.* Autonomous Robots, 2018. **43**(4): p. 967-991.

114. Ozturk, U. and K. Cicek, *Individual collision risk assessment in ship navigation: A systematic literature review.* Ocean Engineering, 2019. **180**: p. 130-143.

115. Motz, F., S. Hockel, M. Baldauf, and K. Benedict, *Development of a Concept for Bridge Alert Management.* Transnav-International Journal on Marine Navigation and Safety of Sea Transportation, 2009. **3**(1): p. 61-66.

116. Mitici, M. and H.A.P. Blom, *Mathematical Models for Air Traffic Conflict and Collision Probability Estimation.* IEEE Transactions on Intelligent Transportation Systems, 2018.

117. Siegwart, R., I.R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. second edition ed. 2011, Cambridge, Massachusetts, London, England: The MIT Press.

118. Naeem, W., G.W. Irwin, and A.L. Yang, *COLREGs-based collision avoidance strategies for unmanned surface vehicles.* Mechatronics, 2012. **22**(6): p. 669-678.

119. Tam, C. and R. Bucknall, *Cooperative path planning algorithm for marine surface vessels.* Ocean Engineering, 2013. **57**: p. 25-33.

120. Praczyk, T., *Neural anti-collision system for Autonomous Surface Vehicle.* Neurocomputing, 2015. **149**: p. 559-572.

121. Perera, L.P., J.P. Carvalho, and C.G. Soares, *Intelligent Ocean Navigation and Fuzzy-Bayesian Decision/Action Formulation.* Ieee Journal of Oceanic Engineering, 2012. **37**(2): p. 204-219.

122. Khatib, O., *Real-time obstacle avoidance for manipulators and mobile robots*, in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. 1985, IEEE: St. Louis, MO, USA. p. 500-505.

123. Borenstein, J. and Y. Koren, *Real‐time obstacle avoidance for fast mobile robots.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 1989. **19**(5): p. 1179 - 1187.

124. Ge, S.S. and Y.J. Cui, *Dynamic motion planning for mobile robots using potential field method.* Autonomous Robots, 2002. **13**(3): p. 207-222.

125. Lyu, H.G. and Y. Yin, *Ship's Trajectory Planning for Collision Avoidance at Sea Based on Modified Artificial Potential Field.* 2017 2nd International Conference on Robotics and Automation Engineering (Icrae), 2017: p. 351-357.

126. Lyu, H. and Y. Yin, *COLREGS-Constrained Real-time Path Planning for Autonomous Ships Using Modified Artificial Potential Fields.* Journal of Navigation, 2018. **72**(3): p. 588-608.

127. Zeng, L.Q. and G.M. Bone, *Mobile Robot Collision Avoidance in Human Environments.* International Journal of Advanced Robotic Systems, 2013. **10**(1): p. 1-14.

128. Soltan, R.A., H. Ashrafiuon, and K.R. Muske, *Trajectory Real-Time Obstacle Avoidance for Underactuated Unmanned Surface Vessels.* 2009: p. 1059-1067.

129. Mahini, F., L. DiWilliams, K. Burke, and H. Ashrafiuon, *An experimental setup for autonomous operation of surface vessels in rough seas.* Robotica, 2013. **31**(05): p. 703-715.

130. Szlapczynski, R. and P. Krata, *Determining and visualizing safe motion parameters of a ship navigating in severe weather conditions.* Ocean Engineering, 2018. **158**: p. 263-274.

131. Szlapczynski, R., *Planning Emergency Manoeuvres.* Journal of Navigation, 2008. **62**(01): p. 79.

132. Loe, Ø.A.G., *Collision Avoidance for Unmanned Surface Vehicles*, in *Department of Engineering Cybernetics*. 2008, Norwegian University of Science and Technology.

133. Serigstad, E., *Hybrid Collision Avoidance for Autonomous Surface Vessels*. 2017, NTNU.

134. Li, S., J. Liu, X. Cao, and Y. Zhang, *A Novel Method for Solving Collision Avoidance Problem in Multiple Ships Encounter Situations.* 2018. **11184**: p. 47-66.

135. Švec, P., A. Thakur, E. Raboin, B.C. Shah, and S.K. Gupta, *Target following with motion prediction for unmanned surface vehicle operating in cluttered environments.* Autonomous Robots, 2013. **36**(4): p. 383-405.

136. Benjamin, M.R., *Autonomous COLREGS Modes and Velocity Functions*. 2017, Mechanical Engineering, MIT: MA, USA.

137. Huang, Y., *Study on marine traffic flow complexity metric*, in *School of Navigation*. 2014, Wuhan University of Technology: Wuhan, CN.

138. Fox, D., W. Burgard, and S. Thrun, *The dynamic window approach to collision avoidance.* Ieee Robotics & Automation Magazine, 1997. **4**(1): p. 23-33.

139. Martinez-Gomez, L., *Safe Navigation for Autonomous Vehicles in Dynamic Environments: an Inevitable Collision State (ICS) Perspective*. 2010, Université de Grenoble: France.

140. Brock, O. and O. Khatib, *High-speed navigation using the global dynamic window approach.* Icra '99: Ieee International Conference on Robotics and Automation, Vols 1-4, Proceedings, 1999: p. 341-346.

141. Seder, M. and I. Petrovic, *Dynamic window based approach to mobile robot motion control in the presence of moving obstacles.* Proceedings of the 2007 Ieee International Conference on Robotics and Automation, Vols 1-10, 2007: p. 1986-+.

142. Bertaska, I.R., B. Shah, K. von Ellenrieder, P. Svec, W. Klinger, A.J. Sinisterra, M. Dhanak, and S.K. Gupta, *Experimental evaluation of automatically-generated behaviors for USV operations.* Ocean Engineering, 2015. **106**: p. 496-514.

143. Eriksen, B.-O.H., M. Breivik, E.F. Wilthil, A.L. Flåten, and E.F. Brekke, *The Branching-Course MPC Algorithm for Maritime Collision Avoidance.* Journal of Field Robotics, submitted.

144. Sun, X.J., G.F. Wang, Y.S. Fan, D.D. Mu, and B.B. Qiu, *Collision Avoidance of Podded Propulsion Unmanned Surface Vehicle With COLREGs Compliance and Its Modeling and Identification.* Ieee Access, 2018. **6**: p. 55473-55491.

145. Pivtoraiko, M., R.A. Knepper, and A. Kelly, *Differentially Constrained Mobile Robot Motion Planning in State Lattices.* Journal of Field Robotics, 2009. **26**(3): p. 308-333.

146. Kushleyev, A. and M. Likhachev, *Time-bounded Lattice for Efficient Planning in Dynamic Environments.* Icra: 2009 Ieee International Conference on Robotics and Automation, Vols 1-7, 2009: p. 4303-4309.

147. Fiorini, P. and Z. Shiller, *Motion planning in dynamic environments using velocity obstacles.* International Journal of Robotics Research, 1998. **17**(7): p. 760-772.

148. Alonso-Mora, J., P. Beardsley, and R. Siegwart, *Cooperative Collision Avoidance for Nonholonomic Robots.* Ieee Transactions on Robotics, 2018. **34**(2): p. 404-420.

149. Zhuang, J.Y., L. Zhang, S.Q. Zhao, J. Cao, B. Wang, and H.B. Sun, *Radar-based collision avoidance for unmanned surface vehicles.* China Ocean Engineering, 2016. **30**(6): p. 867-883.

150. Chakravarthy, A. and D. Ghose, *Obstacle avoidance in a dynamic environment: acollision cone approach.* IEEE Transactions on Systems, Man, and Cybernetics, 1998. **28**: p. 562–574.

151. Savkin, A.V. and C. Wang, *A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles.* Robotica, 2013. **31**(06): p. 993-1001.

152. Xue, Y., D. Clelland, B.S. Lee, and D. Han, *Automatic simulation of ship navigation.* Ocean Engineering, 2011. **38**(17-18): p. 2290-2305.

153. Fan, Y., X. Sun, and G. Wang, *An autonomous dynamic collision avoidance control method for unmanned surface vehicle in unknown ocean environment.* International Journal of Advanced Robotic Systems, 2019. **16**(2).

154. Wiig, M.S., K.Y. Pettersen, and A.V. Savkin, *A Reactive Collision Avoidance Algorithm for Nonholonomic Vehicles.* 2017 Ieee Conference on Control Technology and Applications (Ccta 2017), 2017: p. 1776-1783.

155. Ferranti, L., R.R. Negenborn, T. Keviczky, and J. Alonso-Mora, *Coordination of Multiple Vessels Via Distributed Nonlinear Model Predictive Control*, in *17th European Control Conference (ECC 2018)*. 2018: Limassol, Cyprus. p. 2523-2528.

156. Hu, L., W. Naeem, E. Rajabally, G. Watson, T. Mills, Z. Bhuiyan, and I. Salter, *COLREGs-Compliant Path Planning for Autonomous Surface Vehicles: A Multiobjective Optimization Approach.* Ifac Papersonline, 2017. **50**(1): p. 13662-13667.

157. Liu, Y.C., W.W. Liu, R. Song, and R. Bucknall, *Predictive navigation of unmanned surface vehicles in a dynamic maritime environment when using the fast marching method.* International Journal of Adaptive Control and Signal Processing, 2017. **31**(4): p. 464-488.

158. Smierzchalski, R. and Z. Michalewicz, *Modeling of ship trajectory in collision situations by an evolutionary algorithm.* Ieee Transactions on Evolutionary Computation, 2000. **4**(3): p. 227-241.

159. Szlapczynski, R., *Evolutionary Sets Of Safe Ship Trajectories: A New Approach To Collision Avoidance.* Journal of Navigation, 2011. **64**(1): p. 169-181.

160. Tsou, M.C. and C.K. Hsueh, *The Study of Ship Collision Avoidance Route Planning by Ant Colony Algorithm.* Journal of Marine Science and Technology-Taiwan, 2010. **18**(5): p. 746-756.

161. Tsou, M.C., *Multi-target collision avoidance route planning under an ECDIS framework.* Ocean Engineering, 2016. **121**: p. 268-278.

162. Kang, Y.T., W.J. Chen, D.Q. Zhu, J.H. Wang, and Q.M. Xie, *Collision Avoidance Path Planning for Ships by Particle Swarm Optimization.* Journal of Marine Science and Technology-Taiwan, 2018. **26**(6): p. 777-786.

163. Song, A.L.F., B.Y.R. Su, C.Z.P. Dong, D.W. Shen, E.Z.Q. Xiang, and F.P.X. Mao, *A two-level dynamic obstacle avoidance algorithm for unmanned surface vehicles.* Ocean Engineering, 2018. **170**: p. 351-360.

164. Graziano, A., A.P. Teixeira, and C. Guedes Soares, *Classification of human errors in grounding and collision accidents using the TRACEr taxonomy.* Safety Science, 2016. **86**: p. 245-257.

165. Schwarting, W., J. Alonso-Mora, and D. Rus, *Planning and Decision-Making for Autonomous Vehicles.* Annual Review of Control, Robotics, and Autonomous Systems, 2018. **1**(1): p. 187-210.

166. Althof, M., *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. 2010, Technische Universität München: Germany.

167. Majumdar, A. and R. Tedrake, *Funnel libraries for real-time robust feedback motion planning.* International Journal of Robotics Research, 2017. **36**(8): p. 947-982.

168. UK, M., *BEING A RESPONSIBLE INDUSTRY-An Industry Code of Practice*. 2017, Society of Maritime Industries on behalf of Maritime UK.

169. Sheridan, T.B. and R. Parasuraman, *Human-Automation Interaction.* Reviews of Human Factors and Ergonomics, 2005. **1**(1): p. 89-128.

170. IMO, *International convention for the safety of life at sea*. 1974.

171. Fagnant, D.J. and K. Kockelman, *Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations.* Transportation Research Part A: Policy and Practice, 2015. **77**: p. 167-181.

172. Lee, J.D. and K.A. See, *Trust in automation: designing for appropriate reliance.* Hum Factors, 2004. **46**(1): p. 50-80.

173. Kim, H., D. Kim, J.-U. Shin, H. Kim, and H. Myung, *Angular rate-constrained path planning algorithm for unmanned surface vehicles.* Ocean Engineering, 2014. **84**: p. 37-44.

174. Liu, Y. and R. Bucknall, *The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method.* Applied Ocean Research, 2016. **59**: p. 327-344.

175. Szlapczynski, R. and J. Szlapczynska, *An analysis of domain-based ship collision risk parameters.* Ocean Engineering, 2016. **126**: p. 47-56.

176. Pietrzykowski, Z. and J. Uriasz, *The Ship Domain – A Criterion of Navigational Safety Assessment in an Open Sea Area.* Journal of Navigation, 2008. **62**(01): p. 93.

177. Jenie, Y.I., E.-J. van Kampen, C.C. de Visser, J. Ellerbroek, and J.M. Hoekstra, *Three-Dimensional Velocity Obstacle Method for UAV Deconflicting Maneuvers.* 2015.

178. Wilkie, D., J.v.d. Berg, and D. Manocha. *Generalized velocity obstacles*. in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009.

179. Large, F., C. Laugier, and Z. Shiller, *Navigation among moving obstacles using the NLVO: Principles and applications to intelligent vehicles.* Autonomous Robots, 2005. **19**(2): p. 159-171.

180. van den Berg, J., M. Lin, and D. Manocha, *Reciprocal Velocity Obstacles for real-time multi-agent navigation*, in *2008 IEEE International Conference on Robotics and Automation (ICRA)*. 2008, IEEE: Pasadena, CA, USA. p. 1928-1935.

181. van den Berg, J., S.J. Guy, M. Lin, and D. Manocha, *Reciprocal n-Body Collision Avoidance.* Robotics Research, 2011. **70**: p. 3-19.

182. Bareiss, D. and J. van den Berg, *Generalized reciprocal collision avoidance.* International Journal of Robotics Research, 2015. **34**(12): p. 1501-1514.

183. Bareiss, D. and J. van den Berg, *Reciprocal Collision Avoidance for Robots with Linear Dynamics using LQR-Obstacles.* 2013 Ieee International Conference on Robotics and Automation (Icra), 2013: p. 3847-3853.

184. Coue, C., C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, *Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application.* The International Journal of Robotics Research, 2006. **25**(1): p. 19-30.

185. Lee, B.H., J.D. Jeon, and J.H. Oh, *Velocity obstacle based local collision avoidance for a holonomic elliptic robot.* Autonomous Robots, 2016. **41**(6): p. 1347-1363.

186. Large, F., S. Sekhavat, Z. Shiller, and C. Laugier. *Towards real-time global motion planning in a dynamic environment using the NLVO concept*. in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. 2002.

187. Fulgenzi, C., A. Spalanzani, and C. Laugier, *Combining Probabilistic Velocity Obstacles and Occupancy Grid for safe Navigation in dynamic environments*, in *Workshop on Safe Navigation*. 2007, IEEE ICRA: Rome, Italy. 2007.

188. Alonso-Mora, J., A. Breitenmoser, P. Beardsley, and R. Siegwart, *Reciprocal Collision Avoidance for Multiple Car-like Robots.* 2012 Ieee International Conference on Robotics and Automation (ICRA), 2012: p. 360-366.

189. Velasco, G.A.M., C. Borst, J. Ellerbroek, M.M. van Paassen, and M. Mulder, *The Use of Intent Information in Conflict Detection and Resolution Models Based on Dynamic*

Velocity Obstacles. Ieee Transactions on Intelligent Transportation Systems, 2015. **16**(4): p. 2297-2302.

190. Zhang, W., S. Wei, Y. Teng, J. Zhang, X. Wang, and Z. Yan, *Dynamic Obstacle Avoidance for Unmanned Underwater Vehicles Based on an Improved Velocity Obstacle Method.* Sensors (Basel), 2017. **17**(12).

191. Chen, P., Y. Huang, J. Mou, and P.H.A.J.M. van Gelder, *Ship collision candidate detection method: A velocity obstacle approach.* Ocean Engineering, 2018. **170**: p. 186-198.

192. Van Gelder, P.H.A.J.M., *H-infinity-control design on a spraydryer process. Department of Mathematics and Computer Science*, in *Department of Mathematics and Computer Science*. 1991, TU Eindhoven: Eindhoven, NL.

193. Skjetne, R., Ø.N. Smogeli, and T.I. Fossen, *A Nonlinear Ship Manoeuvering Model: Identification and adaptive control with experiments for a model ship.* Modeling, Identification and Control: A Norwegian Research Bulletin, 2004. **25**(1): p. 3-27.

194. IMO, *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)*, I.M. Organization, Editor. 1972.

195. Chen, L., *Cooperative Multi-Vessel System for Waterborne Transport*, in *3me*. 2019, Delft University of Technology: Delft, the Netherlands.

196. Best, A., S. Narang, D. Barber, and D. Manocha, *AutonoVi: Autonomous Vehicle Planning with Dynamic Maneuvers and Traffic Constraints*, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, IEEE: Vancouver, BC, Canada.

197. Zuhaib, K., A. Khan, J. Iqbal, M. Ali, M. Usman, A. Ali, S. Yaqub, J. Lee, and C. Han, *Collision Avoidance from Multiple Passive Agents with Partially Predictable Behavior.* Applied Sciences, 2017. **7**(9): p. 903.

198. Li, Y., A.C. Landsburg, R.A. Barr, and S.M. Cahsal, *Improving ship maneuverability standards as a means for increasing ship controllability and safety.* Oceans 2005, Vols 1-3, 2005: p. 1972-1981.

199. Aven, T., *The risk concept—historical and recent development trends.* Reliability Engineering & System Safety, 2012. **99**: p. 33-44.

200. Andretta, M., *Some considerations on the definition of risk based on concepts of systems theory and probability.* Risk Anal, 2014. **34**(7): p. 1184-95.

201. Blom, H.A.P. and G.J. Bakker, *Safety Evaluation of Advanced Self-Separation Under Very High En Route Traffic Demand.* Journal of Aerospace Information Systems, 2015. **12**(6): p. 413-427.

202. Bouarfa, S., H.A.P. Blom, and R. Curran, *Agent-Based Modeling and Simulation of Coordination by Airline Operations Control.* IEEE Transactions on Emerging Topics in Computing, 2016. **4**(1): p. 9-20.

203. Vendittelli, M., J.-P. Laumond, and C. Nissoux, *Obstacle Distance for Car-Like Robots.* IEEE Transactions on Robotics and Automation, 1999. **15**(4): p. 678-691.

204. Sahawneh, L.R. and R.W. Beard, *A Probabilistic Framework for Unmanned Aircraft Systems Collision Detection and Risk Estimation.* 2014 Ieee 53rd Annual Conference on Decision and Control (Cdc), 2014: p. 242-247.

205. Liu, J., R. Hekkenberg, E. Rotteveel, and H. Hopman, *Literature review on evaluation and prediction methods of inland vessel manoeuvrability*. Ocean Engineering, 2015. **106**: p. 458-471.

206. Crane, C.L., H. Eda, and A. Landsburg, *Controllability*, in *Principles of Naval Architecture*, E.V. Lewis, Editor. 1989, The SNAME: Jersey, US.

Summary

Ship collision is a classical problem for maritime practitioners and researchers. Human error is a major cause of collision accidents, which motivates researchers to develop automation systems replacing navigators on board. In particular, the developments of self-driving cars, robotics, and unmanned drones boost the confidence of having Maritime Autonomous Surface Ships (MASS). However, before autonomous ships fully replace all conventional ships, supporting the situational awareness of human operators is still a rigid demand, where the design of human-machine interactions (HMIs) become crucial.

This dissertation pays special attention to HMIs during collision avoidance. Specifically, a Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS) is proposed, which allows the human operators and automation to share their intelligence. During collision avoidance, the HMI-CAS can not only offer one (optimal) solution to human users but also indicate the dangerous solutions. So, the decision process of the machine becomes transparent for the human users. The human operators can not only read and understand the solutions found by the machine but also validate and modify the solutions via an interface in the HMI-CAS. Without human interventions, the HMI-CAS works as an autonomous collision avoidance system. Moreover, to support the operators decide in time, a novel measure of collision risk using a concept called "room-for-maneuver" is proposed. Therefore, the measured risk reflects the dangerous level of the approaching ships and the difficulty to handle the encounter.

Human-machine interaction in the developments of MASS

This research found that the focuses of studies for manned and unmanned ships are different and complementary: the research for manned ships concentrates more on conflict detection; the research for unmanned ships concentrates more on conflict resolution. Thus, it is beneficial to learn from each other. However, developing various levels of unmanned ships requires not only

better conflict detection and conflict resolution but also a user-friendly design that supports various HMIs, which is less discussed in the literature.

The benefits of developing HMIs are:

(1) Supporting MASS I-III;
(2) Building trust to ASVs and promoting ASVs;
(3) Borrow knowledge from human operators to develop rule-compliant ASVs.
(4) Improving existing collision alert system with more functions

Functions and framework of the HMI-CAS

The demands on HMIs in collision avoidance contain seven aspects, i.e., "early warning", "solution advice", "automated solution execution", "cooperation support", "trajectory prediction", "data collection", and "turning off/on the system". "Solution advice" and "cooperation support" are less discussed, but they are crucial for supporting collision avoidance and are required in Type I-III MASS and Control Mode 1-4. Accordingly, five possible solution forms supporting HMIs are identified, i.e., one solution, an optimal solution, finite feasible solutions, a set of feasible solutions, all sets of unsafe solutions.

To support autonomous functions, the framework of the HMI-CAS remains the same as the Guidance, Navigation, and Control (GNC) system. Nevertheless, for exchanging information between human operators and the systems, an interface is designed.

A family of algorithms supports various HMIs during collision avoidance

With a comprehensive literature review on the state-of-the-art methods and the analysis on the requirements of HMIs, a family of Velocity Obstacle (VO) algorithms are chosen to be applied in the HMI-CAS. Specifically, these algorithms visualize the solution space of the own-ship (OS), collect the dangerous solutions in VO sets, and then find collision-free solutions out of the VO sets. Thus, the users can easily understand how the system make decisions. Three VO algorithms are employed, which are linear velocity obstacle (LVO) algorithm, non-linear velocity obstacle (NLVO) algorithm, and probabilistic velocity obstacle (PVO) algorithm in this research. Different from traditional methods, the VO algorithms do not require the target ship (TS) to keep constant speed and course. Therefore, these algorithms can incorporate the (non-linear, probabilistically predictable, etc.) predicted trajectories of the obstacles and TSs in collision avoidance.

Improved conflict resolution in the HMI-CAS considering ship dynamics and navigational regulations

The VO algorithms are convenient for human operators to join the conflict resolution in the HMI-CAS. However, the dynamics of ships are ignored when identifying the closed regions of dangerous solutions. Ships usually have poor maneuverability, and therefore more time and space are needed to achieve the desired velocity. That means the collision might happen before the ship achieves the desired velocity in the extreme case. To handle this issue, the generalized velocity obstacle (GVO), which considers the ship's dynamics and controllers, is integrated into the HMI-CAS. Moreover, the interface of the HMI-CAS is modified to be user-friendly. Comparing with the HMI-CAS using VO algorithms, using the GVO in the HMI-CAS has two advantages: (1) supporting collision avoidance in a close distance where the VO algorithms might lead to a collision; (2) avoiding collisions using less evasive actions, which is advocated by navigational regulations, i.e., COLREGs.

Improved conflict detection in the HMI-CAS for supporting evasive actions in time.

The core of conflict detection is a measurement of collision risk that indicates the safety/danger level of the ship. When the risk is high, the collision alarm is triggered for taking evasive actions in the HMI-CAS. There are numerous risk measures which can assess collision risk, but most of them are independent of conflict resolution. As a result, the measured risk is independent of whether the collision is avoidable, which is not suitable for the proposed HMI-CAS.

A Time-varying Collision Risk (TCR) measure is proposed, which defined the collision risk as to the chance of avoiding dangers, i.e., the percentage of controls leading to collisions. High TCR level reflects a condition where the collision event is difficult to solve, and the collision is highly possible to happen. This measure is also compliant with common sense that the risk would not decrease when the ship encounters with more ships or the ship has poor maneuverability. TCR indicator, therefore, is proposed to be an auxiliary tool in the conflict detection module of HMI-CAS. Specifically, human operators can safely postpone evasive actions until the TCR reaches 1.

In summary, this dissertation proposes a prototype of a collision avoidance system that supports human-machine interactions, call HMI-CAS. Instead of replacing humans on board, the proposed system aims at bridging the intelligence of humans and machines, which enables cooperation between humans and machines and enriches the choice of collision avoidance systems for supporting human operators and for developing autonomous ships. It is expected to reduce the collision risk, to facilitate human operators, and to promote the development of Maritime Autonomous Surface Ships.

Curriculum Vitae

Yamin Huang was born in February, 1990 in Xianyou, Fujian, China. He obtained the B.Sc. degree on Maritime Administration in 2011 and the M.Sc. degree on Traffic Information Engineering and Control in 2014, both from Wuhan University of Technology in Wuhan, China.

Starting from September 2014, Yamin is sponsored by China Scholarship Council as a PhD candidate at the Safety and Security Science Group, the Department of Value, Technology, and Innovation, Delft University of Technology, Delft, the Netherlands. In his PhD project, Yamin proposes an advanced collision avoidance system supporting human-machine interactions that not only avoids collision automatically but also supports the human to take over control of the automated collision avoidance. His research interests include nautical risk assessment, human-machine interaction, autonomous ships, and maritime traffic management.

Peer-reviewed publications

Y. Huang, L. Chen, P. Chen, R.R. Negenborn, P.H.A.J.M. van Gelder. 2020. "*Ship collision avoidance methods: state-of-the-art*". Safety Science. 121: 451-473. DOI: 10.1016/j.ssci.2019.09.018.

Y. Huang, L. Chen, P.H.A.J.M. van Gelder. 2019. "*Generalized velocity obstacle algorithm for collision prevention at sea*". Ocean Engineering 173: 142-156. DOI: 10.1016/j.oceaneng.2018.12.053.

Y. Huang, P.H.A.J.M. van Gelder, Y. Wen. 2018. "*Velocity obstacle algorithms for collision prevention at sea*". Ocean Engineering 151:308-321. DOI: 10.1016/j.oceaneng.2018.01.001

Y. Huang, P.H.A.J.M. van Gelder. 2019 (Online). "*Time-varying risk measurement for ship collision prevention*". Risk Analysis. DOI: 10.1111/risa.13293

Y. Huang, T.L. Yip, Y. Wen. 2019. "*Comparative analysis of marine traffic flow in classical models*". Ocean Engineering 187. DOI: 10.1016/j.oceaneng.2019.106195

Y. Huang, P.H.A.J.M. van Gelder. "*A measure of collision risk for triggering evasive actions*". (Submitted)

Y. Huang, L. Chen, R.R. Negenborn, P.H.A.J.M. van Gelder. "*A HMI-CAS for human-machine cooperation during collision avoidance*". (Submitted)

D. Lei, F. Goerlandt, O.A. Valdez Banda, **Y. Huang***, Y. Wen, P. Kujala. "*Improving stand-on ship's situational awareness by estimating the intention of the give-way ship*". Ocean Engineering. (Under Review)

L. Chen, R.R. Negenborn, **Y. Huang**, J.J. Hopman. "*Survey on Cooperative Control for Waterborne Transport*". (Submitted)

L. Chen, **Y. Huang**, H. Zheng, J.J. Hopman, R.R. Negenborn. (Online). "*Cooperative multi-vessel systems in urban waterway networks*". IEEE Transactions on Intelligent Transportation Systems. DOI: 10.1109/TITS.2019.2925536

P. Chen, **Y. Huang**, J. Mou, P.H.A.J.M. van Gelder. 2019. "*Probabilistic risk analysis for ship-ship collision: state-of-the-art*". Safety Science. 117, 108-122. DOI: 10.1016/j.ssci.2019.04.014

P. Chen, **Y. Huang**, J. Mou, P. H. A. J. M. van Gelder. 2018. "*Ship collision candidate detection method: A velocity obstacle approach*". Ocean Engineering 170:186-198. DOI: 10.1016/j.oceaneng.2018.10.023

Y. Wen, **Y. Huang**, C. Zhou, J. Yang, C. Xiao, X. Wu. 2015. "*Modelling of marine traffic flow complexity*". Ocean Engineering 104:500-510. DOI: 10.1016/j.oceaneng.2015.04.051

Conferences

Y. Huang, P.H.A.J.M. 2019. "*Measuring Ship Collision Risk in a Dense Traffic Environment*". 13th International Conference on Marine Navigation and Safety of Sea Transportation, Gdynia, Poland.

Y. Huang, P.H.A.J.M. van Gelder. 2017. "*Non-linear velocity obstacles with applications to the maritime domain*". In the 17th International Congress on Maritime Transportation and Harvesting of Sea Resources (IMAM 2017), eds. Guedes Soares and Teixeira. Lisbon, Portugal: Taylor & Francis Group.

Y. Huang, P.H.A.J.M. van Gelder & M.B. Mendel. 2016. "*Imminent ships collision risk assessment based on velocity obstacle*". Walls, Revie & Bedford (Eds). Risk, Reliability and Safety: Innovating Theory and Practice (ERESL 2016). 693-700., Glasgow, the UK.