MINISTRY OF EDUCATION AND TRAINING

LE QUY DON UNIVERSITY

NGUYEN THI LAN ANH

DEVELOPING EFFICIENT LOCALIZATION AND MOTION PLANNING SYSTEMS FOR A WHEELED MOBILE ROBOT IN A DYNAMIC ENVIRONMENT

DOCTORAL DISSERTATION: CONTROL ENGINEERING AND AUTOMATION

HA NOI - 2021

MINISTRY OF EDUCATION AND TRAINING

LE QUY DON UNIVERSITY

NGUYEN THI LAN ANH

DEVELOPING EFFICIENT LOCALIZATION AND MOTION PLANNING SYSTEMS FOR A WHEELED MOBILE ROBOT IN A DYNAMIC ENVIRONMENT

DOCTORAL DISSERTATION

Major: CONTROL ENGINEERING AND AUTOMATION

Code: 092520216

SUPERVISOR: Assoc. Prof. Dr. Pham Trung Dung

HA NOI - 2021

ASSURANCE

I certify that this dissertation is a research work done by the author under the guidance of the research supervisors. The dissertation has used citation information from many different references, and the citation information is clearly stated. Experimental results presented in the dissertation are completely honest and not published by any other author or work.

Author

Nguyen Thi Lan Anh

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor, Assistant Professor Pham Trung Dung, who has been directly guiding me through the PhD progress. His passionate enthusiasm, unwavering dedication to research, and insightful advice have motivated me to carry out this research. I do appreciate all support and opportunities that he has provided to me.

Then, I wish to thank my my co-supervisor, Dr. Truong Xuan Tung, for his valuable advices on my research. He has given and discussed a lot of new issues with me. Working with Dr. Tung, I have learnt how to do research systematically. His support have motivated me to overcome all challenges in during my PhD journey.

Next, I also would like to thank the leaders and all lecturers of the Faculty of Control Engineering, Le Quy Don University for supporting me with favorable conditions and cheerfully helping me in the study and research process.

Finally, I must express my very profound gratitude to my parents, to my husband for unfailing support me and always encouraging, to my daughter, Tran Nguyen Khanh An, and my son, Tran Duc Anh for trying to grow up by themselves. This accomplishment would not have been possible without them.

Author

Nguyen Thi Lan Anh

CONTENTS

ABBREVIATIONS

No.	Abbreviation	Meaning
1	IMU	Inertial Measurement Unit
2	GPS	Global Position System
3	KF	Kalman Filter
4	EKF	Extended Kalman Filter
5	PF	Particle Filter
6	VO	Velocity Obstacle
7	RVO	Reciprocal Velocity Obstacle
8	HRVO	Hybrid Reciprocal Velocity Obstacle
9	DWA	Dynamic Window Approach
10	EDWA	Enhance Dynamic Window Approach
11	EB	Elastic Band
12	TEB	Time Elastic Band
13	PTEB	Proactive Time Elastic Band
14	ETEB	Extended Time Elastic Band
15	ROS	Robot Operating System
16	PCL	Point Cloud Library

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

1.1. Motivation

Mobility is an essential navigation issue for autonomous mobile robots. To allow the mobile robots to navigate safely in a real-world environment, the mobile robots must deal with four typical functional blocks of the navigation system [1], as shown in Fig. 1.1, including: (i) *perception* – the mobile robots must interpret its sensors to extract meaningful information; (ii) *localization* – the mobile robots must determine their position and orientation in the environment. In other words, it answers the question "Where am I?"; (iii) *motion planning* – includes *path planning techniques* and *obstacle avoidance methods*. The mobile robots utilize it to decide how to act to achieve its goals; and (iv) *motor control* – the mobile robots must modulate their motor outputs to achieve the desired trajectory, i.e. PID control.

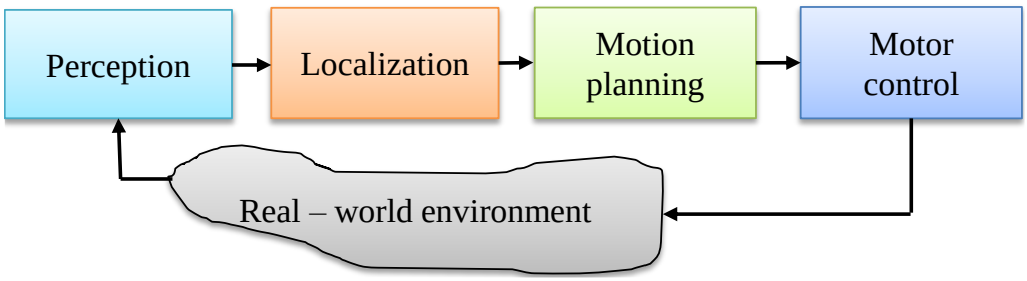


Figure 1.1: A general control scheme for autonomous mobile robots.

It has been known that, the success in robot navigation requires the

success of the four aforementioned fundamental processes, and to improve the performance of the robot's navigation system, the performance of all processes need to be improved.

In recent years, several domestic researches in the field of robotics have been published in recent years, such as the publications of Vietnam Academy of Science and Technology, Institute of Information Technology, Hanoi University of Science and Technology, Vietnam National University, Le Quy Don Technical University and Ho Chi Minh City University of Technology. The domestic works mainly focus on trajectory tracking systems [2], [3], [4] and [5]. Specifically, the authors propose control laws which enable the mobile robots to *follow predefined trajectories*. In this research, the authors also stated that, using these control laws the systems can overcome uncertainties caused by dynamic parameter variations and external disturbances. In other research direction, some studies [6] and [7] propose algorithms based on extended Kalman filter (EKF) to improve the localization system for mobile robots in unknown environments. And a few works [8], [9] develop adaptive control algorithms for *tracking moving targets* by using image features of the target which get from a camera system. Finally, a little researches introduce a trajectory planning method in a static environment with known start point and target point [10]. As a result, the mobile robot navigation system, especially localization and motion planning systems has not been focused and adequately researched.

Therefore, in this research we only focuses on two interesting systems including *localization* and *motion planning systems*, which are the scope

of the thesis.

Localization is the problem of estimating a robot's pose relative to its environment from sensor observations. It has been referred as the most fundamental problem to provide the mobile robot with autonomous competences. The challenges of localization are from the inaccuracy and inadequacies of sensors and effects of noise. Firstly, the errors of the measurement model, or sensor noise, due to the structural characteristics, resolution and error tolerance of different types of sensors or dynamic environments, such as light conditions, obstacles. Clearly, the solution here is to take multiple readings into account or multi-sensor fusion to increase the overall information from inputs. Secondly, the errors also can be caused by systematic errors (deterministic) such as the size of uneven wheels, the distance between two unbalance wheels, and they can be eliminated by proper calibration of the system. However, there are still a number of non-systematic (random) errors that remain, such as slipping on the surface, changes in the contact points of the wheel are uneven [1], leading to uncertainties in position estimation over time. In addition, when the mobile robot navigates in the harsh environmental conditions, the information can be interrupted in a short or long interval of time. Therefore, the mobile robot might have insufficient information for estimating the pose during its navigation.

The localization system has received the greatest research attention in the past decade, and as a result, significant advances have been made on this front [11]. In recent years, several conventional localization systems have been proposed to improve the performance of the robot pose

estimation such as [12], [13], [14], [15], [16], [17] [18], [19], [20]. Although they have achieved significant results in increasing the quality of this localization system, the researchers mainly focus on the case that the localization system receives sufficient information from the observation system. To the best of our knowledge, there are limited researches on localization systems when the sensing signal is interrupted or completely lost. Hence it is necessary to continue researching on these issues.

In other aspect, the motion planning system is crucial issue in navigation system of autonomous mobile robots, especially when they navigate in dynamic environments. Because, the dynamic environments are changeable, uncertain, and clustered environments with the presence of obstacles, humans, vehicles, and even other autonomous devices. Therefore, to ensure the safe navigation of the mobile robot in such environments, various navigation systems have been proposed.

The navigation frameworks can be divided into two categories based on the information used as the input of the motion planning system: (i) position-based approaches and (ii) velocity-based techniques.

In position-based group, only the position of the obstacles is used as the input of the navigation systems, such as [21], [22], [23], [24], [25] [26] and [27]. Some mobile robot navigation systems [22], [23] and [25] are developed using the social force model [28]. In other approach, the researchers utilize the dynamic window approach model proposed by Fox et al. [29] to develop the mobile robot navigation systems [21] and [24]. These navigation systems take into account the dynamics of the mobile robot and utilize the closest distance from the robot to the surround-

ing obstacles for obstacle avoidance. More recently, a few navigation systems [26] and [27] are developed using the timed-elastic-band (TEB) technique for avoiding obstacles. To maintain the separation from the obstacles, the TEB-based navigation systems take into consideration the distance from the proposed robot's trajectory to the surrounding obstacles. Despite the fact that the aforementioned navigation systems have been able to generate the safe navigation for the mobile robots in real-world environments, they do not proactively deal with *potential collision* with the surrounding obstacles. Because, these methods assume moving obstacles to be stationary. As a results, these navigation systems might be sufficient for the obstacle avoidance in static and semi-dynamic environments, but foresighted evasive maneuvers are not possible.

Regarding to the velocity-based techniques, the navigation systems such as [30], [31], [32] and [33], have taken into account the potential collision of the mobile robot with the surrounding obstacles based on the concept of the velocity obstacles presented by Fiorini et al. [34]. Specifically these navigation systems take into account the position and motion of all agents and selects the collision-free velocity command from the two-dimensional velocity space for each agent. Although these navigation systems have been installed in the mobile robot platforms and successfully verified in the real-world environments, they might not be able to handle all collision situations in the dynamic environments [31] and [33]. In addition, these systems only utilize the current position and velocity of the robot and the obstacles to create velocity commands, without considering the dynamics of the mobile robot. Therefore, it is difficult

to directly use this velocity command to control a mobile robot in the real-world environment.

In another approach, the navigation frameworks also can be divided into two categories based on the robot dynamics incorporated into the navigation systems: (i) none robot dynamics-based approaches and (ii) robot dynamics-based techniques.

Regarding to the none robot dynamics-based approaches, some obstacle avoidance techniques and motion control algorithms, such as the artificial potential field [35], vector field histogram [36], elastic band [37], velocity obstacles [34], [31], and social fore model [28], [23] techniques have been proposed for the autonomous mobile robots. These approaches have been achieved certainly in navigating mobile robots in dynamic environments. However, the systems do not directly take into account the dynamics of the mobile robots. Thus, it might be difficult to directly utilize the output control command to driving the mobile robots in the real-world environments, especially for non-holonomic mobile robots.

In order to address that issue, *several robot dynamics-based approaches* have been proposed in the recent years, such as the dynamic window approach [29], randomized kinodynamic planning [38], [39] and timed elastic band (TEB) [40] methods. Although, these approaches have been successfully applied in real-world environments, they might not suitable with the dynamic environments, because the mobile robots equipped with these techniques are unable to transit across obstacles in dynamic environments.

To deal with that problem, recently Rosmann et al. [41, 42] proposed

extensions of the TEB technique by using parallel trajectory planning in spatially distinctive topologies. Using this technique, mobile robots can switch to the current globally optimal trajectory among the candidate trajectories of distinctive topologies, which are maintained and optimized in parallel. However, these approaches only take into account the position of the surrounding obstacles and do not incorporate the potential collision of the mobile robots with the obstacles. Therefore, such developed navigation systems lack robustness in diverse situations in the dynamic environments.

In order to solve the aforementioned issues, it is necessary to proposed an efficient navigation system for autonomous mobile robots in dynamic environments. This navigation can drive the robots in both sufficient and insufficient sensing information, and takes into account both the robot dynamics and potential collision of the robots and the surrounding obstacles.

1.2. Objectives

The main objective of this thesis is to develop *an efficient navigation system* that enables a mobile robot to navigate *autonomously*, *safely* and *proactively* in a dynamic environment. In our study, the dynamic environment has characteristics, as follows: uncertain sensing, static and dynamic obstacles, changing environment and flat floor. In order to obtain the objective, we can break down into two sub-objectives: (i) improving the accuracy of the *localization system* and (ii) enhancing the *performance of the motion planning system*. In the former, localization

algorithms for the mobile robot in the dynamic environment with sufficient as well as insufficient information are proposed. In the later, we propose new local planning algorithms for the motion planning system of the mobile robot in the dynamic environment.

1.3. Methodology

In order to achieve the aforementioned objective, the dissertation makes use of an approach that combines theoretical analysis with experiments, and utilizes techniques in the fields of random signals, statistics, optimization to improve the performance of the mobile robot navigation system. In order to verify and evaluate the effectiveness of the proposed models we implement and install them in our mobile robot platform. The experiments are conducted in both simulation and real-world environments. The experimental results of the proposed system are compared with the conventional systems.

The dissertation utilizes software and tools for programming and testing the proposed algorithms including MATLAB and run on Intel core i7 desktop computer for simulation and quickly testing our ideas; the operating system is Ubuntu Linux; the programming language is C/C++; the Robot Operating System (ROS) [43] is the software core of the robots; of several methods for the localization system and new methods for motion planning system. The dissertation also makes use of the Stage robot simulator [44], OpenCV Library [45], ROS Toolbox in MATLAB[1] and the Point Cloud Library (PCL) [46] for conducting experiments in real-world

[1] https://www.mathworks.com/help/ros/index.html

environments.

1.4. Contributions

The main contributions of the dissertation are outlined as follows:

- Two sensor fusion-based localization algorithms are proposed. Firstly, the EKF -based localization algorithm is used to enable the robot to estimate its position with higher accuracy than conventional localization systems. Secondly, using the Particle filter (PF)-based localization algorithm, the robot may be able to determine its position when moving in the environments with sufficient information as well as the interrupted signal situation.

- Three new local planning algorithms in the motion planning system for autonomous mobile robots in dynamic environments are proposed, including *enhanced dynamic window approach (EDWA)*, *proactive time elastic band (PTEB)* and *extended time elastic band (ETEB)* algorithms. The mobile robots equipped with the proposed algorithms are capable of *proactively avoiding* dynamic obstacles and potential collisions, and *navigating safely* towards the given goal.

- The integrated navigation system based on the previous proposed algorithms including the EKF-based localization algorithm with the ETEB algorithm is utilized in real - world environments. The experimental results illustrate that the proposed system is efficient and feasibility in dynamic environments.

The results of the dissertation have been published on 5 papers includ-

ing 02 International conference papers and 02 Domestic journal papers, and 01 ISI journal paper.

1.5. Dissertation outline

The dissertation is composed of five chapters except for references, each of them dealing with difference aspect of the mobile robot navigation systems. The remainder of the dissertation is organized as follows.

Chapter 2 is introductions and reviews background information in the navigation frame of mobile robots. The chapter is subdivided into three parts. Part 1 introduce two type of mobile robot platforms which will be utilized experiments in real-world environments. Moreover, we describes a typical type of kinematic models that will be used in our simulations and experiments. Part 2 presents existing algorithms which are then used to improve the accuracy of the localization system for mobile robots. The final part presents conventional local planning algorithms in the motion planning system of mobile robots.

Chapter 3 presents two algorithms to enhance the performance of the localization system. The chapter consists of two parts. Part 1 addresses the issue of the multiple sensor fusion-based localization system using the EKF algorithm in the sufficient information case. Part 2 introduces the proposed localization algorithm based on the PF algorithm which can be use effectively in the interrupted information of the situation.

Chapter 4 introduces three new local planning algorithms of the motion planning system for the mobile robot. These proposed models enable the mobile robot to navigate *safely* and *proactively* in dynamic en-

vironments. We conduct experiments in both simulations and real-world environments to validate the effective of the proposed algorithms.

Conclusions and future works are drawn in the chapter 5. The main contributions of the graduation thesis have been reached.

Chapter 2
BACKGROUND

This chapter provides important insights for the next chapters in the thesis. Firstly, we introduce two types of mobile robot platforms and kinematic model of a differential-drive robot in Section 2.1 which are used in both simulation and real-world experiments to validate the effectiveness of the proposed algorithms. Secondly, the extended Kalman filter (EKF) algorithm and the Particle filter (PF) algorithm used in the proposed localization systems by L.A. Nguyen et al.[47], [48], are briefly described in Section 2.2. Then, three typical local planning (obstacle avoidance) algorithms adopted by L.A. Nguyen et al. [49], [50] and [51], including Dynamic Window Approach (DWA), Hybrid Reciprocal Velocity Obstacle (HRVO) and Time Elastic Band (TEB) algorithms, are presented in Section 2.3. Finally, Section 2.4 presents a conclusion summarizing what we wish to develop to achieve more efficient models .

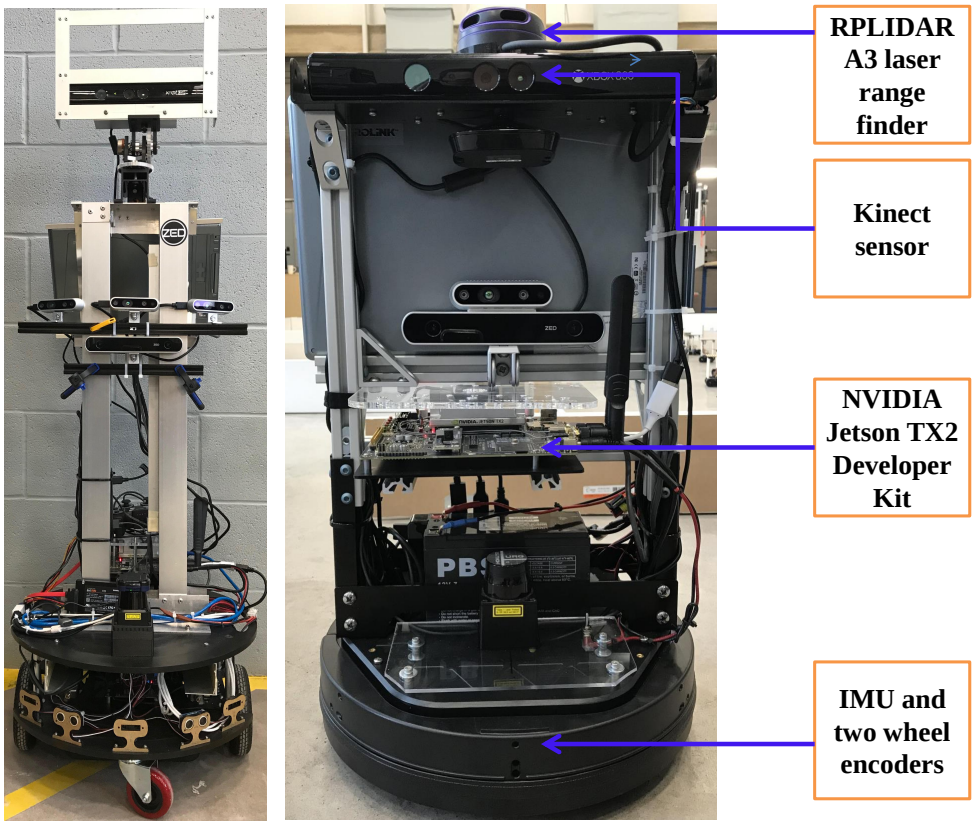
2.1. Mobile robot models

In order to verify the performance, efficiency and feasibility of proposed algorithms, that are going to be presented in the thesis, two robot platforms in The-More-Than-One Robot Laboratory, University of Prince Edward Island, Canada[1] which used in our experiments are

[1] http://morelab.org

firstly presented. Secondly, the typical kinematic model of differential-drive robots will be used in simulations in the next chapters is also introduced.

2.1.1. Mobile robot platforms



(a) (b)

Figure 2.1: Two mobile robot platforms under the study.

Two robot platforms that will be used in our experiments in chapter 4 including an Eddie mobile robot platform as shown in Fig. 2.1(a) and QBot-2e mobile robot platform in Fig. 2.1(b).

a. The Eddie mobile robot platform

The Eddie mobile robot platform is a two-wheel differential drive mobile platform and equipped with a Hokuyo UTM-30LX-EW laser

rangefinder and a NVIDIA Jetson AGX Xavier Developer Kit. The laser rangefinder positioned at the height of 0.4[m], provides the distance measurements up to 30.0[m] in the angular field of view 270^o, and is used for detecting obstacles (humans) in the vicinity of the mobile robot to provide information for the motion planning system.

The platform is equipped with two-wheel encoders with resolution 2.5^o and an internal measurement unit (IMU) for a localization system with resolution (X-axis, Y - axis) $0.02^o/s$.

The desktop computer and the Xavier board are connected via a wifi router using ROS Toolbox in MATLAB[2].

b. The QBot-2e mobile robot platform

The QBot-2e mobile robot platform[3] is also a two-wheel differential drive mobile robot platform with two additional caster wheels, as shown in Fig. 2.1(b). This platform is equipped with two-wheel encoders and an internal measurement unit (IMU) for a localization system. The encoders provide 2578 counts per revolution. In other words, the resolution of the encoders is 0.14^o. The IMU is a device that includes compass, gyroscopes and accelerometers to estimate the position, orientation, velocity and acceleration of a mobile robot. In this thesis, we only used the orientation component of IMU for localization systems (with resolution (X-axis, Y - axis) $0.02^o/s$).

The QBot-2e mobile robot platform is added a RPLIDAR A3 laser rangefinder and a NVIDIA Jetson TX2 Developer Kit, as shown in

[2]https://www.mathworks.com/help/ros/index.html
[3]https://www.quanser.com/products/qbot-2e

Fig. 2.1(b). The laser rangefinder positioned at the height of 0.45[m], provides distance measurements up to 25[m] in the angular field of view 360^o, and the resolution 0.33^o. The laser rangefinder is used for the localization system and detecting objects in the robot's vicinity, and providing information for the robot motion planning system. Whereas, the NVIDIA board is a embedded computer, which is used to install the entire navigation system and other software packages of the mobile robot.

2.1.2. Kinematic model of differential-drive robot

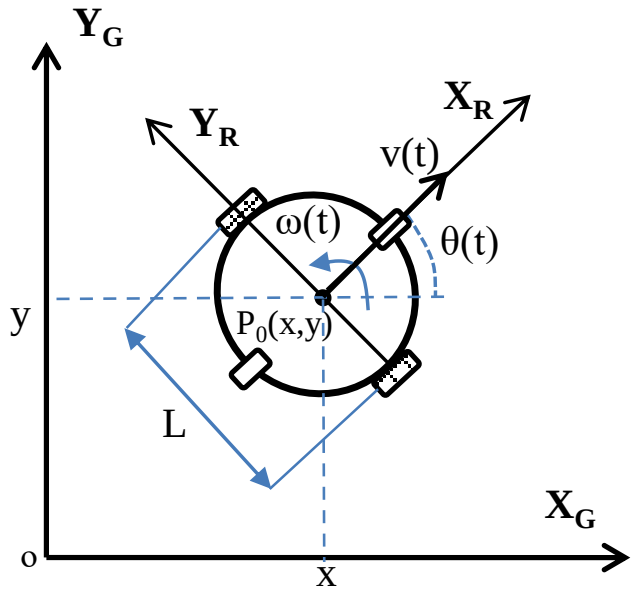


Figure 2.2: The global reference frame and the robot reference frame.

This subsection recapitulates the kinematic model of a differential-drive robot. In order to specify the position of the robot on the plane, a relation between the global frame of the plane X_GOY_G and the local reference frame of the robot X_ROY_R are established, shown in Fig. 2.2, also called as the current robot configuration. The position of the robot

(x, y) in the global reference frame is the position of the center point P_0 on the robot's chassis. The angular between the global and the local reference frame is given by θ. The pose of the robot can be describe as a vector with three elements $[x, y, \theta]^T$.

For the differential drive robot, shown in Fig. 2.2, the position can be estimated starting from a known position by summing the incremental travel distances. Thus, for a discrete system with a fixed sampling interval Δt, the incremental travel distances of the robot $\triangle x$ and $\triangle y$ on the global reference frame are

$$\triangle x = \triangle s \cos(\theta + \frac{\triangle \theta}{2}); \quad \triangle y = \triangle s \sin(\theta + \frac{\triangle \theta}{2}) \tag{2.1}$$

Assume that $\triangle s_r$ and $\triangle s_l$ are distances traveled by the right and left wheels of the robot in interval time Δt, respectively. As a result, the linear incremental displacement $\triangle s$ and the orientation $\triangle \theta$ of the robot are defined as follows:

$$\triangle s = \frac{\triangle s_l + \triangle s_r}{2}; \quad \triangle \theta = \frac{\triangle s_r - \triangle s_l}{L} \tag{2.2}$$

where, L denotes the distance between two robot's wheels.

Thus, we get the updated state of the mobile robot at the time k governed [1] as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \triangle s_{k-1} \cos(\theta_{k-1} + \frac{\triangle \theta_{k-1}}{2}) \\ \triangle s_{k-1} \sin(\theta_{k-1} + \frac{\triangle \theta_{k-1}}{2}) \\ \triangle \theta_{k-1} \end{bmatrix} \tag{2.3}$$

Equation (2.3) is also known as the *odometry motion model* of the mobile robot [11].

Let $\mathbf{u}_{k-1} = [v_{k-1}, \omega_{k-1}]^T$ denotes the control command at time k-1. Suppose that we keep the control command $\mathbf{u}_{k-1} = [v_{k-1}, \omega_{k-1}]^T$ constant for some time Δt, with the linear velocity command v_{k-1} and the angular velocity command ω_{k-1}. Moreover, $\triangle s_{k-1} = v_{k-1}\Delta t$, and $\triangle\theta_{k-1} = \omega_{k-1}\Delta t$. Thus, after the duration Δt the *velocity motion model* of the robot is as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} v_{k-1}\Delta t \cos(\theta_{k-1} + \frac{\omega_{k-1}\Delta t}{2}) \\ v_{k-1}\Delta t \sin(\theta_{k-1} + \frac{\omega_{k-1}\Delta t}{2}) \\ \omega_{k-1}\Delta t \end{bmatrix} \tag{2.4}$$

The motion model (2.3) and (2.4) will be used in the following Chapters.

2.2. Bayesian filters for localization systems

Before discussing conventional probabilistic localization methods in detail, the general probability robot localization problem is presented. Because the data coming from the robot sensors are affected by measurement errors, thus we can only compute the probability that the robot is in a given configuration. The key idea in probabilistic robotics is to represent uncertainty using probability theory: instead of giving a single best estimate of the current robot configuration, probabilistic robotics represents the robot configuration as a probability distribution over all possible robot poses. This probability is called *belief*.

Consider a mobile robot moving in a realistic environment, it can keep track of its position over time using odometry. Due to odometry uncertainty, after some movement, the robot will become very uncertain about its position. To keep the uncertainty about the position not grow-

ing, the robot must localize the relationship of itself to its environment. Thus, the robot might use its exteroceptive sensors (e.g., laser, vision sensors) to make observations of the environment. After that combining the information got from such exteroceptive observations with the information provided by the robot's odometry can enable the robot to localize more precisely. The process of updating all information can be divided into two steps [1], called prediction update and perception (or correction)update. More specifically, during the prediction update step, the robot estimating its configuration uses its proprioceptive sensor such as encoders to estimate its motion. However, errors introduced by encoders, lead to the robot's belief about the position is uncertain. Therefore, motion is somewhat non-deterministic. On the other hand, the perception update step, the robot uses the information from its exteroceptive sensors to correct the position estimated during the prediction phase. As result, the belief state of the robot is refined. In the following subsections, two different methods of probabilistic localization are described, including the Extended Kalman Filter localization and Particle filter localization.

2.2.1. Extended Kalman filter algorithm

Kalman filter [52] is widely used in many different applications, especially in the field of mobile robots. However, the linear Kalman filter addresses the general problem of trying to estimate the state of a discrete-time controlled process, that is governed by a linear stochastic difference equation. But what happens if the process to be estimated

and/or the measurement relationship to the process are non-linear? In this case, the extended Kalman filter is made use of.

Using the first order of the Taylor series expansion, the estimation can be linearized around the current estimation using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships [52]. Let us assume that a process has a state vector $\mathbf{x}_k \in \Re^n$ and the state of the process is governed by the non-linear stochastic difference equation:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \tag{2.5}$$

with a measurement $\mathbf{z} \in \Re^m$, that is

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \tag{2.6}$$

where, $\mathbf{x}_k, \mathbf{z}_k$ are the state and measurement vectors in the time step k, respectively; f is a non-linear function, that relates the state at the previous time step $k-1$ to the state at the current time step k. It is also included a driving function $\mathbf{u}_k$ and a zero-mean process noise $\mathbf{w}_k$; h is the non-linear function that relates the state $\mathbf{x}_k$ to the measurement $\mathbf{z}_k$. Each parameters in the vector is an observable value from each sensors at the time step k; $\mathbf{w}_k$, $\mathbf{v}_k$ are the random variables and represent the process and measurement noises, respectively. They are assumed to be independent to each other with normal probability distributions (2.7).

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k); \quad \mathbf{v}_k \sim N(0, \mathbf{R}_k); \quad E(\mathbf{w}_k, \mathbf{v}_k) = 0 \tag{2.7}$$

In practice of course one does not know the individual values of the noise at each time step. However, one can approximate the state and

measurement vector without them as follows:

$$\widehat{\mathbf{x}}_k^- = f(\widehat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \tag{2.8}$$

and

$$\widehat{\mathbf{z}}_k = h(\widehat{\mathbf{x}}_k) \tag{2.9}$$

The basic operation of the EKF filter is the same as the linear discrete Kalman filter. Hence, the equations for the EKF filter also fall into two phases: (i) time update equations, and (ii) measurement update equations. The time update equations are responsible for projecting forward the current state and error covariance estimates in time, to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimation.

EKF filter time update equations:

$$\widehat{\mathbf{x}}_k^- = f(\widehat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \tag{2.10}$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \tag{2.11}$$

EKF filter measurement update equations:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \tag{2.12}$$

$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\widehat{\mathbf{x}}_k^-)) \tag{2.13}$$

$$\mathbf{P}_k = (I - \mathbf{K}_k \mathbf{H}_k) + \mathbf{P}_k^- \tag{2.14}$$

where, $\widehat{\mathbf{x}}_k^- \in \Re^n$ is a priori state estimation at step k given knowledge of the process prior to step $k-1$; $\widehat{\mathbf{x}}_k \in \Re^n$ is a posteriori state estimation at

step k given measurement $\mathbf{z}_k$; $\mathbf{P}_k^-$ is an a priori estimation error covariance matrix; $\mathbf{P}_k$ is an a posteriori estimation error covariance matrix; $\mathbf{Q}_k$ is the process noise covariance (the covariance of the noise associated to the motion model); $\mathbf{R}_k$ is the measurement noise covariance at step k (Note subscript allowing it to change with each measurement); $\mathbf{K}_k$ is the Kalman gain; $\mathbf{F}_k$ and $\mathbf{H}_k$ are the Jacobian matrix of partial derivatives of the function f and h with respect to $\mathbf{x}$, respectively, and computed in (2.15).

$$\mathbf{F}_k = \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k)}{\partial \mathbf{x}_{k-1}}; \quad \mathbf{H}_k = \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} \tag{2.15}$$

$\mathbf{W}_k$ is the Jacobian matrix of partial derivatives of the function f with respect to $\mathbf{w}$; and $\mathbf{V}_k$ is the Jacobian matrix of partial derivatives of the function h with respect to $\mathbf{v}$.

$$\mathbf{W}_k = \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k)}{\partial \mathbf{w}}; \quad \mathbf{V}_k = \frac{\partial h(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{v}} \tag{2.16}$$

2.2.2. The particle filter algorithm

PDF (Particle Density Filter), the particle filter is developed based on the framework of Bayesian theory [53] [54]. Particle filter(also as known as Monte - Carlo) is a widely used localization method in the field of mobile robots [55]. Consider the evolution of the state sequence of a target given by

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \tag{2.17}$$

Where $f_k : \Re_x^n \times \Re_v^n \to \Re_x^n$ is a possibly nonlinear function of the state $\mathbf{x}_{k-1}$; $\mathbf{w}_{k-1}$ is a process noise sequence, n_x, n_v are dimensions of the state and process noise vectors, respectively. N is the set of natural numbers.

The objective of tracking is to recursively estimate $\mathbf{x}_k$ from measurements

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \tag{2.18}$$

Where $h_k : \Re_x^n \times \Re_n^n \rightarrow \Re_z^n$ is a possibly nonlinear function; $\mathbf{v}_k$ is a measurement noise sequence; n_z, n_n are dimensions of the measurement noise vectors, respectively. In particular, filtered estimates are sought of based on the set of all available measurements $\mathbf{z}_{1:k} = \{\mathbf{z}_i, i = 0, ..., k\}$ up to time k. To recursively calculate some degree of belief in the state at time k, following a Bayesian perspective, taking different values, given the data up to time k. Therefore, it is required to construct the pdf $\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1})$. In order to present the detail of the algorithm, let $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_s}$ denote a random measure that characterizes the posterior probability density distribution $p\{\mathbf{x}_{0:k}^i \| \mathbf{z}_{1:k}^i\}$. Where $\{\mathbf{x}_{0:k}^i, i = 1, ..., N\}$ is a set of particles with their associated weights $\{w_k^i, i = 1, ..., N\}$, and $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0, ..., k\}$ is the set of all states up to time k. N is the number of particles used in the approximation. Then, the posterior density at k can be approximated as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{k-1})p(\mathbf{x}_k|\mathbf{z}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \approx \sum_{k=1}^{N} w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i) \tag{2.19}$$

Where $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ the normalizing constant depends on the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ defined by the measurement model. The normalized weights are chosen using the principle of importance sampling [53]. If samples $\mathbf{x}_k^i$ were drawn from an importance density $q(\mathbf{x}_k|\mathbf{z}_k)$, then the

weights in (2.21) can be shown as

$$\tilde{w}_k^i \propto \frac{p(\mathbf{x}_k|\mathbf{z}_k)}{q(\mathbf{x}_k|\mathbf{z}_k)} \tag{2.20}$$

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{k=1}^{N}(\tilde{w}_{k-1}^i)} \tag{2.21}$$

The choice of the proposal distribution is one of the most critical design issues for successful particle filter. Two of those critical reasons are as follows: samples are drawn from the proposal distribution, and the proposal distribution is used to evaluate important weights. In practical terms, this means that after a certain number of recursive steps, all but one particle will have negligible. Degeneracy can be reduced by using good importance sampling function and resamepling. Resampling is a scheme to eliminate particles small weights and to concentrate and replicate on particles with large weights. Multiply/Suppress samples $\mathbf{x}_k^i$ with high/low importance weights w_k^i , respectively. To obtain N new random samples $\mathbf{x}_k^i$ approximately distributed according to $p(\mathbf{x}_i|\mathbf{z}_k)$ and set $w_k^i = 1/N$. Thus, the Particle Filter algorithm can be divided into three main steps:

1. Sample the particles using the proposal distribution.

2. Compute the importance weights: weight = target distribution/proposal distribution.

3. Resampling: draw samples with probability and repeat N times.

The Particle Filter algorithm also presents as Algorithm 1.

Algorithm 1: Basic Particle Filter algorithm

input : Particle filter input $(\mathbf{X}_{k-1}, \mathbf{u}_k, \mathbf{z}_k)$
output: $\overline{\mathbf{X}}_k$

begin
  Initialize parameter set $\overline{\mathbf{X}}_k = \mathbf{X}_k = \emptyset$
  for i=1 **to** N **do**
    Generate a particle
      $\mathbf{x}_k^{[i]} \sim p(\mathbf{x}_k^i | \mathbf{u}_k^i, \mathbf{x}_{k-1})$
    Calculate an importance weight
      $w_k^{[i]} = \frac{p\left(\mathbf{x}_k^{[j]}\right)}{q\left(\mathbf{x}_k^{[i]}\right)}$
    $\overline{\mathbf{X}}_k = \overline{\mathbf{X}}_k + \left(\mathbf{x}_k^{[j]}, w_k^{[j]}\right)$
  end for
  for i=1 **to** N **do**
    Draw $\mathbf{x}_k^{[i]}$ with importance weights $w_k^{[i]}$
    Add $\mathbf{x}_k^{[i]}$ to $\overline{\mathbf{X}}_k$
  end for

2.3. Typical obstacle avoidance algorithms

Obstacle avoidance algorithms can be called local planning algorithms. If full information is available about the obstacle set and a robot is present, global path planning methods may be used to find the optimal path. But only local information is available, sensor-based methods are used. A subset of sensor-based methods is reactive methods, which may be expressed as a mapping between the sensor state and control signal, with no memory present. Local obstacle avoidance focuses on changing the robot's trajectory as informed by its sensors during robot motion. The resulting robot motion is both a function of the robot's current or recent sensor readings and its goal position and relative location to the goal position. Local or reactive approaches, on the other hand, use only a small fraction of the environment model, to generate robot

control. However, this comes at the obvious disadvantage that they cannot produce optimal solutions.

The key advantage of local techniques over global ones lies in their low computational complexity, which is particularly important when the world model is updated frequently based on sensor information. The obstacle avoidance algorithms presented here depend to varying degrees on the existence of a global map and on the robot's precise knowledge of its location relative to the map. Despite their differences, all of the algorithms can be termed obstacle avoidance algorithms because the robot's local sensor readings play an important role in the robot's future trajectory. A great variety of approaches have demonstrated competent obstacle avoidance, such as Bug algorithm [56], bubble band technique [37], [57], curvature velocity method (CVM) [58], dynamic window approach [29], [59], [60]. In this section, some typical obstacle avoidance algorithms that have been used successfully in mobile robotics are presented.

2.3.1. The dynamic window approach algorithm

Dynamic window approach (DWA) model proposed by Fox et al. [29] is a classical method for obstacle avoidance, which is successfully applied for autonomous mobile robots in real-world environments [21], [24]. The DWA algorithm can be carried out in two stages: (i) search space, and (ii) optimization. *In the first stage*, the search space of the possible velocities (v, ω) is reduced using the circular trajectories and the dynamics of the mobile robot. *In the second stage*, the optimal velocities are chosen

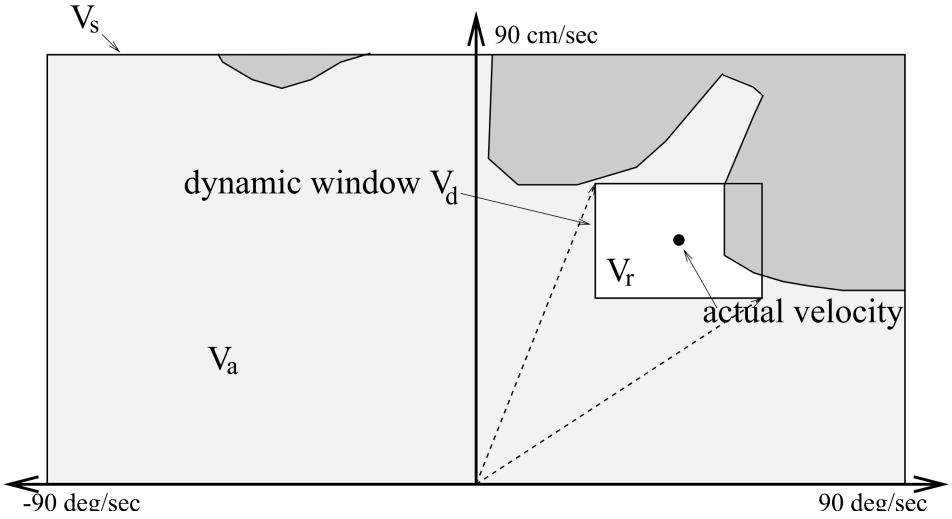


Figure 2.3: The velocity space of the dynamic window approach model. V_s, V_a, V_d are the possible velocities, admissible velocities, and dynamic window, respectively.

from the remaining velocities by maximizing the objective function.

2.3.1.1. Search space

The velocity space V_s is all possible sets of tuples (v, ω). Assume that given the current robot speed, the algorithm selects a *dynamic window* of all tuples (v, ω) [29], as shown in Fig. 2.3 that can be reached within the next sample period, taking into account the acceleration capabilities of the robot and the cycle time.

The search space V_r from the possible velocities can be performed in three steps. In the first step, the search space is reduced into two dimensions in the velocity search space in which each curvature is determined uniquely by the velocity pair (v, ω), which is translational and rotational velocities, respectively. In the second step, to avoid obstacles an admissible velocity set V_a is determined to ensure that only safe trajectories are considered. In other words, V_a is the set of velocities (v, ω) presented

by light gray area which allow the robot to stop without colliding with the obstacle. Finally, to consider the limited acceleration of the mobile robot by the motors, the search space is reduced to the dynamic window, which contains only the velocities that the robot can be reached within the next time interval. The dynamic window V_d is centered around the actual velocity and the extensions of its depend on the accelerations of the robot. In other words, V_d is an area which velocities can be reached in the next time interval from a current velocity. As a result, the search space V_r is an area, which is defined as the intersection of the V_s, V_a and V_d as follows:

$$V_r = V_s \cap V_a \cap V_d \tag{2.22}$$

with

$$V_s = \{0 \quad max_v \quad -[max_\omega] \quad [max_\omega]\} \tag{2.23}$$

$$V_a = \{(v,\omega)|v \leq \sqrt{2.dist(v,\omega).\dot{v}_b} \wedge \omega \leq \sqrt{2.dist(v,\omega).\dot{\omega}_b}\} \tag{2.24}$$

$$V_d = \{(v,\omega)|v \in [v_a - \dot{v}.t, v_a + \dot{v}.t] \wedge \omega \in [\omega_a - \dot{\omega}.t, \omega_a + \dot{\omega}.t]\} \tag{2.25}$$

Where, max_v and max_ω are a maximum velocity and acceleration of the robot. $dist(v,\omega)$ is the distance to the closest obstacle on robot's trajectory; and $\dot{v}_b$ and $\dot{w}_b$ are corresponding accelerations for breakage. (v_a, w_a) is the actual velocity; and $\dot{v}$ and $\dot{w}$ are accelerations in during the time interval t.

Figure 2.3 shows an example of the the resulting search space when the mobile robot is moving in a corridor. In this case V_r is represented by the white area.

2.3.1.2. Optimization using objective function

As presented in the previous subsection, the output of the search space is the set of velocities V_r. Obviously, there are a lot of velocity pairs among this area. Therefore, DWA technique predicts the results of each velocity pair candidates concerning target heading angle, the minimum distance to obstacles and linear velocity values and chooses the optimum speed pair by maximizing the objective function (2.26) using the weighted sum method.

$$G(\upsilon, \omega) = \alpha head(\upsilon, \omega) + \beta dist(\upsilon, \omega) + \gamma vel(\upsilon, \omega) \tag{2.26}$$

where, α, β, γ are the weights of the target heading, obstacle clearance and velocity, and predefined values.

- **Target heading:** The target heading function measures the progress toward the goal and is computed as follows:

$$head(\upsilon, \omega) = 180^o - |\theta^{goal} - \theta_r| \tag{2.27}$$

  where, θ^{goal} is the orientation of the vector pointing from the predicted position of the robot to the goal, and θ_r is the predicted orientation of the robot at the predicted position. To determine the predicted position assuming that the robot moves with the selected velocity during the next time interval using the robot kinematic model. Clearly, it has a maximal value when the robot moves directly to the target.

- **Obstacle clearance:** The function obstacle clearance $dist(\upsilon, \omega)$ represents the distance from the mobile robot to the closest obstacle

that intersects with its trajectory. If there is no obstacle on the trajectory, its value is set to a large constant.

- **Velocity:** The velocity function $vel(\upsilon, \omega)$ ensures that the robot operates at a maximum admissible linear speed. Therefore, it forces the robot to move faster on the trajectory. This function provides the admissible maximum speed effect to the objective function except it is near to the goal. Maximizing the objective function by choosing suitable weighted parameters results in the mobile robot navigates smoothly and safely, and guarantees the robot to reach the target as fast as possible.

The DWA technique has been proven to be well suited for robots operating at high speed and is successfully applied to several robots [1]. Thus this method takes robot dynamics into consideration. In other words, the pair of $vel(\upsilon, \omega)$ are created based on the current velocity and the dynamic constraints of the robot. More precisely, it considers the actual speed of the robot, its acceleration, and the robot physical limits. Therefore, it eliminates unreachable velocities coming form the limited accelerations of the robot. Moreover, all speed pairs in which are not able to stop before colliding with obstacles are also eliminated. Briefly, DWA has advantages such as: faster, safer and more goal - oriented signals from the velocity space of the robot. However, this method has some drawbacks: (i) the local minima problem occurs since it only takes into account the admissible velocity in the dynamic window while the connectivity of the free space is not considered; (ii) it does not proac-

tively deal with potential collisions when the robot moves close to the obstacles, because of considering only forward motions.

2.3.2. Hybrid reciprocal velocity obstacle model

The hybrid reciprocal velocity obstacles (HRVO) technique introduced by Snape et al. [31] is an extension of the reciprocal velocity obstacles method [30]. The HRVO method is a velocity obstacles-based approach [34] taking the motion of other agents into account for collision avoidance in multi-agent systems. The HRVO model has successfully been applied to multi-robot collision avoidance [61]. Thus, the HRVO technique can be also understood as a control policy where each agent selects a collision-free velocity from the two-dimensional velocity space in the $xy-plane$. A construction of the HRVO model of a robot and an obstacle is illustrated in Fig. 2.4.

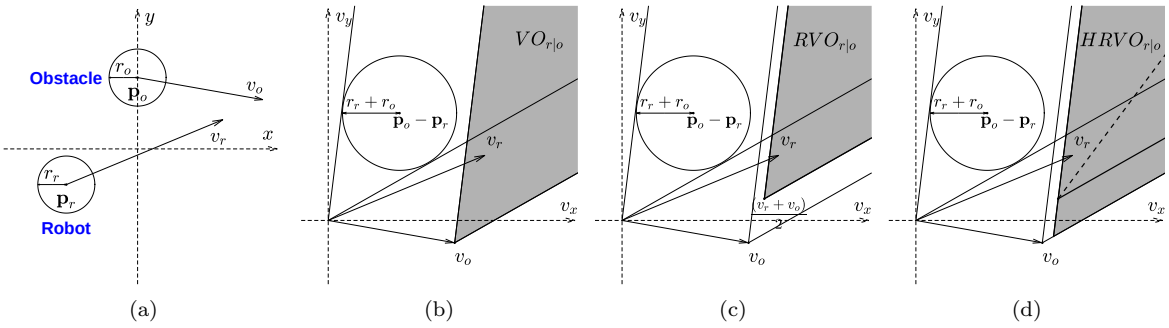


Figure 2.4: Procedure of the hybrid reciprocal velocity obstacles of a robot and an obstacle.

Figure. 2.4 contains (a) A configuration of a disc-shaped robot and a obstacle in the $xy-plane$ with radii r_r and r_o, positions $\mathbf{p}_r$ and $\mathbf{p}_o$, and velocities v_r and v_o, respectively; (b) The velocity obstacle (VO) [34] for the robot induced by the obstacle; (c) The reciprocal velocity

obstacle (RVO) [30] for the robot induced by the obstacle; (d) The hybrid reciprocal velocity obstacles (HRVO) [31] for the robot induced by the obstacle. Suppose that a set of dynamic and static obstacles O appear in the robot's vicinity. The combined $HRVO$ for the mobile robot given in the existence of several obstacles is the union of all the $HRVOs$ induced by all the obstacles:

$$HRVO_r = \bigcup_{o \in O} HRVO_{r|o} \tag{2.28}$$

According to [31], to avoid collisions with obstacles, the velocity $\mathbf{v}_r^{hrvo}$ of the mobile robot should be selected out side outside the $HRVO_r$ and close to the preferred velocity vector of the robot $\mathbf{v}_r^{pref}$. In other words, $\mathbf{v}_r^{hrvo}$ is calculated as follows:

$$\mathbf{v}_r^{hrvo} = arg \min_{\mathbf{v} \notin HRVO_r} \|\mathbf{v} - \mathbf{v}_r^{pref}\|_2 \tag{2.29}$$

where, $\mathbf{v}_r^{pref}$ is computed as follows:

$$\mathbf{v}_r^{pref} = v_r^{pref} \frac{\mathbf{p}_r - \mathbf{p}_g}{\|\mathbf{p}_r - \mathbf{p}_g\|_2} \tag{2.30}$$

where, $\mathbf{p}_r$ is the current position of the robot, $\mathbf{p}_g$ is the goal position, and v_r^{pref} is the preferred speed of the mobile robot.

2.3.3. Timed elastic band technique

The elastic band [37] is a well-known motion planning technique, which deforms a path to the goal by applying an *internal contraction force* resulting in the shortest path and *external repulsive forces* radiating from the obstacles to receive a collision-free path. Nevertheless, this approach does not take into account time information. In other words, the robot's

kinodynamic constraints are not considered explicitly, and hence a dedicated path following controller is required. In order to solve that issue, Rosmann et al. [40] presented an online trajectory planning algorithm for online collision avoidance, called timed elastic band (TEB) approach, which locally optimizes the robot's trajectory by minimizing the trajectory execution time, separation from obstacles and compliance with kinodynamic constraints such as satisfying limitations of velocities and accelerations.

In this subsection, the TEB algorithm described in [40] is briefly presented. Assuming that a discretized trajectory $\mathbf{B}$ is defined in terms of a finite-dimensional parameter vector including of an ordered sequence of mobile robot states $\mathbf{s}_k = [x_r^k, y_r^k, \theta_r^k]^T$, with k = 1, 2, ..., N and time stamps ΔT_k with k = 1, 2, ..., N-1. Thus the set of parameters $\mathbf{B}$ subject to optimization is defined as follows:

$$\mathbf{B} = [\mathbf{s}_1, \Delta T_1, \mathbf{s}_2, \Delta T_2, ..., \mathbf{s}_{N-1}, \Delta T_{N-1}, \mathbf{s}_N]^T \tag{2.31}$$

where, ΔT_k represents the time interval that the mobile robot have to requires to transit between two consecutive poses $\mathbf{s}_k$ and $\mathbf{s}_{k+1}$. An sample trajectory with three poses is depicted in Fig. 2.5 [41].

The main purpose of TEB method in the open-loop optimization is to find controls in order to move the robot from an initial pose s_s to a final pose s_f with a minimal time interval while guaranteeing kinodynamic constraints and separating from obstacles with a safe distance. The optimized TEB B^* is obtained by solving the following nonlinear program

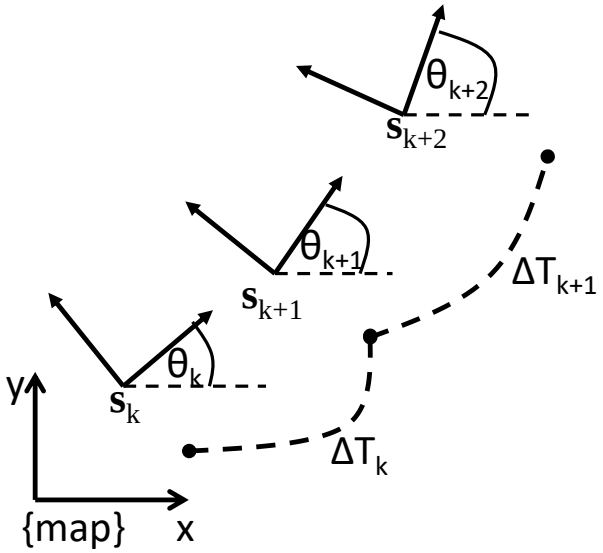


Figure 2.5: TEB trajectory representation with n=3 poses

(NLP):

$$\min_{B} \sum_{k=1}^{N-1} \Delta T_k^2 \tag{2.32}$$

subject to:

$0 \leq \Delta T_k \leq \Delta T_{max}$,

$\mathbf{h}_k(\mathbf{s}_{k+1}, \mathbf{s}_k) = 0$, (Nonholonomic kinematics)

$\mathbf{o}_k(\mathbf{s}_k) \geq 0$, (Clearance from surrounding obstacles)

$\nu_k(\mathbf{s}_{k+1}, \mathbf{s}_k, \Delta T_k) \geq 0$, (Limitation of robot's velocities)

$\alpha_k(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{s}_{k-1}, \Delta T_k, \Delta T_{k-1}) \geq 0$ (Limitation of robot's accelerations)

The total transition time is approximated by T $\approx \sum_{k=1}^{N-1} \Delta T_k$, ΔT_{max} is an upper limit of ΔT_k in order for the robot moving smoothly in the real time. Minimizing the sum of squared $\sum_{k=1}^{N-1} \Delta T_k^2$ enforces to achieve uniform time intervals $\Delta T_k = T/N$. The aforementioned equality and inequality equations represent the constraint of the environment with the robot, such as nonholonomic kinematics, clearance from obstacles and bounds on velocities and accelerations. All of the constraints are incor-

porated into the objective function (2.33) as additional penalty terms.

$$V(\mathbf{B}) = \sum_{k=1}^{N-1}[\ \Delta T_k^2 + \delta_h \|\mathbf{h}_k\|_2^2 + \delta_v \|min\{\mathbf{0}, \nu_k\}\|_2^2 + \delta_o \|min\{\mathbf{0}, \mathbf{o}_k\}\|_2^2 + \delta_\alpha \|min\{\mathbf{0}, \alpha_k\}\|_2^2\ ] = \mathbf{w}^T f(\mathbf{B}) \quad (2.33)$$

where, the notation $\mathbf{B} \setminus \{s_1, s_N\}$ implies that neither the start pose $\mathbf{s}_1$ nor the goal pose $\mathbf{s}_N$ are subject to optimization. It is noted that, during optimization the trajectory is clipped at the current robot pose $\mathbf{s}_k$ and the desired goal pose $\mathbf{s}_N$. Quadratic penalties function are applied according to [62] with user defined weights δ. For the remainder of (2.33), the cost function $V(\mathbf{B})$ is expressed in terms of the dot product, in which $\mathbf{w}$ captures individual weights and $f(\mathbf{B})$ contains individual cost terms. With objective function $V(\mathbf{B})$ the overall optimization problem is defined by:

$$\mathbf{B}^* = arg \min_{\mathbf{B}\setminus\{\mathbf{s}_1, \mathbf{s}_N\}} V(\mathbf{B}) \quad (2.34)$$

The TEB approach utilized the Levenberg-Marquardt (LM) algorithm [62] to solve (2.34) and obtained the optimal trajectory $\mathbf{B}^*$ of the mobile robot. Finally, the desired control command of robot $u_r^k = [\upsilon_r^k\ ,\ \omega_r^k]$ can be directly calculated from the optimal trajectory $\mathbf{B}^*$, and (2.35) and (2.36). The mean translational and rotational velocities are approximated using finite differences according to the Euclidean respectively angular distance between two consecutive configurations s_k and s_{k+1} and the time interval ΔT_i for transition between both poses.

$$\upsilon_r^k = \Delta T_k^{-1} \| [x_{k+1} - x_k, y_{k+1} - y_k]^T \| \gamma(s_k, s_{k+1}) \quad (2.35)$$

$$\omega_r^k = \Delta T_k^{-1} (\beta_{k+1} - \beta_k) \quad (2.36)$$

where, $\gamma(s_k, s_{k+1})$ denotes a function that extracts the sign of the translation velocity, whether the robot moves forwards or backwards.

Considering a differential drive mobile robot, the relationship between the two-wheel velocities and the translational and rotational velocities υ_r and ω^r of the robot center point are computed according to:

$$\upsilon_r^r = \upsilon_r + \frac{\omega_r L}{2} \tag{2.37}$$

$$\upsilon_r^l = \upsilon_r - \frac{\omega_r L}{2} \tag{2.38}$$

in which the parameter L denotes the distance between two wheel.

The classical TEB technique has been applied in real-world environment and has achieved considerable success with some advantages such as online trajectory optimization, reaching goal in minimal time, taking into account kinodynamic constraints, applying for non - holonomic kinematics and having alternative trajectories in distinctive topologies. However, it still has a drawback of only optimizing a single trajectory leading to stuck to a locally optimal trajectory somewhere. In dynamic environments, the presence of obstacles introduces multiple local minima, hence finding local minima coincides with the extraction of distinctive topologies. To tackle this problem, recently the TEB approach was extended to parallel trajectory planning in spatially distinctive topologies [41] and [42], which enable the robot to switch to the current globally optimal trajectory among the candidate trajectories of distinctive topologies.

The extension TEB technique [42] is presented in Fig. 2.6(b) and Algorithm 2, which consists of three major steps: (i) exploration (Lines 2-6) of the Algorithm 2), (ii) optimization (Lines 7-11 of the Algorithm

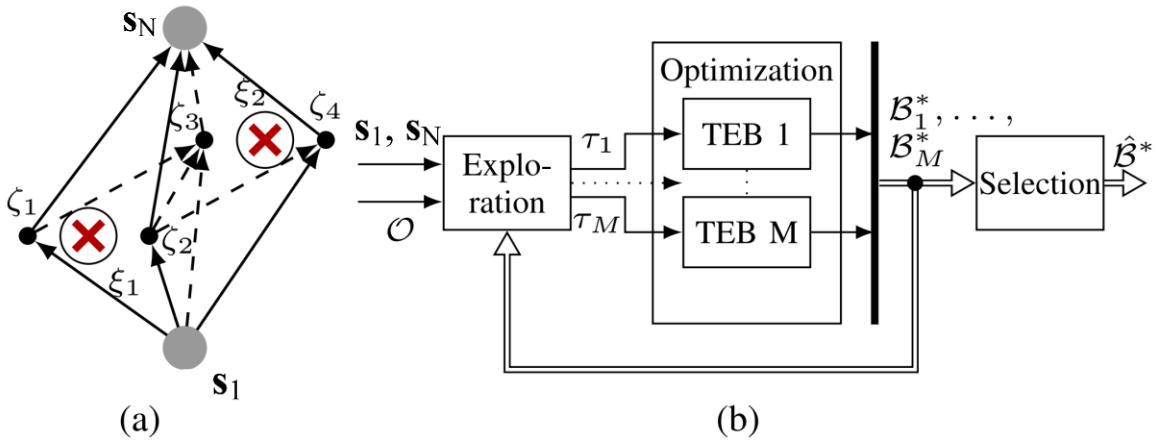


Figure 2.6: The example of exploration graph (a). The block diagram of parallel trajectory planning of time elastic bands (b).

Algorithm 2: Timed elastic band algorithm

input : robot state $\mathbf{s}_r$, start pose $\mathbf{p}_s$, goal pose $\mathbf{p}_g$, set of obstacles $\mathbf{O}$
output: Control command $\mathbf{u}_r$

1 **begin**
2 $\mathbf{G} \leftarrow$ createGraph($\mathbf{s}_r$, $\mathbf{p}_s$, $\mathbf{p}_g$, $\mathbf{O}$);
3 $\mathbf{D} \leftarrow$ depthFirstSearch($\mathbf{G}$);
4 $\mathbf{H} \leftarrow$ computeH-Signature($\mathbf{D}$, $\mathbf{G}$);
5 $\mathbf{R} \leftarrow$ removeRedundantPath($\mathbf{D}$, $\mathbf{H}$, $\mathbf{G}$);
6 $\mathbf{T} \leftarrow$ initializeTrajectories($\mathbf{R}$, $\mathbf{G}$);
7 **for** each trajectory $\mathbf{B}_p \in \mathbf{T}$ **do**
8 $\mathbf{V} \leftarrow$ objectiveFunction(); $\triangleright$ using (2.33)
9 $\mathbf{B}_p^* \leftarrow$ Optimizer($\mathbf{B}_p, \mathbf{O}, \mathbf{V}$); $\triangleright$ Solve (2.34)
10 $\mathbf{B}^* \leftarrow$ storeLocalOptimalTrajectory($\mathbf{B}_p^*$);
11 **end for**
12 $\mathbf{V}_c \leftarrow$ newObjectiveFunction(); $\triangleright$ using (2.40)
13 $\hat{\mathbf{B}}^* \leftarrow$ Optimizer($\mathbf{B}^*, \mathbf{O}, \mathbf{V}_c$); $\triangleright$ Solve (2.39)
14 $\mathbf{u}_r \leftarrow$ According to (2.35), (2.36) and $\hat{\mathbf{B}}^*$
15 Return $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$;

2) and (iii) selection (Lines 12-13 of the Algorithm 2). The input of the Algorithm 2 is the robot state $\mathbf{s}_r$, start pose $\mathbf{p}_s$, goal pose $\mathbf{p}_g$ and set of obstacles $\mathbf{O}$, and the output is the control command $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$ of the mobile robot.

In the exploration step, a graph $\mathbf{G}$ is generated to connect from $\mathbf{p}_s$ to

$\mathbf{p}_g$ by forward directed edges. The graph is then filter using depths-first search algorithm to keep only the acyclic graph. Finally, the H-Signature [63] technique is utilized to filters redundant paths that have the same H-Signature; as a result, a set of M primitive candidate paths that belong to alternative distinctive topologies are obtained.

In the second step, locally optimal trajectories for all M alternative topologies are planned in parallel by using the TEB optimization with respect to the objective function 2.33, which generates M locally optimal trajectories respectively $\mathbf{B}_p^*$, with p = 1, 2, ..., M.

In the final step, the best TEB $\hat{\mathbf{B}}^*$ or the least-cost trajectory is selected from the set of alternatives $\mathbf{B}_p^*$ obtained by solving the following equation, which reveals the global minimizer.

$$\hat{\mathbf{B}}^* = arg \min_{\mathbf{B}_p^* \in \{\mathbf{B}_1^*, \mathbf{B}_1^*, ..., \mathbf{B}_M^*\}} V_c(\mathbf{B}_p^*) \tag{2.39}$$

where,the objective function $V_c(\mathbf{B}_p^*)$ is presented as follows:

$$V_c(\mathbf{B}_p^*) = \mathbf{w}_c^T f_c(\mathbf{B}_p^*) \tag{2.40}$$

The extension TEB technique has been successfully applied in dynamic environments [41] and [42]. Nevertheless, the TEB planner only incorporates the position of the obstacles and does not take into account potential collision of the robot with the obstacles, which results in an unintelligent behavior in the dynamic environments.

2.4. Conclusions of the chapter

This chapter have presented the conventional models and techniques, which are then utilized in the next chapters in the thesis. Firstly the

kinematic model of the differential drive robot, and two mobile robot platforms are introduced. Secondly, the extended Kalman filter and the Particle filter algorithms are presented. Finally, the Dynamic Window Approach (DWA), Hybrid Reciprocal Velocity Obstacle (HRVO) and Time Elastic Band (TEB) algorithms are presented, these algorithm are used to develop the proactive motion planning system in Chapter 4. In the next Chapter, we are going to present two proposed localization algorithms based on the multiple sensor fusion methods, including EKF algorithm and the PF algorithm.

Chapter 3
SENSOR DATA FUSION-BASED LOCALIZATION ALGORITHMS

Localization is the problem of estimating robot's pose relative to its working space from sensor observations. It has been referred as the most fundamental problem to provide mobile robots with autonomous capabilities. Because, to achieve autonomous navigation, the mobile robot must maintain an accurate knowledge of its position and orientation.

The localization system suffers from two main problems, including inaccuracy or/and incompleteness of sensors (or sensor noise), and with Gaussian/Non-Gaussian distribution of noises, when a robot moves in a real-world environment. In order to deal with these problems effectively, two multiple sensor fusion-based localization algorithms are proposed to improve the performance of the localization system with *two different cases*, respectively. *The first case* is sufficient information and Gaussian distribution noises using extended Kalman filter(EKF)-based localization. *The second case* is insufficient information and Non-Gaussian/Gaussian distribution noises using Particle filter (PF)-based localization. The main idea of two algorithms is to fuse the data from different sensors composing of wheel encoders, IMU and GPS sensors to get more accurate estimations of robot's pose.

This chapter is organized as follows. To deal with the first case, the

extended Kalman filter-based localization algorithm is presented in Section 3.1. In Section 3.2, the proposed localization algorithm based on Particle filter, which is utilized to solve the second case, is introduced. Finally, conclusion is draw in Section 3.3.

3.1. Extended Kalman filter-based localization algorithm

While a robot moving in a real-world environment, it is equipped with a sensor system to know its position and orientation. Therefore, to localize successfully the robot has to determine both motion model and measurement model exactly [1]. The errors can be caused by systematic errors, such as the size of uneven wheels, the distance between two unbalance wheels, and non-systematic errors, such as slipping on the surface, changes in the contact points of the wheel are uneven. In addition, the error caused by the measurement model is due to the structural characteristics, resolution and error tolerance of different types of sensors or dynamic environment, such as light or obstacles that robots move there. In particular, some of the errors might be deterministic. Thus, they can be eliminated by proper calibration of the system, such as errors occurring in mechanical design. However, there are still a number of non-deterministic errors that remains, leading to uncertainties in robot pose estimation over time. Each error bases on a distribution rule, either Gaussian or non-Gaussian. Therefore, in order to improve the accuracy of the localization system in the dynamic environment, it is necessary to first determine the cause of the disturbance, the type of noise distribution and how to eliminate such noise [11].

Because the noise randomly occurs during the robot's navigation, thus one of the common solutions used to compensate the noise caused by motion model and the measurement model is the sensor fusion with different precision. In addition, each sensor measures only once or two parameters of the environment with limited accuracy. Moreover, using more sensors with higher accuracy will increase a quality of measurement. This is why the sensor fusion algorithms are thriving.

Several mobile robot localization systems have been proposed in recent years to improve the performance of the robot pose estimation [13], [14], [15], [16], [17] and [18]. An enhanced low-cost 3-D localization system is presented in [13]. In this paper, the authors made use of the Kalman filter algorithm to integrate the data from wheel encoders, MEMS-based inertial sensors, and GPS. In [14], the researchers presented a localization system of a mobile robot, which is equipped with 3–axis inertial measurement unit, an active beacon system, and wheel encoders. To do that, they utilized a low-pass filter and a Kalman filter algorithm to reduce noise of input sensors data and obtain more precise robot position and robot movement in real-time. In [15], the authors proposed a localization system of a mobile robot along an uneven path, where it cannot solely rely on encoders, GPS or accelerometer individually. In this system, the Kalman filter-based sensor fusion algorithm was implemented in order to get the best position estimation. A self-localization technique for autonomous mobile robot based on particle filtering in active beacon system is presented in [16]. Using ultrasonic sensor as a particle filter is applied to eliminate process and measurement noise. In [17], the

authors presented a sensor fusion framework, that improves the localization system of mobile robots with limited computational resources. To do that, they employs an event based Kalman filter to combine the measurements of a global sensor and an inertial measurement unit on an event based schedule, using fewer resources but with similar performance when compared to the conventional methods. In [18], an adaptive neuron fuzzy inference system was proposed for fusing the GPS and IMU measurements to enhance performance estimation in low cost navigation system when the robot moves in a dynamic environment or slippery ground surfaces and uneven road conditions. Moving on the motivation of using EKF to improve the accuracy of the localization system of the autonomous mobile robot, which is equipped with wheel encoders, GPS and IMU sensors, is proposed.

The remainder of this section is organized as follows. Section 3.1.1 presents a construction of EKF - based localization algorithm with the measurement vector design in three different approaches. Section 3.1.2 shows the simulation results and discussions.

3.1.1. Construction of EKF-based localization algorithm

A wide variety of sensors is used in mobile robots. Each sensor has different characteristics and functions. We can devide sensors into two groups: (i) proprioceptive group and (ii) exteroceptive group. Proprioceptive sensors measure values internal to the robot, for example, wheel/motor sensors and IMUs (gyroscope and accelerometers). Meanwhile, exteroceptive sensors are used to extract typical environment fea-

tures as well as acquire information on objects in the robot's vicinity, or even to measure directly a robot's global position. Thus, the robot may know the robot's position and the state of its surroundings. There are several common exteroceptive sensors, such as active ultrasonic beacons, GPSs or vision sensors, and so on.

In the proposed localization system, we utilize wheel encoders, IMU (9-axis family) and GPS to determine the position and orientation of the mobile robot in the dynamic environment.

The robot uses wheel encoders to estimate its pose or odometry motion model. As it starts to move from a precisely known location, it can keep track of its motion using odometry motion model. Due to odometry uncertainty, after long time the robot will become very uncertain about its position. In other words, the uncertainty of the robot configuration increases due to the integration of the odometric error over time.

Meanwhile, IMU is a device that uses accelerometers, gyroscopes and compasses to estimate a relative position, velocity, and acceleration of a moving robot. The accerlerometers are used to estimate the instantaneous acceleration. The acceleration is then integrated to obtain the velocity and then integrated again to obtain the position. While the gyroscope and compass data are fused and integrated to estimate the robot orientation. As presented in Section 2.1.1, in this study we only use the orientation component of the IMU sensor data to correct the orientation estimated from the wheel encoders. However, after long period of operation, all IMUs drift.

To eliminate this drift of IMU and accumulated error of encoders,

GPS is used to correct the estimated pose every time the GPS signal is received. GPS provides the absolute position and heading of the mobile robot. It's depended on various environments as well as different applications that we chose an appropriate GPS system. If the mobile robot moves in indoor environments, the high resolution GPS should be applied such as differential GPS (DGPS) which makes use of a second static receiver, can achieve 1 cm or sub -1 cm resolution [1]. While the lower resolution GPS will be chosen, when the robot drives in outdoor environments.

Moreover, each sensor has its own advantages and disadvantages. Thus, the extended Kalman filter algorithm to fuse the data from aforementioned sensors was utilized to improve the accuracy of the localization system. To accomplish that, a block diagram of the mobile robot localization system was proposed as shown in Fig. 3.1.

The EKF uses two steps including prediction and correction process to estimate the states of the robot. In the prediction step, the robot's state predictions are made based on a nonlinear kinematic motion model (in this case motion model is odometry motion model). In the correction step, the predicted states are corrected based on measurement observations from multiple sensor (GPS/IMU). A detailed description of the sources for each of the previously mentioned measurements along with associated measurement models [64] is presented in Fig. 3.2.

The EKF filter is a well-known method that linearizes the state of a system about an estimation of the current mean and covariance. This can be done only if the linearization errors are small in the update time

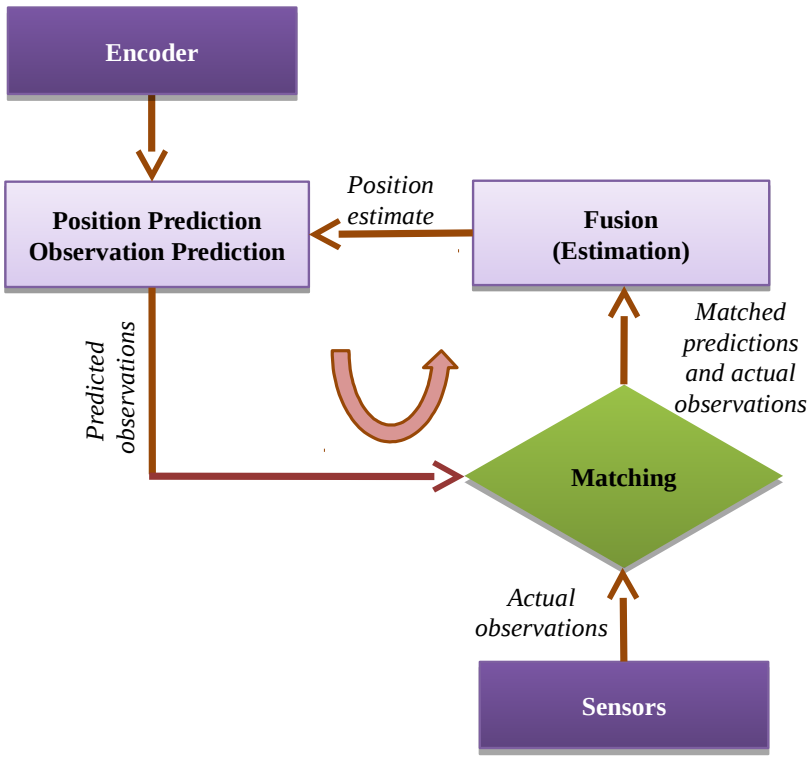


Figure 3.1: The block diagram of the proposed autonomous mobile robot localization systems based on the multiple sensor fusion methods.

interval [65]. In this work, assuming that the sensor frequencies are high enough, leading to the time interval is short enough. Therefore, the linear estimation is a valid hypothesis. Furthermore, without loss of generality, suppose that the process noise is non–additive and measurement noise is the additive. Thus the measurement model modified from (2.6) is presented in (3.1).

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{3.1}$$

where, $\mathbf{x}_k = [x_k, y_k, \theta_k]^T$ is the state vector defined in (2.4); $\mathbf{z}_k$ is the measurement vector; $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$ is the measurement noise with the covariance matrix $\mathbf{R}_k$; $\mathbf{u}_k = [v_k, \omega_k]^T$ and $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ in (2.5) are the input control vector of the robot and the process noise with the covariance matrix $\mathbf{Q}_k$, respectively; v_k is the linear velocity and ω_k is the

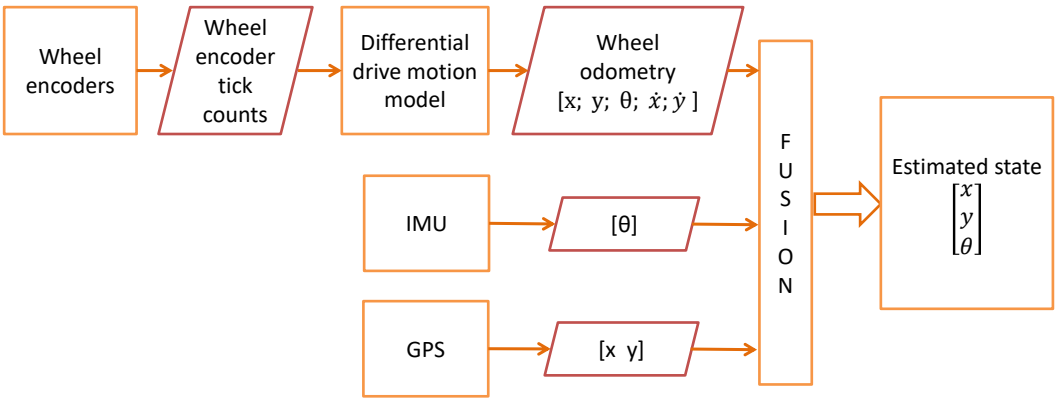


Figure 3.2: The data flow from sensors into the EKF for robot localization.

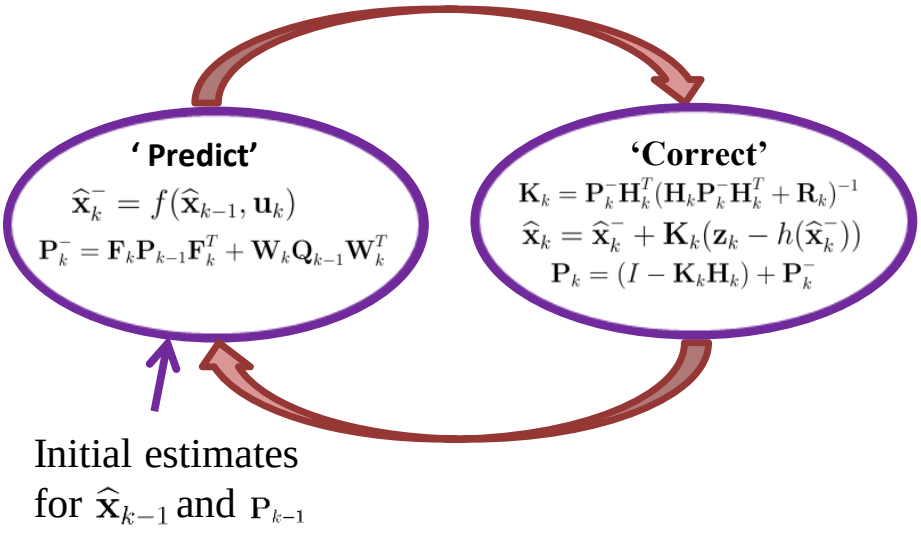


Figure 3.3: The extended Kalman filter-based mobile robot localization system.

angular velocity of the robot; and $\mathbf{Q}_k$ is defined in (3.2), as presented in [1].

$$\mathbf{Q}_k = \begin{bmatrix} (\alpha_1|v_k| + \alpha_2|\omega_k|)^2 & 0 \\ 0 & (\alpha_3|v_k| + \alpha_4|\omega_k|)^2 \end{bmatrix} \tag{3.2}$$

In addition, because the measurement noise is the additive noise, therefore (2.12) is rewritten as follows:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{3.3}$$

As a result, the entire extended Kalman filter-based localization sys-

tems for autonomous mobile robots is shown in Fig. 3.3.

In reality, the robot motion is subject to noise. The actual velocities differ from the control command (or measured ones, if the robot possesses a sensor for measuring velocity). This difference will be modeled by a zero-centered random variable with finite variance. More precisely, let us assume the actual velocities are given by:

$$\begin{bmatrix} v'_k \\ \omega'_k \end{bmatrix} = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} \varepsilon_{\alpha_1|v_k|+\alpha_2|\omega|} \\ \varepsilon_{\alpha_3|v_k|+\alpha_4|\omega|} \end{bmatrix} \tag{3.4}$$

Here ε_b is a zero – mean error variable with variance b. Thus, the true velocity equals the commanded velocity plus some small, additive error (noise). In our model, the variance of the error is proportional to the commanded velocity. The parameters α_1 to α_4 are robot specific error parameters. They model the accuracy of the robot. The less accurate a robot, the larger these parameters. The common choice for the error ε_b is a normal distribution. The normal distribution with zero mean and variance b is given by the density function:

$$\varepsilon_b(a) = \frac{1}{\sqrt{2\pi.b}} e^{-\frac{1}{2}\frac{a^2}{b}} \tag{3.5}$$

Actually, any meaningful posterior distribution is of course not degenerate, and poses can be within a three dimensional space of variations in x, y, and θ. To generalize our motion model accordingly, assuming that the robot performs a rotation γ' when it arrives at its final pose. Thus instead of computing $\theta_k = \theta_{k-1} + \omega_{k-1}\Delta t$ (as seen (2.4)), the final orientation is modeled by:

$$\theta_k = \theta_{k-1} + \omega'_{k-1}\Delta t + \gamma'\Delta t \tag{3.6}$$

with $\gamma' = \varepsilon_{\alpha_5|v|+\alpha_6|\omega|}$

where, α_5 and α_6 are additional robot – specific parameters that determine the variance of the additional rotational noise. The actual pose $\mathbf{x}_k = [x_k,\ y_k,\ \theta_k]^T$ after executing the motion command $\mathbf{u}_{k-1} = [v_{k-1},\ \omega_{k-1}]^T$ at $\mathbf{x}_{k-1} = [x_{k-1},\ y_{k-1},\ \theta_{k-1}]^T$ is thus the real resulting motion model as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} v'_{k-1}\Delta t\cos\left[\theta_{k-1} + \frac{1}{2}\omega'_{k-1}\Delta t\right] \\ v'_{k-1}\Delta t\sin\left[\theta_{k-1} + \frac{1}{2}\omega'_{k-1}\Delta t\right] \\ \omega'_{k-1}\Delta t + \gamma'\Delta t \end{bmatrix} \tag{3.7}$$

.

The real motion model of the robot is presented in (3.7) and $\mathbf{w}$ is replaced by $\mathbf{u}$ in (2.16). Thus, the Jacobian matrix $\mathbf{W}$ defined in (2.16) is rewritten as follows:

$$\mathbf{W}_k = \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k)}{\partial \mathbf{u}_k} = \begin{bmatrix} T_s\cos\beta_{k-1} & -0.5v'_{k-1}T_s^2\sin\beta_{k-1} \\ T_s\sin\beta_{k-1} & 0.5v'_{k-1}T_s^2\cos\beta_{k-1} \\ 0 & T_s \end{bmatrix} \tag{3.8}$$

where, $\beta_{k-1} = \theta_{k-1} + 0.5T_s\omega'_k$. $\mathbf{F}_k$ is the Jacobian matrix of partial derivatives of the function f with respect to state $\mathbf{x}$ and is defined in (3.9):

$$\mathbf{F}_k = \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k)}{\partial \mathbf{x}_{k-1}} = \begin{bmatrix} 1 & 0 & -v'_{k-1}T_s\sin\beta_{k-1} \\ 0 & 1 & v'_{k-1}T_s\cos\beta_{k-1} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The size of matrices $\mathbf{F}$ and $\mathbf{Q}$ depend on the structure of the mobile robot. Whereas, the size of matrices $\mathbf{z}$, $\mathbf{H}$ and $\mathbf{R}$ depend on a number of measurements or sensors. In addition, Jacobian matrix $\mathbf{H}_k$ is defined

by the sensor types, that are utilized for measurement. In this section, making use of encoder, GPS, IMU sensors. GPS sensor provides the position of the mobile robot (x, y). IMU sensor measures the robot's heading θ. Thus, the measurement model and Jacobian matrix $\mathbf{H}_k$ were derived for each sensor in the next subsections.

Moreover, to eliminate the cumulative error when using only the wheel encoders or the dead - reckoning method, and improve the performance of the localization system, in this section, three approaches are presented, as shown in Fig 3.4. Figure 3.4 shows three approaches: (I) Combining

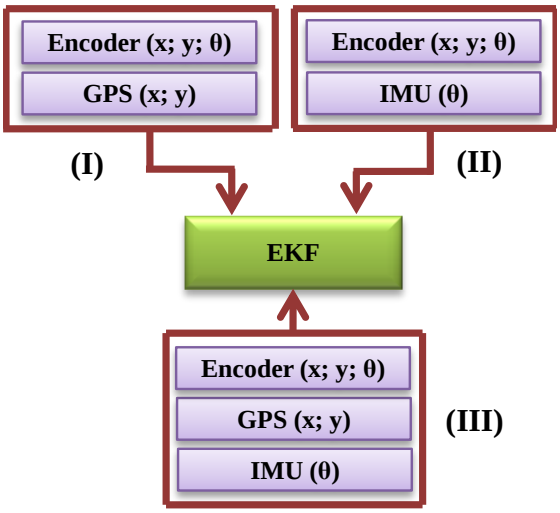


Figure 3.4: The proposed approaches

Encoder and GPS; (II) Combining Encoder and IMU; (III) Combining Encoder, GPS and IMU.

a. Global Position Systems (GPS)

GPS provides the absolute position of the robot in the environment. We have a measurement vector, as follows:

$$\mathbf{z}_{k\_gps} = \begin{bmatrix} x_{gps} \\ y_{gps} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \tag{3.10}$$

Therefore, the Jacobian matrix $\mathbf{H}$ is determined as follows:

$$\mathbf{H}_{gps} = \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.11}$$

b. Inertial measurement unit (IMU)

We have the measurement vector which is the orientation component of IMU, as follows:

$$\mathbf{z}_{k\_imu} = [\theta_{imu}] = [\theta_k] \tag{3.12}$$

Then, we obtained matrix $\mathbf{H}$ as follow:

$$\mathbf{H}_{imu} = \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

c. Combining GPS and IMU in the measurement model

Then the measurement vector is

$$\mathbf{z}_{k\_gps\_imu} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{gps} \\ y_{gps} \\ \theta_{imu} \end{bmatrix} \tag{3.14}$$

Therefore, the Jacobian matrix $\mathbf{H}_k$ is defined as follows:

$$\mathbf{H}_{gps\_imu} = \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

3.1.2. Results and discussions

a. Simulation setup

To verify the usefulness of our localization system, the proposed system have been implemented and tested in a Matlab based simulation. The kinematic model of the differential drive mobile robot introduced in Section 2.1.2 is made use of. In order to accomplish that, two scenarios are created, which their sizes are 75 x 60$[m^2]$ and 55 x 40$[m^2]$, as shown in Fig. 3.5(a) and Fig. 3.6(a), respectively.

In the first scenario, the trajectory of the mobile robot is the sinusoidal trajectory. The initial pose of robot is $(0, 0, 0)$, the goal pose is $(76, 32, 28)$, and the traveling time is 52[s]. Whereas, the trajectory of the robot is the circular trajectory with anti-clockwise direction in the second scenario. The initial pose of the robot is $(0, 0, 0)$, the diameter of the circle is 40[m], and the traveling time is 63[s]. The maximum linear velocity and angular velocity of the mobile robot are 2[m/s] and 0.4[rad/s], respectively, in both scenarios. The sampling time of the EKF filter is 10 [ms].

From the equations of the process shown in Fig. 3.3, it is recognizable that the efficiency of the EKF filter mainly depends on the estimation of white Gaussian noises $\mathbf{w}_k$ and $\mathbf{v}_k$. Moreover, the noises are featured by covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively. Therefore, determining the values for the parameters of $\mathbf{Q}_k$ (3.2) and $\mathbf{R}_k$ adjudicate the quality of the EKF filter. The covariance matrix $\mathbf{Q}_k$ is predetermined as constant

by our experiences, as follows:

$$\mathbf{Q} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.0685 \end{bmatrix} \tag{3.16}$$

The measurement noise $\mathbf{v}_k$ with the covariance matrix $\mathbf{R}_k$ is predefined in (3.17). In which, if the measurement data is the coordinates (x, y), the covariance matrix is $\mathbf{R}_{(x,y)}$, while the measurement data are both the coordinates and the angle (x, y, θ), the covariance matrix is $\mathbf{R}_{(x,y,\theta)}$. In addition, if the measurement signal is only the angle θ, the covariance matrix is $R_\theta = 0.0685$

$$\mathbf{R}_{(x,y)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \mathbf{R}_{(x,y,\theta)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.0685 \end{bmatrix} \tag{3.17}$$

To compare the simulation results between approaches, a statistical data analysis of all the simulations is carried out. To accomplish that, the Mean Error (ME) and Mean Square Error (MSE), which are computed in (3.18), are utilized in this section.

$$ME = \frac{1}{n}\sum_{k=1}^{n} NE_k; \quad MSE = \frac{1}{n}\sum_{k=1}^{n}(NE_k - ME)^2 \tag{3.18}$$

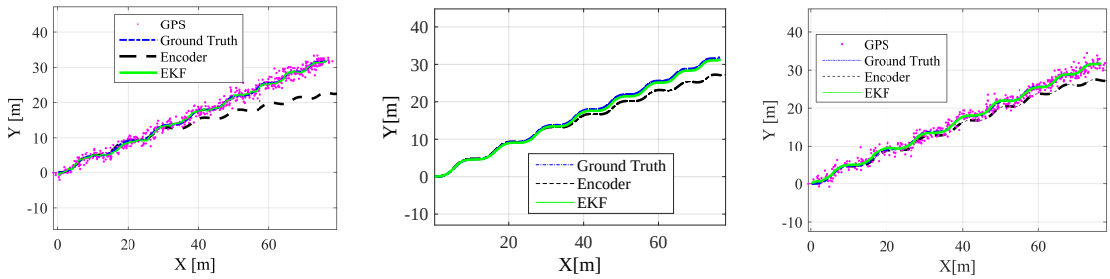
where, n is the number of samples, NE is calculated in (3.19).

$$NE_k = \sqrt{(x_{ekf} - x_{true})^2 + (y_{ekf} - y_{true})^2} \tag{3.19}$$

In this study, 520000 samples in the sinusoidal scenario and 630000 in the circular scenario have been collected .

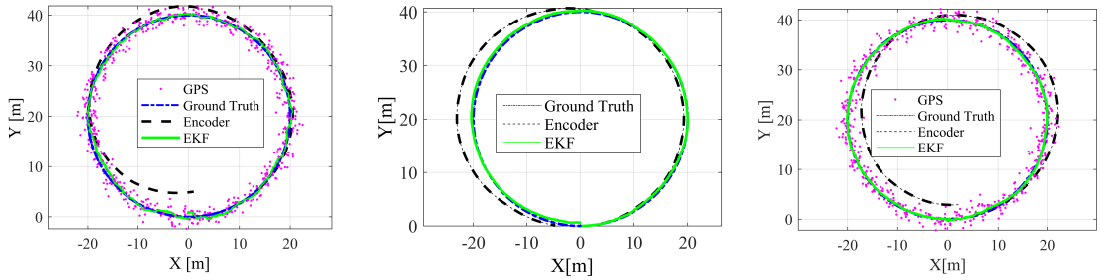
b. Simulation results

The simulation results of the two conducted simulations are shown in Figs. 3.5, 3.6, 3.7, Table 3.1, and Table 3.2. Figure 3.5 and 3.6 show the trajectories of the mobile robot, including: The blue dash lines are the expected trajectories (ground truth); The black dash lines are the trajectories derived directly from the encoder data; The magenta dots are the GPS data; and The green dash lines are the estimated trajectories of the mobile robot using our proposed localization system. Whereas, Table 3.1, Table 3.2, and Figure 3.7 illustrate the statistical data analysis of two conducted simulations. As can be seen in Figs. 3.5 and 3.6,



(a) Encoder and GPS (b) Encoder and IMU (c) Encoder, GPS and IMU

Figure 3.5: The sinusoidal trajectories of the mobile robot in three approaches.



(a) Encoder and GPS (b) Encoder and IMU (c) Encoder, GPS and IMU

Figure 3.6: The circular trajectories of the mobile robot in three approaches.

the black dash lines are very far from the ground truth trajectories of

the mobile robot, because the errors are accumulated over time during the robot's navigation. This is the weakness of the encoder-based odometry method, which derives directly from encoders. In contrast, the green lines are approximated to the real robot's trajectories in three approaches and both sinusoidal and circular trajectories. This illustrates that, the proposed extended Kalman filter-based localization system is able to provide the higher accuracy than the encoder-based odometry method. In addition, Table 3.1, Table 3.2 and Figure 3.7 depict that using approach III the values of the mean error and mean square error are smallest, and those values in the encoder method are biggest, in both sinusoidal and circular trajectories. This indicates that, the proposed localization system with approach III outperforms conventional system in terms of accuracy of estimating the pose of the mobile robot in dynamic environments.

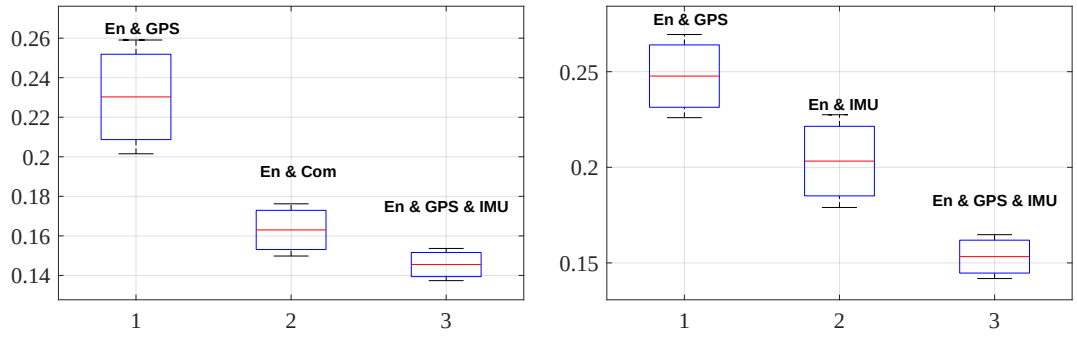
Table 3.1: Mean error for the three approaches

Sensors	**Sinusoidal trajectory**	**Circular trajectory**
Encoder	37.4074	35.1793
Encoder + GPS	0.2303	0.2477
Encoder + IMU	0.1630	0.2032
Encoder + GPS + IMU	**0.1455**	**0.1533**

Table 3.2: Mean square error for the three approaches

Sensors	**Sinusoidal trajectory**	**Circular trajectory**
Encoder	6.1162	5.9312
Encoder + GPS	0.0289	0.0217
Encoder + IMU	0.0189	0.0243
Encoder + GPS + IMU	**0.0081**	**0.0115**

In summary, the simulation results show that the goal of the proposed method is using the sensor fusion method with the higher precision sen-



(a) Sinusoidal trajectory (b) Circular trajectory

Figure 3.7: The mean error and mean square error of the robot's position of three approaches in two simulations.

sor, the lower the error. In detail the noise of GPS is greater than the noise of IMU in the simulation, due to the result of the second approach is more accurate than the first approach and the third approach have the most accurate in three approaches. Final, the proposed localization system is capable of providing higher accuracy mobile robot's pose than the conventional localization systems, which uses the encoder-based odometry method.

3.2. Particle filter-based localization algorithm

As presented in Section 3.1, there are many works on the localization system to improve the performance of the localization systems and mobile robot navigation systems. These conventional systems have achieved certain results. For instance, several systems have used KF/EKF algorithms to fuse the data from sensors [12], [13], [15]. Other researchers used particle filter (PF) to improve the accuracy of the localization system [19], [20]. A self-localization technique for a mobile robot using a PF algorithm and the active beacon system is proposed in [19]. The

main idea of the method is to estimate the position and heading value of the mobile robot using the ultrasonic sensor. The particle filter is used to eliminate process and measurement noise. In [20], the authors propose a novel hybrid particle/finite impulse response filtering algorithm for improving the reliability of particle filter-based localization schemes in cluttered and noisy environments. The hybrid system detects the particle filter failures and recovers the failed particle filter by resetting the particle filter using the output of an auxiliary finite impulse response filter. In another recent study [66] researchers used the odometric model of the golf cart to compute the vehicle position and orientation. In their paper, the authors utilize a neural network model in order to learn the odometric model from sensor data.

Although the aforementioned alorithms are capable of improving the performance of the localization systems. They have mainly proposed in the case the localization system receives all information from the observation system. To the best of our knowledge, there have not been satisfactory researches on localization systems in the interrupted information or when the signal is completely lost. Therefore, in this section, an effective localization system based on the particle filter and fusion sensor technique is proposed to estimate and predict the pose of the mobile robot equipped with an encoder, GPS and IMU sensors.

The remainder of the section is organized as follows. Subsection 3.2.1 presents the proposed particle filter-based mobile robot localization system. Subsection 3.2.2 shows the simulation results and discussion.

3.2.1. Construction of PF-based localization algorithm

The mobile robot is being tracked in the dynamic environment. The pose of the robot is measured by some external system, such a GPS, or/and IMU. Along the path, it will drive through a roofed area or where no measurement can be made or the partially lost signal. We also know the motion commands $\mathbf{u} = [v, w]^T$ send to the mobile robot, but the robot will not execute the exact commanded motion due to model in accuracy. Therefore, in this subsection, the particle filter is utilized to reduce the effects of noise in the measurement data and get a more accurate estimation of the pose of the robot. The particle filter is ideally suited for estimating the state of such kind of systems, as it can deal with the inherent non – linearity. Two models are used in the particle filter, these are the motion model and the measurement model presented in (3.7), (3.14), respectively.

The main objective of particle filtering is to "track" a variable of interest as it evolves over time, typically with a non – Gaussian and potentially multi – modal probability density function. The basis of the method is to construct a sample – based representation of the entire probability density function. A series of actions are taken, each one modifying the state of the variable of interest according to some model. Moreover, at certain times, an observation arrives that constrains the state of the variable of interest at that time. Multiple particles of the variable of interest are used, each one associated with a weight that signifies the quality of that specific particles. An estimate of the variable

of interest is obtained by the weighted sum of all the particles.

The algorithm is recursive in nature and operates in two phases: **prediction** and **update**. After each action, each particle is modified according to the existing model (prediction stage), the addition of random noise in order to the existing model to simulate the effect of noise on the variable of interest. Then, each particle's weight is re-evaluated based on the latest sensory information available (**update stage**). At times the particles with small weights are eliminated, a process called **resampling**. More formally, the variable of interest at time k is represented as a particle set $\mathbf{S}_k$ of N samples, each particle is $S_k^{[j]} = \left[\mathbf{x}_k^{[j]},\ w_k^{[j]}\right] :\ j = 1 \ldots N$ (in our case the pose of moving robot $\mathbf{x}_k = [x_k,\ y_k,\ \theta_k]^T$), where the index j denotes the particle and not the robot, each particle consisting of a copy of the variable of interest and a weight ($w_k^{[j]}$) that defines the contribution of this particle to the overall estimate of the variable. If we know the particle density filter (pdf) of the robot at the previous instant (*time k-1*) then the effect of the action is modeled to obtain a *prior of the pdf* at time k (prediction).

Prediction: The step uses the *previous state* to predict the current state based on the system model (3.7). In order to predict the probability distribution of the pose of the moving robot after a motion needs to have a model of the effect of noise on the resulting pose. Many different approaches have been used (see [67] for an overview), most of which use an additive Gausian noise model for the motion. If the robot′s previous pose is $\mathbf{x}_{k-1} = [x_{k-1},\ y_{k-1},\ \theta_{k-1}]^T$, the resulting pose $\mathbf{x}_k = [x_k,\ y_k,\ \theta_k]^T$ is defined following (3.7). The noise model is applied separately to each of

Algorithm 3: Proposed PF algorithm

input : Particle filter input $(\mathbf{S}_{k-1}, \mathbf{u}_{k-1}, \mathbf{z}_k)$
output: $\overline{\mathbf{S}}_k,\ w_k^{[j]}$

begin
  Initialize parameter set $\overline{\mathbf{S}}_k = \mathbf{S}_k = \emptyset$
  for j=1 **to** N **do**
    Generate a particle
      //Motion model
      $$\mathbf{x}_k^j = \begin{bmatrix} x_k^j \\ y_k^j \\ \theta_k^j \end{bmatrix} == \begin{bmatrix} x_{k-1}^j \\ y_{k-1}^j \\ \theta_{k-1}^j \end{bmatrix} + \begin{bmatrix} v'_{k-1}\Delta t\cos\left[\theta_{k-1} + \frac{1}{2}\omega'_{k-1}\Delta t\right] \\ v'_{k-1}\Delta t\sin\left[\theta_{k-1} + \frac{1}{2}\omega'_{k-1}\Delta t\right] \\ \omega'_{k-1}\Delta t + \gamma'\Delta t \end{bmatrix}$$
      //Measurement model
      $$\mathbf{z}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{gps} \\ y_{gps} \\ \theta_{imu} \end{bmatrix}$$
    //Calculate an important weight
      $\mathbf{E}_k^j = \mathbf{x}_k^j - \mathbf{z}_k^j$
    if$(\hat{\mathbf{z}}_k^j = [\hat{x}_k,\ \hat{y}_k]^T)$
      $EN^{[j]} = Sqrt(\left[\mathbf{E}_k^j(1)\right]^2 + \left[\mathbf{E}_k^j(2)\right]^2)$
    elseif$(\hat{\mathbf{z}}_k^j = \left[\hat{\theta}_k\right])$
      $EN^{[j]} = Sqrt(\left[\mathbf{E}_k^j(3)\right]^2)$
    elseif$(\hat{\mathbf{z}}_k^j = \left[\hat{x}_k,\ \hat{y}_k, \hat{\theta}_k\right]^T)$
      $EN^{[j]} = Sqrt(\left[\mathbf{E}_k^j(1)\right]^2 + \left[\mathbf{E}_k^j(2)\right]^2 + \left[\mathbf{E}_k^j(3)\right]^2)$
    Endif
    $w_k^{[j]} = |2\pi R_k|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}\left(EN^{[j]}\right)^{0.4}\right\}$
    $\bar{\mathbf{S}}_k = \bar{\mathbf{S}}_k + \left[\mathbf{x}_k^{[j]}, w_k^{[j]}\right]$
  Normalize: $w_k^{[j]} = \frac{w_k^{[j]}}{\sum_{j=1}^{N} w_k^j}$
  Resampling using algorithm 4
  end for

the types of motion (rotations and translation) because they are assumed independently. With the motion command $\mathbf{u}'_k = [v',\ \omega']^T$ where $v' = v + sd1 * rand_N$; $\omega' = w + sd2 * rand_N$; $\gamma' = sd3 * rand_N$ (sd1, sd2, sd3 are standard deviation errors). They represent the uncertainty in the linear velocity, the angular velocity, and the orientation respectively.

Replacing v', ω', γ' into (3.7) obtains the model with noise.

Update: The phase uses information obtained from to update the particle weights to accurately describe the pdf of the robot. In another word, using a current sensor measurement (3.14) to correct the predicted state. At line 20 (in Algorithm 3) calculates the importance of weight $w_k^{[j]}$ for each particle $\mathbf{x}_k^{[j]}$. The importance of weight is used to correct the mismatch between the proposal distribution and the desired target distribution [53]. According to the measurement model, weights are assigned by likelihood response (in Algorithm 3). The measurement likelihood function computes the likelihood for each predicted particle based on the error norm (EN) between predicted measurement and actual measurement (line 18). In this particular situation, predicted particle considered as the predicted is an Nx3 vector and measurement is a 1x n vector (n is the number of measured elements, in this section has three different cases of measurement).

Resampling (for more detail in algorithm 4): One of the problems that appear with the use of PF is the depletion of the population after a few iterations. Most of the particles have drifted far enough for their weight to become too small to contribute to the pdf of the moving robot. In this section, the Multinomial method has been chosen.

The proposed PF-based localization algorithm is summarized in Algorithm 3. It inputs the particle set $\mathbf{S}_{k-1}$ at time k-1, motion control $\mathbf{u}_{k-1}$, measurements $\mathbf{z}_k$. It outputs the particle set $\mathbf{S}_k$. Here, N denotes the total number of particles used in this algorithm.

Algorithm 4: Presents resampling methods based on the cumulative sum of the normalized weights

input : Particle filter input $(\mathbf{S}_k, w_k)$
output: $\overline{\mathbf{S}}_k$

begin
  $[\hat{\mathbf{x}}_{\mathrm{k}}^{[\mathrm{i}]}] = \mathrm{Resembel}[\{\mathbf{x}_{\mathrm{k}}^{[\mathrm{j}]}, w_k^{[j]}\},\ N]$
  Multinomial choice
    N = length(w);
    Q = CumulativeSum(w);
    Q(N)=1;
    i=0;
    While(i≤N)
      Sampl = rand(1,1);
      j=1;
      While($Q_k^{[j]} < Sampl$)
        j=j+1;
      endwhile
      Index(i) =j;
      i= i+1;
      $\hat{\mathbf{x}}_{\mathrm{k}}^{[\mathrm{i}]} = \mathbf{x}_{\mathrm{k}}^{[\mathrm{j}]}$
      add $\hat{\mathbf{x}}_{\mathrm{k}}^{[\mathrm{i}]}$ to $\overline{\mathbf{S}}_k$
    endwhile
  Return $\overline{\mathbf{S}}_k$

3.2.2. Results and discussions

3.2.2.1. Simulation setup

To verify the usefulness of the proposed localization system, the proposed algorithm have been implemented and tested in a MATLAB based simulation. In order to accomplish that, four scenarios are created , which their size are $12 \times 8[m]^2$, as shown in Fig. 3.8. The robot kinematic model and measurement model are presented in (3.7), (3.14), respectively. The trajectory of the mobile robot is the curve trajectory. The initial pose of the robot is $[0,\ 0,\ 0]^T$. The robot will navigate following the trajectory with the traveling time of 20[s]. The maximum linear velocity and angular velocity of mobile robot are 80[cm/s] and

0.008[rad/s], respectively. Empirically setting the number of particles is 500 particles. The state of all particles is randomly initialized using a normal distribution. Each particle contains 3 state variables [x, y, θ]. Assume that, the measurement of all three pose components have the same error distribution. Here the sensor reading is just simulated by adding Gaussian noise to the ground truth data (with the noise standard deviation sd = 0.2) and defused noise into the system model with noise standard deviations: sd1 = 4; sd2 = 1.2; sd3 = 0.35. The trajectory of the mobile robot is divided into three parts to show the performance of the particle filter-based localization system of the mobile robot, as shown in Fig. 3.8.

In the first part, the mobile robot navigates in a good environment condition, where the observation system works well. In other words, the localization system receives all the sensor data.

In the second part, the mobile robot navigates into the roofed area. In this area, all of the signals are lost or a part of them is received by the localization system. If the entire signals are lost, the particle filter-based localization only uses the prediction model to predict the pose of the robot. While in the later, the localization system can use the available signal to correct the prediction step.

In the third part, the mobile robot goes out of the roofed area. With new measurements, the estimated pose gradually converges back to the actual pose.

In order to compare the simulation results between approaches in terms of quantity, statistical data analysis of all the simulations is car-

ried out. To accomplish that, the mean error (ME), which is computed in (3.20), is utilized In this section.

$$ME = \frac{1}{n}\sum_{k=1}^{n} NE_k \tag{3.20}$$

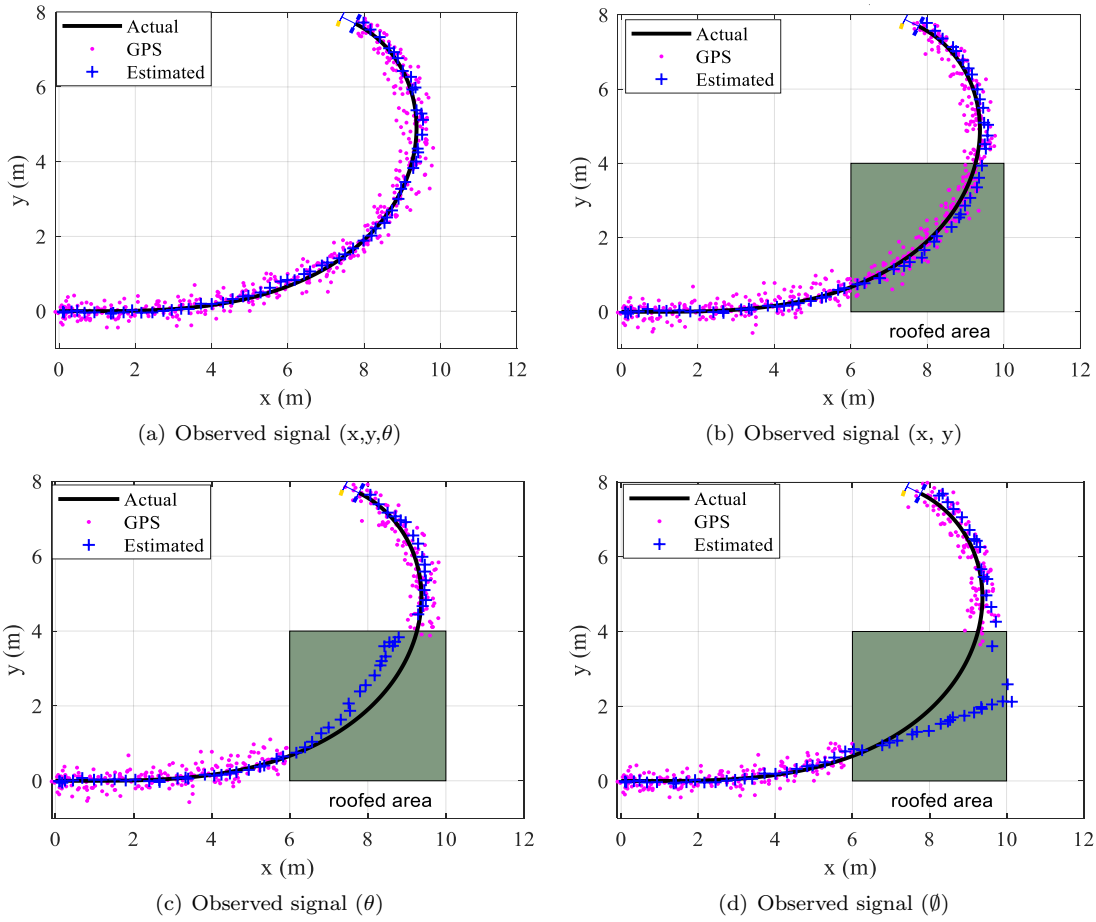
where n=500 is the number of samples, NE is calculated in

$$NE_k = \sqrt{(x_{PF} - x_{true})^2 + (y_{PF} - y_{true})^2} \tag{3.21}$$

3.2.2.2. Simulation results

The simulation results are shown in Fig. 3.8, and Table 3.3. The Figure 3.8 shows four scenarios such as (a) robot receives [x, y, θ] in the entire trajectory, (b) robot gets [x, y] in the roofed area, (c) the [θ] information is available in the roofed area, (d) all the sensor signals are lost in the roofed area. The figure also illustrates the trajectories of the mobile robot, including the black line is the desired trajectory; the magenta dots are the GPS data, and the blue crosses are the estimated trajectory of the mobile robot using the proposed algorithm. Whereas, Table 3.3 illustrates the mean square errors of four approaches when the localization system uses or does not use the particle filter algorithm. Figure 3.8(a) shows the simulation results when the localization system of the mobile robot can receive all signals [x, y, θ] from the observation system. As can be seen in Fig. 3.8(a), the estimated trajectory is approximated to the actual trajectory of the robot. As a result, the value of mean error 0.1098 is the smallest, as seen in Table 3.3.

In Fig. 3.8(b), the mobile robot navigates through the roofed area and the signal θ is lost. In other words, the localization system can only use



(a) Observed signal (x,y,θ)

(b) Observed signal (x, y)

(c) Observed signal (θ)

(d) Observed signal ($\emptyset$)

Figure 3.8: The simulation results using PF localization

the observed [x, y] signal in the correction phase. In this case, the mean error value is 0.1451. It indicates that the performance of the localization system is decreased, as also shown in Fig. 3.8(b).

In Fig. 3.8(c), the mobile robot navigates through the roofed area and the signals [x, y] are lost. In another word, the localization system can only use the observed θ signal in the correction phase. In this case, the mean error value is 0.2406. It indicates that the performance of the localization system is decreased, as also shown in Fig. 3.8(c).

In the final case, all of the signals [x, y, θ] from the sensor system are lost (observed [$\emptyset$]). In other words, in the roofed area, the localization

only uses the prediction phase to estimate the pose of the mobile robot. As a result, mean error value 0.3957 is greatest and the estimated robot trajectory is far from the actual trajectory, as shown in the last row in Table 3.3 and Fig. 3.8(d).

The comparison of estimated trajectories among the three approaches illustrates that using the particle filter algorithm the localization can track the robot's position and trajectory after going out of the roofed area. In other words, after lost signal in interval time the mobile robot still likely follows the trajectory. However, the performance of the measured parameters is gradually reduced according to the measured parameters. In addition to the results shown in Fig. 3.8, the comparison

Table 3.3: The mean error of the proposed localization system and existing localization system.

Observed Signal	Encoder-based odometry algorithm	Proposed localization algorithm
$[x, y, \theta]$	0.2510	**0.1098**
$[x, y]$	0.2479	**0.1451**
$[\theta]$	0.9648	**0.2406**
$[\emptyset]$	0.9722	**0.3957**

of mean error of the proposed particle filter-based localization algorithm and the conventional algorithm are made. The comparison results are shown in Table 3.3. The first column shows the mean error values of the conventional algorithm, while the second column shows the mean error values of our proposed system. The results in Table I illustrates that the mobile robot equipped with our proposed localization system outperforms the conventional localization system in term of mean errors.

In summary, the simulation results indicate that the proposed particle filter-based localization algorithm is able to apply and improve the

performance of the autonomous mobile robot when it *navigates in interrupted sensor data information.*

3.3. Remarks and discussions

In this chapter, two efficient localization algorithms have been proposed for autonomous mobile robots in dynamic environments including EKF-based localization algorithm and PF-based localization algorithm. The main idea of the proposed algorithms are to fuse the data sources from the sensor system including wheel encoders, IMU, GPS sensors, to enhance the accuracy of the robot's pose estimation when it navigates in *two different cases.*

The first case, when sensor signals are *sufficient* and noise distributions are *Gaussian* distribution, the EKF - based localization algorithm has been made of used. Two simulation results corresponding with two sinusoidal and circular trajectories of the autonomous mobile robots have conducted. The simulation results indicate that, the proposed localization algorithm is capable of providing higher accuracy mobile robot's pose than conventional localization systems.

The second case, when information get from sensor systems is *insufficient* or the sensor data signals are *interrupted*, and noises have *Non-Gaussian/Gaussian* distribution, PF - based localization algorithm has been proposed. To illustrate the effectiveness of the proposed system, we implement it and conduct simulation in a simulation environment. The simulation results show that, although the mobile robot navigates into areas where a part or entire of the signals can be lost, the robot's

positions are still close to ground truth data. In other words, the simulation results illustrate that the performance of the proposed localization system is significantly improved in terms of accuracy.

The output of the proposed localization systems are the robot's pose including robot's position and orientation, which are then used as the input of the motion planning system, as shown in Fig. 1.1. In the next chapter, three proposed motion planning systems are going to present for the autonomous mobile robots in the dynamic environments.

Note that the researches of this chapter have been published in [C1] and [J2] .

Chapter 4
DEVELOPING EFFICIENT MOTION PLANNING SYSTEMS

The ability to autonomously and safely navigate in real-world dynamic environments is crucial for autonomous mobile robots. To achieve that ability, the most important issue is that the mobile robots must avoid both static and dynamic obstacles during its navigation and navigate towards a given target. Therefore, it is very necessary to develop navigation systems which are able to drive the mobile robots in real-world environments with dynamic obstacles. In the previous chapter, we proposed the two localization algorithms which were used to improve accuracy of output signals of the localization system. In other words, the input signals of the motion planning system including robot's position and orientation are enhanced significantly, as shown in Fig. 4.1.

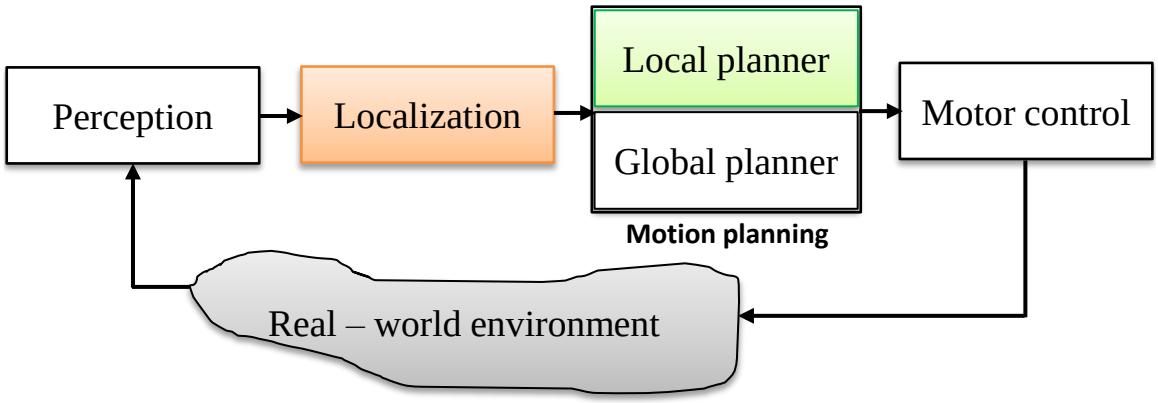


Figure 4.1: The navigation framework for autonomous mobile robot.

In this chapter, we propose motion planning systems of the mobile robots which are capable of driving the mobile robots to *proactively* and *safely avoid* dynamic obstacles in the real-world environments.

The motion planning systems include of two sub-systems, as shown in Fig. 4.1: (i) global planner (or path planning); (ii) local planner (or obstacle avoidance). The *Global planner* is used to construct safe and collision free paths of the robot from an initial point to the given goal point with a given map. In contrast, the*local planner* means recalculating the constructed paths to avoid possible collision, especially moving obstacles. Therefore, in order for the mobile robots to move safely in the dynamic environments, we focus on developing the *local planning algorithms* (or obstacle avoidance algorithms) in the motion planning systems. The proposed motion planning systems should take into account *the kinodynamic constraints* of the robot, and the *potential collisions* of the robots with surrounding obstacles. Particularly, the robots should predict *obstacle's future states* including position and orientation as well as *future trajectory* of the obstacles in their vicinity.

To address the aforementioned issues, in this chapter, *three new local planners* of the motion planing system for the autonomous mobile robots in the dynamic environment are proposed, including the *enhanced dynamic window approach (EDWA)*, *proactive timed elastic band (PTEB)*, and *extended timed elastic band (ETEB)* algorithm. In addition, an *efficient navigation system*, which integrates the proposed EKF-based localization algorithm and a proposed ETEB algorithm for online trajectory planning, is also introduced. The effectiveness and feasibility of

the proposed algorithms are validated through a series of experiments in both simulated and real-world environments.

This chapter is organized as follows. The proposed hybrid (EDWA) algorithm based on DWA technique and HRVO model to enhance the performance navigation system of the robot in the dynamic environment is presented in Section 4.1. To deal with the limitation of the EDWA algorithm, the description of the proposed PTEB algorithm, combining HRVO technique and TEB algorithm, is given in Section 4.2. In Section 4.3, we introduce the proposed ETEB algorithm: using motion prediction algorithm incorporating into TEB model. In order to demonstrate the feasibility of the proposed algorithms in our thesis, a completed navigation system for the autonomous mobile robot in real -world environment is shown in Section 4.4. Finally, we reach a conclusion in Section 4.5.

4.1. Proposed enhanced dynamic window approach algorithm

The dynamic environments are dynamic, uncertain, and clustered environments with the presence of humans, vehicles, and even other autonomous devices. Therefore, various navigation systems have been proposed to ensure the safe navigation of the mobile robot in such environments. The navigation frameworks can be divided into two categories based on the information used as the input of the motion planning system: (i) position-based approaches and (ii) velocity-based techniques. In the first group, only the position of the obstacles is used as the input of the system. Whereas in the second group, both position and motion

of the obstacles are utilized to develop the navigation system. Detail discussion of two groups is going to the following.

Several position-based navigation systems have been proposed for the autonomous mobile robots in recent years [21], [22], [23], [24], [25] [26] and [27]. Some mobile robot navigation systems [22], [23] and [25] are developed using the social force model [28]. The social force-based navigation systems utilize the attractive force to the goal and the repulsive forces from the obstacles to develop the motion control model. In addition, the repulsive forces is computed using the distance form the robot to the obstacles.

In an alternative approach, the researchers utilize the dynamic window approach algorithm proposed by Fox et al. [29] to develop the mobile robot navigation systems [21] and [24]. These navigation systems take into account the motion dynamic of the mobile robot and utilize the closest distance from the robot to the surrounding obstacles for obstacle avoidance.

More recently, a few navigation systems [26] and [27] are developed using the TEB technique for avoiding obstacles. To maintain the separation from the obstacles, the TEB-based navigation systems take into consideration the distance from the proposed robot's trajectory to the surrounding obstacles.

Despite the fact that the aforementioned navigation systems have been able to generate the safe navigation for the mobile robots in real-world environments, they do not proactively deal with potential collision with the surrounding obstacles. Because, these methods are typically a reac-

tive control technique, and the navigation systems only take into account the position of the obstacles. In other words, moving obstacles are assumed to be stationary. As a results, these navigation systems might be sufficient for the obstacle avoidance in static and semi-dynamic environments, but foresighted evasive maneuvers are not possible.

In order take into account the potential collision of the mobile robot with the surrounding obstacles, a number of autonomous robot navigation systems [30], [31], [32] and [33] based on the concept of the velocity obstacles presented by Fiorini et al. [34], have been proposed. The velocity obstacles-based navigation system takes into account the position and motion of all agents and selects the collision-free velocity command from the two-dimensional velocity space in the $xy-plane$ for each agent. Although these methods have been successfully verified in the real-world environments, they might not be able to handle all collision situations in The dynamic environments [31] and [33]. In addition, the velocity obstacles-based navigation system only utilizes the current position and velocity of the robot and the obstacles to generate the velocity command for the mobile robot. Moreover, the system does not take into account the motion dynamic of the mobile robot. Thus, it is difficult to directly utilize this velocity command to control the mobile robot in the real-world environments.

In order to overcome the aforementioned drawbacks, in this section, an EDWA technique for autonomous mobile robot navigation systems in dynamic environments is proposed . The main idea of the proposed technique is to combine the advantages of the DWA technique and the HRVO

model. Particularly, incorporating the orientation of the velocity vector generated by the HRVO model into the DWA model. By incorporating this information into the DWA technique, the mobile robot equipped with the proposed EDWA algorithm can proactively avoid obstacles and safely navigate to the given goal.

In the section, we introduce an typical example scenario to performance the proposed framework in Subsection 4.1.1. Then, the proposed EDWA algorithm is presented in Subsection 4.1.2. In Subsection 4.1.3, the proposed navigation frame work that uses EDWA model is shown. Subsection 4.1.4 and Subsection 4.1.4 show the simulation and experimental results, respectively. Finally, the last Subsection 4.1.5 contains remarks.

4.1.1. Problem description

In this study, a dynamic environment with the presence of an autonomous mobile robot and O obstacles o_1 and o_2 in the robot's vicinity is considered, as shown in Fig 4.2. The robot is requested to navigate to a goal while safely avoid the obstacles during its navigation. The curved dashed line is the intended trajectory of the mobile robot.

Assuming that the robot state is $\mathbf{s}_r = [x_r, y_r, \theta_r, v_r, \omega_r]^T$, where $\mathbf{p}_r = [x_r, y_r]^T$ is the position, θ_r is the orientation, v_r is the linear velocity, and ω_r is the angular velocity. The dynamic motion of the mobile robot is $(v_{min}, v_{max}, \omega_{min}, \omega_{max}, \dot{v}_{max}, \dot{\omega}_{max})$, where, v_{min}, ω_{min} are the minimum linear and angular velocities, respectively, v_{max}, ω_{max} are maximum linear and angular velocities, and $\dot{v}_{max}, \dot{\omega}_{max}$ are maximum linear and

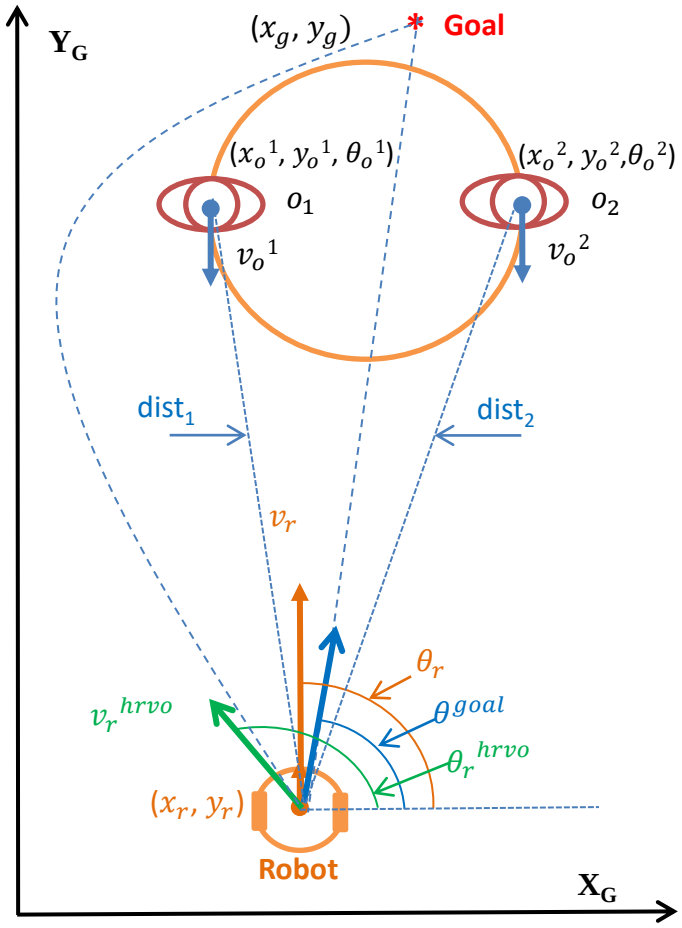


Figure 4.2: The example scenario of the dynamic environments including a mobile robot and two dynamic obstacles.

angular accelerations, respectively. The goal position of the robot is $\mathbf{p}_g = [x_g, y_g]^T$. There are N obstacles appearing in the vicinity of the robot $O = \{\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_N\}$, where $\mathbf{o}_i$ is the i^{th} obstacle. The state of the obstacle $\mathbf{o}_i$ is represented as $\mathbf{s}_o^i = [x_o^i, y_o^i, \theta_o^i, v_o^i]^T$, where $\mathbf{p}_o^i = [x_o^i, y_o^i]^T$ is the position, θ_o^i is the orientation, and v_o^i is the linear velocity. The radius of the robot and obstacle are r_o and r_r, respectively.

Note that all of the terms in this section will also be used in the rest of the thesis.

4.1.2. Construction of the EDWA algorithm

The dynamic window approach (DWA) model is successfully applied to several mobile robot navigation systems in real-world environments [21], [1] and [24]. Because, they take into account the dynamic motion of the mobile robot, including actual speed, acceleration and physical limits. Moreover, they consider the surrounding obstacles in their vicinity. Therefore, it can eliminate unreachable velocities coming from the limited accelerations of the robot. In addition, all velocity pairs, which are not able to stop the mobile robot before colliding with the obstacles, are also eliminated. However, this method is typically a reactive control, so it does not proactively deal with potential collisions when the robot moves close to the obstacles, especially the moving obstacles. Thus, it might be insufficient to apply these algorithms to the mobile robot in dynamic environments.

Whereas, the HRVO model [31] is a velocity-based approach synthesizing the motion of neighbourhood agents for collision avoidance. It has a big advantage of proactive collision avoidance by taking into account the motion of the obstacles. Thus, the robot is able to avoid the potential collision with the surrounding obstacles. As a results, this model is successfully applied to several navigation systems of the mobile robot [33] and [32]. However, the HRVO model only takes into account the current velocity of the robot and the obstacles to generate the robot velocity command $\mathbf{v}_r^{hrvo}$ for the next time step. Therefore, it is difficult to directly use this velocity to control the mobile robot in real-world

environments.

To exploit the advantages of both DWA and HRVO algorithms, an EDWA algorithm for autonomous mobile robot in dynamic social environments is proposed . In other words, the developed EDWA algorithm take into account both the motion dynamic of the mobile robot and its potential collision with the surrounding obstacles. To accomplish this, in the objective function (2.26), the target heading function $head(\upsilon, \omega)$ is modified. Particularly, in (2.27) instead of using the predicted orientation of the mobile robot θ_r, the orientation of the velocity vector generated by the HRVO model is utilized. More specifically, the orientation θ_r^{hrvo} of the velocity vector $\mathbf{v}_r^{hrvo} = [\upsilon_x, \upsilon_y]^T$ generated by the HRVO model in (2.29) is used to compute the new target heading function as follows:

$$head^{hrvo}(\upsilon, \omega) = 180^o - |\theta^{goal} - \theta_r^{hrvo}| \tag{4.1}$$

$$\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x) \tag{4.2}$$

Finally, the objective function of the DWA model in (2.26) is replaced by the new objective function as follows:

$$G'(\upsilon, \omega) = \alpha head^{hrvo}(\upsilon, \omega) + \beta dist(\upsilon, \omega) + \gamma vel(\upsilon, \omega) \tag{4.3}$$

The proposed EDWA algorithm is presented in detail in Algorithm 5. Its inputs are the robot state $\mathbf{s}_r$, the goal position $\mathbf{p}_g$, the obstacle state $\mathbf{s}_o$, and its outputs is the control command $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$. The proposed EDWA algorithm consists of three steps including: (i) calculate the search space of the velocities V_r (Lines 4–7 of the Algorithm 5), (ii) compute the orientation of the velocity vector generated by the HRVO

model (Lines 8–9 of the Algorithm 5), and (iii) select the efficient velocity control command (Lines 10–19 of the Algorithm 5). The motion dynamic of the mobile robot $(v_{max}, \omega_{max}, \dot{v}_{max}, \dot{\omega}_{max})$ is utilized to compute the search space V_r. Using the proposed EDWA algorithm, the robot navigation system is capable of generating an efficient velocity command $\mathbf{u}_r$=$[v_r, \omega_r]^T$, which proactively handle potential collisions with dynamic obstacles in its surrounding environment and approach a given goal.

Algorithm 5: Proposed enhance dynamic window approach algorithm

input : robot state $\mathbf{s}_r$, goal position $\mathbf{p}_g$, obstacle state $\mathbf{s}_o$
output: Control command $\mathbf{u} = [v_r, \omega_r]^T$

1 **begin**
2 Initialize parameter set α, β, γ
3 Set motion dynamic $v_{max}, \omega_{max}, \dot{v}_{max}, \dot{\omega}_{max}$
4 Compute V_s = possible velocities
5 Compute V_a = admissible velocities
6 Compute V_d = reachable velocities
7 Compute $V_r = V_s \cap V_a \cap V_d$
8 Run HRVO to generate $\mathbf{v}_r^{hrvo} = [v_y, v_x]^T$
9 Compute $\theta_r^{hrvo} = atan2(v_y, v_x)$
10 **for** each pair of velocity $(v_i, \omega_i) \in V_r$ **do**
11 Predict robot position (x_i, y_i) using (2.4)
12 $\theta_i^{goal} = atan2(y_g - y_i, x_g - x_i)$
13 $head_i^{hrvo} = 180^o - |\theta_i^{goal} - \theta_r^{hrvo}|$
14 Compute obstacle clearance function $dist_i$ using the closest distance to obstacles
15 Compute velocity function $vel_i = |v_i|$
16 Compute the $score_i$ using (4.3)
17 Store $score_i$ in the score vector $\mathbf{S}$
18 **end for**
19 Select $\mathbf{u} = [v_r, \omega_r]^T$ using maximum score from $\mathbf{S}$

In this study, we utilize a two-wheel differential drive mobile robot platform, with the state of the robot at the time k is $s_r^k = [x_k, y_k, \theta_k]^T$. Therefore, the state of the robot at the time (k+1) is governed by (2.4) (Line 11). Then, to generate directly control signals for the motor control

model (v_r^r, v_r^l) which are the linear velocity commands of the right and left wheels of the robot, respectively, are computed using the velocity control command $\mathbf{u}_r$ as follows:

$$v_r^r = v_r + \frac{L\omega_r}{2}dt \tag{4.4}$$

$$v_r^l = v_r - \frac{L\omega_r}{2}dt \tag{4.5}$$

4.1.3. The EDWA algorithm-based navigation framework

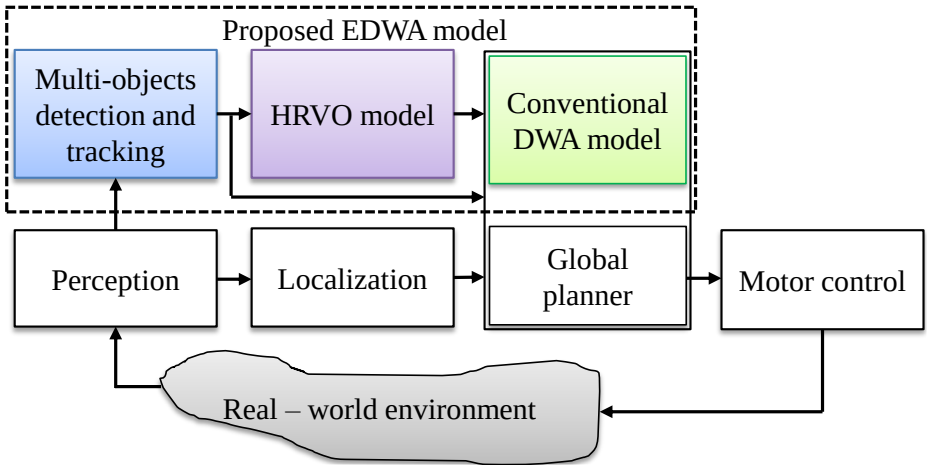


Figure 4.3: The efficient navigation system based on the EDWA algorithm

To navigate safely in dynamic social environments, autonomous mobile robots should incorporate not only the position of the surrounding obstacles but also their potential collision into navigation systems. An extended navigation scheme based on the conventional navigation scheme introduced in [1] is proposed to accomplish this, as shown in Fig. 4.3. The proposed system consists of two major parts: (i) the conventional navigation scheme, and (ii) the extended part.

The conventional navigation scheme is typically based on the composition of four functional blocks: perception, localization, motion planning,

and motor control. In the extended part, the multi-objects detection and tracking block is used to detect and track objects in the vicinity of the robot. Then the HRVO algorithm utilizes the object state including position, orientation and velocity, to model the potential collision of the robot with the surrounding objects. This information and the object state are then used as the inputs of the EDWA algorithm. The proposed EDWA algorithm functions as a local planner for any the motion planning system, and guarantees the safe navigation for the mobile robot in The dynamic environments.

4.1.4. Algorithm validation by simulations and experiments

To verify the usefulness of the proposed EDWA algorithm, we implemented and tested in a MATLAB-based simulation environment as well as conducted experiments on the Eddie mobile robot platform in the real-world environment.

a. Simulation setup

Table 4.1: Parameters set in experiments

Parameters	Value	Parameters	Value	Parameters	Value
α	3	r_r, r_o	$0.3[m]$	v_{max}	$1[m/s]$
β, γ	0.1	t_{sim}	$3[s]$	ω_{max}	$0.35[rad/s]$
α_{vision}	270^o	r_{vision}	$8[m]$	Δt	$0.25[s]$

Four typical scenarios have been created , including: (i) scenario 1 – a mobile robot avoids an obstacle moving toward shown as Fig .4.4 (a), (b), and (c); (ii) scenario 2 – a mobile robot avoids two obstacles moving toward shown as Fig .4.4 (d), (e), and (f); (iii) scenario 3 – a mobile robot avoids two obstacles in a crossing situation as shown in

Fig. 4.5 (a), (b), and (c); and (4) scenario 4 – a mobile robot avoids three dynamic obstacles in a complex situation, as shown in Fig. 4.5 (d), (e), and (f).

In each scenario, three experiments corresponding to three pairs of reactive motion planning algorithms to compare the proposed EDWA algorithm with the conventional DWA algorithm are conducted, including DWA-DWA the mobile robot and the obstacles are both installed DWA algorithm; EDWA-DWA – the mobile robot is installed the EDWA algorithm and the obstacles are installed DWA algorithm; and EDWA-EDWA – the mobile robot and the obstacles are both installed EDWA algorithm. Then comparing the experimental results between these methods to illustrate the performance of the proposed EDWA algorithm. It is noted that, the obstacle equipped with the DWA algorithm incorporates only the state of other obstacles into the motion planning system. Whereas the obstacle takes the state of both robot and other obstacles into account when it is installed the EDWA algorithm.

The mobile robot is requested to navigate from the starting position $(0, -10)$ to the goal position $(0, 9)$ while avoiding the obstacles in the surrounding environment, in each scenario. The initial velocity of the robot is set to $0.0[m/s]$. The range and field of view of the perception system of the robot and the obstacles are set to $r_{vision} = 8.0[m]$ and $\alpha_{vision} = 270^o$, respectively. The time to forward the simulated trajectory of the DWA and EDWA algorithms is set to $t_{sim} = 3[s]$. In all experiments, the values of the parameters of the DWA and proposed EDWA algorithms are empirical set and presented in Table 4.1.

In order to compare the proposed EDWA algorithm and the conventional DWA algorithm we make use of both qualitative and quantitative evaluations. Regarding to the qualitative evaluation, the trajectory of the mobile robot and the obstacles are visualized in the same figure. Whereas, in term of quantitative evaluation, we utilize three matrices, including the velocity and average velocity of the mobile robot, and the minimum distance from the robot to the surrounding obstacles. The velocity and average velocity are used to indicate the proactive robot trajectory, while the minimum distance illustrates the safe navigation of the mobile robot. In addition, the minimum distance is normalized as follows:

$$\delta_{min}(t) = e^{(-\frac{d_{min}(t)^2}{3})} \tag{4.6}$$

where $d_{min}(t)$ are the closest distances between the boundary of the robot and the boundary of all obstacles at time t. Therefore, the closer the robot to an obstacle is, the closer the value of $\delta_{min}(t)$ to 1 is.

b. Simulation results

The results in the simulation environments are shown in Fig. 4.4, Fig. 4.5, Fig. 4.6, Fig. 4.7, and Table 4.2. A video clip of our simulation results can be found at this link[1].

- **Scenario 1:** Aiming of this scenario is to verify the response of the mobile robot when it avoids an obstacle moving toward the robot in three pairs DWA-DWA, EDWA-DWA and EDWA-EDWA. The trajectories of the robot and the obstacle are shown in Fig. 4.4(a),

[1]https://youtu.be/oypDiSQTYPQ

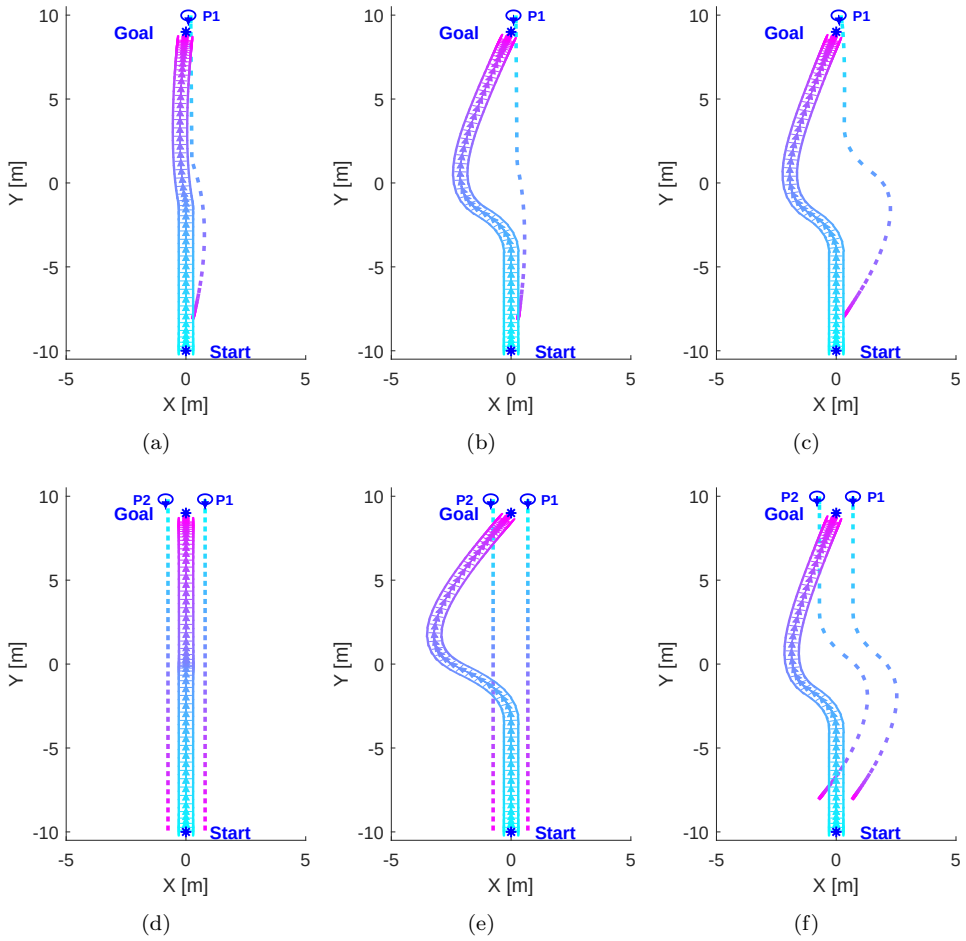


Figure 4.4: The trajectory of the mobile robot and obstacles in Scenario 1 and 2.

4.4(b) and 4.4(c), respectively. As can be seen in these figures, the mobile robot can safely avoid the obstacle and approach the given goal in all three experiments. However, the robot avoids the obstacle at the short distance, and the trajectory of the robot is very close to the trajectory of the obstacle in the pair DWA-DWA, as shown in Fig. 4.4(a). That is because the robot equipped with the conventional DWA algorithm only take the position of the obstacle into account.

In contrast, the robot equipped with the proposed EDWA algorithm

proactively avoids obstacle when the obstacle in the robot field of view ($r_{vision} = 8m$), as shown in Fig. 4.4(b). That is because the robot incorporates the potential collision of the robot with the obstacles into the motion planning system.

In addition, the robot and the obstacle can smoothly avoid each other if both of them are installed the proposed EDWA algorithm, as shown in Fig. 4.4(c). This is very properly in dynamic social environments, where the humans, vehicles, autonomous devices and the mobile robot are both shared the responsibility of avoiding each other.

- **Scenario 2:** In this scenario, the avoiding action of the mobile robot is examined when it avoids two obstacles moving toward to the robot. The trajectory of the robot and the obstacles are shown in Fig. 4.4(d), 4.4(e) and 4.4(f). The mobile robot equipped with the DWA algorithm passes the space between two obstacles, as shown in Fig. 4.4(d), while the robot equipped with the proposed EDWA algorithm can proactively avoid two moving obstacles, shown as Fig. 4.4(e). Similar to Scenario 1, when the robot and the obstacles are both installed the proposed EDWA algorithm the robot smoothly and proactively avoid two obstacles and safely moving toward the given goal.

Table 4.2: The average passing velocity of the robot

Robot and Obstacles	Scenario 1 [m/s]	Scenario 2 [m/s]	Scenario 3 [m/s]	Scenario 4 [m/s]
DWA-DWA	0.8895	0.8693	0.8910	0.7937
EDWA-DWA	**0.9531**	**0.9441**	**0.9439**	**0.9361**
EDWA-EDWA	**0.9711**	**0.9630**	**0.9527**	**0.9461**

- **Scenario 3:** Aiming of this scenario is to examine the behavior

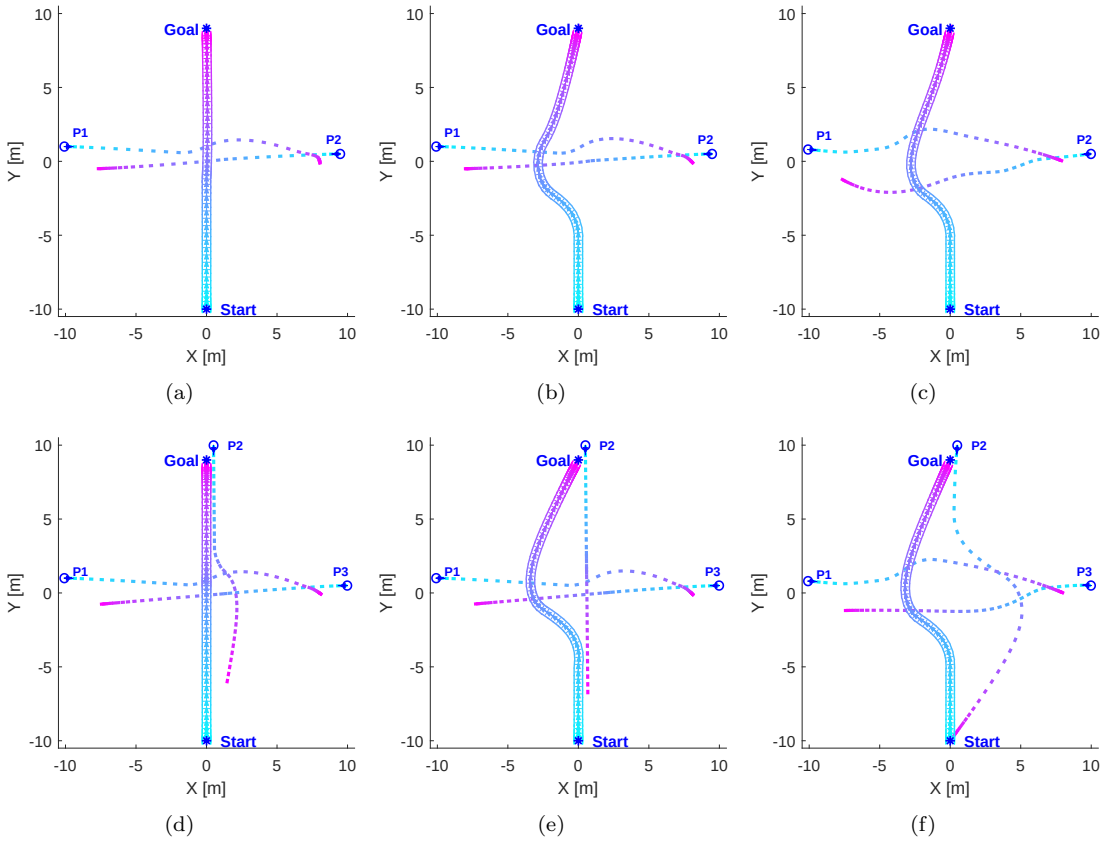


Figure 4.5: The trajectory of the mobile robot and obstacles in Scenario 3 and 4.

of the mobile robot when it avoids two obstacles moving toward to each other. The trajectory of the robot and the obstacles are shown in Fig. 4.5(a), 4.5(b) and 4.5(c). Similar to Scenario 1 and 2 the mobile robot equipped with the conventional DWA algorithm can avoid two obstacles at the intersection area, and safely navigate to the given goal. However, the distance between the robot and the obstacles at the intersection is very close. Whereas in Fig. 4.5(b) and 4.5(c) the robot can proactively avoid the moving obstacles in its vicinity, because it takes the potential collision into account by using the proposed EDWA algorithm.

- **Scenario 4:** In this scenario, to further clarify the effectiveness

of the proposed EDWA algorithm in a dynamic scenario, in which the obstacles P_1 and P_3 are moving toward to each other, while obstacle P_2 is moving toward to the robot. The trajectory of the robot and the obstacles are shown in Fig 4.5(d), 4.5(e) and Fig 4.5(f). Similar to the aforementioned scenarios, the robot equipped with the conventional DWA move very close the obstacles. While the robot has ability to proactively avoid obstacles and safely navigate to the given goal when it is installed the proposed EDWA algorithm.

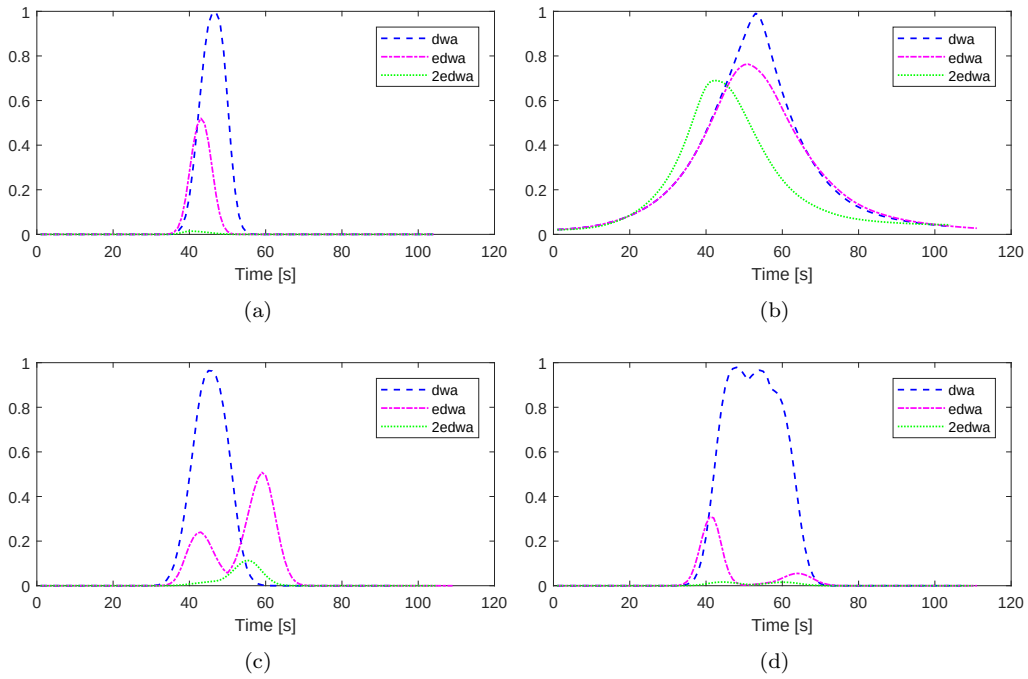


Figure 4.6: The minimum passing distance along the robot's trajectory.

We further investigate the effectiveness of our proposed EDWA algorithm through quantitative analysis. As can be seen in Fig. 4.6, in all four experiment scenarios, the value of the normalized minimum distance is close to 1, it indicates that the mobile robot equipped with the DWA algorithm move close to the obstacle. In contrast, the EDWA enable

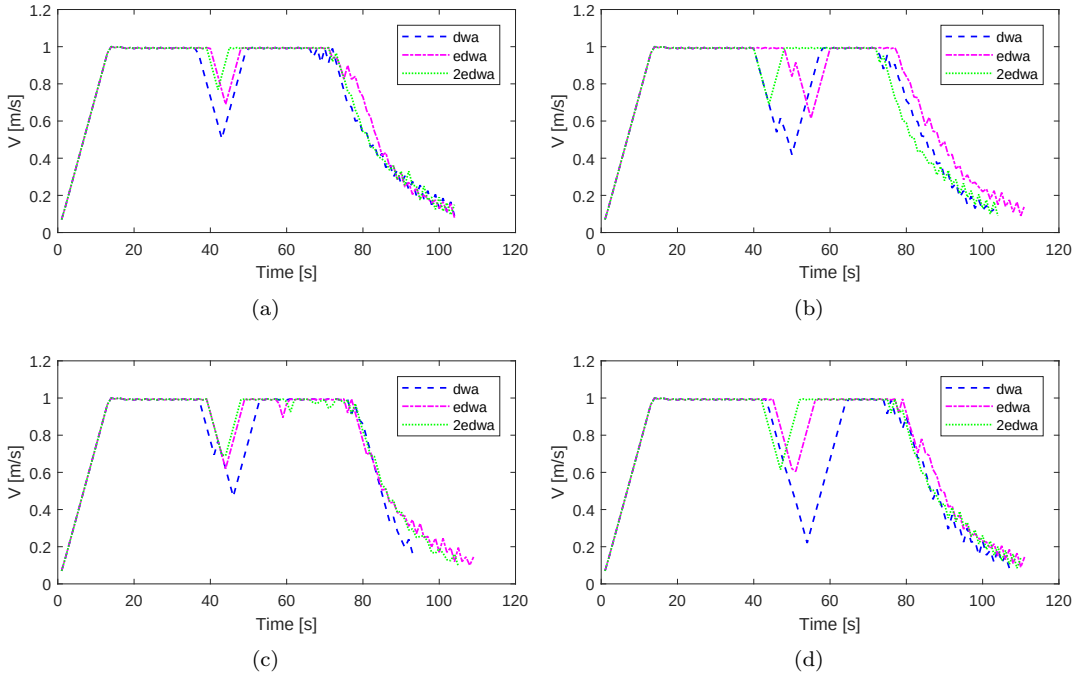
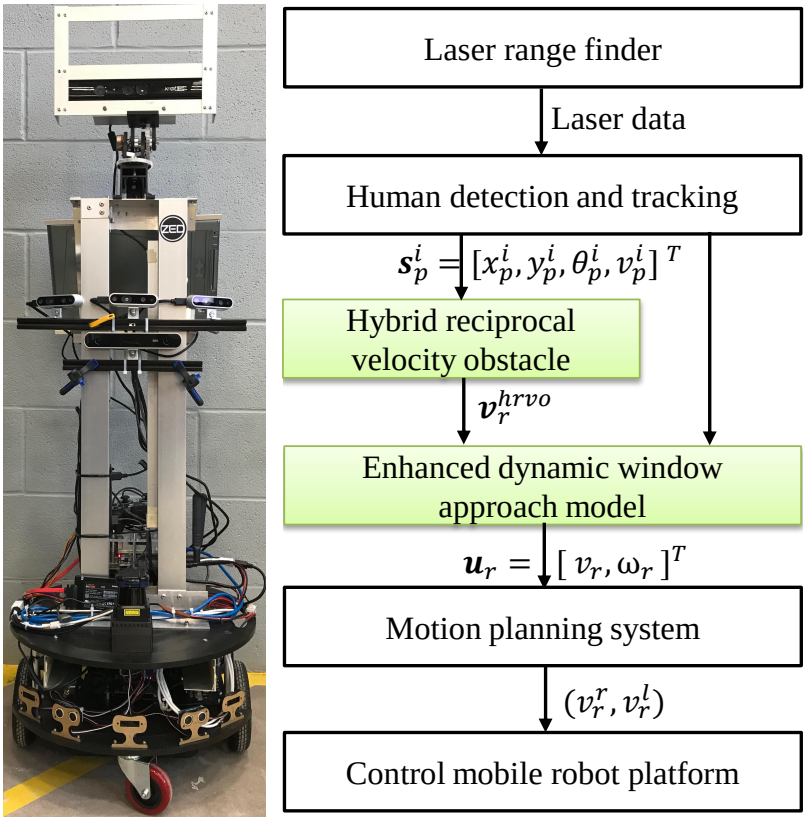


Figure 4.7: The robot's velocity along the trajectory of mobile robot.

the mobile robot to proactively avoid obstacle, because the value of the normalized minimum distance is small and even approaching 0. In addition, Fig. 4.7 and Table 4.2 illustrate that, the mobile robot equipped with our proposed EDWA is capable of smoothly avoiding the obstacles in the vicinity of the mobile robot.

In summary, the simulation results shown in Fig. 4.4, Fig. 4.5, Fig. 4.6, Fig. 4.7, and Table 4.2 illustrate that, our proposed EDWA algorithm is capable of driving the mobile robot to deal with potential collisions with various situations in the surrounding environment of the robot in dynamic environments. Moreover, to validate the effectiveness and feasibility of the proposed algorithm, we continue to conduct experiments on a mobile robot platform in a real-world environment.



(a) (b)

Figure 4.8: (a) The Eddie mobile robot platform equipped with a laser rangefinder and a NVIDIA Xavier Developer Kit; (b) The data flow diagram of the proposed framework.

c. Experimental setup and results

The proposed EDWA algorithm has been installed on the mobile robot platform Fig. 4.8(a). Four experiments in a laboratory-like environment are then conducted to examine whether the robot equipped with the proposed EDWA algorithm could safely and proactively avoid obstacles. In this study, using humans as moving obstacles in all experiments is made. The mobile robot platform is used in this research demonstrated in detail in Section 2.1.1 and Table. 4.3, and the data flow diagram of the proposed framework is illustrated in Fig. 4.8(b). Moreover, all experiments

Table 4.3: Parameters of the Eddie mobile robot platform

Parameters of the laser rangefinder	**Value**
Distance measurement	$\leq$ 30 m
Angular field of view	270^o
Resolution	0.25^o
Parameters of Encoder	
Counts per revolution	144 pulses
Resolution	2.5^o

are executed with the parameter values presented in Table 4.1.

Four experiments are conducted to examine whether the mobile robot was able to safely avoid one moving people or two moving people and two crossing people and three moving people, as shown in Fig. 4.9(a), 4.9(b), 4.9(c), and 4.9(d), respectively. In each experiment, the mobile robot is positioned at (0,0) and is requested to navigate to a given goal positioned at (0,8), while avoids the humans in the surrounding environment of the robot. .

The experimental results of the four experiments are shown in the second row in Fig. 4.9 and the first row shows the snapshot of the scenarios. A video with our experimental results can be found at the hyperlink[2].

Figure 4.9(e), 4.9(f), 4.9(g), and 4.9(h) shows the trajectories of the mobile robot and the humans in its vicinity. As can be seen in these figures, the robot can successfully and proactively avoid the humans and safely navigate to the given goal in all four experiments. Because, the mobile robot equipped with the proposed EDWA algorithm can detect and track the humans in the surrounding environment, and then incorporates the human position and motion into its motion planning system.

Overall, the experimental results shown in Fig. 4.9 illustrates that, the

[2]https://youtu.be/wAfgDIxm0Ak

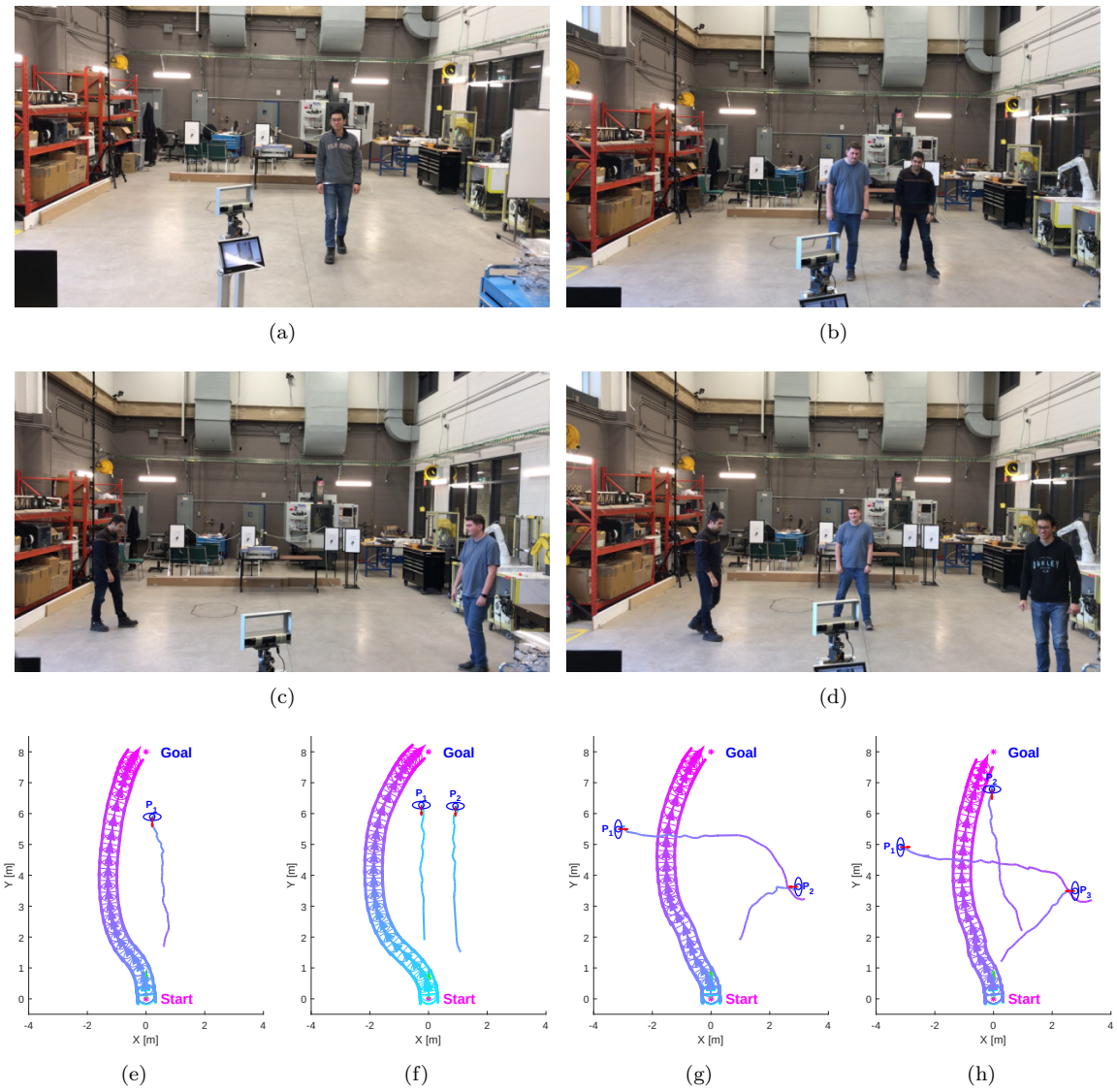


Figure 4.9: The experimental results of four experiments.

proposed EDWA algorithm is feasibility and effectiveness in real-world environments. It enables the mobile robot to proactively avoid dynamic humans in the vicinity of the robot, and safely navigate to the given goal. As a results, the proposed algorithm is able to apply to the real-world environments.

4.1.5. Remarks

In this section, an EDWA algorithm has been presented for autonomous mobile robot navigation systems. By incorporating the velocity vector generated by the HRVO model into the objective function of the conventional DWA technique, the proposed EDWA algorithm could be capable of driving the mobile robots to proactively avoid dynamic obstacles in the robot's vicinity. The simulation results illustrate that the proposed EDWA algorithm outperform the conventional DWA.

However, the proposed EDWA algorithm still suffer from limitations. That is the robot equipped the EDWA algorithm sometimes gets stuck in a local optimal trajectory. Therefore, the robot is impossible to reach the given goal. Moreover, it is so difficult the robot to *transit across obstacles* if they are close to the robot. In order to overcome these weaknesses, in the next section, an effective local planing algorithm is suggested.

4.2. Proposed proactive timed elastic band algorithm

In the previous section, the navigation frameworks could be divided into two categories based on the information used as the input of the motion planning system. However, in another approach the navigation frameworks also can be divided into two categories based on the robot dynamics: (i) none robot dynamics-based approaches and (ii) robot dynamics-based techniques. In the former, the methods do not directly take into account the dynamic constraints of the mobile robots. While in the later, the robot dynamics such as the kynodynamic con-

straints, velocity and acceleration limitations, are directly incorporated into the motion planning system.

Regarding to the none robot dynamics-based techniques, a number of obstacle avoidance and motion control algorithms such as the artificial potential field(APF) [35], vector field histogram (VFH) [36], elastic band (EB) [37], velocity obstacles(VO) [34], [31], and social fore model [28], [23] techniques have been proposed for the autonomous mobile robots. These approaches have been evaluated that the robots are capable of safely avoiding the obstacles in the robot's vicinity, and navigating towards to the given goal. Nevertheless, the systems do not directly take into account the dynamic constrains of the mobile robots. Hence, it might be difficult to directly utilize the output control command to control the mobile robots in the real-world environments, especially for non-holonomic mobile robots.

In order to address that issue, several robot dynamics-based approaches have been proposed in the recent years, such as the dynamic window approach (DWA) [29], randomized kinodynamic planning [38], [39] and an original timed elastic band (TEB) [40] methods. Although, these approaches have been successfully applied in real-world environments, they might not suitable with the dynamic environments. Because the robots equipped with these techniques get often stuck in a local optimal trajectory, as they are unable to transit across obstacles in the dynamic environments. To deal with that problem, recently Rosmann et al. [41, 42] proposed extensions of the TEB technique by using parallel trajectory planning in spatially distinctive topologies. Using this tech-

nique, the mobile robots can switch to the current globally optimal trajectory among the candidate trajectories of distinctive topologies, which are maintained and optimized in parallel. However, these approaches only take into account the current position of the obstacle and do not anticipate obstacle's future trajectory as well as do not incorporate the potential collision with the surrounding obstacle. Therefore, such developed navigation systems lack robustness in diverse situations in the dynamic environments.

In order to overcome the aforementioned shortcomings, in this section, a proactive timed elastic band (PTEB) technique for autonomous mobile robot navigation systems in dynamic social environments is proposed. Because, the TEB technique takes into account the velocity and acceleration limitations, kinodynamic (kinematic and dynamic) and nonholonomic constraints of the mobile robots, and the safety distance of the obstacles and their geometric. In addition, the technique operates in real-time and thereby directly generates commands for the underlying robot motion controller [68]. While the hybrid reciprocal velocity obstacle [31] (HRVO) model utilizes the obstacle's states including current position, orientation and velocity, to compute the robot's future trajectory in order to avoid collisions. Therefore, the robot enables to predict the potential collision with the surrounding obstacles.

The main idea of the proposed technique is to combine the advantages of the TEB technique and the HRVO model. In more detail, this algorithm incorporates the orientation of the velocity vector generated by the HRVO model into the objective function of the TEB algorithm.

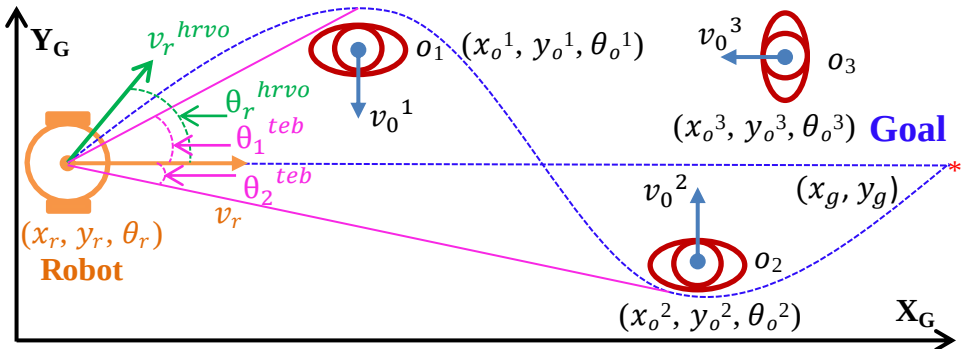


Figure 4.10: The example scenario of the dynamic social environments including a mobile robot and three dynamic obstacles. The robot is requested to navigate to the given goal while avoiding two crossing obstacles o_1 and o_2, and a moving forward obstacle o_3. The curved dashed line is the intended optimal trajectory of the mobile robot.

By incorporating the potential collision between the robots and the obstacles into the TEB technique, the mobile robots equipped with the proposed PTEB algorithm can proactively avoid obstacles and safely navigate towards the given goal.

The remainder of the section is structured as follows. A typical scenario of the problem is presented in Subsection 4.2.1. Subsection 4.2.2 suggests PTEB algorithm to solve it. Subsection 4.2.3 describes the navigation system of the robot that using the PTEB algorithm. The experimental results in two simulation environments are described and discussed in Subsection 4.2.4 and Subsection 4.2.4. Finally, conclusion is drawn in Subsection 4.2.5.

4.2.1. Problem description

The scenario in Fig 4.10 is utilized to describe and prove the efficiency of the proposed planning algorithm. In this scenario, the mobile robot is requested to navigate from the starting pose $\mathbf{s}_1 = [x_r^1, y_r^1, \theta_r^1]^T$ to the goal pose $\mathbf{s}_N = [x_r^N, y_r^N, \theta_r^N]^T$, while avoiding three obstacles. The figure

shows an example of a crossing scenario, that the TEB technique might generate an optimal trajectory in front of the obstacles, which may not be feasible. Therefore, to deal with the problem, it's necessary to propose the improved algorithm.

4.2.2. Construction of the PTEB algorithm

The main idea is to exploit the advantages of both conventional TEB technique presented in Section 2.3.3 and HRVO model described in Section 2.3.2, a proactive timed elastic band (PTEB) algorithm is suggested. In other words, the developed PTEB algorithm takes into account both the dynamic constraints of the mobile robot and its potential collision with the surrounding obstacles. To accomplish this, in the objective function in (2.40), one more factor using the orientation of the velocity vector generated by the HRVO model is added. More specifically, the orientation θ_r^{hrvo} of the velocity vector $\mathbf{v}_r^{hrvo} = [\upsilon_x, \upsilon_y]^T$ generated by the HRVO model in (2.29) is used to compute the difference between it and the angles θ_p^{teb} of the M locally optimal trajectories, with p = 1, 2, ..., M.

$$\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x) \tag{4.7}$$

$$\theta_p^{teb} = atan2(y_p^{teb} - y_r, x_p^{teb} - x_r) \tag{4.8}$$

$$\Delta\theta_p^{teb} = |\theta_r^{hrvo} - \theta_p^{teb}| \tag{4.9}$$

where, (x_r, y_r) is the current position of the mobile robot, (x_p^{teb}, y_p^{teb}) is the coordinates of the node ζ_p, which is added beside the obstacles. It is noted that, the value of $\Delta\theta_p^{teb}$ ranges from 0 to π, and the numbers of $\Delta\theta_p^{teb}$ are equal to the numbers of individuals TEB. Finally, the new

objective function of the PTEB is obtained as follows:

$$\hat{\mathbf{V}}_c(\mathbf{B}_p^*) = V_c(\mathbf{B}_p^*) + \delta_{hrvo}\Delta\theta_p^{teb} \tag{4.10}$$

where, δ_{hrvo} is a predefined value. Using the new objective function (4.10), the result of solving (2.39), give the optimal trajectory, as presented in Fig. 4.10. Figure 4.10 shows an example of the proactive TEB algorithm. In this case, the curved dashed line is the intended optimal trajectory of the mobile robot. In other words, the mobile robot equipped with the proactive TEB algorithm is capable of avoiding potential collision with the three obstacles. Because it takes into account the orientation of the velocity vector generated by the HRVO model. It is noted that, the difference between the θ_r^{hrvo} and the θ_1^{teb} is smallest.

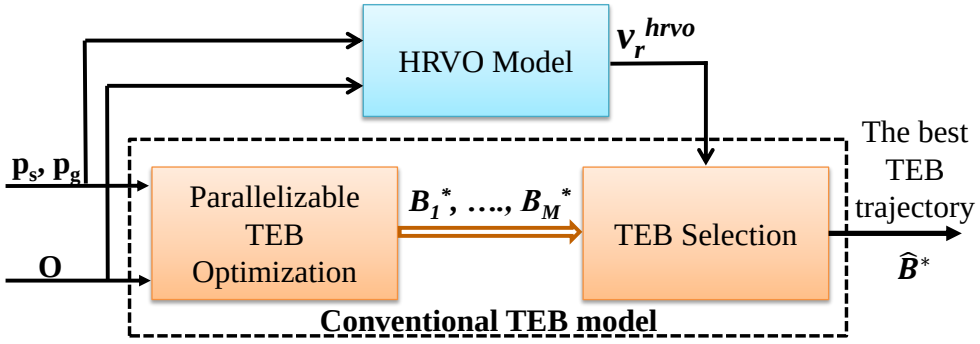


Figure 4.11: The flowchart of the proposed proactive TEB algorithm.

Figure. 4.11 shows the flowchart, and Algorithm 6 presents in detail the proposed PTEB algorithm. Algorithm 6 is develop based on the conventional TEB algorithm presented in Algorithm 2 in Section 2.3.3. The input of the proposed algorithm is robot state $\mathbf{s}_r$, start pose $\mathbf{p}_s$, goal pose $\mathbf{p}_g$ and a set of obstacles $\mathbf{O}$. The output is the optimal robot trajectory $\hat{\mathbf{B}}^*$, which can be used to directly extract the velocity control command $\mathbf{u} = [\upsilon_r, \omega_r]^T$ of the mobile robot.

Firstly, we generate M locally optimal trajectories $\mathbf{B}_p^*$, with p = 1, 2, ..., M based on the conventional TEB model (Lines 2-11 of the Algorithm 6). Secondly, we compute $\Delta\theta^{teb}$ (Lines 12-15 of the Algorithm 6) by combining the orientation of θ_r^{hrvo} and the angles θ_p^{teb} of the candidate optimal trajectories in the previous step. Thirdly, a new component $\Delta\theta^{teb}$ is added into the conventional objective function. As a result, we obtain a new objective function $\hat{\mathbf{V}}_c(\mathbf{B}_p^*)$ (Lines 16-17 of the Algorithm 6). Finally, the optimal trajectory $\hat{\mathbf{B}}^*$ and the optimal control command of the mobile robot are generated in Lines 18 - 20 of Algorithm 6, respectively.

Algorithm 6: Proposed PTEB algorithm

input : robot state $\mathbf{s}_r$, start pose $\mathbf{p}_s$, goal pose $\mathbf{p}_g$, set of obstacles $\mathbf{O}$
output: Control command $\mathbf{u}_r$

1 **begin**
2 $\mathbf{G} \leftarrow$ createGraph($\mathbf{s}_r$, $\mathbf{p}_s$, $\mathbf{p}_g$, $\mathbf{O}$);
3 $\mathbf{D} \leftarrow$ depthFirstSearch($\mathbf{G}$);
4 $\mathbf{H} \leftarrow$ computeH-Signature($\mathbf{D}$, $\mathbf{G}$);
5 $\mathbf{R} \leftarrow$ removeRedundantPath($\mathbf{D}$, $\mathbf{H}$, $\mathbf{G}$);
6 $\mathbf{T} \leftarrow$ initializeTrajectories($\mathbf{R}$, $\mathbf{G}$);
7 **for** each trajectory $\mathbf{B}_p \in \mathbf{T}$ **do**
8 $\mathbf{V} \leftarrow$ objectiveFunction(); ▷ using (2.33)
9 $\mathbf{B}_p^* \leftarrow$ Optimizer($\mathbf{B}_p, \mathbf{O}, \mathbf{V}$); ▷ Solve (2.34)
10 $\mathbf{B}^* \leftarrow$ storeLocalOptimalTrajectory($\mathbf{B}_p^*$);
11 **end for**
12 $\mathbf{v}_r^{hrvo} = [\upsilon_y, \upsilon_x]^T \leftarrow$ Run HRVO($\mathbf{s}_r$, $\mathbf{O}$)
13 $\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x)$
14 $\theta_p^{teb} = atan2(y_p^{teb} - y_r, x_p^{teb} - y_r)$
15 $\Delta\theta^{teb} = min(|\theta_r^{hrvo} - \theta_p^{teb}|)$ with p= 1, 2, ..., M.
16 $\mathbf{V}_c \leftarrow$ newObjectiveFunction(); ▷ using (2.40)
17 $\hat{\mathbf{V}}_c(\mathbf{B}_p^*) = V_c(\mathbf{B}_p^*) + \delta_{hrvo}\Delta\theta^{teb}$ using (4.10)
18 $\hat{\mathbf{B}}^* \leftarrow$ Call Optimizer($\mathbf{B}^*, \mathbf{O}, \hat{\mathbf{V}}_c$) ▷ Solve (2.39)
19 $\mathbf{u}_r \leftarrow$ According to (2.35), (2.36) and $\hat{\mathbf{B}}^*$
20 Return $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$

4.2.3. The PTEB algorithm-based navigation framework

Similar to Section 4.1.3, in order to conduct experiments in simulation and real-world environments, the proposed PTEB algorithm is also integrated into the conventional navigation scheme, as presented in Fig. 4.12. In the extended part, the multi-objects detection and tracking block is used to detect and track objects in the vicinity of the robot. Then the HRVO model utilizes the object state including position, orientation and velocity, to model the potential collision of the robot with the surrounding objects. This information and the object state are then used as the inputs of the proposed PTEB algorithm. Once the optimal trajectory is generated by the proposed PTEB algorithm, the motion control command $\mathbf{u}_r=[\upsilon_r,\ \omega_r]^T$ is extracted and used to drive the mobile robot to proactively avoid the obstacles in the robot's vicinity and approach a given goal. In this study, a two-wheel differential drive mobile robot platform as well as the robot's motion model similar to presented in the previous section 4.1.4 are utilized .

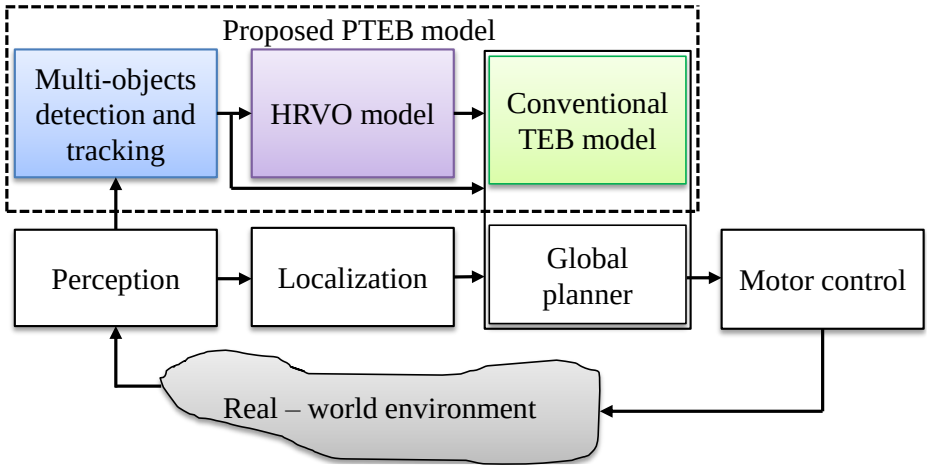
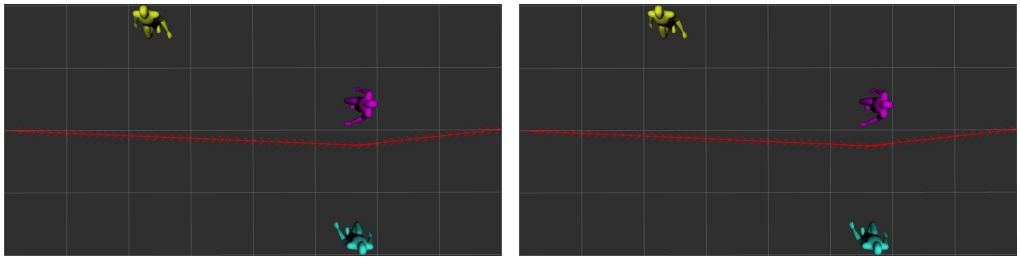
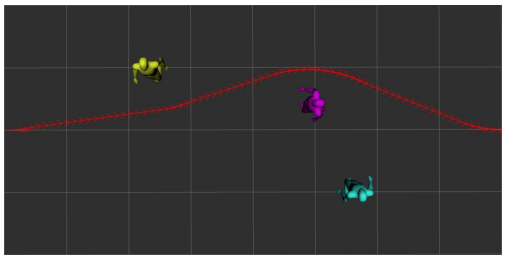


Figure 4.12: The navigation framework based on the PTEB algorithm.

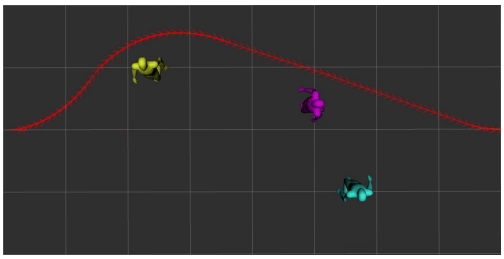


(a) T1 [s]

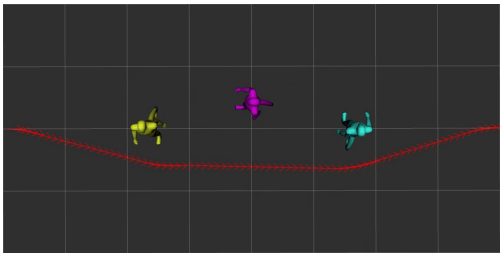
(b) T1 [s]



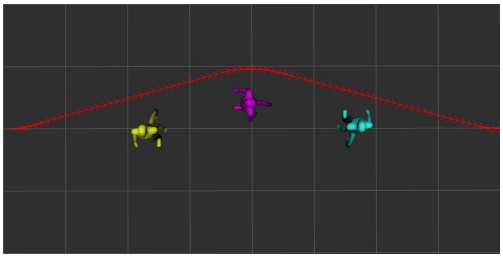
(c) T2 [s]



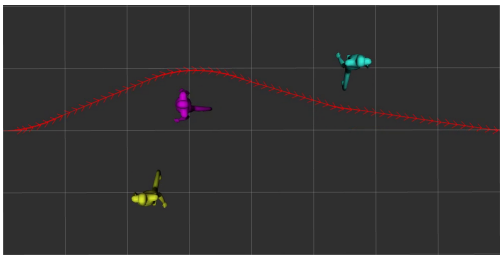
(d) T2 [s]



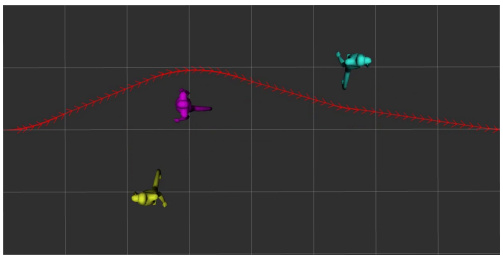
(e) T3 [s]



(f) T3 [s]



(g) T4 [s]



(h) T4 [s]

Figure 4.13: Four snapshots at four timestamps of the two experiments in the simulation environment.

4.2.4. Simulation results

a. Simulation experiment in RViz Environment

In this study, firstly examining the proposed PTEB algorithm in a simple simulation environment, and visualizing the results in RViz en-

Table 4.4: Parameters set in experiments

Parameters	Value	Parameters	Value
v^r_{max}	1 [m/s]	r_r, r_o	$0.3[m]$
ω^r_{max}	$2.5[m/s^2]$	δ_{hrvo}	0.5
δ_v	2.0	δ_h	1000
δ_o	50	δ_α	1.0

vironment[3] with parameters set up in Table 4.4. The mobile robot is requested to navigate from left to right, while avoiding dynamic people. The simulation results are shown in Fig. 4.13. In this figure, the left column shows the results of the conventional TEB algorithm, whereas the right column presents the results of the proposed PTEB algorithm. The green curves illustrate the distinctive candidate trajectories, while the green curve with red arrows depicts the optimal selected trajectory of the mobile robot.

At the time stamps T1 and T4, the simulation results of the conventional TEB algorithm and the proposed PTEB algorithm are similar. Because, at the time stamp T1 the two crossing humans are approaching the straight line between the starting position and the goal position but they are far from straight line, as shown in Figs. 4.13(a) and 4.13(b), or at the time stamp T4 the two crossing humans are close to the straight line but they are moving away the straight line, as shown in Figs. 4.13(g) and 4.13(h).

At the time stamps T2 and T3, the globally obtimal trajectory is generated in font of two crossing people, as shown in Figs. 4.13(c) and 4.13(e), in these cases, the mobile robot can safely avoid people but its behavior might not be smooth. In contrast, the globally optimal

[3]http://wiki.ros.org/rviz

trajectory is generated behind the left crossing person, it illustrates that, the robot is able to proactively avoid people, as shown in Figs. 4.13(d) and 4.13(f). Because, the proposed PTEB algorithm takes into account the potential collision of the robot with the surrounding humans.

Secondly, in order to narrow the gap between simulation and real-world experiments, in the next section we have created a hallway-like scenario with walls, objects, humans and goals based on the Stage robot simulator [44].

b. Simulation experiment in Stage environment

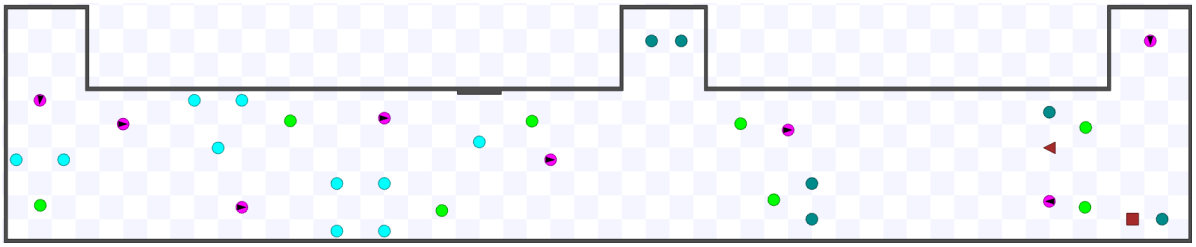


Figure 4.14: A hallway-like scenario with walls, objects, humans, and goals.

The simulation is implemented in Stage environment and parameters of the system as well as of the objective function (4.10) are set up in Table 4.4, as depicted in Fig. 4.14. The figure shows the mobile robots (magenta dots), 10 stationary people (cyan dots), 6 moving people (dark blue dots), and two moving object (brown triangle and square), and 8 goals (green dots) are distributed in the scenario. The robot is assigned a task to navigate to approach goals while avoiding humans and objects. The moving humans and objects are controled using social force model proposed by Helbing et al. [28] and the available software platform[4].

We have installed the proposed PTEB algorithm on the mobile robot

[4]http://pedsim.silmaril.org

to validate its effectiveness. We conducted two experiments in the simulated Stage environment to examine whether our mobile robot equipped with the proposed PTEB algorithm could safely and proactively avoid dynamic obstacles while navigating safely in the environments. In first experiment, the mobile robot is requested to navigate from each starting position (magenta dot) to a corresponding goal (green dot), while avoiding the static and moving objects in the hallway. In the second experiment, a mobile robot is requested to navigate to approach each goal in the scenario while avoiding dynamic objects during its navigation.

In addition, to quantitatively validate the proposed PTEB algorithm, we adopted the collision index (CI) proposed by Truong et al. [24]. Specifically, the CI value is applied to measure the physical safety of the robot and each individual obstacle, and calculated using Eq. 4.11 [24].

$$CI = \max_{i=1:N} \exp\Big(- \Big(\big(\frac{x_r - x_o^i}{\sqrt{2}\sigma_o^x}\big)^2 + \big(\frac{y_r - y_o^i}{\sqrt{2}\sigma_o^y}\big)^2 \Big) \Big) \tag{4.11}$$

Where, (x_o^i, y_o^i) is the position of obstacle o_i, (x_r, y_r) denotes the position of robot, σ_o^x and σ_o^y are standard deviations of the basic the obstacle space, which are set equal to 0.45 in [24], and N is number of the obstacles in the vicinity of the robot. The range of the CI value is from 0.0 to 1.0, where the higher the value of CI, the closer the relative distance between the robot and the obstacle. The robot crashed into the surrounding object if the CI value is greater than the threshold 0.54, presented in [24].

The experimental results in the Stage environment are illustrated in

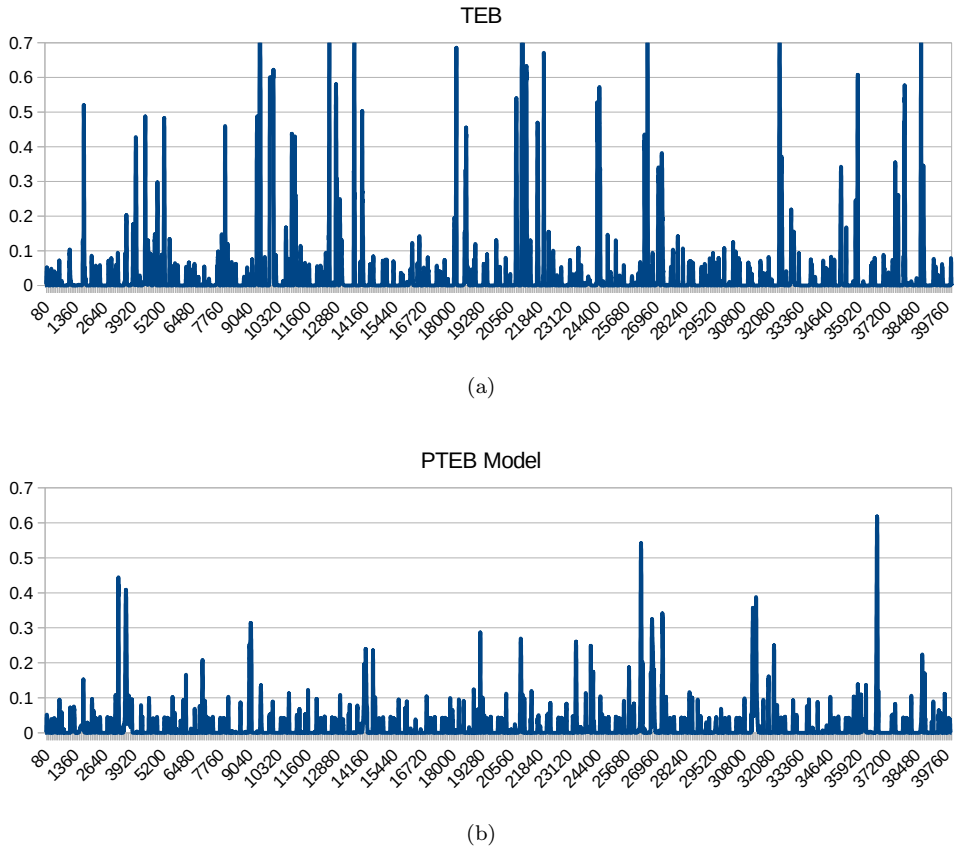


Figure 4.15: The simulation results of the two experiments. The first row shows the collision index of the conventional TEB algorithm. Whereas, the second row illustrates the collision index of the PTEB technique.

Fig. 4.15. As can be seen in Fig. 4.15(b), the CI value is maintained as lower than 0.54 along the robot trajectory. It indicates that, the mobile robot equipped with the proposed PTEB algorithm is able to proactively avoid static and dynamic object in the vicinity of the robot, and safely navigate to the given goal. In contrast, the mobile robot installed the conventional TEB technique move closely to the obstacles. The robot even crashed into the obstacles several time during the robot navigation, as shown in Fig. 4.15(a).

In summary, the experimental results clearly demonstrated that the mobile robot equipped with the proposed PTEB algorithm significantly

reduces collision than that with the conventional TEB algorithm.

4.2.5. Remarks and discussion

A *proactive* timed elastic band algorithm (PTEB) have been presented in this section. The main idea of the proposed PTEB algorithm is to combine the advantages of the TEB algorithm and the HRVO model by incorporating the velocity vector generated by the HRVO model into the objective function of the TEB algorithm. As a result, the proactive optimal trajectory was achieved from the output of the proposed PTEB algorithm which is used to control the mobile robots. The effectiveness of the proposed algorithm were validated through a series of experiments in simulation environments. The simulation results reported in this section demonstrated that, the proposed proactive motion planning model with proposed PTEB algorithm is capable of driving the mobile robots to proactively avoid static and dynamic obstacles, and providing smooth and safe navigation for the robots.

Although the proposed PTEB algorithm has been achieved consider successes, it lacks of robustness in various environments, because it only incorporates the velocity obstacles-based potential collision. Therefore, in order to deal with this issue, in the next section, an extended timed elastic band (ETEB) algorithm, which takes into account the *future states of the surrounding obstacles*, will be proposed for autonomous mobile robots.

4.3. Proposed extended timed elastic band algorithm

Similar as described in former sections (Section 4.1 and Section 4.2), here we can split into existing navigation frameworks into two groups based on the techniques used to develop the systems: (i) reaction-based methods and (ii) trajectory-based approaches. In the first group, the reactive local planners, e.g., the artificial potential field [35], vector field histogram [36], the dynamic window approach [29], velocity obstacles [34], [31], and social fore model [28], [23], are used to develop the systems. Although these approaches have been evaluated such that the robots are capable of safely avoiding the obstacles in the robot's vicinity, they are often short-sighted in time. To solve that problem, trajectory-based methods, such as elastic band [37], randomized kinodynamic planning [38], [39], timed elastic band (TEB) [40], Rosmann et al. [41], [42], compute plans on longer timescale to produce smoother robot's trajectory. However, these approaches only take into account the position of the obstacle and do not incorporate the potential collisions between the robots and the surrounding obstacles

Furthermore, the conventional navigation frameworks also can be classified into two categories according to the information used as input of the motion planning systems in an other approach: (i) current states-based approaches and (ii) future states-based techniques. In the former, the methods only take into account the current states position of the surrounding obstacles. While in the later, the current and future states of the obstacles are incorporated into the motion planning systems. Sev-

eral navigation systems have been developed for the mobile robots using the current states of the surrounding obstacle, such as the dynamic window approach [29], randomized kinodynamic planning [38], [39] and timed elastic band (TEB) [40] methods. Although, these approaches have been successfully applied in real-world environments, they might not suitable with the dynamic environments, because these approaches only take into account the current position of the obstacle and do not incorporate the future states of the surrounding obstacles. Therefore, such developed navigation systems lack robustness in diverse situations in the dynamic environments. To address that issue, a few mobile robot navigation systems take into account the future states of the obstalces [31], [23], [69] and [70]. However, these systems do not directly take into account the motion dynamics of the mobile robots such as the kynodynamic constraints, velocity and acceleration limitations. Hence, it might be difficult to directly utilize the output control command to control the mobile robots in the real-world environments.

In order to overcome the aforementioned shortcomings, in this section, we propose an extended timed elastic band (ETEB) technique for mobile robot navigation systems using motion prediction algorithm. Because, the TEB technique takes into account the velocity and acceleration limitations, kinodynamic and nonholonomic constraints of the mobile robot, and the safety distance of the obstacles and their geometric. And the motion prediction model utilizes the obstacle's states including position, orientation and velocity, to predict future position of the surrounding obstacles. By incorporating this potential collision between the robots

and the obstacles into the TEB technique the mobile robots equipped with our proposed ETEB algorithm can proactively avoid obstacles and safely navigate to the given goal.

In this section, we present a typical scenario of this problem in Subsection 4.3.1. Then, the proposed ETEB algorithm for solving the problem is given in Subsection 4.3.2. The experimental results are demonstrated in Subsection 4.3.3. Finally, conclusion is drawn in Subsection 4.3.4.

4.3.1. Problem description

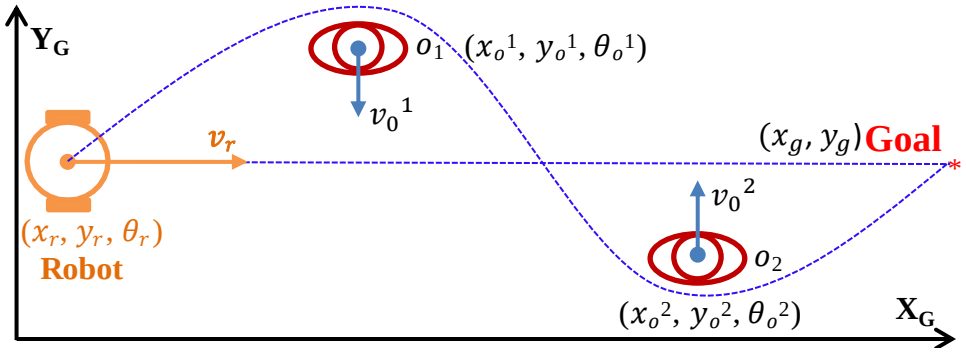


Figure 4.16: The example scenario including a mobile robot and two dynamic obstacles. The curved dashed line is the intended optimal trajectory of the mobile robot.

To demonstrate the efficiency of the proposed algorithm, we consider a dynamic social environment with the presence of an autonomous mobile robot and O obstacles in the robot's vicinity, as shown in Fig 4.16. The robot is requested to navigate to a goal while safely avoid the obstacles during its navigation. In this scenario, two obstacles from both sides facing each other and move across the robot.

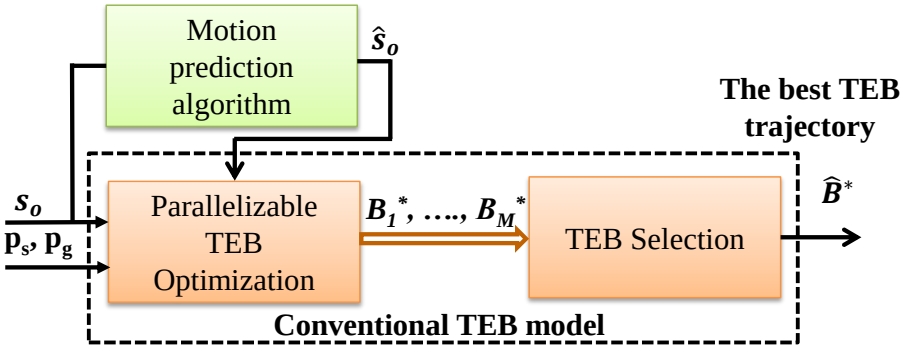


Figure 4.17: The proposed extended TEB algorithm

4.3.2. Construction of the ETEB algorithm

The mobile robot equipped the PTEB algorithm is able to transit across obstacles, and safely proactively navigate in dynamic environments. However, it only takes into account current states and potential collision of the surrounding obstacles. Therefore, it lacks of robustness in dynamic environments and might not be feasible in various real-world situations.

In order to address that issue, in this section, an extended timed elastic band (ETEB) algorithm is proposed, as presented in Fig 4.17 and Algorithm 7. To accomplish that, the future states of the surrounding obstacles is firstly predicted by using the extended Kalman filter algorithm [52] and the data association technique [71]. The output of the motion prediction model is the future states of the obstacles $\hat{\mathbf{s}}_o$, as shown in Fig 4.17. Then the proposed algorithm incorporates both the current states $\mathbf{s}_o^i$ and the future states $\hat{\mathbf{s}}_o^i$ of the obstacles $\mathbf{o}_i$ into the conventional TEB algorithm, as shown in Fig. 4.17. In other words, in stead of using only current states of the surrounding obstacles as TEB algorithm or the current states and potential collision as PTEB algorithm, the proposed

ETEB algorithm takes *both the current and future states* into account.

Assuming that the number of obstacles in the robot's vicinity at time k is R ($\mathbf{O}_i$, i=1, 2, ..., R). The future state of the obstacles predicted by the robot is $\hat{s}_o$ ($\hat{\mathbf{O}}_i$, i=1, 2, ..., R). Therefore, the *total number of obstacles* used as input of the the conventional TEB algorithm becomes 2R. As a result, the objective function in (2.33) is added a new part, as presented in (4.12).

$$\hat{\mathbf{V}}(\mathbf{B}_p) = \mathbf{V}(\mathbf{B}_p) + \delta_o \|min\{\mathbf{0}, \hat{\mathbf{O}}\}\|_2^2 \tag{4.12}$$

Algorithm 7: Proposed ETEB algorithm

input : robot state $\mathbf{s}_r$, start pose $\mathbf{p}_s$, goal pose $\mathbf{p}_g$, set of obstacles $\mathbf{O}$
output: Control command $\mathbf{u}_r$

begin
- $\mathbf{G} \leftarrow$ createGraph($\mathbf{s}_r$, $\mathbf{p}_s$, $\mathbf{p}_g$, $\mathbf{O}$);
- $\mathbf{D} \leftarrow$ depthFirstSearch($\mathbf{G}$);
- $\mathbf{H} \leftarrow$ computeH-Signature($\mathbf{D}$, $\mathbf{G}$);
- $\mathbf{R} \leftarrow$ removeRedundantPath($\mathbf{D}$, $\mathbf{H}$, $\mathbf{G}$);
- $\mathbf{T} \leftarrow$ initializeTrajectories($\mathbf{R}$, $\mathbf{G}$);
- $\hat{\mathbf{O}}^k \leftarrow$ Motion prediction of obstacles;
- **for** each trajectory $\mathbf{B}_p \in \mathbf{T}$ **do**
  - $\mathbf{V} \leftarrow$ objectiveFunction(); ▷ using (2.33)
  - $\hat{\mathbf{V}}(\mathbf{B}_p) = \mathbf{V}(\mathbf{B}_p) + \delta_o \|min\{\mathbf{0}, \hat{\mathbf{O}}^k\}\|_2^2$ using (4.12);
  - $\mathbf{B}_p^* \leftarrow$ Optimizer($\mathbf{B}_p, \mathbf{O}, \hat{\mathbf{V}}$); ▷ Solve (2.34)
  - $\mathbf{B}^* \leftarrow$ storeLocalOptimalTrajectory($\mathbf{B}_p^*$);
- **end for**;
- $\mathbf{V}_c \leftarrow$ newObjectiveFunction(); ▷ using (2.40);
- $\hat{\mathbf{B}}^* \leftarrow$ Call Optimizer($\mathbf{B}^*, \mathbf{O}, \mathbf{V}_c$) ▷ Solve (2.39;)
- $\mathbf{u}_r \leftarrow$ According to (2.35), (2.36) and $\hat{\mathbf{B}}^*$
- Return $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$

The proposed ETEB algorithm is presented detail in Algorithm 7. The input of Algorithm 7 consists of the robot state $\mathbf{s}_r$, start pose $\mathbf{p}_s$, goal pose $\mathbf{p}_g$ and a set of obstacles $\mathbf{O}$ including (x_o, y_o, v_o); and the output of Algorithm 7 is the control command of the mobile robot $\mathbf{u}^*$. In the first

step, we generate M locally optimal trajectories $\mathbf{B}_p^*$) with p=1, 2,..., M by using the TEB optimization in parallel with respect to the objective function(2.33) (Lines 2-6 of the Algorithm 7). In the second step, the future sates of the surrounding obstacles, which is adopted from the motion prediction model, are incorporated into the conventional TEB model (Line 7 of the Algorithm 7). In the third step, the future states of obstacles are added into the conventional objective function (Lines 10 of Algorithm 7). As a result, we obtain a new objective function, as presented in (4.12). In the fourth step, the optimal robot trajectory $\hat{\mathbf{B}}^*$ is selected from the set of alternatives $\mathbf{B}_p^*$ by solving (2.39) (Line 14-15 of the Algorithm 7). Finally, the control command $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$ of the mobile robot is extracted directly from the selected trajectory $\hat{\mathbf{B}}^*$ (Line 16) . This control command is then utilized (Line 17)to control the mobile robot to safely avoid the obstacles in the robot vicinity.

4.3.3. Simulation results

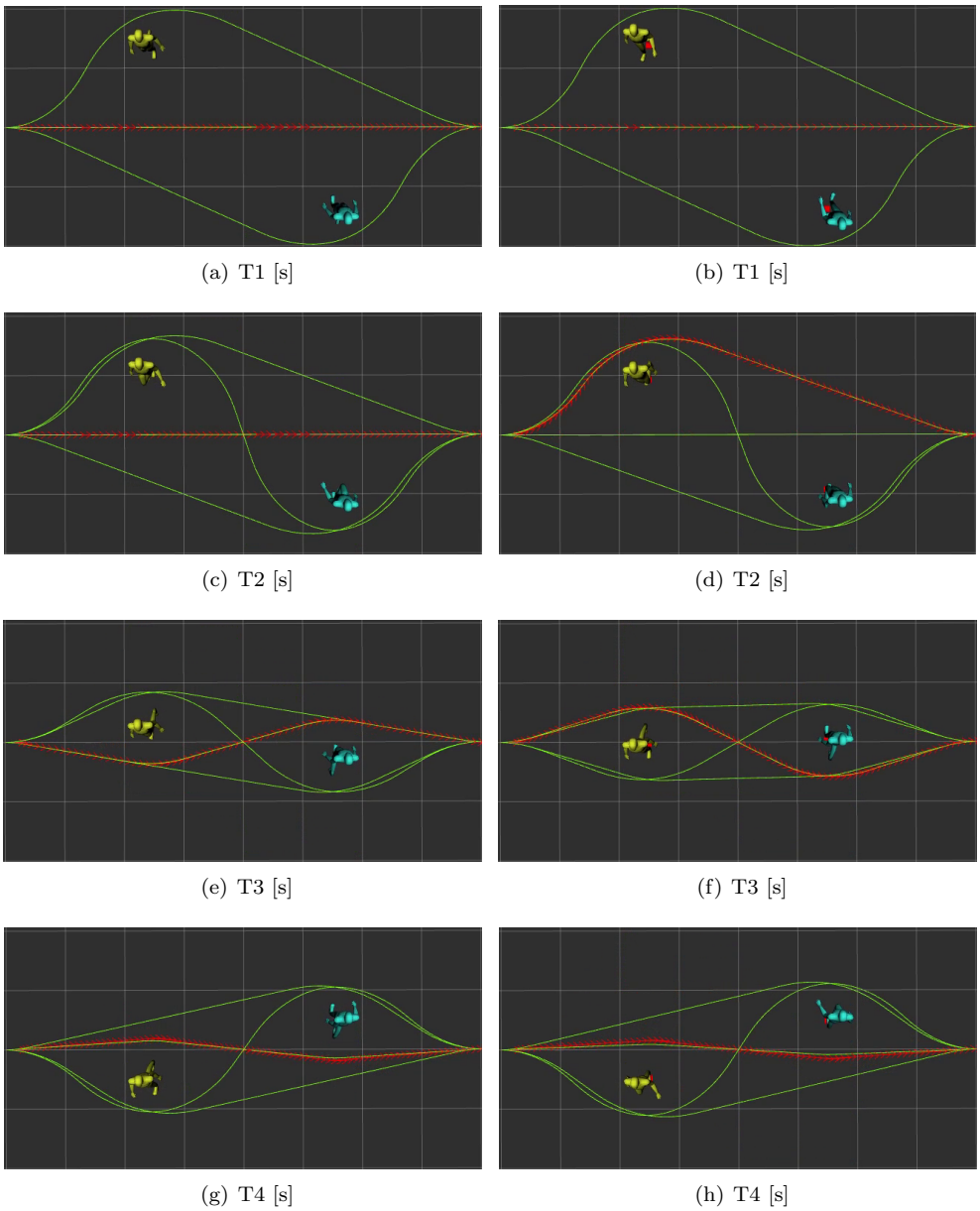
The proposed ETEB algorithm is firstly examined in RViz environment[5] and Stage simulator[6] to verify the effectiveness. Parameters set in simulation are shown in the Table 4.4.

a. Simulation experiment in RViz Environment

The mobile robot is requested to navigate from left to right, while avoiding two crossing people (can be seen Fig. 4.16). The simulation results are shown in Fig. 4.18. The first column shows the results of the conventional TEB algorithm, whereas the second column presents the

[5]http://wiki.ros.org/rviz
[6]http://pedsim.silmaril.org

results of the proposed ETEB algorithm; the green curves illustrate the distinctive candidate trajectories, while the green curve with red arrows depicts the optimal selected trajectory of the mobile robot.



(a) T1 [s] (b) T1 [s]

(c) T2 [s] (d) T2 [s]

(e) T3 [s] (f) T3 [s]

(g) T4 [s] (h) T4 [s]

Figure 4.18: Four snapshots at four timestamps of the two experiments in the simulation environment.

At the time stamps T1 and T4, the simulation results of the conventional TEB algorithm and the proposed ETEB algorithm are similar.

Because, at the time stamp T1 the humans are approaching the straight line between the starting position and the goal position but they are far from straight line, as shown in Figs. 4.18(a) and 4.18(b), or at the time stamp T4 the humans are close to the straight line but they are moving away the straight line, as shown in Figs. 4.18(g) and 4.18(h).

At the time stamps T2 and T3, the optimal trajectory is generated in font of two people, as shown in Figs. 4.18(c) and 4.18(e). In these cases, the mobile robot can safely avoid people but its behavior might not smoothly. In contrast, the optimal trajectory is generated behind the left person, it illustrates that, the robot is able to proactively avoid people, as shown in Figs. 4.18(d) and 4.18(f). Because, the proposed ETEB algorithm takes into account the potential collision of the robot with the surrounding humans.

b. Simulation experiment in Stage Environment

We also validate the effectiveness of the proposed ETEB algorithm in Stage simulator. To accomplish that, we incorporate the proposed ETEB algorithm into the conventional navigation scheme, and develop the ETEB algorithm-based mobile robot navigation system, as presented in Fig. 4.19. Similar to Section 4.2.4, the results of ETEB algorithm are compared with the PTEB algorithm using the collision index proposed by Truong et al. [24]. We then installed the developed navigation system on the mobile robot platform built in Stage enviroment, and conducted experiments in the scenario shown in Fig. 4.14. The parameters set in experiments is presented detail in section 4.2.4. The experimental results

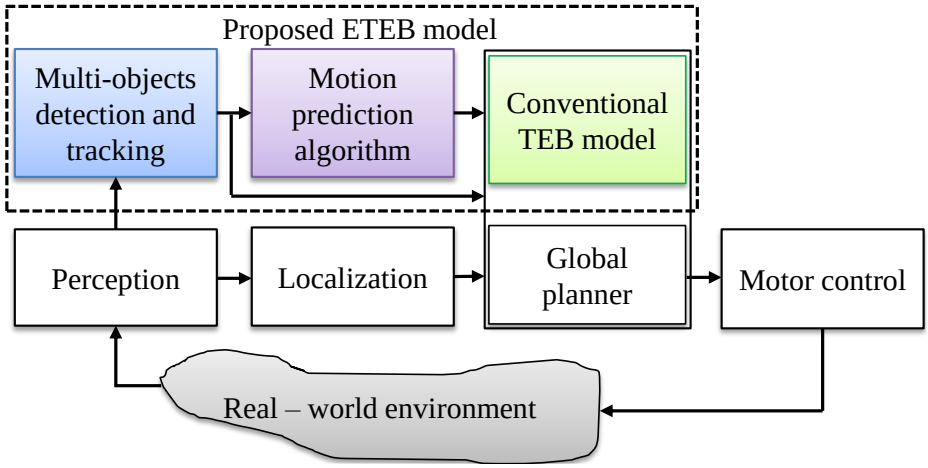
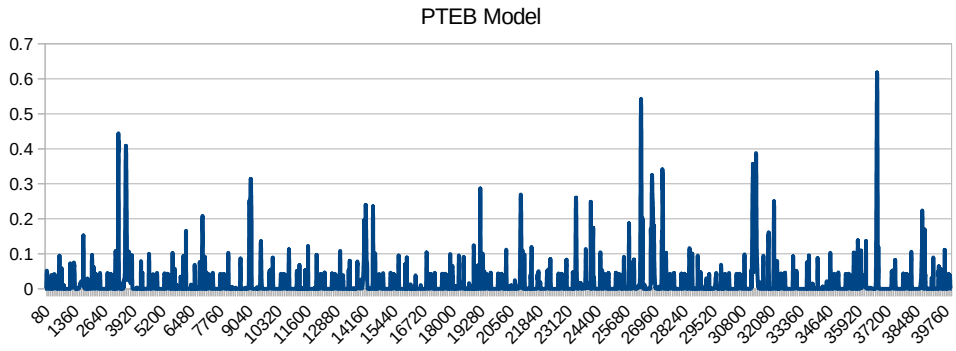
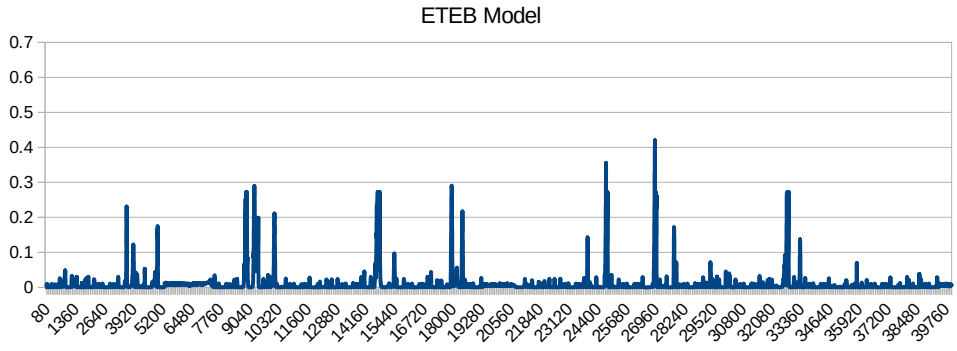


Figure 4.19: The navigation framework based on the ETEB algorithm



(a)



(b)

Figure 4.20: The simulation results of the two experiments. The first row shows the collision index of the conventional PTEB algorithm. Whereas, the second row illustrates the collision index of the ETEB technique.

in the Stage environment are illustrated in Fig. 4.20. As can be seen in Fig. 4.20(b), the CI value is always maintained as lower than 0.54 along

the robot trajectory. It indicates that the mobile robot equipped with the proposed ETEB algorithm is able to proactively avoid static and dynamic objects in the vicinity of the robot, and more safely navigate to the given goal. Meanwhile, the mobile robot equipped PTEB algorithm moves more closely to the obstacles. The robot even crashed into the obstacles several times when the CI value is larger than the threshold 0.54, as shown in Fig. 4.20(a).

4.3.4. Remarks

We have presented an extended timed elastic band algorithm (ETEB) for navigation system of the autonomous mobile robots in dynamic environments. The main idea of the proposed technique is to incorporate both the *current and future sates* of the surrounding obstacles into the conventional TEB algorithm. The simulation results prove that the proposed ETEB algorithm is more effective than PTEB algorithm in terms of proactively avoiding potential collisions in dynamic environment.

4.4. Proposed integrated navigation system

In order to demonstrate the feasibility and usefulness of the proposed algorithms in the thesis, this section presents a completed navigation system for the autonomous mobile robot in a dynamic environment. It has been known that, a completed navigation system is combined from four fundamental models including perception, localization, motion planning, and motor control model. Therefore, this work gathers the proposed efficient algorithms presented in the previous parts of the thesis into a navigation system. The main contribution of this section is to demon-

strate the integration of using EKF-based localization algorithm with an extended timed elastic band (ETEB) algorithm for a differential-wheel model in the real-world environment. In which, EKF-based localization system was presented in Section 3.1. And the ETEB technique-based local planner is proposed in Section 4.3. This work has been tested and validated on the QBot-2e mobile robot platform in indoor environment. All experimental results show that the robot can achieve the goal without any collision, and avoid dynamic obstacles safely and proactively.

The section is organized as follows. Subsection 4.4.1 presents the completed navigation framework in which it will be used the proposed localization algorithm (in Section 3.1) and the proposed motion planning model (in Section 4.3). The experimental results in real-world environments are described in Subsection 4.4.2. The conclusions of the section are provided in Subsection 4.4.3.

4.4.1. Completed navigation framework

Mobility is the most essential navigation issue of mobile robots. To allow the mobile robots to navigate safely in a real-world environment, the mobile robots must deal with typical functional blocks of the navigation system, including perception, localization, motion planning, and motor control, as explained by Siegwart et al. [1]. To accomplish this, in this research project, an extended navigation scheme based on the conventional navigation scheme introduced by Siegwart et al. is proposed, as shown in Fig. 4.21. Figure 4.21 illustrates the proposed navigation scheme for autonomous mobile robots in dynamic environments. The

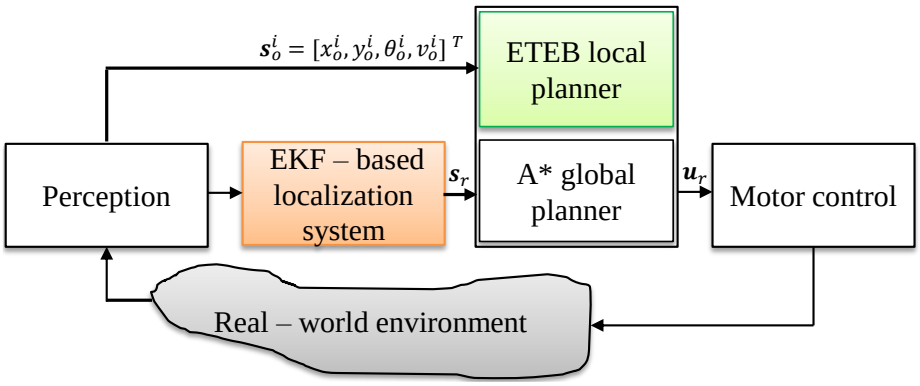


Figure 4.21: The block diagram of the mobile robot navigation system utilize the EKF-based localization algorithm and the ETEB-based obstacle avoidance algorithm.

navigation system consists of four functional blocks: perception, localization, motion planning, and motor control. In first block, the obstacles in the robot's vicinity are detected and tracked. The extended Kalman filter-based localization system is used to estimate the position of the mobile robot in the environment. In the next block, the A* algorithm-based global path planning algorithm is utilized to find the path from the starting position to the given goal. Then the proposed ETEB-based local planner, which is presented in previous Section 4.3, is used to generate the optimal trajectory of the robot from the current position of the robot to the local target. Once the control command of the robot is obtained, it is used as input of the motor control block, which navigate the mobile robot to proactively avoid obstacles and approach the given goal. In the following subsections, the proposed navigation system is presented in details.

a. Perception system

To estimate the states of the interesting objects including humans in the robot's vicinity, the human detection and tracking algorithm developed in [72] were utilized to estimate the human position and velocity. The basic idea of this approach is to fuse the human information detected by laser data as presented in [73] and Kinect sensor data as explained in [74] using a particle filter. A detailed description of the technique can be found in [72]. As a result, the human states are $\mathbf{s}_o^i = [x_o^i, y_o^i, \theta_o^i, v_o^i]^T$. This information is then used as the inputs of the motion planning system, as shown in Fig. 4.21. Figure 4.22 shows an example of the human detection and tracking system, in which the green dot is the human position in the navigation plane of the mobile robot.

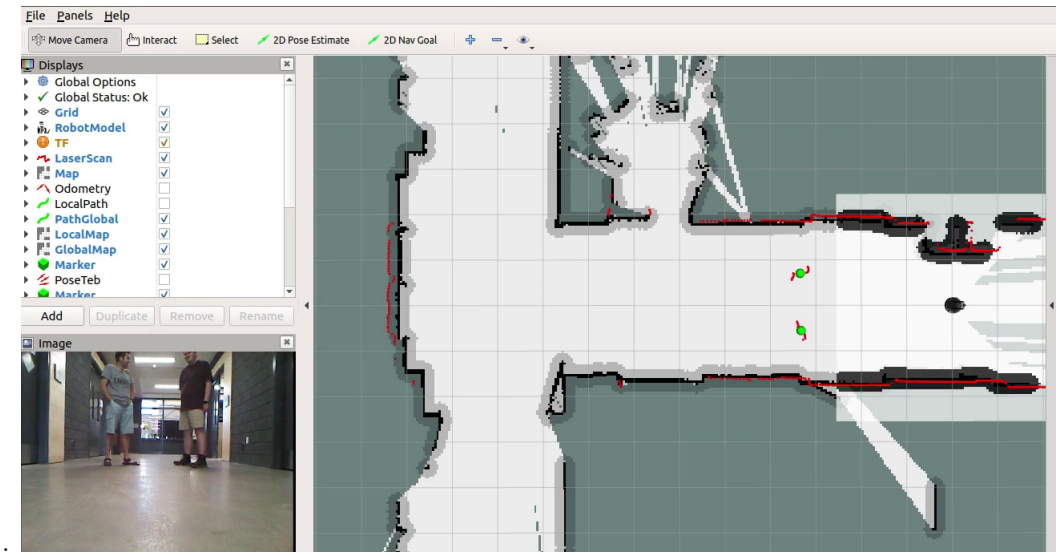


Figure 4.22: The example of the human detection and tracking algorithm.

The human position is then filtered and predict using the Kalman filter algorithm [52] and the data association technique [71]. Thus the output of the human prediction system is $O = [o_1, o_2, ..., o_N]$, where $o_i = (x_i^o, y_i^o)$.

b. Localization system based on EKF filter

In order to navigate autonomously in a dynamic environment, a mobile robot must maintain an accurate knowledge of its position and orientation related to such environment. In this study, the localization system presented in [75] is utilized. Because the mobile robot drives in the indoor environment where have no GPS signal, the author only utilize EKF algorithm to fuse the data from wheel encoders and the IMU sensor (approach II in [75]), to enhance the localization system. It will be better performance localization system, in case GPS system are supported by the infrastructure. Figure 4.23 shows an example of the comparison

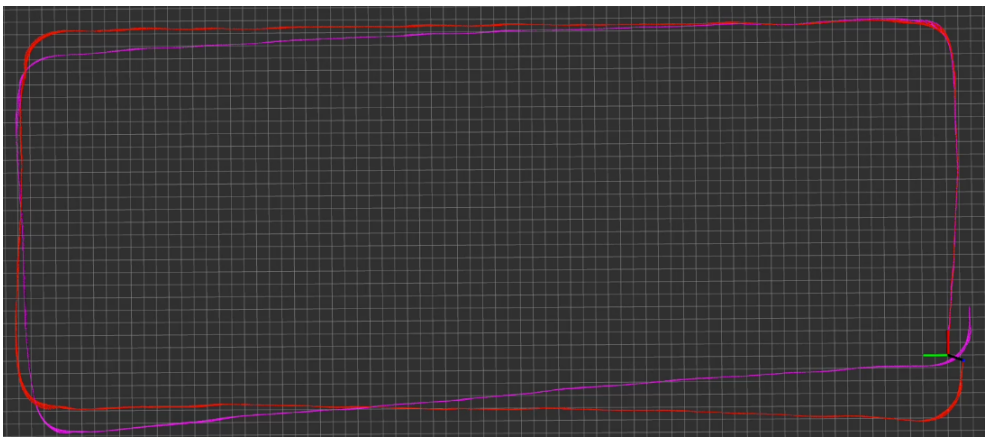


Figure 4.23: The comparison result of localization systems.

between the proposed localization system and the conventional system. In which, the red curve illustrates the estimated robot's position using the proposed localization system, while the magenta curve depicts the robot's position using only odometry information. As a result, the proposed localization system outperforms the conventional algorithm in term of accuracy.

c. Proposed motion planning system

The aim of this system is not only to find one solution that drives the robot from the starting point to the goal point, but also to find the optimal solution with the minimum distance and smoothest maneuvers and without collision obstacles. Normally, the motion planning system is divided into two parts including: (i) the global planner and (ii) the local planner. In which, the global planner is utilize as a prior information of the environment to create the best possible path. Meanwhile, the local planner recalculates the initial plan to avoid possible dynamic obstacles.

In the first part, the A* path planning algorithm [76] is used for finding the global path from the initial position to the final position of the mobile robot. It is the one of the most popular choices for path planning, because it's fairly simple and flexible. This algorithm considers the map as a two-dimensional grid with each tile being a node or a vertex. And it is based on graph search methods and works by visiting vertices (nodes) in a graph starting with the current position of the robot all the way to the goal. The key to the algorithm is identifying the appropriate successor node at each step. Given the information regarding the goal node, the current node, and the obstacle nodes, an educated guess can be made to find the best next node and add it to the list. A* algorithm uses a heuristic algorithm to guide the search while ensuring that it will compute a path with minimum cost. It calculates the distance (also called the cost) between the current location in the workspace, or the current node, and the target location. It then evaluates all the adjacent

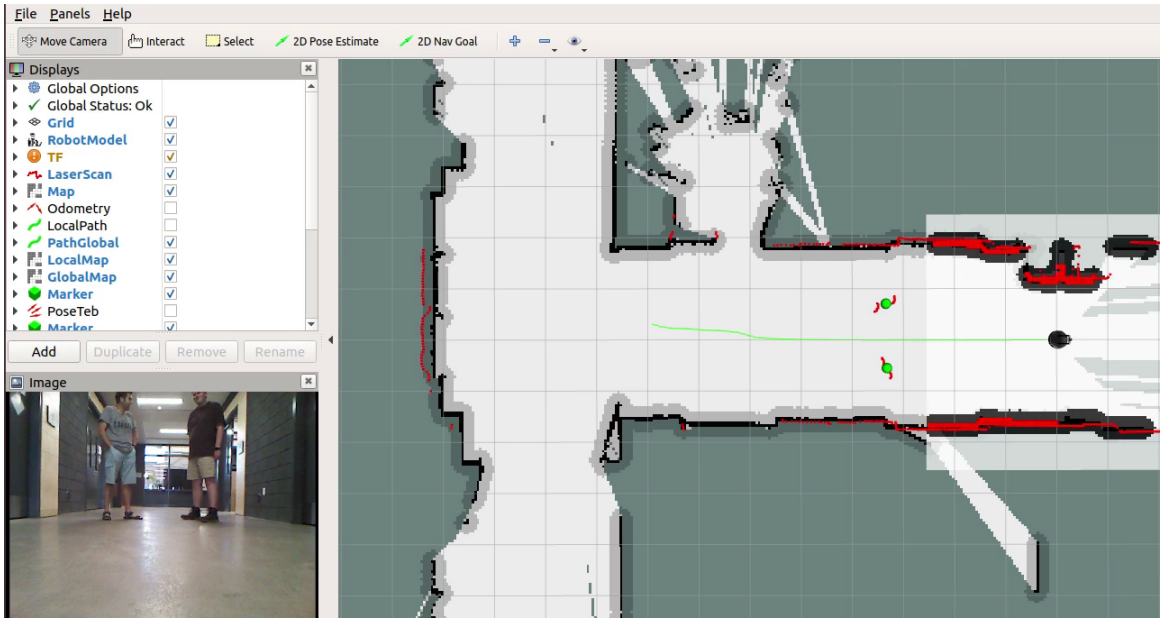


Figure 4.24: The example result of the A*-based path planing algorithm (the green curve).

nodes that are open (i.e., not an obstacle nor already visited) for the expected distance or the heuristic estimated cost from them to the target, also called the heuristic cost $h(n)$. It also determines the cost to move from the current node to the next node, called the path cost $g(n)$. Thus, the total cost to get to the target node $f(n) = h(n)+g(n)$ is computed for each successor node and the node with the smallest cost is chosen as the next point. Figure 4.24 depicts an example of the path planing system using A* algorithm. The green curve is the found path from the initial position to the final position of the mobile robot. Figure 4.24 depicts an example of the path planing system using A* algorithm. The green curve is the found path from the initial position to the final position of the mobile robot.

In the second part, ETEB -based local planner, which is presented in Section. 4.3, is utilized to drive the autonomous mobile robot to the

given goal. The ETEB algorithm incorporates robot dynamics, and current and future positions of the surrounding obstacles to generates the safe and proactive trajectory for the mobile robot. As a result, the robot equipped with the proposed ETEB algorithm is able to safely and proactively navigate in dynamic environments.

b. Motor control

Once the optimal trajectory is generated by the proposed ETEB algorithm, the motion control command $\mathbf{u}_r=[\upsilon_r, \omega_r]^T$ is extracted and used to drive the mobile robot to safely and proactively avoid the obstacles in the robot's vicinity and approach a given goal. In this study, a two-wheel differential drive mobile robot platform presented in Section 2.1.2 is made use of. Therefore the state of the mobile robot is governed by the equation presented in (2.4).

4.4.2. Experimental setup and results

To validate its feasibility and effectiveness have been implemented and installed the entire navigation system on the mobile robot platform. We then conduct experiments in a corridor-like environment to examine whether the robot equipped with the proposed ETEB algorithm could safely and proactively avoid obstacles. In this study, the humans is made use of as moving obstacles in all experiments.

Experimental Setup

We used a QBot-2e mobile robot platform[7] equipped with a RPLIDAR A3 laser rangefinder as shown in Fig. 4.25 which is used for detecting

[7]https://www.quanser.com/products/qbot-2e

humans in the vicinity of the mobile robot. The parameters of the mobile robot platform is presented in detail in Section 2.1.1 and Table 4.5.

Table 4.5: Parameters of the QBot 2e mobile robot platform

Parameters of laser rangefinder	**Value**
Distance measurement	$\leq$ 25 m
Angular field of view	360^o
Resolution	0.33^o
Parameters of Kinect	
Depth sensor ranged	0.8 m to 6 m
Vertical and horizontal viewing angle	43^o and 57^o
Parameter of Encoder	
Counts per revolution	2578 pulses
Resolution	0.14^o

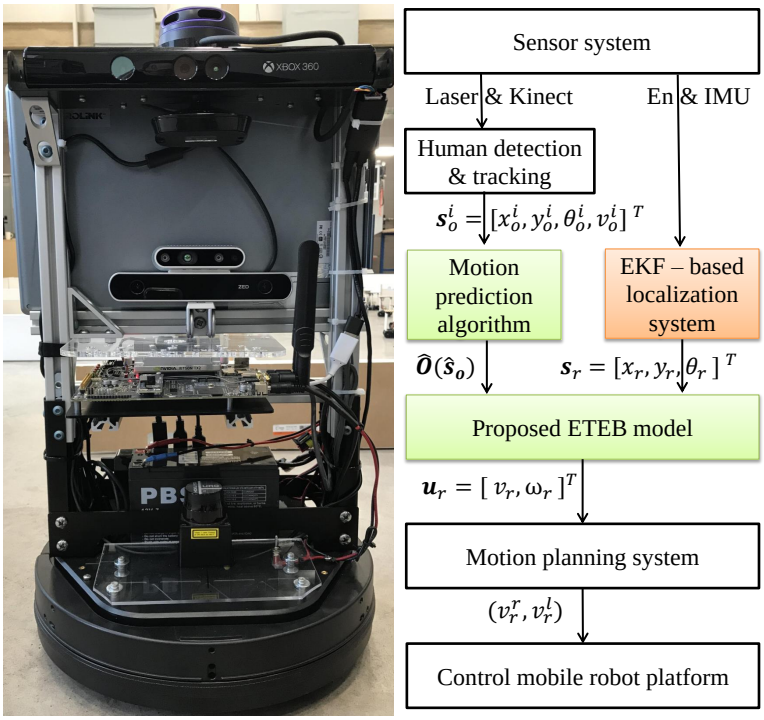


Figure 4.25: The QBot-2e mobile robot platforms and the data flow diagram of the proposed framework.

The software of the proposed framework is implemented using the C/C++ programming language. The entire navivation framework are developed based on the Robot Operating System (ROS) [43] and run on Jetson TX2 board. The conventional TEB package[8] was inherited and

[8]http://wiki.ros.org/teb_local_planner

modified for developing the proposed extended TEB algorithm. The human detection and tracking algorithm presented in Section 4.4.1 is utilized to estimate the human states $\mathbf{s}_p^i$ including the position and motion of the humans in the robot's vicinity.

Experimental Results

We conducted an experiment in a corridor to examine whether the mobile robot could avoid humans, obstacles while navigating safely in the real-world environment. The robot is requested to navigate from the initial position to the final goal, while avoiding the obstacles during its navigation. Notice that obstacles are always moving in the corridor. The snapshots of the experimental results are shown in Fig. 4.26. A video with our experimental results can be found at the hyperlink[9]. The

(a) 46[s] (b) 48[s] (c) 50[s]

Figure 4.26: Three snapshots at three timestamps of the experiment in the sparse environment.

experimental results reveal that, the proposed ETEB algorithm is feasibility and effectiveness in real-world environments. It enables the mobile robot to safely avoid dynamic humans in the vicinity of the robot. In other words, the proposed extended timed elastic band algorithm is able to apply to the real-world environments.

[9]https://www.youtube.com/watch?v=LmIf26qeTg8

4.4.3. Remarks

In this section, we have incorporated the proposed EKF-based localization algorithm and the proposed ETEB motion planning model into an integrated mobile robot navigation system. The effectiveness and feasibility of the proposed framework were proved through a series of experiments in real-world environments. The experimental results illustrated that, the entire navigation system including perception, localization, motion planning also motor control systems is capable of driving the mobile robots to safely and proactively avoid dynamic obstacles in the surrounding environment, providing the safe navigation for the robots.

4.5. Conclusions and discussion

In this chapter, three effective local planning algorithms in the motion planning system for autonomous mobile robots in dynamic environments have proposed, including the EDWA algorithm, the PTEB algorithm and the ETEB algorithm.

Firstly, the EDWA algorithm combines traditional DWA technique with the HRVO model to generate a proactive obstacle avoidance system. As a result, the mobile robots could navigate safely and smoothly to its goal. Although this proposed algorithm has been verified in real-world environments, they are sparse dynamic environments. Therefore, the EDWA algorithm might not suitable with various dynamic environments, because the robots are not able to transit across obstacles in their vicinity.

To overcome the aforementioned issue, the proposed PTEB algorithm

using parallel trajectory planning in spatially distinctive topologies was investigated. The simulation results demonstrate significant benefits of incorporating the TEB technique with HRVO model. The mobile robots equipped the proposed PTEB algorithm are able to transit across obstacles and proactively avoid static and dynamic obstacle in the robot's vicinity. Although the PTEB algorithm outperform the EDWA algorithm in this issue, it is not yet foreseeable future states of obstacles.

To deal with these weaknesses and improve the performance of the navigation system, the proposed ETEB algorithm, which integrated the future states of the surrounding obstacles into the conventional TEB technique, has been proposed.

Finally, an entire navigation system including four typical components have been presented. We conducted experiments in both simulation and real-world environments. The results demonstrated the effectiveness and feasibility of the proposed algorithms.

Note that the research reported in this chapter gave rise to our publications [J3], [C4] and [J6].

Chapter 5
CONCLUSIONS AND FUTURE WORKS

In this chapter we draw conclusions derived on each chapter of this thesis, and we gather them into a joint reflection to illustrate the more important issues. We then suggest some future research directions that steam out from this work.

5.1. Conclusions

We have described the motivation, the contribution and the outline of the thesis in Chapter 1. We then have provided the conventional techniques which are utilized in our proposed systems in Chapter 2. In Chapter 3 and 4, we have presented the proposed localization algorithms as well as local planning algorithms in the motion planning system.

In Chapter 3, the efficient localization algorithms have been proposed to improve the accuracy of estimating the robot's pose in the dynamic environment, including EKF and PF-based localization algorithms. The EKF-based localization algorithm utilizes the EKF algorithm to fuse data getting from the sensor system consisting of the wheel encoders, GPS and IMU sensors. As a result, this simple model has highly efficient and suitable for the robot driving in the environments with sufficient information. Whereas, the PF-based localization algorithm is utilized in environments with incomplete information or interrupted signal of

the sensor system. The proposed algorithms are tested in the simulation environment with different scenarios. The simulation results showed that the mobile robot equipped with the proposed localization algorithms have higher accuracy of estimating pose than the existing systems.

In Chapter 4, three new efficient local planning algorithms of the motion planning model have been proposed for autonomous mobile robots in dynamic environments, including EDWA, PTEP and ETEB algorithms.

The main idea of the proposed EDWA algorithm is to incorporate the velocity vector generated by the HRVO model into the objective function of the conventional DWA technique. We validate the effectiveness of the proposed algorithm through a series of experiments in both simulated and real-world environments. The experimental results show that, our proposed EDWA algorithm is capable of driving the mobile robots to proactively avoid dynamic obstacles in the robot field of view, providing the safe navigation for the mobile robots.

To enable the autonomous mobile robots transit across obstacles and proactively navigate in dynamic environments we have been proposed the PTEB algorithm for online trajectory planning by incorporating the potential collision generated by the HRVO model into the objective function of the conventional TEB technique. The output of the proposed PTEB algorithm is the optimal trajectory, which is utilized to control the mobile robots. A series of experiments in various simulation environments is conducted to validate the effectiveness of the proposed algorithm. The simulation results demonstrate that our proposed PTEB algorithm is able to drive the mobile robot to proactively avoid dynamic

obstacles and safely transit across obstacles in the dynamic environment.

In addition, we propose an ETEB algorithm for online trajectory planning, which takes both future and current states of the surrounding obstacles into account the TEB model of the motion planning model. Therefore, it allows the mobile robot to navigate more effectively in terms of proactively avoiding potential collisions in the dynamic environment. We validate the effectiveness of the proposed algorithm through a series of experiments in simulated environments.

Finally, the completed navigation system for the autonomous mobile robot in a dynamic environment have been presented. In which, we have presented an integrated navigation system for the autonomous mobile robot in the dynamic environment by incorporating the techniques proposed in our previous studies, including the EKF localization and ETEB local planning algorithms, into a completed navigation system. We conducted experiments the completed navigation system on the robot platform in a real-world environment. The experimental results demonstrate that, the proposed algorithms have feasibility and the proposed navigation system is capable of driving the mobile robots to proactively avoid dynamic obstacles, providing the safe navigation for the robots.

5.2. Limitations

Although the results in simulation as well as real-world environments illustrate the effectiveness of the proposed algorithms, the dissertation still suffers from some limitations.

The dissertation lacks of examining the proposed PF based-localization

algorithm on the mobile robot platform in real-world environments. All of the proposed navigation systems in this dissertation are only verify the effectiveness in sparse and semi dynamic environments. The proposed PTEB algorithm is only examined in simulation environments. The rest two proposed navigation algorithms are installed in our mobile robot platform and examined in real-world environments. However, we only conduced experiments in indoor environments.

5.3. Future works

The potential future directions for research based on the results presented in this thesis are given in the rest of this section.

Firstly, we will install the complete navigation system on the our mobile robot platform and conduct experiments in various type of environments including indoor and outdoor, semi-dynamic and dynamic environments to verify the effectiveness of the proposed algorithms.

Secondly, applying powerful techniques [77] and [78] for predicting the future position and trajectory of obstacles in the robot's vicinity and then incorporating this information into the motion planning system of the mobile robot.

Thirdly, fast and efficient motion planning systems should be proposed for mobile robot navigation in crowded dynamic environments.

Finally, to adapt with different dynamic environments, deep neural networks [79] and deep reinforcement learning techniques [80] should also be considered to improve the learning efficiency and navigation performance of the mobile robot.

PUBLICATIONS

[C1] **L. A. Nguyen**, P. T. Dung, T. D. Ngo, X. T. Truong, "Improving the accuracy of the autonomous mobile robot localization systems based on the multiple sensor fusion methods," in: *2019 3rd IEEE International Conference on Recent Advances in Signal Processing, Telecommunications Computing (SigTelCom)*, Hanoi, Vietnam, pp. 33–37, March 2019, doi: 10.1109/SIGTELCOM.2019.8696103.

[J2] **L. A. Nguyen**, L.T. Nghia, D. N. Thang, P. T. Dung, X. T. Truong, "Localization system based on the particle filter algorithm and sensor fusion technique for autonomous mobile robots in the interrupted sensor data," in: *Special issue on Measurement, Control and Automation*, pp. 46–53, Dec. 2019.

[J3] **L. A. Nguyen**, P. T. Dung, X. T. Truong, "An Integrated Navigation System for Autonomous Mobile Robot in Dynamic Environments," in: *Journal of Military Science and Technology*, pp. 32–46, May 2020.

[C4] **L. A. Nguyen**, P. T. Dung, X. T. Truong, "A Proactive Trajectory Planning Algorithm for Autonomous Mobile Robots in Dynamic Social Environments," in: *2020 17th International Conference on Ubiquitous Robots (UR)*, Kyoto, Japan, pp. 309–314, June 2020, doi: 10.1109/UR49135.2020.9144925.

[C5] Van Bay Hoang, **L. A. Nguyen** and Xuan Tung Truong, "Social

constraints-based socially aware navigation framework for mobile service robots," in: *NAFOSTED Conference on Information and Computer Science*, Nov. 2020.

[J6] **L. A. Nguyen**, P. T. Dung, T. D. Ngo, X. T. Truong, "An Efficient Navigation System for Autonomous Mobile Robots in Dynamic Social Environments," in: *International Journal of Robotics and Automation, ACTA Press* (ISI-SCIE). DOI: 10.2316/J.2021.206-0490, Dec. 2020.

REFERENCES

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. The MIT Press, February 2011.

[2] N. V. Tinh, N. T. Linh, P. T. Cat, P. M. Tuan, M. N. Anh, and N. P. Anh, "Modeling and feedback linearization control of a nonholonomic wheeled mobile robot with longitudinal, lateral slips," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 996–1001.

[3] N. V. Tinh, K. Nguyentien, T. Do, and P. M. Tuan, "Neural network-based adaptive sliding mode control method for tracking of a nonholonomic wheeled mobile robot with unknown wheel slips, model uncertainties, and unknown bounded external disturbances," *Acta Polytechnica Hungarica*, vol. 15, no. 2, pp. 103–123, 2018.

[4] K. Nguyentien, L. Le, T. Do, N. V. Tinh, and P. M. Tuan, "Robust control for a wheeled mobile robot to track a predefined trajectory in the presence of unknown wheel slips," *Vietnam Journal of Mechanics*, vol. 40, no. 2, pp. 141–154, 2018.

[5] N. V. Tinh, T. Hoang, P. M. Tuan, and D. P. Nam, "A gaussian wavelet network-based robust adaptive tracking controller for a wheeled mobile robot with unknown wheel slips," *International Journal of Control*, vol. 92, no. 11, pp. 2681–2692, 2019.

[6] T. T. Hoang, P. M. Duong, N. T. T. Van, D. A. Viet, and T. Q. Vinh, "Development of an ekf-based localization algorithm using compass sensor and lrf," in *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)*, 2012, pp. 341–346.

[7] N. V. Tinh, M. D. Phung, T. H. Tran, and Q. V. Tran, "Mobile robot localization using fuzzy neural network based extended kalman filter," in *2012 IEEE International Conference on Control System, Computing and Engineering*, 2012, pp. 416–421.

[8] N. M. Tien and P. X. Minh, "Research deverlopment of mobile robot mounted camera automatically seaching and tracking moving target," in *The Vietnam Conference of Control and Automation*, no. 133, 2011.

[9] N. V. Tinh, P. T. Cat, P. M. Tuan, and T. T. Q. Bui, "Visual control of integrated mobile robot-pan tilt-camera system for tracking a moving target," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2014.

[10] N. V. Tinh, P. M. Tuan, P. T. Cat, and T. T. Q. Bui, "Trajectory planning and control of mobile robot for transportation in warehouse," in *2011 International Conference on Robotics and Automation*, 2011, pp. 1566–1571.

[11] T. Sebastian, B. Wolfram, and F. Dieter, *Probabilistic Robotics*. The MIT Press, August 2005.

[12] M. Brossard and S. Bonnabel, "Learning wheel odometry and imu errors for localization," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 291–297.

[13] E. North, J. Georgy, M. Tarbouchi, U. Iqbal, and A. Noureldin, "Enhanced mobile robot outdoor localization using ins/gps integration," in *2009 International Conference on Computer Engineering Systems*, 2009, pp. 127–132.

[14] T. Lee, J. Shin, and D. Cho, "Position estimation for mobile robot using in-plane 3-axis imu and active beacon," in *2009 IEEE International Symposium on Industrial Electronics*, July 2009, pp. 1956–1961.

[15] D. M. G. A. I. N. Sumanarathna, I. A. S. R. Senevirathna, K. L. U. Sirisena, H. G. N. Sandamali, M. B. Pillai, and A. M. H. S. Abeykoon, "Simulation of mobile

robot navigation with sensor fusion on an uneven path," in *2014 International Conference on Circuits, Power and Computing Technologies*, March 2014, pp. 388–393.

[16] J. Woo, Y. Kim, J. Lee, and M. Lim, "Localization of mobile robot using particle filter," in *2006 SICE-ICASE International Joint Conference*, Oct 2006, pp. 3031–3034.

[17] L. Marin, M. Valles, A. Soriano, A. Valera, and P. Albertos, "Multi sensor fusion framework for indoor-outdoor localization of limited resource mobile robots," *Sensors*, vol. 13, no. 10, pp. 14 133–14 160, 2013.

[18] M. A. K. Jaradat and M. F. Abdel-Hafez, "Enhanced, delay dependent, intelligent fusion for ins/gps navigation system," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1545–1554, May 2014.

[19] J. Woo, Y. Kim, J. Lee, and M. Lim, "Localization of mobile robot using particle filter," in *2006 SICE-ICASE International Joint Conference*, Oct 2006, pp. 3031–3034.

[20] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/fir filtering," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1089–1098, Oct 2015.

[21] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The Office Marathon: Robust navigation in an indoor office environment," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 300–307.

[22] G. Ferrer, A. Garrell, and A. Sanfeliu, "Social-aware robot navigation in urban environments," in *European Conference on Mobile Robots (ECMR)*, September 2013, pp. 331–336.

[23] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using

a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.

[24] X. T. Truong and T. D. Ngo, "Dynamic social zone based mobile robot navigation for human comfortable safety in social environments," *International Journal of Social Robotics*, pp. 1–22, 2016.

[25] X. T. Truong, N. Y. Voo, and T. D. Ngo, "Socially aware robot navigation system in human interactive environments," *Intelligent Service Robotics*, vol. 10, no. 4, pp. 287–295, 2017.

[26] C. Rosmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[27] ——, "Kinodynamic trajectory optimization and control for car-like robots," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2017, pp. 5681–5686.

[28] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, pp. 4282–4286, 1995.

[29] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Transactions on Robotics and Automation*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[30] J. van den Berg, C. L. Ming, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 1928–1935.

[31] J. Snape, J. Van den Berg, S. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, August 2011.

[32] D. Zhang, Z. Xie, P. Li, J. Yu, and X. Chen, "Real-time navigation in dynamic human environments using optimal reciprocal collision avoidance," in *Proceedings of the 2015 IEEE International Conference on Mechatronics and Automation*, August 2015, pp. 2232–2237.

[33] D. Claes, D. Hennes, and K. Tuyls, "Towards human-safe navigation with proactive collision avoidance in a shared workspace," in *Workshop on On-line decision-making in multi-robot coordination*, 2015.

[34] P. Fiorini and Z. Shillert, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, pp. 760–772, 1998.

[35] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, March 1985, pp. 500–505.

[36] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, June 1991.

[37] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, May 1993, pp. 802–807.

[38] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[39] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[40] C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *7th German Conference on Robotics*, May 2012, pp. 1–6.

[41] C. Rosmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[42] C. Rosmann, M. Oeljeklaus, F. Hoffmann, and T. Bertram, "Online trajectory prediction and planning for social robot navigation," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2017, pp. 1255–1260.

[43] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 32, 2009, pp. 151–170.

[44] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *In Proceedings of the 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.

[45] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[46] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation*, May 2011, pp. 1–4.

[47] L. A. Nguyen, P. T. Dung, T. D. Ngo, and X. T. Truong, "Improving the accuracy of the autonomous mobile robot localization systems based on the multiple sensor fusion methods," in *2019 3rd IEEE International Conference on Recent Advances in Signal Processing, Telecommunications Computing (SigTelCom)*, 2019, pp. 33–37.

[48] L. A. Nguyen, L. Nghia, D. N. Thang, P. T. Dung, and X. T. Truong, "Localization system based on the particle filter algorithm and sensor fusion technique for autonomous mobile robots in the interrupted sensor data," *Special issue on Measurement, Control and Automation*, vol. 4, no. 2, pp. 46–53, December 2019.

[49] L. A. Nguyen, P. T. Dung, T. D. Ngo, and X. T. Truong, "A proactive trajectory planning algorithm for autonomous mobile robots in dynamic social environ-

ments," in *2020 17th International Conference on Ubiquitous Robots (UR), Kyoto, Japan*, 2020, pp. 309–314.

[50] ——, "An efficient navigation system for autonomous mobile robots in dynamic social environments," *International Journal of Robotics and Automation, ACTA Press(ISI-SCIE)*, December 2020.

[51] ——, "An integrated navigation system for autonomous mobile robot in dynamic environments," *Journal of Military Science and Technology*, May 2020.

[52] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA: Tech. Rep. TR-95-041, University of North Carolina at Chapel Hill, 2006.

[53] D. F. S. Thrun, W. Burgard, *Probabilistic Robotics*. MIT Press, 2006.

[54] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[55] D. Hennes, D. Claes, W. Meeussen, and K. Tuyls, "Multi-robot collision avoidance with localization uncertainty," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, 2012, pp. 147–154.

[56] V. J. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 5, pp. 1058–1069, Sep. 1990.

[57] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," 05 1997, pp. 2920 – 2925.

[58] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *In Proc. of the IEEE International Conference on Robotics and Automation*, 1996, pp. 3375–3382.

[59] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, May 1999, pp. 341–346.

[60] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991.

[61] D. Claes, D. Hennes, K. Tuyls, and W. Meeussen, "Collision avoidance under bounded localization uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2012, pp. 1192–1198.

[62] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer series in operations research, 1999.

[63] S. Bhattacharya, *Topological and geometric techniques in graph-search based robot planning*. Ph.D. dissertation, University of Pennsylvania, 2012.

[64] S.Phillips and S.Narasimhan, "Automating data collection for robotic bridge inspections," *Journal of Bridge Engineering*, vol. 8, no. 24, 2019.

[65] H. Deilamsalehy and T. C. Havens, "Sensor fused three-dimensional localization using imu, camera and lidar," in *2016 IEEE SENSORS*, Oct 2016, pp. 1–3.

[66] J. D. Toledo, Jonay, R. Arnay, D. Acosta, and L. Acosta, "Improving odometric accuracy for an autonomous electric cart," *Sensors*, vol. 18, no. 1, 2018.

[67] J. Borenstein and Liqiang Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869–880, Dec 1996.

[68] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *2013 European Conference on Mobile Robots*, Sep. 2013, pp. 138–143.

[69] G. Ferrer and A. Sanfeliu, "Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, September 2014, pp. 1730–1735.

[70] X. T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1743–1760, October 2017.

[71] H. W. Kuhn, *The Hungarian Method for the Assignment Problem.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 29–47.

[72] X. T. Truong, N. Y. Voo, and T. D. Ngo, "RGB-D and laser data fusion-based human detection and tracking for socially aware robot navigation framework," in *Proceedings of the 2015 IEEE Conference on Robotics and Biomimetics*, Dcember 2015, pp. 608–613.

[73] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 3402–3407.

[74] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with rgb-d data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2012, pp. 2101–2107.

[75] L. A. Nguyen, P. T. Dung, T. D. Ngo, and X. T. Truong, "Improving the accuracy of the autonomous mobile robot localization systems based on the multiple sensor fusion methods," in *International Conference on Recent Advances in Signal Processing, Telecommunications Computing*, 2019, pp. 33–37.

[76] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[77] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 961–971.

[78] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *https://arxiv.org/abs/1905.06113v3*, 2019.

[79] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, p. 85–117, Jan 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003

[80] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved.*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[81] C. S. Raveena, R. S. Sravya, R. V. Kumar and A. Chavan, "Sensor Fusion Module Using IMU and GPS Sensors For Autonomous Car," *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 2020, pp. 1-6, doi: 10.1109/INOCON50539.2020.9298316.

[82] G. T. Lee, S. B. Seo and W. S. Jeon, "Indoor localization by Kalman filter based combining of UWB-positioning and PDR", *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, pp. 1-6, Jan. 2021.

[83] L. Alsmadi, X. Kong, K. Sandrasegaran and G. Fang, "An Improved Indoor Positioning Accuracy Using Filtered RSSI and Beacon Weight," *in IEEE Sensors Journal*, vol. 21, no. 16, pp. 18205-18213, 15 Aug.15, 2021, doi: 10.1109/JSEN.2021.3085323.

[84] V. B. Hoang, L. Anh Nguyen, P. V. Nguyen and X. -T. Truong, "A Time-Dependent Motion Planning System for Mobile Service Robots in Dynamic Social Environments," *2021 International Conference on System Science and Engineering (ICSSE)*, 2021, pp. 464-469, doi: 10.1109/ICSSE52999.2021.9538489.

[85] Nguyen D.T., Hoang V.B., Le T.N., Nguyen T.H., Truong X.T, "Social Hybrid Reciprocal Velocity Obstacle-Based Socially Aware Mobile Robot Navigation Framework," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, INISCOM 2021, vol 379. Springer, Cham.*

[86] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen and J. Zhang, "Learning Local Planners for Human-aware Navigation in Indoor Environments," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6053-6060, doi: 10.1109/IROS45743.2020.9341783.