MINISTRY OF NATIONAL DEFENSE
MILITARY TECHNICAL ACADEMY

NGUYEN THI LAN ANH

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

Major: Control engineering and automation
Code: 9.52.02.16

SUMMARY OF TECHNICAL DOCTORAL DISSERTATION

Ha Noi - 2021

THIS WORK IS COMPLETED AT MILITARY TECHNICAL ACADEMY - MINISTRY OF NATIONAL DEFENSE

Supervisor:

Assoc. Prof. Dr. Pham Trung Dung

Opponent 1:

Prof. Dr. Chử Đức Trình

Opponent 2:

Assoc. Prof. Dr. Lê Thị Lan

Opponent 3:

Prof. Dr. Lê Hùng Lân

This thesis will be defended before The Academy-Level Doctoral Examination Board according to the Decision No 3515/QĐ-HV date 06 month 9 year 2021 of the President of Military Technical Academy, meeting at the Military Technical Academy at time ... date ... month ... year .....

This thesis could be found at:
- National Library of Vietnam
- Library of Military Technical Academy

PUBLICATIONS

[1] **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.

[2] **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.

[3] **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.

[4] **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.

[5] 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.

[6] **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.

CONCLUSIONS AND FUTURE WORK

This dissertation focuses on developing an efficient navigation system that enables a mobile robot to navigate autonomously, safely and proactively in the dynamic environment. The dissertation has the following main contributions.

- First, two sensor fusion-based localization algorithms are proposed to improve accuracy of the conventional localization systems, including the EKF -based localization algorithm and the Particle filter (PF)-based localization algorithm, when the robot moves in the environments with sufficient information and the interrupted signal situation, respectively.

- Second, three new local planning algorithms, including EDWA, PTEB and 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.

- Third, the integrated navigation system based on the proposed algorithms, including the EKF-based localization algorithm and the ETEB algorithm, is utilized in real - world environments to illustrate efficient and feasibility of the proposed system.

However, the dissertation still suffers from some limitations. The dissertation lacks of examining the proposed PF based-localization and proposed PTEB algorithm on the mobile robot platform in real-world environments. And we only conduced experiments in indoor environments.

Building upon this research, there are a number of directions for future work arisen from the dissertation. Firstly, we will conduct the experiments in various type of environments including indoor and outdoor, semi-dynamic and dynamic environments. Secondly, applying powerful techniques [77] and [78] for predicting the future position and trajectory of obstacles in the robot's vicinity and then incorporating into the motion planning system of the mobile robot. Thirdly, efficient motion planning systems should be proposed for a mobile robot in crowded dynamic environments. Finally, deep neural networks [79] and deep reinforcement learning techniques [80] should also be considered to improve navigation performance of the mobile robot.

INTRODUCTION

Navigation is an essential issue for an autonomy of mobile robots in a dynamic environment. To develop an efficient navigation system that enables a mobile robot to navigate autonomously, safely and proactively in a dynamic environment, we can break down into two 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. The main contributions of the dissertation are outlined as follows.

- Two sensor fusion-based localization algorithms are proposed, including EKF -based localization and the Particle filter (PF)-based localization algorithms. We used these algorithms to improve the accuracy of the localization system when the mobile robot moves in the environments with sufficient information as well as the interrupted signal situation.

- Three new local planning algorithms for the motion planning system of autonomous mobile robots in dynamic environments are proposed, including EDWA, PTEB and 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.

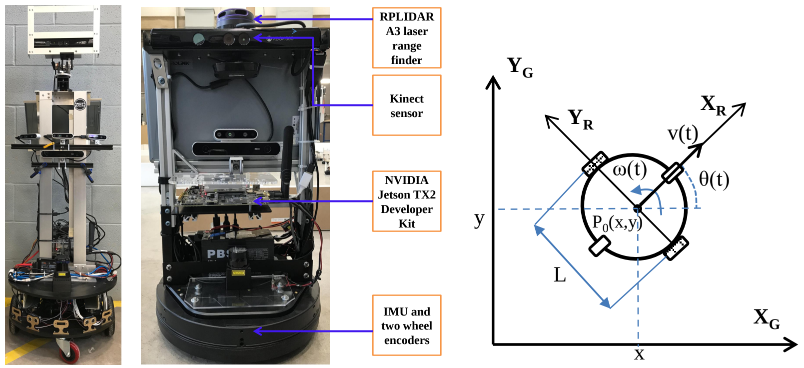
The dissertation is organized into fives chapters except for references. Chapter 2 gives the backgrounds related to this research. Chapter 3 presents two proposed localization algorithms. Chapter 3 introduces three new proposed local planning algorithms to enhancing performance of the motion planning system and conduct experiments in both simulation and real-world environments. The final, conclusions and future works are drawn in the Chapter 5.

Chapter 1
BACKGROUND

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

1.1.1 Mobile robot platforms



(a) (b) (c)

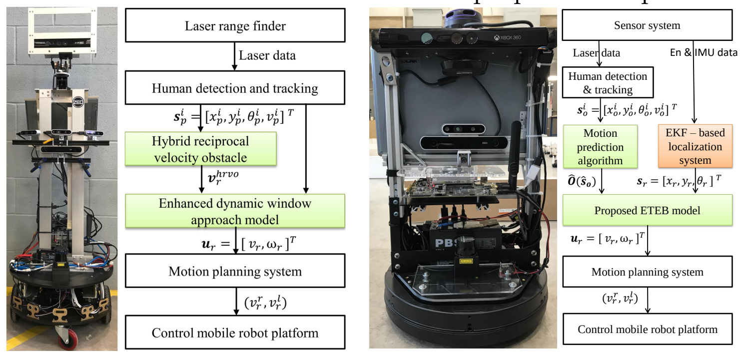
Figure 1.1: (a) Eddie mobile robot platform;(b) QBot-2e mobile robot platform; (c) The global reference frame and the robot reference frame.

1.1.2 Kinematic model of a differential - drive robot

In our studies, kinematic model of the differential drive robot is utilized in both simulations and experiments. For the differential drive robot, shown in Fig. 1.1(c), the position can be estimated starting from a known position by the incremental travel distances in an interval time Δt. 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

[1]http://morelab.org

position. 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 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 and used as input of the motor control block. We then installed the proposed completed navigation



(a) (b)

Figure 3.13: (a) The Eddie mobile robot platform; (b) The QBot-2e mobile robot platforms.

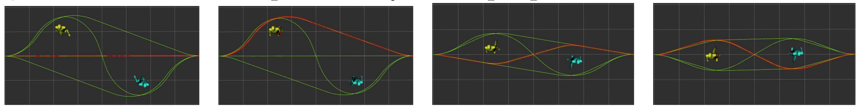
system on the robot platform seen as Fig. 3.13(b) and conducted experiments in a corridor-like environment to examine the effectiveness and feasibility of its. A video with our experimental results can be found at the hyperlink[6]. The experimental results illustrated that, the proposed entire navigation system 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.

3.5 Conclusions

Three effective local planning algorithms in the motion planning system for autonomous mobile robots in dynamic environments have proposed, including EDWA, PTEB and ETEB algorithms. The 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.

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

in Figs. 3.11(b), 3.11(d). are generated behind the left person, it illustrates that, the robot is able to proactively avoid people.



(a) T2 [s]   (b) T2 [s]   (c) T3 [s]   (d) T3 [s]

Figure 3.11: Snapshots at two times tamps of the two experiments.

b. Simulation experiment in Stage environment

We continue to validate the effectiveness of the TEB algorithm in terms of quantitative by experimenting in the Stage environment. Firstly, we incorporate the proposed ETEB algorithm into the conventional navigation scheme. Then the developed ETEB-based navigation system, as presented in Fig. 3.10(b) are built in Stage environment and conducted two experiments in the scenario shown in Fig. 3.8(b). The set of parameter is also presented in Table 3.3 and the threshold of the CI value is 0.54. The experiments results Fig. 3.12 prove that

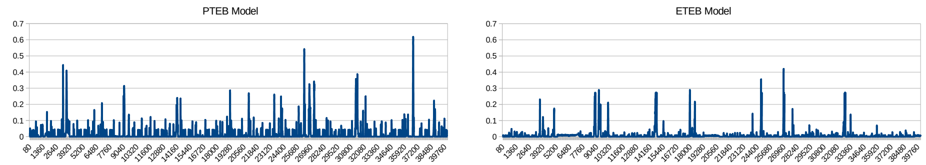


Figure 3.12: The simulation results of the two experiments.

the ETEB algorithm incorporating both the current and future sates of the surrounding obstacles into the conventional TEB algorithm is more effective than PTEB algorithm in terms of proactively avoiding potential collisions in dynamic environment.

3.4 Proposed integrated navigation system

The main contribution of this section is to demonstrate the integration of using EKF-based localization algorithm with ETEB algorithm. A completed navigation system is combined from four fundamental models including perception, localization, motion planning, and motor control model, as shown in Fig. 3.1(b). In first block, the obstacles in the robot's vicinity are detected and tracked by the human detection and tracking algorithm developed in [75]. The proposed EKF-based localization system is used to estimate the robot's

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}. After the duration Δt the velocity motion model of the robot is as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} v_{k-1}\Delta t \cos(\theta_{k-1} + \frac{\omega_{k-1}\Delta t}{2}) \\ v_{k-1}\Delta t \sin(\theta_{k-1} + \frac{\omega_{k-1}\Delta t}{2}) \\ \omega_{k-1}\Delta t \end{bmatrix} \tag{1.1}$$

1.2 Bayesian filters for localization systems

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 growing, the robot must localize the relationship of itself to its environment. Thus, the robot might use its exteroceptive 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. Two different methods of probabilistic localization are described, including the Extended Kalman Filter (EKF)-based localization and Particle filter (PF)- based localization.

1.3 Typical obstacle avoidance algorithms

The motion planning systems include of two sub-systems: (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. In order to the mobile robots move safely in the dynamic environments, we focus on developing the local planning algorithms (or obstacle avoidance algorithms) for the motion planning system. Some typical local planning algorithms are presented in this section, including Dynamic Window Approach(DWA), Hybrid Reciprocal Velocity Obstacle(HRVO) and Time Elastic Band (TEB) algorithms.

Chapter 2

SENSOR DATA FUSION - BASED LOCALIZATION ALGORITHMS

2.1 Introduction

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, including sufficient information and Gaussian distribution noises, and insufficient information and Non-Gaussian/Gaussian distribution noises, respectively. 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.

2.2 Extended Kalman Filter - based localization algorithm

In the first case, 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. Due to odometry uncertainty, the uncertainty of the robot configuration increases due to the integration of the odometric error over time. Meanwhile, IMU (accelerometers, gyroscopes and compasses) is used to estimate a relative position, velocity, and acceleration of a moving robot. 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. 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.

The EKF -based localization algorithm composes of two steps as shown

Algorithm 4: 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 (3.13)
    $\hat{\mathbf{V}}(\mathbf{B}_p) = \mathbf{V}(\mathbf{B}_p) + \delta_o \| min\{\mathbf{0}, \hat{\mathbf{O}}^k\}\|_2^2$ using (3.14);
    $\mathbf{B}_p^* \leftarrow$ Optimizer($\mathbf{B}_p, \mathbf{O}, \hat{\mathbf{V}}$);
    $\mathbf{B}^* \leftarrow$ storeLocalOptimalTrajectory($\mathbf{B}_p^*$);
  end for;
  $\mathbf{V}_c \leftarrow$ newObjectiveFunction(); ▷ using (3.11);
  $\hat{\mathbf{B}}^* \leftarrow$ Call Optimizer($\mathbf{B}^*, \mathbf{O}, \mathbf{V}_c$) ▷ Solve (3.10;)
  $\mathbf{u}_r \leftarrow$ According to (2.35)(2.36) and $\hat{\mathbf{B}}^*$
  Return $\mathbf{u}_r = [v_r, \omega_r]^T$

using the TEB optimization in parallel. 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. In the third step, the future states of obstacles are added into the conventional objective function. In the fourth step, the optimal robot trajectory $\hat{\mathbf{B}}^*$ is selected from the set of alternatives $\mathbf{B}_p^*$ by solving (3.10). Finally, the control command $\mathbf{u}_r = [v_r, \omega_r]^T$ of the mobile robot is extracted directly from the selected trajectory $\hat{\mathbf{B}}^*$. This control command is then utilized to control the mobile robot.

3.3.2 Algorithm validation by simulations

To verify the effectiveness of the proposed ETEB algorithm, we conducted examinations in RViz environment[4] and Stage simulator[5] with the set of parameters shown in the Table 3.3.

a. Simulation experiment in RViz Environment

The mobile robot is requested to navigate from left to right, while avoiding two crossing people. Figure. 3.11(a), 3.11(c) show the results of the TEB algorithm, whereas Figs. 3.11(b), 3.11(d) present the results of the proposed ETEB algorithm. The optimal trajectories(the green curve with red arrows) as shown

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

[42]. The mobile robots equipped with the ETEB algorithm can proactively avoid obstacles better and safely navigate to the given goal.

3.3.1 Construction of the ETEB algorithm

The ETEB algorithm is presented in Fig 3.10(a) and Algorithm 4. The future states of the surrounding obstacles is firstly predicted by using the extended Kalman filter algorithm [52] and the data association technique [74]. The output of the motion prediction model is the future states of the obstacles $\hat{\mathbf{s}}_o$, as shown in Fig 3.10(a). Then the proposed algorithm incorporates both the current states $\mathbf{s}_o$ and the future states $\hat{\mathbf{s}}_o$ of the obstacles $\mathbf{O}$ into the exploration step of the TEB algorithm, as shown in Fig. 3.10(a). In stead of using only current states and potential collision as PTEB algorithm, the ETEB algorithm takes both the current and future states into account.

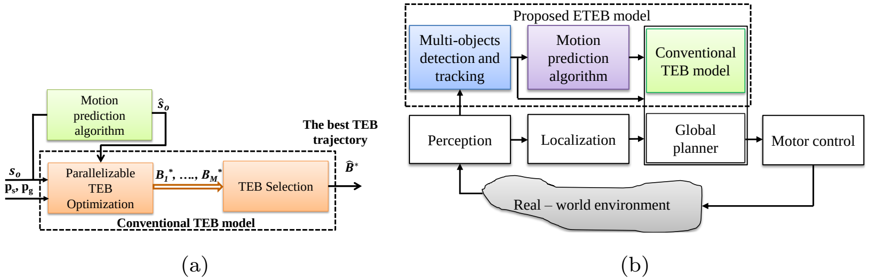


Figure 3.10: (a) The flowchart of proposed extended TEB algorithm; (b) The navigation framework based on the ETEB algorithm.

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 (3.13) is added a new part, as presented in (3.14).

$$V(\mathbf{B}) = \sum_{k=1}^{N-1} [\ \Delta T_k^2 + \delta_h \|\mathbf{h}_k\|_2^2 + \delta_v \|min\{\mathbf{0}, \nu_k\}\|_2^2 + \delta_o \|min\{\mathbf{0}, \mathbf{o}_k\}\|_2^2 + \delta_\alpha \|min\{\mathbf{0}, \alpha_k\}\|_2^2\ ] \quad (3.13)$$

$$\hat{\mathbf{V}}(\mathbf{B}_p) = \mathbf{V}(\mathbf{B}_p) + \delta_o \|min\{\mathbf{0}, \hat{\mathbf{O}}\}\|_2^2 \quad (3.14)$$

The proposed ETEB algorithm is presented detail in Algorithm 4. In the first step, we generate M locally optimal trajectories $\mathbf{B}_p^*$ with p=1, 2,..., M by

in Fig. 2.2, including (i) prediction and (ii) correction step. In the first step, the robot's state predictions are made based on a kinematic motion model (odometry motion model) using encoders. In the second step, the predicted states are corrected based on measurement observations from the sensor system (GPS/IMU).

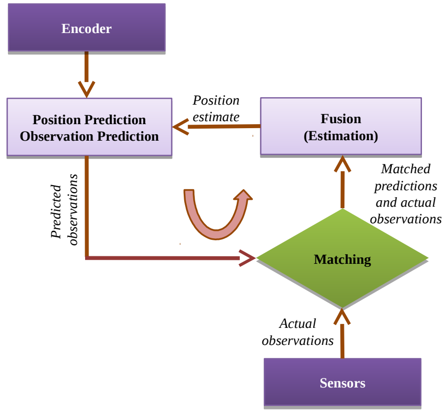


Figure 2.1: The block diagram of the proposed EKF - based localization systems

The EKF algorithm uses a 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).

To verify the usefulness of the EKF-based localization algorithm, we implemented and tested this algorithm in Matlab - based simulations with the kinematic model of the differential drive mobile robot. Three approaches in two scenarios are presented, including (I) Combining Encoder and GPS; (II) Combining Encoder and IMU; (III) Combining Encoder, GPS and IMU. In addition, a statistical data analysis of all the simulations is carried out by using the Mean Error (ME) (2.1) and Mean Square Error (MSE)(2.2).

The simulation results of the two scenarios are shown in Figs. 2.2, Fig. 2.2 and Fig. 2.4. Figure 2.2 shows the robot trajectories. Figure. 2.2 and Fig. 2.4

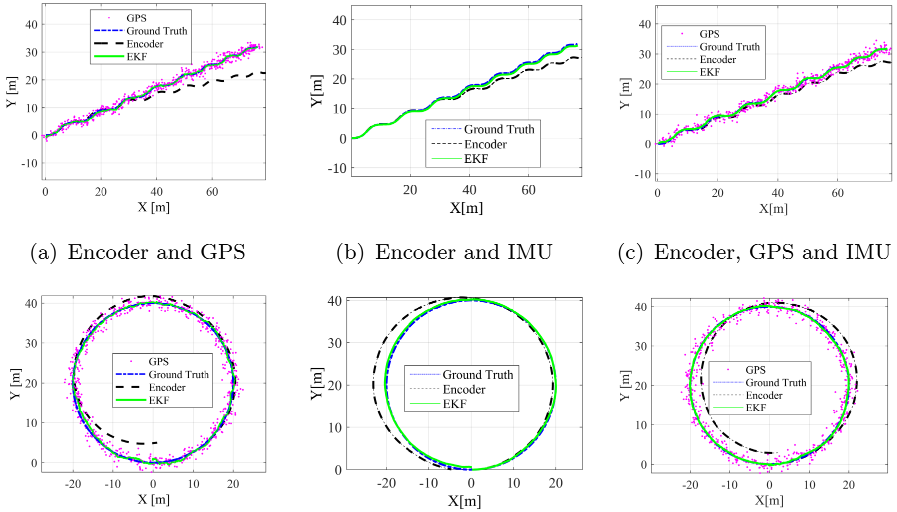
illustrate the statistical data analysis of the two conducted simulations.

$$ME = \frac{1}{n}\sum_{k=1}^{n} NE_k; \quad MSE = \frac{1}{n}\sum_{k=1}^{n}(NE_k - ME)^2 \tag{2.1}$$

where, n is the number of samples, NE is calculated in (2.2).

$$NE_k = \sqrt{(x_{ekf} - x_{true})^2 + (y_{ekf} - y_{true})^2} \tag{2.2}$$

In summary, the simulation results show that the goal of the proposed EKF-



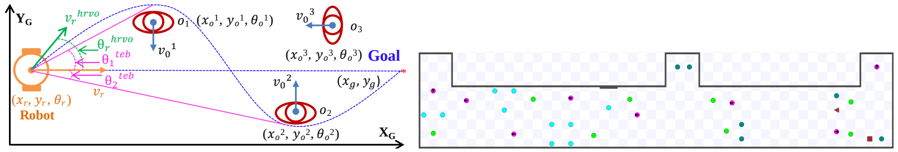
(a) Encoder and GPS (b) Encoder and IMU (c) Encoder, GPS and IMU

(d) Encoder and GPS (e) Encoder and IMU (f) Encoder, GPS and IMU

Figure 2.2: The circular and sinusoidal trajectories in three approaches.

Sensors	Mean Error		Mean Square Error	
	Sinusoidal trajectory	Circular trajectory	Sinusoidal trajectory	Circular trajectory
Encoder	37.4074	35.1793	6.1162	5.9312
Encoder + GPS	0.2303	0.2477	0.0289	0.0217
Encoder + Compass	0.1630	0.2032	0.0189	0.0243
Encoder + GPS + Compass	**0.1455**	**0.1533**	**0.0081**	**0.0115**

Figure 2.3: ME and MSE for the three approaches



(a) (b)

Figure 3.8: (a) The example scenario;(b) A hallway-like scenario with walls, objects, humans, and goals.

threshold value is 0.54) is applied to measure the physical safety of the robot and each individual obstacle.

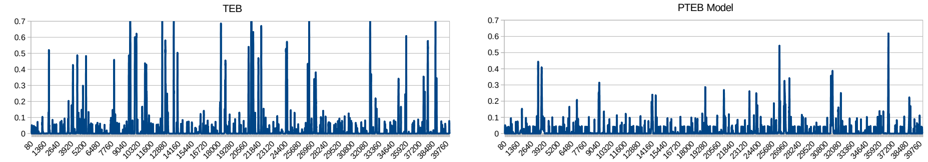


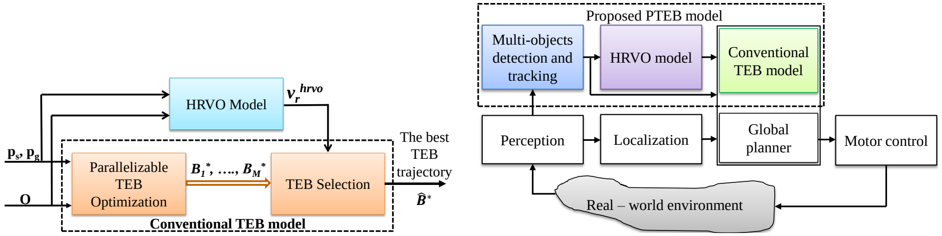
Figure 3.9: The simulation results of the two experiments.

As can be seen in Fig. 3.9(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 dynamic objects in the vicinity of the robot, and significantly reduces collision than that with the TEB algorithm, as shown in 3.9(a).

Although the proposed PTEB algorithm has been achieved consider successes. It still lacks of robustness in various environments, because it only incorporates the velocity obstacles-based potential collision. In order to deal with this weakness, in the next study, an extended timed elastic band (ETEB) algorithm, which takes into account future states of the surrounding obstacles, will be proposed.

3.3 Proposed extended timed elastic band algorithm

In this section, we propose an extended timed elastic band (ETEB) algorithm for the mobile robot navigation system using motion prediction algorithm. The motion prediction model utilizes the obstacle's states including position, orientation and velocity, to predict future positions of the surrounding obstacles. The main idea of the ETEB is incorporating both current and future states of the obstacles into the exploration step of the extensionTEB model

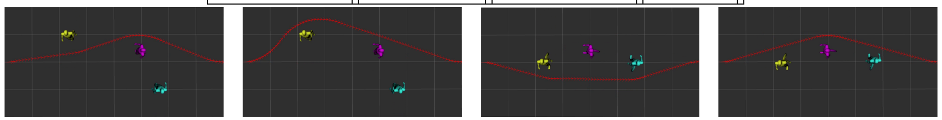


(a) (b)

Figure 3.6: (a) The flowchart of the PTEB algorithm; (b) The PTEB - based navigation framework.

Table 3.3: Parameters set in experiments - PTEB algorithm

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



(a) T2 [s] (b) T2 [s] (c) T3 [s] (d) T3 [s]

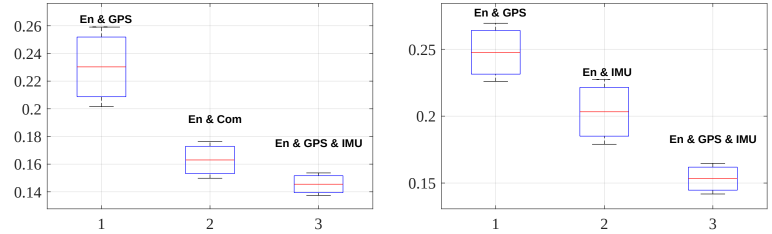
Figure 3.7: Snapshots at two timestamps of the two simulations.

crossing person, it illustrates that, the robot is able to proactively avoid people, as shown in Figs. 3.7(b) and 3.7(d). Because, the proposed PTEB algorithm takes into account the potential collision of the robot with the surrounding humans.

b. Simulation experiment in Stage environment

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. 3.6(b). We conducted two experiments in the simulated Stage environment (seen in Fig. 3.8(b)) to examine the effectiveness of proposed PTEB algorithm. Parameters of the system as well as of the objective function (3.12) are set up in Table 3.3.

We adopted the collision index (CI) proposed by Truong et al.[24] to quantitatively validate the proposed PTEB algorithm. Specifically, the CI value (the



(a) Sinusoidal trajectory (b) Circular trajectory

Figure 2.4: ME and MSE of three approaches in two scenarios.

based localization algorithm is using the sensor fusion method with the higher precision sensor, the lower the error. Thus, 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.

2.3 Particle filter-based localization algorithm

In the second case, 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 PF - based localization algorithm includes two steps.

Prediction: The step uses the previous state to predict the current state based on the system model (1.1). 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.

Update: Using a current sensor measurement to correct the predicted state. According to the measurement model, weights are assigned by likelihood response (in Algorithm 1). The measurement likelihood function computes the likelihood for each predicted particle based on the error norm (EN) between predicted measurement and actual measurement.

The proposed PF-based localization algorithm is summarized in Algorithm 1. To verify the usefulness of the PF-based localization algorithm, we created four scenarios: (a) robot receives [x, y, θ] in the entire trajectory, (b) robot gets [x, y], (c) the [θ] information is available, (d) all the sensor signals are lost, in the roofed area. 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

Algorithm 1: 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
      $\mathbf{x}_k^j$ using (1.1)
      //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
    $w_k^{[j]} = |2\pi R_k|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(EN^{[j]})^{0.4}\right\}$
    $\bar{\mathbf{S}}_k = \bar{\mathbf{S}}_k + \left[\mathbf{x}_k^{[j]}, w_k^{[j]}\right]$
  Normalize: $w_k^{[j]} = \frac{w_k^{[j]}}{\sum_{j=1}^{N} w_k^j}$
  Resampling using algorithm 4
  end for

= 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. 2.5. In the first part, the mobile robot navigates in a good environment condition, where 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

where, δ_{hrvo} is a predefined value. Using the new objective function (3.12), the result of solving (3.10), give the optimal trajectory (curved dashed line), as presented in Fig. 3.8(a). The proposed PTEB algorithm is presented more detail in Figure. 3.6(a) and Algorithm 3.

Algorithm 3: 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 $\mathbf{u}_r$
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($\mathbf{R}$, $\mathbf{G}$);
  for each trajectory $\mathbf{B}_p \in \mathbf{T}$ **do**
    $\mathbf{V} \leftarrow$ objectiveFunction(); $\triangleright$ using (3.13)
    $\mathbf{B}_p^* \leftarrow$ Optimizer($\mathbf{B}_p, \mathbf{O}, \mathbf{V}$);
    $\mathbf{B}^* \leftarrow$ storeLocalOptimalTrajectory($\mathbf{B}_p^*$);
  end for
  $\mathbf{v}_r^{hrvo} = [\upsilon_y, \upsilon_x]^T \leftarrow$ Run HRVO($\mathbf{s}_r$, $\mathbf{O}$)
  $\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x)$
  $\theta_p^{teb} = atan2(y_p^{teb} - y_r, x_p^{teb} - y_r)$
  $\Delta\theta^{teb} = min(|\theta_r^{hrvo} - \theta_p^{teb}|)$ with p= 1, 2, ..., M.
  $\mathbf{V}_c \leftarrow$ newObjectiveFunction(); $\triangleright$ using (3.11)
  $\hat{\mathbf{V}}_c(\mathbf{B}_p^*) = V_c(\mathbf{B}_p^*) + \delta_{hrvo}\Delta\theta^{teb}$ using (3.12)
  $\hat{\mathbf{B}}^* \leftarrow$ Call Optimizer($\mathbf{B}^*, \mathbf{O}, \hat{\mathbf{V}}_c$) $\triangleright$ Solve (3.10)
  $\mathbf{u}_r \leftarrow$ According to (Eq. 2.35 and 2.36) and $\hat{\mathbf{B}}^*$
  Return $\mathbf{u}_r = [\upsilon_r, \omega_r]^T$

3.2.2 Algorithm validation by simulations

The scenario in Fig 3.8(a) is utilized to describe and prove the efficiency of the proposed planning algorithm.

a. Simulation experiment in RViz Environment

Firstly examining the proposed PTEB algorithm in a simple simulation environment, and visualizing the results in RViz environment[3] with parameters set up in Table 3.3.

At the time stamps T2 and T3, the globally obtimal trajectory is generated in font of two crossing people, as shown in Figs. 3.7(a) and 3.7(c), in these cases, the mobile robot can safely avoid people but its behavior might not be smooth. In contrast, the globally optimal trajectory is generated behind the left

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

trajectory among the candidate trajectories of distinctive topologies. Therefore, it enables the robot to transit across obstacles. 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. Thus, such developed navigation systems lack robustness in diverse situations in the dynamic environments.

A proactive timed elastic band (PTEB) algorithm is proposed to overcome these shortcomings. The main idea of the proposed algorithm is to combine the advantages of the TEB technique and the HRVO model by incorporating the potential collision between the robots and the obstacles into the selection step of the extension TEB model [42].

3.2.1 *Construction of the PTEB algorithm*

The 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 of conventional extension TEB, one more factor using the orientation of the velocity vector generated by the HRVO model is added. The orientation θ_r^{hrvo} of the velocity vector $\mathbf{v}_r^{hrvo} = [\upsilon_x, \upsilon_y]^T$ generated by the HRVO model in (3.3) 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.

$$\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x) \tag{3.7}$$

$$\theta_p^{teb} = atan2(y_p^{teb} - y_r, x_p^{teb} - x_r) \tag{3.8}$$

$$\Delta\theta_p^{teb} = |\theta_r^{hrvo} - \theta_p^{teb}| \tag{3.9}$$

where, (x_p^{teb}, y_p^{teb}) is the coordinates of the node ζ_p, which is added beside the obstacles.

$$\hat{\mathbf{B}}^* = arg \min_{\mathbf{B}_p^* \in \{\mathbf{B}_1^*, \mathbf{B}_1^*, \ldots, \mathbf{B}_M^*\}} V_c(\mathbf{B}_p^*) \tag{3.10}$$

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

$$V_c(\mathbf{B}_p^*) = \mathbf{w}_c^T f_c(\mathbf{B}_p^*) \tag{3.11}$$

Finally, the new objective function of PTEB algorithm is obtained as follows:

$$\hat{\mathbf{V}}_c(\mathbf{B}_p^*) = V_c(\mathbf{B}_p^*) + \delta_{hrvo}\Delta\theta_p^{teb} \tag{3.12}$$

converges back to the actual pose.

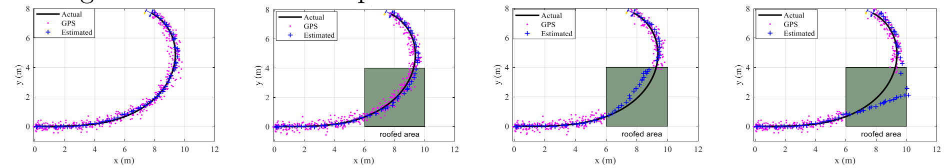


Figure 2.5: The simulation results using PF localization

Table 2.1: ME of robot's position.

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**
[]	0.9722	**0.3957**

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.

2.4 Conclusion

In this chapter, two efficient localization algorithms have been proposed including EKF-based and PF-based localization algorithm. The first case, when sensor signals are sufficient and noise distributions are Gaussian distribution, the EKF - based localization algorithm has been made of used. 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. 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. 3.1(a).

Chapter 3

DEVELOPING EFFICIENT MOTION PLANNING SYSTEMS

The motion planning systems include of two sub-systems, as shown in Fig. 3.1(a): (i) global planner (or path planning); (ii) local planner (or obstacle avoidance). We only focus on developing the local planning algorithms for the motion planning systems which are capable of driving the mobile robots to proactively and safely avoid dynamic obstacles in the real-world environments. To accomplish that motion planning systems should take into account robot's kinodynamic constraints, the potential collisions of the robots with surrounding obstacles and obstacle's future states as well as future trajectory of the obstacles in their vicinity. Three new local planning algorithms of the motion planing system for the mobile robots 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, is also introduced.

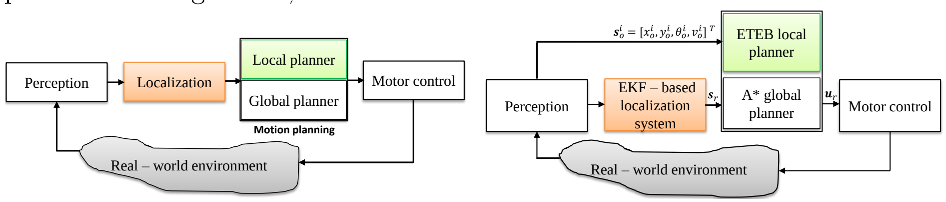


Figure 3.1: (a) The conventional navigation framework; (b)The proposed integrated navigation system.

3.1 Proposed enhanced dynamic window approach - based algorithm

Various navigation systems have been proposed to ensure the safe navigation of the mobile robot in dynamic 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

Table 3.2: The average passing velocity [m/s] of the robot

Robot and Obstacles	Seario 1	Seario 2	Seario 3	Seario 4
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**

introduced in [1] is proposed to accomplish this, as shown in Fig. 3.2(b). The proposed system consists of two major parts: (i) the conventional navigation scheme, and (ii) the extended part. The proposed EDWA algorithm has been installed on the mobile robot platform with data flow diagram, á shown in Fig. 3.13(a). Four experiments in a laboratory-like environment are then conducted. In this study, using humans as moving obstacles in all experiments is made. The experimental results of the four experiments are shown in the second row in Fig. 3.5 and the first row shows the snapshot of the scenarios. A video with our experimental results can be found at the hyperlink[2].

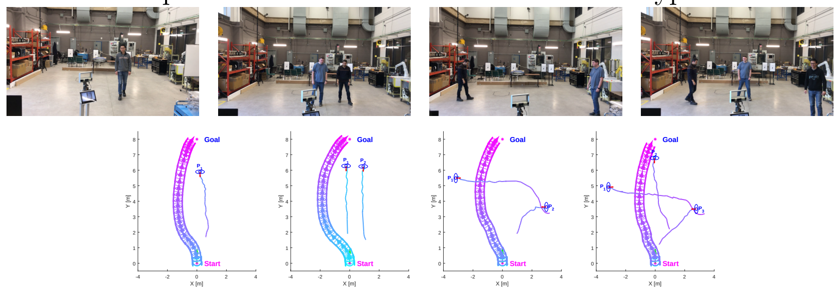


Figure 3.5: The experimental results of four experiments.

Overall, the simulation and experimental results shown illustrate that, the 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. However, the robot equipped the EDWA algorithm sometimes gets stuck in a locally optimal trajectory and unable to transit across obstacles if they are very close to it.

3.2 Proposed proactive timed elastic band algorithm

Recently Rosmann et al. [54] proposed extensions of the TEB technique by using parallel trajectory planning in spatially distinctive topologies. Using this technique, the mobile robots can switch to the current globally optimal

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

closer the value of $\delta_{min}(t)$ to 1 is. The simulation results shown in Fig. 3.3,

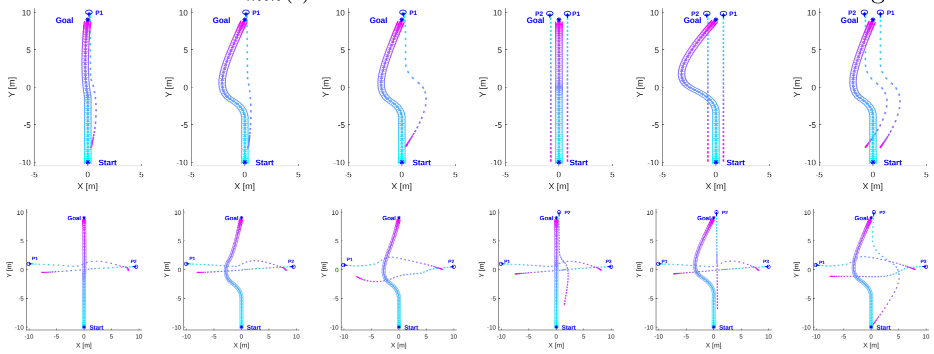


Figure 3.3: Trajectories of the robot and obstacles in four Scenarios.

Table 3.1: Parameters set in experiments - EDWA algorithm

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	$0.25[s]$

Fig. 3.4 and Table 3.2 (a video clip of our simulation results can be found at this link[1]) 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.

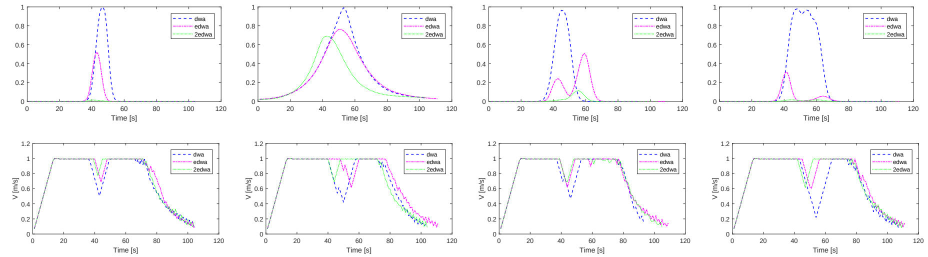


Figure 3.4: (First row) The minimum passing distance; (Second row) Robot velocity along the robot's trajectory.

b. Experimental setup and results

An extended navigation scheme based on the conventional navigation scheme

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

techniques. In the first group, the navigation systems take into account the robot dynamics, including actual speed, acceleration and physical limits. However, these methods only incorporate current positions of obstacles, so it does not proactively deal with potential collisions. Whereas in the second group, the navigation systems have a big advantage of proactive collision avoidance by incorporating both the current position and velocity of the obstacles. Thus, the robot is able to avoid the potential collision with the surrounding obstacles. Nevertheless, the systems does not consider robot dynamics. Thus, it is difficult to directly use this velocity to control the mobile robot in real-world environments. In order to overcome the mentioned drawbacks, an EDWA algorithm is proposed. The main idea of the EDWA algorithm is to combine the advantages of the DWA technique and the HRVO model, which are typical techiques in the two aforementioned groups.

3.1.1 Construction of the EDWA algorithm

The EDWA algorithm takes into account both the robot dynamics and its potential collision with the surrounding obstacles. To accomplish this, in the objective function (3.1) of conventional DWA model, the target heading function $head(v, \omega)$ is modified.

$$G(v, \omega) = \alpha head(v, \omega) + \beta dist(v, \omega) + \gamma vel(v, \omega) \tag{3.1}$$

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

$$head(v, \omega) = 180^o - |\theta^{goal} - \theta_r| \tag{3.2}$$

where, θ^{goal} is the orientation of the vector pointing from the predicted position of the robot to the goal

$$\mathbf{v}_r^{hrvo} = arg \min_{\mathbf{v} \notin HRVO_r} \|\mathbf{v} - \mathbf{v}_r^{pref}\|_2 \tag{3.3}$$

Particularly, in (3.2) 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 $\mathbf{v}_r^{hrvo} = [v_x, v_y]^T$ generated by the HRVO model in (3.3) is used to compute the new target heading function as follows:

$$head^{hrvo}(v, \omega) = 180^o - |\theta^{goal} - \theta_r^{hrvo}| \tag{3.4}$$

$$\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x) \tag{3.5}$$

Finally, the objective function of the DWA model in (3.1) is replaced by the new objective function as follows:

$$G'(\upsilon,\omega) = \alpha head^{hrvo}(\upsilon,\omega) + \beta dist(\upsilon,\omega) + \gamma vel(\upsilon,\omega) \tag{3.6}$$

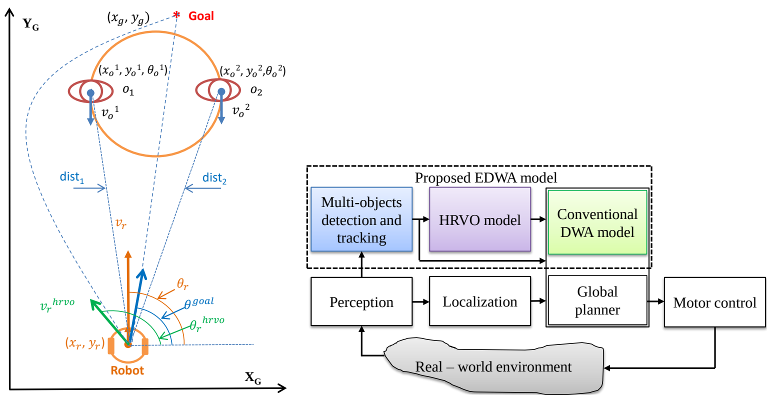
The proposed EDWA algorithm consists of three steps including: (i) calculate the search space of the velocities V_r , (ii) compute the orientation of the velocity vector generated by the HRVO model, and (iii) select the efficient velocity control command. Using the proposed EDWA algorithm is to generate an efficient velocity command $\mathbf{u}_r$=$[\upsilon_r,\ \omega_r]^T$. Then, to generate directly control signals for the motor control model (υ_r^r, υ_r^l) which are the linear velocity commands of the right and left wheels of the robot, respectively.

Algorithm 2: 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 $\mathbf{u} = [\upsilon_r, \omega_r]^T$
begin
  Initialize parameter set α, β, γ
  Set motion dynamic $\upsilon_{max}, \omega_{max}, \dot{\upsilon}_{max}, \dot{\omega}_{max}$
  Compute V_s = possible velocities
  Compute V_a = admissible velocities
  Compute V_d = reachable velocities
  Compute $V_r = V_s \cap V_a \cap V_d$
  Run HRVO to generate $\mathbf{v}_r^{hrvo} = [\upsilon_y, \upsilon_x]^T$
  Compute $\theta_r^{hrvo} = atan2(\upsilon_y, \upsilon_x)$
  for each pair of velocity $(\upsilon_i, \omega_i) \in V_r$ **do**
    Predict robot position (x_i, y_i) using (1.1)
    $\theta_i^{goal} = atan2(y_g - y_i, x_g - x_i)$
    $head_i^{hrvo} = 180^o - |\theta_i^{goal} - \theta_r^{hrvo}|$
    Compute obstacle clearance function $dist_i$ using the closest distance to obstacles
    Compute velocity function $vel_i = |\upsilon_i|$
    Compute the $score_i$ using (3.6)
    Store $score_i$ in the score vector **S**
  end for
  Select $\mathbf{u} = [\upsilon_r, \omega_r]^T$ using maximum score from **S**

3.1.2 *Algorithm validation by simulations and experiments*

To verify effectiveness of the proposed algorithm, we created a scenario, as shown Fig. 3.2(a). Assuming that the robot state is $\mathbf{s}_r = [x_r, y_r, \theta_r, \upsilon_r, \omega_r]^T$. The robot's goal position is $\mathbf{p}_g = [x_g, y_g]^T$. There are N obstacles appearing



(a) (b)

Figure 3.2: (a) The example scenario; (b) The efficient navigation system based on the EDWA algorithm

in the vicinity of the robot $O = \{\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_N\}$. The state of the obstacle $\mathbf{o}_i$ is $\mathbf{s}_o^i = [x_o^i, y_o^i, \theta_o^i, \upsilon_o^i]^T$. The radius of the robot and obstacle are r_o and r_r, respectively. Robot is requested to navigate safely from initial point to given goal point.

a. Simulation setup and results

Four typical scenarios have been created. In each scenario, three experiments corresponding to three pairs of reactive motion planning algorithms (DWA-DWA; EDWA-DWA; EDWA-EDWA)to compare the proposed EDWA algorithm with the conventional DWA algorithm are conducted. In order to compare the proposed EDWA algorithm and the conventional DWA algorithm, we made 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, as shown in Fig. 3.3. Whereas, in term of quantitative evaluation, we utilize three matrices, as shown in Fig. 3.4 and Table 3.2. The velocity and average velocity are used to indicate the proactive robot trajectory, while the minimum distance from the robot to the surrounding obstacles illustrates the safe navigation of the mobile robot. Note that the minimum distance is normalized as follows: $\delta_{min}(t) = e^{(-\frac{d_{min}(t)^2}{3})}$ where, $d_{min}(t)$ is 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