Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Hongyu Zhou

Autonomous guidance, stepwise path planning, and path-following control with anti-collision for autonomous marine robots

Master's thesis in Marine Technology
Supervisor: Prof. Roger Skjetne
June 2020

NTNU
Kunnskap for en bedre verden

Hongyu Zhou

Autonomous guidance, stepwise path planning, and path-following control with anti-collision for autonomous marine robots

Master's thesis in Marine Technology
Supervisor: Prof. Roger Skjetne
June 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Abstract

This thesis presents methods for autonomous guidance, stepwise path planning, and path-following control with anti-collision and COLREGs compliance. The system is validated through simulations.

Two waypoint generation methods are developed, i.e., the navigation function method and the stream function method. These two methods are modified to achieve COLREGs compliance. The navigation function method guarantees global convergence. While the stream function method gives shorter and smoother path.

Each waypoint generation method can be integrated with a path generation algorithm. The path generation algorithm formulates a quadratic programming problem to minimize path length between two waypoints. A controller is designed using backstepping and a thrust allocation algorithm is implemented to perform path following.

Preface

This master thesis concludes my Master's Degree and two-year study in Marine Technology at the Norwegian University of Science and Technology. It is written in the spring of 2020, and is to some extent a continuation on the specilization project carried out in the fall of 2019. The thesis contains a literature review on relevant topics for the thesis and development of autonomous guidance methods. Results have been obtained from simulations.

The work on the thesis starts from January, 2020. The project process first includes a literature review. Based on the review and project description, relevant methods are selected and implemented. These methods then are assessed by simulations.

A weekly report should be sent to my supervisor on latest progress and the goal for next week. Based on the report, the supervisor is able to give necessary guidance and suggestions. Besides, a meeting is held every two week by my co-advisor, Zhengru Ren. In the meeting, certain problems can be discussed, such as English writing, thesis organization, and recommendations for future work.

Acknowledgement

The work on my master's thesis has been challenging but it is also a precise and interesting experience. I have received much help during my work, and I would like to acknowledge them here.

First I would like to thank my supervisor Prof. Roger Skjetne for providing me with an opportunity to work on the problem of path planning, and his supervision on my work.

I would also like to thank my co-advisor, Zhengru Ren, who generously gives me suggestions on improvement of my work.

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER OF TECHNOLOGY THESIS DEFINITION (30 SP)

Name of the candidate: Zhou, Hongyu

Field of study: Marine control engineering

Thesis title (Norwegian): Autonom guidance, stegvis baneplanlegging, og automatisk banefølging med kollisjonsunngåelse for autonome marine roboter

Thesis title (English): Autonomous guidance, stepwise path planning, and path-following control with anti-collision for autonomous marine robots

Background

Autonomous path-planning and path-generation with obstacle avoidance is an important issue for autonomous surface vessels, marine robots, and other robotic platforms. In order to perform path-planning, a guidance model is needed. This model must represent the workspace of the robot and contain information of the environment necessary to safely guide the vehicle according to its mission objective. Typical objectives are either transit to one or more targets and complete coverage of the work space. *Transit* typically implies finding the optimal path (shortest, minimum energy, minimum time, etc.) from the robot's initial pose to the target pose. *Complete coverage* is the typical objective of robot vacuum cleaners and lawn movers to completely cover all positions in the workspace. Possible guidance models for these robots is the navigation function, stream function, etc., which embeds information of static obstacles (and dynamic if slow enough) and the targets, and it guarantees dynamically to provide a path to the target(s). However, the path may neither be nice, nor optimal.

The objective of this thesis is to study navigation function and other relevant guidance models in static and dynamic environment as guidance model(s) for autonomous robot/vessel guidance and maneuvering problems. The guidance model should frequently update as the workspace of autonomous robot/vessel changes. Based on the guidance model, the robot/vessel performs path planning and path following using a suitable control law, e.g. maneuvering, reaching the target position with collision avoidance.

Work description

1. Perform a background and literature review to provide information and relevant references on:
   - Collision avoidance methods for moving obstacles.
   - Dynamic guidance models such as navigation function, stream function, etc.
   - Relevant dynamical model of a marine robot – to be used in a case study.
   - Relevant COLREG rules and COLREG compliance.
   - Method(s) for generation of feasible path segments
   - Maneuvering control methods for 2D path-following control of marine autonomous robots.

   Write a list with abbreviations and definitions of terms and symbols, relevant to the literature study and project report.

2. Propose a simplified dynamics of a marine autonomous robot. Define also a few workspace environments for the robot, defining target(s) and static and moving obstacles. Use these setups as simulation cases in your analysis. COLREG compliance should be included in the case studies. Some setup(s) should challenge your underlying guidance models and maneuvering methods.

3. Study relevant guidance models that are applicable to your obstacle-avoidance case study, incl. the navigation function method. Present and show insight into the performance and properties of each method studied.

4. Implement the guidance models, path-planning, path-generation, and maneuvering control for each proposed guidance model for your marine robot. Plan the path from starting point to one or more

destinations by means of the guidance model, while avoiding obstacles. Assess their COLREG compliance during maneuvers for avoiding the obstacles.

5. For each guidance model method proposed and implemented, test on your simulation cases, analyze the results by comparing/contrasting the resulting performance with evaluation of the key performance indicators. Discuss specifically COLREG behavior in each case.

Specifications

Every weekend throughout the project period, the candidate shall send a status email to the supervisor and co-advisors, providing two brief bulleted lists: 1) work done recent week, and 2) work planned to be done next week.

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the various steps in the deduction.

The report shall be organized in a logical structure to give a clear exposition of background, problem, design, results, and critical assessments. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is not expected to be longer than 70 A4-pages, 100 B5-pages, from introduction to conclusion, unless otherwise agreed upon. It shall be written in English (preferably US) and contain the elements: Title page, abstract, acknowledgement, project definition, list of symbols and acronyms, table of contents, introduction (project motivation, objectives, scope and delimitations), background/literature review, problem formulation, method, results, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct, which is taken very seriously by the university and cause consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with an electronic copy to the main supervisor and department according to NTNU administrative procedures. The final revised version of this thesis description shall be included after the title page. Computer code, pictures, videos, dataseries, etc., shall be included electronically with the report.

Start date: 15 January, 2020 **Due date:** As specified by the administration.

Supervisor: Roger Skjetne
Co-advisor(s): Zhengru Ren, Mathias Marley

Trondheim, 23.03.2020

_____

Roger Skjetne
Supervisor

Table of Contents

List of Tables

List of Figures

Nomenclature

Acronyms

UAV	=	Unmanned Aerial Vehicle
AV	=	Autonomous Vehicle
COLREGs	=	International Regulations for Preventing Collisions at Sea
LoA	=	Level of Autonomy
DOF	=	Degree of Freedom
APF	=	Artificial Potential Field
GNC	=	Guidance, Navigation, and Control
NED	=	North-East-Down
CLF	=	Control Lyapunov Function
VSP	=	Voith-Schneider Propeller

Symbol

N_o	=	The number of obstacles
$p_d(s)$	=	Desired path
$\boldsymbol{\eta}$	=	Ship pose
$\boldsymbol{\eta}_d$	=	Desired ship pose
s	=	Path variable
v_s	=	Desired path speed
x	=	North position
y	=	East position
ψ	=	Heading
$\mathbf{M}$	=	System mass matrix
$\mathbf{C}$	=	Coriolis and centripetal matrices
$\mathbf{D}$	=	Damping matrix
$\mathbf{g}$, $\mathbf{g}_0$	=	Restoring force vectors
$\boldsymbol{\nu}_r$	=	Relative velocity vector
$\boldsymbol{\nu}_c$	=	Current velocity vector
$\boldsymbol{\nu}$	=	Linear and angular velocities in BODY frame
$\boldsymbol{\tau}$	=	Propulsion forces and moments
$\boldsymbol{\tau}_{wind}$	=	Wind forces and moments
$\boldsymbol{\tau}_{wave}$	=	Wave forces and moments
$\mathbf{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})$	=	Six degree rotation matrix
$\mathbf{R}(\psi)$	=	Rotation matrix
$\mathbf{B}(\boldsymbol{\alpha})$	=	Configuration matrix
$\boldsymbol{\alpha}$	=	Vector of thruster angles
$\mathbf{K}$	=	Diagonal force coefficient matrix
$\mathbf{u}$	=	Normalized thruster forces vector
L_x, L_y	=	Size of the workspace
N_x, N_y	=	The number of grid points in x-axis and y-axis

$U(\mathbf{p})$	=	Potential field
$\mathbf{p_t}$	=	Position of the target
$\mathbf{p}_i$	=	Position of the ith obstacle
$\mathbf{v}_i$	=	Velocity of the ith obstacle
κ	=	Tuning parameter in a navigation function
$\delta U(\mathbf{p})$	=	The difference of potential between current and next sample time
$\mathbf{p}_c$	=	Position of the current waypoint
$\mathbf{p}_w$	=	The set of possible next waypoints
$\mathbf{p}_{next}$	=	Position of the next waypoint
n_h, R, L, γ_1	=	Tuning parameters for choosing waypoint in the navigation function method
$\psi_v(\mathbf{p})$	=	Stream function for multiple obstacles
$\psi_i(\mathbf{p}$	=	Stream function generated by the ith obstacle
ψ_{v_i}	=	Stream function of a vortex flow
C_{v_i}, γ_2	=	Tuning parameter in a stream function
n_r	=	Range of the current waypoint
$\mathbf{B}(s)$	=	A Bézier curve
$\mathbf{P}_{i,k}$	=	The ith control point for kth path segment
$\mathbf{WP}_k$	=	The kth waypoint
δ, μ_0	=	Tuning parameters in pragmatic approach
α_k	=	The kth path angel
ζ, ϵ	=	Tuning parameters in optimization problem
$\mathbf{K}_1$, $\mathbf{K}_2$, μ	=	Tuning parameters in controller design

Chapter 1

Introduction

1.1 Motivation

The research on autonomy is gaining increasing attention in the robotics community. It is a promising area because of its extensive application in practice. For example, unmanned aerial vehicles (UAVs) can be used to monitor forest and agriculture, autonomous vehicles (AVs) can reduce crashes, congestion, and emissions, and reducing crewing on autonomous ships. In autonomous robotics, a central problem is path planning. For humans, it is trivial to plan a path. While for robots, such a task is challenging.

Guidance models are important for solving the problem of path planning. They should be able to characterize the surrounding terrain to be used to find a path and guide the robot to its targets dynamically. However, the path found by these guidance models might not be optimal. In addition, these guidance models have their own limitations and, therefore, have their own application domains. For example, for underactuated vehicles, such as marine surface vehicles, a path that has large heading change is not preferred. In addition, a set of pre-set rules should be considered to guide collision avoidance. A popular set of rules is "International Regulations for Preventing Collisions at Sea" (COLREGs).

The problem of path planning is to find an optimal or suboptimal path from the starting position to a robot's goal position. To achieve the task of path planing, an appropriate guidance model is necessary. The guidance model must be able to accurately represent the robot's workspace and embed information that can safely and efficiently guide the robot according to its task requirement. Typical objectives of path planning are either transit (to one or more target) or complete coverage. Transit means finding the optimal path (shortest, minimum energy, minimum time, etc.) from the robot's initial position to the target position. Complete coverage is the typical objective of robot vacuum cleaners, lawn movers and robots for search and rescue.

1.2 Objectives

The objective of this thesis is to study navigation function and other relevant guidance models in static and dynamic environment as guidance model(s) for autonomous robot/vessel guidance and maneuvering problems. The guidance model should frequently update as the workspace of autonomous robot/vessel changes. Based on the guidance model, the robot/vessel performs path planning and path following using a suitable control law, e.g. maneuvering, reaching the target position with collision avoidance.

This is achieved by:

- Perform a background and literature review to provide information and relevant references on collision avoidance methods, relevant COLREGs rules, path generation method, maneuvering control methods, etc.

- Propose a simplified dynamics of an autonomous robot and workspace of the robot. Define a few workspace environments, that include target(s) and static and moving obstacles. Use these setups as simulation cases in your analysis. COLREGs compliance should be included.

- Study relevant guidance models. Present and show insight into the performance and properties of each method studied.

- Implement the guidance models, path-planning, path-generation, and maneuvering control for each proposed guidance model for proposed marine robot.

- For each guidance model method proposed and implemented, test on simulation cases, analyze the results by comparing the resulting performance with evaluation of the key performance indicators. Discuss specifically COLREGs behavior in each case.

1.3 Scope and delimitations

The scope of this thesis is to present insight on the theory and methods relative to autonomous guidance and path planning methods and illustrate them through sufficient examples. Those methods include navigation function, stream function, maneuvering theory, etc. Results of computer simulations are use to evaluate those path planning algorithms. In addition, selected COLREGs rules are used to assess the resulting performance.

In this work, we assume that the workspace is a two-dimensional (2D) horizontal plane. A simplified dynamics of a marine robot is proposed, where only horizontal degree of freedoms (DOFs) are considered. It is also assumed that the information of the workspace is known, i.e., the positions and velocities of target(s) and obstacles.

1.4 Contributions and thesis outline

The contribution of this thesis is the development of two guidance models, i.e., the navigation function method and the stream function method, for autonomous guidance and stepwise path planning with COLREGs compliance. The proposed guidance models decide the discrete waypoints recursively that guide a robot toward its target(s). A feasible path is then generated using a path-generation algorithm. The marine control system is designed to perform path following. The whole system has been validated through simulations.

The outline of the thesis is as follows:

- Chapter 2 presents relevant background information on autonomous system architecture, path-planning methods, path-generation methods, control design methods, and relevant COLREGs rules.

- Chapter 3 presents the system architecture and problem formulation.

- Chapter 4 presents the dynamic model for control design, thruster dynamics, and workspace representation.

- Chapter 5 explains the algorithms for waypoint generation with COLREGs compliance, and path generation. Two waypoint generation methods are modified to achieve COLREGs compliance.
- Chapter 6 presents the controller design and thrust allocation.
- Chapter 7 presents the simulation results.
- Chapter 8 gives the final conclusions and suggestions for further work.

Chapter 2

Background

2.1 Autonomous systems

2.1.1 Deliberation for autonomous robots

This section is primarily based on Ingrand and Ghallab (2017). Autonomous robots interact with their working environment through their sensory systems and have to act deliberately in order to fulfill their missions. The meaning of acting deliberately is that taking actions through sound and justifiable reasoning based on their objectives.

Autonomous robots perform various tasks involving a variety of interactions with various environments, which cannot be programmed at the design stage to make them able to take all possible actions. Therefore, for autonomous robots working in open environments and performing various tasks to successfully complete their missions, deliberation is necessary. In summary, autonomy plus variety require deliberation.

Acting deliberately requires several distinct functions. In Ingrand and Ghallab (2017), the requirement is divided into six deliberation functions, i.e., planning, acting, observing, monitoring, goal reasoning and learning as shown in Figure 2.1.

Their definitions are given below, see Ingrand and Ghallab (2017) for further elaboration. It should be noticed that their borderlines are blurry and depend on specific applications.

- Planning: motivated by missions or purpose, combines prediction and search to find a feasible trajectory in the working space based on predictive models of the environment and feasible actions.
- Acting: refines planned actions and outputs commands appropriate for the current situation and reacts to changes of the robot and its working space; both refinement and reaction may rely on a organized sequence of actions. Stimulus input from sensors are transformed to output to actuators through an organized and appropriate sequence of actions, in order to achieve certain purpose.
- Observing: detects and recognizes characteristic features of the working space, as well as other relevant information for the objective such as actions and plans. This function combines bottom-up processes, from sensors to meaningful data, with top-down activities such as sensing actions and planning for information gathering.
- Monitoring: compares predicted robots' states and observed information of the working space. Its function is to recognize and interpret discrepancies, perform diagnosis and trigger recovery actions when they are necessary.

- Goal reasoning is a type of monitoring but only at the mission level. It concerns about commitments and goals. This function assess the relation between commitments and goals, based on observed new information, such as new constraints, new opportunities and possible failures. Its task is to assess if some commitments should be abandoned and if it is necessary to update the current goals.

- Learning is the process of acquiring new knowledge, becoming adaptive and improving robots themselves through past experience the models needed for deliberation.

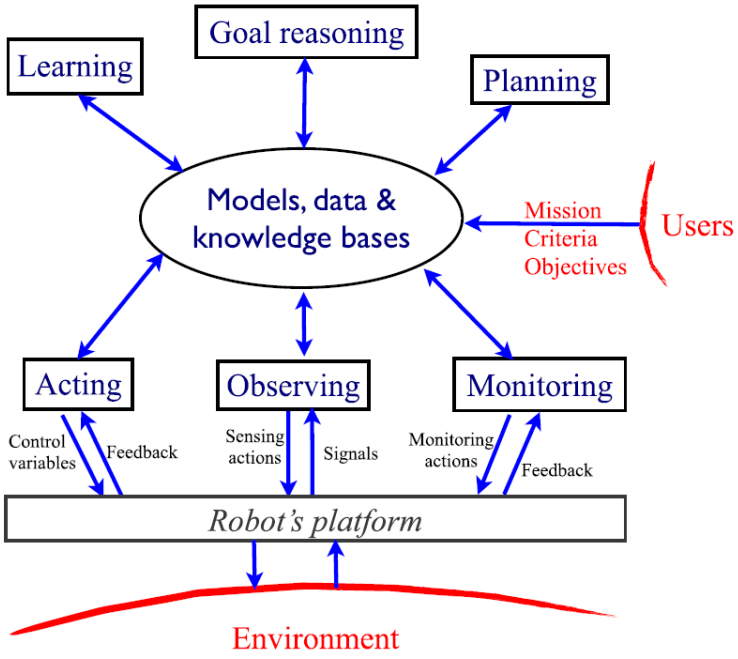


Figure 2.1: Overview of deliberation functions. Courtesy: Ingrand and Ghallab (2017).

2.1.2 Autonomous marine robots

The concept of autonomous or unmanned ships was introduced decades ago. Nicola Tesla's demonstration of a radio-controlled model boat in New York's Madison Square in 1898 may also fall into the concept of autonomous ships. Autonomous vehicles have been researched by some transport modes since the 1970s. From 1982 to 1988, Japan investigated remote control of ships in the Project *Highly Reliable Intelligent Ship* (Rødseth, 2019).

The EU-project *MUNIN* running from 2012 to 2015 is the first large scale study on unmanned and autonomous merchant ships (Rodseth and Burmeister, 2012). The objective of the project *MUNIN* is to investigate the possibility of transforming a Handymax dry bulk carrier into an unmanned autonomous ship. The results showed that it was not feasible from a commercial standpoint. But it would be possible to build a fully autonomous ship for other operations. Since then, new researches and studies on autonomous ships have been increasing steadily.

The general definition of the level of autonomy (LoA) of ships is given in Rødseth et al. (2018). From the perspective of collision avoidance, the specific definition of LoA is provided in Huang et al. (2020):

- LoA 0: No machine is involved and human takes the responsibility for collision avoidance.

- LoA 1: Machines provide supports and certain service in conflict detection, while the human operators directly control the ship.

- LoA 2: The human operators directly control the ship supported by machines in conflict detection and conflict resolution.

- LoA 3: Machines control the ship with human monitor. Machines need to help the human understand conflict resolution. The human can indirectly control the ship via machines or directly controls the ship via the on-board operators.

- LoA 4: Machines control the ship independently. Under emergent circumstances, machines informs and alarms the human who can take indirectly control via machines.

- LoA 5: The ship controlled by machines is fully autonomous and the human cannot control the ship.

The interactions between the human and machines are increasing as the LoA increases (Huang et al., 2020). From LoA 2 to LoA 4, machines are required to have more functions than conflict detection. To be specific, machines need to find anti-collision solutions, validate these solutions, and help the human in decision making.

In the process of collision avoidance, five components shown in Figure 2.2 are involved (Huang et al., 2020):

- Observer: Sensory systems that offer data to support other components.

- Motion prediction module: A fundamental module in this process. It estimates the trajectories of ships and surrounding obstacles. To predict the motion of own ships, holonomic model, kinematic model or dynamic model can be used. To predict the motion of target ships, physics-based methods, maneuver-based methods, and interaction-aware methods are proposed.

- Conflict detection module: It assesses the risk of collision and send alarms if necessary.

- Conflict resolution module: A core component in this process. It provides available solutions for collision avoidance and validate chosen solutions.

- Actuator: A component that implements the solutions found by conflict resolution module.

With respect to collision avoidance, manned ships and autonomous unmanned ships have common framework shown in Figure 2.2. On the contrary, the focuses of manned ships and autonomous unmanned ships are different. For manned ships, the main focuses are conflict detection, i.e., how to assess potential risks and remind the human if necessary. For autonomous unmanned ships, conflict detection is less of the focus. The main concentration is conflict resolution, i.e., how to find an collision-free trajectory. The different decision process of manned ships and autonomous unmanned ships are shown in Figure 2.3 and 2.4.

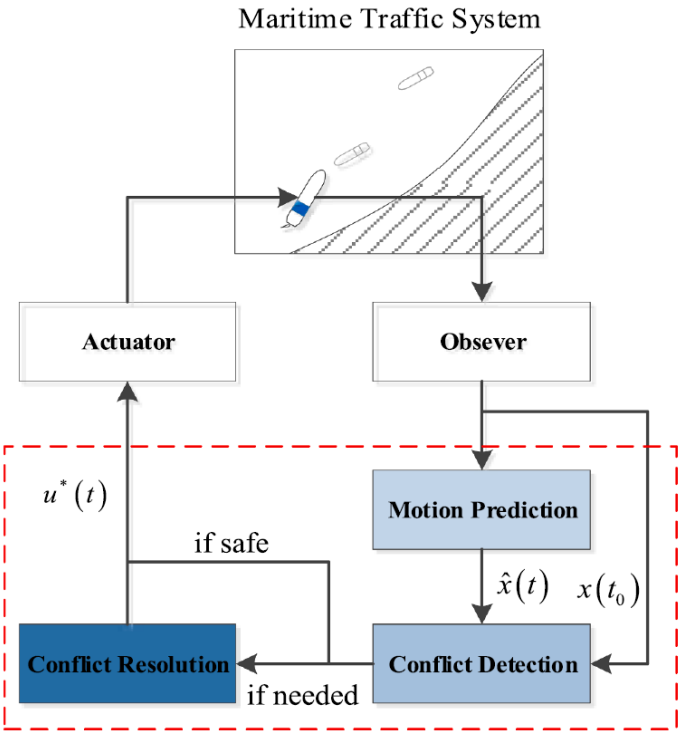


Figure 2.2: The information flow of collision prevention. Courtesy: Huang et al. (2020).

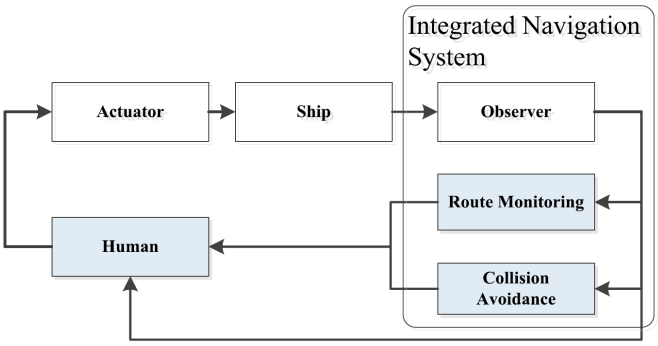


Figure 2.3: The decision process in a manned ship. Courtesy: Huang et al. (2020).

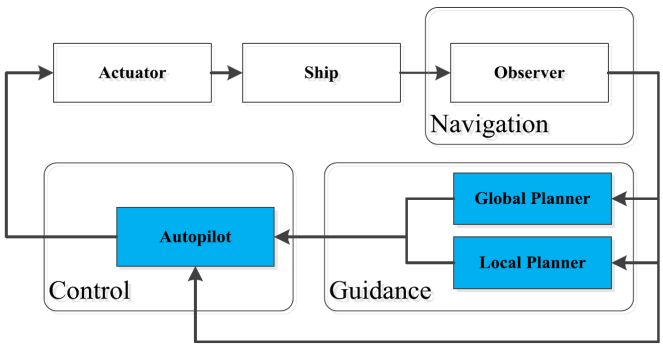


Figure 2.4: The decision process in a unmmanned ship. Courtesy: Huang et al. (2020).

2.2 Collision avoidance methods

2.2.1 Map representation

Map representation that provides accurate information of a robot's workspace is important for solving the problem of path planning. The methods of map representation can be divided into two groups (Campbell et al., 2012):

- Qualitative or topological mapping: uses nodes, arcs, and vertices to represent features. Numerical date is not referred, and therefore the features are not geometrically accurate.

- Quantitative mapping or metric mapping: represents features with numerical data. Its applications are waypoints-based path planning.

Since in this thesis it is assumed that the information of the workspace, such as positions and velocities of target(s) and obstacles, is known, we do not focus on the methods of map representation. But the importance of accurate map representation should be clearly specified. One popular and basic method of map representation is the occupancy grid mapping, shown in Figure 2.5.

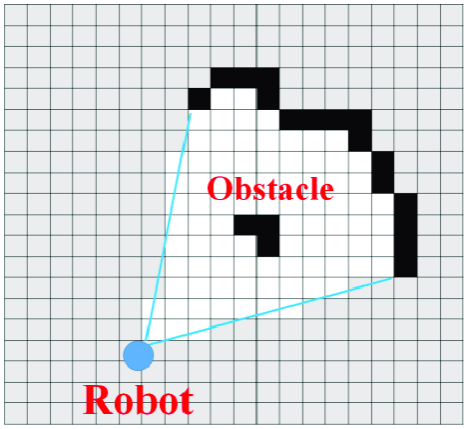


Figure 2.5: Occupancy grid mapping. Courtesy: Nam et al. (2017).

Occupancy grid mapping

The occupancy grid mapping employs a multidimensional (typically two or three-dimensional) tessellation of space into grids. Each grid stores a probabilistic estimate of its state (Elfes, 1989). The possible state might be unknown, empty or occupied by a obstacle.

To construct the occupancy grid, the robot needs to use sensors to obtain information. Hence, recovery of the workspace from sensor data is best modeled as an estimation theory problem. Here the Bayesian update rule is applied. The algorithm of the occupancy grid mapping is presented below (Thrun et al., 2005).

Algorithm 1: The occupancy grid algorithm

Algorithm occupancy_grid_mapping ($\{l_{t-1,i}\}, x_t, z_t$):
for *all girds* m_i **do**
  if m_i *in perceptual field of* z_t **then**
    $\{l_{t,i}\} = \{l_{t-1,i}\}$+inverse_sensor_model$(m_i, x_t, z_t) - l_0$;
  else
    $\{l_{t,i}\} = \{l_{t-1,i}\}$;
  end
end
return $\{l_{t,i}\}$

Here i indicates the grid index, x_t is the pose of the robot at time t, z_t is the measurement at time t, $\{l_{t,i}\}$ is log-odds representation of occupancy defined in Equation (2.2) and the inverse_sensor_model function is the inverse measurement model log-odds form given in Equation (2.3).

$$l_0 = log\frac{p(m_i = 1)}{1 - p(m_i = 1)}, \tag{2.1}$$

$$l_{t,i} = log\frac{p(m_i|x_{1:t}, z_{1:t})}{1 - p(m_i|x_{1:t}, z_{1:t})}. \tag{2.2}$$

Here, $p(m_i|x_{1:t}, z_{1:t})$ is the probability of grid m_i being occupied given measurements and pose from time 1 to t. The inverse sensor model is

$$inverse\_sensor\_model(m_i, x_t, z_t) = p(m_i|z_t, z_t). \tag{2.3}$$

The measurement distance and directions should be considered in the sensor model.

Related works

The occupancy grid mapping can be applied to a three-dimensional workspace. Dryanovski et al. (2010) proposes a Multi-Volume Occupancy Grid, which is a novel way to represent three-dimensional workspace for micro aerial vehicle mapping and navigation. The proposed method stores positive and negative sensor readings and groups the data into volume lists attached to a two-dimensional grid. The advantage of this method is incremental addition of obstacles and free space.

In order to deal with a dynamic environment, in Meyer-Delius et al. (2012), a Hidden Markov Model is used to represent the occupancy grid and model how it changes over time. Their method is tested in simulation and using real-world data. The model is suitable in dynamic environment and help to improve the mobile robots' path planning performance.

In occupancy grid method, sensors are important. Moghadam et al. (2008) points out the limitations of using either LIDAR or stereo vision for indoor mobile robot. Range sensors are able to provide 2D information while stereo vision provides 3D information. The occupancy grid is created by fusing the data from range sensors and a stereo vision system. The methods for sensor fusion need to be considered here.

2.2.2 Collision avoidance

There are abundant methods for path planning and collision avoidance, and these techniques can be classified into six categories (Huang et al., 2020):

- Rule-based method: incorporates rules in finding collision-free solutions such that the solutions are rule-compliance.
- Virtual vector method: generates a virtual vector field for collision avoidance process.
- Discretization of solutions with collision check method: finds a solution in the discrete solution-space.
- Continuous solutions with collision constraints method: finds the optimal solution in continuous solution-space where collision is considered as constraints.
- Re-planning method: finds collision-free solution in free configuration space.
- Hybrid method: takes advantage of and combines several methods for collision avoidance process.

In the following, several methods for path planning and collision avoidance are introduced. Other relevant methods can be found in Quan et al. (2020), Huang et al. (2019), Patle et al. (2019), and Mac et al. (2016).

A star

The A star (A*) (Cormen et al., 2009) is a widely used search method in path planning. It evaluate a grid point by using a cost function

$$f(n) = g(n) + h(n), \tag{2.4}$$

where $g(n)$ is the path cost from the starting point to the grid point n, and $h(n)$ is the estimated cost from grid point n to the target point. If $h(n)$ is admissible, which means it never overestimates the acutal cost from grid point n to the target point, then A* is guaranteed to find a optimal path. An admissible $h(n)$ can be chosen as the Euclidean distance between grid point n and the target point, since the straight-line path is shortest between two points.

Artificial potential field method

The artificial potential field (APF) method is a classical approach for robot navigation, and it is widely applied for real-time collision-free path planning. In this method, the workspace of the robot is represented by an artificial potential field where the target attracts the robot and the obstacles repulse the robot away.

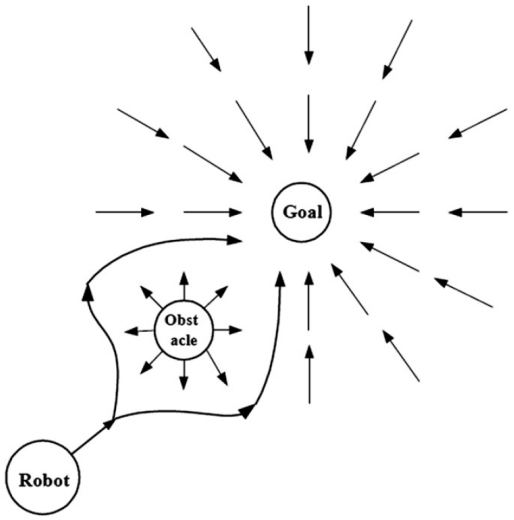


Figure 2.6: Concept illustration of the APF method. Courtesy: Lee et al. (2004).

The APF method is mathematically simple and easy to implement. The classical way to define potential function is using the positions of the robot, the target(s), and the obstacles. This can be easily modified to include more information, and the modified APF method is able to provide good result in non-static environment. In addition, this method has the advantage of combining easily with optimal control and other heuristics methods.

The problem is that the path found by the APF method is not guaranteed to be optimal, and it suffers from local minima. These problems can result in increasing energy cost and time to reach the target(s).

The APF that fills the workspace is divided into two components: attractive potential $U_{att}(\mathbf{p})$ and repulsive potential $U_{rep}(\mathbf{p})$, where $\mathbf{p}$ is the position of the robot. The total potential energy $U_{tot}(\mathbf{p})$ is defined as

$$U_{tot}(\mathbf{p}) = U_{att}(\mathbf{p}) + U_{rep}(\mathbf{p}), \tag{2.5}$$

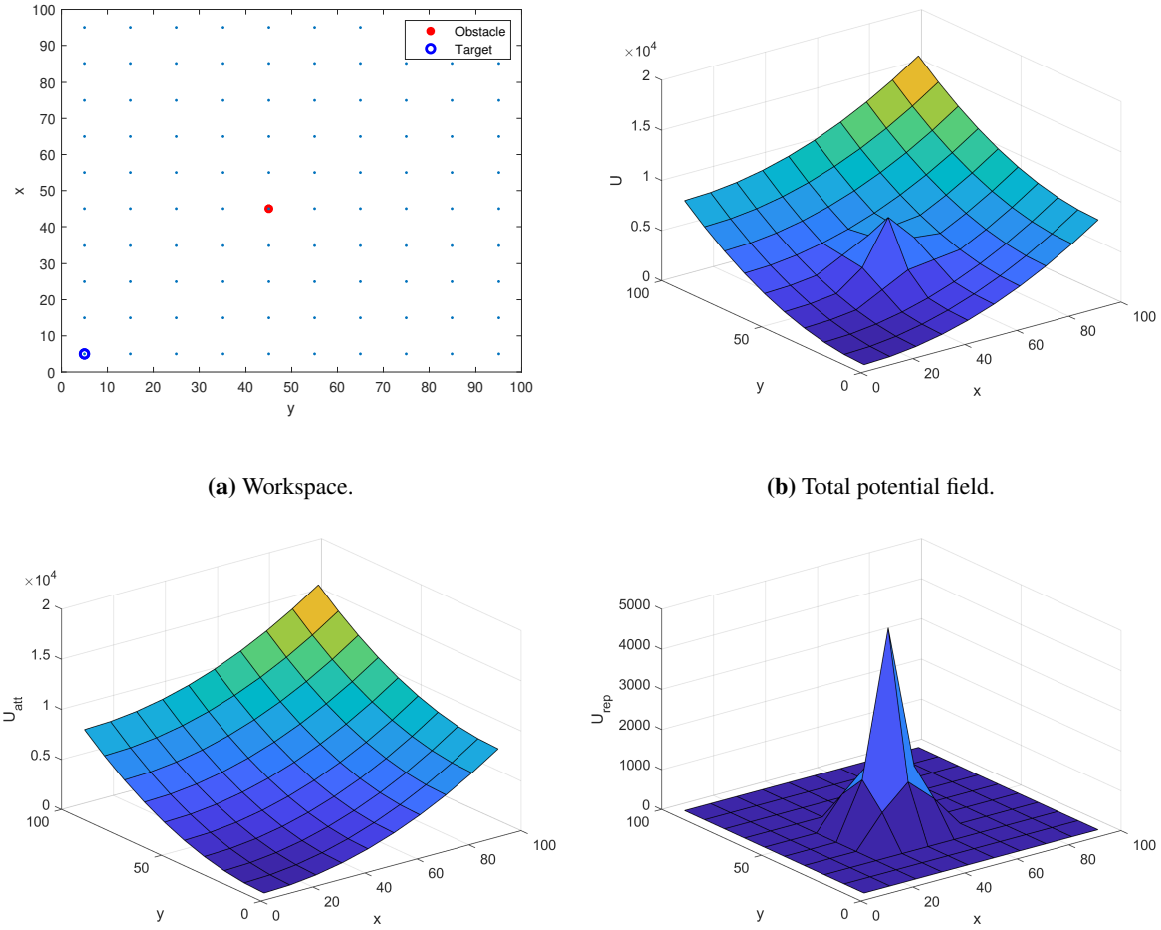
where the repulsive potential is defined within a range of obstacles. The repulsive potential increases as the robot becomes closer to the obstacles. The traditional definition of the attractive potential is a function of relative position between the robot and its targets. The attractive potential usually decreases as the robot gets closer to its target. See Figure 2.7.

Based on the definition of potential, the force acting on the robot results from the potential field is given by

$$\mathbf{F}_{tot} = \mathbf{F}_{att} + \mathbf{F}_{rep}, \tag{2.6}$$
$$\mathbf{F}_{att} = -\nabla U_{att}(\mathbf{p}), \tag{2.7}$$
$$\mathbf{F}_{rep} = -\nabla U_{rep}(\mathbf{p}), \tag{2.8}$$

where the $\mathbf{F}_{att}$ is an attractive force the robot to its targets, and $\mathbf{F}_{rep}$ is a repulsive force making the robot move away from the obstacles. Hence, the trajectory generated by this method is giving gradient vector of the potential field.

This method is particularly attractive because of its elegant mathematical analysis and simplicity (Mac et al., 2016). But it has a problem of local minima, shown in Figure 2.8. The robot, the obstacle and the target are located in a straight line. The obstacle generates a repulsive force that prevent the robot moving near to the obstacle, while the attractive force generated by the target make the robot move forwards. Consequently, the robot is trapped in the local minima and cannot reach its target position. This is due to the balance between the attractive force and the repulsive force.



(a) Workspace.

(b) Total potential field.

(c) Attractive potential field.

(d) Repulsive potential field.

Figure 2.7: Illustration of the potential field given a robot's workspace.

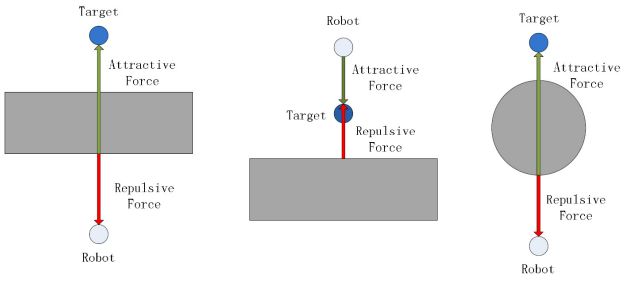


Figure 2.8: Local minima. Courtesy: Li et al. (2012).

Related works

The concept of APF is first introduced in Khatib (1986), where this method is applied to solve the problem of real-time manipulator control.

Traditional APF method can also be used for mobile robot navigation, presented in Tan et al. (2010). Using a 2D laser range finder, the robot obtains the information of its workspace. Then the robot uses a hybrid localization method and a weight average method to find optimal pose according to an error model. The APF method is used for path planning that satisfies certain optimization criteria.

In Huang (2009), the conventional potential function is used, while it is extended for direction and speed planning for a mobile robot to track a moving target among obstacles. The direction and the speed of the robot are derived as functions of the velocities and positions of the robot, the target, and the obstacles.

The traditional definition of potential field can be easily modified in order to embed more information, which is an advantage of the APF method. In Ge and Cui (2002), the attractive potential is a function of the relative position and velocity between the target and the robot. Similarly, the repulsive potential is also defined as a function of relative position and velocity of the robot with respect to the obstacles. To construct the potential method, these definitions require that the information of targets and obstacles is either known or can be measured online by sensors. The simulation studies and hardware experiments using omnidirectional mobile robots demonstrates the effectiveness of this method in dynamic environments.

The potential function is further extended in Yin and Yin (2008), where acceleration is also included. The attractive potential function therefore is a function of relative position, velocity, and acceleration between the robot and its target. The repulsive potential function is a function of relative position, velocity, and acceleration between the robot and obstacles.

In Lyu and Yin (2017), the potential function is modified to solve path planning problem with collision avoidance. The proposed APF method can meet requirements of COLREGs rules and the ship is able to track a dynamic target. The modified APF method finds a safe path for the own ship in restricted waters where there are several static and dynamic obstacles. And the simulations take into account the manoeuvre capability of ships, which results in a smooth path.

Another advantages of the APF method is that it can be easily combined with other methods, such as optimal control. In Chen et al. (2016), to achieve motion control of UAV, the control force is combined with potential force generated by the APF method for the UAV path planning problem. The optimization problem is formed in order to minimize the control force, i.e., the cost of energy. Using optimal control, the length of the path is shorter and smoother than that using traditional method. The APF method can be also combined with different heuristics methods, e.g. ant colony optimization algorithm in Zhao and Yi (2006).

In Chiang et al. (2015), an integrated method named path-guided Artificial Potential Field with Stochastic Reachable Sets (Path-Guided APF-SR) including sampling-based technique and APF is proposed. The sampling-based technique is used to find a path to avoid static obstacles, while APF is implemented to avoid moving obstacles. This method has low computation cost and works flexible in crowded environments with various moving obstacles.

The problem of artificial potential field method is that the path might not be optimal and there may exist local minima. Possible ways to address these problems are presented in Li et al. (2012), where the

Regression Search Method is applied to optimize the trajectory. The problem of local minima is solved by changing the direction of repulsive force, shown in Figure 2.9. Other method such as Wall Following (Yun and Tan, 1997) can also be used to solve the problem of local minima.

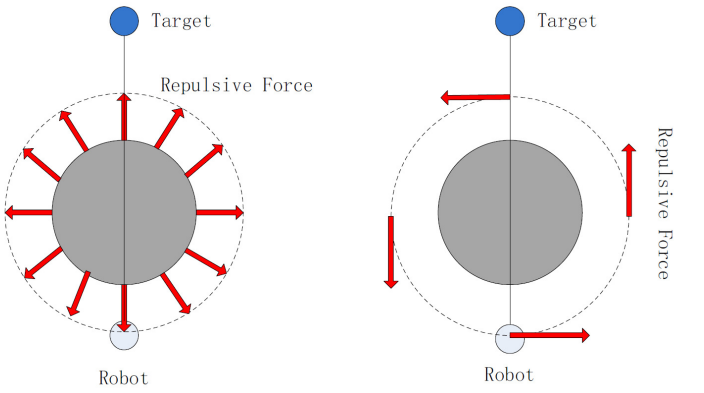


Figure 2.9: Repulsive field of circle obstacle. Left: original direction. Right: changed direction. Courtesy: Li et al. (2012).

Navigation function method

As mentioned before, the problem of the APF method is local minima. The navigation function approach, which is a potential field based method introduced in Koditschek and Rimon (1990) and Rimon and Koditschek (1992), is a solution to the local minima. In the navigation function approach, these undesired local minima can be transformed into saddle points that have regions of attraction of measure zero.

The information of the robot's, target's and obstacles' positions are used to construct the vector field, while the target is usually the origin of the planar workspace. The method has almost global convergence with good convergence rate and collision avoidance.

In the navigation function method, the workspace is usually ball-shaped and the robot is located inside the ball-shaped workspace. The ball-shaped workspace is not a special case since any star-shaped region can be diffeomorphically transformed into a sphere (Rimon and Koditschek, 1992). Therefore, each obstacle are considered to be described by a sphere, and they are all assumend isolated. The boundary of the robot's workspace is also considered as an obstacle. An example of the potential generated by a navigation function is shown in Figure 2.10.

By following the negated direction of the vector field, this will converge to the origin of the workspace described by a closed ball from almost all initial conditions. Strictly global navigation is not possible (Koditschek and Rimon, 1990). There is at least one saddle point for each obstacle in the generated vector field. And paths starting from some initial positions will end up in these saddle points. In practice, however, this is not an important issue since these initial conditions have zero measure and they are rare (Roussos et al., 2010). Hence, in the navigation function based approaches it is common to assume that there are not saddle points that can be reached.

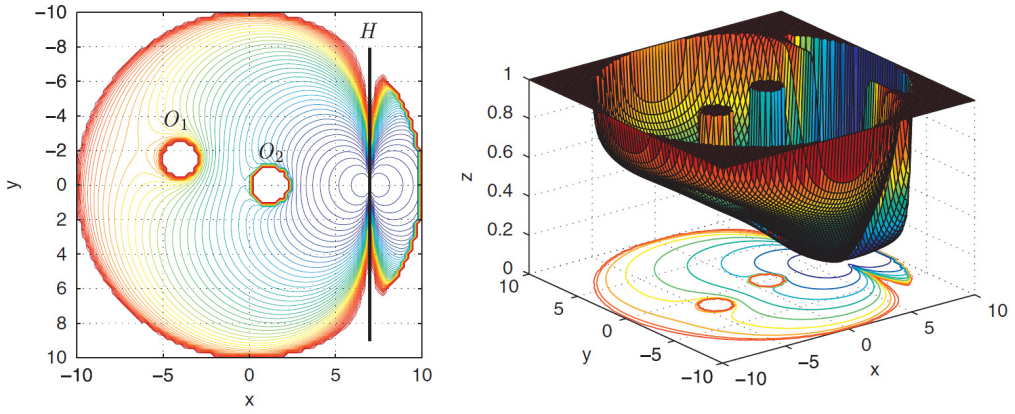


Figure 2.10: An example of a potential field generated by a navigation function in a 2D workspace with two obstacles O_1 and O_2. The target is at $[7\ 0]$ orientation $\psi_d = 0$. The desired orientation is achieved by a nonholonomic obstacle H, which is the line $x = 7$. Courtesy: Roussos et al. (2010).

Related works

The navigation function method can be applied to single-agent systems and multi-agent systems in a 2D or 3D environment. Related works are Valbuena and Tanner (2012), Kallem et al. (2011), Tanner and Kyriakopoulos (2000), Tanner et al. (2001), Tanner et al. (2003), Urakubo et al. (2004), Loizu et al. (2004), Rahmani et al. (2008), Roussos et al. (2010), Roelofsen et al. (2015), and Dimarogonas et al. (2006). These works specifically explore the area of control design and motion planning for systems with nonholonomic constraints.

In Tanner and Kyriakopoulos (2000), a discontinuous feedback control and a navigation function are applied to mobile manipulators. The proposed method not only gives obstacle avoidance but also the avoidance of the manipulator singular configurations.

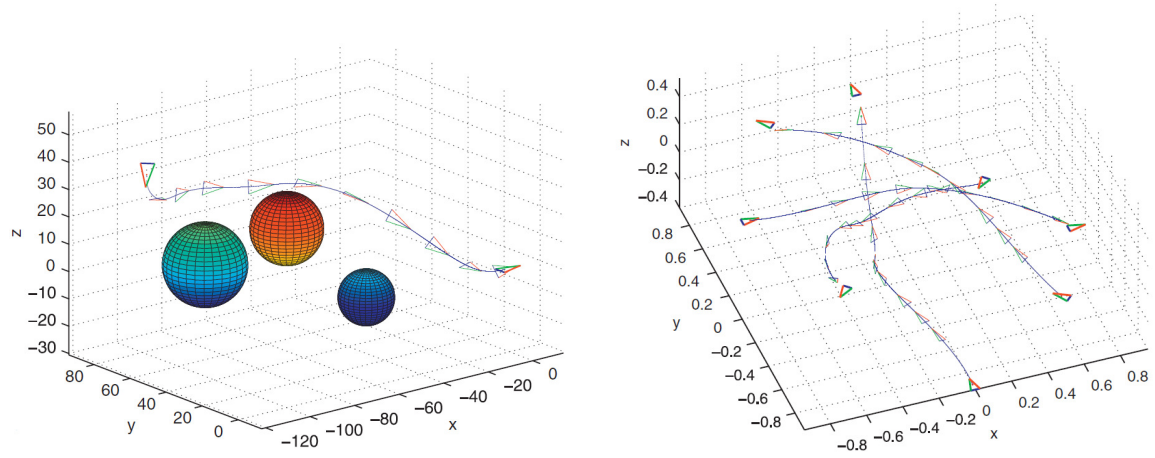
Based on Tanner and Kyriakopoulos (2000), Tanner et al. (2001) redesign the control law to improve convergence rates and modify the admissible workspace by taking the robot volume into account. This is the first complete methodology for real-time nonholonomic navigation amongst obstacles.

In Valbuena and Tanner (2012), the navigation function is combined with motion planning with nonholonomic control for nonholonomic mobile robots moving on the planar workspace. Their approach allows the mobile robot to reach its goal position. Also, the orientation of the robot is well controlled. The problems in their method are nonuniform convergence rate and sensitivity to implementation limitations such that the robot overshoots its goal position.

In Tanner et al. (2003), the navigation function is applied to multiple nonholonomic mobile manipulators for handling of deformable material. The method yeilds asymptotic convengence rate, collision avoidance and nonholonomic navigation, motion coordination for the multi-robot system, boundedness of object deformations, and singularity avoidance for the manipulator mechanisms. Navigation for a multi-robot system using the navigation function method which has guaranteed convergence rate and collision avoidance is also present in Loizu et al. (2004). In Rahmani et al. (2008), a deconfliction algorithm for the multi-vehicle system based on an appropriate navigation function is proposed. In Dimarogonas et al. (2006), the navigation function method is used to solve the problem of decentralized navigation.

The aforementioned research results are applied in two-dimensional workspace, such as ground vehicles or aircraft flying at constant attitude. The navigation function applied to three-dimension problem

is present in Roussos et al. (2010), shown in Figure 2.11. Similar to the methods in Tanner and Kyriakopoulos (2000) and Tanner et al. (2001), the method offers an advantage: the integral lines of the resulting potential field are all tangent to the desired orientation at the destination. This means that in-place rotation is not needed. In the single agent case, the agent is navigated to its target following a collision-free path. In the multi-agent case, the distributed control scheme steers the agents towards their targets and away from collisions with each other. In Roelofsen et al. (2015), the navigation function method is applied to a multi-agent UAV system in a real-time environment.



(a) Navigation for single agent

(b) Navigation for four agents

Figure 2.11: The navigation function for three-dimension problem. Courtesy: Roussos et al. (2010).

Stream function method

Stream function, also called potential flow, is derived from hydrodynamics. It is a subset of the APF method, since it generates a vector field. The stream function is inspired by the fluid flowing around any obstacles in its way. Streamlines, which are the trajectory of fluid particles, are collision-free paths. Hence, to model the flow based on potential flow theory enables us to find safe path for the robot.

The stream function is fast and efficient. It is able to generate smoother path compared to the APF method, since it models the flow of fluid. The stream function do not suffer from local minima. Instead, it has the problem of stagnation points, which can be addressed by adding a time-varying complex potential of the vortex, using Particle Swarm Optimization or using other effective techniques. In addition, it is not guaranteed to find optimal path.

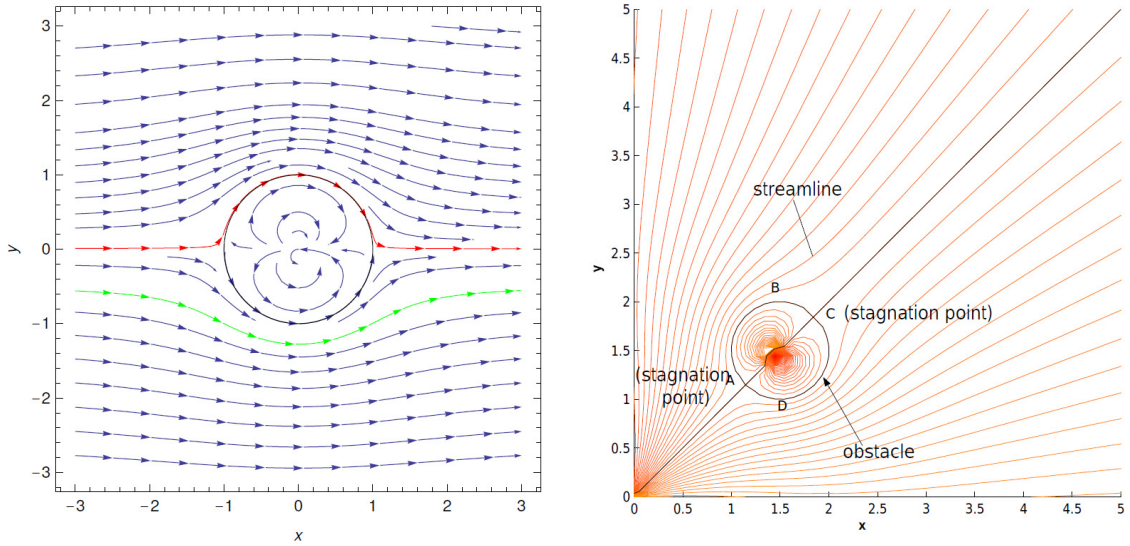
Based on the observation that fluid flows around a circular cylinder, the stream function is obtained from the closed form velocity equations for incompressible fluid flow. For detailed information, please refer to Milne-Thomson (1996), Currie (2016) and Axler et al. (2013).

In a 2D workspace, the stream function for a circular obstacle in a uniform flow is shown in Figure 2.12a. An example of feasible path is the green path.

The target is usually represented by a sink flow, at which the fluid particles end. The circular obstacle in a sink flow is shown in Figure 2.12b.

In order to model a circular obstacle, the object, called doublet in potential theory is created. The effect of doublet depends on the position of the robot and its target. Therefore, compared to the APF method, a stream function is able to generate smoother paths. The boundary condition, i.e., the fluid cannot transverse walls and obstacles, also help to generate smooth path, which is suitable to aircraft-like vehicles (Waydo and Murray, 2003).

If there are multiple obstacles, Laplace's equation ($\nabla^2\phi_{pf} = 0$, where ϕ_{pf} is the velocity potential) with multiple boundary conditions must be solved. This is analytically impossible. Instead, a method called addition and thresholding can be used (Waydo and Murray, 2003; Sullivan et al., 2003). This will be explained in Chapter 5.



(a) Circular obstacle in a uniform flow. Courtesy: Pedersen and Fossen (2012).

(b) Circular obstacle in a sink flow. Courtesy: Ye et al. (2005).

Figure 2.12: Circular obstacle in different flows.

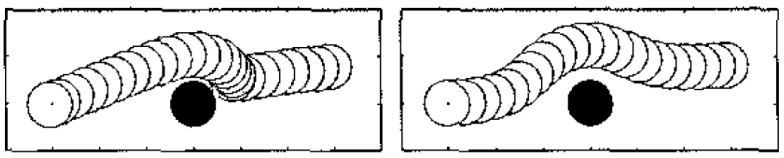


Figure 2.13: Comparison between paths generated by the APF method (left) and stream function (right). Courtesy: Waydo and Murray (2003).

Related works

In Waydo and Murray (2003), a path planner based on a stream function is used for the robot to perform complex behavior, including formation flying, target tracking, and interception. They conclude that the stream function is fast, efficient and adapt to changing condition, which can be used in real-time path planning.

Stream functions do not have the problem of local minima. However, the problem of stream function is that it has stagnation points, where the velocity of fluid is zero. In Figure 2.12b, there are two stagnation points. The stagnation points are located at the places where the streamline first reaches or starts to leave

the obstacle.

One technique to solve this problem is present in Ye et al. (2005). According on fluid dynamics, adding the complex potential of vortex to a circular obstacle in certain types of flow, it will change the positions of stagnation points. An example is showed in Figure 2.14. If the strength of vortex is time-varying, the positions of stagnation points will change with time. Therefore once a robot gets onto a stagnation point, at next time step when the stagnation point updates its position, the robot will be out of the stagnation point. The problem is successfully solved, and a stream function is implemented for coordinated control of swarms navigating through multiple obstacles. In Hu et al. (2007), the Particle Swarm Optimization is used to tackle the stagnation problem.

Another problem is that correct solutions are not guaranteed in the case of multiple obstacles (Sullivan et al., 2003). An example is presented in Figure 2.15. In addition, Figure 2.15 points out that the path length is not guaranteed to be optimal, i.e., it can be arbitrarily long.

Other related works are Kim et al. (2011), Wang and Ju (2015), etc.

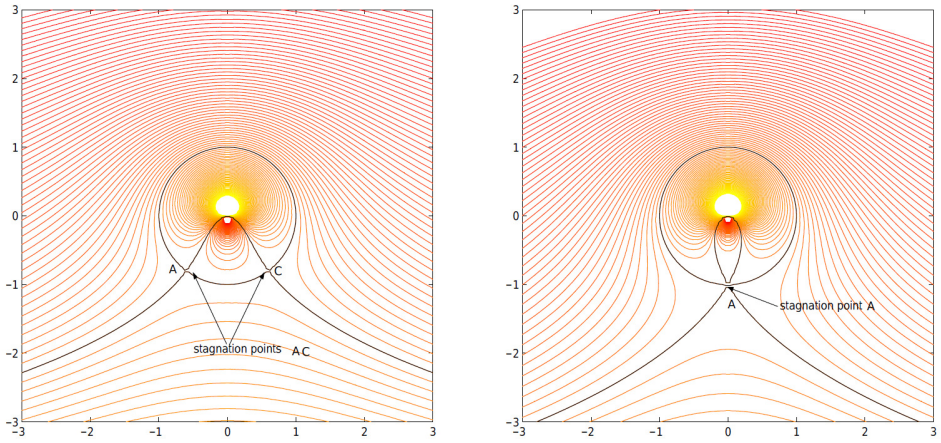


Figure 2.14: Stagnation points shifting. Courtesy: Ye et al. (2005).

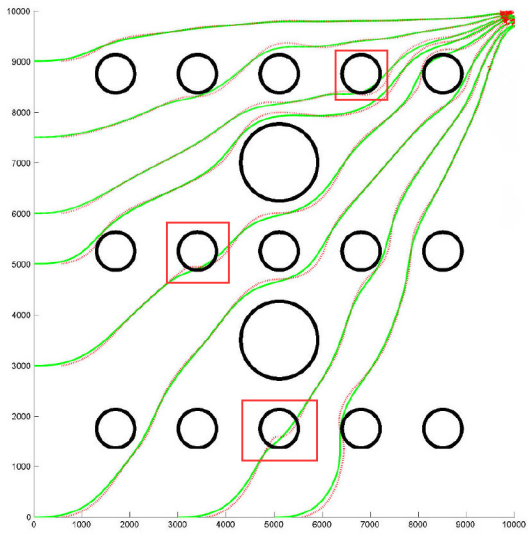


Figure 2.15: Trajectory generated by stream function in the case of multiple obstacles.The red blocks indicate collision. Courtesy: Pedersen and Fossen (2012).

2.3 Path generation

One way to generate a path is to use the Bézier curves. A Bézier curve is a parametric curve, which is named after Pierre Bézier, who used it in the 1960s for designing curves for the bodywork of Renault cars.

There are various works using Bézier curves for path generation and optimization. In Gao et al. (2018a), the trajectory of a UAV is represented as piecewise Bézier curve. The trajectory is further optimized by formulating a quadratic programming problem. Based on Gao et al. (2018a), Gao et al. (2018b) improve the method by minimizing total trajectory time.

Another method is to utilize the B-spline. A B-spline is a generalization of the Bézier curves. A B-spline can be obtained by writing spline curves in terms of certain basis functions.

In Usenko et al. (2017), the problem of local trajectory replanning is solved by formulating a B-spline optimization problem. Similarly, Zhou et al. (2019) improve the smoothness and clearance of the trajectory by a B-spline optimization.

2.4 The maneuvering problem

The maneuvering problem is introduced in Skjetne (2005). For a system with output $\boldsymbol{y} \in \mathbb{R}^m$, we consider a desired path defined by a set

$$\mathcal{P} := \{\boldsymbol{y} \in \mathbb{R}^m : \exists s \in \mathbb{R}\ s.t.\ \boldsymbol{y} = \boldsymbol{y}_d(s)\} \tag{2.9}$$

where y_d is continuously parametrized by a path variable s.

Given the desired path (2.9), the maneuvering problem has two tasks, i.e., the geometric task and the dynamic task:

- Geometric Task: Force the output of the given system y to converge to the desired path $y_d(s)$.
- Dynamic Task: Satisfy one of the following tasks:
  - Time Assignment: Force the path variable θ to converge to a desired time signal $v_t(t)$.
  - Speed Assignment: Force the path speed $\dot{s}$ to converge to a desired speed $v_s(s,t)$.
  - Acceleration Assignment: Force the path acceleration $\ddot{s}$ to converge to a desired acceleration $v_a(\dot{s}, s, t)$.

The maneuvering problem defined above only highlights the goal of convergence to the path and the chosen dynamic assignment. For the control design, additional conditions should be considered, such as feasibility of the geometric and dynamic tasks and boundedness of the system states.

The maneuvering problem is a convenient problem statement that can be used to solve many problems. Many nonlinear control design methods, including backstepping, sliding mode control, etc., can be used to solve the maneuvering problem. In the control design, the geometric task will be solved first, then an suitable update law, which bridges the geometric task with the dynamic task, will be chosen to complete the control design. Examples of such update laws are gradient update laws, unit-tangent gradient update law, etc.

2.5 COLREGs rules

The International Regulations for Preventing Collisions at Sea 1972 (COLREGs) were published by the International Maritime Organization (IMO). Human operators should follow COLREGs rules when operating all kinds of vessels or watercraft. For autonomous ships without on-board human operators to be lawfully operated at sea, these rules must still be followed. This is to avoid unexpected and incorrect actions that may cause confusion and potential collision among other vessels.

The COLREGs include 41 rules that are divided into six sections(Commandant, 1999; IMO, 2020):

- Part A - General (Rules 1-3): Outlining the application, responsibility and general definitions of the regulations.
- Part B- Steering and Sailing (Rules 4-19): Part B consists of three sections, i.e., the conduct of vessels in any condition of visibility, the conduct of vessels in sight of one another and the conduct of vessels in restricted visibility.
- Part C - Lights and Shapes (Rules 20-31): Covers light requirements for various vessels.
- Part D - Sound and Light Signals (Rules 32-37): Covers definitions of sound and light signals and protocols for the use of these signals.
- Part E - Exemptions (Rule 38): States that ships which comply with the 1960 Collision Regulations and were built or already under construction when the 1972 Collision Regulations entered into force may be exempted from some requirements for light and sound signals for specified periods.
- Part F - Verification of compliance with the provisions of the Convention (Rules 39-41).

The part of the Steering and Sailing Rules is probably the most relevant section for fully autonomous ships (Campbell et al., 2012). Some of these main rules are listed below.

- Rule 13 – Overtaking: *Any vessel overtaking any other shall keep out of the way of the vessel being overtaken.*
- Rule 14 – Head-on Situation: *When two power-driven vessels are meeting on reciprocal or 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.* See Figure 2.16.
- Rule 15 – Crossing Situation: *When two power-driven vessels are crossing so as to involve risk of collision, 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.* See Figure 2.17.
- Rule 16 – Action by give-way vessel: *Every vessel which is directed to keep clear of another vessel shall, so far as possible, take early and substantial action to keep well clear.*
- Rule 17 – Action by stand-on vessel: *Where one of two vessels is to keep out of the way, the other shall keep her course and speed. The latter vessel may however take action to avoid collision by her manoeuvre alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these rules.*

Rule 18 specifies the responsibilities between vessels. An example of Rule 18 is that a power-driven vessel underway shall keep out of the way of a vessel not under command.

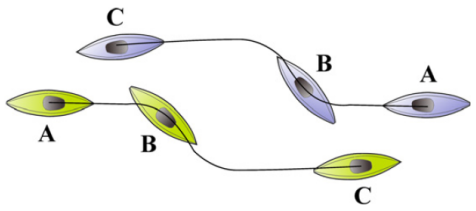


Figure 2.16: Head-on situation on COLREGs. Both of the ships start from point A and end at point C. Courtesy: Campbell et al. (2012).

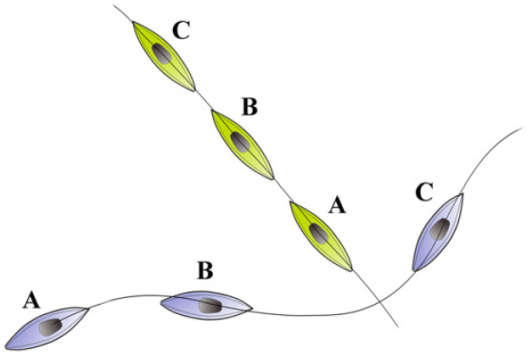


Figure 2.17: Crossing situation on COLREGs. Both of the ships start from point A and end at point C. Courtesy: Campbell et al. (2012).

In recent years, part of COLREGs rules has been incorporated in finding a rule-compliant collision-free solution. Relevant researches are Perera et al. (2012), Praczyk (2015), Kuwata et al. (2013), Johansen et al. (2016), Lyu and Yin (2017) and Lyu and Yin (2019). Some rules are frequently used, such as Rule 13–19. Despite the development of rule-compliant collision-free methods, incorporating all regulation rules still need more effort in the future (He et al., 2017; Woerner et al., 2019).

Chapter 3

Problem formulation

In this thesis, we consider a fully actuated marine surface vehicle moving at low speed. The marine vehicle is working in a 2D plane, and only three DOFs are considered, i.e., surge, sway, and yaw. There are several dynamic obstacles in the workspace. The environmental loads caused by waves, currents and wind are ignored. It is also assumed that the information of the workspace is known, i.e., the positions and the velocities of target(s) and obstacles, and therefore, sensors are not included. The problem to solve is to find a collision-free path with COLREGs compliance that guides the marine vehicle reach its target(s) from its initial position.

3.1 System description

The guidance, navigation, and control (GNC) system for marine robots is composed of three subsystems (Fossen, 2011), illustrated in Figure 3.1.

- The guidance system: continuously calculates the desired position, velocity and attitude of a marine craft to be used by the control system.
- The navigation system: determines a marine craft's position, course, and distance traveled. In some cases velocity and acceleration are determined as well.
- The control system: continuously calculates the desired forces and moments applied to a marine craft motivated by a certain control objective.

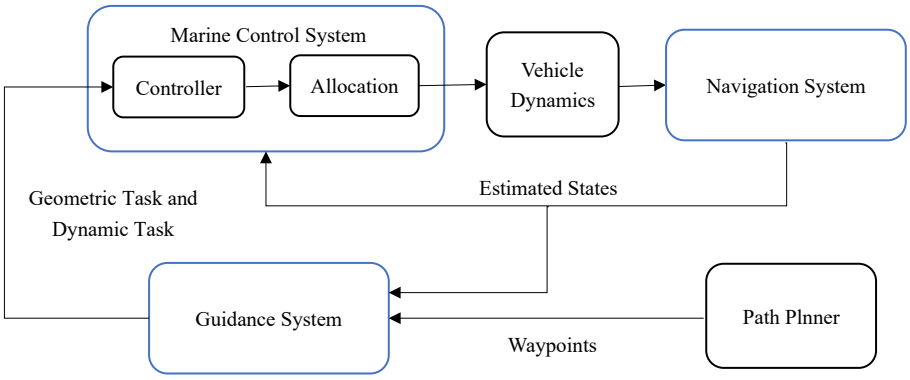


Figure 3.1: The GNC system for a marine craft.

In this thesis, the functions of each subsystem and module are:

- Path Planner: determines discrete waypoints to be used by the Guidance System to generate desired position, heading and path speed.
- Navigation System is not considered as we assume that the workspace is known.
- Guidance System: given waypoints from Path Planner module, find the desired position, heading and path speed, i.e., the Geometric Task and the Dynamic Task.
- Controller: calculates the desired forces and moments to satisfy the geometric task and the dynamic task.
- Allocation: distributes the desired forces and moments to each thrusters.
- Vehicle Dynamics: models the dynamics of a marine robot using a control design model.

3.2 Problem statement

3.2.1 Stepwise path planning

Given the positions of target(s) and obstacles, the problem of path planning is to find a set of discrete waypoints to guide a marine robot reach its target position from its starting position with collision avoidance.

Suitable path planning methods should be chosen as a guidance model among the relevant methods studied. Generally, the information incorporated in a guidance model is the position information of target(s) and obstacles. This inspires a possible way to obtain better performance of those guidance models: to incorporate other relevant information, such as the velocity of the obstacles. It is assumed that the positions and velocities of the obstacles are known.

Consider the number of obstacles in the workspace is N_o and only one target. The workspcae information includes the the positions of the marine robot $[x\ y]^\top$, the target $[x_t\ y_t]^\top$ and the obstacles $[x_i\ y_i]^\top$ where $i \in \{1, 2, ..., N_o\}$. Given the workspace information, the task of path planner module is to find a next waypoint $[x_{next}\ y_{next}]^\top$. Since the obstacles or the target can move, the path planner module needs to update the workspace information and finds a next waypoint using latest information. Once the marine robot reaches the next waypoint, the path planner module will update the workspace information and determine a new waypoint. This procedure will be done recursively until the marine robot gets to its target.

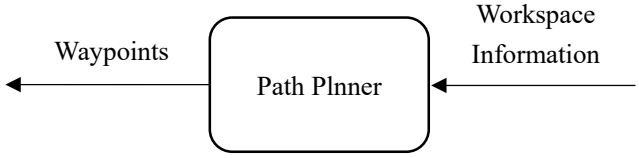


Figure 3.2: Path planner module.

3.2.2 Path generation

The problem of path generation is to generate a feasible path through given waypoints, which are determined by a chosen guidance model. A path can be a straight line or a curve, but the smooth path with continuous path derivatives at waypoints is preferred.

In the guidance system, the position of the marine robot $[x\ y]^\top$ from workspace information and the next waypoint $[x_{next}\ y_{next}]^\top$ are used. The task of the guidance system is to find a path, represented by Equation (3.1), from the position of the marine robot $[x\ y]^\top$ to the next waypoint $[x_{next}\ y_{next}]^\top$, based on the geometric task and the dynamic task.

A path in 2D plane with path variable s can be represented by

$$p_d(s) = [x_d(s), y_d(s)]^\top, \exists s \in \mathbb{R}. \tag{3.1}$$

For the marine robot, the desired heading is defined as

$$\psi_d(s) = atan(\frac{y_d^s(s)}{x_d^s(s)}). \tag{3.2}$$

Besides, desired path speed $v_s(s,t)$ should be specified. For simplicity, $v_s(s,t)$ can be a constant.

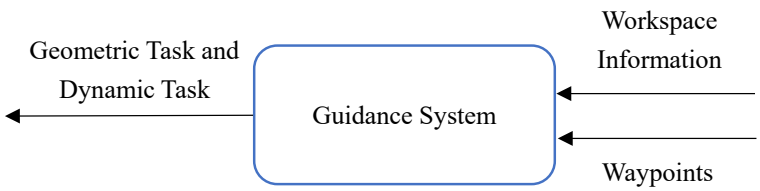


Figure 3.3: Guidance system module.

3.2.3 Control objective

The control objective is constructed as a maneuvering problem, as introduced in Skjetne (2005). This means the control objective is to satisfy the geometric task and the dynamic task. The dynamic task is chosen as the speed assignment. The control objectives then are:

- Geometric Task: Force the output of the given system $\boldsymbol{\eta}(t)$ to converge to the desired pose $\boldsymbol{\eta}_d(s(t))$,

$$\lim_{t\to\infty} |\boldsymbol{\eta}(t) - \boldsymbol{\eta}_d(s(t))| = 0. \tag{3.3}$$

- Speed Assignment: Force the path speed $\dot{s}$ to converge to a desired speed $v_s(s,t)$,

$$\lim_{t\to\infty} |\dot{s}(t) - v_s(s(t),t)| = 0. \tag{3.4}$$

The output of a marine robot working in horizontal plane is its North-East position and heading:

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}. \tag{3.5}$$

The desired path $\eta_d(s(t))$, calculated by the guidance system, is given as

$$\boldsymbol{\eta}_d(s) = \begin{bmatrix} p_d(s) \\ \psi_d(s) \end{bmatrix} = \begin{bmatrix} x_d(s) \\ y_d(s) \\ \psi_d(s) \end{bmatrix}, \quad s \in \mathbb{R}. \tag{3.6}$$

The maneuvering control objective is to find a control input $\boldsymbol{\tau} = [X\ Y\ N]^\top$ using suitable control design method such that

$$\begin{aligned} \boldsymbol{\eta}(t) &= \boldsymbol{\eta}_d(s(t)),\ as\ t \to \infty. \\ \dot{s}(t) &= v_s(s(t), t),\ as\ t \to \infty. \end{aligned} \tag{3.7}$$

Given a commanded control input $\boldsymbol{\tau}$, the task of allocation is to assigns a normalized force vector $\mathbf{u} = [u_1\ u_2\ ..\ u_r]^\top$ to each individual thruster, where r is the number of thrusters.

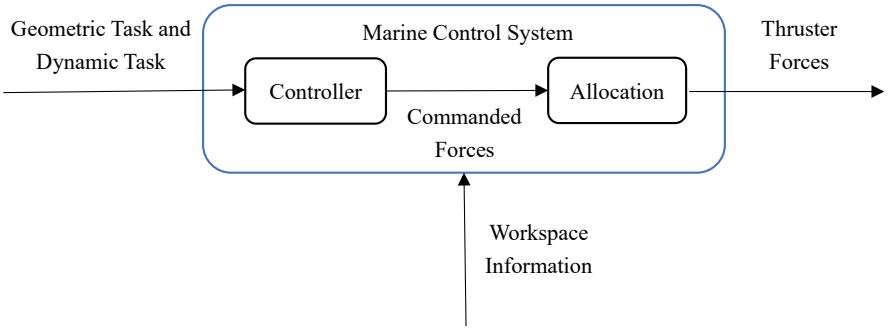


Figure 3.4: Marine control system module.

3.2.4 COLREGs compliance

To assess the effectiveness of the guidance models, three rules from COLERGs are selected:

- Rule 13 – Overtaking: *Any vessel overtaking any other shall keep out of the way of the vessel being overtaken.*
- Rule 14 – Head-on Situation: *When two power-driven vessels are meeting on reciprocal or 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 so as to involve risk of collision, 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.*

3.2.5 Assumptions

In this thesis, some simplifications and assumptions are made:

- Only computer simulations are performed.
- It is assumed that the workspace of the marine robot is a 2D plane.
- It is assumed that the dynamics of the marine robot is simplified.
- The volume of the marine robot is ignored.
- It is assumed that the information of the robot's workspace is known, i.e., the positions and velocities of the target(s) and the obstacles.
- Only selected COLREGs rules are used to assess the resulting performance.

Chapter 4

Modeling

4.1 Notation

According to SNAME (1950), a marine robot can be described by six motion variables in 6 DOFs. Those variables are defined in Table 4.1 and illustrated in Figure 4.1.

Table 4.1: The notation of SNAME (1950) for marine vessels.

DOF	Definition	Forces and moments	Linear and angular velocities	Positions and Euler angle
1	motions in the x-direction (surge)	X	u	x
2	motions in the y-direction (sway)	Y	v	y
3	motions in the z-direction (heave)	Z	w	z
4	rotation about the x-axis (roll)	K	p	ϕ
5	rotation about the y-axis (pitch)	M	q	θ
6	rotation about the z-axis (yaw)	N	r	ψ

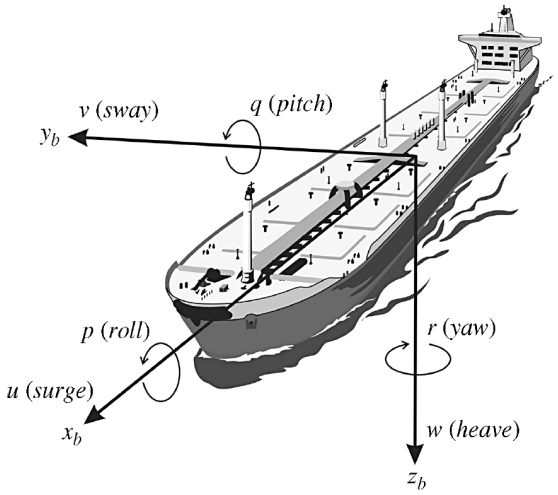


Figure 4.1: Illustration of motion variables for marine vehicles. Courtesy: Fossen (2011).

4.2 Reference frame

Two reference frames are used in this thesis, namely the North-East-Down (NED) frame and the body-fixed reference (BODY) frame, see Fossen (2011) for further explanation:

- The NED frame is denoted $\{n\}$ = (x, y, z), where x is the axis pointing towards ture North, y is the axis pointing towards ture East, z is the axis pointing dpwnwards normal to Earth's surface. The origin of the NED frame is defined relative to the Earth's reference ellipsoid (Mularie, 2000). This is the coordinate system we refer to in our everyday life and it is used for local navigation.
- The BODY frame is denoted $\{b\}$ = (x_b, y_b, z_b), where x_b is the longitudinal axis (directed from aft to fore), y_b is the transversal axis (directed to starboard), and z_b is the normal axis (directed from top to bottom). This is a moving frame where the origin is fixed to the marine robot. For marine robots, the body axes are chosen to coincide with the principal axes of inertia.

4.3 Simulation model

The simulation model is the most accurate description of a system. This model includes features that affect model accuracy, even though these features are not used for control and observer design.

A 6-DOF high-fidelity simulation model for a marine robot can be given as (Fossen, 2011)

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \mathbf{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})\boldsymbol{\nu} \\ \mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{g}(\boldsymbol{\eta}) + \mathbf{g}_0 &= \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \end{aligned}, \tag{4.1}$$

where $\boldsymbol{\eta}$ is position and Euler angle in NED frame, $\mathbf{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})$ is rotation matrix from BODY to NED frame, $\boldsymbol{\nu}$ is linear and angular velocities in BODY frame, $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ is relative velocity vector between the marine robot velocity vector $\boldsymbol{\nu}$ and the current velocity vector $\boldsymbol{\nu}_c$, $\mathbf{M}$ is the system mass matrix, $\mathbf{C}(\boldsymbol{\nu}_r)$ is the coriolis and centripetal matrices, $\mathbf{D}(\boldsymbol{\nu}_r)$ is the damping matrix, $\mathbf{g}(\boldsymbol{\eta})$ and $\mathbf{g}_0$ are the restoring force vectors, $\boldsymbol{\tau}$, $\boldsymbol{\tau}_{wind}$, and $\boldsymbol{\tau}_{wave}$ are the propulsion, wind, and wave forces and moments.

4.4 Control design model

The control design model is used for the design of motion control system. Compared to the simulation model, the control design model is simplified. For marine surface vehicles, it is traditional that only horizontal motion, i.e., surge, sway, and yaw motion are considered.

The control design model chosen in this thesis is CSE1 in the Marine Cybernetics Laboratory (MCLab), Department of Marine Technology, NTNU. The model is valid for low-speed scenario. The proposed control design model is given as

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \mathbf{R}(\psi)\boldsymbol{\nu} \\ \mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} &= \boldsymbol{\tau} \end{aligned}, \tag{4.2}$$

where

- $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are the ship pose and velocity vector, given as

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ r \end{bmatrix}. \tag{4.3}$$

- $\boldsymbol{\tau}$ is the thrust forces and moment vector, given as

$$\boldsymbol{\tau} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix}. \tag{4.4}$$

- $\mathbf{R}(\psi)$ is the 3-DOF rotation matrix, given as

$$\mathbf{R}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \dot{\mathbf{R}}(\psi) = r\mathbf{R}(\psi)\mathbf{S} = r\mathbf{R}(\psi) \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.5}$$

- $\mathbf{M}$ is the inertia matrix, given as

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix}. \tag{4.6}$$

- $\mathbf{C}(\boldsymbol{\nu})$ is the coriolis and centripetal matrix, given as

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & (-mx_g + Y_{\dot{r}})r + (-m + Y_{\dot{v}})v \\ 0 & 0 & (m - X_{\dot{u}})u \\ (mx_g - Y_{\dot{r}})r + (m - Y_{\dot{v}})v & (-m + X_{\dot{u}})u & 0 \end{bmatrix}. \tag{4.7}$$

- $\mathbf{D}(\boldsymbol{\nu})$ is the damping matrix, for simplicity the damping matrix is considered linear, given as

$$\mathbf{D}(\boldsymbol{\nu}) = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}. \tag{4.8}$$

4.5 Thruster dynamics

For a marine vehicle equipped with r in 3 DOFs, the relation between commanded forces and moments and thruster forces is given as

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\alpha})\mathbf{K}\mathbf{u}, \tag{4.9}$$

where $\boldsymbol{\tau} = [X\ Y\ N]^\top$, $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ ..\ \alpha_p]^\top \in \mathbb{R}^p$ is a vector of azimuth angles, $\mathbf{B}(\boldsymbol{\alpha}) \in \mathbb{R}^{3\times r}$ is configuration matrix which describes the geometry and location of the thrusters, $\mathbf{K} \in \mathbb{R}^{r\times r}$ is the diagonal force coefficient matrix, and $\mathbf{u} = [u_1\ u_2\ ..\ u_r]^\top \in \mathbb{R}^r$ is the thruster forces vector. To calculate the thruster forces vector, numerous methods can be found in Johansen and Fossen (2013).

For a azimuth thruster, the i^{th} coloum of $\mathbf{B}(\boldsymbol{\alpha})$ is

$$\boldsymbol{t_i} = \begin{bmatrix} cos(\alpha_i) \\ sin(\alpha_i) \\ l_{xi}sin(\alpha_i) - l_{yi}cos(\alpha_i) \end{bmatrix}, \tag{4.10}$$

where $[l_{xi}\ l_{yi}]$ is the position of the thruster.

For a main propeller, the i^{th} coloum of $\mathbf{B}(\boldsymbol{\alpha})$ is

$$\boldsymbol{t_i} = \begin{bmatrix} 1 \\ 0 \\ -l_{yi} \end{bmatrix}. \tag{4.11}$$

For a tunnel thruster, the i^{th} coloum of $\mathbf{B}(\boldsymbol{\alpha})$ is

$$\boldsymbol{t_i} = \begin{bmatrix} 0 \\ 1 \\ l_{xi} \end{bmatrix}. \tag{4.12}$$

4.6 Workspace representation

The workspace of the marine robot is represented by discrete grid points. An example is shown in Figure 4.2. The number of grid pints in the simulations will be larger than that shown in Figure 4.2, in order to represent the workspace accurately. The size of the workspace is $L_y(m) \times L_x(m)$. We place N_x and N_y equidistant points in x-axis and y-axis respectively, with $d_x = \frac{L_x}{N_x}$, and $d_y = \frac{L_y}{N_y}$, which results in $N = N_x \times N_y$ grid points.

The obstacles are represented by circles. The center of each obstacle is located at a grid point with radius r_i. An example is shown in Figure 4.2.

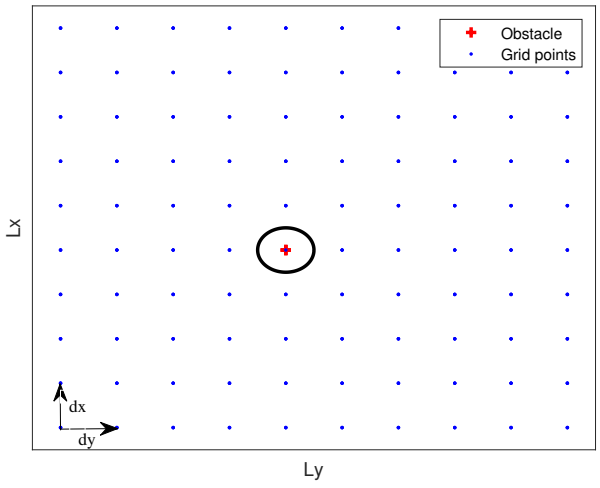


Figure 4.2: Illustration of workspace representation.

Chapter 5

Waypoint generation and path generation

In this chapter, the methods for stepwise waypoint generation and path generation will be introduced. There are two different methods to generate a next waypoint: the navigation function method (Section 5.1) and the stream function method (Section 5.2) . After a next waypoint is decided, path generation using Bézier curves (Section 5.3) will be performed, which gives a soomth path segment from the current waypoint to the chosen next waypoint.

5.1 Waypoint generation using navigation function method

5.1.1 Generating the potential field

To generate the potential field in the workspace, the following equation is used.

$$U(\mathbf{p}) = \frac{\gamma}{(\gamma^{\kappa} + \prod_{i=0}^{N_o} \beta_i(\mathbf{p}))^{\frac{1}{\kappa}}}, \tag{5.1}$$

where $\gamma = \|\mathbf{p} - \mathbf{p_t}\|^2$, $\mathbf{p}$ is the horizontal position of the marine robot, $\mathbf{p_t}$ is the position of the taret, N_o is the number of obstacles, $\beta_i(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_i\|^2$, where $i \in \{1, 2, ...N_o\}$, for obstacles that are not the boundary of the workspace, $\beta_0(\mathbf{p}) = r_0^2 - \|\mathbf{p}\|^2$ for the boundary of the workspace, r_0 is radius of the workspace, $\mathbf{p}_i$ is position of the center of ith obstacle, and κ is a positive parameter which should be sufficiently large.

From Equation (5.1), it is easy to see that the potential is zero when $\gamma = \|\mathbf{p} - \mathbf{p_t}\|^2 = 0$, which implies $\mathbf{p} = \mathbf{p_t}$. Also, $\beta_i(\mathbf{p})$ is zero when $\mathbf{p}$ is on the boundary of the obstacles, then the potential at these points is 1. Except the boundary of the workspace, $\beta_i(\mathbf{p})$, where $i \in \{1, 2, ...N_o\}$, is negative when $\mathbf{p}$ is inside the obstacles, which may give a negative or complex value of potential. In order to avoid this problem, inside the obstacles $\beta_i(\mathbf{p})$ should be 0, where the potential becomes 1.

Therefore, for $\beta_i(\mathbf{p})$ except the boundary of the workspace, where $i \in \{1, 2, ...N_o\}$, we have

$$\beta_i(\mathbf{p}) = \begin{cases} \|\mathbf{p} - \mathbf{p}_i\|^2 & , \quad \|\mathbf{p} - \mathbf{p}_i\|^2 > r_i^2 \\ 0 & , \quad \|\mathbf{p} - \mathbf{p}_i\|^2 \leq r_i^2 \end{cases}, \tag{5.2}$$

where r_i is the radius of ith obstacle.

Similarly, for $i = 0$, we have

$$\beta_0(\mathbf{p}) = \begin{cases} r_0^2 - \|\mathbf{p}\|^2 & , \quad r_0^2 > \|\mathbf{p}\|^2 \\ 0 & , \quad r_0^2 \leq \|\mathbf{p}\|^2 \end{cases} . \tag{5.3}$$

An example of the potential field generated by a navigation function is shown in Figure 5.1, with three obstacles and one target. As we expect, the largest potential value is 1 on the boundary of obstacles, and it decreases to 0 at target position.

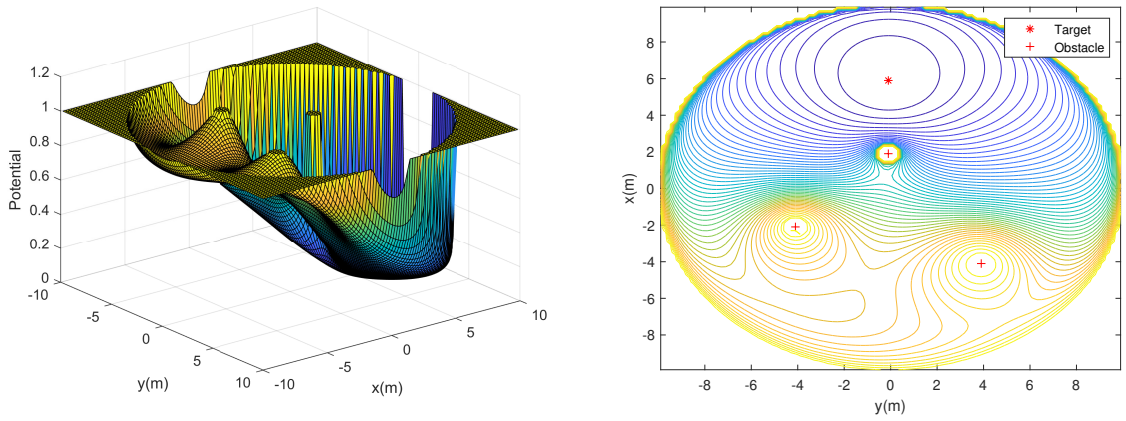


Figure 5.1: Potential field generated by the navigation function method.

5.1.2 The influence of obstacle velocity

To understand the influence of obstacle velocity on the potential field, we consider only one obstacle for simplicity. From Equation (5.1), it is clear that the potential value becomes smaller as $\beta_i(\mathbf{p})$ increases. In other words, the potential value drops as the distance to an obstacle increases. Consider a fixed point at position $\mathbf{p}$ and a moving obstacle in the workspace. The potential increases at $\mathbf{p}$ if the obstacle moves closer to the fixed point, while the potential drops if the obstacle becomes further away from the fixed point. This logic can be extended to the whole workspace.

If the velocity of the obstacle is known, we can calculate the difference of potential of the workspace between current and next sample time, i.e.

$$\delta U(\mathbf{p}) = U_{next}(\mathbf{p}) - U_{current}(\mathbf{p}), \tag{5.4}$$

where $U_{current}(\mathbf{p})$ is the potential field at current sample time, and $U_{next}(\mathbf{p})$ is the potential field at next sample time. Then this information can play a role in choosing a next waypoint, e.g. choosing a next waypoint with $\delta U(\mathbf{p}) < 0$, and preventing the robot from getting closer to an obstacle. An example is illustrated in Figure 5.2. This simple strategy can also be applied to multiple obstacles, since $\delta U(\mathbf{p})$ has a strong local effect.

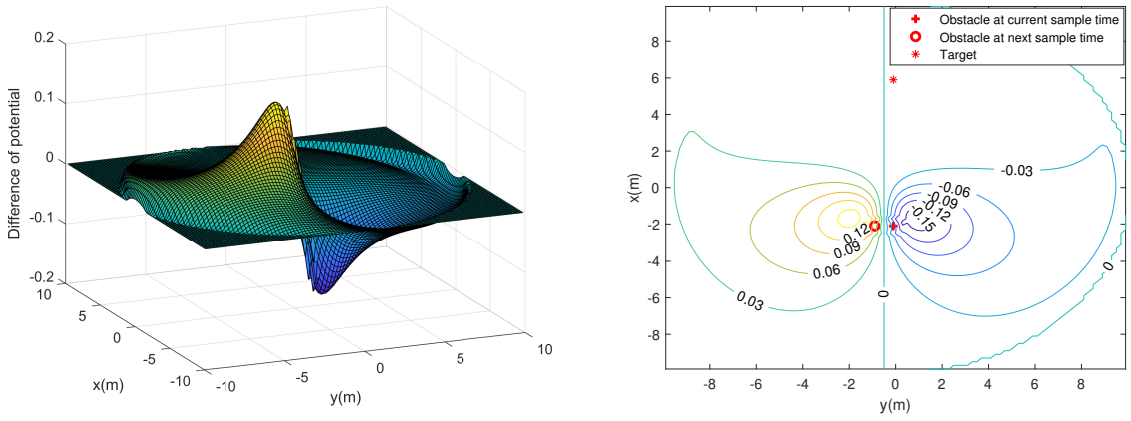


Figure 5.2: Influence of obstacle velocity on the potential field.

5.1.3 Choosing the next waypoint

Decision-making module

Under the head-on situation, the original navigation function does not force the robot to past on from the port side of a obstacle. Hence, in order to enforce CLOREGs compliance, a decision-making module should be built up to change strategy of choosing waypoint in head-on situation.

Consider we have current waypoint $\mathbf{p}_c = [x_c \ y_c]^\top$, target position $\mathbf{p}_t = [x_t \ y_t]^\top$, and the position $\mathbf{p}_i = [x_i \ y_i]^\top$ and velocity $\mathbf{v}_i = [v_{x_i} \ v_{y_i}]^\top$ of a obstacle, where $i \in \{1, 2, ...N_o\}$. The head-on situation is simplified in this thesis, shown in Figure 5.3a. We consider:

- An obstacle moving in the x direction, which implies $v_{y_i} = 0$;
- The robot is close to an obstacle in the y direction, which implies $|y_i - y_c| \leq n_h dy$, where n_h is a positive parameter;
- An obstacle is located in between the robot and the target in x direction, and the marine robot is moving from the upper part of the workspace to the lower part, which implies $x_c > x_i > x_t$;

These conditions are summarized as

$$\begin{cases} v_{y_i} & = 0 \\ |y_i - y_c| & \leq n_h dy \ , \quad i \in \{1, 2, ...N_o\} \\ (x_i - x_c)(x_i - x_t) & < 0 \end{cases} \tag{5.5}$$

If Equation (5.5) is satisfied, then the strategy for choosing waypoint should force the robot to past on

from the port side of a obstacle. The algorithm for decision making is given in Algorithm 2.

Algorithm 2: The decision-making module

```
Algorithm navigation_function_decision (p_c ,p_i ,p_t ,n_h):
head_on_flag← 1;
for all i ∈ {1, 2, ..., N_o} do
    if Equation (5.5) satisfied then
        head_on_flag← 0;
        break;
    end
end
return head_on_flag
```

Waypoint strategy

When there is no head-on situation, where $head\_on\_flag = 1$, the next waypoint is the solution of an optimization problem:

$$\mathbf{p}_{next} := \min_{\mathbf{p}_w} \quad U_{current}(\mathbf{p}_w) + \gamma_1 \left\| \mathbf{p}_w - \mathbf{p}_t \right\|$$

$$\text{s.t.} \quad \mathbf{p}_w \in \mathcal{P}_w \qquad ,$$

$$\mathcal{P}_w := \begin{cases} \{\mathbf{p} \mid \{max\{\frac{|x_c - x|}{d_x}, \frac{|y_c - y|}{d_y}\} \leq n_r\} \cap \{\delta U(\mathbf{p}_w) \leq \delta U(\mathbf{p})\}, & if \ \|\mathbf{p} - \mathbf{p}_i\| \leq R, \ i \in \{1, 2, ..., N_o\} \\ \{\mathbf{p} \mid max\{\frac{|x_c - x|}{d_x}, \frac{|y_c - y|}{d_y}\} \leq n_r\}, & otherwise \end{cases} \tag{5.6}$$

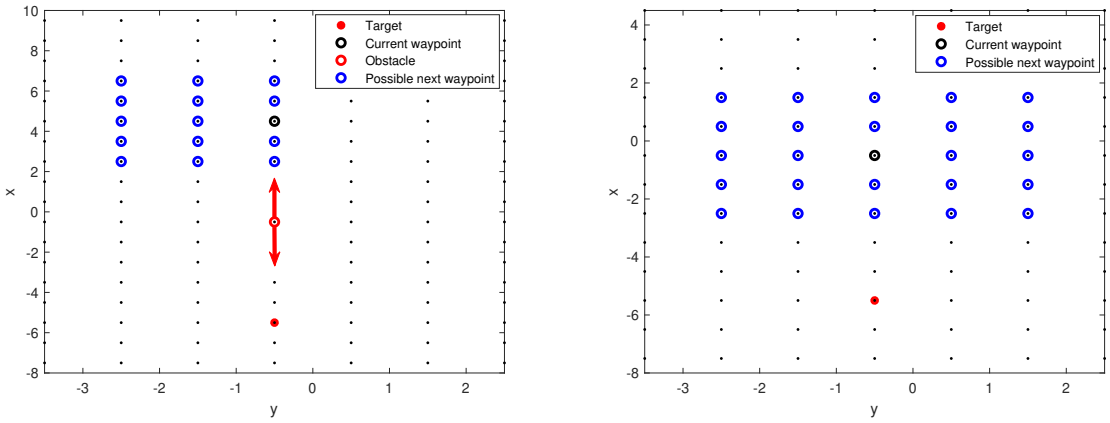
where n_r, R, and γ_1 are tuning parameters.

First we explain how to choose the set of the next possible waypoints $\mathcal{P}_w$. The set $\mathcal{P}_w$ includes those grid points within a small range of the current waypoint, which satisfies: $max\{\frac{|x_c - x|}{d_x}, \frac{|y_c - y|}{d_y}\} \leq n_r$. This is shown in Figure 5.3b. If there is a obstacle close to the current waypoint, which means $\|\mathbf{p} - \mathbf{p}_i\| \leq R$, the collision avoidance with COLREGs compliance should be adopted. This is achieved by satisfying $\delta U(\mathbf{p}_w) \leq \delta U(\mathbf{p})$. By adding this condition, the robot will be further apart from the obstacle at next sample time; therefore, the robot moves behind the obstacle and avoids collision.

The original navigation function chooses the waypoint with lowest potential $U_{current}(\mathbf{p}_w)$. However, this always leads to a conservative and long path. Instead, we use A* to decide the next waypoint by minimizing $f(\mathbf{p}_w)$:

$$\begin{aligned} f(\mathbf{p}_w) &= g(\mathbf{p}_w) + h(\mathbf{p}_w) \\ &= U_{current}(\mathbf{p}_w) + \gamma_1 \left\| \mathbf{p}_w - \mathbf{p}_t \right\|. \end{aligned} \tag{5.7}$$

When $head\_on\_flag = 0$, the overall strategy for $head\_on\_flag = 1$ still holds. The difference is that those possible next waypoints on the port side of the robot are eliminated from the set $\mathcal{P}_w$. A comparison is given in Figure 5.3. In Figure 5.3b, when $head\_on\_flag = 1$, the marine robot can move either port side or starboard side. While in Figure 5.3a, the robot is not allowed to choose the grid points on port side.



(a) Head-on situation.

(b) Other situations.

Figure 5.3: Illustration of choosing next waypoint with $n_r = 2$.

5.1.4 Parameter tuning

The parameters need to be tuned are κ, n_r, R, and γ_1. Small n_r means the robot will update the workspace information more frequently. R can be considered as the range of conflict detection. These two parameters should be reasonable and reflect real-world conditions. γ_1 is the weight between $U_{current}(\mathbf{p}_w)$ and $\|\mathbf{p}_w - \mathbf{p}_t\|$. It should be small enough, since $\|\mathbf{p}_w - \mathbf{p}_t\|$ is much larger than $U_{current}(\mathbf{p}_w)$.

Next, we discuss the influence of κ. In order to simplify parameter tuning, n_r, R, and γ_1 are fixed for all simulation cases, and the performance of the navigation function method is determined by the tuning of κ.

The level step of contour lines in figures 5.4 - 5.7 is 0.02. In Figure 5.5, where κ is larger than N_o, most of grid points have potential value of 1. The potential starts to decrease within a range of the target. On the other hand, in Figure 5.6, where κ is smaller than N_o, only within a small range of the obstacles is the potential close to 1. The problem of these potential fields in figures 5.5 - 5.6 is that $U_{current}(\mathbf{p}_w)$ is almost the same over the whole workspace. Then, for a given γ_1, $\|\mathbf{p}_w - \mathbf{p}_t\|$ is more important when choosing a next waypoint, which means the shortest path is preferred.

In figures 5.4 and 5.7, where κ is pretty close to the number of the obstacle(s) N_o, both 3D and 2D plots give good visualization of the potential fields. We can see clearly how the potential changes within the workspace. The potential varies within a neighborhood of a given point. With γ_1 fixed, these potential fields benefits the process of choosing a next waypoint by considering the effect of both $U_{current}(\mathbf{p}_w)$ and $\|\mathbf{p}_w - \mathbf{p}_t\|$, even the number of obstacles are different.

A good choice of κ gives better visualization and help find a good next waypoint. It is recommended to choose $\kappa \approx N_o$ as a starting point.

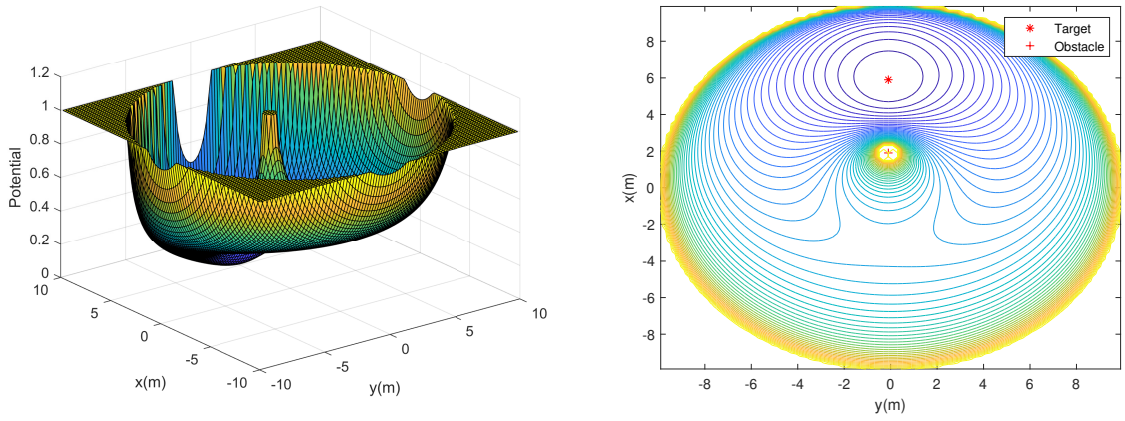


Figure 5.4: Potential field generated by the navigation function method with 1 obstacle and $\kappa = 1.5$.

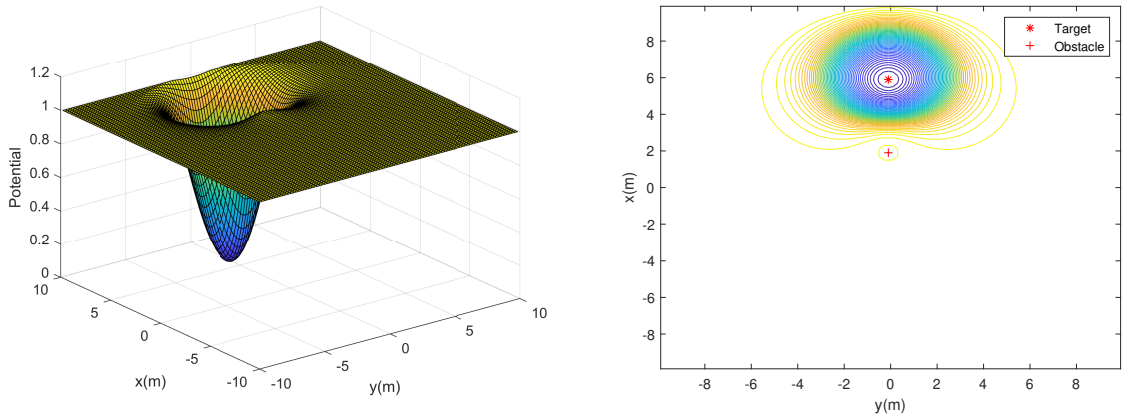


Figure 5.5: Potential field generated by the navigation function method with 1 obstacle and $\kappa = 3$.

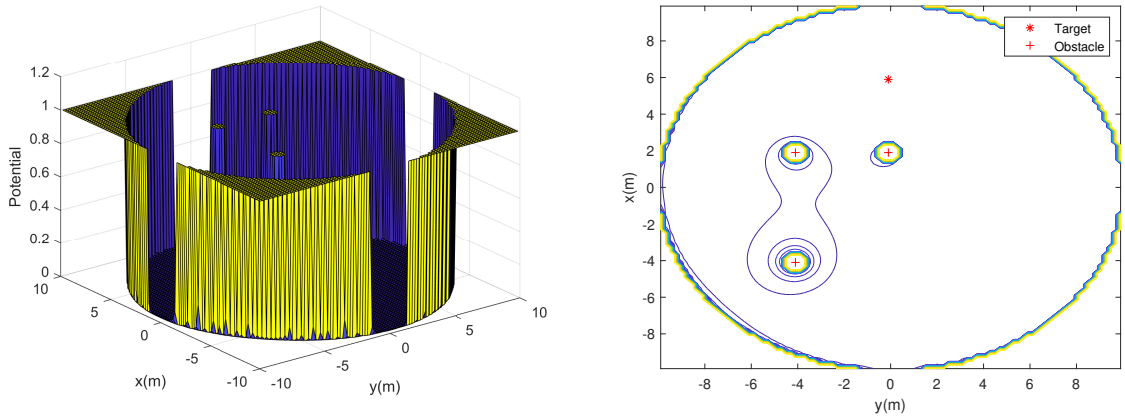


Figure 5.6: Potential field generated by the navigation function method with 3 obstacle and $\kappa = 1.5$.

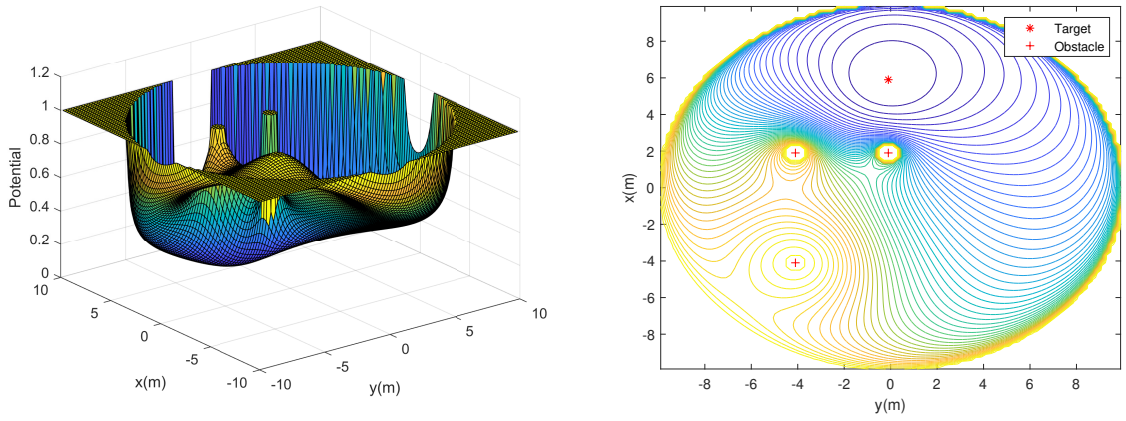


Figure 5.7: Potential field generated by the navigation function method with 3 obstacle and $\kappa = 3$.

5.2 Waypoint generation using stream function method

5.2.1 Potential flows and complex potential

In this section, some important concepts from hydrodynamics are introduced. Detailed information can be found in Milne-Thomson (1996), Currie (2016) and Axler et al. (2013).

Potential flows and velocity potential

Consider the flow in a 2D plane. The flow is irrotational if the vorticity vector $\omega_f \in \mathbb{C}$ is zero everywhere in the fluid, which is defined as

$$\omega_f = \nabla \times \nu_f = 0, \tag{5.8}$$

where $\nu_f = \nu_x + \mathrm{i}\nu_y$ is the velocity of the flow, and i is the imaginary unit.

For any scalar function ϕ_f, Equation (5.9) holds, given below:

$$\nabla \times \nabla \phi_f = 0. \tag{5.9}$$

The condition of irrotationality can then be satisfied by combining equations (5.8) and (5.9):

$$\begin{aligned} \nu_f &= \nabla \phi_f, \\ \nabla \nu_f &= 0, \end{aligned} \tag{5.10}$$

where ϕ_f is called velocity potential. The irrotational flow fields are called potential flows, represented by $\nu_f = \nabla \phi_f$.

The velocity of the flow can be calculated from a velocity potential,

$$\nu_x = \frac{\partial \phi_f}{\partial x}, \quad \nu_y = \frac{\partial \phi_f}{\partial y}. \tag{5.11}$$

The velocity potential also satisfies Laplace's equation $\nabla^2 \phi_f = 0$, since an idea flow must satisfy the condition of continuity $\nabla \nu_f = 0$.

Stream function

In cartesian coordinates, the condition of continuity $\nabla \nu_f = 0$ can be expressed as

$$\frac{\partial \nu_x}{\partial x} + \frac{\partial \nu_y}{\partial y} = 0. \tag{5.12}$$

Now introducing a stream function ψ_f which is defined as

$$\nu_x = \frac{\partial \phi_f}{\partial y}, \quad \nu_y = -\frac{\partial \phi_f}{\partial x}. \tag{5.13}$$

Then the stream function also satisfies the Laplace's equation by its definition, i.e., $\nabla^2 \psi_f = 0$. We also introduce the concept of streamlines. A streamline is the line along which $\psi_f = constant$.

Complex potential

With the velocity potential and the stream function, we can now define an irrotational and inviscid 2D flow $\omega_f(z)$

$$\omega_f(z) = \phi_f + \mathrm{i}\psi_f, \tag{5.14}$$

where $z = x + \mathrm{i}y$. We can see that the real part of $\omega_f(z)$ is the velocity potential, and the imaginary part the stream function.

By equating the velocity components, the velocity potential and the stream function satisfy Cauchy-Riemann equation

$$\frac{\partial \phi_f}{\partial x} = \frac{\partial \psi_f}{\partial y}, \quad \frac{\partial \phi_f}{\partial y} = -\frac{\partial \psi_f}{\partial x}. \tag{5.15}$$

Basic flow types

The uniform flow, sink, source and vortex are the most important flow types.

- For the uniform flow, the complex potential is defined as

$$\omega_f(z) = \nu_f z. \tag{5.16}$$

- The complex potential for sink and source are

$$\omega_f(z) = Cln(z), \tag{5.17}$$

  where $C > 0$ denotes sources and $C < 0$ denotes sinks. The sink flow can be used to represent a target

- The complex potential for vortex is

$$\omega_f(z) = C\mathrm{i}ln(z), \tag{5.18}$$

  where $C > 0$.

5.2.2 Obstacle representation

Circle theorem

The circle theorem is important to represent a circular obstacle. The theorem is given in the following and it is valid for a obstacle at an arbitrary position.

Theorem 5.1 (Circle theorem (Milne-Thomson, 1996))**.** *Let there be irrotational two-dimensional flow of incompressible inviscid fluid in the z-plane. Let there be no rigid boundaries and let the complex potential of the flow be $f(z)$, where the singularities of $f(z)$ are all at a distance greater than a from the point b. Let $\bar{f}(z)$ and $\bar{b}$ be the conjugate function of $f(z)$ and b respectively. If a circular cylinder, typified by its cross-section the circle $|z-b| = a$, is introduced into the flow, the complex potential becomes*

$$\omega(z)_f = \phi_f + \mathrm{i}\psi_f = f(z) + \bar{f}(\frac{a^2}{z-b} + \bar{b}). \tag{5.19}$$

By applying the circular theorem, the stream function for a circular obstacle in a uniform flow is given as

$$\psi_f = Cy(1 - \frac{a^2}{x^2+y^2}), \tag{5.20}$$

where (x, y) is the position of the robot, a is the radius of the obstacle. An example is shown in Figure 2.12a.

The circular obstacle in a sink flow is given as

$$\psi_f = -Catan(\frac{y}{x}) + Catan\left(\frac{\frac{a^2(y-y_{obs})}{(x-x_{obs})^2+(y-y_{obs})^2} + y_{obs}}{\frac{a^2(x-x_{obs})}{(x-x_{obs})^2+(y-y_{obs})^2} + x_{obs}}\right), \tag{5.21}$$

where (x_{obs}, y_{obs}) is the position of the obstacle. An example of obstacle in a sink flow is shown in Figure 2.12b.

When using circle theorem for obstacle avoidance, the parameter a in Equation (5.19) is usually larger than the radius of the obstacle to ensure a safety distance, so that if the robot stops moving at a stagnation point on the surface of the obstacle, it will not cause collision.

Multiple obstacles

If in the flow field there are multiple circular obstacles, the Laplace's equation with multiple boundary conditions must be solved to obtain the stream function. However, this is analytically impossible. In the case of multiple obstacles, a method called addition and thresholding, presented in Waydo and Murray (2003) and Sullivan et al. (2003).

The basic idea of addition and thresholding is that each obstacle has its influence area. This method removes the influence of those obstacles that should not have a direct impact on the streamlines, while the influences of the other obstacles are added together to create the vector field. An example is shown in Figure 5.8.

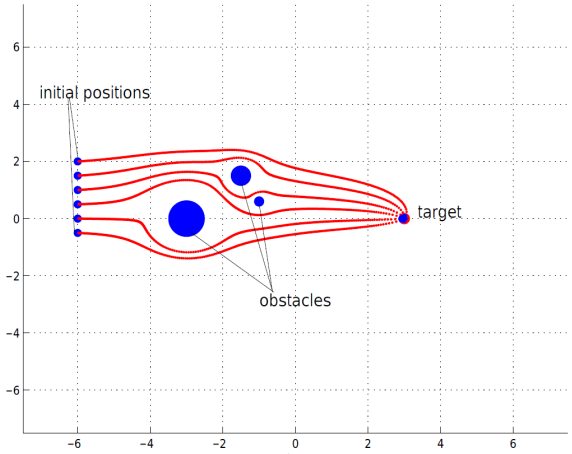


Figure 5.8: Avoidance of multiple obstacles. Courtesy: Ye et al. (2005).

To apply this method in a scenario with N_o obstacles, assume ψ_i is the stream function generated by the ith obstacle, and $i \in \{1, 2, ..., N_o\}$. For each obstacle, there is a range l_i, within which the stream function is only influenced by the ith obstacle. Otherwise, the stream function is influenced by all the obstacles. The stream function for the whole workspace is given as

$$\psi_f(\mathbf{p}) = \sum_{i=1}^{N_o} c_i \psi_i(\mathbf{p}), \quad c_i = \begin{cases} 1, & if \ \|\mathbf{p} - \mathbf{p}_i\| \leq l_i \\ 0, & if \ \|\mathbf{p} - \mathbf{p}_i\| > l_i \end{cases}. \tag{5.22}$$

Avoidance of a moving obstacle

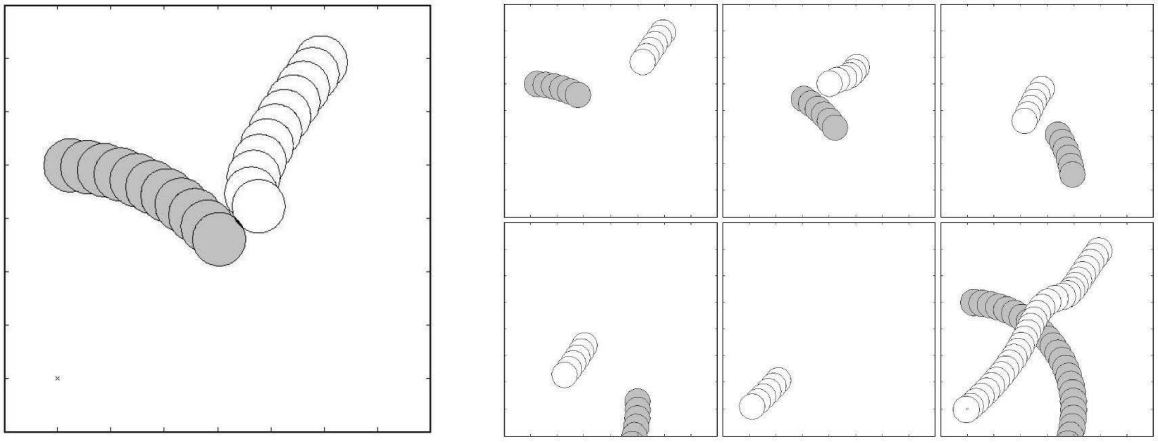
If the robot needs to avoid moving obstacles, using a stream function calculated at each sample time is insufficient. Instead, a dynamic component should be added to the original stream function. This method is developed in Waydo and Murray (2003).

Theorem 5.2 (Stream function for a moving obstacle)**.** *Consider a circular obstacle in an arbitrary irrotational two-dimensional flow of incompressible inviscid fluid with complex potential $f(z)$ as in the Circle Theorem above. Let the obstacle be moving at constant velocity $v_o = v_x + \mathrm{i}v_y$. The complex potential for the flow about the obstacle is given by*

$$\begin{aligned} \omega_f(z) &= \omega_s(z) + \omega_d(z) \\ &= \omega_s(z) - v_x\left(\frac{a^2}{z-b} + \bar{b}\right) - \mathrm{i}v_y\left(\frac{a^2}{z-b} + \bar{b}\right), \end{aligned} \tag{5.23}$$

where $\omega_s(z)$ is the static stream function that would be derived if the obstacle were not moving, and $\omega_d(z)$ is the dynamic component in order to avoid a moving obstacle.

However, this method might not work for multiple moving obstacles. Consider two obstacles moving in opposite directions, the dynamic components will cancel out each other. Also, the boundary condition for multiple obstacles becomes a problem.



(a) Collision due to no dynamic component.

(b) Collision avoidance With dynamic component.

Figure 5.9: Avoidance of a moving obstacle. The Vehicle is the white circle and the obstacle is the black one. Courtesy: Waydo and Murray (2003).

Avoidance of multiple moving obstacles

For the avoidance of multiple moving obstacles, the vortex flows with centers at $\mathbf{p}_i$ are added to the original stream function.

The stream function of a vortex flow is

$$\psi_{vi}(\mathbf{p}) = C_{v_i} ln(\|\mathbf{p} - \mathbf{p}_i\|^2), \quad i \in \{1, 2, ..., N_o\}, \tag{5.24}$$

where C_{v_i} is a tuning parameter.

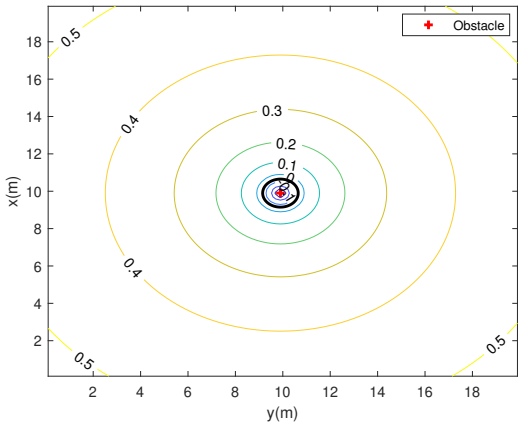


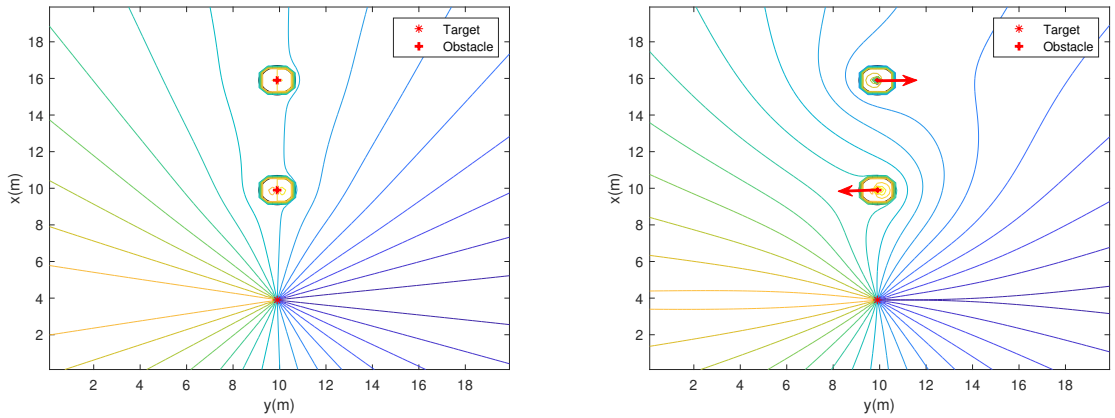
Figure 5.10: The streamlines of a vortex flow.

Now the stream function for the whole workspace becomes

$$\psi_v(\mathbf{p}) = \sum_{i=1}^{N_o} c_i(\psi_i(\mathbf{p}) + \psi_{v_i}(\mathbf{p})), \quad c_i = \begin{cases} 1, & if \ \|\mathbf{p} - \mathbf{p}_i\| \leq l_i \\ 0, & if \ \|\mathbf{p} - \mathbf{p}_i\| > l_i \end{cases}. \tag{5.25}$$

By adding vortex flows with well tuned C_{v_i}, the streamlines close to the obstacles will be moved behind the obstacles, or in other words, towards the opposite direction of obstacles' velocities. A robot following

a constant streamline then will also move behind the obstacles. This is presented in Figure 5.11. In this way, using a stream function calculated at each sample time gives collision avoidance in dynamic environments, since it incorporates the information of obstacles' velocity by vortex flows. The streamlines far away from the obstacles are only slightly influenced. This means a robot following a streamline only performs collision avoidance when it gets close to an obstacle, otherwise it will move directly towards its target.



(a) The flow without vortex.

(b) The flow with vortex.

Figure 5.11: Avoidance of multiple moving obstacles by adding vortex flows. The red arrows indicate the directions of obstacles' velocities.

5.2.3 Choosing the next waypoint

In the stream function method, a robot follows a constant streamline ψ_f will eventually reach its target, which is modelled as a sink flow. However, since the workspace is discretized by a set of grid points, it is impossible to choose a next waypoint $\mathbf{p}_{next}$ with the same ψ_f as the current waypoint $\mathbf{p}_c$. Instead, we choose a next waypoint with the closest ψ_f. Similar to the navigation function method, the possible next waypoints $\mathbf{p}_w$ should be in a small range of current waypoint.

When the current waypoint $\mathbf{p}_c = [x_c\ y_c]^\top$ and the target $\mathbf{p}_t = [x_t\ y_t]^\top$ are far apart, illustrated in Figure 5.12a, the next waypoint is also chosen as the solution of an optimization problem:

$$\begin{aligned} \mathbf{p}_{next} := \min_{\mathbf{p}_w} \quad & |\psi_v(\mathbf{p}_w) - \psi_v(\mathbf{p}_c)| + \gamma_2 \|\mathbf{p}_w - \mathbf{p}_t\| \\ \text{s.t.} \quad & \mathbf{p}_w \in \mathcal{P}_w \\ & \mathcal{P}_w := \{\mathbf{p} \mid max\{\tfrac{|x_c - x|}{d_x}, \tfrac{|y_c - y|}{d_y}\} = n_r\} \end{aligned} \tag{5.26}$$

where γ_2 is a tuning parameter which should be sufficiently small.

Similarly, the set of the next possible waypoints $\mathcal{P}_w$ should satisfy: $max\{\frac{|x_c-x|}{d_x}, \frac{|y_c-y|}{d_y}\} = n_r$. The robot will move toward the target and following a streamline, by minimizing the defined cost function. The reason to penalize the term $\gamma_2 \|\mathbf{p}_w - \mathbf{p}_t\|$ is that the streamlines do not give any information about the direction of moving. If we only consider minimizing $|\psi_v(\mathbf{p}_w) - \psi_v(\mathbf{p}_c)|$, the robot might be guided away from the target. An example is shown in Figure 5.13.

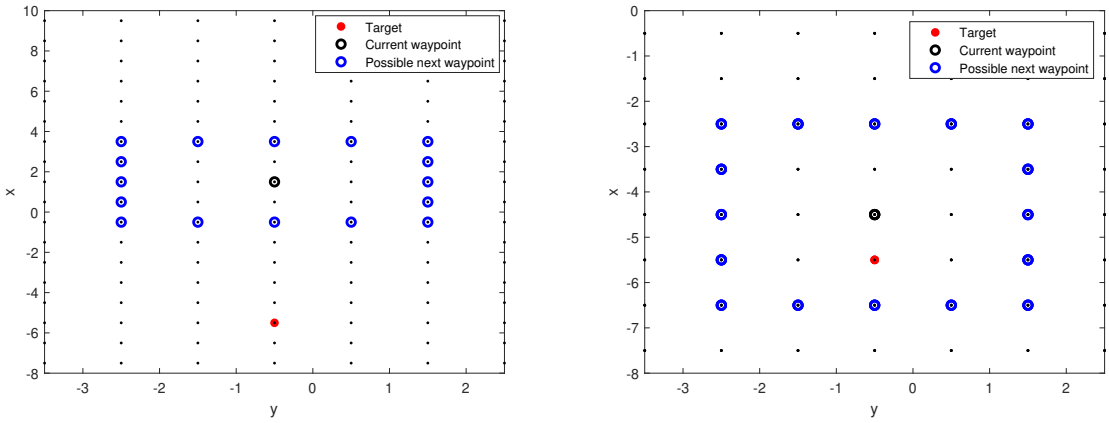
When the current waypoint is close to the target, the waypoint chosen by Equation (5.26) may past the target. In this situation, if Equation (5.27) is satisfied, the next waypoint should be the target position.

$$\mathbf{p}_{next} := \mathbf{p}_t \quad if\ max\{\frac{|x_c - x|}{d_x}, \frac{|y_c - y|}{d_y}\} < n_r \tag{5.27}$$

The algorithm is summarized as follow:

Algorithm 3: The stream function method

Algorithm stream_function_waypoint ($\psi_f(\mathbf{p})$,$\mathbf{p}_c$,n_r):
if *Equation* (5.27) *satisfied* **then**
  $\mathbf{p}_{next} = \mathbf{p}_t$
else
  Find $\mathbf{p}_{next}$ by Equation (5.26)
end
$\mathbf{p}_c \leftarrow \mathbf{p}_{next}$
return $\mathbf{p}_{next}$



(a) **(b)**

Figure 5.12: Illustration of choosing next waypoint with $n_r = 2$.

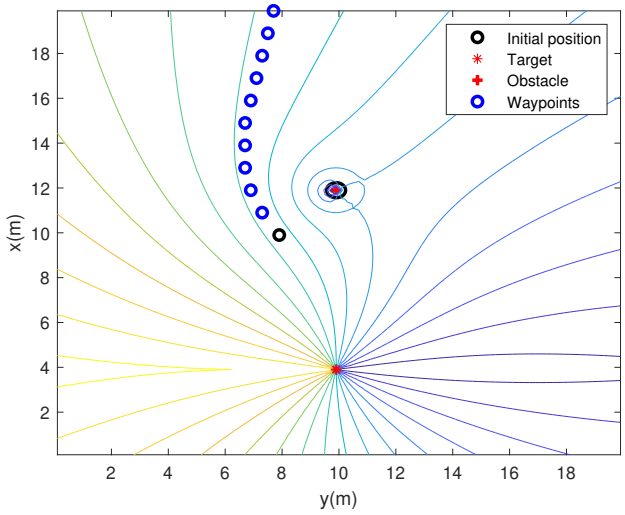


Figure 5.13: Waypoints that guide the robot away from the target.

5.3 Path generation using Bézier curves

The method of path generation is based on Knædal (2019). Before introducing the quadratic programming approach used in this thesis, the basic concept of Bézier curves and a pramatic approach will be introduced.

A Bézier curve of n degree is defined by $n+1$ control points $\mathbf{P}_i$, where $i \in [0, 1, ..., n]$. The first and the last control points are the end points of a curve. Let a Bézier curve parameterized by $s \in [0, 1]$ denoted $\mathbf{B}(s) = [x(s), y(s)]^\top$. Then $\mathbf{B}(s)$ can be expressed as a linear combination of the control points:

$$\begin{aligned}\mathbf{B}(s) &= b_0^n(s)\mathbf{P}_0 + b_1^n(s)\mathbf{P}_1 + ...b_n^n(s)\mathbf{P}_n \\ &= [b_0^n(s)\ b_1^n(s)\ ...\ b_n^n(s)] \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_n \end{bmatrix} \\ &= [1\ s\ s^2\ ...\ s^n] \begin{bmatrix} b_{0,0} & 0 & 0 & ... & 0 \\ b_{1,0} & b_{1,1} & 0 & ... & 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n,0} & b_{n,1} & b_{n,2} & ... & b_{n,n} \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_n \end{bmatrix},\end{aligned} \tag{5.28}$$

where $b_i^n(s)$ is the Bernstein basis polynomial of degree n,

$$b_i^n(s) = \frac{n!}{i!(n-i)!} s^i (1 - s^{n-i}),\ i \in [0, 1, ..., n], \tag{5.29}$$

and $b_{i,j}$ are given by (Joy, 2000b):

$$b_{i,j} = (-1)^{i-j} \frac{n!}{i!(n-i)!} \frac{i!}{j!(i-j)!}. \tag{5.30}$$

The derivative of a nth order Bézier curve is still a Bézier curves but of $n-1$ order. Since the control points are independent of variable s, the derivative of a Bézier curve can be obtained by calculating the derivative of the Bernstein basis function $b_i^n(s)$, given in Joy (2000a)

$$\frac{\mathrm{d}}{\mathrm{d}s} b_i^n(s) = n(b_{i-1}^{n-1}(s) - b_i^{n-1}(s)). \tag{5.31}$$

Then the derivative of a Bézier curve is

$$\frac{\mathrm{d}}{\mathrm{d}s} \mathbf{B}(s) = n \sum_{i=0}^{n-1} b_i^{n-1}(s)(\mathbf{P}_{i+1} - \mathbf{P}_i). \tag{5.32}$$

By using equations (5.31) and (5.32), one can find higher-order derivatives of a Bézier curve.

5.3.1 Pragmatic approach

The goal of stepwise path generation in this thesis is to generate a path of $\mathcal{C}^3$-continuity from the current waypoint to the next waypoint using a pragmatic approach. For detailed definition and explanation, see

Knædal (2019).

To satisfy the requirement of $\mathcal{C}^3$-continuity, the degree of a Bézier curve should be $n = 7$ (Knædal, 2019). This means for each path segment, there are 8 control points. Assuming that we have waypoints $\mathbf{WP}_k = [x_k \ y_k]^\top$, $\mathbf{WP}_{k+1} = [x_{k+1} \ y_{k+1}]^\top$, and $\mathbf{WP}_{k+2} = [x_{k+2} \ y_{k+2}]^\top$ in NED frame. The path segment from $\mathbf{WP}_k$ to $\mathbf{WP}_{k+1}$ is generated by a set of control points $\mathbf{P}_{i,k}$, where $i \in [0, 1, ..., 7]$. Now we need to find the control points $\mathbf{P}_{i,k+1}$, where $i \in [0, 1, ..., 7]$ to generate the path segment from $\mathbf{WP}_{k+1}$ to $\mathbf{WP}_{k+2}$. This is achieved by solving equations (5.33) and (5.34). If $k = 1$, then the path segment is a straight line from $\mathbf{WP}_1$ to $\mathbf{WP}_2$.

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{1,k+1} \\ \mathbf{P}_{2,k+1} \\ \mathbf{P}_{3,k+1} \end{bmatrix} = \begin{bmatrix} 2\mathbf{P}_{7,k} - \mathbf{P}_{6,k} \\ -2\mathbf{P}_{6,k} + \mathbf{P}_{5,k} \\ 2\mathbf{P}_{7,k} - 3\mathbf{P}_{6,k} + 3\mathbf{P}_{5,k} - \mathbf{P}_{4,k} \end{bmatrix}, \tag{5.33}$$

$$\begin{cases} \mathbf{P}_{0,k+1} &= \mathbf{WP}_k \\ \mathbf{P}_{7,k+1} &= \mathbf{WP}_{k+1} \\ \mathbf{P}_{4,k+1} &= \mathbf{WP}_{k+1} - \delta[cos(\psi_{k+1}), sin(\psi_{k+1})] \\ \mathbf{P}_{5,k+1} &= \mathbf{WP}_{k+1} - \frac{\delta}{\mu_0}[cos(\psi_{k+1}), sin(\psi_{k+1})] \\ \mathbf{P}_{6,k+1} &= \mathbf{WP}_{k+1} - \frac{\delta}{2\mu_0}[cos(\psi_{k+1}), sin(\psi_{k+1})] \end{cases}, \tag{5.34}$$

where δ and μ_0 are tuning parameters, and

$$\psi_{k+1} = atan2(y_{k+1} - y_k, x_{k+1} - x_k). \tag{5.35}$$

Once the control points are found, the desired path $\boldsymbol{\eta}_d(s)$ in Equation (3.1) can be calculated by using equations (5.28) and (3.2).

5.3.2 Optimal control points

As presented in Knædal (2020), the pragmatic approach can be improved by formulating a quadratic programming problem which minimizes the length of each path segment.

From Equation (5.33), we can see that $\mathbf{P}_{1,k+1}$, $\mathbf{P}_{2,k+1}$, and $\mathbf{P}_{3,k+1}$ are uniquely decided by $\mathbf{P}_{7,k}$, $\mathbf{P}_{6,k}$, $\mathbf{P}_{5,k}$, and $\mathbf{P}_{4,k}$. In Equation (5.34), $\mathbf{P}_{0,k+1}$ and $\mathbf{P}_{7,k+1}$ are end points of a path segment. Therefore, only $\mathbf{P}_{4,k+1}$, $\mathbf{P}_{5,k+1}$, and $\mathbf{P}_{6,k+1}$ need to be optimized.

The decision variables can be reduced by expressing $\mathbf{P}_{4,k+1}$, $\mathbf{P}_{5,k+1}$, and $\mathbf{P}_{6,k+1}$ in a path-fixed reference frame, shown in Fiugre 5.14. The path-fixed reference frame is rotated by a path angle α_k with respect to the NED frame, where

$$\alpha_k = atan2(x_{k+1} - x_k, y_{k+1} - y_k). \tag{5.36}$$

In the path-fixed reference frame, the decision variables are only the y-coordinates of the control points, $\boldsymbol{\chi}_{3\times1} = [y_{4,k+1} \ y_{5,k+1} \ y_{6,k-1}]^\top = [\chi_1 \ \chi_2 \ \chi_3]^\top$. Then the objective function is

$$J = \min_{\boldsymbol{\chi}} \int_0^1 \left\| \frac{\mathrm{d}}{\mathrm{d}s}\mathbf{B}(s) \right\|^2 \mathrm{d}s. \tag{5.37}$$

Next, the constraints on this optimization problem is briefly introduced. First, in Figure 5.14, we can see $\mathbf{P}_{4,k}$, $\mathbf{P}_{5,k}$, $\mathbf{P}_{6,k}$, $\mathbf{WP}_{k+1}$, $\mathbf{P}_{1,k+1}$, $\mathbf{P}_{2,k+1}$, and $\mathbf{P}_{3,k+1}$ are arranged in order. Similarly, the constraint on $\boldsymbol{\chi}$ is

$$0 \leq \chi_1 \leq \chi_2 \leq \chi_3 \leq y_{7,k+1} \leq y_{1,k+2} \leq y_{2,k+2} \leq y_{3,k+2}. \tag{5.38}$$

Besides, we want the path segment inside a corridor, presented in Figure 5.15. This gives

$$y_{1,k+2} \ \leq \ y_{7,k+1} + \frac{\zeta}{2}, \tag{5.39a}$$

$$y_{2,k+2} \ \leq \ y_{7,k+1} + \frac{\zeta}{2}, \tag{5.39b}$$

$$y_{3,k+2} \ \leq \ y_{7,k+1} + \frac{\zeta}{2}, \tag{5.39c}$$

where ζ is the corridor width.

Writing Equation (5.39) in terms of $\boldsymbol{\chi}$ gives

$$-\chi_3 \ \leq \ y_{7,k+1} + \frac{\zeta}{2}, \tag{5.40a}$$

$$\chi_2 - 4\chi_3 \ \leq \ -3y_{7,k+1} + \frac{\zeta}{2}, \tag{5.40b}$$

$$-\chi_1 + 6\chi_2 - 12\chi_3 \ \leq \ -7y_{7,k+1} + \frac{\zeta}{2}, \tag{5.40c}$$

In addition, we also want to keep the curvature low. The constraint is (Knædal, 2020):

$$\chi_3 \ \leq \ y_{7,k+1} - \epsilon, \tag{5.41a}$$

$$-\chi_2 + 4\chi_3 \ \leq \ -3y_{7,k+1} - \epsilon, \tag{5.41b}$$

$$\chi_1 - 6\chi_2 + 12\chi_3 \ \leq \ -7y_{7,k+1}\epsilon, \tag{5.41c}$$

where ϵ is a tuning parameter.

The quadratic optimization problem now can be summarized as follow,

$$\begin{aligned} J = \min_{\boldsymbol{\chi}} \quad & \int_0^1 \left\| \tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{B}(s) \right\|_2^2 \mathrm{d}s \\ \text{s.t.} \quad & \mathbf{A}\boldsymbol{\chi} \ \leq \ \mathbf{b} \\ & \boldsymbol{\chi} \ \geq \ \mathbf{0} \\ & \chi_i \ \leq \ y_{7,k+1},\ i \in [1,\ 2,\ 3] \end{aligned} \tag{5.42}$$

where

$$\mathbf{A}^\top = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 \\ -1 & 1 & 0 & 1 & 6 & 0 & -1 & -6 & -1 & -5 \\ 0 & -1 & -1 & -4 & -12 & 1 & 4 & 12 & 3 & 8 \end{bmatrix}, \tag{5.43}$$

$$\mathbf{b}^\top = \begin{bmatrix} 0 & 0 & y_{7,k+1} + \frac{\zeta}{2} & -3y_{7,k+1} + \frac{\zeta}{2} & -7y_{7,k+1} + \frac{\zeta}{2} & y_{7,k+1} - \epsilon & -3y_{7,k+1} - \epsilon & -7y_{7,k+1}\epsilon & 2y_{7,k+1} & 4y_{7,k+1} \end{bmatrix}. \tag{5.44}$$

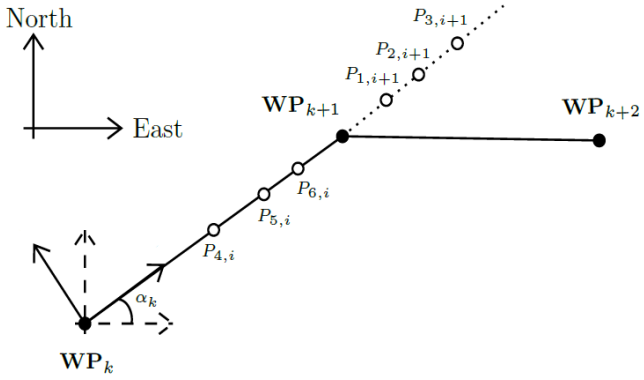


Figure 5.14: A path-fixed reference frame with origin in $\mathbf{WP}_k = [x_k \; y_k]^\top$ rotated by a an angle α_k relative to the NED-frame. Courtesy: Knædal (2020).

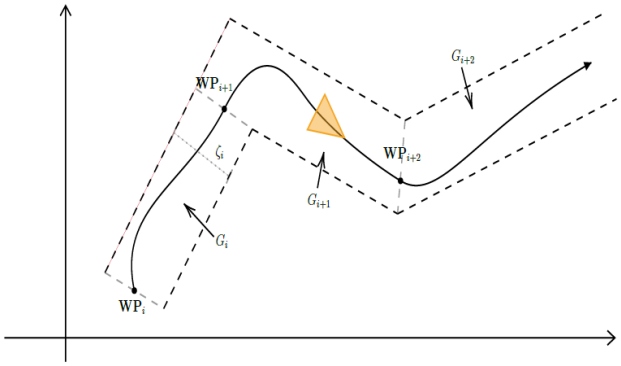


Figure 5.15: The constraint of corridor on path segment. Courtesy: Knædal (2020).

Chapter 6

Control system design

In this chapter, maneuvering control design and thrust allocation algorithm will be introduced. The control system is designed to follow the path segment generated by path generation method using Bézier curves.

6.1 Backstepping maneuvering control design

The following design is based on Skjetne (2005). Using the control design model in Equation (4.2), we choose the LgV backstepping design and define:

$$\begin{cases} z_1 := \mathbf{R}(\psi)^\top(\boldsymbol{\eta} - \boldsymbol{\eta}_d(s)) \\ z_2 := \boldsymbol{\nu} - \alpha_1, \\ \omega := \dot{s} - v_s(t,s) \end{cases} . \tag{6.1}$$

6.1.1 Path parametrization

In the maneuvering control design, the desired parametrized pose $\boldsymbol{\eta}_d(s) = [x_d\ y_d\ \psi_d]^\top$ is now available by using the path generation method. For a path of $\mathcal{C}^3$-continuity, the relevant derivatives of desired pose $\boldsymbol{\eta}_d$ with respect to s are

$$\boldsymbol{\eta}_d^s = \begin{bmatrix} x_d^s \\ y_d^s \\ \psi_d^s \end{bmatrix}, \quad \boldsymbol{\eta}_d^{2s} = \begin{bmatrix} x_d^{2s} \\ y_d^{2s} \\ \psi_d^{2s} \end{bmatrix}, \tag{6.2}$$

where

$$\psi_d^s = \frac{x_d^s y_d^{2s} - y_d^s x_d^{2s}}{(x_d^s)^2 + (y_d^s)^2}, \tag{6.3}$$

$$\psi_d^{2s} = \frac{x_d^s y_d^{3s} - y_d^s x_d^{3s}}{(x_d^s)^2 + (y_d^s)^2} - \frac{(x_d^s y_d^{2s} - y_d^s x_d^{2s})(2x_d^s x_d^{2s} + 2y_d^s y_d^{2s})}{((x_d^s)^2 + (y_d^s)^2)^2}. \tag{6.4}$$

6.1.2 Backstepping – step 1

In the step 1, we define the first control Lyapunov function (CLF)

$$V_1 = \frac{1}{2} z_1^\top z_1 \tag{6.5}$$

The design in the step 1 follows these steps:

$$\begin{aligned}\dot{z}_1 &= \mathbf{R}(\psi)^\top(\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d^s(s)\dot{s}) + \dot{\mathbf{R}}(\psi)^\top(\boldsymbol{\eta} - \boldsymbol{\eta}_d) \\ &= -r\mathbf{S}z_1 + z_2 + \alpha_1 - \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s(\omega + v_s),\end{aligned} \tag{6.6}$$

The derivative of V_1 with respect to time is

$$\begin{aligned}\dot{V}_1 &= z_1^\top\dot{z}_1 \\ &= -z_1^\top r\mathbf{S}z_1 + z_1^\top z_2 + z_1^\top(\alpha_1 - \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s(\omega + v_s)) \\ &= -z_1^\top\mathbf{K}_1 z_1 + z_1^\top z_2 + \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s\omega \\ &= -z_1^\top\mathbf{K}_1 z_1 + z_1^\top z_2 + \rho_1\omega,\end{aligned} \tag{6.7}$$

by choosing

$$\alpha_1 = -\mathbf{K}_1 z_1 + \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s v_s, \tag{6.8}$$

where $\mathbf{K}_1$ is positive definite, and $\rho_1 = \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s$ is the first tuning function.

We postpone dealing with z_2 until next step. However, we will deal with ω now.

6.1.3 Update laws acting in output space

We now need to decide the update law in order for it to only act in the output space of $\boldsymbol{\eta}$. After step 1, we have

$$\begin{aligned}\dot{V}_1 &= -z_1^\top\mathbf{K}_1 z_1 + z_1^\top z_2 + \rho_1\omega \\ \rho_1 &= \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s\end{aligned}. \tag{6.9}$$

Unit-tangent gradient update law

We choose

$$\begin{aligned}\omega &= \mu\frac{\boldsymbol{\eta}_d^{s\top}}{|\boldsymbol{\eta}_d^s|}\mathbf{R}(\psi)z_1 \\ &\implies \dot{s} = v_s(t,s) + \mu\frac{\boldsymbol{\eta}_d^{s\top}}{|\boldsymbol{\eta}_d^s|}\mathbf{R}(\psi)z_1\end{aligned}. \tag{6.10}$$

This gives $\rho_1\omega \leq 0$. Concluding step 1, we have

$$\begin{cases}\dot{z}_1 = -r\mathbf{S}z_1 + z_2 + \alpha_1 - \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s(\omega + v_s) \\ \quad\dot{s} = v_s(t,s) + \mu\dfrac{\boldsymbol{\eta}_d^{s\top}}{|\boldsymbol{\eta}_d^s|}\mathbf{R}(\psi)z_1 \\ \dot{V}_1 \leq -z_1^\top\mathbf{K}_1 z_1 + z_1^\top z_2 \\ \alpha_1 = -\mathbf{K}_1 z_1 + \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s v_s\end{cases}. \tag{6.11}$$

After choosing the update law, $\dot{\alpha}_1$ need to be calculated since it must be cancled in the next step. We get

$$\dot{\alpha_1} = \mathbf{K}_1(\mathbf{K}_1 + r\mathbf{S})z_1 - \mathbf{K}_1 z_2 - r\mathbf{S}\mathbf{R}^\top\dot{\boldsymbol{\eta}}_d + \mathbf{R}^\top\ddot{\boldsymbol{\eta}}_d. \tag{6.12}$$

6.1.4 Backstepping – step 2

In the step 2, the second CLF is defined as

$$V_2 = V_1 + \frac{1}{2} z_2^\top \mathbf{M} z_2. \tag{6.13}$$

The design in step 2 follows these steps:

$$\begin{aligned}
\mathbf{M}\dot{z}_2 &= \mathbf{M}\dot{\boldsymbol{\nu}} - \mathbf{M}\dot{\alpha}_1 \\
&= \mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{M}\mathbf{K}_1\dot{z}_1 + r\mathbf{M}\mathbf{S}\mathbf{R}^\top\dot{\boldsymbol{\eta}}_d - \mathbf{M}\mathbf{R}^\top\ddot{\boldsymbol{\eta}}_d \\
&= -(\mathbf{C}+\mathbf{D})\boldsymbol{\nu} + \boldsymbol{\tau} - \mathbf{M}\dot{\alpha}_1 \\
&= -(\mathbf{C}+\mathbf{D})(z_2+\alpha_1) + \boldsymbol{\tau} - \mathbf{M}\dot{\alpha}_1.
\end{aligned}$$

Then differentiate V_2:

$$\begin{aligned}
\dot{V}_2 &= -z_1^\top\mathbf{K}_1 z_1 + z_2^\top(-(\mathbf{C}+\mathbf{D})(z_2+\alpha_1) + \boldsymbol{\tau} - \mathbf{M}\dot{\alpha}_1) \\
&= -z_1^\top\mathbf{K}_1 z_1 + z_2^\top(-(\mathbf{C}+\mathbf{D})\alpha_1 + \boldsymbol{\tau} - \mathbf{M}\dot{\alpha}_1 + z_1) - z_2^\top(\mathbf{C}+\mathbf{D})z_2.
\end{aligned} \tag{6.13}$$

Now we need to purpose a control law for $\boldsymbol{\tau}$ that renders $\dot{V}_2$ negative definite. We choose

$$\boldsymbol{\tau} = -z_1 + (\mathbf{C}+\mathbf{D})\alpha_1 + \mathbf{M}\dot{\alpha}_1 - \mathbf{K}_2 z_2, \tag{6.13}$$

where $\mathbf{K}_2$ is positive definite. This gives

$$\begin{aligned}
\dot{V}_2 &= -z_1^\top\mathbf{K}_1 z_1 - z_2^\top(\mathbf{K}_2+\mathbf{C}+\mathbf{D})z_2 \\
&\leq -z_1^\top\mathbf{K}_1 z_1 - z_2^\top\mathbf{K}_2 z_2.
\end{aligned} \tag{6.13}$$

The final control law and closed-loop system becomes

$$\begin{cases}
\dot{z}_1 = -r\mathbf{S}z_1 + z_2 + \alpha_1 - \mathbf{R}(\psi)^\top\boldsymbol{\eta}_d^s(\omega + v_s) \\
\mathbf{M}\dot{z}_2 = -z_1 - (\mathbf{K}_2+\mathbf{C}+\mathbf{D})z_2 \\
\dot{s} = v_s(t,s) + \mu\dfrac{\boldsymbol{\eta}_d^{s\top}}{|\boldsymbol{\eta}_d^s|}\mathbf{R}(\psi)z_1 \\
\dot{V}_2 \leq -z_1^\top\mathbf{K}_1 z_1 - z_2^\top(\mathbf{K}_2)z_2.
\end{cases} . \tag{6.13}$$

6.2 Thrust allocation

Given a set of commanded forces and moments from the controller, the thrust allocation algorithm assigns a normalized force vector to each individual thruster.

CES1 has two Voith-Schneider propellers (VSPs) and a bow tunnel thruster.The thruster location and data are given in Figure 6.1 and Table 6.1. Note that the angles are defined as 0 pointing towards the bow with a clockwise rotation giving a positive angle.

Table 6.1: Thruster data.

Thruster	$l_{i,x}$ [m]	$l_{i,y}$ [m]	α [rad]	F_{max} [N]	α_{min} [rad]	α_{max} [rad]
1	-0.4574	-0.055	α_1	1.03	-π	π
2	-0.4574	0.055	α_2	1.03	-π	π
3	0.3875	0	$\frac{\pi}{2}$	2.629	-	-

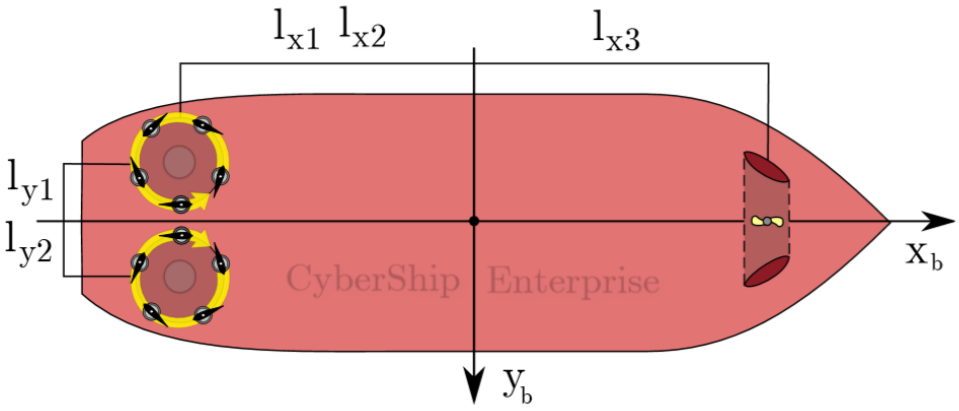


Figure 6.1: CSE1 with positive directions and thruster locations.

The generalized commanded forces and moments τ are related to the thruster setpoints by

$$\boldsymbol{\tau} = \sum_{i=1}^{3} \tau_i = \mathbf{Bf} = \mathbf{BKu}, \tag{6.13}$$

where $\mathbf{f}$ is thrust vector, $\mathbf{B}$ is the thrust configuration matrix, $\mathbf{K}$ is a scaling gain matrix, and $\mathbf{u}$ is the (normalized) thruster forces. For the VSPs, The normalized thrust vector must satisfy $u_i \in [0, 1],\ i = 1,\ 2$. For the bow thruster $u_3 \in [-1, 1]$. Assuming the linear mapping $F_i = k_i u_i$ for the thrust produced by thruster i, it follows from these u_i and the tabulated max thruster forces that $k_1 = k_2 = 1.03N$ N and $k_3 = 2.629N$.

6.2.1 Thruster allocation in rectangular coordinates

For the thrust mapping in rectangular coordinates, we can instead choose to decompose each thruster into its x- and y- components and work in rectangular cooordinates. This gives us the thrust vector

$$\mathbf{f} = \begin{bmatrix} f_{1,x} & f_{1,y} & f_{2,x} & f_{2,y} & f_3 \end{bmatrix}^\top. \tag{6.13}$$

For the VSPs, we can then calculate the corresponding angle α_i as

$$\alpha_i = \tan^{-1} \frac{f_{i,y}}{f_{i,y}}, \tag{6.13}$$

while the thrust input $u_i \in [0, 1]$ becomes

$$u_i = \left| \frac{\sqrt{f_{i,x}^2 + f_{i,y}^2}}{F_{i,max}} \right|. \tag{6.13}$$

For the bow tunnel thruster, $\mathbf{u}$ is simply calculated as

$$u_3 = \frac{f_3}{F_{3,max}}. \tag{6.13}$$

By decomposing the thrust vector of a VSP and regarding it as two independent thrusters, we can extend the thrust configuration matrix into a 3×5 matrix that is not dependent on $\boldsymbol{\alpha}$. This gives us

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ -l_{1,y} & l_{1,x} & -l_{2,y} & l_{2,x} & l_{3,x} \end{bmatrix}, \tag{6.13}$$

so we can write the thrust mapping as

$$\boldsymbol{\tau} = \mathbf{B}\mathbf{f} = \mathbf{B}\mathbf{K}\mathbf{u}, \tag{6.13}$$

where $\mathbf{u}$ and $\mathbf{K}$ are given as:

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 \\ 0 & k_1 & 0 & 0 & 0 \\ 0 & 0 & k_2 & 0 & 0 \\ 0 & 0 & 0 & k_2 & 0 \\ 0 & 0 & 0 & 0 & k_3 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{1,x} \\ u_{1,y} \\ u_{2,x} \\ u_{2,y} \\ u_3 \end{bmatrix}. \tag{6.13}$$

6.2.2 Moore-Penrose pseudo inverse

Here, the CSE1 is overactuated and the inverse matrix can not be found using basic linear transformation. In order to find the (normalized) thruster forces, Moore-Penrose pseudo inverse is used. The pseudo inverse of $\mathbf{B}$ matrix is defined as

$$\mathbf{B}^{\dagger} = \mathbf{B}^{\top}(\mathbf{B}\mathbf{B}^{\top})^{-1}. \tag{6.13}$$

Then the (normalized) thruster forces is given by

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{B}^{\dagger}\boldsymbol{\tau}. \tag{6.13}$$

Chapter 7

Results

7.1 Simulation overview

Eight simulation cases are designed to test the proposed methods, including overtaking, head-on, crossing, and complex situations. The path length and COLREGs compliance are considered for assessment. The parameters for the control design model and the workspace are given in tables 7.1 and 7.2. Tuning parameters for the navigation function method and the stream function method are given in tables 7.3 and 7.4. All obstacles in the simulation have constant speeds following straight-line trajectories.

Table 7.1: CSE1 parameters.

Parameter	Value	Parameter	Value	Parameter	Value
m	14.11	$X_{\dot{u}}$	-2	X_u	-0.6555
I_z	1.76	$Y_{\dot{v}}$	-10	Y_V	-1.33
x_g	0.0375	$Y_{\dot{r}}$	0	Y_R	-7.25
y_g	0.0	$N_{\dot{r}}$	-1	N_v	0.0
				N_r	-1.9

Table 7.2: Workspace representation.

Parameter	Value	Parameter	Value	Parameter	Value
L_y	$20m$	N_y	100	d_y	$0.02m$
L_x	$20m$	N_x	100	d_x	$0.02m$

Table 7.3: Parameters for the navigation function method.

Parameter \ Scenario	1	2	3	4	5	6	7	8
κ	1.6	1.6	2.2	2.2	3.8	3.8	4.8	5.4
γ_1	0.05							
n_h	5							
n_r	5							
R	2							
r_i	1.5							

Table 7.4: Parameters for the stream function method.

Parameter \ Scenario	1	2	3	4	5	6	7	8
C_{v_i}	$\begin{bmatrix}0.1\end{bmatrix}$	$\begin{bmatrix}0.1\end{bmatrix}$	$\begin{bmatrix}-0.1\end{bmatrix}$	$\begin{bmatrix}0.05\\0.2\\0.1\end{bmatrix}$	$\begin{bmatrix}0.1\\0.1\\0.1\end{bmatrix}$	$\begin{bmatrix}-0.1\\0.15\\-0.2\end{bmatrix}$	$\begin{bmatrix}0.1\\-0.1\\-0.1\\0.15\\0.2\end{bmatrix}$	$\begin{bmatrix}0.1\\-0.1\\-0.1\\-0.1\\0.2\end{bmatrix}$
γ_2	0.2							
n_h	5							
n_r	5							
R	2							
r_i	1.5							

7.2 Scenario 1 - Overtaking

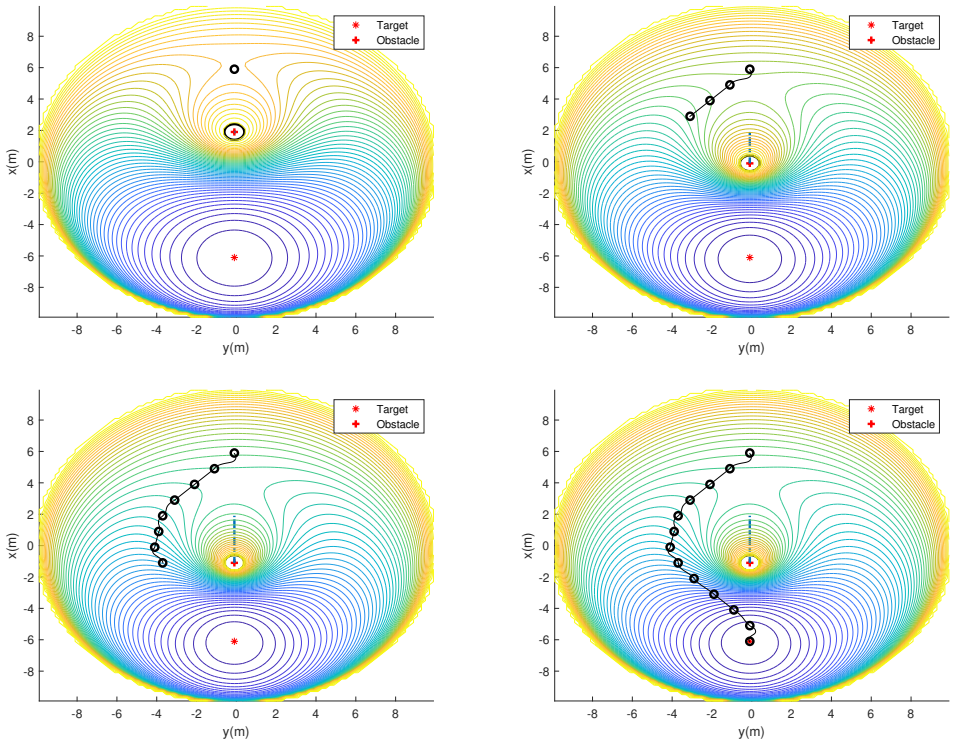


Figure 7.1: Results of scenario 1 using navigation function method.

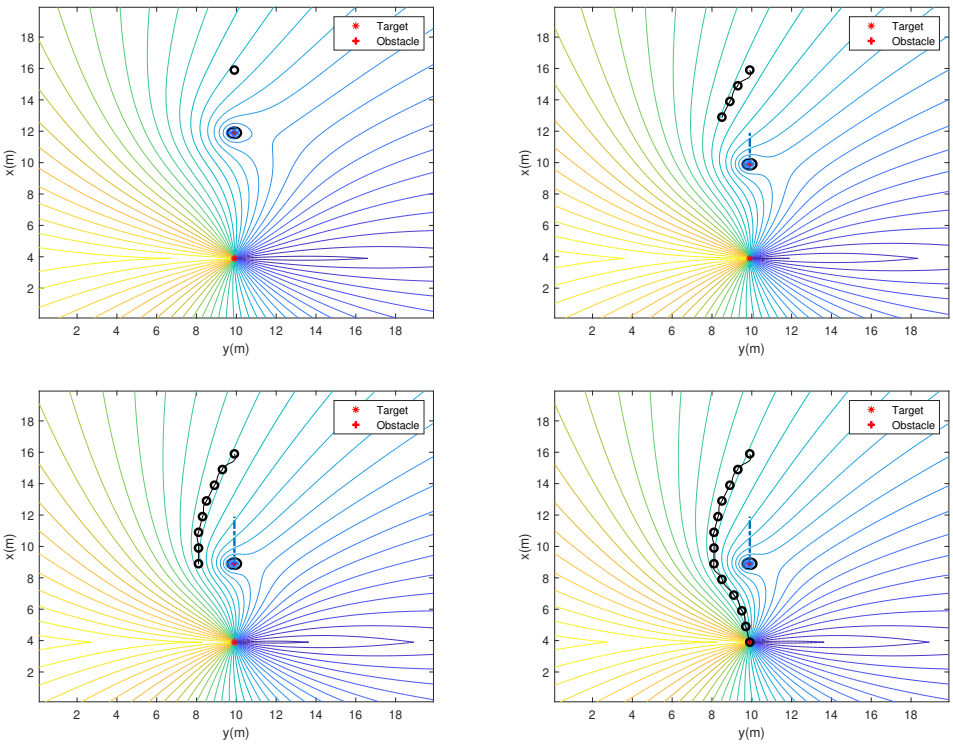


Figure 7.2: Results of scenario 1 using stream function method.

7.3 Scenario 2 - Head-on situation

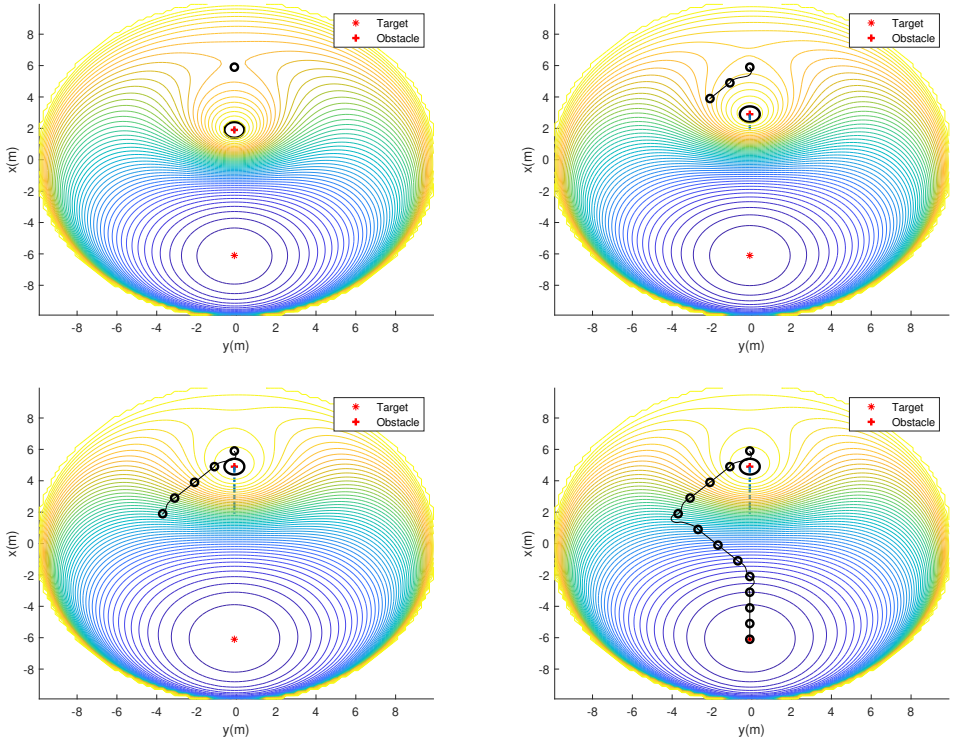


Figure 7.3: Results of scenario 1 using navigation function method.

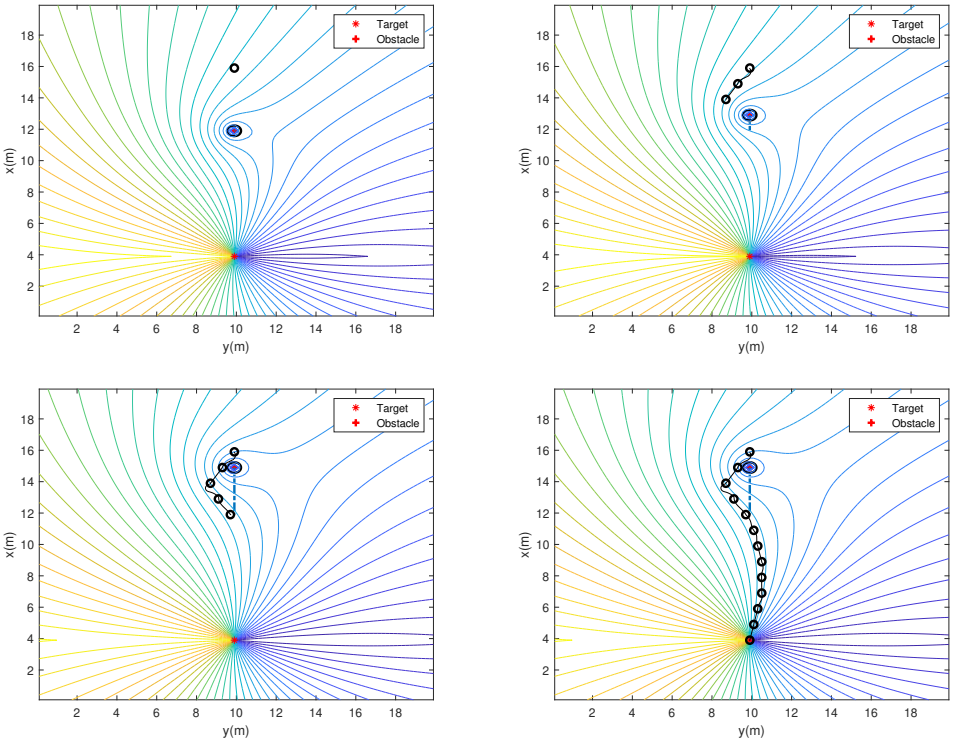


Figure 7.4: Results of scenario 2 using stream function method.

7.4 Scenario 3 - Crossing situation

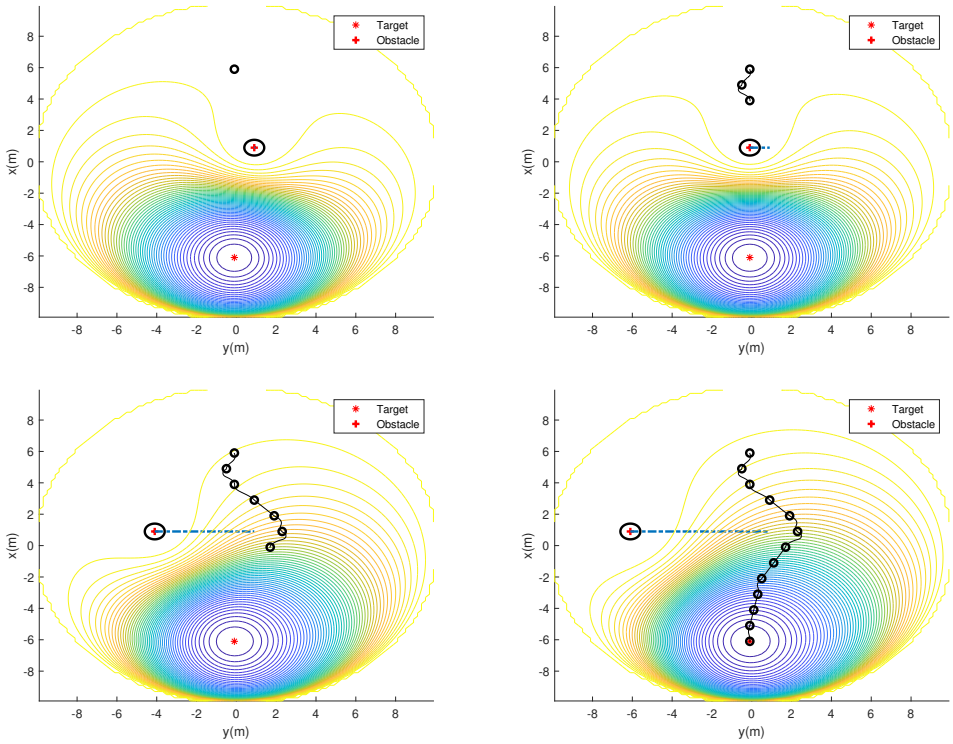


Figure 7.5: Results of scenario 3 using navigation function method.

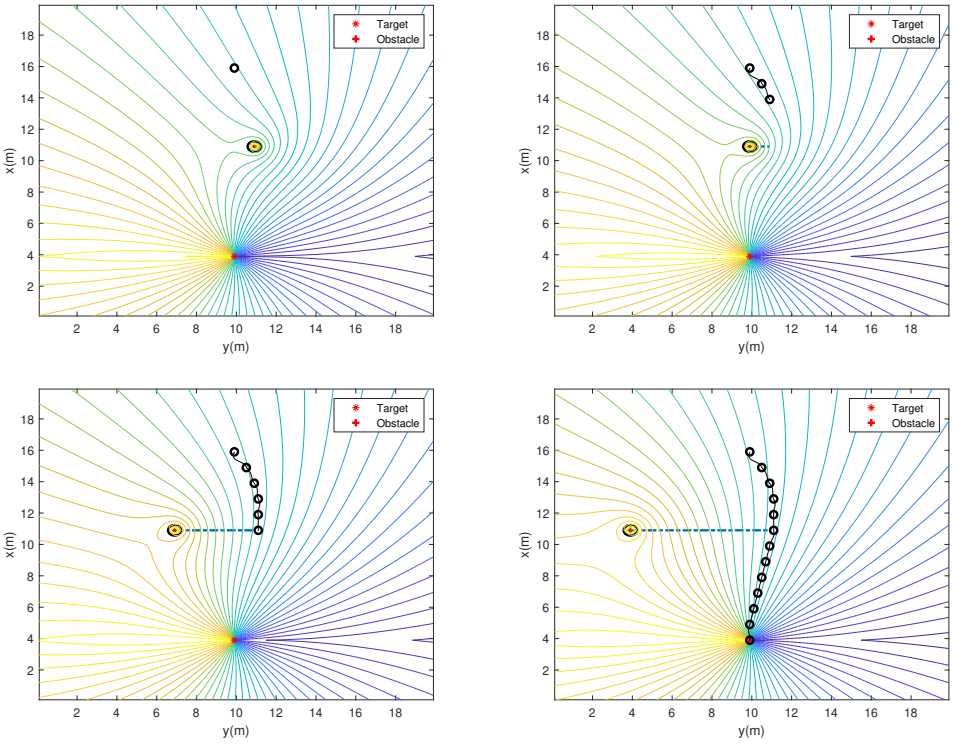


Figure 7.6: Results of scenario 3 using stream function method.

7.5 Scenario 4 - Head-on situation

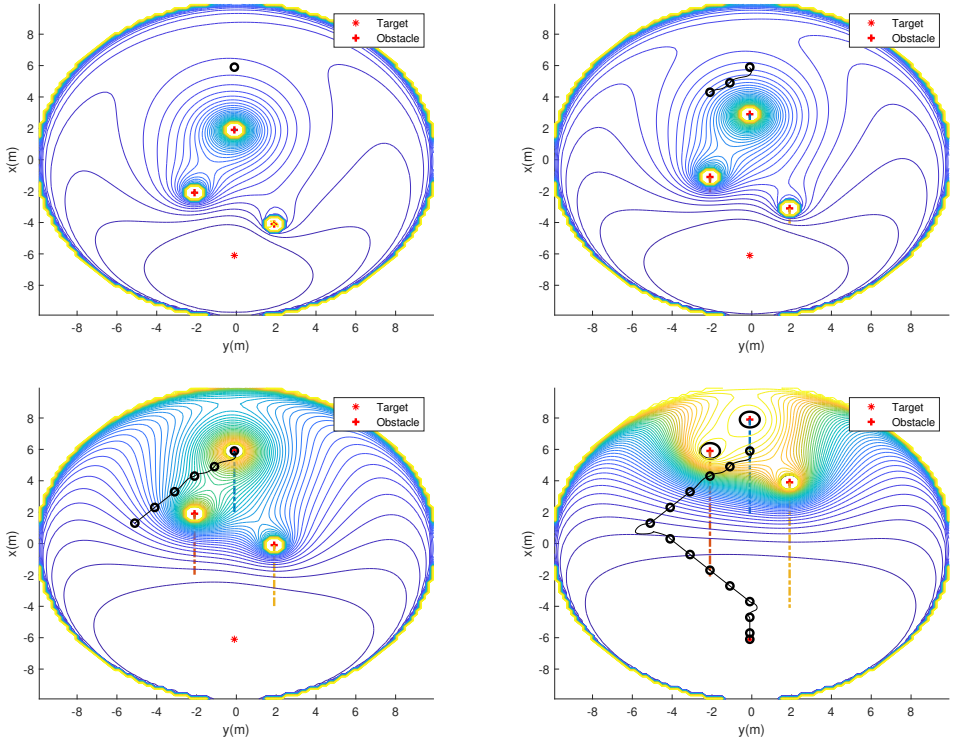


Figure 7.7: Results of scenario 4 using navigation function method.

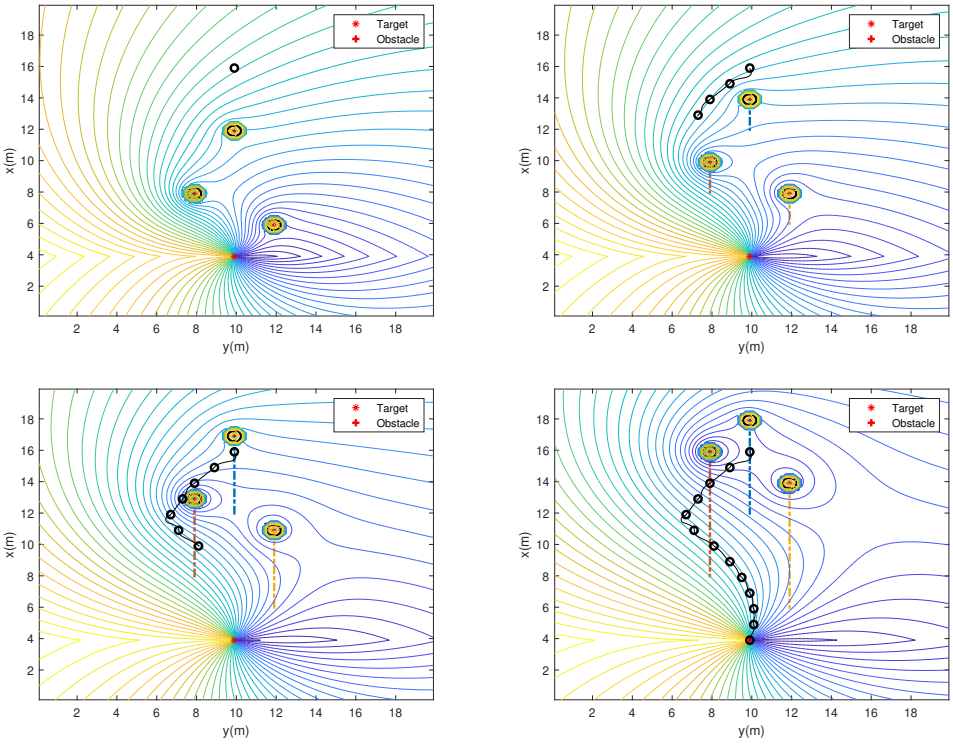


Figure 7.8: Results of scenario 4 using stream function method.

7.6 Scenario 5 - Crossing situation

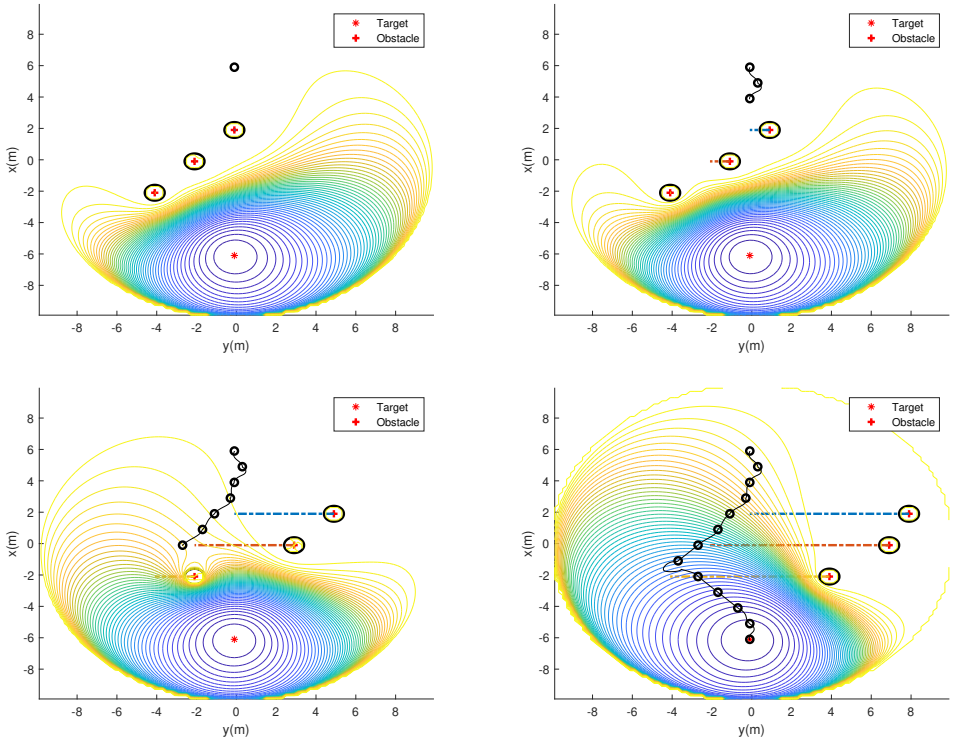


Figure 7.9: Results of scenario 5 using navigation function method.

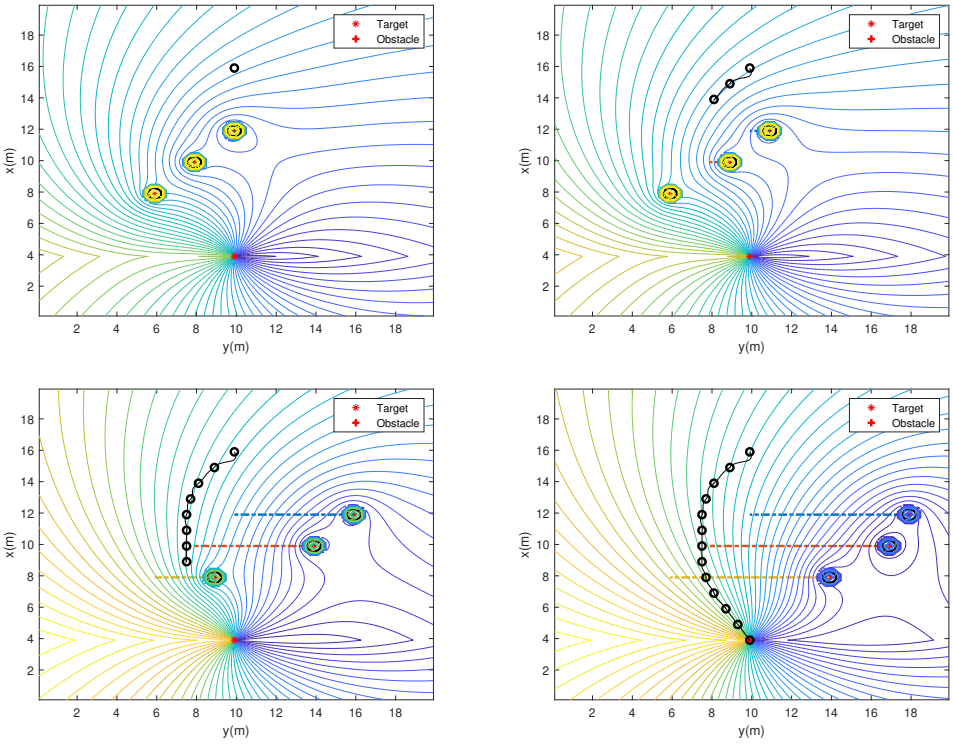


Figure 7.10: Results of scenario 5 using stream function method.

7.7 Scenario 6 - Crossing situation

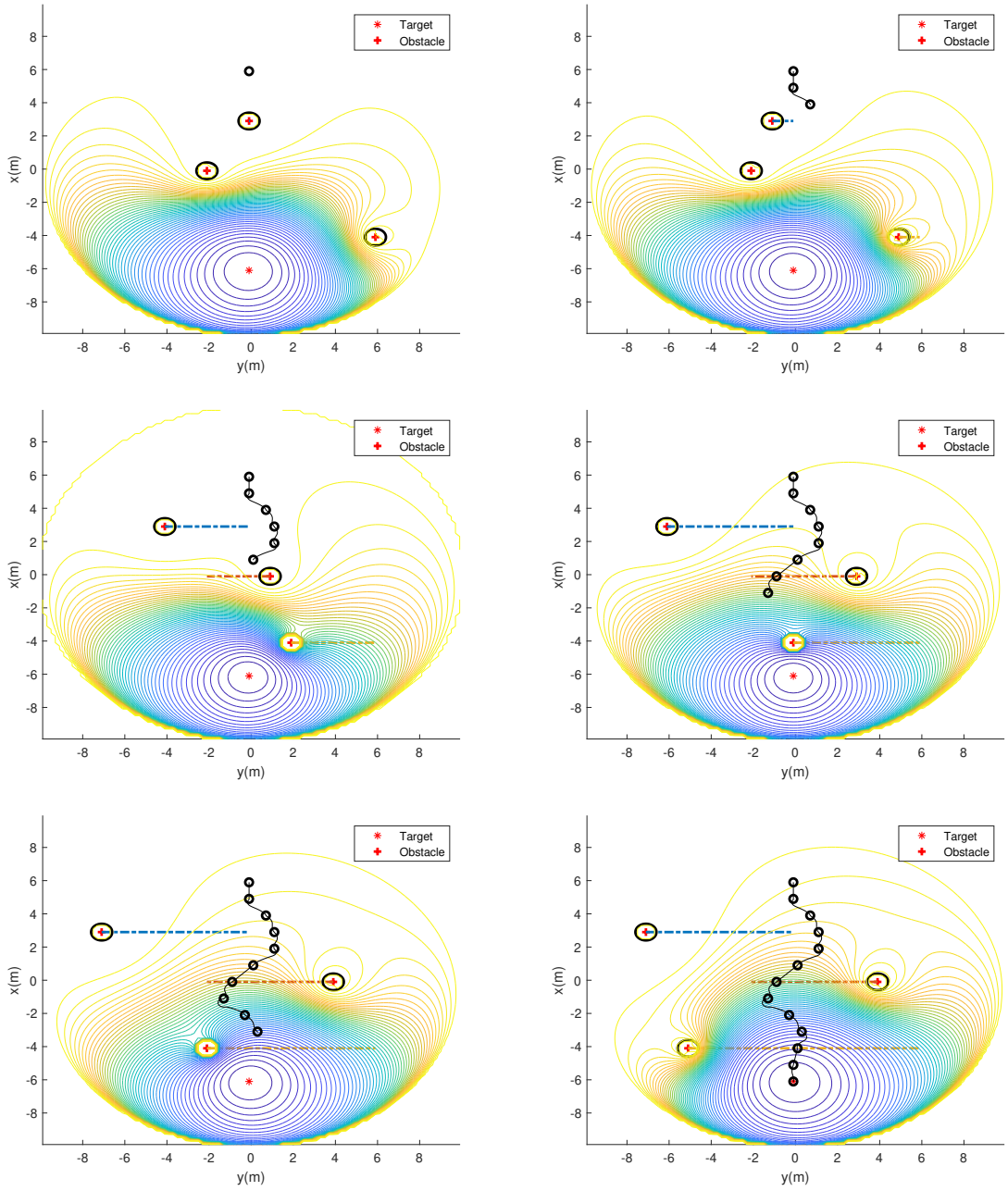


Figure 7.11: Results of scenario 6 using navigation function method.

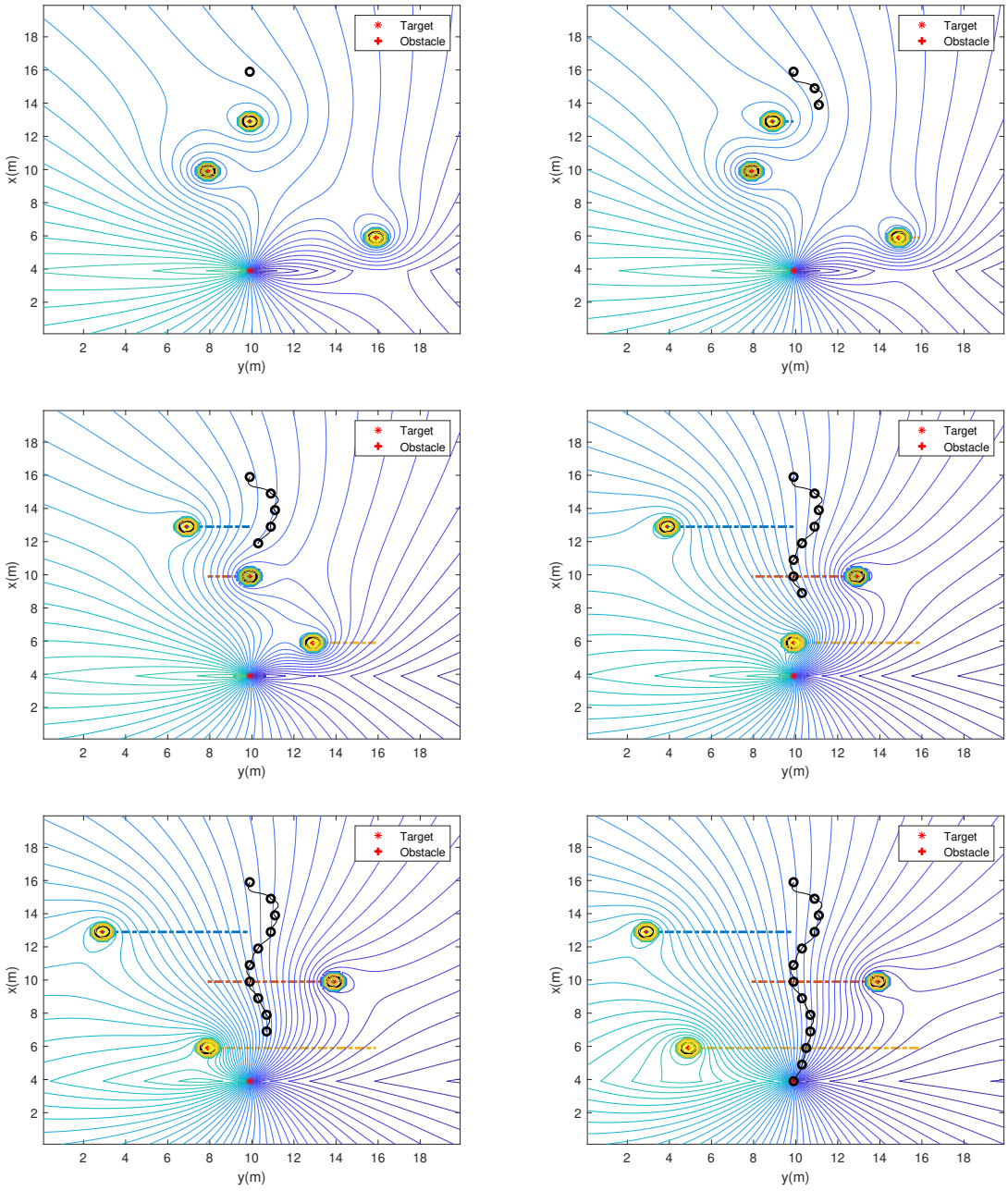


Figure 7.12: Results of scenario 6 using stream function method.

7.8 Scenario 7 - Complex situation

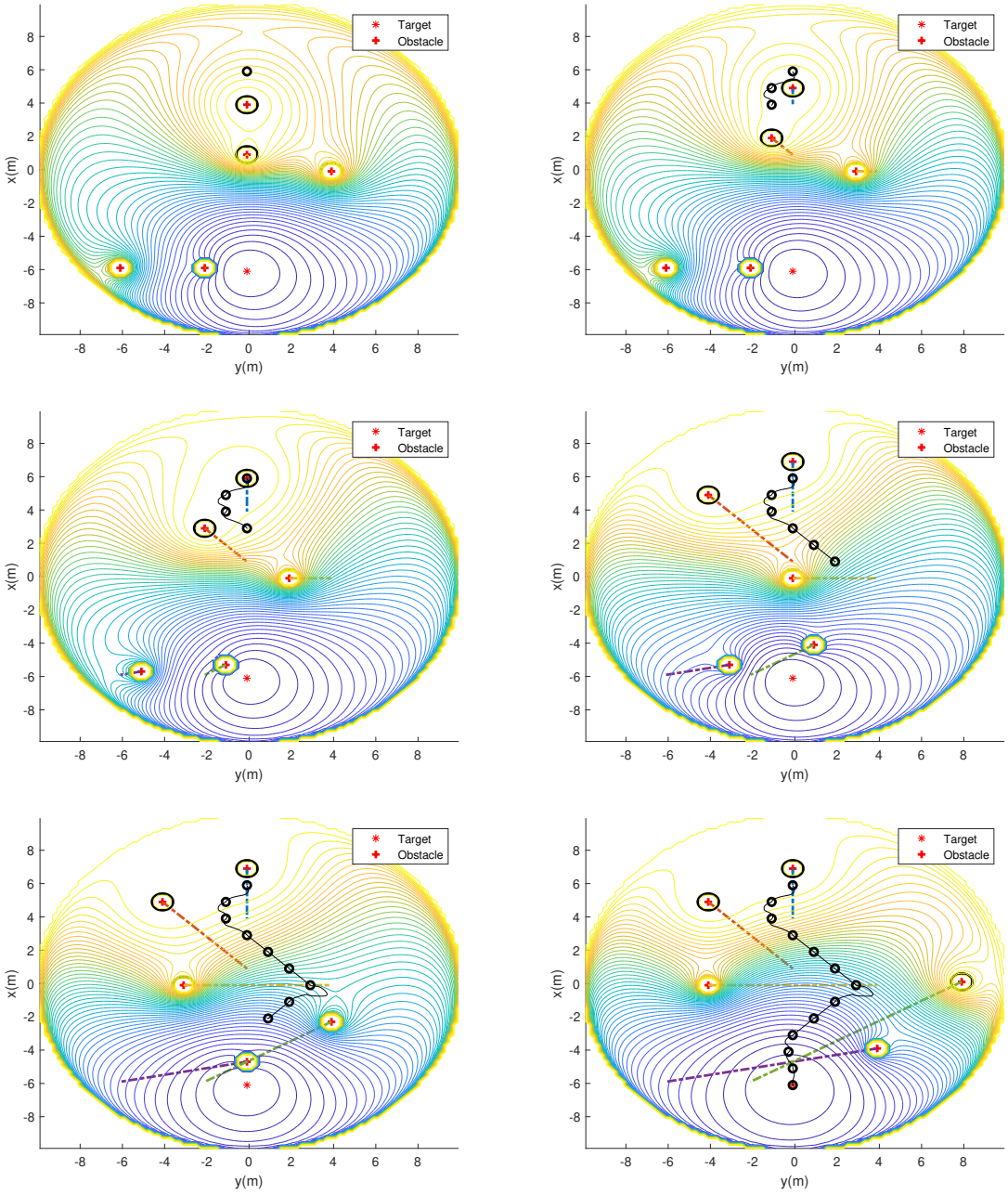


Figure 7.13: Results of scenario 7 using navigation function method.

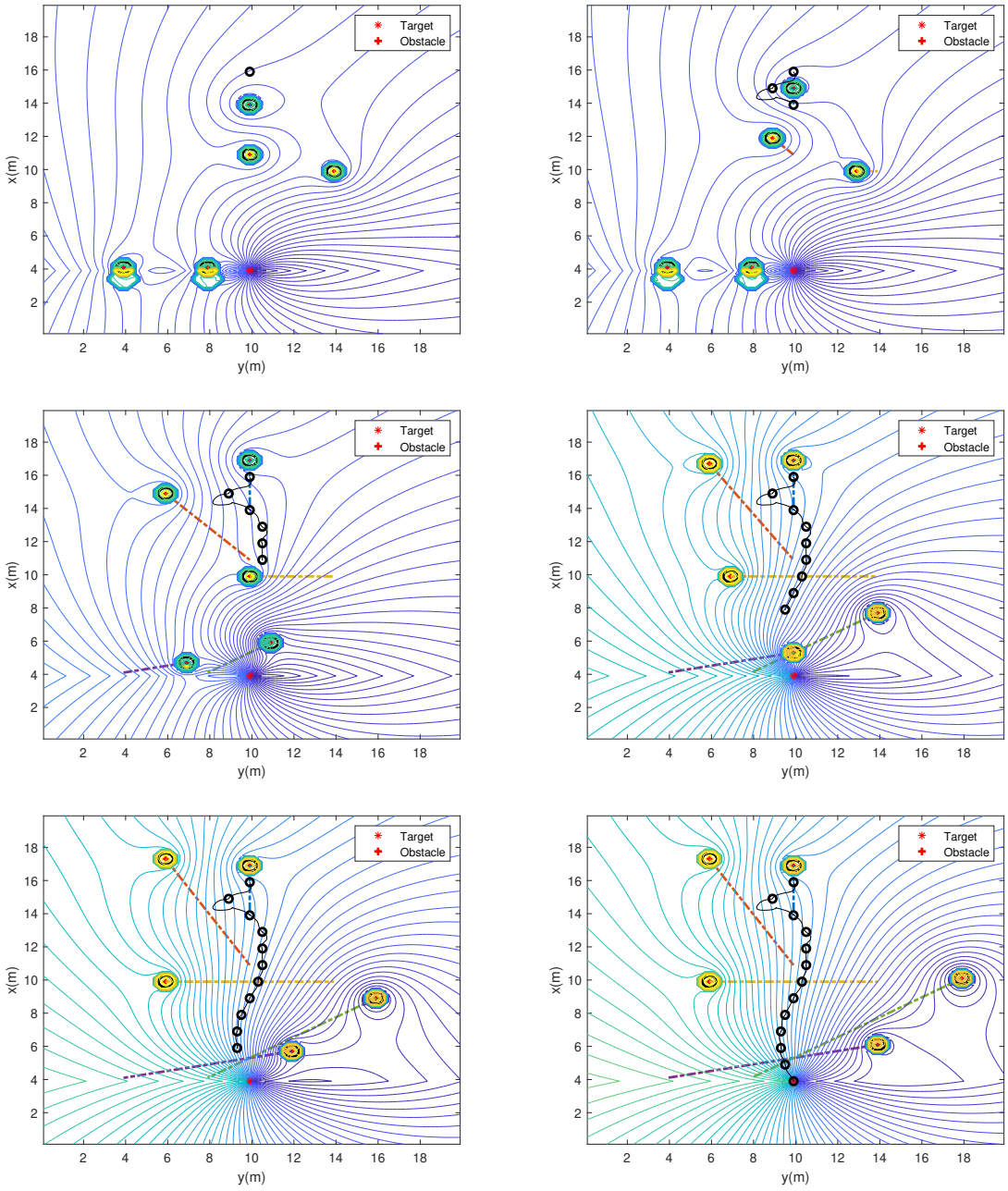


Figure 7.14: Results of scenario 7 using stream function method.

7.9 Scenario 8 - Complex situation

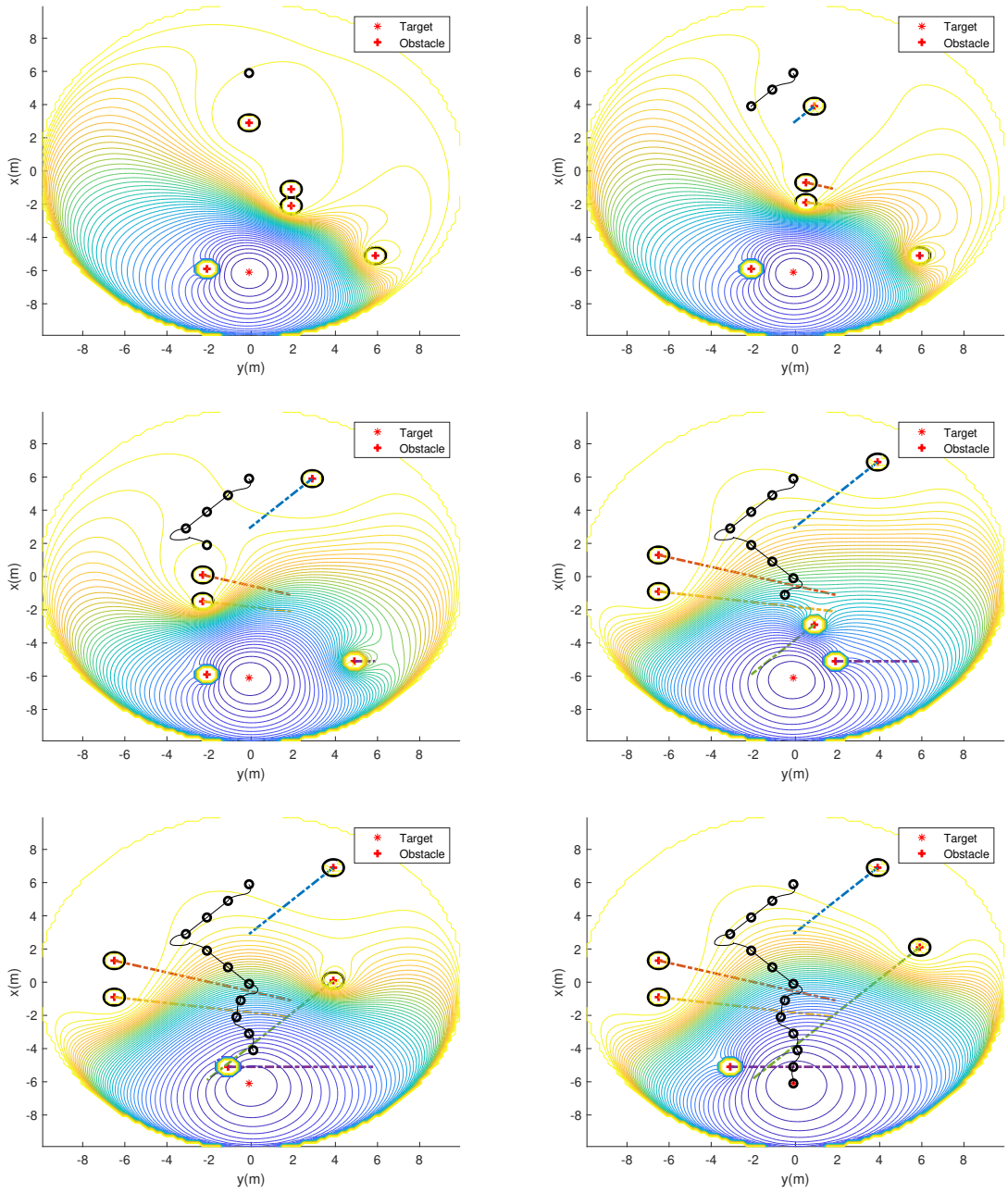


Figure 7.15: Results of scenario 8 using navigation function method.

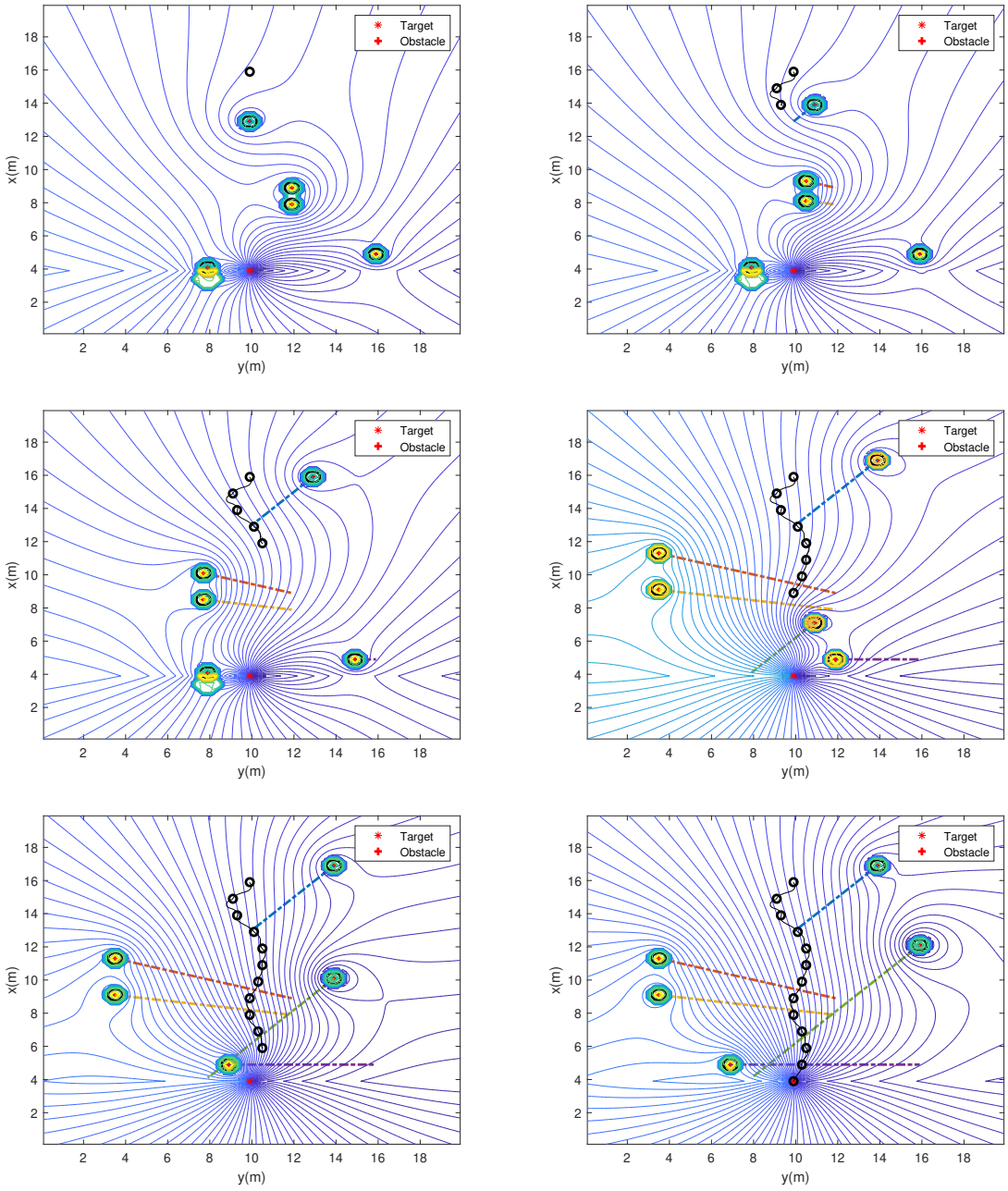


Figure 7.16: Results of scenario 8 using stream function method.

7.10 Discussion

Simulation results are summarized in Table 7.5, which compares two methods in terms of path length and COLREGs compliance. Additional results using original navigation function and original stream function methods are given in figures 8.5 to 8.14. The original navigation function method means choosing waypoints without using the decision-making module and $\delta U(\mathbf{p})$. The original stream function method means choosing waypoints without adding vortex flows.

Both of the original methods determine waypoints that violate COLREGs rules, see e.g. figures 8.7 and 8.12. Even worse, using the original stream function method in some cases leads to collision, see figures 8.8 and 8.12. Compared to the original methods, the proposed waypoint generation methods are able to

guide a marine robot to its target with both collision avoidance and COLREGs compliance.

In Table 7.5, the path length using the navigation function method is longer in all simulation cases. The reason is that, using Equation (5.1), an obstacle affects the potential field globally (means the whole workspace), i.e., an obstacle influences the potential of the whole workspace. Based on such potential field, a robot starts to avoid collision from its initial position. Therefore, waypoint generation is conservative, and gives a longer path. On the contrary, an obstacle only affects the flow field locally, i.e., an obstacle curves the streamlines within a small range, e.g. see Figure 5.11a. A robot following a constant streamline then only avoid collision when it gets close to an obstacle. Hence, the stream function method leads to a shorter path. But this can make the robot get too close to an obstacle.

The stream function method also gives a smoother path, since it models a potential flow. While the navigation function method can give a sudden change in heading, and it is not desired for an underactuated robot, see figures 7.7, 7.9, and 7.13. In addition, based on experience in tuning, it is easier and more convenient to tune C_{v_i} to obtain a desired path using the stream function method.

The advantage of the navigation function method is the guarantee of global convergence. For the stream function method, after adding vortex flows, not all streamlines end at target position, see figures 7.12, 7.14, and 7.16. The global convergence could not be guaranteed.

Table 7.5: Simulation results.

Scenario	Path length (m)		COLREGs compliance	
	Navigation function	Stream function	Navigation function	Stream function
1	15.1459	12.7948	Yes	Yes
2	15.1153	12.9042	Yes	Yes
3	13.8484	12.4396	Yes	Yes
4	16.7435	14.4477	Yes	Yes
5	15.2661	13.5094	Yes	Yes
6	14.2345	13.2100	Yes	Yes
7	15.9217	13.6886	Yes	Yes
8	15.3928	13.2343	Yes	Yes

Chapter 8

Conclusion

This thesis has presented methods for autonomous guidance, stepwise path planning, and path-following control with anti-collision and COLREGs compliance. The system has been validated through simulations.

In the simulation studies, eight scenarios were presented, including overtaking, head-on, crossing, and complex situations. Two waypoint generation methods, i.e., the navigation function method and the stream function method were compared. The results showed that the navigation function method guarantees global convergence, while the stream function method gives shorter and smoother path.

Each waypoint generation method was integrated with a path generation algorithm, which formulates a quadratic programming problem to minimize the path length between two waypoints. The marine control system was designed to perform path following.

In the future work, more COLREGs rules or other regulations can be used to assess proposed methods. In this thesis, only three rules were considered. By incorporating more rules, more complex situations can be studied.

Also, the decision-making module in the navigation function method was simplified, with only the relative position between the robot and obstacles being considered. Therefore, it only works for limited situations. For more general applications, the relative position between the robot and obstacles, the velocity of the robot, the velocities of obstacles, and other information can be included when making a decision.

Another direction is to test the system under uncertainties and environmental disturbances. In the thesis, the trajectories of the obstacles were known and the environmental loads caused by wind, waves, and currents were ignored. It would be interesting to add a motion prediction module to estimate obstacles' trajectories, and assess the robustness of proposed methods.

Besides, the collision avoidance problem can be treated as a cooperative control problem of a multi-robot system, where inter-robots information exchange is available. Each robot should cooperate together and avoid collision with COLREGs compliance.

Bibliography

Axler, S., Bourdon, P., Wade, R., 2013. Harmonic function theory. volume 137. Springer Science & Business Media.

Campbell, S., Naeem, W., Irwin, G.W., 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. Annual Reviews in Control 36, 267–283.

Chen, Y.b., Luo, G.c., Mei, Y.s., Yu, J.q., Su, X.l., 2016. Uav path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science 47, 1407–1420.

Chiang, H.T., Malone, N., Lesser, K., Oishi, M., Tapia, L., 2015. Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 2347–2354.

Commandant, U., 1999. International regulations for prevention of collisions at sea, 1972 (72 colregs). US Department of Transportation, US Coast Guard, COMMANDANT INSTRUCTION M 16672.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to algorithms. MIT press.

Currie, I.G., 2016. Fundamental mechanics of fluids. CRC press.

Dimarogonas, D.V., Loizou, S.G., Kyriakopoulos, K.J., Zavlanos, M.M., 2006. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. Automatica 42, 229–243.

Dryanovski, I., Morris, W., Xiao, J., 2010. Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 1553–1559.

Elfes, A., 1989. Using occupancy grids for mobile robot perception and navigation. Computer 22, 46–57.

Fossen, T.I., 2011. Handbook of marine craft hydrodynamics and motion control. John Wiley & Sons.

Gao, F., Wu, W., Lin, Y., Shen, S., 2018a. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 344–351.

Gao, F., Wu, W., Pan, J., Zhou, B., Shen, S., 2018b. Optimal time allocation for quadrotor trajectory generation, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 4715–4722.

Ge, S.S., Cui, Y.J., 2002. Dynamic motion planning for mobile robots using potential field method. Autonomous robots 13, 207–222.

He, Y., Jin, Y., Huang, L., Xiong, Y., Chen, P., Mou, J., 2017. Quantitative analysis of colreg rules and seamanship for autonomous collision avoidance at open sea. Ocean Engineering 140, 281–291.

Hu, C., Wu, X., Liang, Q., Wang, Y., 2007. Autonomous robot path planning based on swarm intelligence and stream functions, in: International Conference on Evolvable Systems, Springer. pp. 277–284.

Huang, L., 2009. Velocity planning for a mobile robot to track a moving target—a potential field approach. Robotics and Autonomous Systems 57, 55–63.

Huang, S., Teo, R.S.H., Tan, K.K., 2019. Collision avoidance of multi unmanned aerial vehicles: A review. Annual Reviews in Control .

Huang, Y., Chen, L., Chen, P., Negenborn, R.R., van Gelder, P., 2020. Ship collision avoidance methods: State-of-the-art. Safety science 121, 451–473.

IMO, 2020. Convention on the international regulations for preventing collisions at sea, 1972 (colregs). Website [Accessed March 27, 2020]. http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx.

Ingrand, F., Ghallab, M., 2017. Deliberation for autonomous robots: A survey. Artificial Intelligence 247, 10–44.

Johansen, T.A., Fossen, T.I., 2013. Control allocation—a survey. Automatica 49, 1087–1103.

Johansen, T.A., Perez, T., Cristofaro, A., 2016. Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. IEEE transactions on intelligent transportation systems 17, 3407–3422.

Joy, K.I., 2000a. Bernstein polynomials. visualization and graphics research group, department of computer science, university of california, davis. .

Joy, K.I., 2000b. A matrix formulation of the cubic bezier curve. matrix 1, 3.

Kallem, V., Komoroski, A.T., Kumar, V., 2011. Sequential composition for navigating a nonholonomic cart in the presence of obstacles. IEEE Transactions on Robotics 27, 1152–1159.

Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots, in: Autonomous robot vehicles. Springer, pp. 396–404.

Kim, M.H., Sur, J., et al., 2011. A stream function approach to design obstacle avoidance path planning algorithm for autonomous underwater vehicles, in: The Twenty-first International Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers.

Knædal, M., 2019. Stepwise path-generation using bézier curves. specilization project, NTNU .

Knædal, M., 2020. Stepwise path-generation using bézier curves. Technical note, Rev.B .

Koditschek, D.E., Rimon, E., 1990. Robot navigation functions on manifolds with boundary. Advances in applied mathematics 11, 412–442.

Kuwata, Y., Wolf, M.T., Zarzhitsky, D., Huntsberger, T.L., 2013. Safe maritime autonomous navigation with colregs, using velocity obstacles. IEEE Journal of Oceanic Engineering 39, 110–119.

Lee, S.M., Kwon, K.Y., Joh, J., 2004. A fuzzy logic for autonomous navigation of marine vehicles satisfying colreg guidelines. International Journal of Control, Automation, and Systems 2, 171–181.

Li, G., Yamashita, A., Asama, H., Tamura, Y., 2012. An efficient improved artificial potential field based regression search method for robot path planning, in: 2012 IEEE International Conference on Mechatronics and Automation, IEEE. pp. 1227–1232.

Loizu, S., Dimarogonas, D.V., Kyriakopoulos, K.J., 2004. Decentralized feedback stabilization of multiple nonholonomic agents, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, IEEE. pp. 3012–3017.

Lyu, H., Yin, Y., 2017. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field, in: 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), IEEE. pp. 351–357.

Lyu, H., Yin, Y., 2019. Colregs-constrained real-time path planning for autonomous ships using modified artificial potential fields. The Journal of Navigation 72, 588–608.

Mac, T.T., Copot, C., Tran, D.T., De Keyser, R., 2016. Heuristic approaches in robot path planning: A survey. Robotics and Autonomous Systems 86, 13–28.

Meyer-Delius, D., Beinhofer, M., Burgard, W., 2012. Occupancy grid models for robot mapping in changing environments, in: Twenty-Sixth AAAI Conference on Artificial Intelligence.

Milne-Thomson, L.M., 1996. Theoretical hydrodynamics. Courier Corporation.

Moghadam, P., Wijesoma, W.S., Feng, D.J., 2008. Improving path planning and mapping based on stereo vision and lidar, in: 2008 10th International Conference on Control, Automation, Robotics and Vision, IEEE. pp. 384–389.

Mularie, W., 2000. World geodetic system 1984–its definition and relationships with local geodetic systems. Department of Defense, NIMA USA .

Nam, T.H., Shim, J.H., Cho, Y.I., 2017. A 2.5 d map-based mobile robot localization via cooperation of aerial and ground robots. Sensors 17, 2730.

Patle, B., Pandey, A., Parhi, D., Jagadeesh, A., et al., 2019. A review: On path planning strategies for navigation of mobile robot. Defence Technology .

Pedersen, M.D., Fossen, T.I., 2012. Marine vessel path planning & guidance using potential flow. IFAC Proceedings Volumes 45, 188–193.

Perera, L., Carvalho, J., Soares, C.G., 2012. Intelligent ocean navigation and fuzzy-bayesian decision/action formulation. IEEE Journal of Oceanic Engineering 37, 204–219.

Praczyk, T., 2015. Neural anti-collision system for autonomous surface vehicle. Neurocomputing 149, 559–572.

Quan, L., Han, L., Zhou, B., Shen, S., Gao, F., 2020. Survey of uav motion planning. IET Cyber-systems and Robotics 2, 14–21.

Rahmani, A., Kosuge, K., Tsukamaki, T., Mesbahi, M., 2008. Multiple uav deconfliction via navigation functions, in: AIAA Guidance, Navigation and Control Conference and Exhibit, p. 6626.

Rimon, E., Koditschek, D.E., 1992. Exact robot navigation using artificial potential functions. Departmental Papers (ESE) , 323.

Rødseth, Ø., 2019. Why autonomous ships? Website, Norwegian Forum for Autonomous Ship (NFAS) [Accessed March 26, 2020]. `http://nfas.autonomous-ship.org/why-en.html#AUT`.

Rodseth, O., Burmeister, H., 2012. Developments toward the unmanned ship, proceedings of internatinal symposium information on ships. ISIS , 30–31.

Rødseth, Ø.J., Nordahl, H., Hoem, Å., 2018. Characterization of autonomy in merchant ships, in: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), IEEE. pp. 1–7.

Roelofsen, S., Gillet, D., Martinoli, A., 2015. Reciprocal collision avoidance for quadrotors using on-board visual detection, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 4810–4817.

Roussos, G., Dimarogonas, D.V., Kyriakopoulos, K.J., 2010. 3d navigation and collision avoidance for nonholonomic aircraft-like vehicles. International Journal of Adaptive Control and Signal Processing 24, 900–920.

Skjetne, R., 2005. The maneuvering problem. NTNU, PhD-thesis 1.

SNAME, T., 1950. Nomenclature for treating the motion of a submerged body through a fluid. The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin No , 1–5.

Sullivan, J., Waydo, S., Campbell, M., 2003. Using stream functions for complex behavior and path generation, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 5800.

Tan, F., Yang, J., Huang, J., Jia, T., Chen, W., Wang, J., 2010. A navigation system for family indoor monitor mobile robot, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 5978–5983.

Tanner, H.G., Kyriakopoulos, K.J., 2000. Nonholonomic motion planning for mobile manipulators, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), IEEE. pp. 1233–1238.

Tanner, H.G., Loizou, S., Kyriakopoulos, K.J., 2001. Nonholonomic stabilization with collision avoidance for mobile robots, in: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180), IEEE. pp. 1220–1225.

Tanner, H.G., Loizou, S.G., Kyriakopoulos, K.J., 2003. Nonholonomic navigation and control of cooperating mobile manipulators. IEEE Transactions on robotics and automation 19, 53–64.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic robotics. MIT press.

Urakubo, T., Okuma, K., Tada, Y., 2004. Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), IEEE. pp. 2428–2433.

Usenko, V., von Stumberg, L., Pangercic, A., Cremers, D., 2017. Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 215–222.

Valbuena, L., Tanner, H.G., 2012. Hybrid potential field based control of differential drive mobile robots. Journal of intelligent & robotic systems 68, 307–322.

Wang, X., Ju, Y., 2015. Application of stream function method in local route planning for uuvs, in: 2015 International Industrial Informatics and Computer Engineering Conference, Atlantis Press.

Waydo, S., Murray, R.M., 2003. Vehicle motion planning using stream functions, in: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), IEEE. pp. 2484–2491.

Woerner, K., Benjamin, M.R., Novitzky, M., Leonard, J.J., 2019. Quantifying protocol evaluation for autonomous collision avoidance. Autonomous Robots 43, 967–991.

Ye, G., Wang, H.O., Tanaka, K., 2005. Coordinated motion control of swarms with dynamic connectivity in potential flows. IFAC Proceedings Volumes 38, 136–141.

Yin, L., Yin, Y., 2008. An improved potential field method for mobile robot path planning in dynamic environments, in: 2008 7th World Congress on Intelligent Control and Automation, IEEE. pp. 4847–4852.

Yun, X., Tan, K.C., 1997. A wall-following method for escaping local minima in potential field based motion planning, in: 1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97, IEEE. pp. 421–426.

Zhao, D., Yi, J., 2006. Robot planning with artificial potential field guided ant colony optimization algorithm, in: International Conference on Natural Computation, Springer. pp. 222–231.

Zhou, B., Gao, F., Wang, L., Liu, C., Shen, S., 2019. Robust and efficient quadrotor trajectory generation for fast autonomous flight. IEEE Robotics and Automation Letters 4, 3529–3536.

Appendix

Simulink diagrams

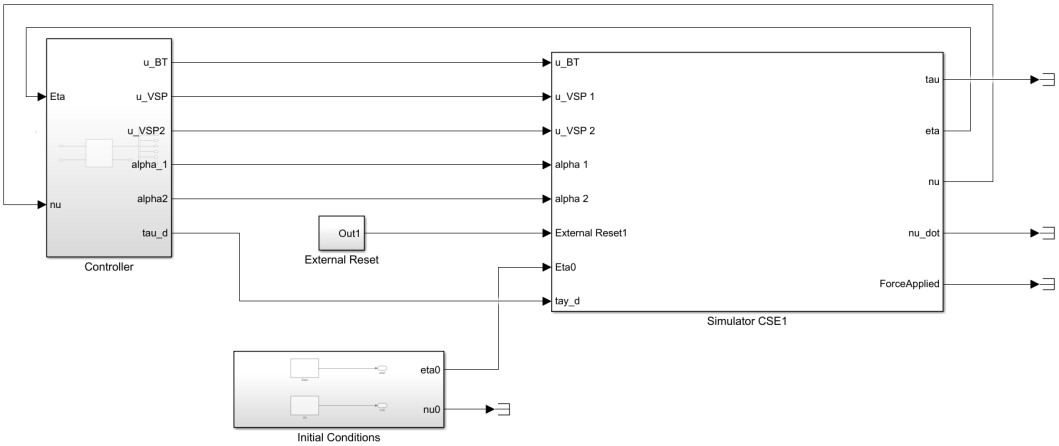


Figure 8.1: System overview.

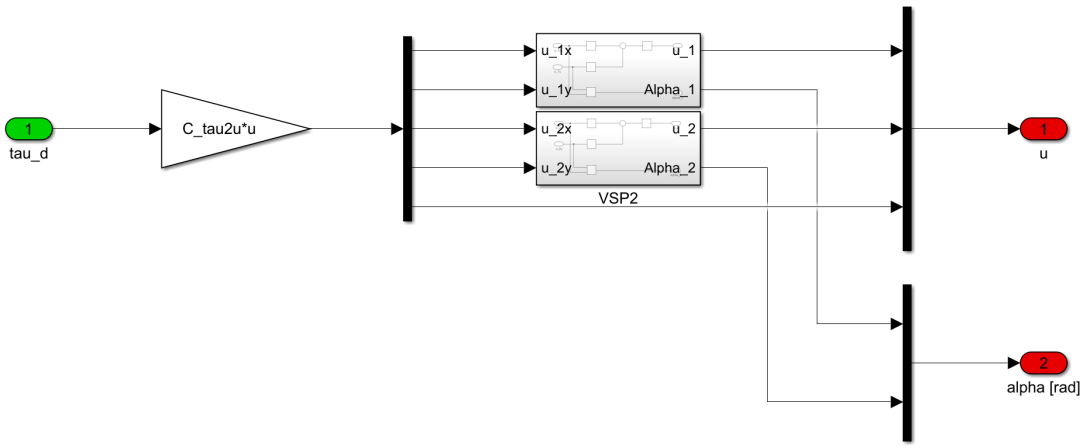


Figure 8.2: Thrust allocation.

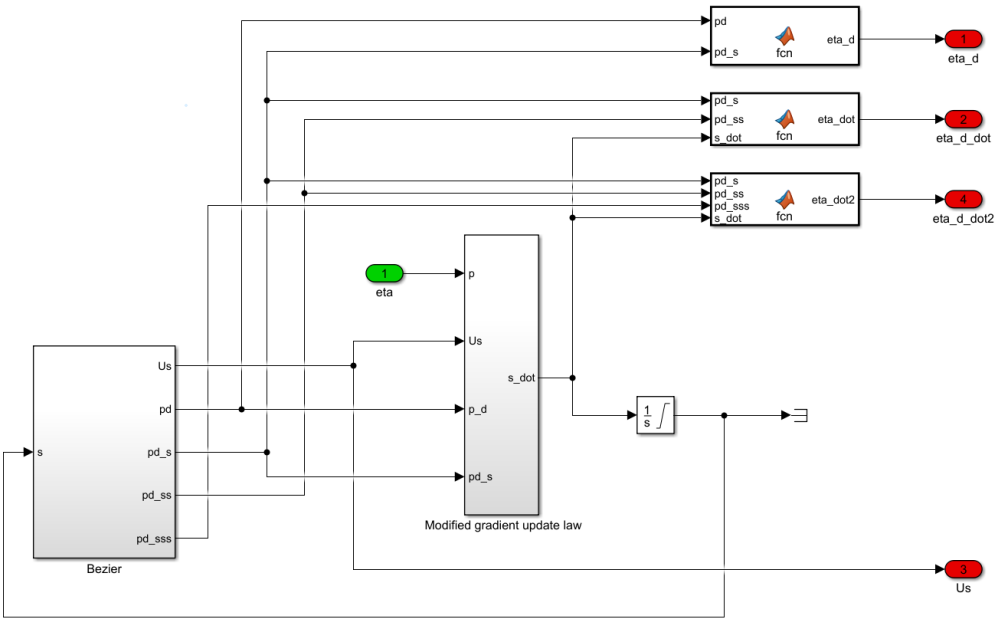


Figure 8.3: Path generation and update law.

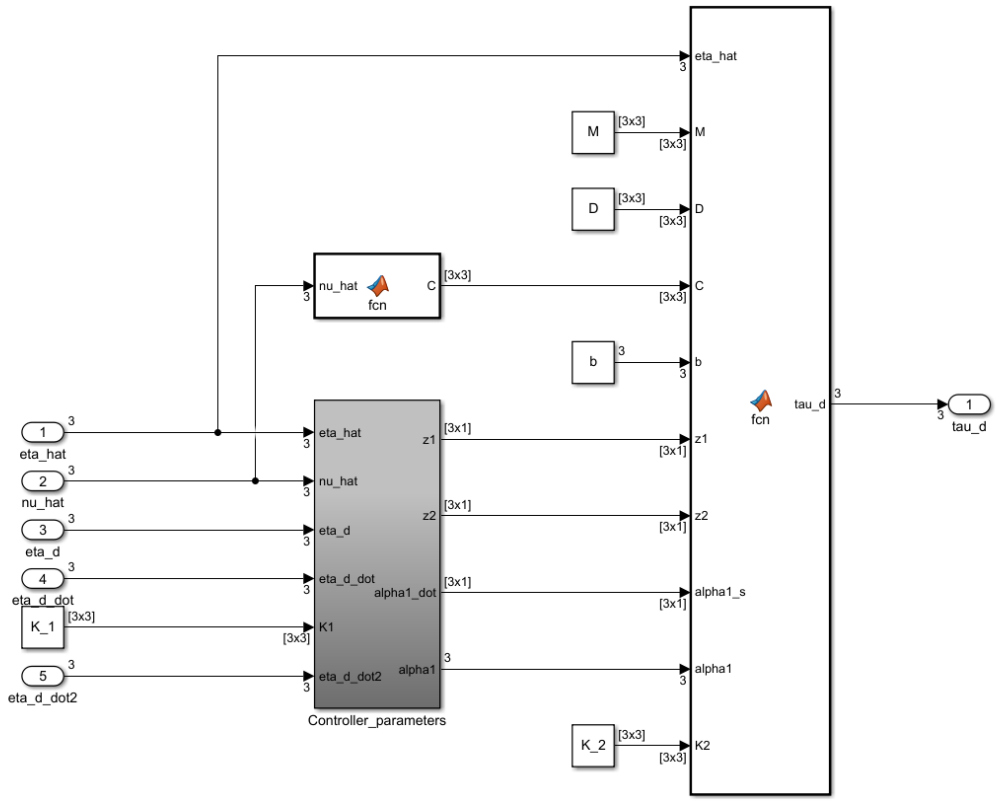


Figure 8.4: Controller.

Additional results

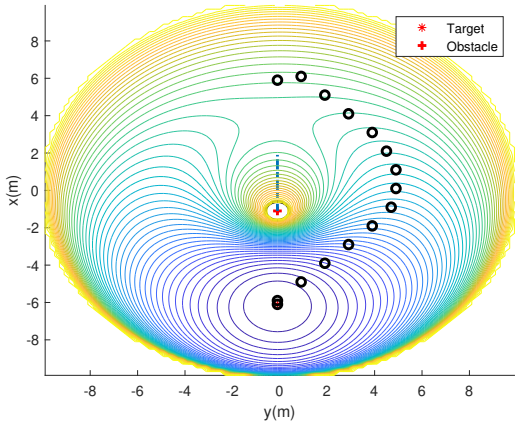


Figure 8.5: Scenario 1: waypoint generation using original navigation function method.

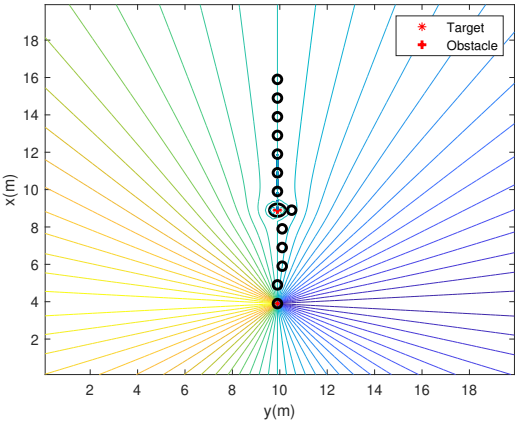


Figure 8.6: Scenario 1: waypoint generation using original stream function method.

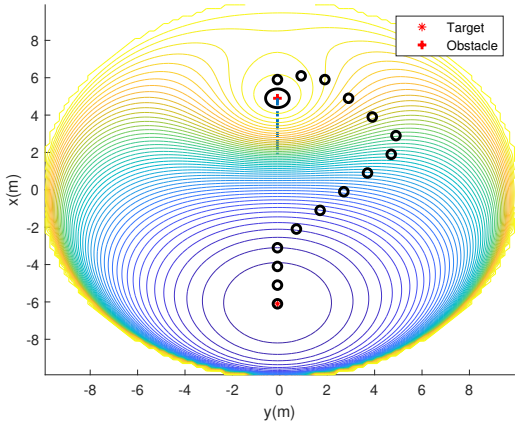


Figure 8.7: Scenario 2: waypoint generation using original navigation function method.

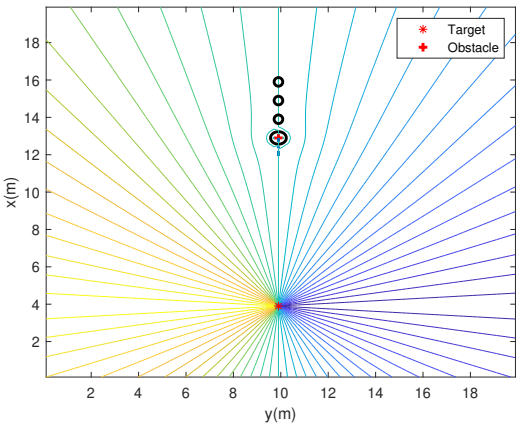


Figure 8.8: Scenario 2: waypoint generation using original stream function method.

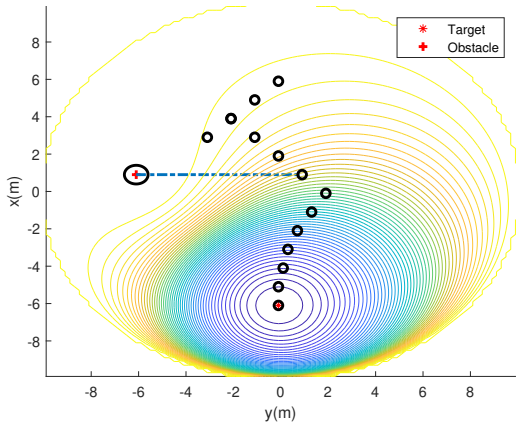


Figure 8.9: Scenario 3: waypoint generation using original navigation function method.

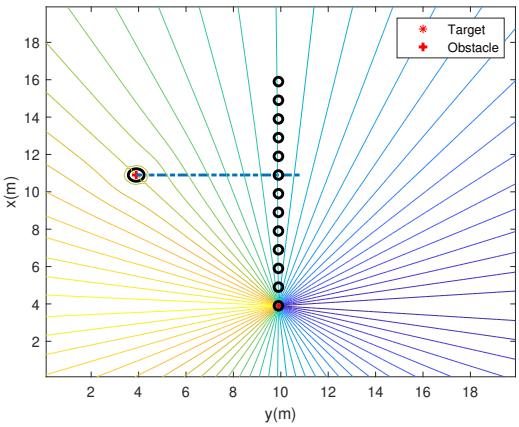


Figure 8.10: Scenario 3: waypoint generation using original stream function method.

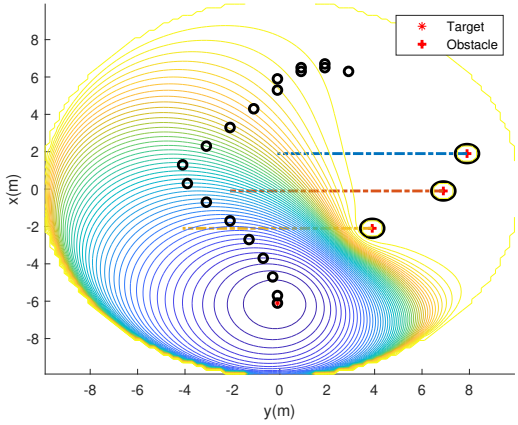


Figure 8.11: Scenario 6: waypoint generation using original navigation function method.

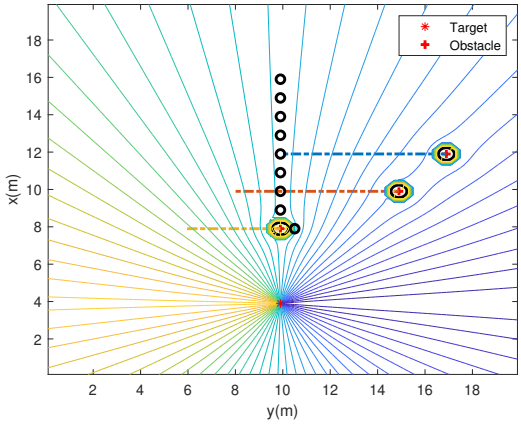


Figure 8.12: Scenario 6: waypoint generation using original stream function method.

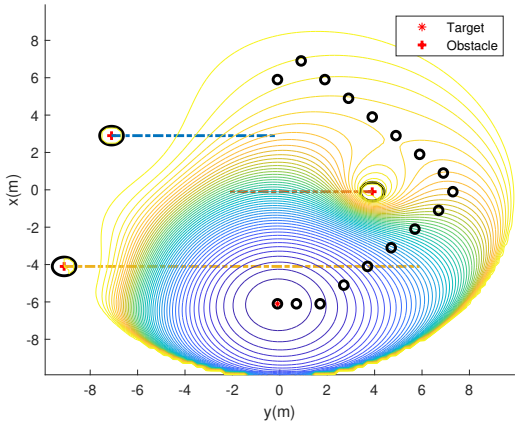


Figure 8.13: Scenario 7: waypoint generation using original navigation function method.

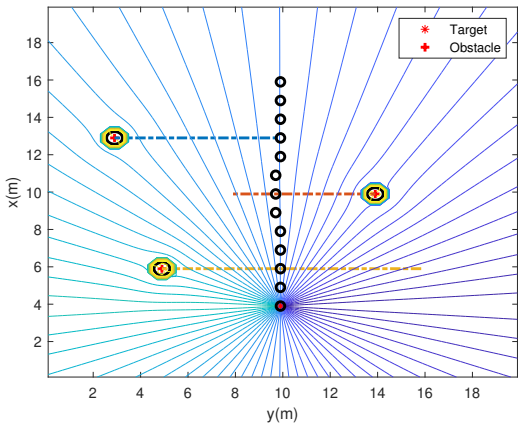
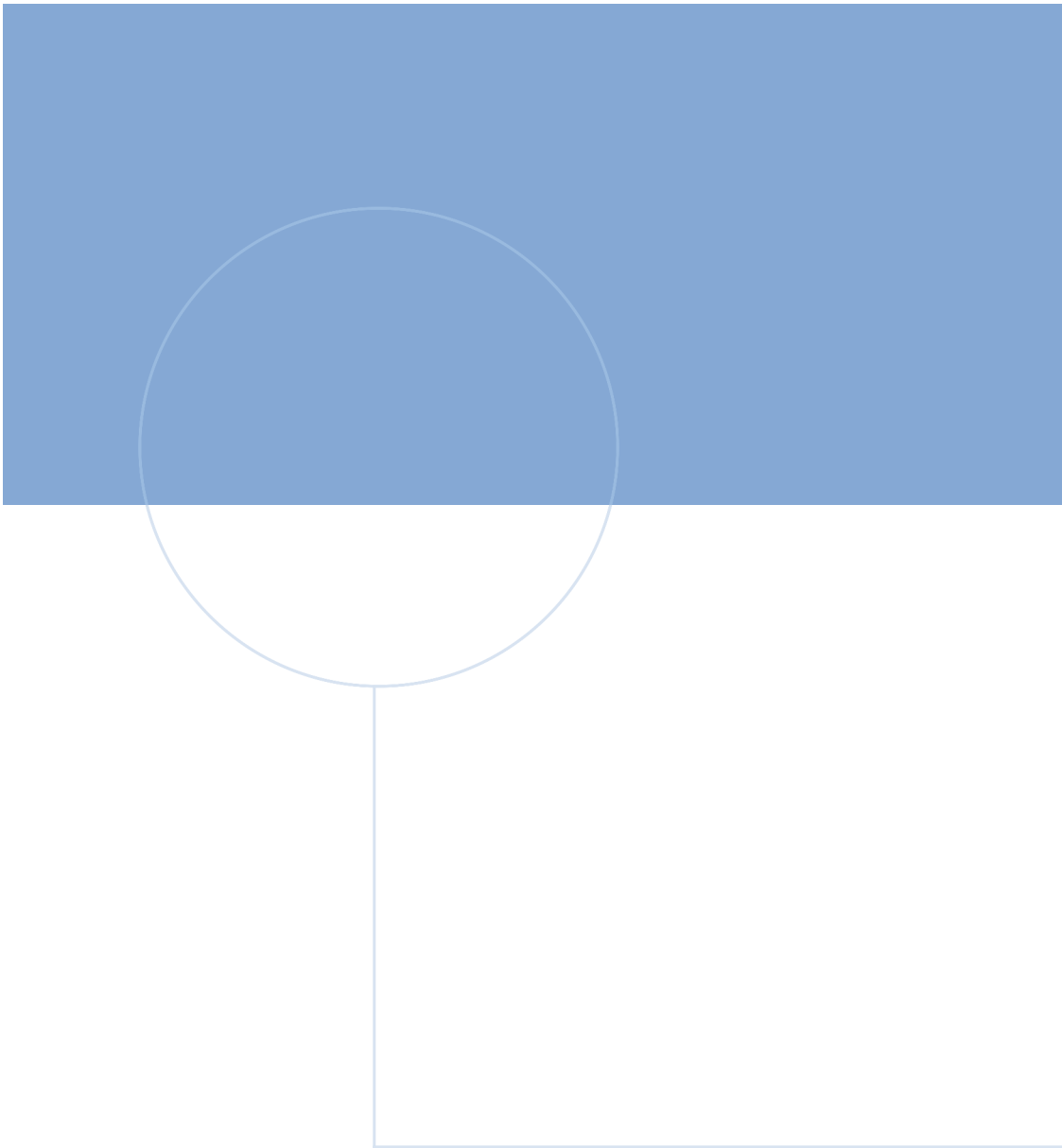


Figure 8.14: Scenario 7: waypoint generation using original stream function method.



NTNU

Kunnskap for en bedre verden