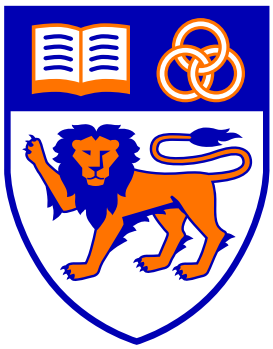


NUS

National University
of Singapore

DYNAMIC PATH PLANNING OF MULTIPLE MOBILE ROBOTS

LIU, Xin

(B.Eng, M.Eng)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2006

Acknowledgements

First of all, I would like to express sincere appreciation to my supervisors Dr. Prahlad Vadakkepat and Prof. Lee Tong Heng for their valuable guidance and constant encouragement in the course of my research study. This thesis would never have come out without their expert guidance and enthusiastic help. Working with them has been a very rewarding and pleasurable experience that has greatly benefited my education.

I would like to thank Dr. Tan Kay Chen, Dr. Abdullah Al Mamun, Dr. Ge Shu Zhi and Dr. Xu Jian Xin for their kind help and suggestions in my research work. Especially, I would like to thank Mr. Jason Chan Kit Wai, Dr. Wang Zhuping, Dr. Xiao Peng and Ms. Liu Jing for the valuable discussions with them.

I am also grateful to all the members of the Mechatronics & Automation Laboratory, Department of Electronical & Computer Engineering, National University of Singapore, for providing the research facilities for my study and for making a pleasant and friendly environment for my campus life.

Acknowledgement is extended to National University of Singapore for giving me the opportunity to pursue my PhD study and to do the research work with university facilities.

Finally, I dedicate this thesis to my parents, my sister and lovely Yifan, who have given me the unerring love and continuous supports through all these years.

Contents

Summary

The main aim of the thesis is to develop dynamic path planning methods for mobile robots in dynamic environments. This research consists of multi-agents mobile robot system construction and online path planning methods for mobile wheeled robot.

A multiple mobile robotic system, Robot Soccer System, is constructed. The behavior hierarchy of robot strategies, formations and actions, successfully organize a robot team to coordinate. The kinematic and dynamic models of the nonholonomic mobile robot are studied. A tracking controller is designed based on the models and the models are validated through simulation and experiments.

Path planning is one of the main issues associated with mobile robots. An artificial potential field (APF) based approach is presented to navigate the multiple robots while avoiding obstacles in a dynamic environment. It is observed that the APF approach is a simple and flexible method for path planning. Another potential field approach, electrostatic potential field (EPF) is studied and its effectiveness is verified.

In order to improve the performance, multi-objectives evolutionary algorithm (MOEA) tools are applied to optimize the APF parameters during the potential construction, providing sub-optimal solutions with multiple objectives. The local minima problem in APF is also tackled with a heuristic method in which an escape force is designed to push the robot out of the local minimal positions.

Effective prediction of the positions of the moving objects paves the way for

effective motion planning. Particle filter is utilized to predict the position of the mobile robot which in turn is combined with the APF algorithm to plan the motion of the robots.

Finally, conclusions about the research are drawn, and suggestion for further research are presented.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Since its inception, robots have been regarded as human assistants or even replacement to some extent. The last century has seen successful applications of classical control algorithms and the robots being utilized extensively in industries, military and space exploration [1][2][3].

With the rapid development of computing facilities in recent years, the performance of robotic systems has dramatically improved by using high speed computers and advanced control algorithms. Robotic systems play more and more important roles not only in the labor-lack situations but also in the entertainment world, especially by the mobile robots. Path planning is one of the central issues in mobile robot research. The path-planning problem is to identify a collision free path from the current robot position to a destination point, satisfying certain constrains such as smoothness in motion, minimum path length, etc. Path planner has a significant part in mobile robot control research and the algorithms should be capable of providing fast adaptive control in dynamic environments.

In this thesis, a multiple mobile robot system (Robot Soccer System) is studied and multiple mobile robot navigation algorithms are proposed which are verified

through hardware implementation.

1.2 Previous Work

The conventional definition of the robot is a mechanical device with perception module to collect environmental information, actuation module and a control system processing the information and providing appropriate instructions [4][5][6].

No matter where the robots are used, whether in factory locations for non-trivial tasks and hazardous environments such as mining, nuclear power station, tunnelling or fire fighting, one of the major problems system designers face is the controller design.

Since the robot actions could be decomposed into behaviors, Behavior-based robotics obtained wide acceptance [7]. The behaviors are defined according to the features of the robotic system. There are many novel approaches in various applications, especially in simulation experiments [8][9][10][11].

1.2.1 Mobile Robot Path Planning

Path planning is the central issue in mobile robotic systems and algorithms for mobile robot path planning have been intensively researched for years. The path planner is required to find a trajectory that allows the robot to navigate from the given starting Point A to the destination Point B with a safe distance from obstacles in the environment.

The main approaches for collision-free and deadlock-free paths include: road map approaches, cell decomposition approaches, artificial potential field approaches and neural network models. The roadmap approach is mostly used to design a collection of path segments to avoid the indoors obstacles [12]. Visibility graphs and Voronoi diagrams [13] are commonly used to build the paths from the initial to target configuration. Cell decomposition approaches decompose the obstacle free

space into regions, called cells or grids, and connect the appropriate successive cells into a path for mobile robots [14][15].

Artificial potential field (APF) approaches navigate the robots by the artificial potential forces constructed virtually by simulating the natural potential fields. APF was first proposed in [16] and applied later in path planning [17][3]. In the construction of APF, heuristic methods are also utilized [18]. The local minimal problem is the main shortcoming of APF [19]. Researchers have developed different methods to overcome the local minima [20][21][22]. APF has been used widely in mobile robot path planning [23][24][25][26].

Neural networks are also used to generate robot paths through training and learning. In [27] a generalized predictive control method based on self-recurrent wavelet neural network (NN) trained with the adaptive learning rates is proposed for stable path tracking of mobile robot. In [28] a robust adaptive controller is designed with adaptive neural networks. An adaptive fuzzy logic system [29] is used to estimate the uncertainty of environment in wheeled mobile robot control. The real time control is obtained by online tuning of the parameters of fuzzy logic system. In [30] hierarchical fuzzy control is designed for autonomous navigation of wheeled robots where the controller is decomposed into three fuzzy subsystems, fuzzy steering, fuzzy linear velocity control and fuzzy angular velocity control where each rule is constructed manually. Furthermore, the coupling effect between linear and angular motion dynamics is considered in fuzzy steering by appropriate rules.

Meanwhile the research on non-holonomic robot model has attracted wide attention due to the fact that mobile robots always have motion constrains [31][32][33]. An appropriate model of the robot is a significant element to design a precise controller. The kinematic model of the system alone is insufficient to describe the system behavior [34][35]. The generalized non-holonomic kinematic and dynamic models are specified in individual application cases. In [36] the dynamic model of a wheeled inverted pendulum is analyzed from a controllability and feedback linearizability points of view. A sliding-mode control method is proposed for mobile

robots with kinematics in 2-D polar coordinates [37][38]. Some methods design the steering controller directly from the spectral information [39][40].

Evolutionary computing, fuzzy computing and neurocomputing are catalogued into Computational Intelligence, or soft computing. The soft computing techniques, artificial neural networks (ANN), fuzzy logic (FL) and evolutionary algorithms (EA), have been combined with robot control designs [41][42][43][44]. ANN and FL act as identifiers in various areas [45], while EA shows its advantages in system parameter optimization [46].

Tracking problem is one of the typical navigation problems, which has been studied extensively in recent years [47][48][49][32]. In [28] wheeled robot tracking controller is designed by adaptive neural networks, while in [50] Fuzzy logic is used to design the robot controller.

Trajectory prediction is closely connected with trajectory tracking which has been widely studied. In this work the particle filter is used to predict robot motion. Particle filtering, a sequential important sampling algorithm, is widely used in Bayesian tracking recursions for general nonlinear and non-Gaussian models [51]. In particle filtering, the target distribution is represented by a set of samples, called particles, with associated importance weights which are propagated through time. The target trajectory prediction is to estimate the state of the target of interest at the current time and at a point in future.

Particle filters have been applied successfully in various state estimation problems [52][53]. Improved particle filter (IPF) is successfully applied in randomly moving object tracking [49].

1.2.2 Evolutionary Algorithms

Evolutionary Algorithm (EA) [46][54][55] is a term used to describe a catalogue of algorithms which are inspired by biological evolutionary processes in nature.

The major EAs are: Genetic Algorithms, Evolutionary Programming, Evolutionary Strategies, Classifier Systems, and Genetic Programming. In these algorithms the evolution procedures of species (selection, mutation, and reproduction), are simulated in computational models to solve optimization problems in complicated search space.

The main applications of EAs in robotic systems are along model structure or parameter optimization. The optimization problems on mobile robots could be path planning problems, trajectory planning problems and task planning problems.

In [56] an algorithm based on EA is utilized to learn safe navigation in multiple robot systems. The robots shared information to speed up the learning process. As well defined artificial potential could be integrated with EA for fast and efficient trajectory searching mechanism [57]. Differential Evolution and Genetic Algorithms are applied for the optimum design of fuzzy controllers for mobile robot trajectory tracking [58]. Moreover, EA is programmed into the onboard software to learn dynamic gaits of the entertainment robot AIBO by Sony [59].

1.2.3 Multi-Objective Evolutionary Algorithms

Many real world problems involve multiple measures of performance, or objectives, which should be optimized simultaneously [60]. In certain cases, objective functions may be optimized separately. However, suitable solutions to the overall problem can seldom be found in this way. Optimal performance according to one objective often implies unacceptable low performance in one or more of the other objective dimensions, creating the need for a compromise to be reached. EAs have been recognized to be possibly well-suited to multi-objective optimization since early in their development. It is possible to search for multiple solutions in parallel, eventually taking advantage of any similarities available in the family of possible solutions to the problem. Multiple Objective Evolutionary Algorithm has been proposed for multi-objective optimization problems [61][62][63]. In [64] another multi-objective combinatorial optimization algorithm other than MOEA was proposed to improve

the global searching ability while maintaining the parallel computing ability.

There are several approaches in MOEA : Plain aggregating approaches, population-based non-Pareto approaches, and Pareto-based approaches.

Plain aggregating approaches Optimize a combination of the objectives with the advantage of producing a single compromise solution. In population-based non-Pareto approaches each objective is effectively weighted proportionally to the size of each sub-population and, more importantly, proportional to the inverse of the average fitness (in terms of that objective) of the whole population at each generation. Pareto-based fitness assignment is a means of assigning equal probability of reproduction to all non-dominated individuals in the population.

1.3 Work in the Thesis

A robot soccer system (RSS) is used in this work to test the algorithms. In Chapter 2 RSS is studied. The RSS integrates robotics, intelligent control and computer technology. In the system robots moving inside a wooden field are controlled via RF commands from a host computer. The information about the environment is conceived by an overhead CCD camera. The mobile robots are $7.5cm$ cubic in size and are capable of locomotion on a surface through the actuation of wheel assemblies mounted on the robot and in contact with the surface. In Chapter 3, the kinematic and dynamic model of the soccer robot are analyzed for further application in controller design.

In Chapter 4, an Adaptive Window based Electrostatic Potential Field (AW-EPF) is proposed to bring down the computational time and to improve the real time performance of the EPF with simple steps before solving for the maximal current path. In the proposed AW-EPF, an effective window area is set according to the current positions of the robot and target, and the obstacles that are in the immediate vicinity are identified. The electrical potential is calculated with respect to the effective window to determine a nearly optimal direction for the robot's

next travel log. The proposed approach is able to generate a shorter path. The proposed approach, also partially solved a problem that the empty space between two obstacles cannot be passed through even if the space is large enough for the robot.

In Chapter 5, an APF based on Evolutionary optimization (EAPF) is built to provide the guide forces to the robot avoiding collisions. The environment data is converted to steering commands and the robot reacts directly by small time expense without decision making. The workspace of robot soccer system is placid and continuous with fixed bounds, and APF approach can be applied for path-planning. EAPF is applied in a robot soccer system where the environment changes dynamically. The input of the EAPF controller is the potential gradient instead of the potential value and hence the involved computation is simple. Several parameters are introduced to construct the artificial attractive and repulsive forces. As path smoothness, safety and path length play roles in the evaluation of the planned path, a multi-objective optimization algorithm is utilized to search for sub-optimal solutions. With the help of MOEA, the proposed EAPF is implemented on a robot soccer system.

In Chapter 6, the particle filter workframe is discussed and used in the mobile robot trajectory prediction. Combing the prediction algorithm with the mobile robot system management and path planning modules, the robot is able to chase the target on a better scale.

Finally in Chapter 7, conclusions and suggestions on further research are presented.

Chapter 2

Multiple Mobile Robotic System

Research in mobile robots has reached a level of maturity where robotic systems can be expected to efficiently perform complex missions in real-world, and capable teams of cooperative mobile robots could provide a valuable service in risk-intensive environments. Through the distribution of computation, perception, and action, a multiple robot team is more powerful [65].

Multiple mobile robot systems are more capable than a single robot in real-world applications, for the reason that complicated missions with interdependencies between the robots become feasible.

The issues associated with multiple mobile robot systems include motion planning, mission planning, and distributed tasks cooperation [66] [67] [68].

Path planning is one of the fundamental problems in mobile robots. In the context of autonomous robots, path planning techniques are required to simultaneously solve two complementary tasks: minimize the length of the trajectory from the starting position to the target position, and maximize the distance to obstacle in order to minimize the risk of collision. The problem becomes harder in multiple robot systems, since the size of state space of the robots grows exponentially with the number of robots [12]. There are two categories of methods for multiple robots motion planning: centralized approach in which the configuration spaces of the

individual robots are combined into one composite configuration space and then a path is searched in the whole composite system, and the decoupled approach in which the individual robot paths are determined and further possible collisions are resolved.

There are different techniques that have been used in dynamic path planning. In [69] a probabilistic model is used to estimate the risk of collision in a typical office environment. In [70] an augmented Lagrangian decomposition and coordination technique based distributed route planning method is applied to minimize the total transportation time without collision among automated guided vehicles in semiconductor fabrication bays. To avoid conflicts, reactive navigation by collaborative resolution of multiple moving agents is proposed as a cooperative scheme associated with real time robot parameters [71].

It is also considered to plan motion of robots one by one according to their priorities in the system [65]. Complex trajectory planning problem is transformed into path planning and velocity planning to reduce the complexity [72][73].

Formation methods of multiple mobile robot systems have been reported in terms of cooperation. The first method is Behavior-Based Strategy [74]. This approach places weightings on certain actions for each robot and the group dynamics emerge. The advantage of this strategy is that the group dynamics contain formation feedback by coupling the weightings of the actions. The second one is Multi-Agent System Strategy which applies a game theoretical approach to the design of closed-loop feedback laws [75][76][77] . Virtual Structure Strategy presents a control scheme for improving multiple mobile robots in formation [78]. The advantage of this strategy is that it makes it easy to prescribe formation strategy, with guaranteed stability, and to add robustness to the formation through the use of group dynamics. The disadvantage of both strategies is the difficulty in controlling mobile robots in formation with a decentralized system. Another one is Leader-Following strategy [79][80]. The advantages of this strategy is that it is easy to control multiple robots in a desired formation using only two, controllers

and it is suitable for describing the formation of robots.

2.1 Robot Soccer System Overview

Robot Soccer System (RSS) is an intriguing multiple mobile robot system for research and entertainment by providing a platform for distributed intelligence algorithms as well as for competition. The idea of robotic soccer was published in early 1990's [81], and the Robot World Cup Initiative (RoboCup) [82] and The Federation of International Robot-soccer Association (FIRA) [83] were estalished in mid of 1990's as major robot soccer league organizations. Robot soccer covers many research topics such as mobile robot control, communication, image processing and mechatronics.

The MiroSot system consists of mobile robots, a radio transceiver, a host computer and a CCD camera (Figure 2.2) [84] [85] [86].

The aim of Robot Soccer Games is to inculcate in the general public an understanding and appreciation of robotics and automation; to educate the general public on the things robots can do that are quite apart from industrial tasks; to help in the technology development by providing benchmarks for practical robotics research and development.

The target of the robot soccer system is to build a team of robots to play 3-a-side (or more robots in a team) football against an opponent robot team. Each robot soccer team shall setup a global vision system, which is above the football field, to keep track of their robots' and the ball positions. A host computer processes the vision information and sends the motion commands to soccer robots through radio frequency communication. The robot soccer designers have to take up the challenges such as to identify their own robots, the ball, and the opponent robots through the vision information, and to establish a reliable protocol for the radio frequency communication. They also need to implement various strategies among the team robots for attacking and defending, and to manage the fouls that comprise

of free ball, penalty kick, goal kick, and free kick.

The soccer robot has driving mechanism, communication parts, and computational parts for velocity control and for processing the data received from the host computer.

Robot Soccer System is an example of distributed robotic systems, which consists of multiple robotic agents whose tasks are distributed. In a distributed system, the agents may be robots, modules, computers, processors or sensors; for the distributed characters, they could be multi-robot, distributed sensing, distributed planning or control, cooperative control or shared autonomy [87]. Problems in RSS include motion plan, path planning, cooperation strategies and so on. According to the tasks and construction of RSS, a top-down analytic behavior based approach is used to design the control software.

The robot soccer system in this work belongs to small league Micro-Robot Soccer Tournament (MiroSot) of FIRA (Figure 2.1). Organized by FIRA, various scales of MiroSot soccer competitions are held annually in different countries. In the small league MiroSot rules, two teams of three robots each, start to goal against the other team during two sessions of game time. The soccer field is black colored wooden platform of 1.5m $\times$ 1.3m, and the ball color is orange. Once the game starts, no human intervention is allowed until the referee's whistle. MiroSot robots are homogeneous because they share the same size, shape, and hardware structure.

The overall system structure is shown in Figure 2.3. In each control loop, the camera captures the image of the field and sends the analogue frame signals to the computer; the image signals are then converted into digital ones by the capture card and processed by vision module of the system software. Information about the robots is processed by the vision processing and becomes the input of the system control module. After the behavior management and trajectory planning, the commands to each robot are transmitted by the Radio communication module.

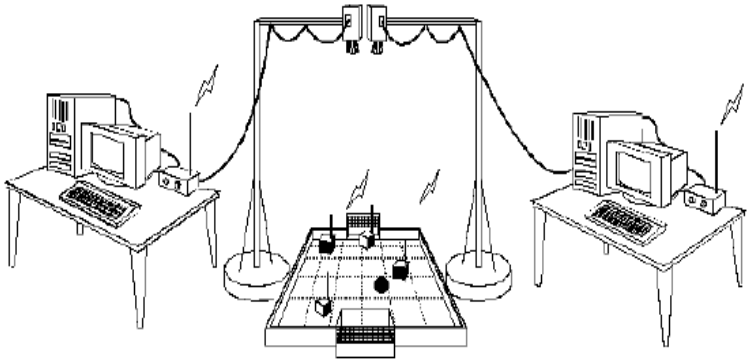


Figure 2.1: Micro-Robot Soccer System (MiroSot)

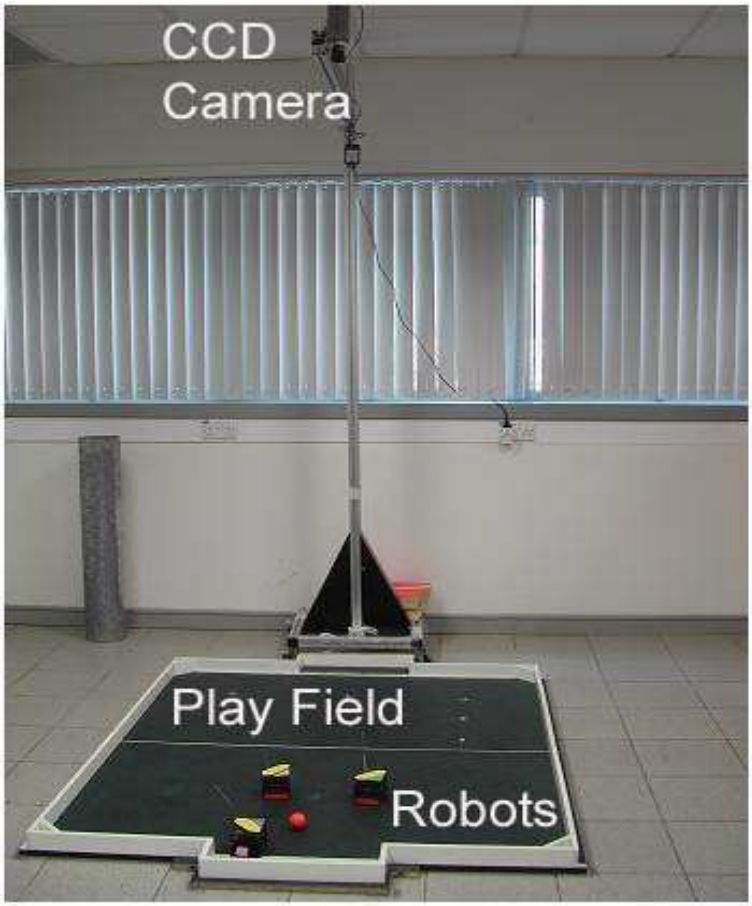


Figure 2.2: Real Robot Soccer System

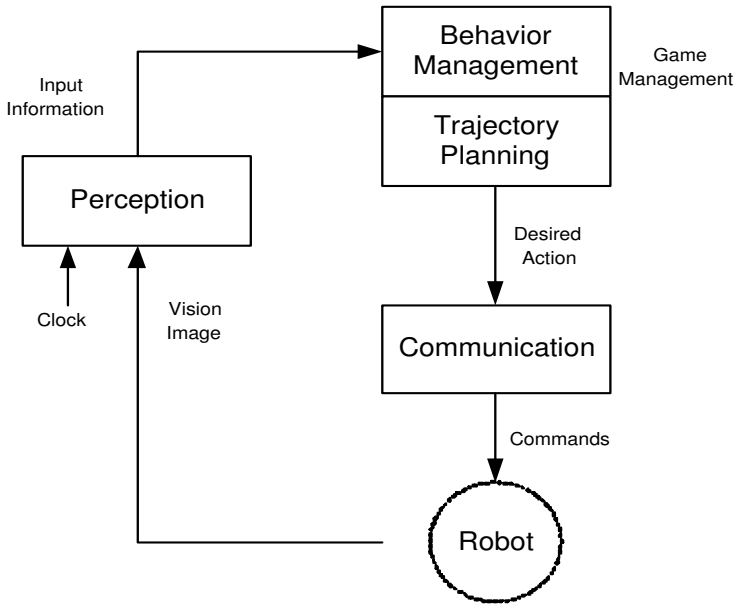


Figure 2.3: Robot Soccer System overall structure

2.2 Mobile Robot Hardware

The mobile robots in a Micro Robot Soccer System are capable of moving on a surface through the actuation of wheel assemblies mounted on the robot and in contact with the surface. It is assumed there is no slip between the wheel and surface. The wheel assembly provides or allows motion between its mount and surface on which it is intended to have a single point of rolling contact. Here bi-wheel type robot with independent motor control is utilized for robot soccer for smooth motion. The robot appearance is shown in Figure 2.4 and the hardware structure in Figure 2.7. In this work the host computer is a DELL GX260 (CPU 2.4GHz) with Windows/2000 platform and a Samsung CCD camera is used.

The robot developed in NUS is powered by a 7.2v battery and is embedded with communication module, microprocessor, and power control unit (Figure 2.5). The robot is symmetrical with a size of $7.5cm$ cubic and has a low center of gravity. Low center of gravity results in high mobility in robot movement.

A Micro-controller chip (PIC16F67X) with flash memory, data memory and

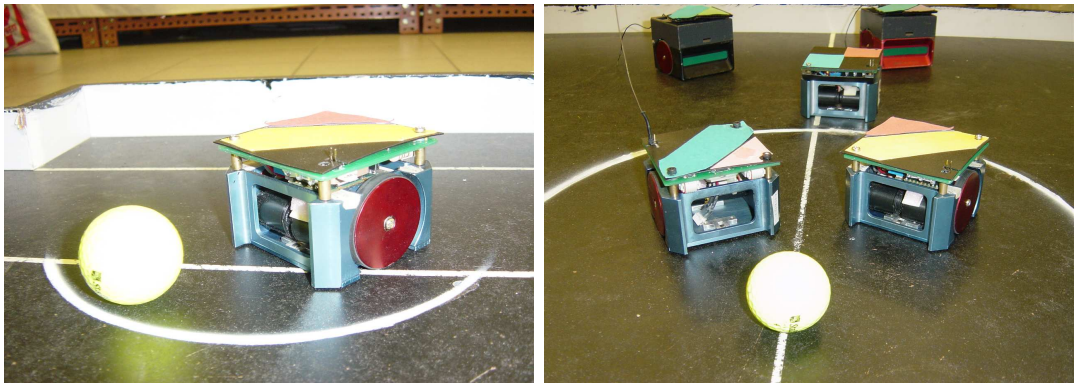


Figure 2.4: Mobile Robots

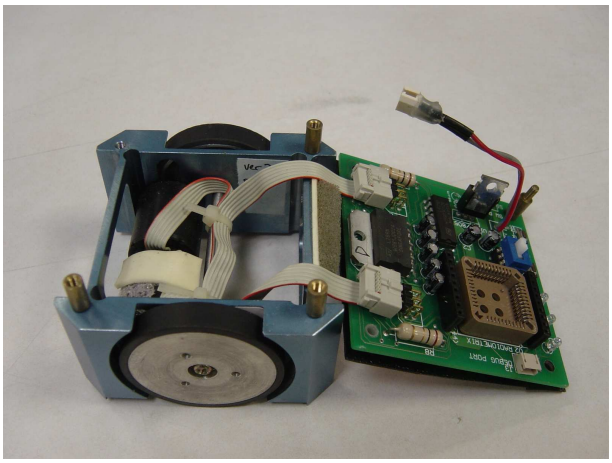


Figure 2.5: Hardware construction

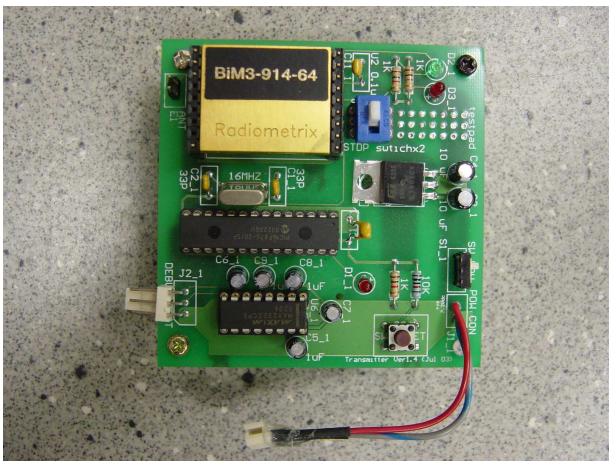


Figure 2.6: Radio transmitter circuit

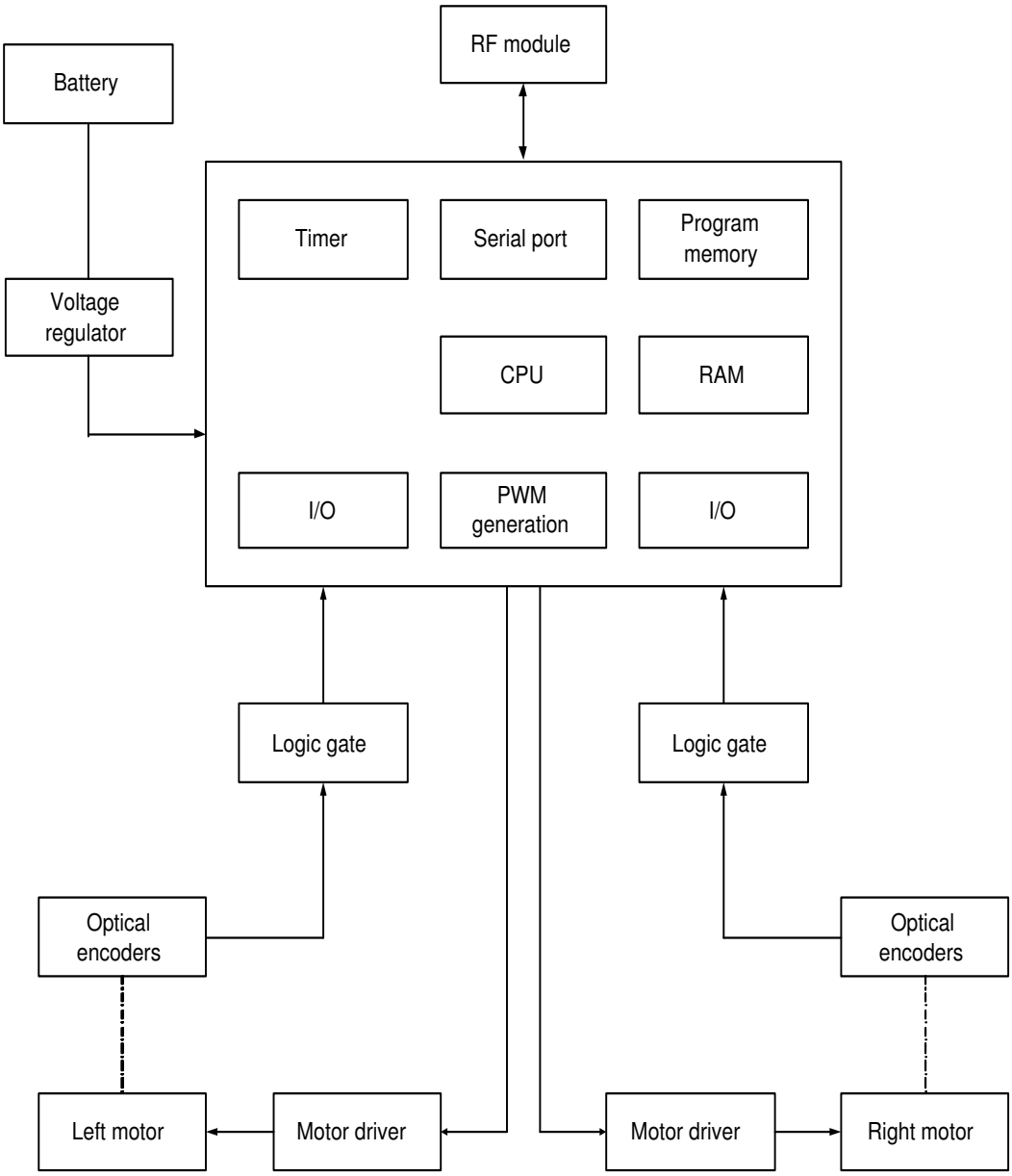


Figure 2.7: Robot hardware structure

EEPROM is selected to process the RF signal and generate Pulse Width Modulation (PWM) motor controls.

In RSS, communication is important to connect the host computer and robots. The prime mode of communication is sending data from the host to the robot. A transceiver receives the encoded data via serial port and broadcasts them to the robots. The data is decoded into wheel controls by individual robots. To improve the interaction ability, the robot communication module is bi-directional. The transceiver module is a single frequency device and can be configured as a transmitter or a receiver using two I/O ports from the micro-controller. Time-division multiplexing (TDMA) is applied as delivery mode in RF. Control data for each robot are combined into one string and each robot reads the allocated bytes from the received string. A common protocol is applied to handle the communication system, which consists of 21 bytes in each protocol string including preamble bytes, lock byte, header, data, tail and checksum. Preamble bytes is to make sure that a reliable data is received.

2.3 System Software

Robotic soccer is motivated by the human soccer competition and encouraged by the inherent research potential on a wide range of topics, hence it is natural to design the system software based on system behaviors. Behavior based robot systems decompose complicated behaviors into layered simple behaviors [7]. The flexible architecture of behaviors enables the system to improve system capabilities by incorporation of new subsets of behaviors.

RSS is a multiagent robot system in which the robots share a common goal with teammates against adversaries [88]. In a multiagent system, the behavior architecture provides a framework for each agent to communicate with environment, make decisions, and decompose the tasks. The entire structure should be in mind at the time of creating the multiagent system. On the other hand, the structure

should be flexible for function changes and improvement.

RSS is a time-critical environment where the robots are assigned roles according to the latest updated information. The process structure of the whole software is shown in Figure 2.8. The game control interface is shown in Figure 2.9, which is programmed in Visual C++. There are three functional modules in the whole system:

- Perception module. It captures the image in vision memory and identifies the objects on the playground.
- Game management module. It takes charge of behavior management and trajectory planner.
- Communication module. It generates and transfers the commands via serial port.

As the center of the software, game management architecture is illustrated in Figure 2.10. From the information obtained through vision processing, the game management module reads the current game (robot positions), decides the actions for each robot, and sends the commands to the communication module. There are three catalogues of behaviors designed in the control software to decompose the complicated team tasks:

- Supervisory level behaviors: Event and Stragety supervision. Event supervision deals with the game events, such as game kick off, game over, free ball and penalty, then sets the system inner status. Base on the robot position information, Strategy supervision decides the team attitude, either attack or defend.
- Basic robot behaviors: Robot basic behaviors are predefined movements with direct targets (Table 2.3).
- Supplementary behaviors: These behaviors are relatively independent with the public functions like path planning algorithms plug-in.

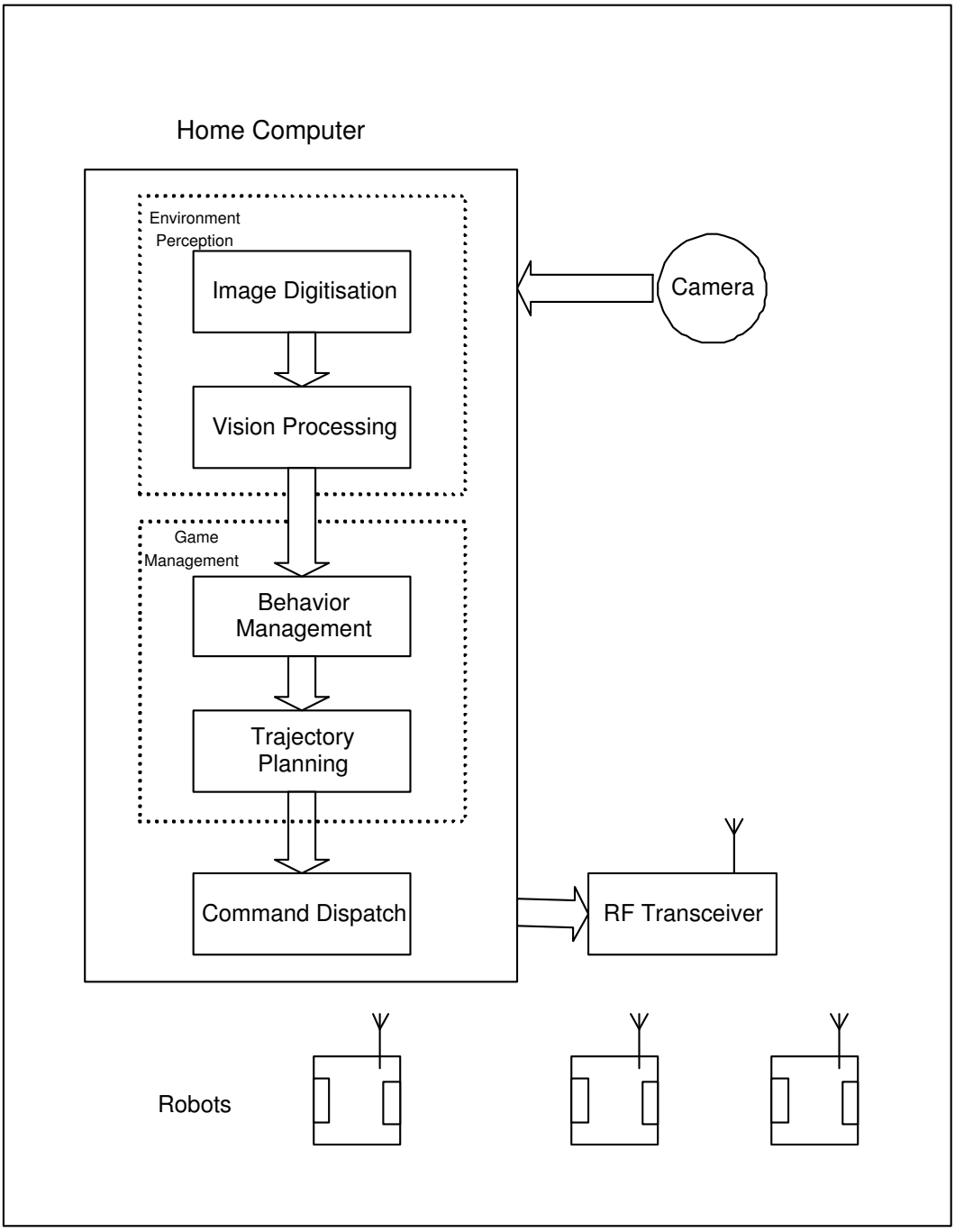


Figure 2.8: System process illustration

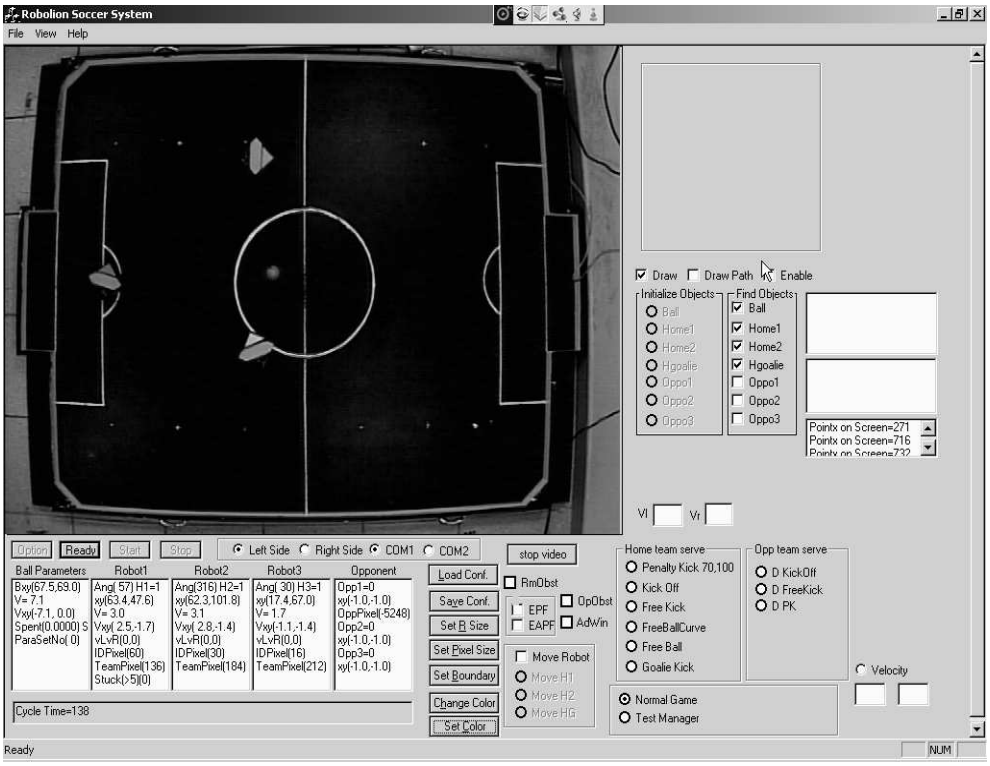


Figure 2.9: Robot Soccer System control panel

Table 2.1: Frequently-used Behaviors

Basic Behaviors	Function	Note
DirestShoot	Rush shoot for goal	
SpinShoot	Shoot by turning the body for goal	
Goal area Attack	Movement in Goal area	
CleanBall	Clean the ball out of home side	
GetBall	Close to get control to the ball	
LineGuard Defense	Stay along the dangerous line to protect	
PushBallInDangerZone	Push the ball out of penalty area	
Escape From Goal	Avoid to own goal	only for Goalie
GuardByEstimation	Protective patrol according to the ball position to	only for Goalie
ReturnGuardLine	Back to guard line after running out of penalty area	only for Goalie
AvoidBoundary	Check if the movement is too close to the boundary	for each robot

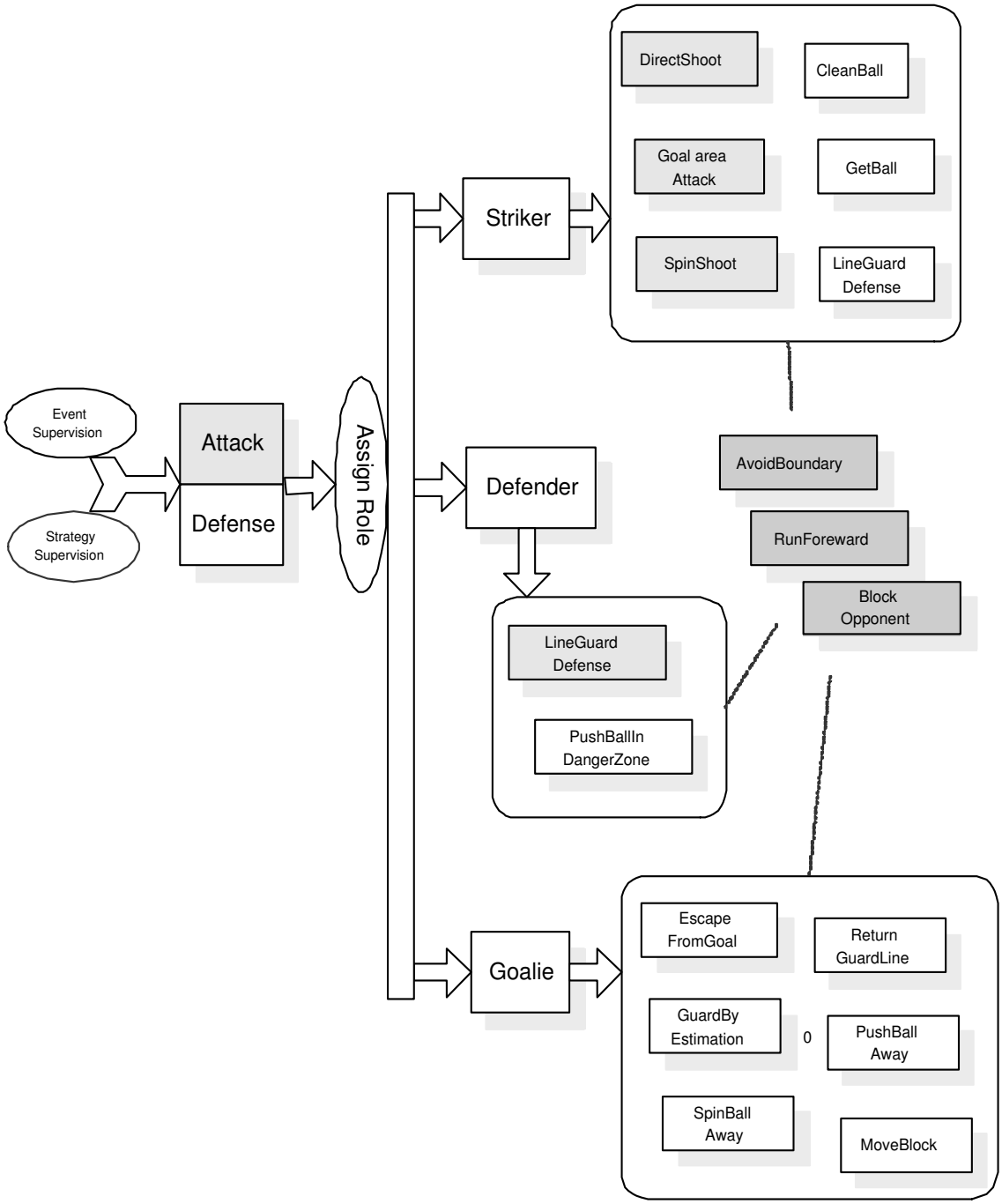


Figure 2.10: Robot Soccer game management architecture

The system decision-making starts from the top level of the whole system, Event supervision and Strategy supervision. In each computing loop, they dispatch the status to Role assignment, then the Striker, Defender, and Goalie roles are sequentially assigned to each robot according to the team attitude and the robot distribution. The final action of each robot is generated by the robot agent with basic behaviors. Note that in this system the robot agents are homogeneous both in software and in hardware. The individual robot object is generated from the same generic classes.

2.4 Discussions

In this chapter, the Robot Soccer System and the individual robots are discussed. The mobile robots are coordinated by the host computer through RF commands, and the whole game management is implemented hierarchically.

Chapter 3

Robot Modelling and Tracking Controller Design

3.1 Introduction

In this chapter, the wheeled mobile robot, a nonholonomic robotic system, used is discussed.

The research on control of nonholonomic systems has grown largely since last century [89][90][12]. Nonholonomic systems commonly arise in finite dimensional mechanical systems with constraints. Nonholonomic control systems have been studied in the domains of robot manipulation, mobile robots, and space robotics [91][92][93][31].

3.2 Wheeled-Robot Model

For a wheel mobile robot as shown in Figure 2.4, in 2-D space X-Y coordination system, the nonholonomic constraint is,

$$\dot{y}cos\theta - \dot{x}sin\theta = 0 \tag{3.1}$$

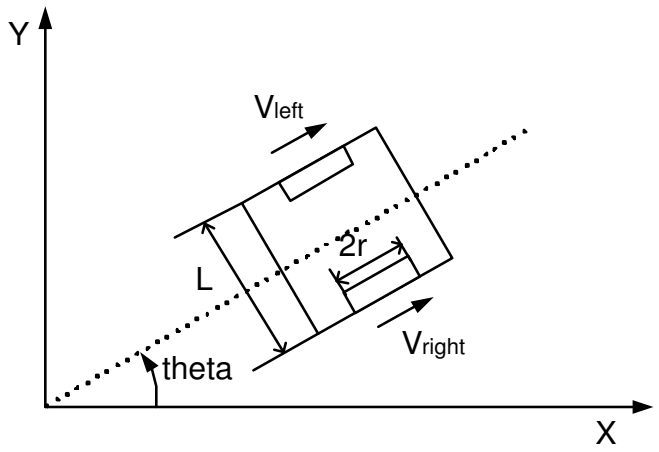


Figure 3.1: Robot posture in X-Y Coordination system

It equals to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ sin\theta & cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix}, \tag{3.2}$$

or,

$$\dot{q} = \mathbf{S}(\mathbf{q})\nu(\mathbf{t}) \tag{3.3}$$

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \tag{3.4}$$

where (x, y) is the robot position in X-Y coordinates, θ denotes the robot orientation, posture $q = [x, y, \theta]^T$, $\nu(\mathbf{t}) = [\nu, \omega]^{\mathbf{T}}$. Since the mass center and geometric center of the robot are identical, $d = 0$.

Driven by DC motor with Pulse Width Modulation (PWM), the dynamic model of the mobile robot is deduced in the following. For a DC motor, the torque τ_m is determined by the effective moment of inertia J_m and the angular velocity of the shaft $\dot{\omega}_m$, and in a relation with motor torque constant K_m and the current supplied to the armature i at the same time, as

$$\tau_m = J_m\dot{\omega}_m = K_m i. \tag{3.5}$$

A gear ratio of K_g reduces the rpm of DC motor to drive the wheel at higher torque τ_l, with effective moment of inertia J_l at the wheel axle and angular velocity ω_l,

$$\tau_l = J_l\dot{\omega}_l = K_g i\tau_m \tag{3.6}$$

Replacing τ_m with (3.5), we get

$$\omega_l = \frac{1}{K_g}\omega_m \tag{3.7}$$

which is equivalent to,

$$\dot{\omega}_l = \frac{1}{K_g}\dot{\omega}_m \tag{3.8}$$

$$\tau_l = J_l\dot{\omega}_l = K_g J_m\dot{\omega}_m. \tag{3.9}$$

Replacing ω_l with (3.7), we get

$$J_l = K_g^2 J_m. \tag{3.10}$$

According to Kirchhoff's Voltage Law, the voltage v_{in} is

$$v_{in} = iR_a + e_b = iR_a + K_m\omega_m \tag{3.11}$$

where R_a is armature resistance and e_b is back-emf induced in the armature which can be expressed by angular velocity ω_m and motor constant K_m,

$$v_{in} = \frac{J_m\dot{\omega}_m R_a}{K_m} + K_m\omega_m \tag{3.12}$$

$$\dot{\omega}_l = -\frac{K_m^2 K_g^2}{J_l R_a}\omega_l + \frac{K_m K_g}{J_l R_a}\omega_l v_{in} \tag{3.13}$$

There is a linear relationship between the duty cycle and the average output voltage level [94][95],

$$v_{in} = \beta\mu, \tag{3.14}$$

where μ is the control signal corresponding to the duty cycle, and β is a proportionality constant. Hence,

$$\dot{\omega}_l = -\frac{K_m^2 K_g^2}{J_l R_a}\omega_l + \frac{K_m K_g\beta}{J_l R_a}\omega_l\mu. \tag{3.15}$$

Let $k_1 = \frac{K_m^2 K_g^2}{J_l R_a}$ and $k_2 = \frac{K_m K_g \beta}{J_l R_a}$, then (3.15) can be rewritten as

$$\dot{\omega}_l = -k_1 \omega_l + k_2 v. \tag{3.16}$$

The relationship among the robot linear velocity ν, angular velocity ω and wheel speeds $(\nu_{left}, \nu_{right})$ is,

$$\nu = \frac{\nu_{right} + \nu_{left}}{2}, \tag{3.17}$$

$$\omega = \frac{\nu_{right} - \nu_{left}}{L}. \tag{3.18}$$

By Newton's Second Law of Motion,

$$\dot{\nu} = \frac{f_{right} + f_{left}}{m} = \frac{\tau_{right} + \tau_{left}}{mr} \tag{3.19}$$

where r is the wheel radius. Replace the torques with (3.6), and using (3.17),

$$\dot{\nu}_{right} + \dot{\nu}_{left} = \frac{2J_m(\dot{\omega}_{right} + \dot{\omega}_{right})}{mr} \tag{3.20}$$

The total torque τ in relation with forces on wheels (f_{left}, f_{right}) is

$$\tau = \frac{L}{2}(f_{right} - f_{left}) = \frac{L}{2r}(\tau_{right} - \tau_{left}) \tag{3.21}$$

According to (3.5), (3.21) can be written as

$$I\dot{\omega} = \frac{L}{2r} J_m(\dot{\omega}_{right} - \dot{\omega}_{right}) \tag{3.22}$$

where I is the moment of inertia, hence,

$$\dot{\omega} = \frac{L}{2Ir} J_m(\dot{\omega}_{right} - \dot{\omega}_{right}), \tag{3.23}$$

$$\dot{\nu}_{right} - \dot{\nu}_{right} = \frac{L^2}{2Ir} J_m(\dot{\omega}_{right} - \dot{\omega}_{right}). \tag{3.24}$$

Let $\mathbf{v} = \begin{bmatrix} \nu_{left} \\ \nu_{right} \end{bmatrix}$, and $\mathbf{u} = \begin{bmatrix} \mu_{left} \\ \mu_{right} \end{bmatrix}$. Combining (3.20) and (3.24),

$$\dot{\mathbf{v}} = \begin{bmatrix} \dot{\nu}_{left} \\ \dot{\nu}_{right} \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} \dot{\omega}_{left} \\ \dot{\omega}_{right} \end{bmatrix}, \tag{3.25}$$

where

$$\mathbf{H} = \begin{bmatrix} \frac{J_l}{mr} + \frac{L^2 J_l}{4Ir} & \frac{J_l}{mr} - \frac{L^2 J_l}{4Ir} \\ \\ \frac{J_l}{mr} - \frac{L^2 J_l}{4Ir} & \frac{J_l}{mr} + \frac{L^2 J_l}{4Ir} \end{bmatrix}. \tag{3.26}$$

Substituting (3.16) into (3.25),

$$\dot{\mathbf{v}} = \mathbf{K_1 v} + \mathbf{K_2 u}, \tag{3.27}$$

where $\mathbf{K_1} = -k_1\mathbf{H}$ and $\mathbf{K}_2 = k_2\mathbf{H}$. With the robot specifications: $r = 0.022m$, $R_a = 1.94$, $m = 0.650kg$, $Kg = 9.68$, $K_m = 6.92e^{-3}Nm/A$, $J_m = 2.7e^{-7}Kgm^2$, $L = 0.075m$, we get,

$$\mathbf{K_1} = \begin{bmatrix} -18.3799 & 3.6760 \\ 3.6760 & -18.3799 \end{bmatrix}, \mathbf{K_2} = \begin{bmatrix} 0.2158 & -4.32 \\ -4.32 & 21.58 \end{bmatrix}.$$

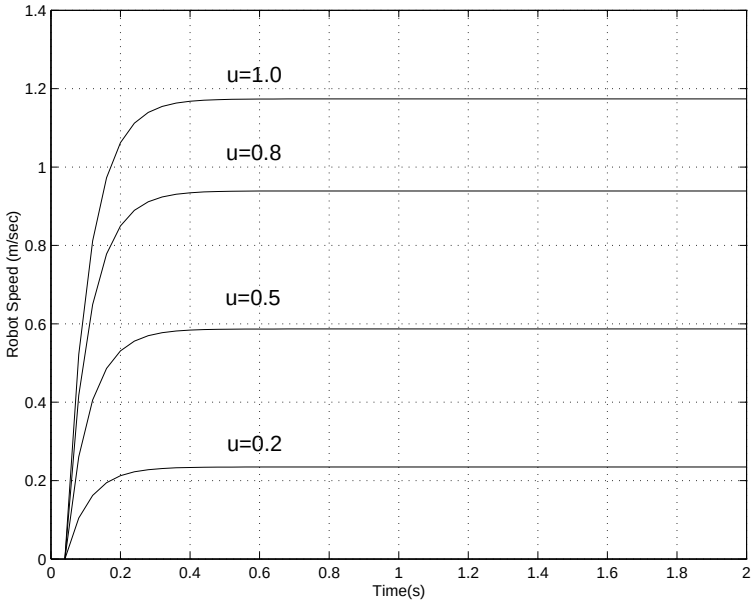


Figure 3.2: Robot response to different command inputs.

Once the robot speed becomes constant, $\dot{\mathbf{v}} = \mathbf{0}$, and,

$$\mathbf{K_1 v} + \mathbf{K_2 u} = \mathbf{0} \tag{3.28}$$

Assuming $\mathbf{G} = -\mathbf{K_1^{-1}K_2}$,

$$\mathbf{v} = \mathbf{Gu} \tag{3.29}$$

G acts as a gain of input controls to the final wheel speeds. With the condition that the motors are designed symmetrically and the wheel motors are decoupled, **G** is a diagonal matrix. The relationship between the input and the output speed for each wheel is simulated as,

$$\mathbf{v} = \mathbf{gEu}, \tag{3.30}$$

where **E** is the unit matrix, $\mathbf{g} \approx \mathbf{1.7}$ according to specifications provided earlier.

The simplified robot model is used to develop a game simulator to test the strategies and algorithms in small time load without damage on the real results. From Figure 3.2 we can see that it takes less than 0.4sec to reach the desired speeds with a 0.04sec time interval, i.e. within 10 run cycles the robot can reach the desired speed.

3.3 Tracking Controller

The proposed approach to generate commands for desired motion is verified in simulation and experiment. For target tracking problem, a heuristic fuzzy based method in a similar robot system, where fuzzy rule gains are tuned manually [50]. A sliding mode control for wheeled mobile robots is proposed in [32].

To generate the control command **u** for the robot, an auxiliary velocity signal $\mathbf{v}^*$ is considered [48],

$$\mathbf{v}^* = \begin{bmatrix} \nu_{tar} \cos e_3 + a_1 e_1 \\ \omega_{tar} + a_2 \nu_{tar} e_2 + a_3 \nu_{tar} \sin e_3 \end{bmatrix} \tag{3.31}$$

where ν_{tar}, ω_{tar} is the target object velocity and angular velocity respectively, and

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{tar} - x \\ y_{tar} - y \\ \theta_{tar} - \theta \end{bmatrix} \tag{3.32}$$

where a_1, a_2, and a_3 are positive constants and are optimized by EA to get acceptable performance. By (3.32), the errors between the robot and target in workspace

coordinates (Figure 3.1) are transformed to an error posture based on robot local coordinates. For example, when the target is ahead of the robot, $e_1 > 0$; when the target is behind of the robot, $e_1 < 0$.

From (3.31), ν is associated with the heading angle error $(\theta_{tar} - \theta)$, target velocity ν_{tar} and the distance error along the robot orientation e_1, while ω is generated by the errors of the heading angle and angular velocity ω_{tar}, and distance error along the perpendicular to the robot orientation e_2.

The command signal to achieve the auxiliary velocity $\mathbf{v}^*$ is $\mathbf{u}^*$ which satisfies (3.33).

$$\mathbf{K_2 u}^* = \mathbf{K_1 v} + \dot{\mathbf{v}}^* - \mathbf{k_e e_v} \tag{3.33}$$

where k_e is positive constant, $\mathbf{e_v} = \dot{\mathbf{v}} - \dot{\mathbf{v}}^*$, is the error between the desired velocity and robot velocity. Substituting (3.33) into (3.27), we get,

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}^* - \mathbf{k_e e_v}, \tag{3.34}$$

$$\dot{\mathbf{e_v}} = -k_e \mathbf{e_v}. \tag{3.35}$$

$$\tag{3.36}$$

Since k_e is a positive constant, $\lim_{t=+\infty}(\mathbf{e_v}) = \mathbf{0}$.

Hence the desired control signal $\mathbf{u}^*$ is

$$\mathbf{u}^* = -\mathbf{K_2^{-1} K_1 v} + \mathbf{K_2^{-1}} \dot{\mathbf{v}}^* - \mathbf{k_e K_2^{-1} e_v} \tag{3.37}$$

3.4 Simulation Results

The command generation method is tested for line path following and curve path following. In Figure 3.3, the robot moves from an initial position $(0.1, 0.13)m$, with initial speed 0, catches up with the moving ball initiated from $(0.27, 0.28)m$ with speed 0.8m/sec within 1 sec. The commands to left and right wheels rise to nearly maximum at the start, and converge to the target velocity rapidly with

stable distance to the target (Figure 3.6). Considering its size, the robot is closer to the target. When the target moves with sharp turning (Figure 3.7), the robot starts from the initial position $(0.1, 0.13)m$, with an initial speed of 0, follows up with the randomly moving ball initiated from $(0.27, 0.28)m$ within 1 sec. The robot makes a similar trajectory as the target. The distance between the robot and target decreases to $0.06m$, considering the sizes of the robot and ball, the robot moves to the target.

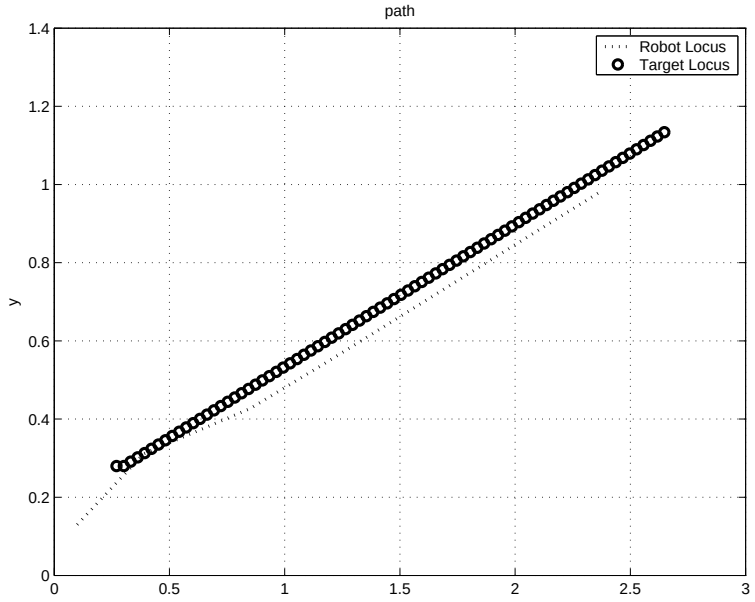


Figure 3.3: Robot following a line.

3.5 Experimental Results

The robot shown in Figure 2.4 is used in the experiments. The specifications are: $r = 0.022m$, $R_a = 1.94$, $m = 0.650kg$, $Kg = 9.68$, $K_m = 6.92e^{-3}Nm/A$, $J_m = 2.7e^{-7}Kgm^2$, $L = 0.075m$.

In Figure 3.9, the robot follows the ball and kicks it from of danger area. In Figure 3.10, the robot follows the robot with ball and intercepts it when closes to the goal area. Assuming Robot1 is in team defence, it starts from the down left position

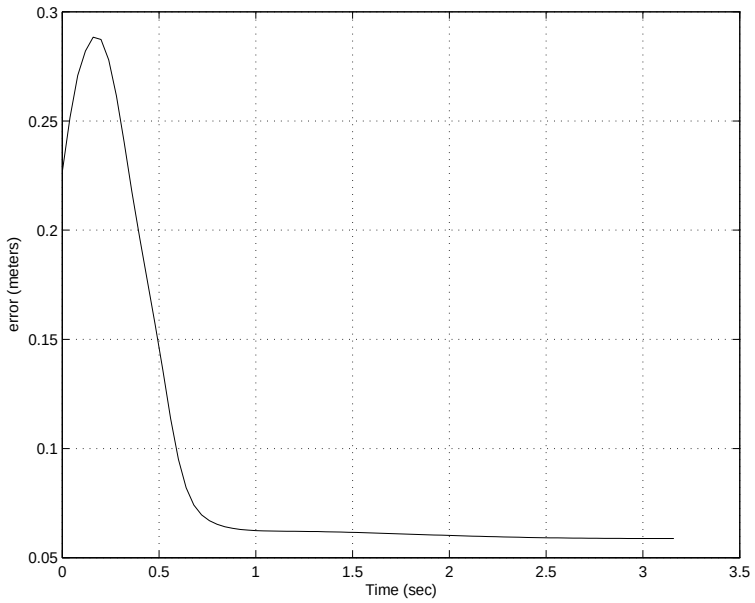


Figure 3.4: Distance error

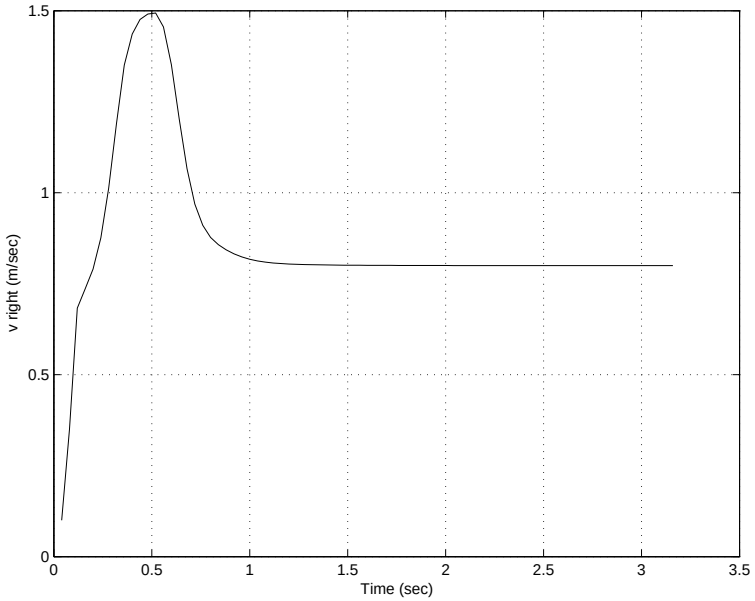


Figure 3.5: Velocity of right wheel

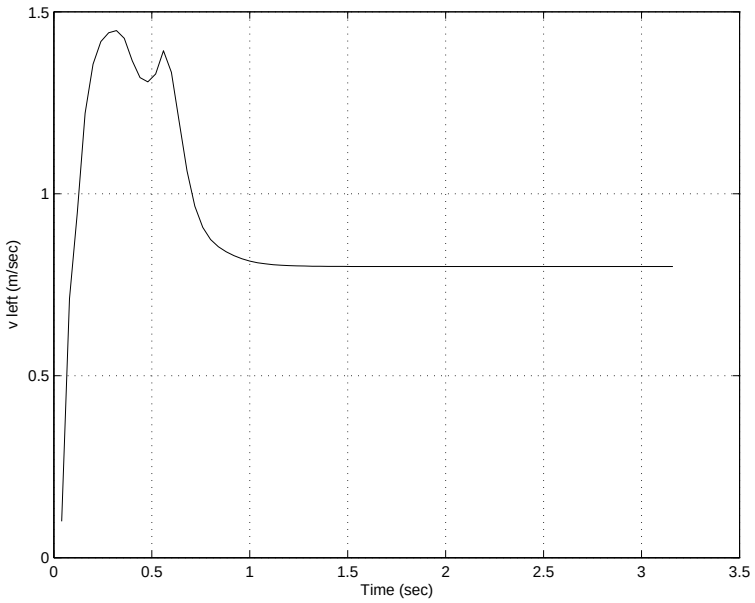


Figure 3.6: Velocity of left wheel

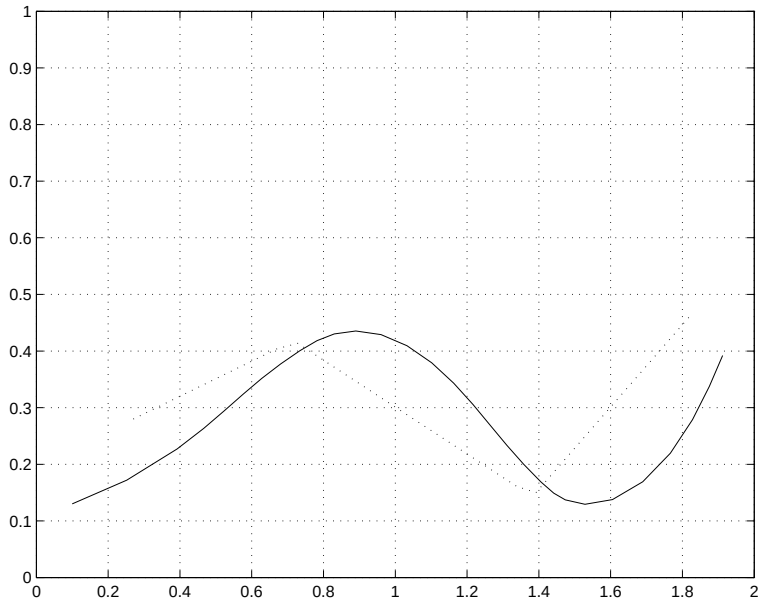


Figure 3.7: Robot following a line with sharp turnings

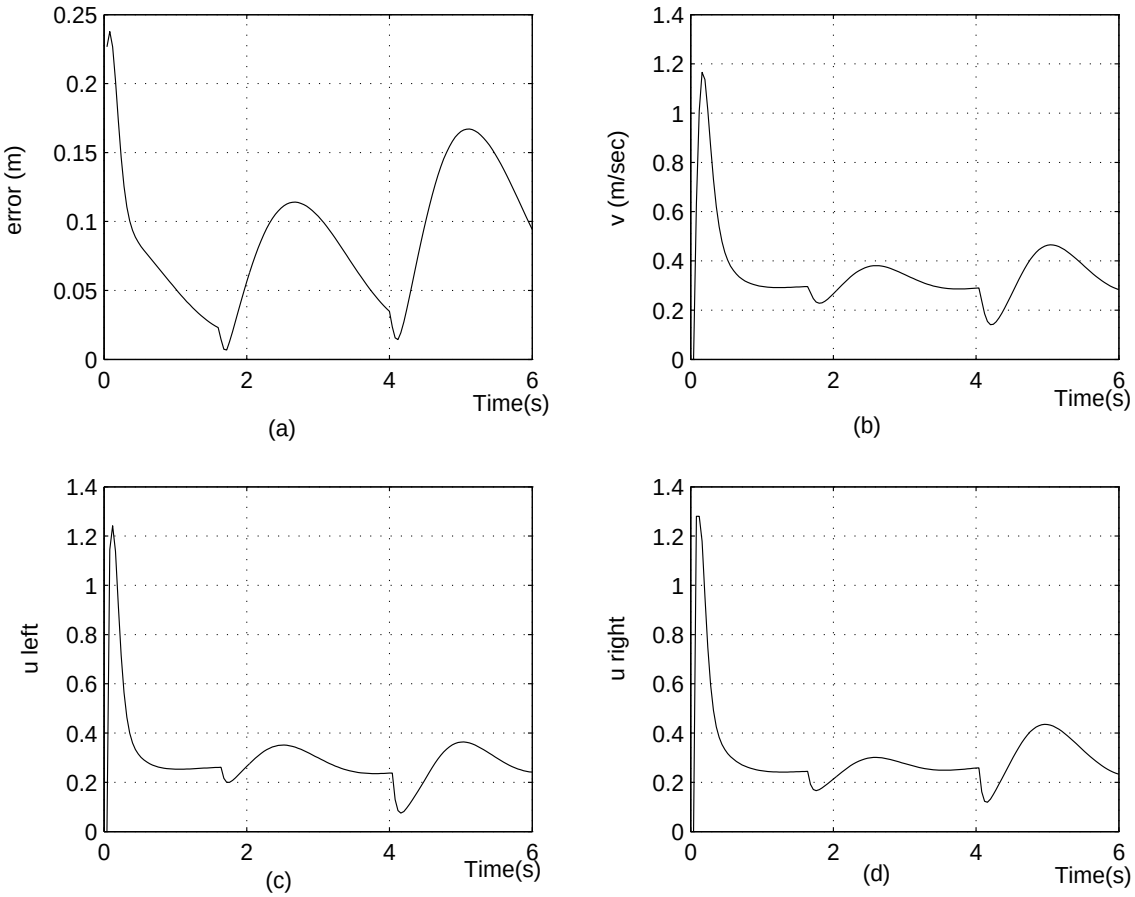


Figure 3.8: (a) The distance error between the robot and target (b) robot velocity profile (c) control command to the left wheel (d) control command to the right wheel

of the field, to block opponent team member Robot2 attacking from the up left position of the field. Robot1 succeeds blocking Robot2 at the interception position noted as in the figure and avoids the opponent attack attempt. In Figure 3.11, starting from the left side of the field, Robot1 runs longer to block the opponent Robot2 attacking along the boundary direction.

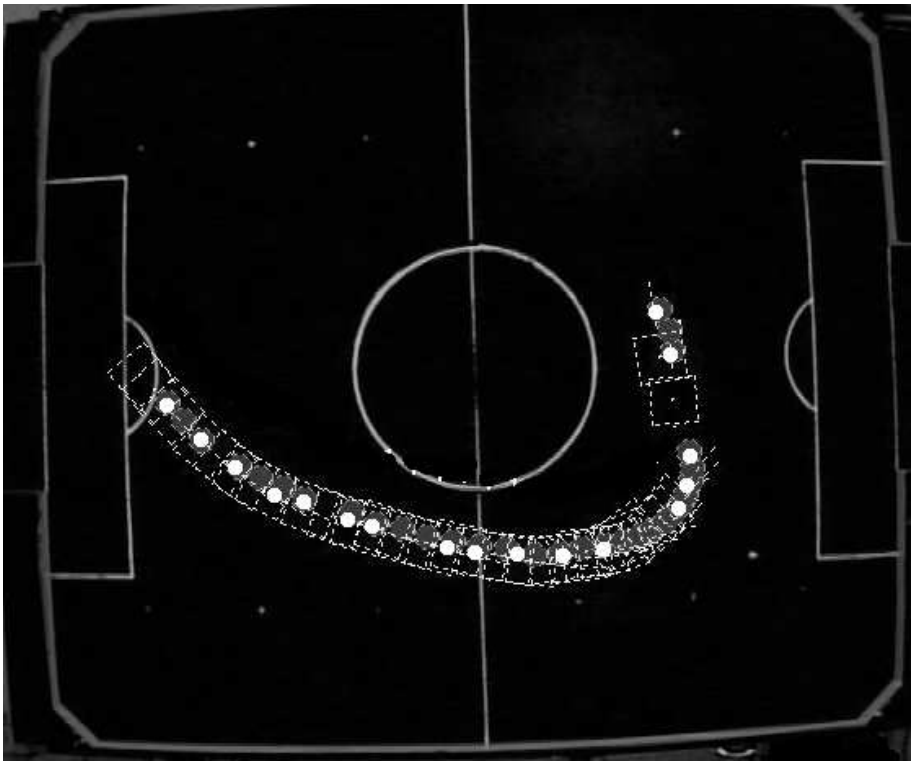


Figure 3.9: Robot blocking possible shoot

3.6 Discussions

In this chapter, the kinematic and dynamic model of the wheeled-robot used in RSS. The experimental results show the practicality of the specified models obtained from the non-holonomic general expressions.

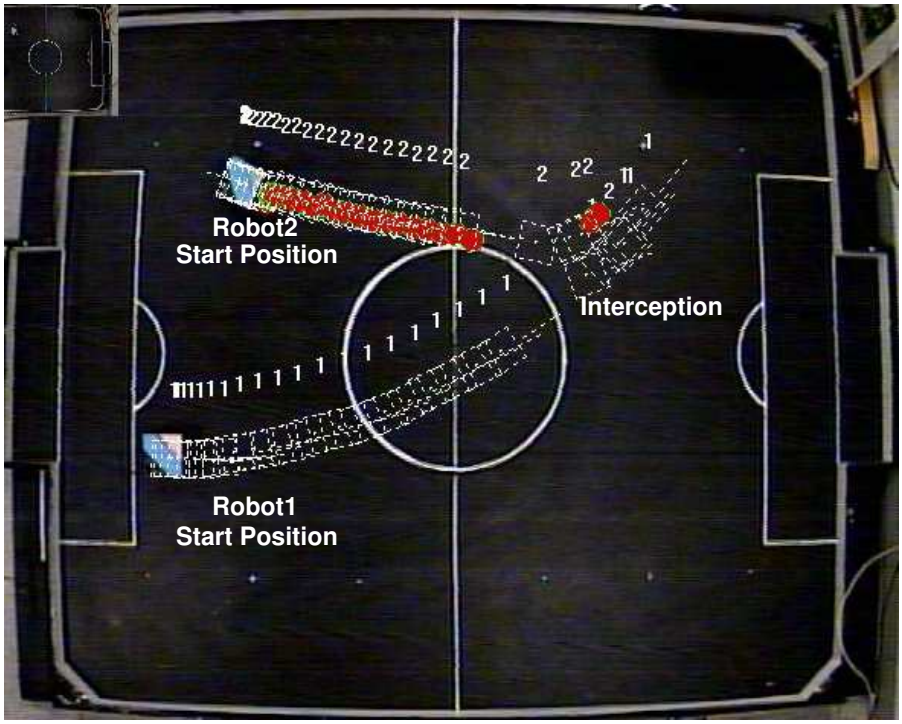


Figure 3.10: Robot blocking the opponent (case 1)

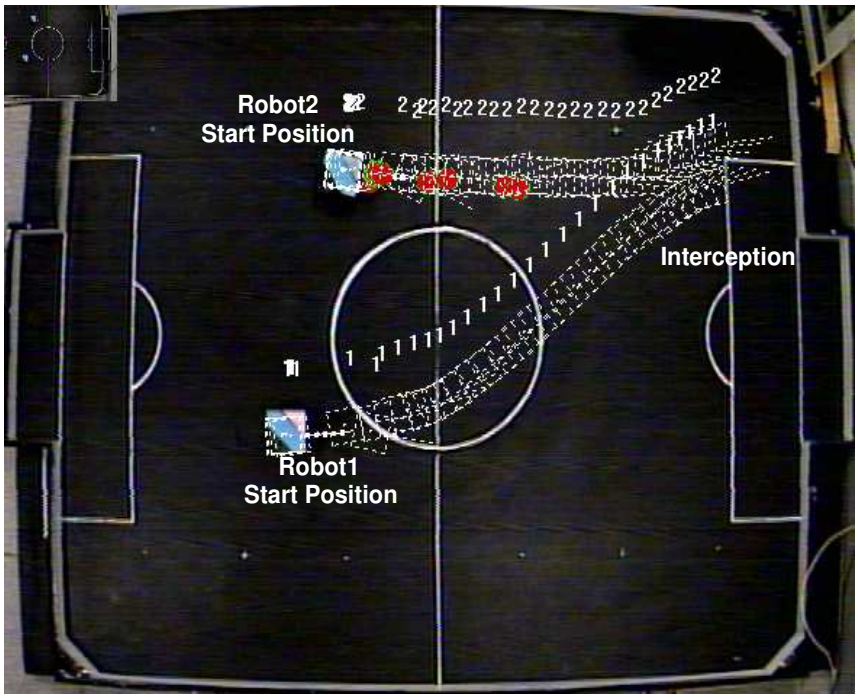


Figure 3.11: Robot blocking the opponent (case 2)

Chapter 4

Electrostatic Potential Field Based Path Planning

4.1 Introduction

In this chapter, structured from physical potential fields directly, Electrostatic Potential Field (EPF) is discussed as a Potential Field approach in path planning.

Kimon proposed the EPF [96] for indoor autonomous robot navigation in 2-D space. With no prior information about the environment, an occupancy map is established by sensor models in real time and converted into a resistor matrix. The electrical potential is calculated by Kirchhoff Current Law on the assumption that a robot is a current source and a target absorbs all the current. As a result, the point where the robot is positioned has the highest potential and the target the lowest. The mobile robot path is the maximal current path in a sheet of conductance networks mapped from the environment occupancy situation. EPF is proved to have a sole minimal at the sink point by classical electrostatic laws according to the potential function criteria suggested by Elon Rimon [97]. Therefore EPF is a local-minimal free navigation function. Smaller cell sizes result in better resolution of the path as the robot path is a connection of cells.

EPF is a basic navigation methodology and can be combined with other techniques. EPF is combined with Fuzzy Logic (FL) to fuse the sensor readings and predict possible collisions [98]. A two-layer FL inference engine is suggested in [98] to process the sonar sensor readings and guarantee collision avoidance. Once the detection module detects high collision possibility, the path is updated.

The main disadvantage of EPF is the associated high computational requirement. In the EPF method, for a stationary environment, the robot path would remain unaltered throughout. In a dynamic system, it is preferable for the robot to react promptly to avoid dangers. Unfortunately the computing speed of most mobile robot systems is not that fast enough and it is reasonable to formulate ways to reduce the computing time. EPF's computational requirement would increase tremendously with increase in the map resolution. Algorithms are required to tackle the issue associated with the computational requirement.

In this chapter, an Adaptive Window based EPF (AW-EPF) is proposed to bring down the computational time and to improve the real time performance of the EPF. In the proposed AW-EPF, an effective window area is set according to the current positions of the robot and the target. The obstacles in the immediate vicinity of the robot are identified. The electrical potential is calculated with respect to the effective window to determine a nearly optimal direction for the robot's next travel log. The proposed approach is able to generate a shorter path. The proposed approach, also partially solved a problem associated that the empty space between two obstacles which generally cannot be passed through even if the space is large enough for the robot.

The AW-EPF path planner instructs the path direction on spot and generates the robot direction for the following step. As the complete path is not generated in a single iteration, the navigation efficiency is improved, especially if the environment is dynamic.

AW-EPF is implemented in a Robot Soccer System. The robots are controlled to move inside a wooden field via RF commands from a host computer. The

information of the environment is conceived by a video camera placed over the field. In the mapping from the environment to the resistor network, the obstacle cells are assigned with high resistance instead of treating the cells as open circuits to simplify the implementation. The experimental results show that the AW-EPF can reduce computations and is able to generate shorter paths.

4.2 Electrostatic Potential Field Construction

It is proved that the unique solution to a resistor network system is equivalent to minimizing the instantaneous power consumed by the network and to identify a maximum current path that follows the path of least resistance [96]. The EPF is a natural potential field approach developed from the theories of electrostatics. In EPF approach, the navigation problem is to find the maximal current flow within a resistor network. The maximal current path is taken as the optimal path for the robot to follow.

The expression for electric field in terms of the applied voltage can be expressed in gradient from [99] as,

$$E = -\nabla V. \tag{4.1}$$

The total of the electric flux out of a closed surface is equal to the charge enclosed divided by the permittivity of free space. It is often convenient to construct an imaginary surface, a Gaussian surface, to take advantage of the symmetry in the physical situation. The electric flux through an area is defined as the electric field multiplied by the area of the surface projected in a plane perpendicular to the field. By Gauss's Law, the divergence of the electric field at a point in space is equal to the charge density ρ divided by the permittivity of free space ε_0.

$$\nabla \cdot E = \frac{\rho}{\varepsilon_0} \tag{4.2}$$

Therefore by Poisson's equation the potential is related to the charge density.

$$\nabla \cdot \nabla V = \nabla^2 V = -\frac{\rho}{\varepsilon_0} \tag{4.3}$$

In a charge-free region of space, (4.3) becomes the LaPlace's equation (4.4).

$$\nabla^2 V = 0 \tag{4.4}$$

From (4.4), it is inferred that there is no local minima in the electrostatic potential field.

To calculate the potential, Kirchhoff's Current Law (KCL) is applied. This fundamental law results from the conservation of charges. The current I is the total outward flux through the surface S is,

$$I = -\frac{d}{dt}\int_V \rho \cdot dv. \tag{4.5}$$

The total charge flowing into a node must be the same as the total charge flowing out of that node. In the mapping from the environment to a resistor network, the robot node is the external current source and the target node is the sink.

For a resistor system of $N = n \times n$ nodes, the current from the k^{th} node to the j^{th} node, i_{kj} is calculated by (4.6).

$$i_{kj} = G_{kj}(v_k - v_j) \tag{4.6}$$

where G_{kj} is the conductance of the branch connected between nodes k and j, v_k and v_j are the voltages of the two nodes respectively. From (4.6) a general equation is derived,

$$G_{N\times N} \cdot V_N \;\; = \;\; J_N. \tag{4.7}$$

V_N is the potential value matrix, $G_{N\times N}$ is the admittance matrix and J_N is current matrix [100]. $G_{N\times N} = [g_{jk}]_N$ is formed such that g_{kk} is equal to the sum of all the conductances connected to node k and g_{kj} is equal to the negative sum of all the conductances connected between the nodes k and j. The potential value matrix V_N then can be calculated by (4.8).

$$V_N \;\; = \;\; G^{-1}_{N\times N} \cdot J_N \tag{4.8}$$

However, it is not practical to calculate G^{-1} when the matrix order is usually higher than 20 in real life applications. The Gause Jordan Elimination (GJE) method is

applied in this work to calculate the potential V. Initially an augmented matrix, A (4.9) is formed by G and J.

$$A = [G\ J] = \begin{pmatrix} g_{11} & g_{12} & g_{13} & .. & g_{1N} & i_1 \\ g_{21} & g_{22} & g_{23} & .. & g_{2N} & i_2 \\ g_{31} & g_{32} & g_{33} & .. & g_{3N} & i_3 \\ .. & .. & .. & .. & .. & .. \\ g_{N1} & g_{N2} & g_{N3} & .. & g_{NN} & i_N \end{pmatrix} \tag{4.9}$$

The row reduction is performed until a nearly non-zero diagonal matrix is derived.

$$A = [G'\ J'] = \begin{pmatrix} g'_{11} & g'_{12} & g'_{13} & .. & g'_{1N} & i'_1 \\ 0 & g'_{22} & g'_{23} & .. & g'_{2N} & i'_2 \\ 0 & 0 & g'_{33} & .. & g'_{3N} & i'_3 \\ .. & .. & .. & .. & .. & .. \\ 0 & 0 & 0 & .. & g'_{NN} & i'_N \end{pmatrix} \tag{4.10}$$

The equation (4.7) can be written as in (4.11).

$$G' \cdot V \ = \ J', \tag{4.11}$$

where the potential V can be obtained easily.

In the electrical network, the target is considered as the sink, the navigated robot as the source and obstacles around as high value resistors (Figure 4.1). Free spaces are occupied by low value resistors. The electrostatic potential field created through a discrete network of linear passive resistors is free of all local minima except at the external sources or sinks [96]. The procedure to find an optimal path is to map the workspace to a sheet of resistive network and then to solve the electrostatic matrix to obtain the potential values. For other application domains, the EPF approach can be combined with specific features to obtain satisfactory results.

The EPF is associated with complex matrix calculations and it is critical to reduce the computing time for real time application. No matter the ways to solve the matrix equation, the computational complexity of EPF is $O(N)$ [96], where N is

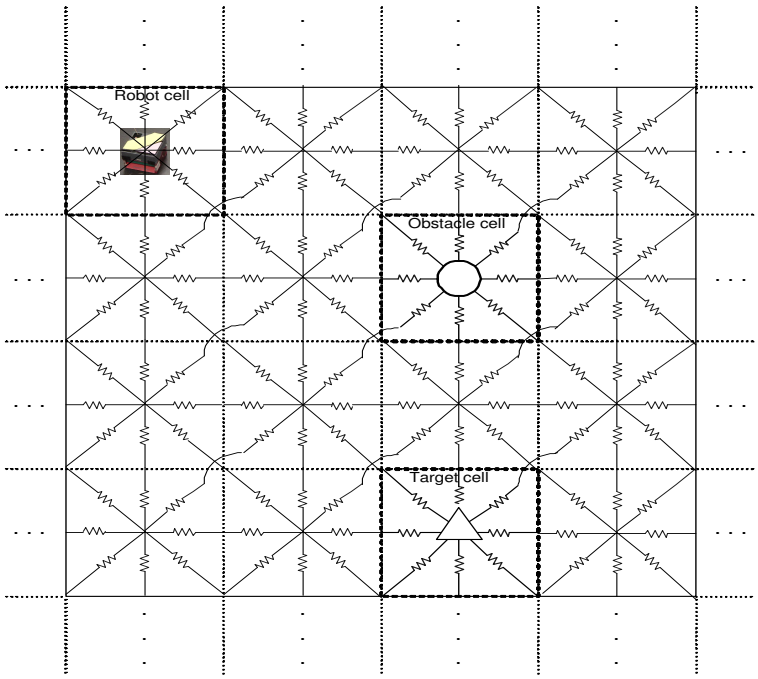


Figure 4.1: In the electrical network, the target is considered as the sink point, the navigated robot as the source and obstacles around as high value resistors, free spaces are occupied by low value resistors.

Table 4.1: Effects of grid size

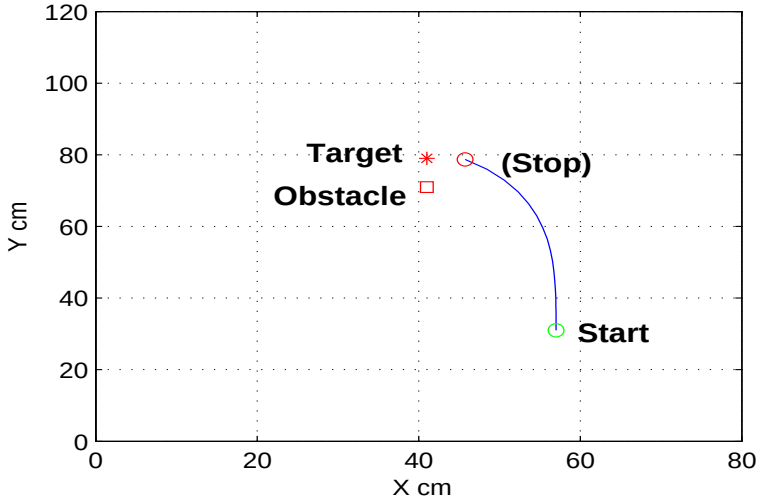
Grid size	Computing time (s)	Run iterations
7x7	0.03	20
9x9	0.12	22
10x10	0.17	28

the number of polygons in the space. The effects of cell numbers on the trajectories are shown by simulation results by comparing the robot trajectories with different grid numbers in the same situation (Figure 4.2). In the simulation, the robot starts from the bottom of the field to the Target on the upper position, avoiding the Obstacle just below the Target. By Figure 4.2 (a), (b) and (c), the trajectories with the grid number 7x7, 9x9, and 10x10 are illustrated respectively, and their computing time for EPF per loop and total run iterations listed in Table 4.1. When the grid size increases to 9x9 from 7x7, the computing time on EPF becomes four times of the previous one. lIt is clear that the computing time increasing dramatically with the higher resolution.

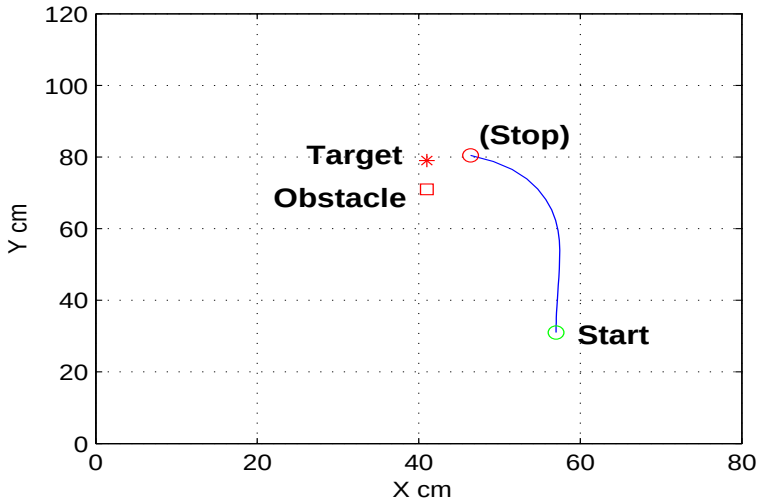
As the computation expense increases largely with the number of cells in the work space, which in turn affects the real time performance seriously, an adaptive window with a fixed grid number would improve the resolution without increase in computational burden.

4.3 Adaptive Window based EPF(AW-EPF)

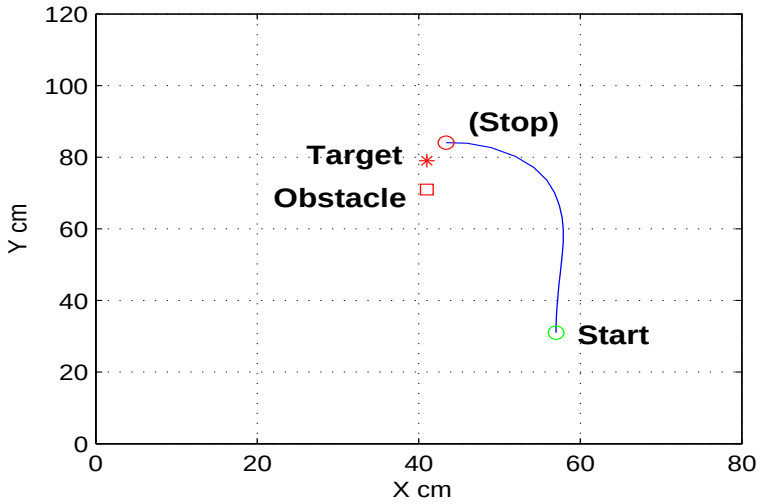
An adaptive window based EPF (AW-EPF) is proposed to bring down the computational burden (Figure 4.3). The steps involved in the AW-EPF approach are: a) set an effective window according to the positions of robot and target, b) identify obstacles inside the effective window, c) convert the workspace to a resistive network, d) solve the network matrix and f) identify the maximal current direction as the robot travel direction. The cells of resistors (Figure 4.1) are all square-shaped in a 2-D space. The resistor values occupied by obstacles are set to a high value.



(a) grid number 7x7



(b) grid number 9x9



(c) grid number 10x10

Figure 4.2: Trajectories with different cell numbers

In AW-EPF, the effective area is a window that moves along with the robot and target. Only the obstacles closer to the robot within a specific circular area are considered as valid obstacles. These two features of the AW-EPF reduce the computational burden.

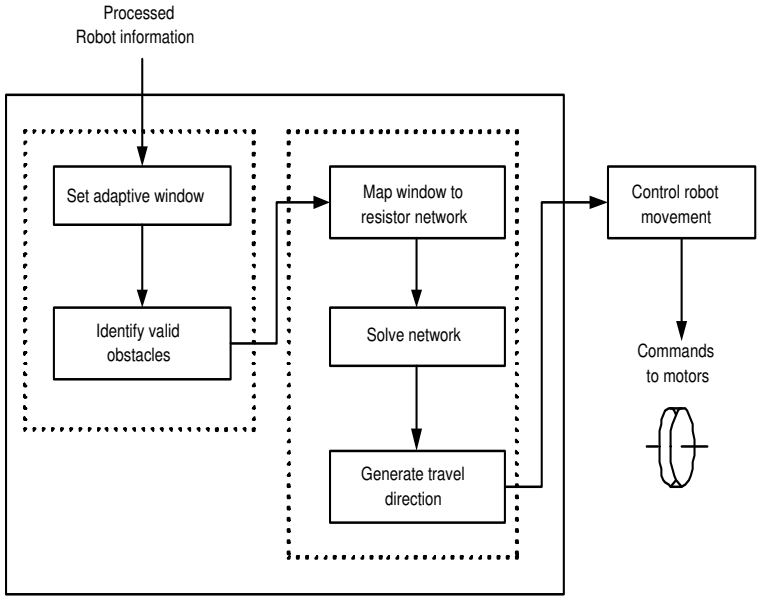


Figure 4.3: Robot information is filtered by the adaptive windows to reduce the computing, then resistor network is mapped and used to navigate the robot movement.

An adaptive window (rectangular frame in Figure 4.4) of constant number of cells is proposed to cut down the computational expense without resolution loss. In Figure 4.4) (a), Obstacle1 is ignored because it is beyond the space between the robot and target; Obstacle3 is also ignored because it is distant for robot current position; Only Obstacle2 is treated as obstacle in this situation; in Figure 4.4) (b) Obstacle1 is ignored because it is beyond the space between the robot and target; Obstacle2 and obstacle3 are both treated as obstacles in this situation.

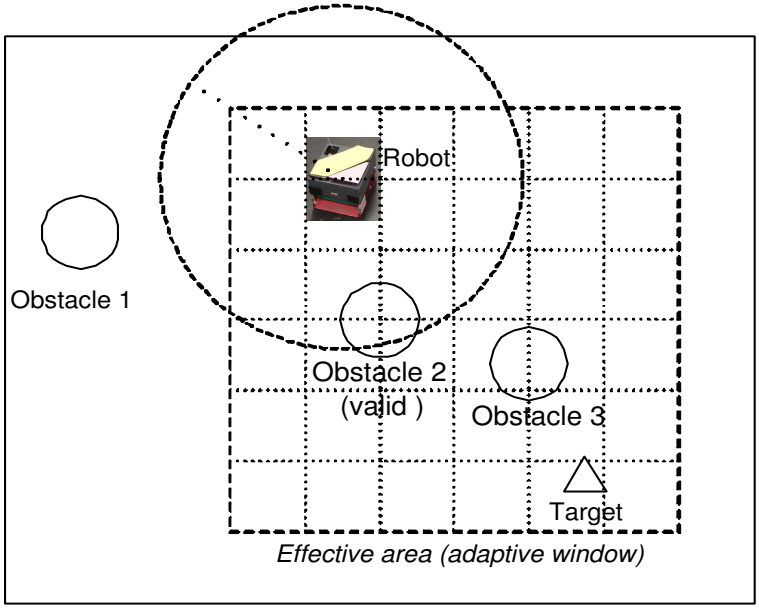
Since the focus of navigation is the space between the robot and the target, the adaptive window boundary is determined by the positions of the target and the robot. When the robot is closer to the target the window size is decreased and with the number of cells within the window remain unchanged, the resolution is

increased. This is beneficial as the path would be more critical when the robot is closer to the target. The effective area window is adaptive in terms that the window size changes at each update iteration according to the positions of the robot and the target. With the adaptive window, the path could be more precise with the same computing time by sacrificing less important areas, and so the resolution of the space is higher without an increase in computational time.

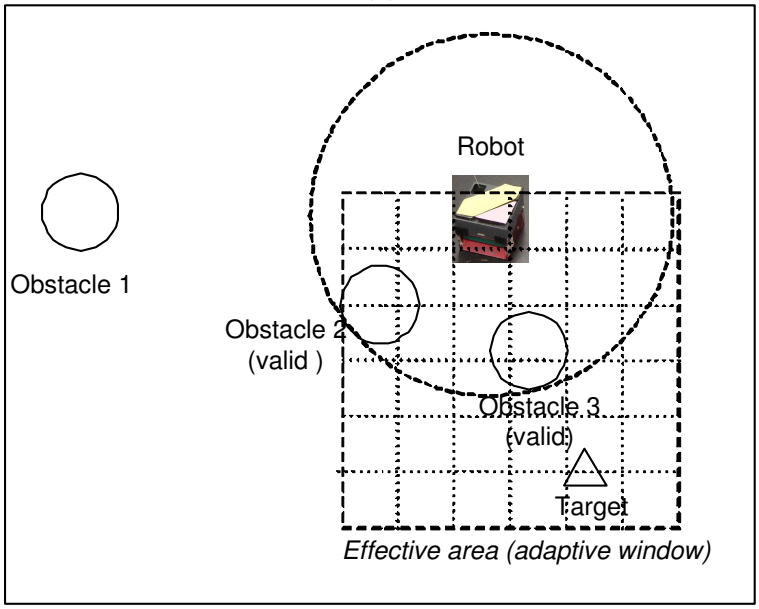
Meanwhile, by eliminating the distant obstacles it may help to keep the robot path direct as long as possible which in turn shortens the path as shown in Figure 4.5. In the artificial potential field, an obstacle would influence the shape of potential distribution and block the robot to pass around in facing direction. It is natural that an obstacle which is away from the robot could be ignored temporarily in path planning to reduce the computation and to make the path smoother.

The obstacles outside a circle of five times of the robot size are not valid for the potential construction (the circle at top left part of Figure 4.4). The robot size unit is the radius of robot circumcircle. Moreover, if two obstacles are far away by distance of three times of robot size, the farthest obstacle with respect to the robot is not considered while planning the path. By this method, the path would be more direct. The path differences for such situations are illustrated in Figure 4.5. The robot starts from the top circle position, avoids the rectangular obstacles and reaches the bottom target circle. Removing the less dangerous obstacles is valuable to partially resolve the problem associated with potential field approaches that the robot may not pass through two obstacles even when the empty space in between is large enough for the robot to pass (Figure 4.5).

From the potential distribution (Figures 4.6 to 4.8), it is observed that the obstacles block the current through it and the potential around the obstacles become higher. It is observed that the AW-EPF based potential field is smoother than the EPF based potential.

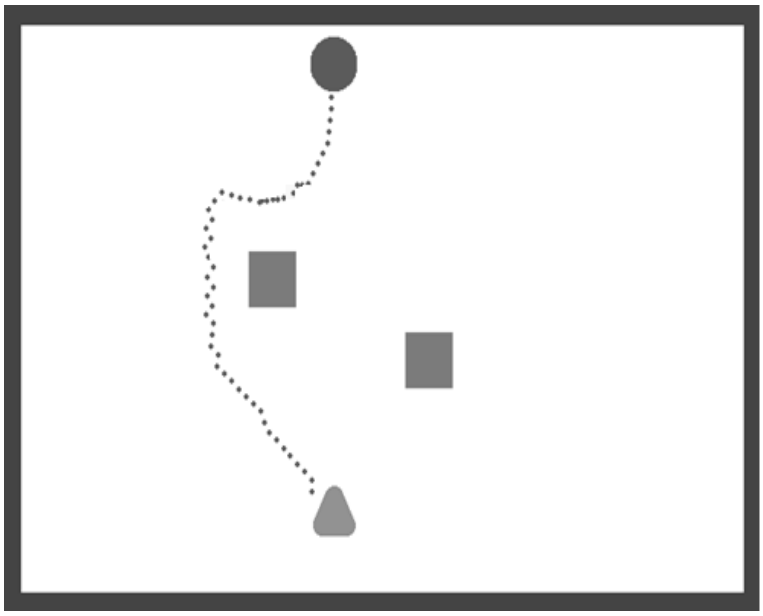


(a)

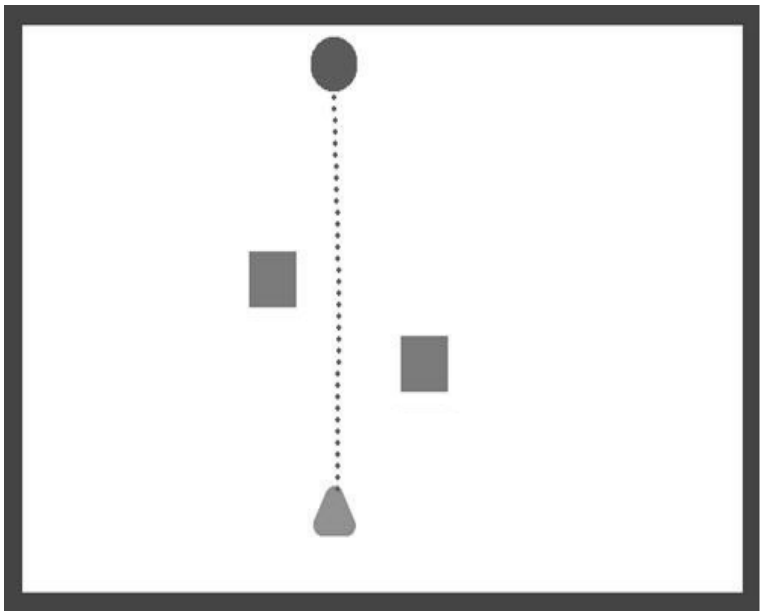


(b)

Figure 4.4: Examples of Adaptive Window work policy



(a)



(b)

Figure 4.5: Simulated paths comparison (2 stationary obstacles), (a)In EPF-based approach, the robot chooses a outside path to avoid both obstacles; (b) In AW-EPF-based approach, the robot passes between the obstacles with shorter pathlength.

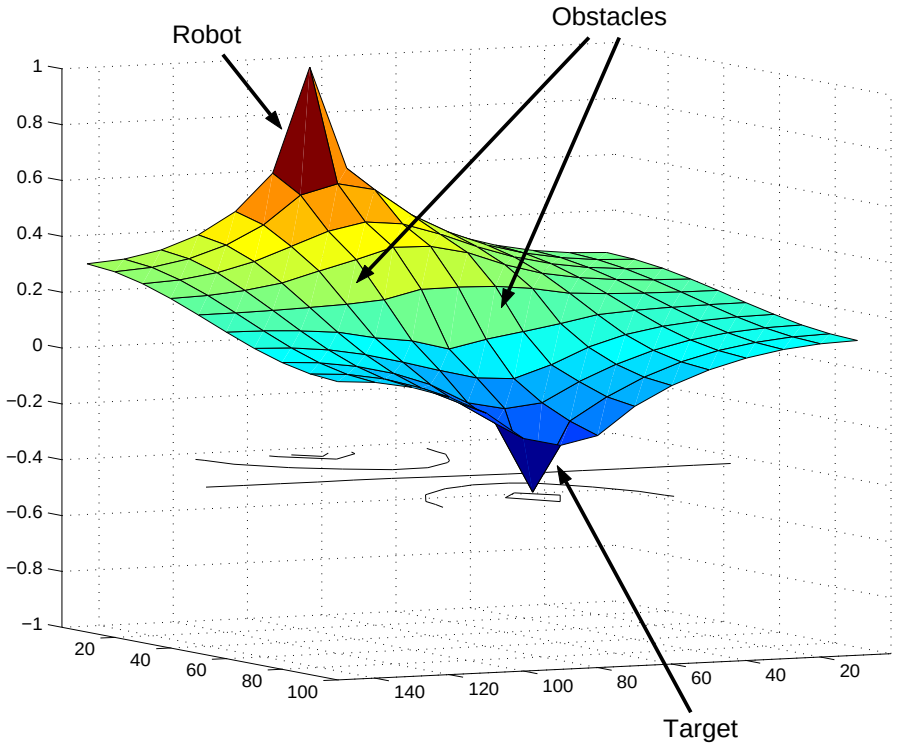
4.4 Experimental Results

The proposed path planner is verified on a multi-robot Soccer System (Chapter 2).

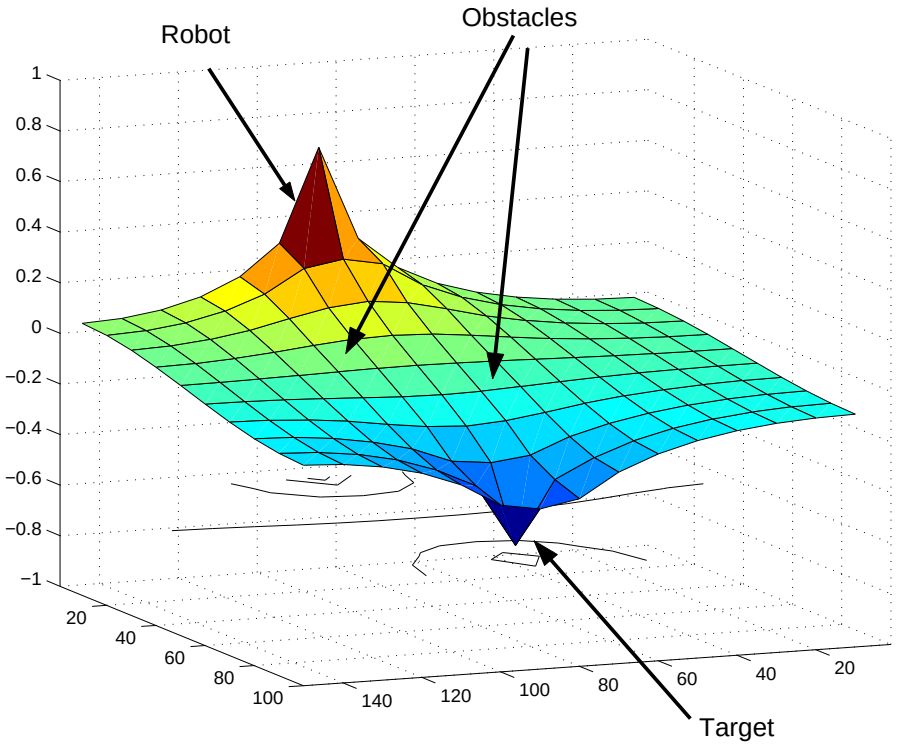
The AW-EPF path planner built-in with the robot motion controller in the host computer enabled the robot to avoid other robots on its way to the target (ball). The robot's positions in the field are identified from the updated vision frame of an overhead camera. The experimental results show that AW-EFP is good as a navigation method for mobile robots for short and quick paths. For different scenarios (Figure 4.9-4.12), the generated paths are more direct with a shorter path and running time. The records are taken from monitor screen, where the main background is the wooden playground.

In Figure 4.9, one robot (Robot 1) is trying to avoid collision with another (Robot 2) to reach the target ball. When the robot closes in to the target, the resolution becomes higher resulting in a smoother path. The rectangular dotted frame in Figure 4.9 on the right is the effective boundary area at the current frame. The window moves and its size changes with respect to the robot motion. With more obstacles, similar results are observed. Especially with 2 obstacles (Figure 4.10), the robot ran through the empty space between the obstacles with the AW-EPF navigation. This result demonstrates an important advantage of AW-EPF in overcoming the 'misleading' repulsive forces in empty space between adjacent obstacles.

The robot followed a shorter path when the obstacles are moving as shown in the following experiments. In Figure 4.11, the robot will move to right of the ball through a stationary Obstacle1 and a straight-line moving Obstacle2 which starts from the upper center of the field. As the trails drown on the fields, the robot with AW-EPF (the lower figure) avoided the obstacle with a smoother path to reach the target ball. In Figure 4.12, Obstacle1 and Obstacle2 are both moving in opposite directions, and the robot with AW-EPF also passes the barriers smoother (the lower figure). It is noted that AW-EPF has better performance happens in cases

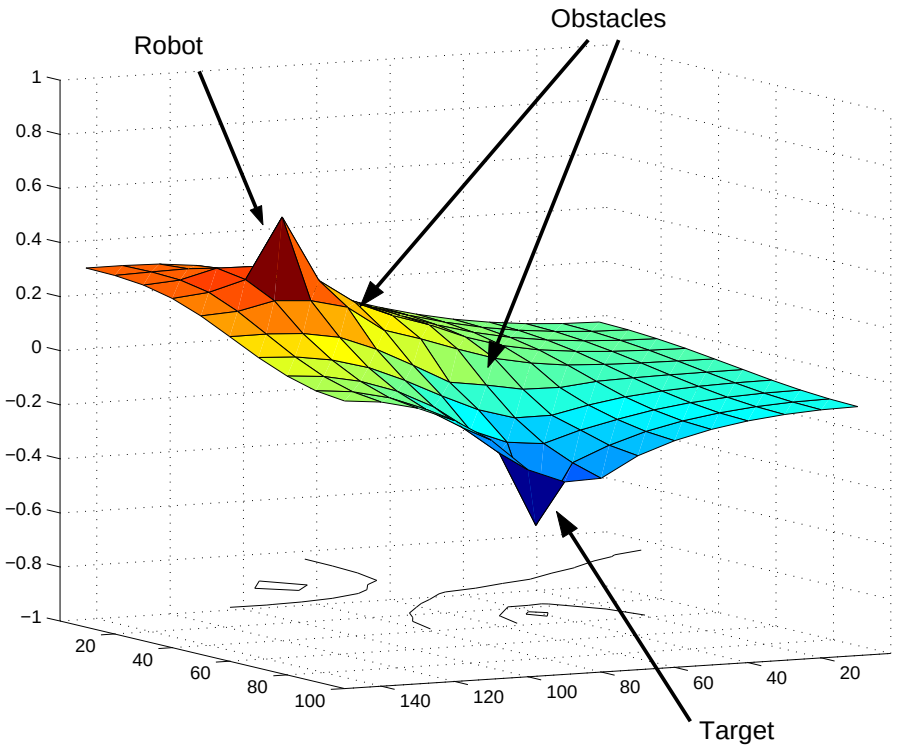


EPF-based

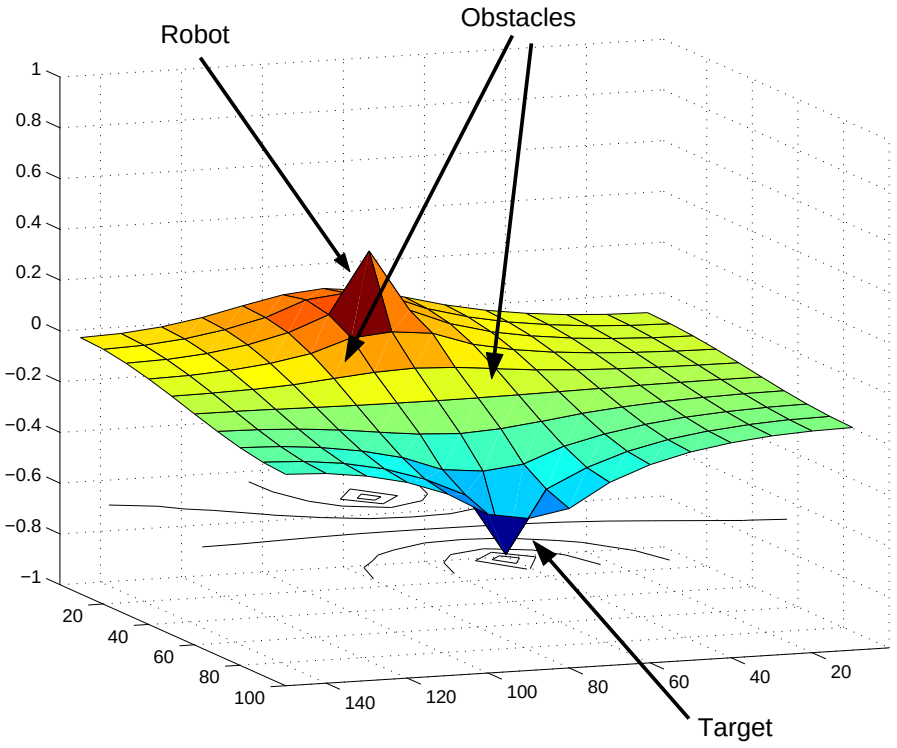


AW-EPF-based

Figure 4.6: Simulated potential comparison (Initial position)

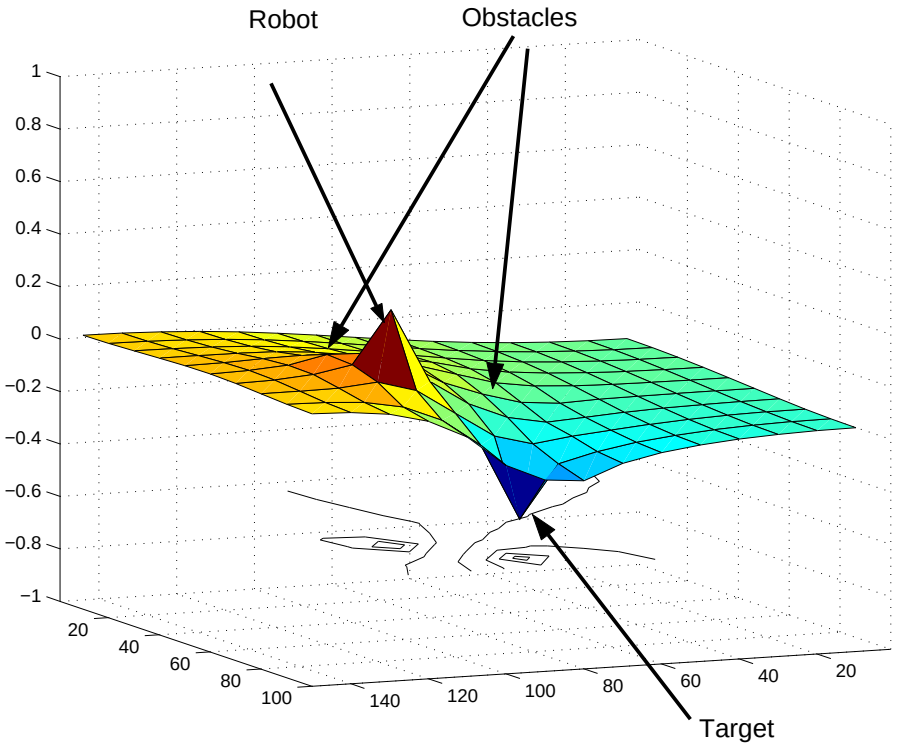


EPF-based

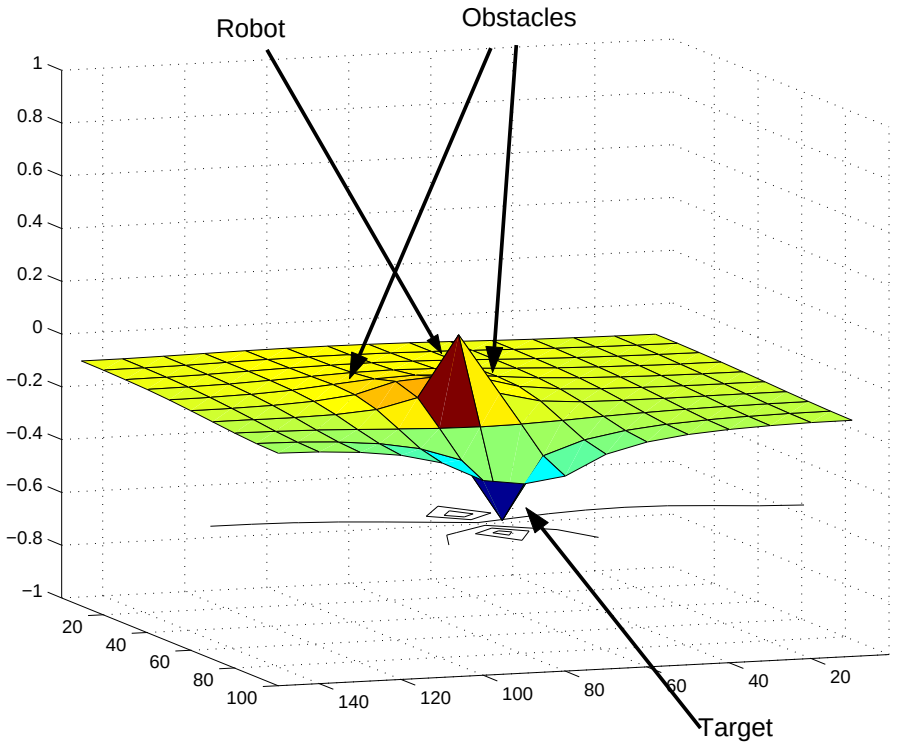


AW-EPF-based

Figure 4.7: Simulated potential comparison (Intermediate I)



EPF-based



AW-EPF-based

Figure 4.8: Potential comparison (Intermediate II)

Table 4.2: AW-EPF time illustration

Case	Loop time (ms)		Run time (s)		
	EPF	AW-EPF	EPF	AW-EPF	Improvement
case1	228	83	16.2	8.4	48%
case2	203	142	13.6	5.0	63%
case3	234	147	9.96	7.95	20%
case4	229	108	12.93	10.7	17%

the obstacles move away from the robot movement direction, and there might be risks when the obstacles running fast into the robot way.

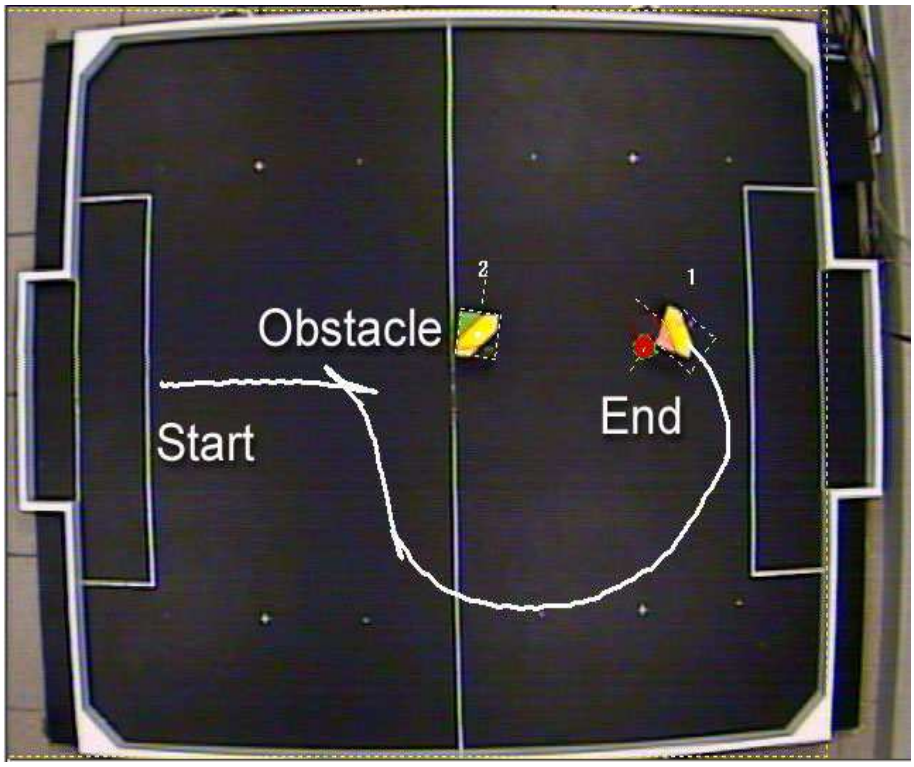
The path is quicker in terms of the running time for each session (Table 4.2) and AW-EPF reduced the run time. It is possible to illustrate the time saved with the help of typical scenarios. The process loop time and whole run time are both decreased considerably. The main reason for this is that the number of cells in AW-EPF is less. Secondly the less important obstacles are ignored depending on the prevailing scenarios.

The most notable improvement in performance is in the one-stationary-obstacle scenario (Figure 4.9). The running time reduces to half of the original and the loop time reduces when the obstacle is not taken into account. The run time is reduced in different ratios depending on the numbers of obstacles and their positions.

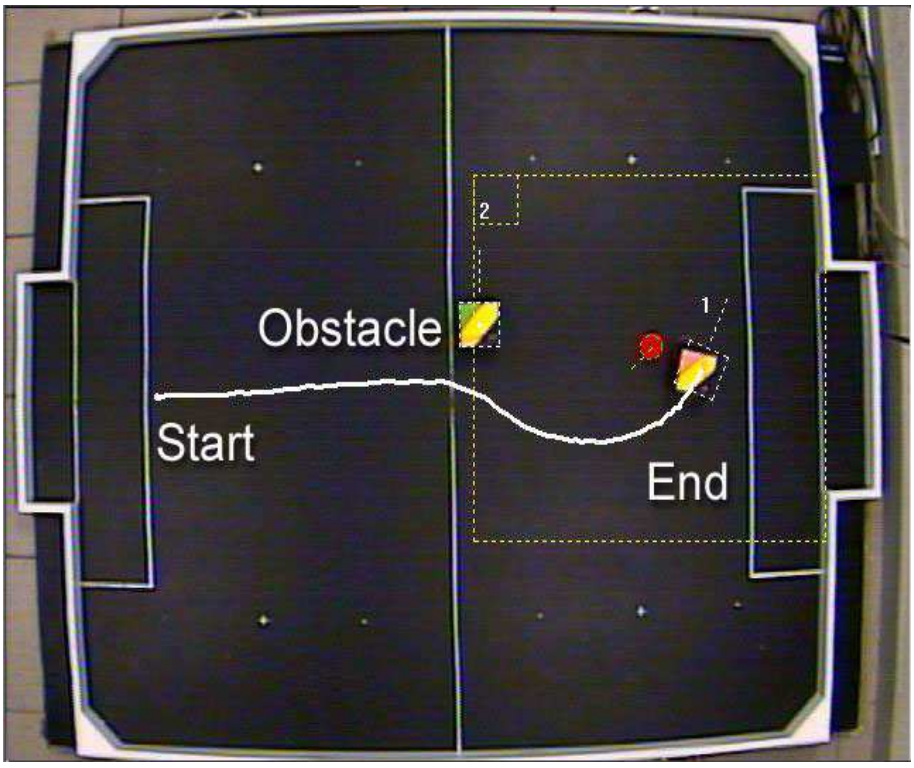
AW-EPF works well as a navigator in mobile robot system to generate smoother and shorter collision free paths. The AW can be applied to other path planning methods as well.

4.5 Discussions

From the experiments, it is observed that AW-EPF with adaptive ability can generate shorter and smoother paths for mobile robots. A path generator with more intelligence handles various cases in real time.

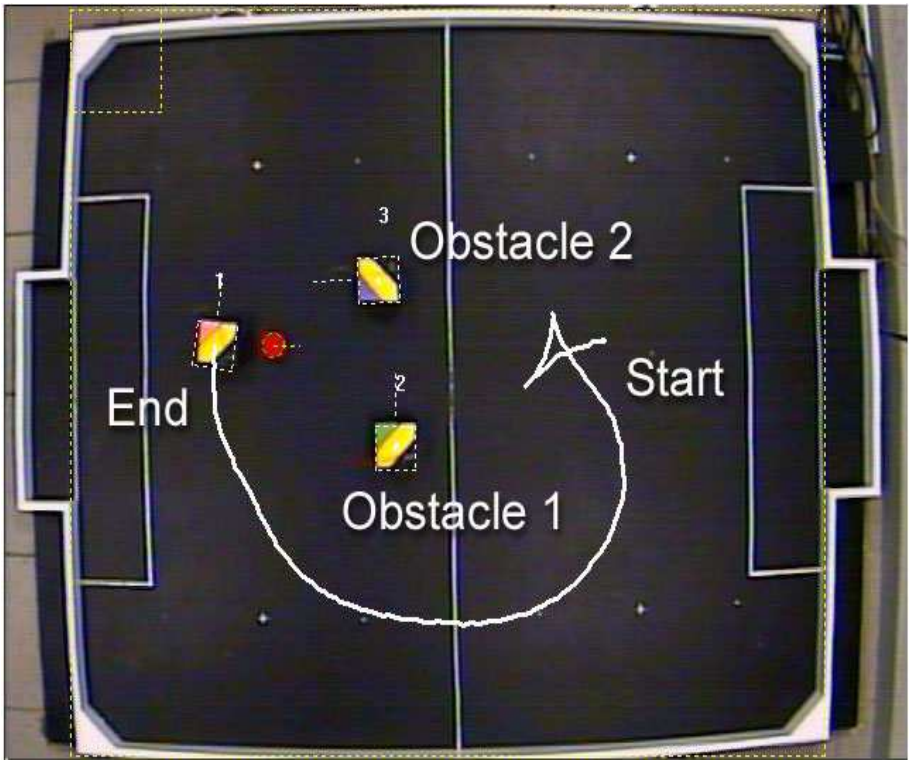


EPF-based path

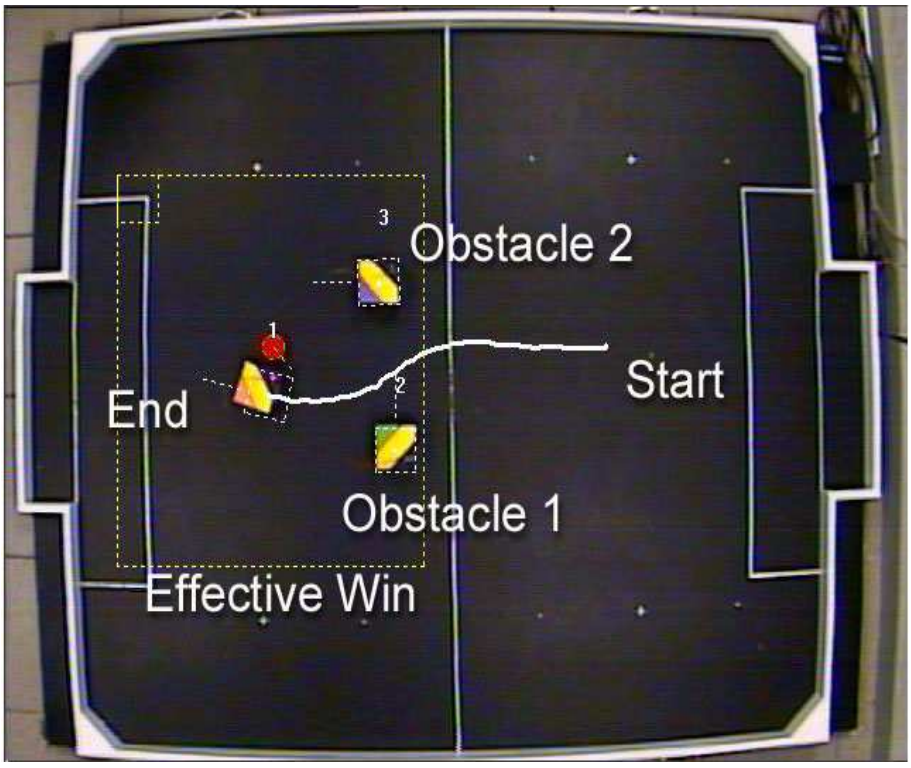


AW-EPF-based path

Figure 4.9: Case 1: Paths comparison (1 stationary obstacle)

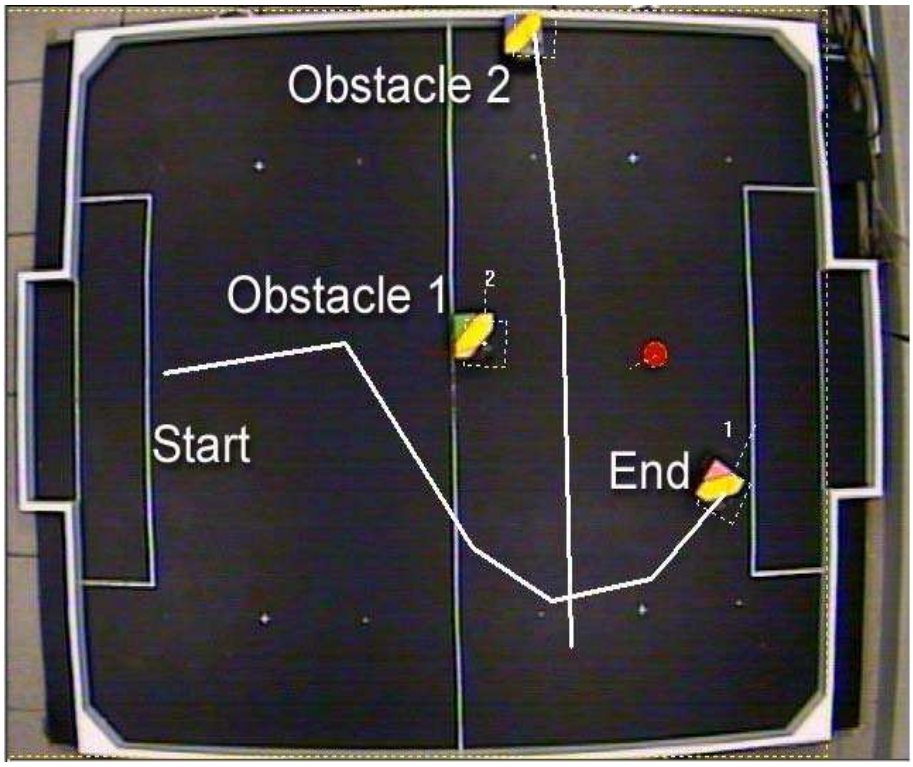


EPF-based path

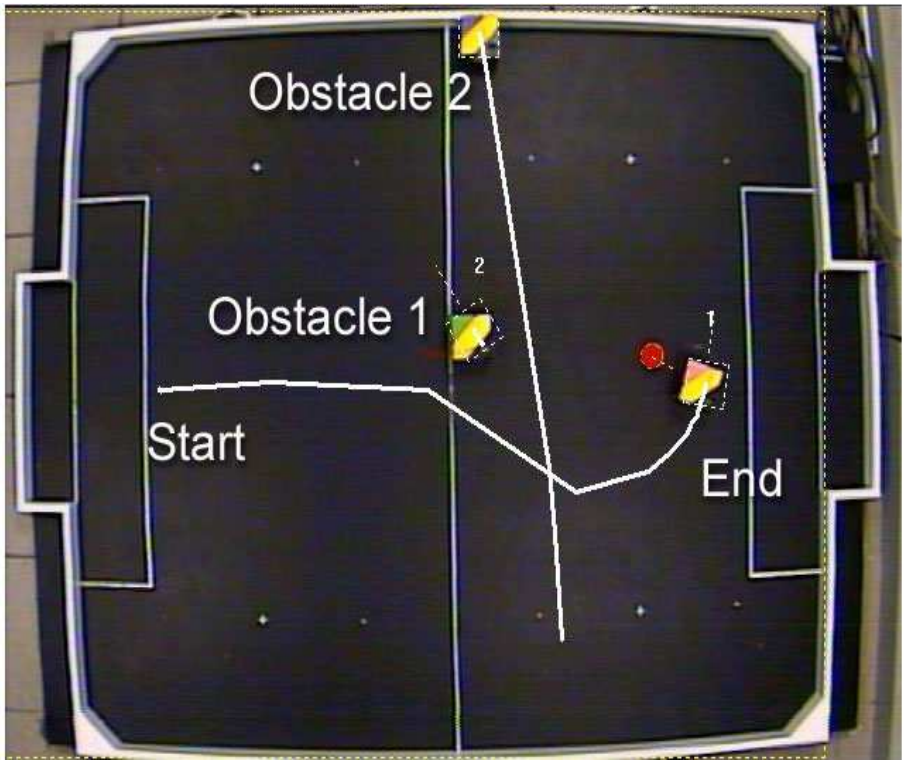


AW-EPF-based path

Figure 4.10: Case 2: Paths comparison (2 stationary obstacles)

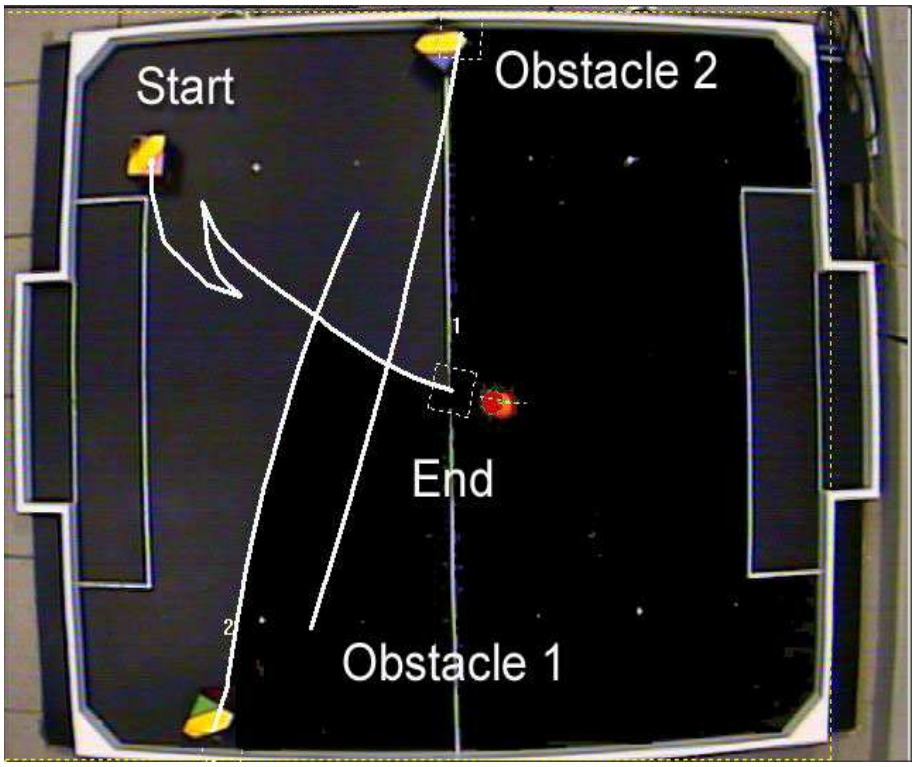


EPF-based path

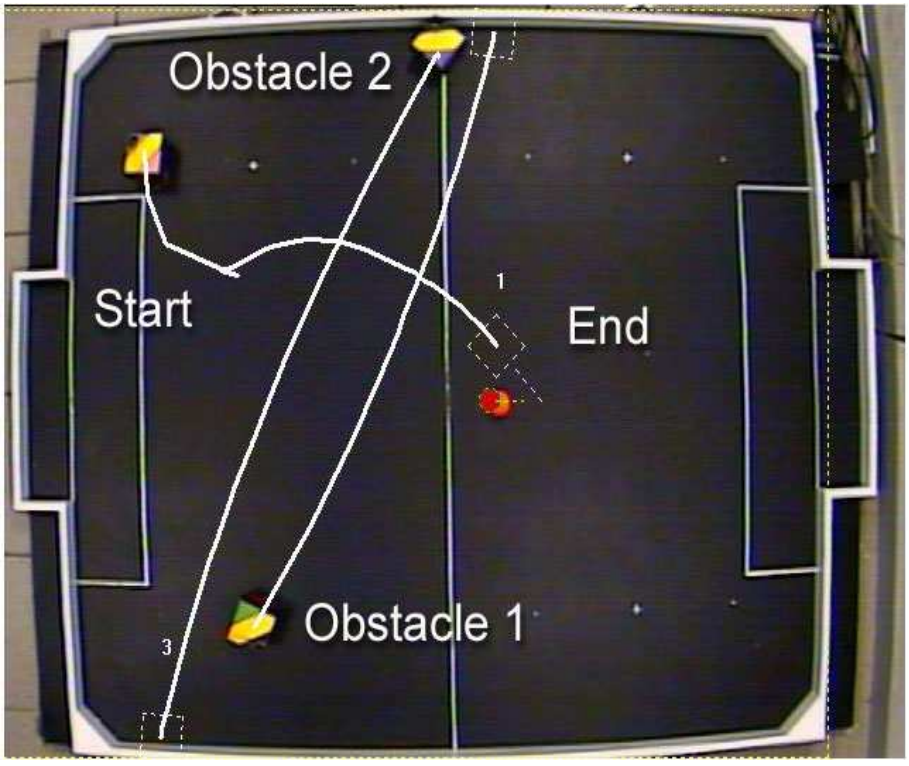


AW-EPF-based path

Figure 4.11: Case 3: Paths comparison (moving obstacle)



EPF-based path

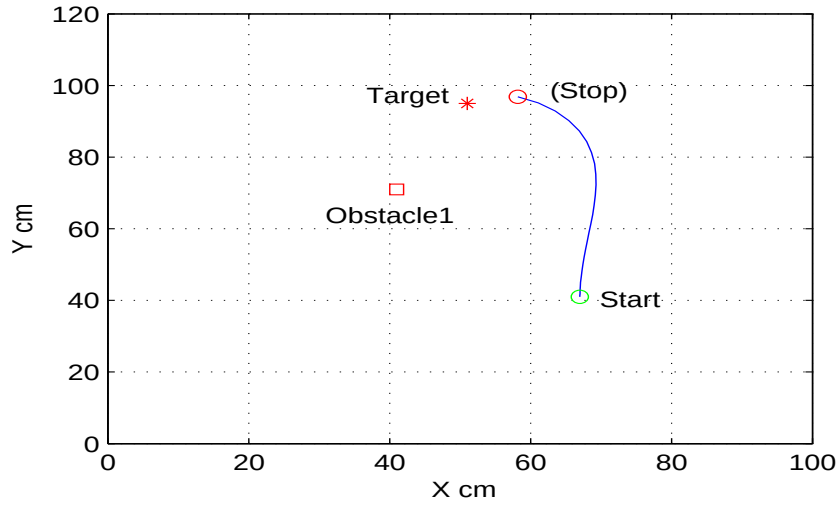


AW-EPF-based path

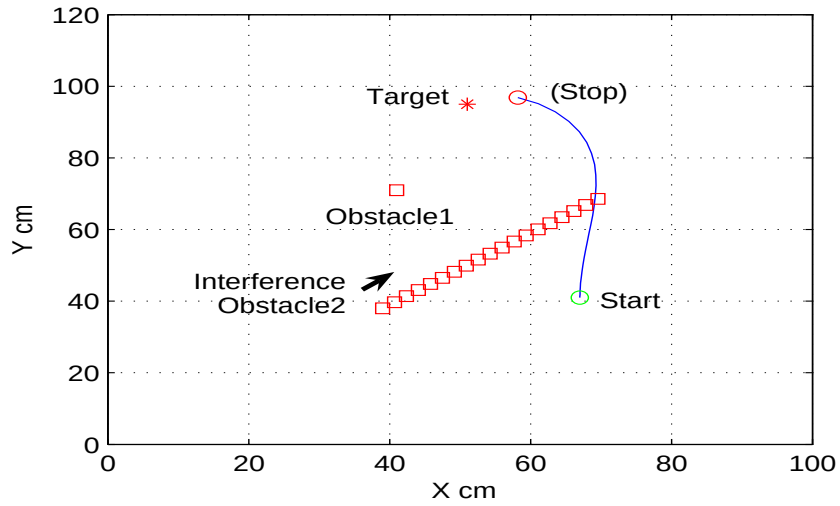
Figure 4.12: Case 4: Paths comparison (two moving obstacles)

Since the EPF approach generates a global path and is proved local minimal free in a bounded environment, AW-EPF inherits its properties within the adaptive window coverage. To study the generality of AW-EPF, consider the situations when a speedy or unforeseen obstacle suddenly appears around the moving robot, which could happen if the window does not cover the whole work space. If an obstacle is running towards the target in high speed from far away, the AW-EPF ignores the obstacle observed earlier and takes it into account when the obstacle is already quite closer to the target. As in Figure 4.13, a moving interference Obstacle2 is placed round the robot after the robot begins move to the Target, avoiding a stationary Obstacle1. The path curves more for the faster obstacle in (c) comparing the paths in (b) with a slow obstacle and the one with no interference. It is implicated that AW-EPF could react to the dynamic environment promptly with a high system refresh rate and to compensate the possible global information inefficiency.

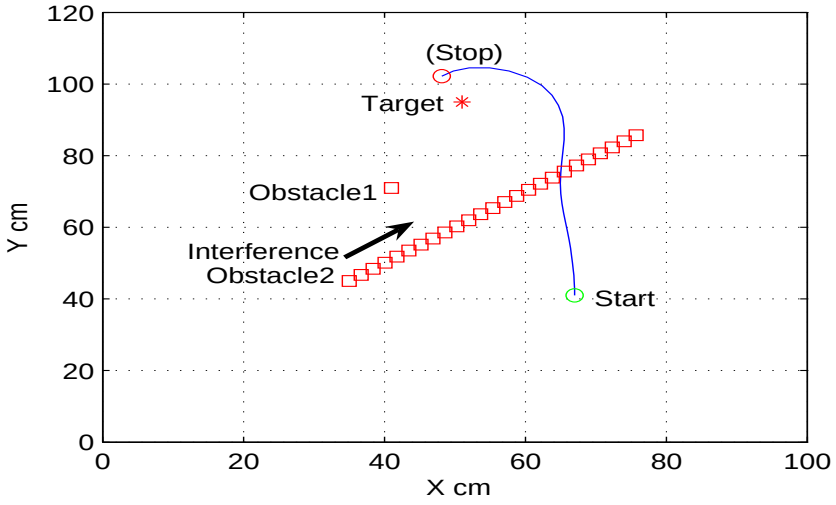
To apply this navigation method in real time with less computation and higher resolution, the adaptive window functions as a subspace of the whole field ground for the robot.



(a)



(b)



(c)

Figure 4.13: AW-EPF performances on unforeseen obstacles

Chapter 5

Evolutionary Artificial Potential Field Based Path Planning

Simple and effective navigation methodologies play important roles in mobile robot systems providing fast adaptive control in dynamic environments. Among these methodologies, the Artificial Potential Field (APF) approach is an effective and convenient path-planning method inspired from physical potential fields [16][3][101][102][103]. By combining its elegant form with other heuristic algorithms, the APF can be utilized effectively for path planning [18].

In this chapter, an Evolutionary Artificial Potential Field (EAPF) approach is proposed for multiple mobile robot real-time path planning. EAPF is an Artificial Potential Field approach which utilizes Evolutionary Algorithms to optimize the associated parameters where an artificial potential field is constructed based on the distribution of the target and obstacles. The repulsive forces from the obstacles and the attractive force towards the target drive the robot around the obstacle while moving towards the target.

To develop a more efficient potential field, the parameters of the proposed Artificial Potential Field need to be optimized. The Multi-objective Evolutionary Algorithm (MOEA) [60][61] is utilized for this purpose, as it provides a way to

obtain nearly optimal results in complex search spaces. The proposed approach is implemented in real-time for mobile robot path planning within a micro-robot soccer setup.

In the later section of this chapter, the experimental results with the multiple mobile robot system show that EAPF is a simple and effective approach for real time mobile robot path-planning.

5.1 Artificial Potential Field

Besides the logic-and-rules based traditional AI approaches, another direction in path planning is the Artificial Potential Field (APF) [97] approach which is flexible and simple to implement. APF approach is inspired from the physical potential fields. Different APFs have been suggested in various application domains from mobile robots, manipulators, and sensor networks [104][105][26][106]. For instance, APF based on heat resistance [107] and electrostatic field [96] are capable of generating the paths for robots in indoor environments. Fractional potential field was also introduced to robot path planning [108]. APF has been also successfully applied in walking robots [109] and in multiple robot environments [110][111]. In [112] the Potential Field approach is utilized for strategy decision making in cooperative robotics.

In [113] a potential field-based cooperative motion planning of a team of semi-autonomous robots is presented in an unknown environment. Gaussian functions are chosen to model the objects and the navigation function is changed with sensor data update. Researchers have developed the APF concept and utilized it for multi-robot coordination [114][115].

The main issue associated with the potential field is the local minima problem. At the local minimum position, the gradient of the potential is zero resulting in a null force input to the robot. Random policy [19] or specific designed strategies [116][117][21][118] can overcome the local minima. However some of the methods

either require a workspace without restrictions on the direction of motion, or that the algorithms are too complicated. In this work an escape force is constructed to the potential field to push the robot away from a local minima. An obstacle filter is designed to simplify the associated processing and to overcome the convex trap formed by obstacles.

In APF, the target and obstacles are assumed to radiate energy potential and the total potential field motivates the robot to the target point which has a lower potential. Usually only the potential generated by the objects in the work space is considered, avoiding the complicated algorithms or rules in path planning. By utilizing appropriate potential field functions, the potential field provides appropriate force to the robot to reach the target safely.

In this work, an APF based on Evolutionary optimization (EAPF) is built to provide the guide forces to the robot along a collision free path in real time. Several parameters are introduced to construct the artificial attractive and repulsive forces and optimized by MOEA.

As path smoothness, safety and path length play roles in the evaluation of the planned path, a multi-objective optimization algorithm is utilized to search for sub-optimal solutions. The details on MOEA is presented in section 5.4.

With the help of MOEA, the proposed EAPF is implemented on a robot soccer system. The workspace of robot soccer system is placid and continuous with fixed bounds, and APF approach can be applied for path-planning.

This chapter is organized as follows. The EAPF functions and the fitness function are presented in Section 5.2. Simulation and experimental results are included in Sections 5.5 and 5.6. Conclusion and future research direction are provided in Section 5.8.

5.2 Evolutionary Artificial Potential Field

In the Artificial Potential Field method, an obstacle is considered as a point of higher potential, and the target as a point of lowest potential.

The 2-D space (S) under-consideration is represented by a Cartesian coordinate system (Figure 5.2). Each point (x, y) in S is represented by vector $\overline{P}$. Let $\overline{P}_T$, $\overline{P}_R$ and $\overline{P}_O$ represent the vectors pointing from the origin (0,0) to the Target, the Robot, and the Obstacle respectively. Given a target (x_T, y_T) and an obstacle (x_O, y_O), the robot is attracted by an attractive force $\overline{F}_a$ towards the target and repelled by a repulsive force $\overline{F}_r$ from an obstacle. The total force $\overline{F}_{total}$ is the geometric sum of the attractive and repulsive forces.

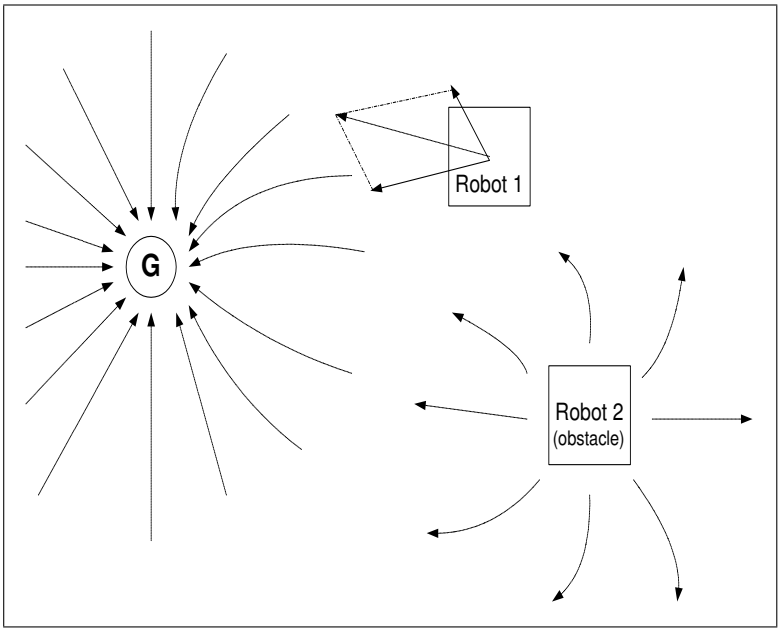


Figure 5.1: Forces in Artificial Potential Field

The attractive force $\overline{F}_a$ and repulsive force $\overline{F}_r$ are defined in (5.3) and (5.4). The magnitude of the attractive force $\overline{F}_a$ is defined as inversely proportional to the distance (D_{TR}) from the robot to the target, while the magnitude of the repulsive force $\overline{F}_r$ is inversely proportional to exponent of the distance (D_{OR_i}) between the

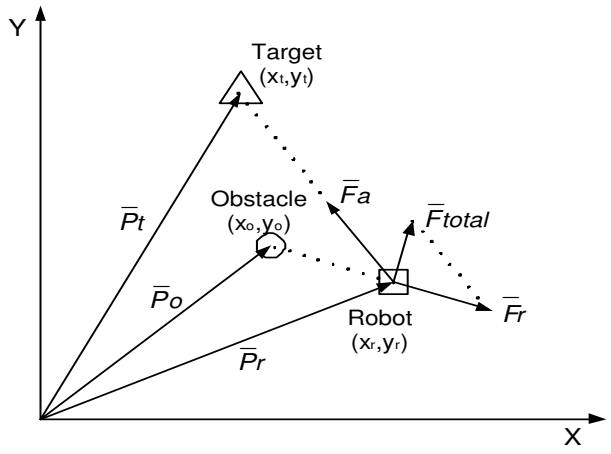


Figure 5.2: Artificial potential force illustration

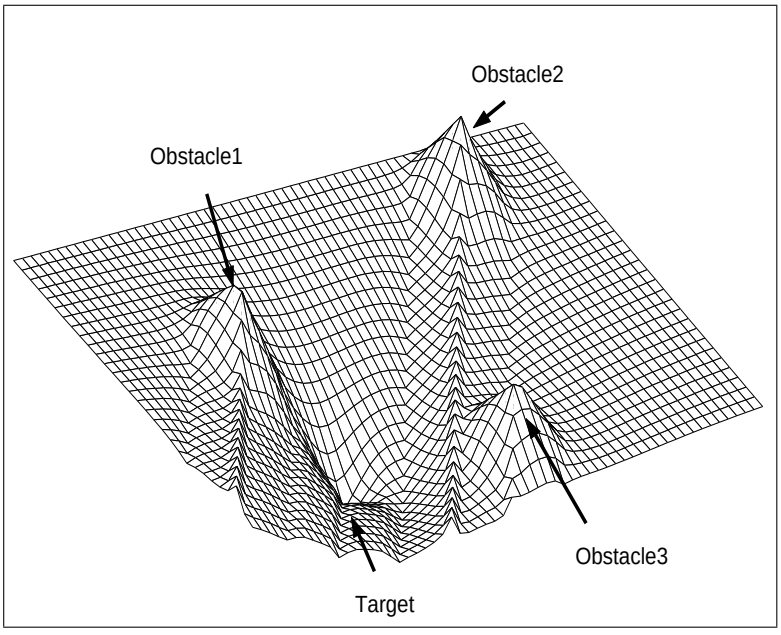


Figure 5.3: Artificial potential field distribution

robot and obstacle, where i is the index of obstacles. The radii of the robots, obstacles and the ball (r_R, r_O and r_B) are considered in the potential field generation. The sum of the radii of the robot and ball/obstacle is subtracted from the distance between the robot and the ball/obstacle in calculating D_{OR} (5.1) and D_{TR} (5.2).

The angles of the forces are defined between $-\pi$ and π. p and n in (5.4) are positive parameters to optimize. For definite j obstacles in S, $\overline{F}_r$ is the geometric sum of all the repulsive force vectors.

$$D_{TR} = \|\overline{P}_T - \overline{P}_R\| - (r_R + r_B), \tag{5.1}$$

$$D_{OR} = \|\overline{P}_R - \overline{P}_{Oi}\| - (r_R + r_O), \tag{5.2}$$

$$\overline{F}_a = D_{TR}^{-1}\angle(\overline{P}_T - \overline{P}_R), \tag{5.3}$$

$$\overline{F}_r = \sum_{i=1}^{j}(pD_{OR_i})^{-n}\angle(\overline{P}_{O_i} - \overline{P}_R), \tag{5.4}$$

$$\overline{F}_{total} = \overline{F}_a + \overline{F}_r. \tag{5.5}$$

The robot orientation is important in control implementation. Taking the heading factor into the force functions, the attractive and repulsive forces then become:

$$\overline{F}_a = (D_{TR})^{-1}(1 + \frac{\Delta\theta_a}{2\pi})\angle(\overline{P}_T - \overline{P}_R), \tag{5.6}$$

$$\overline{F}_r = (pD_{OR})^{-n}(1 + \frac{\Delta\theta_r}{2\pi})\angle(\overline{P}_O - \overline{P}_R). \tag{5.7}$$

where, $\Delta\theta_a$ is the error between the robot orientation and the attractive force direction. The force is enforced by $(1 + \frac{\Delta\theta_a}{2\pi})$ when the robot is not facing the right direction. Similar modification on the repulsive force is brought out with $\Delta\theta_r$ which is the error between the robot orientation and the repulsive force direction.

When the attractive and repulsive forces balance out, the robot is trapped in local minimum. To avoid this, an escape force $\overline{F}_e$ (5.8) is utilized [20], when $\frac{\|\overline{F}_a\| - \|\overline{F}_r\|}{\|\overline{F}_r\|} < b$ and $\cos(\angle\overline{F}_a - \angle\overline{F}_r) < -\cos(c)$, where b, c, d and m are positive parameters.

$$\|\overline{F}_e\| = \frac{1}{dD_{RO}^m}(\Big|\cos(\angle\overline{F}_a - \angle\overline{F}_r) - \cos(c)\Big|) \tag{5.8}$$

In this work, the repulsive force is taken into account in the definition of the escape force (5.9) (5.10). when $\frac{\|\overline{F}_a\|-\|\overline{F}_r\|}{\|\overline{F}_r\|} < b$ and $|\angle\overline{F}_a - \angle\overline{F}_r| < \pi/m$, where b and m are positive integer variables to optimize, and π/m refers to a certain radian. Hence when the robot begins to enter a possible local-minimal area where the force sum is close to zero, the escape force is triggered to pull the robot out of the trap. The condition to apply an escape force is defined from the magnitude relationship of the attractive force and the repulsive force by b and from their direction relationship by m.

$$\|\overline{F}_e\| = \|\overline{F}_r\| \Big| \cos(\angle\overline{F}_a - \angle\overline{F}_r - \pi) - \cos(\pi/m) \Big| \tag{5.9}$$

$$\angle\overline{F}_e = \begin{cases} \angle\overline{F}_r + \pi/2 & \text{if } \sin(\angle\overline{F}_a - \angle\overline{F}_r) > 0 \\ \angle\overline{F}_r - \pi/2 & \text{otherwise} \end{cases} \tag{5.10}$$

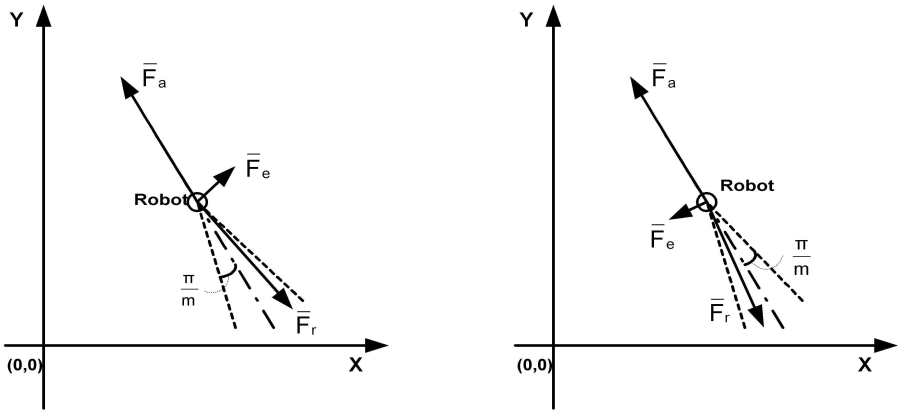


Figure 5.4: Escape force direction determination

When the difference in the attractive and repulsive force angles is less than π/m, falling within a narrow part between the dashed lines in Figure 5.4, the geometric sum of the forces is possible to be close to zero and the potential force might be too small to command the robot to move. The escape force is larger when the difference in the attractive and repulsive force angles is closer to π/m.

The magnitude of the escape force is proportional to the magnitude of the repulsive force and the escape force is large enough to push the robot away from the nearly local minimum position. In the proposed escape force definition, the number of parameters to optimize is reduced from 4 in [20] to 2. In [20], the direction of the escape force is not clearly defined, while the direction of the proposed escape force is provided by (5.10). $\angle\overline{F}_e$ is designed perpendicular to the repulsive force so that the escape force enables the robot to move away from the local minimum swiftly to leave the potential trap, and the positive projection on the direction to the target helps the robot to follow a right direction in movement. The total potential force $\overline{F}_{total}$ is the geographic addition of attractive force, repulsive force and escape force,

$$\overline{F}_{total} = \overline{F}_a + \overline{F}_r + \overline{F}_e. \tag{5.11}$$

Hence there are 4 parameters to be optimized from (5.7),(5.9), (5.10): p, n in the repulsive force and, b, m in the escape force. These parameters determine the distribution of the artificial potential field and affects the robot's path in terms of force magnitude. The values of these parameters are crucial in improving the performance of the path planning algorithm. Since the search space of the parameters is large and it is difficult to find an ideal theoretical solution for this problem, Evolutionary Algorithm is utilized for the parameter optimization.

5.3 EAPF Parameter Analysis

Obviously the parameters involved in potential field configuration affect the field layout. In a simulation setup the effect of various parameters associated with EAPF are studied.

The robot paths for different values of p within the range [0.1, 0.5] are depicted in Figure 5.5 to Figure 5.8. Stationary robots of the same size are the two obstacles nearby.

With the increase of p, the repulsive force from obstacles decrease exponentially,

i.e. the field surface became more smooth. The augment of n has a similar function, with a decrease in the repulsive force. As result, the robot follows a path which can be closer the obstacles.

A smaller p (Figure 5.11) resulted in a larger repulsive force, making the robot path longer. However, when p increases, the parameter's influence fades as the repulsive force diminished to a large extend. For the p parameter, the curvature of the path increases with decreasing of p, while the repulsive force increases. However, when $p > 0.3$, its influence fades because the repulsive force becomes bitterly small. From the view of potential fields, the degree of convex surface decreases with the increase of p, the peak height becomes lower and the influenced area by the convex surface becomes smaller. The same parameter value may lead to different paths in various situations and it is a tradeoff between safety and path length to select an appropriate value.

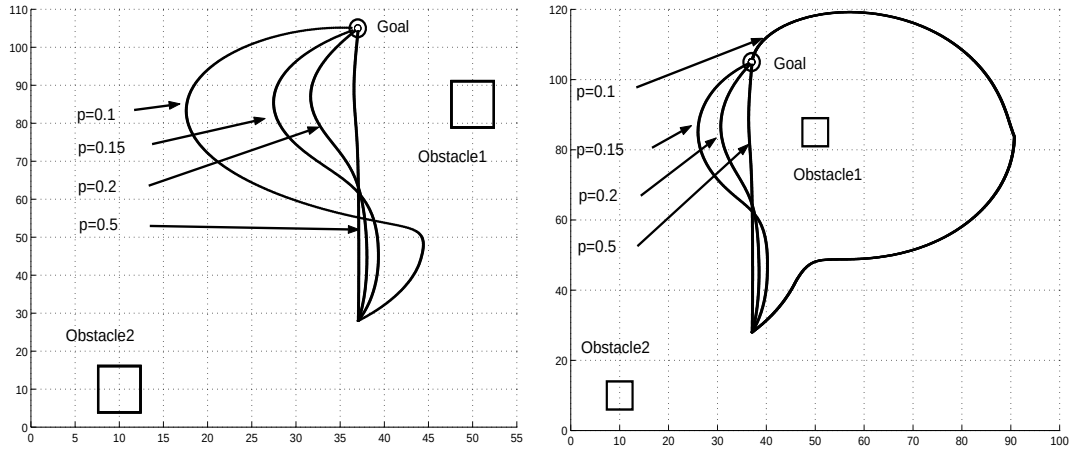
n has more influence on the robot path as it is in the exponent power. b and m influence to the path is activated only when escape force is needed. Bigger b means the escape force is enabled with higher possibility, while bigger m means larger escape force.

When p is 0.5 and above the robot path remained unaltered. Bigger values for n impart smaller repulsive forces (Figure 5.5). Considering the dynamic obstacles in real time implementation, smaller values are preferable for both n and p.

The same parameter set may lead to various cost (path length) in various scenarios, e.g., different robot speeds may lead to different paths. A tradeoff between safety and path length determines the appropriate parameter values for use in different scenarios. Within the possible range of parameters, it may require to fine-tune the parameters.

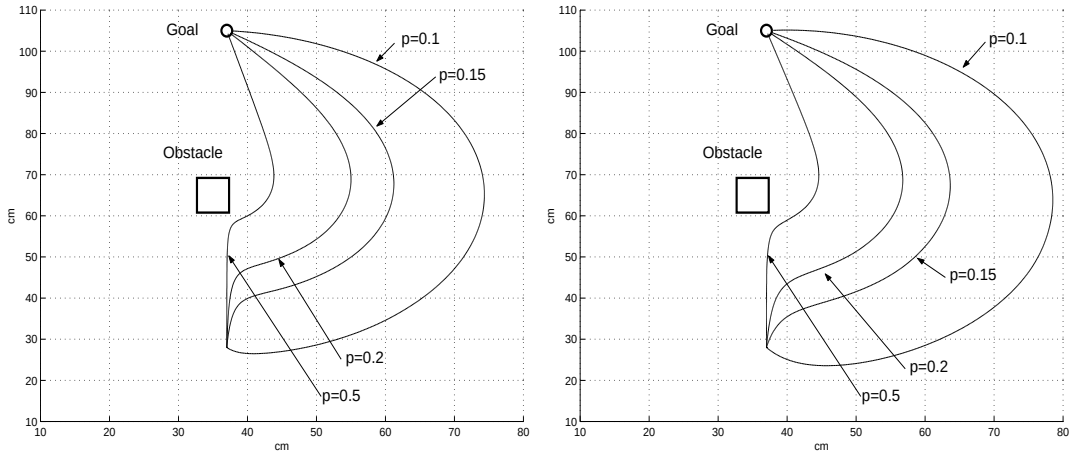
The parameter n has a large influence on the robot path being an exponential in the repulsive force vector (5.4). The influence of parameters b and m to the path are activated only when the escape force is needed. The path generated by different m values are alike.

From the simulation results, it is noted that the parameters p and n play important roles, while b and m are less important as they are only effective around local minima.



$(n = 3.2, b = 8, m = 10)$ $(n = 3, b = 5, m = 12)$

Figure 5.5: Simulated robot trajectories with different p value

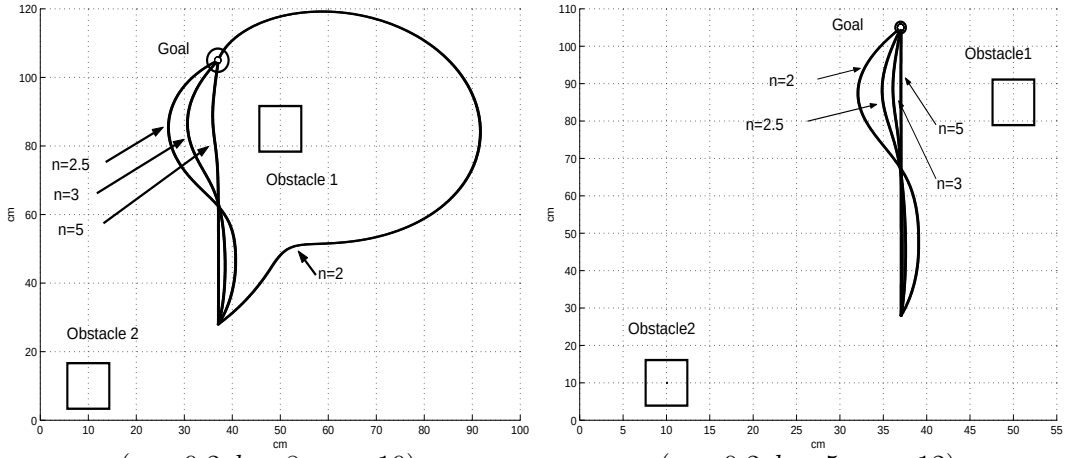


$(n = 3.2, b = 8, m = 10)$ $(n = 3, b = 5, m = 12)$

Figure 5.6: Simulated robot trajectories with different p value

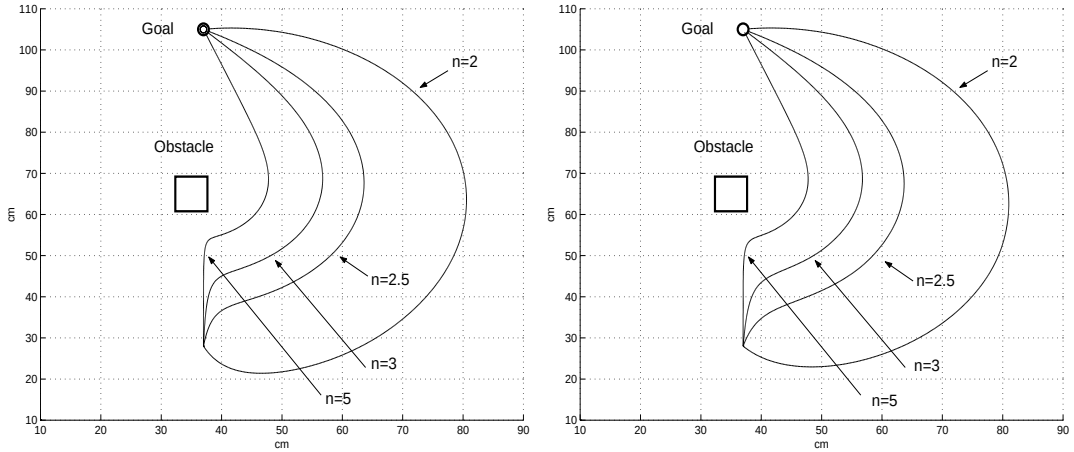
5.4 Parameter Optimization based on MOEA

As path smoothness, safety and path length play roles in the evaluation of the planned path, a multi-objective optimization algorithm is utilized to search sub-optimal solutions. The Multi-objective Evolutionary Algorithm (MOEA) is utilized



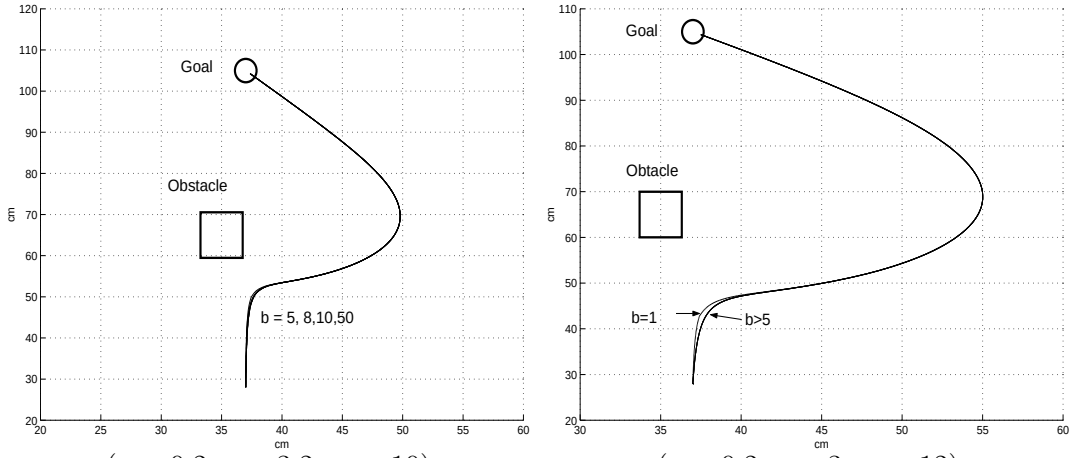
$(p = 0.2, b = 8, m = 10)$ $(p = 0.2, b = 5, m = 12)$

Figure 5.7: Simulated robot trajectories with different n value



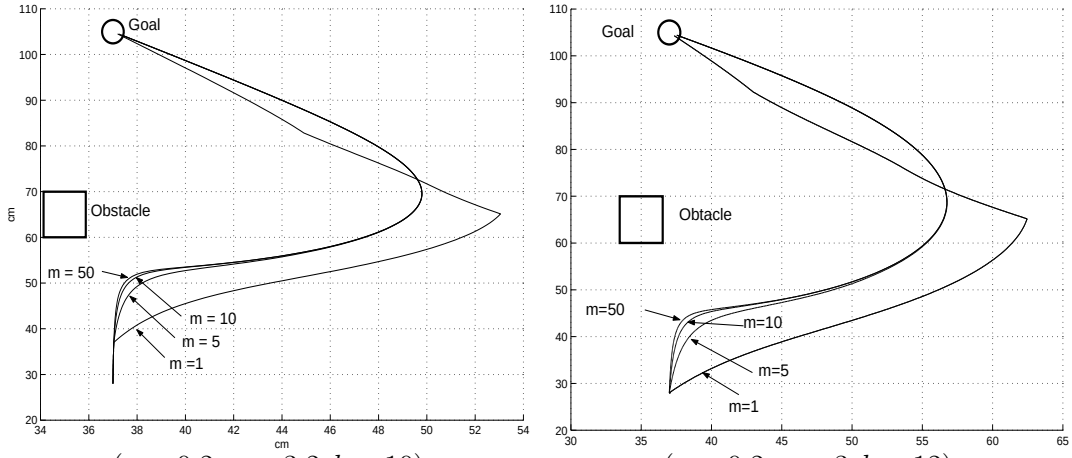
$(p = 0.2, b = 8, m = 10)$ $(p = 0.2, b = 5, m = 12)$

Figure 5.8: Simulated robot trajectories with different n value



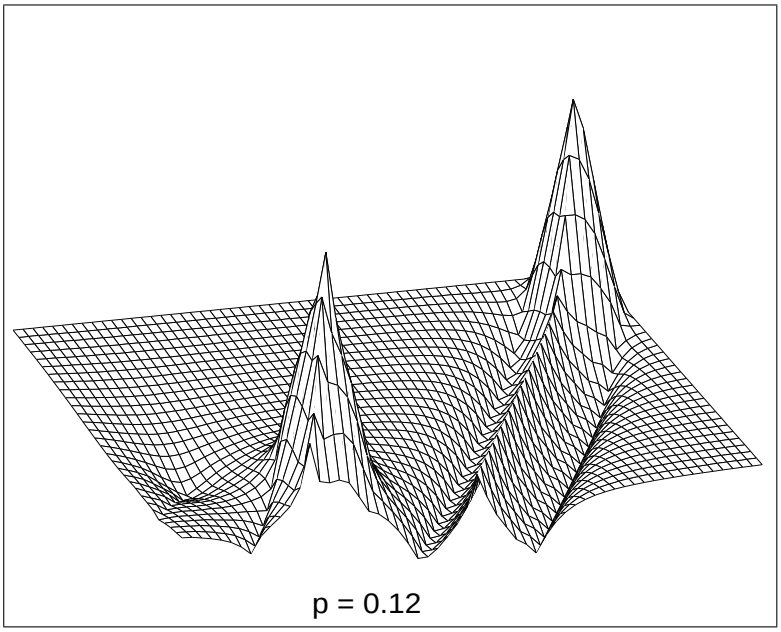
$(p = 0.2, n = 3.2, m = 10)$ $(p = 0.2, n = 3, m = 12)$

Figure 5.9: Simulated robot trajectories with different b value

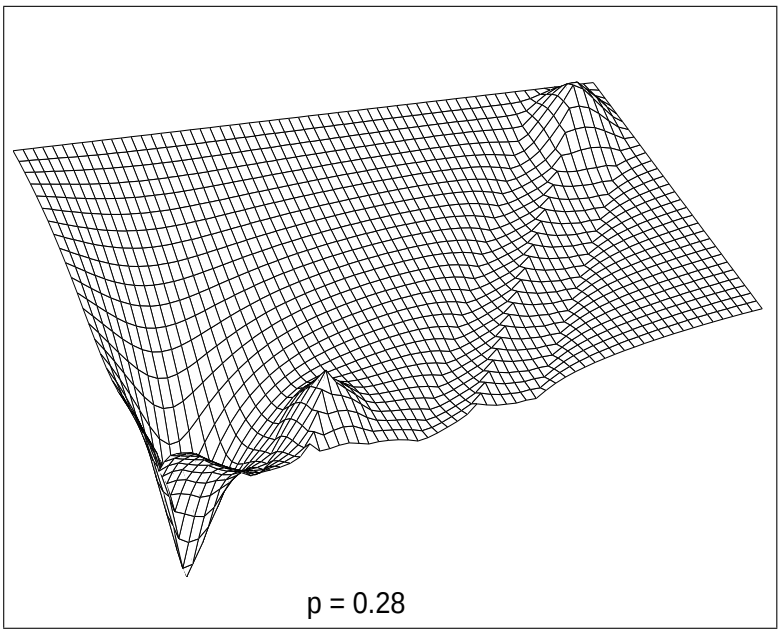


$(p = 0.2, n = 3.2, b = 10)$ $(p = 0.2, n = 3, b = 12)$

Figure 5.10: Simulated robot trajectories with different m value

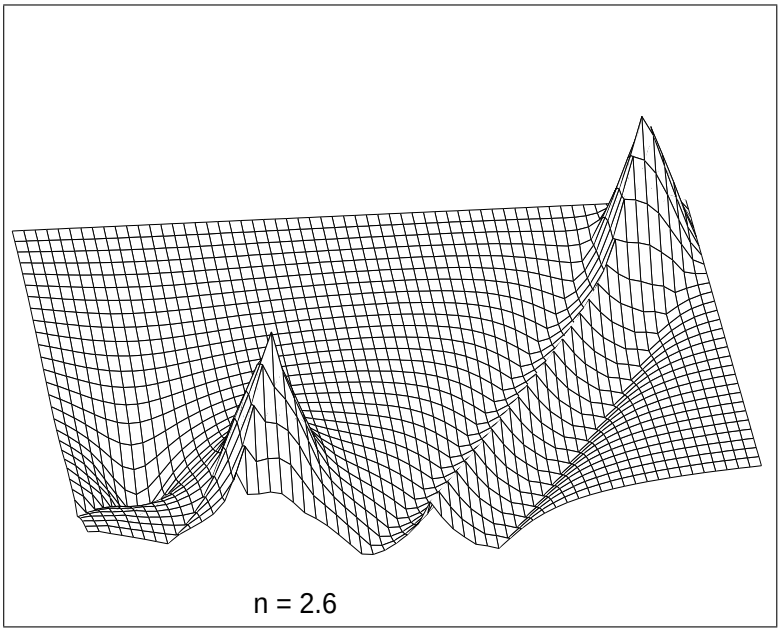


(a)

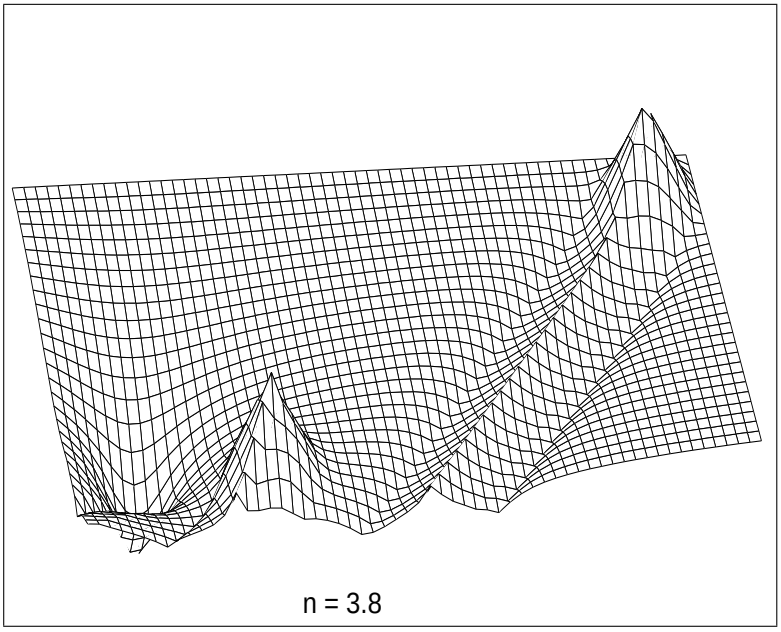


(b)

Figure 5.11: Potential distributions for different p values



(a)



(b)

Figure 5.12: Potential distributions for different n values

to handle the parameter optimization for being powerful in multi-objective problems. Evolutionary Algorithm (EA) is based on models of organic evolution in which initial problem solutions are modelled into evolution to generate satisfactory solutions, and it has been developed decades back and successfully applied in optimization problems in many fields [119], [120], [121], [122], [123]. At the beginning of evolution process, an initial set of populations (solutions) generate offsprings through genetic operations like cross-over, mutation and selection and, the evolution stops when the fitness values of the offsprings reach an expected value. With a single objective, the solution with the lowest fitness is preserved, however, for multiple objectives, the solutions are ranked and those not inferior to any other are preserved. Since 1985 [124] researchers have been developing MOEA techniques to solve multi-objective problems (MOP) and this research field has attracted the interest of both EA researchers and engineers. MOEA [125], [126], [127] has been applied successfully in various engineering fields like filter design [62], pattern identification [128] and robot control [20], [23], [129]. There are many MOEA techniques which could be classified into three categories [130]: a) Aggregating approaches for single objective EA, b) approaches based on Pareto dominance and c) the rest of the approaches. Aggregating approaches are derived from the naive idea that multiple objectives can be combined to a single objective. This method works well if the objectives can be linearly combined or in other appropriate ways. However, it is difficult in many real world problems to find such an aggregation method. As a result, approaches based on special treatment techniques are developed [124], [131], [132].

The use of Pareto optimality in population fitness assignment is suggested in [133]. The approaches based on Pareto selection have become a significant branch of MOEA [134], [130], [135]. In Pareto dominance based approaches, the string in population is mainly ranked by the number of strings it dominates. To keep the diversity of population and to avoid premature convergence, niching, fitness sharing and restrictive mating techniques are proposed in [136], [137], [138], [139].

The Multi-Objective Evolutionary Algorithm (MOEA) is utilized to optimize

the parameters, p, n, m and b, associated with the potential field functions in simulation before implementing on to the robot soccer system.

The MOEA toolbox [140] incorporates Pareto cost assignment scheme and other complementary features of hard constraint specification for constraint handling, dynamic population size, fuzzy boundary local perturbation with interactive local fine-tuning, a novel switching preserved strategy and convergence representation for multi-objective optimization. The users only needs to write the model files and GUI windows provide easy-to-understand monitor information.

The MOEA tool involved in this paper used the Elistist strategy, where the best strings from the previous generation (strings with rank one, for multi-objective) are added to the new population. Then the population goes through evaluation and the excess, inferior strings are removed. With this strategy, good strings are always preserved. The new population is then fed to the model file to get the new cost functions. In case that too many strings are equal, say they are all ranked one, the cost after niching is considered.

Figure 5.14 and Figure 5.15 are the MOEA settings and progress ratio in evolution.

The robot is expected to approach the target point along a collision free path. Let Γ denotes the pool of points the robot passes through. The fitness functions defined (5.12 - 5.14) form the multiple objectives. They may influence the control policy and evolutionary program techniques [141] [142].

$$c_1 = min(D_{TR}), \tag{5.12}$$

$$c_2 = \begin{cases} 0 & min(D_{OR}) > D_{safe} \\ D_{safe} - min(D_{OR}) & otherwise \end{cases}, \tag{5.13}$$

$$c_3 = \int_{\Gamma} ds. \tag{5.14}$$

The function c_1 (5.12) calculates the penalty value associated with the distance

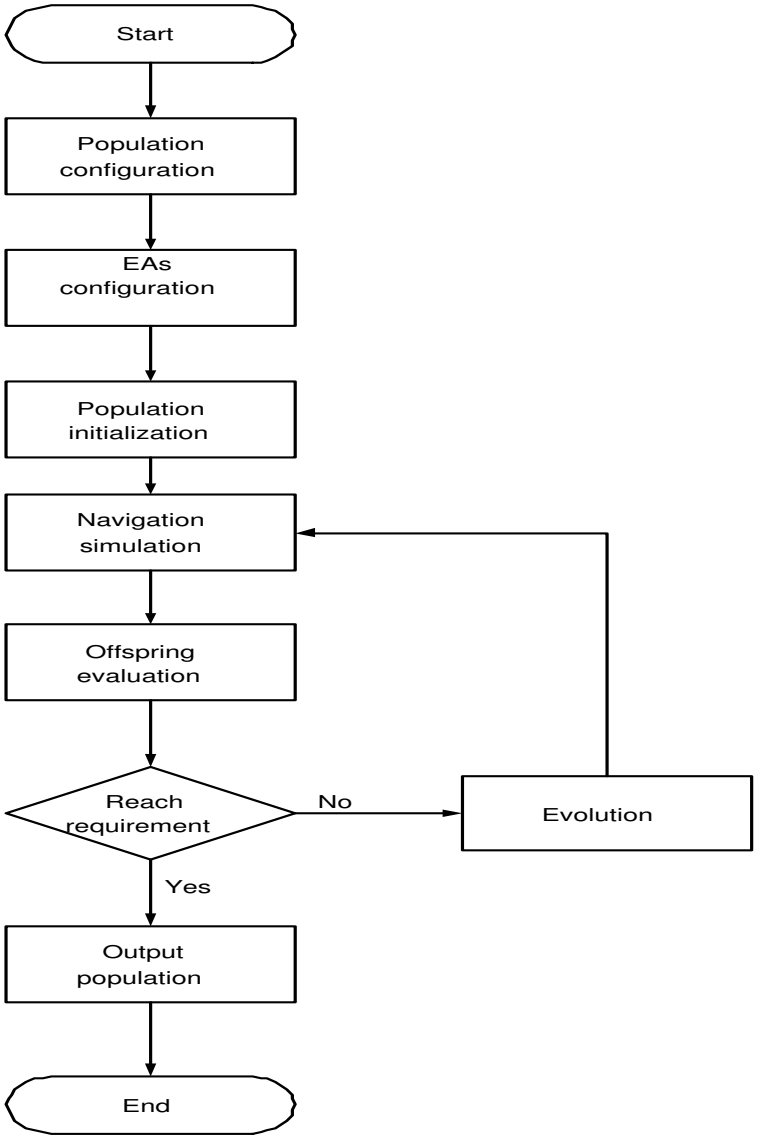


Figure 5.13: Evolution Algorithm procedures flowchart

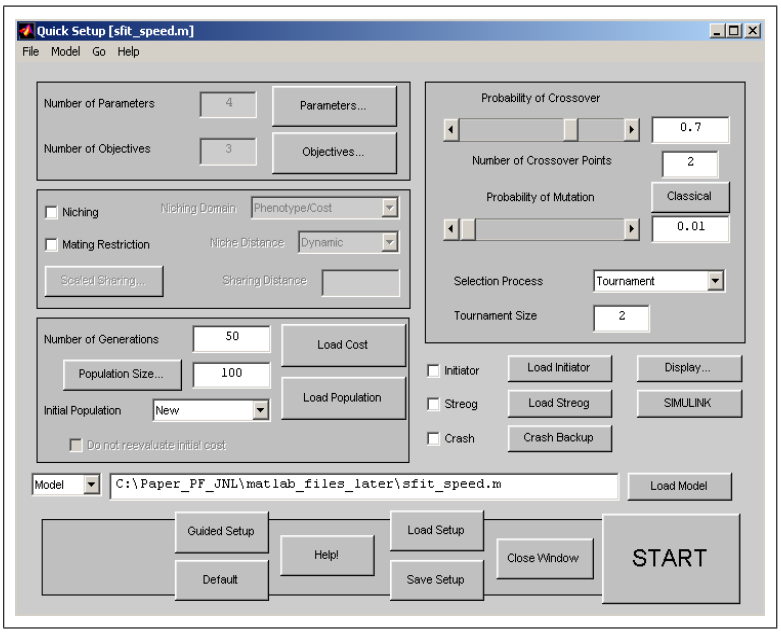


Figure 5.14: MOEA setting

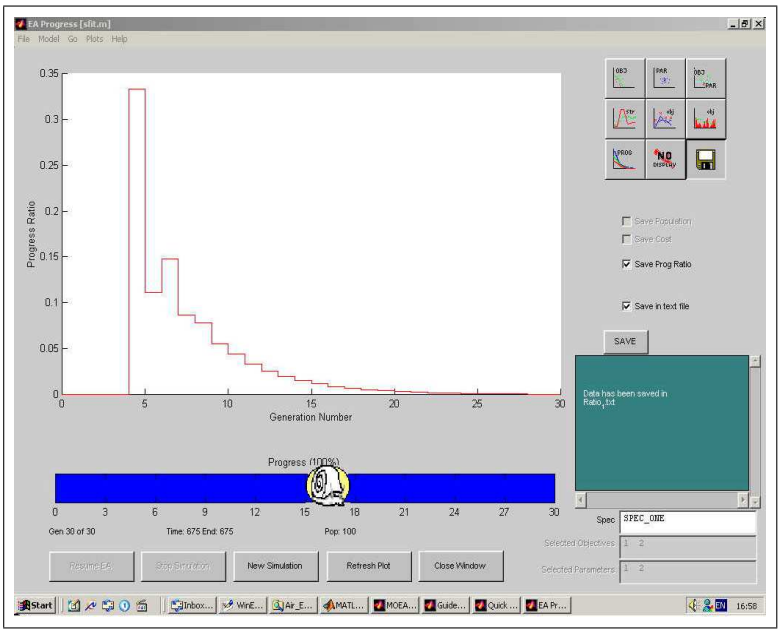


Figure 5.15: Evolution progress ratio

between the robot and target (D_{TR}). c_1 stands for the minimal distance between the robot and the target, and is 0 when the robot is at the target point. c_2 (5.13) denotes the penalty value upon collision and is 0 when the distance (D_{OR}) between the robot and obstacle is more than D_{safe}. When the robot keeps a safe distance from the obstacle ($D_{OR} > D_{safe}$) c_2 is 0. D_{safe} is defined as the accepted safe value for D_{OR}. D_{safe} is considered as the sum of the radii of the circles that respectively contain the robot and the obstacle under consideration. A 10% of the above sum is added on as a safe gap. If $D_{OR} \in (0, D_{safe})$, the robot is very close to the obstacle. c_2 is then the difference between D_{safe} and the shortest possible D_{OR} when the robot is closer to an obstacle. A larger c_2 is more dangerous. The third objective c_3 (5.14) denotes the path length.

c_2 and c_3 conflicts sometimes when the robot path is safer with farther distance to the obstacle, and the whole path length might be increased. During the evolution configuration, the costs could be associated with priorities, c_2 is set with higher priority, Figure 5.16 shows that more population with low c_2 would survive; vice vise, if c_3 is set higher priority more population with low c_3 in Figure 5.17. The Y axle is the normalized value of each cost. During the optimization, the priority configuration could influence the results the robot motion.

The influence of the parameters on potential field distribution is discussed in this section with reference to the simulation.

To ensure the generalization of optimization results, different situations of initial states of the objects are considered to evaluate each candidate, i.e., multiple situations with the speeds and positions of robot, target and obstacles, including the number of obstacles, are designed in each evaluation loop.

5.5 Simulation Results

In the simulation setup the robot and obstacles are represented by circles that can contain a robot as introduced in Chapter 2.

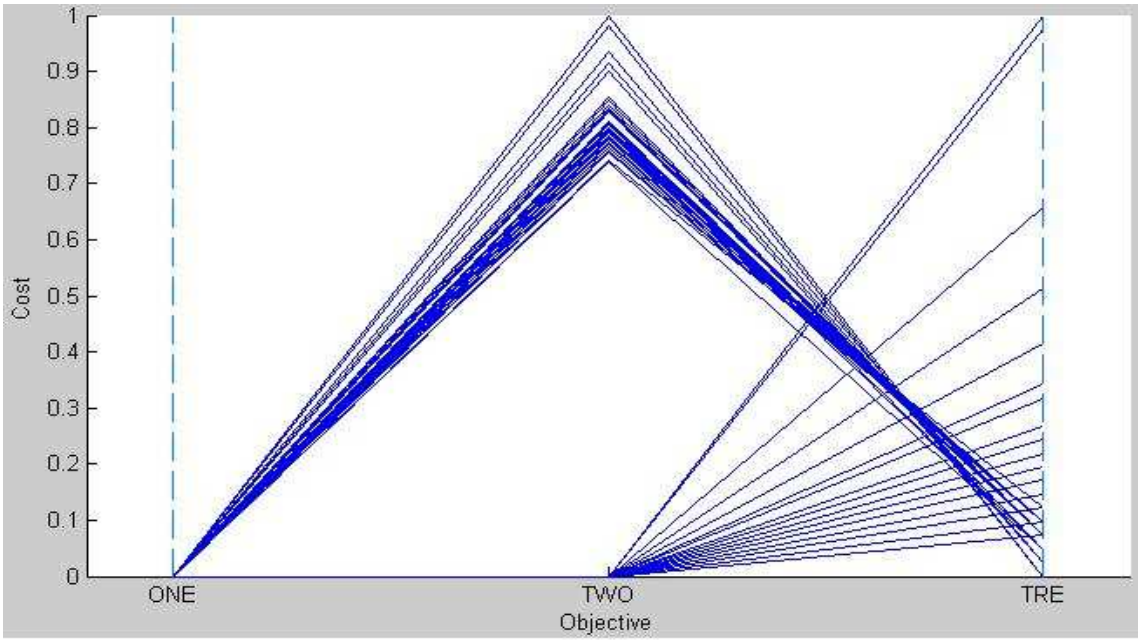


Figure 5.16: Population distribution with higher priority of safe

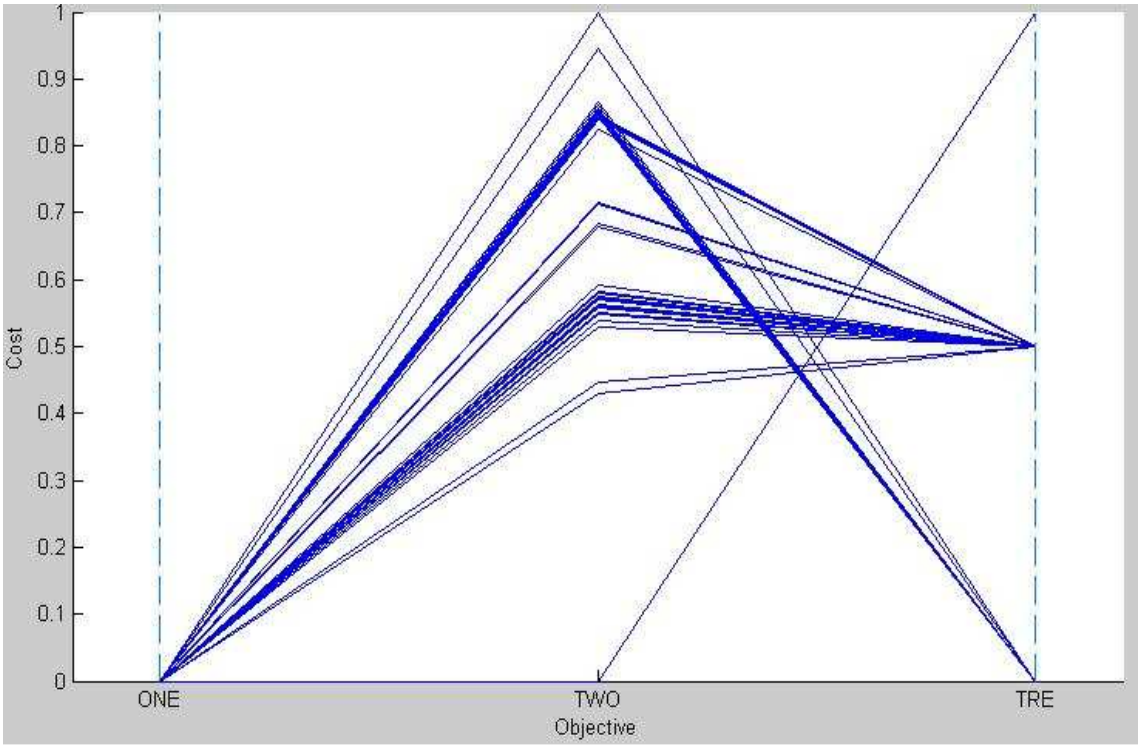
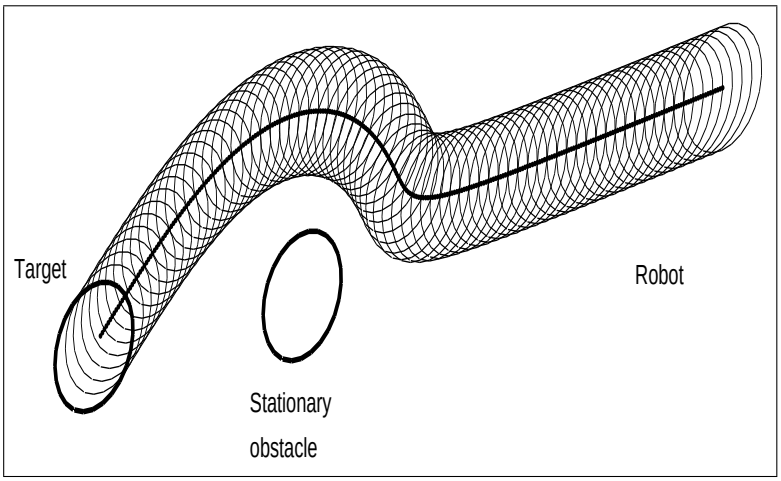
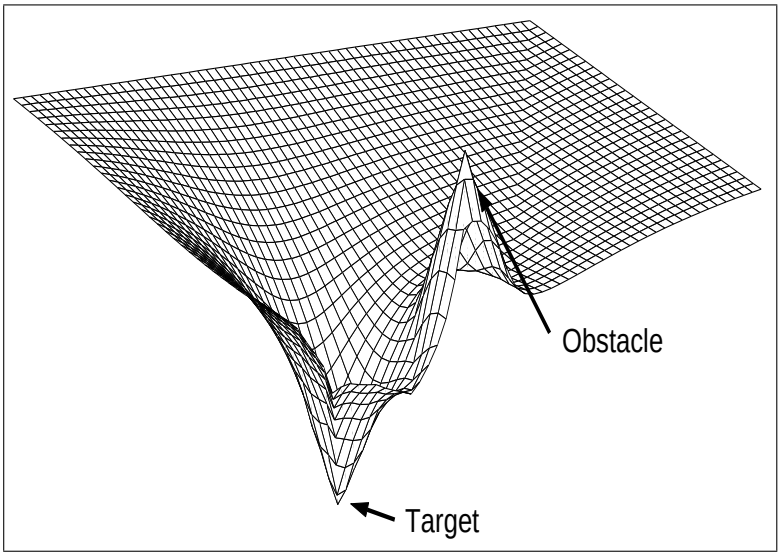


Figure 5.17: Population distribution with higher priority of path length



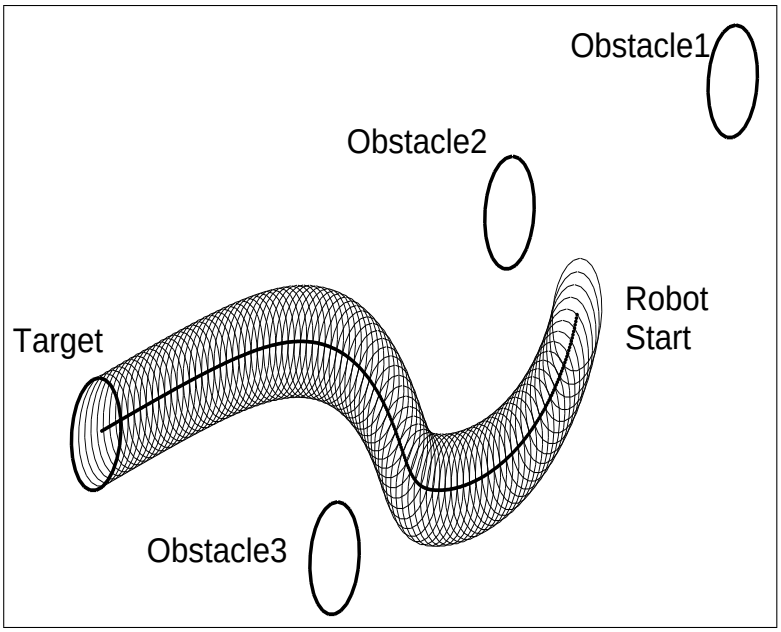
(a) Trajectory



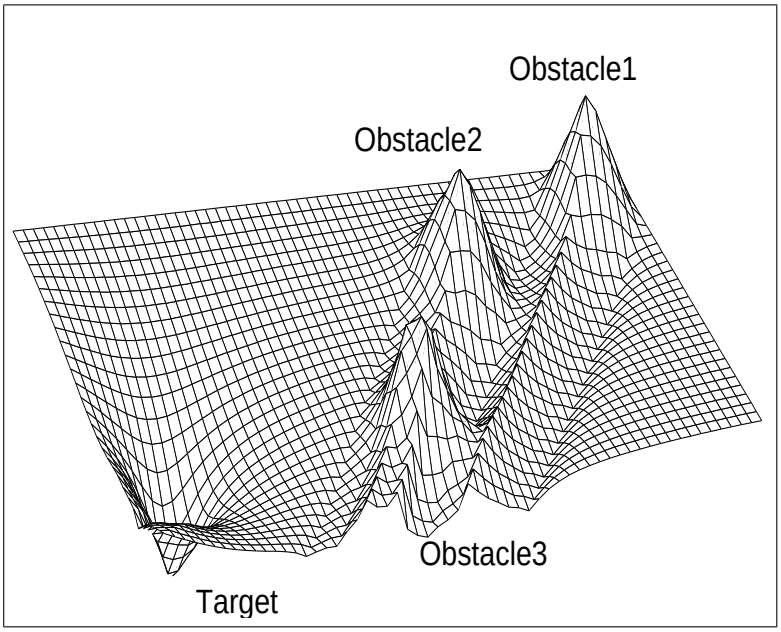
(b) Potential Field

Figure 5.18: Robot avoiding one stationary obstacle

.



(a) Trajectory



(b) Potential Field

Figure 5.19: Robot avoiding multiple obstacles

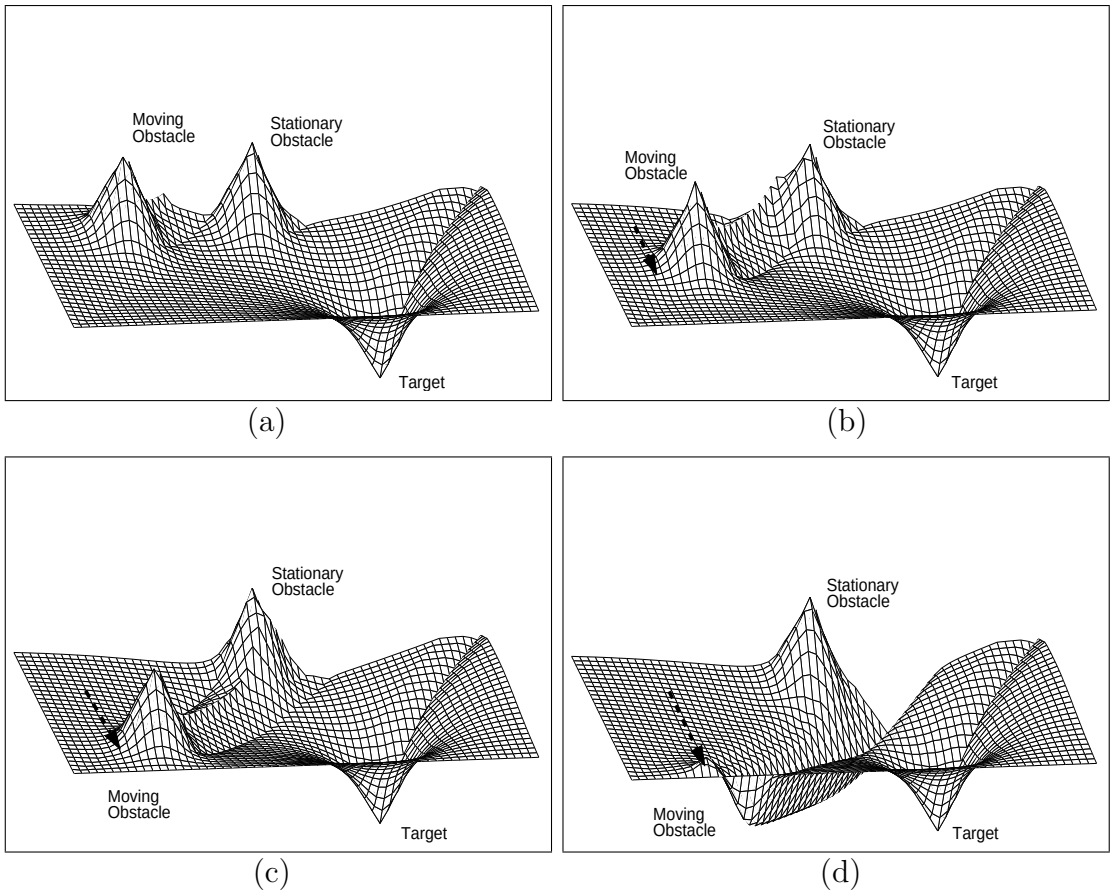


Figure 5.20: Robot avoiding moving obstacle (the moving obstacle starts from the initial position at (a), end at (d))

In Figure 5.18, the robot heads for a stationary target behind a stationary obstacle. The sub-figure on the right (Figure 5.18) is the potential distribution. Following the gradient of the potential, the robot moves to the target. The field remains unchanged as both the target and obstacle are stationary.

The trajectories and potential field distribution with three stationary obstacles are plotted in Figure 5.19. In Figure 5.20, one of the obstacles moves along the dashed line affecting the potential field. The potential fields at different motion phases are plotted to illustrate the way in which the potential field is modified.

Table 5.1: Obstacle Filter Rules by distances and speeds

Speed	Distance	State
Fast	*Far*	Middle
Fast	*Middle*	Middle
Fast	*Near*	Danger
Middle	*Far*	Safe
Middle	*Middle*	Middle
Middle	*Near*	Danger
Slow	*Far*	Safe
Slow	*Middle*	Safe
Slow	*Near*	Danger

5.6 Experimental Results

The robot soccer system (Chapter 2) is the test bed used for EAPF. An adaptive window is constructed to filter the obstacles far away the robot's current position (Figure 4.4). Only obstacles inside the window and within a certain radius are handled as valid obstacles in the potential field. The experiments validate the effectiveness of such a approach.

The optimization is applied to obtain appropriate parameters of the potential field functions.

It is noted that the potential field should adapt to different scenarios. When the obstacle is slow and relatively far, the repulsive force could be gentler, while in the contrast cases, the repulsive force should be larger to avoid collision. A Fuzzy Logic based controller (FLC) is designed to determine current status: safe, moderate and dangerous and three sets of parameters are applied for each respective status.

The FLC inputs are the distance and relative velocity between the mobile robot and the critical obstacle. The FLC rules are shown in Figure 5.21.

EAPF is utilized in the robot soccer system for obstacle avoidance. The play field of the robots is mapped into a 2-D coordinate system. Positions of the robots and ball are represented by respective X-Y coordinates (Figure 5.2). The center

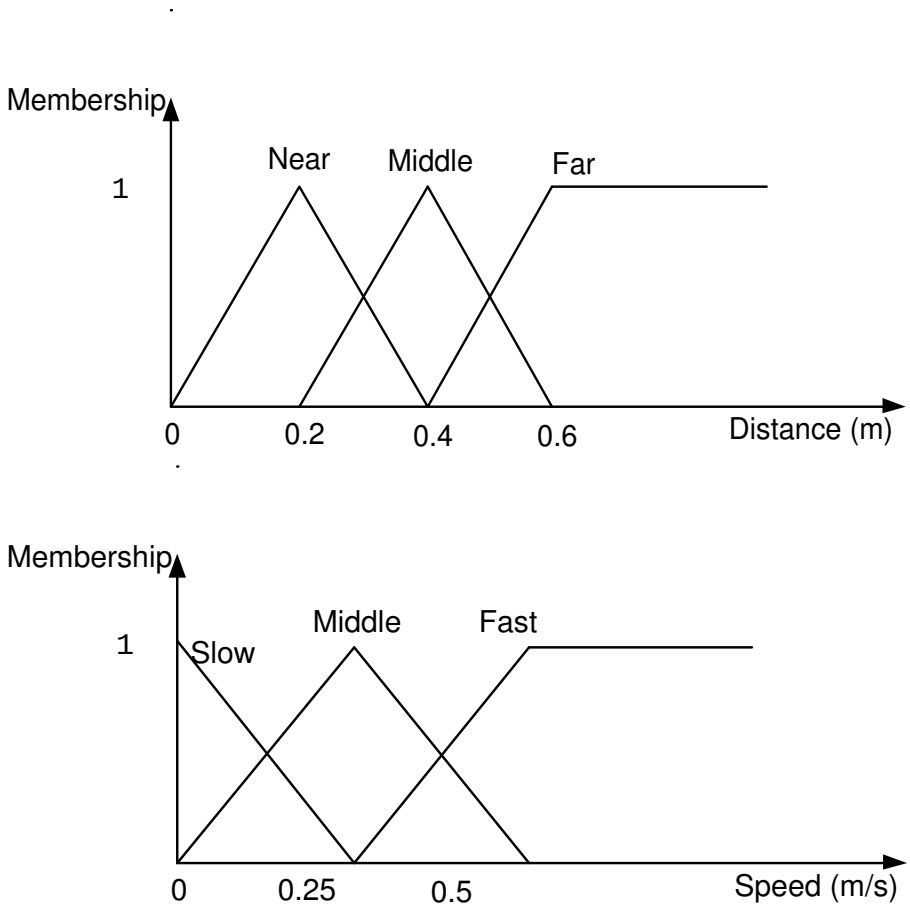


Figure 5.21: Membership functions of linguistic variables for Fuzzy logic rules in judging current status (safe, moderate, dangerous).

point of the robots, obstacles and the ball represent their positions.

In Figure 5.22, Robot1 passes through the obstacles when the empty space is larger than one and half size of the robot. Robot1 started from the upper left position of the field and went straightly ahead through Obstacle1 and Obstacle2. When Robot1 was closer to Obstacle1 and Obstacle2, it turned across the empty space between them and headed for the target.

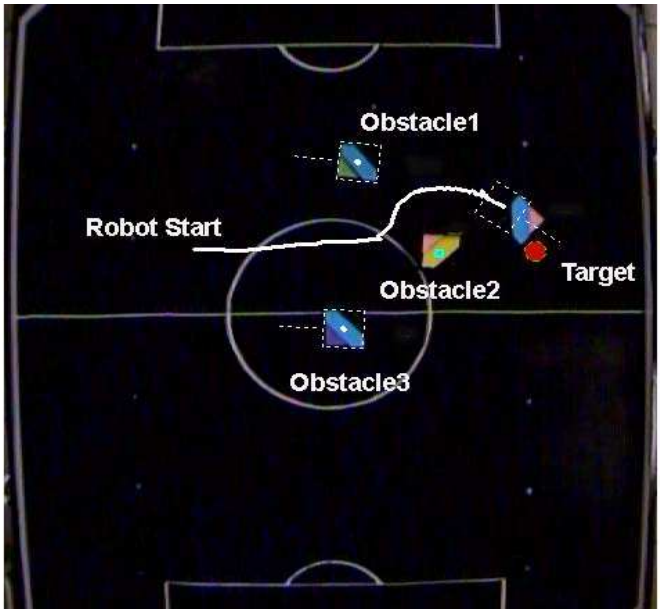
With more obstacles around (Figure 5.23), the robot also could reach the target along a short path. Robot1 started from upper left part of the field. Initially Robot1 encountered Obstacle2 and Obstacle4 which are close to each other, and it could pass through them. Robot1 then headed towards the target and stopped closer to Obstacle3. It changed its course of motion and navigated through the space between Obstacle3 and Obstacle1 effectively. The robot could navigate through the complicated obstacles space with the proposed approach.

The proposed EAPF is applied to two robots (Robot1 and Robot2). Figure 5.24 depicts two scenarios where Robot1 and Robot2 navigate through various static obstacles. Robot1 and Robot2, both with EAPF path planner, starting from different sides of the platform, are required to reach their respective targets. Robot1 moved along an upper path to avoid Robot3 and Robot6; and Robot2 went between Robot4 and Robot5 towards its target.

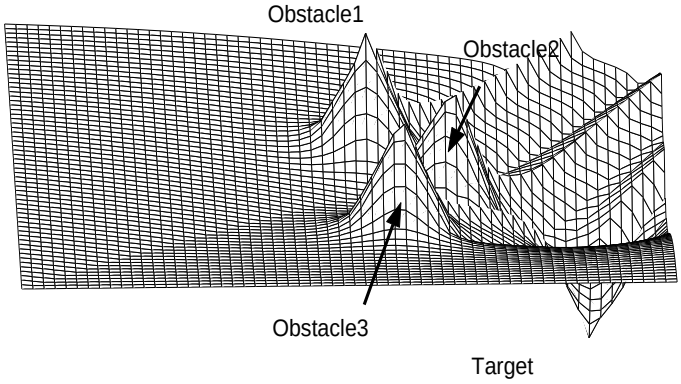
In Figure 5.25 the two robots avoided colliding each other. The changes in the paths of Robot1 and Robot2 around the region 'A' where they crossed the path validated the suitability of EAPF in dynamic path planning.

5.7 Comparison with AW-EPF

Both EAPF and AW-EPF (chapter 4) are based on potential field and comparison of their performances on typical situations could be helpful to identify their

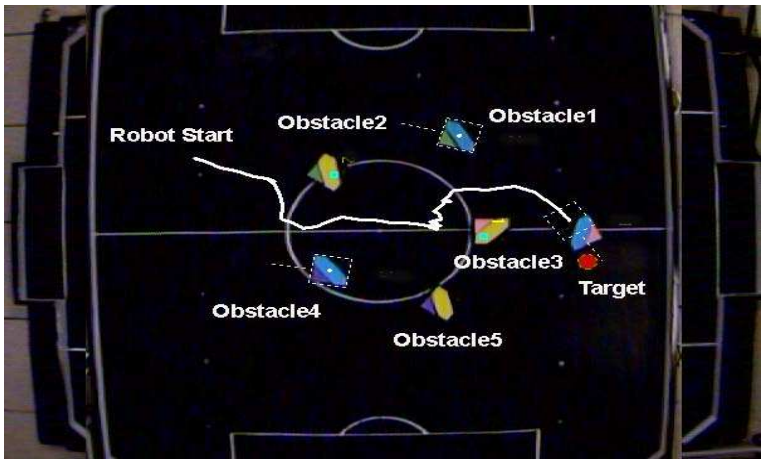


(a) Trajectory

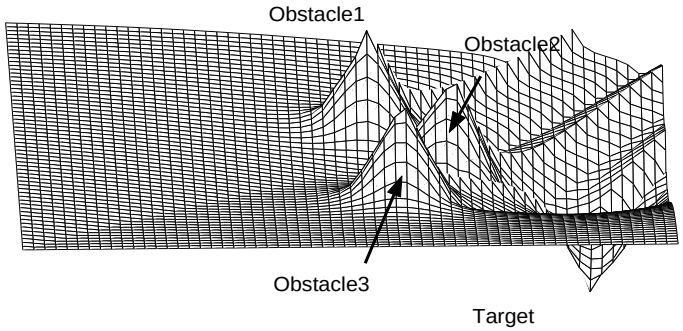


(b) Potential Field

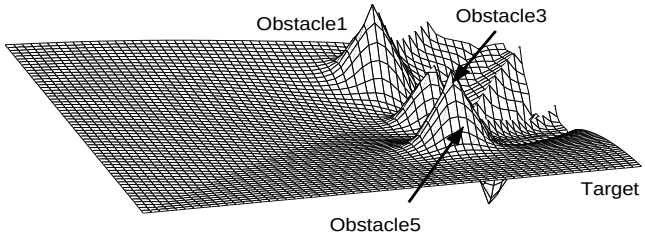
Figure 5.22: Robot avoiding stationary obstacles on the field



(a) Trajectory through a complicate environment



(b) Potential Field at start position



(c) Potential Field at intermediate position

Figure 5.23: Robot passing multiple obstacles on the field

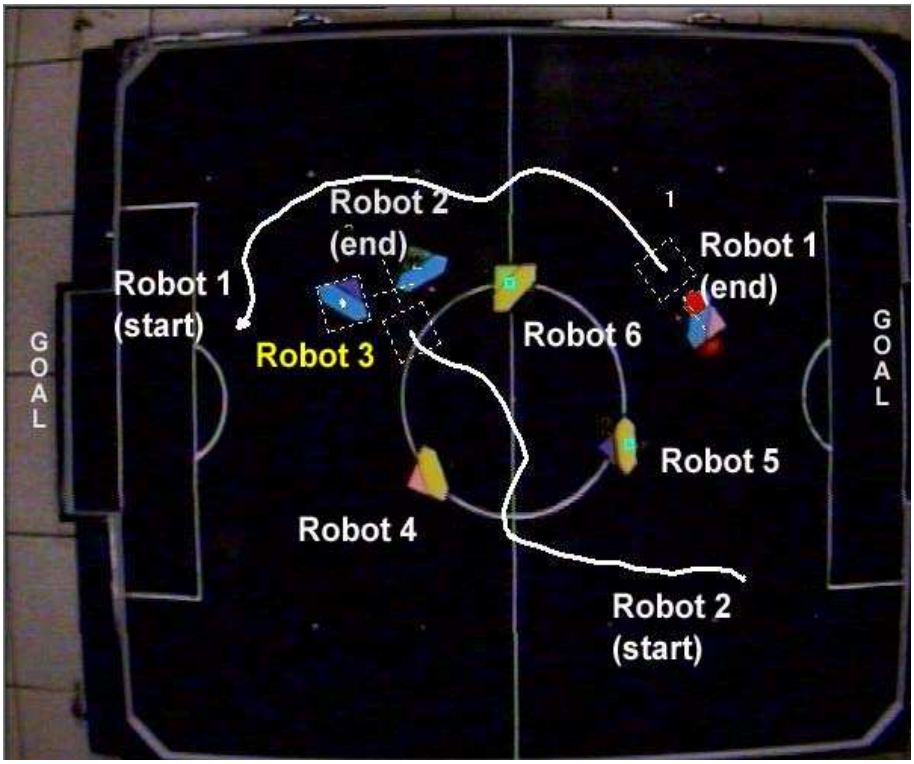


Figure 5.24: EAPF application1 on multiple robots

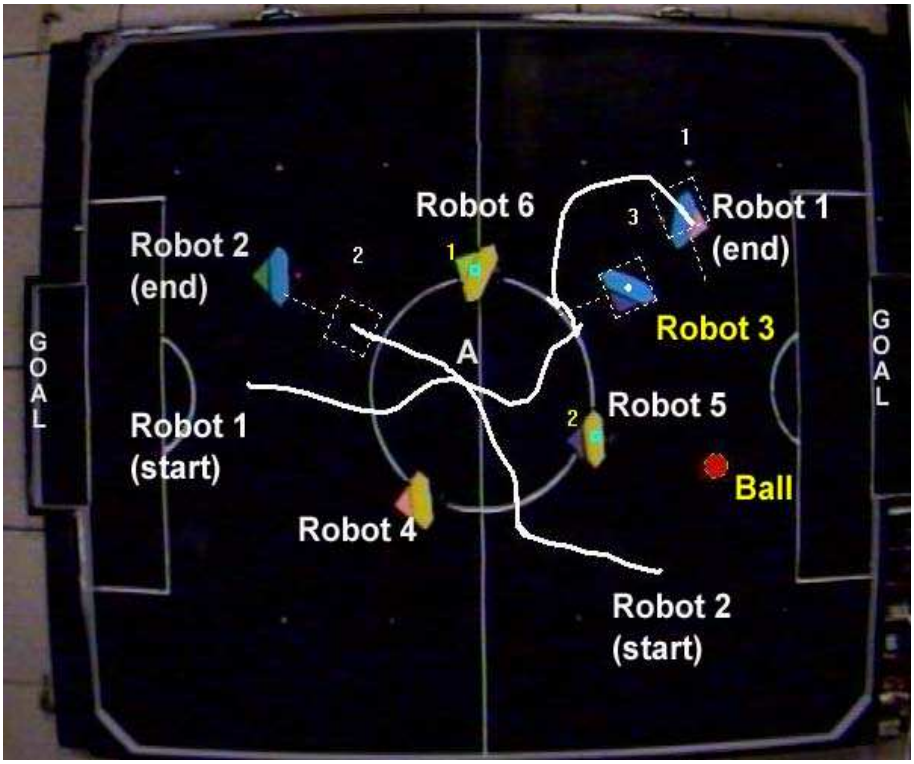


Figure 5.25: EAPF application2 on multiple robots

characteristics. With the optimized configuration in previous experiments, the simulation experiments are carried out and the results are presented in figures below. Application of adaptive window considerately diminishes the computation burden of solving matrix in AW-EPF, but there is no such a concern in EAPF which consumes much less time in computing and it is not combined with EAPF approach in the following experiments.

In Figure 5.26, the robot starts from a right side position and goes to the Target by EAPF and AW-EPF alternatively. The trajectory of dot line is the path navigated by EAPF and the solid one is by AW-EPF. Their running times are almost the same, 5.44 seconds and 5.4 seconds respectively. In EAPF approach, the distance between the obstacles is larger than it is by AW-EPF.

Their performances are also compared in Figure 5.27 with more moving obstacles, Obstacle1 and Obstacle3. The dot line is the path navigated by EAPF and the solid one is by AW-EPF. The robot by EAPF takes extra 0.12 second to reach the Target than by EPF.

It could be implicated from the results that the robot is affected by the obstacles aside more in EAPF based navigation and it keeps further from the obstacles than it does in AW-EPF based navigation. By AW-EPF approach the robot path is more smooth, while by EAPF the robot is more sensitive and reactive to the environment. One of the reasons is EAPF potential field distribution is sharper because of the exponential formula construction. Meanwhile, the usage of grid map with limited resolution smooths AW-EPF potential distribution. Hence the both approach could be chosen to apply according to the sensitivity requirement and computation capability.

5.8 Discussions

For different values of the EAPF parameters, the potential fields and corresponding robot motion are simulated and tested. The EAPF functions proposed are tested in

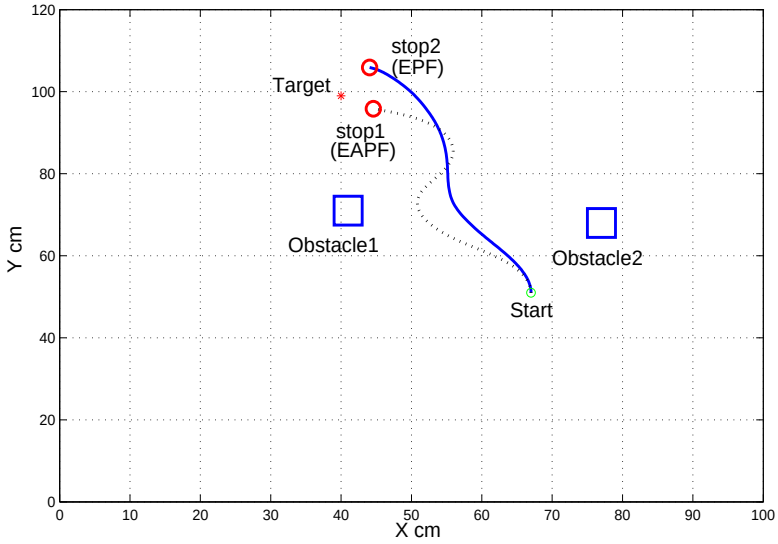


Figure 5.26: Trajectories by AW-EPF and EAPF(case 1)

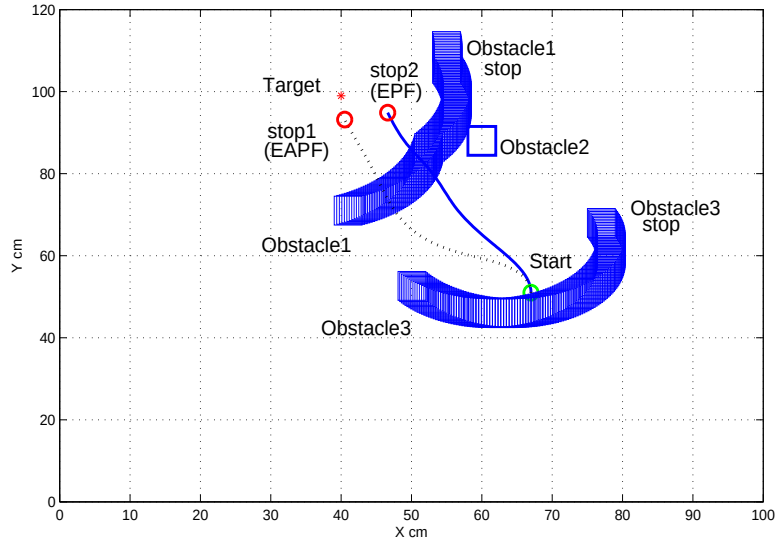


Figure 5.27: Trajectories by AW-EPF and EAPF(case 2)

different scenarios for ball tracking and kicking, while facing competition from other robots. The proposed approach could be applied for real time obstacle avoidance in dynamic environments effectively.

EAPF navigated the robot path effectively to avoid obstacles and reach target finally in various scenarios in the multiple mobile robot system.

For more accurate solutions, it is required to optimize the parameters associated with the EAPF functions in real-time. Further research is needed to improve the path planning performance in the system cooperative behaviors.

Chapter 6

Particle Filter based Trajectory Prediction

6.1 Introduction

In robotics, trajectory prediction is an important issue while planning the motion and different prediction approaches exist. In this chapter, a particle filter based trajectory prediction method for mobile robots is proposed and the method is applied in a multiple mobile robot system.

The problem of predicting moving object trajectory is often encountered in industrial robotic or servo systems where the motion information such as position, velocity, and acceleration, is required for control. Even with high-frequency updating information, a reliable prediction trajectory would be helpful to grasping tasks.

Using prediction procedure, the control system could anticipate the object's position, velocity and acceleration with in order to plan an appropriate path for grasping, catching objects or avoiding obstacles.

In robotic systems, Bayesian framework is widely used to estimate and predict

the system states. Kalman filter is one of the common methods used in sensor fusion based on the assumption that the system is linear and Guassian [143]. Kalman filter is of limited power when the system is nonlinear and non-Guassian in type. The extended Kalman filter approach, requires linearisation about the predicted state with Guassian posterior distribution. The approximation that the posterior distribution is Guassian in type is a gross distortion of the true underlying structure and may lead to filter divergence.

Particle filter, a sequential Monte Carlo method for on-line learning within a Bayesian framework, can be applied to any state-space models [51]. In particle filtering, the target distribution is represented by a set of samples, called particles, with associated importance weights which are propagated through time. Such filters have been variously described as Bayesian bootstrap [144], condensation [145], Monte Carlo [146] and Metropolis-Hasting importance resampling filters [147]. Owing to its advantage of handling nonlinear and non-Guassian system estimation, particle filter is utilized widely in tracking applications [49][148][38]. In [149] and [150], particle filter combined with Gibbs sampler is applied to track multiple moving objects. The targets are assumed to move at nearly constant velocity and only one-dimensional observation variable, the bearing angle, is considered.

Prediction method based on particle filter is to predict the state of the target of interest at a point in the future. By this approach, the position, velocity, and acceleration of the object are predicted by particle filter on the base of previous object information.

In this chapter, the particle filter is used to predict the robot's position and velocity and the experimental results are presented to illustrate the performances of the proposed approach. The experiments of target prediction in robot path planning are carried out in robot soccer system to improve the team competitive ability.

6.2 Generic Particle Filter

The tracking problem is discussed to illustrate the particle filter algorithm. Consider a state sequence $\{X_k, k \in N\}$ of a target is evolved by

$$X_k = f_k(X_{k-1}, V_{k-1}) \tag{6.1}$$

where $f_k : R^{n_x} \times R^{n_v} \longrightarrow R^{n_x}$ is a possibly nonlinear function of the state $X_{k-1}, \{V_{k-1}, k \in N\}$ is an independent and identically distributed process noise sequence, and n_x, n_v are dimensions of the state and process noise vectors respectively. N is the set of natural numbers. The objective of the tracking is recursively estimate X_k from measurements

$$Z_k = h_k(X_k, \mathbf{n_k}) \tag{6.2}$$

where $h_k : R^{n_z} \times R^{n_n} \longrightarrow R^{n_z}$ is a possibly nonlinear function, $\{\mathbf{n_k}, \mathbf{k} \in \mathbf{N}\}$ is an independent and identically distributed measurement noise sequence, and n_z, n_n are dimensions of the state and process noise vectors, respectively. The set of available measurements $Z_{1:k} = \{Z_i, i = 1, ..., k\}$ are the base of estimates of X_k.

In Bayesian approach, the trajectory estimation problem is to recursively calculate some degree of belief in the state X_k at time k, taking different values, given the measurement data $Z_{1:k}$ up to time k. Thus, it is required to construct the posterior density function (pdf) $p(X_k|Z_{1:k})$. It is assumed that the initial pdf $p(X_0|Z_0) \equiv p(X_0)$ of the state vector, which is also known as the prior, is available. Z_0 is the set of initial empty measurements. Then, the pdf $p(X_k|Z_{1:k})$ can be obtained recursively in two stages of prediction and update.

Assume that the required pdf $p(X_{k-1}|Z_{1:k-1})$ at time $k-1$ is available. The prediction stage involves using the system model (6.1) to obtain the prior pdf of the state at time k via the Chapman-Kolmogorov equation,

$$p(X_k|Z_{1:k}) = \int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dX_{k-1}. \tag{6.3}$$

Here it is assumed that $p(X_k|X_{k-1}, Z_{1:k-1}) = p(X_k|X_{k-1})$ since the system model describes a Markov process of order one. The probabilistic model of the

state evolution $p(X_k|X_{k-1})$ is defined by the system equation (6.1) and the known statistics of V_{k-1}.

At time step k, a measurement Z_k becomes available, and this is used to update the prior via Baye's rule,

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k)p(X_k|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})}, \tag{6.4}$$

where the normalizing constant,

$$p(Z_k|Z_{1:k-1}) = \int p(Z_k|X_k)p(X_k|Z_{1:k-1})dX_k, \tag{6.5}$$

depends on the likelihood function $p(Z_k|X_k)$ defined by the measurement model (6.2) and the known statistics of $\mathbf{n_k}$. In the update stage (6.4), the measurement Z_k is used to modify the prior density to obtain the required posterior density of the current state.

The recurrence relations of (6.3) and (6.4) form the basis for the optimal Bayesian solution, which solves the problem of recursively calculating the exact posterior density. This recursive propagation of the posterior density is a conceptual solution in general, and cannot be determined analytically. Optimal Bayesian solutions exist in a restrictive set of cases, including the Kalman filter and grid-based filters, meanwhile, suboptimal algorithms like extended Kalman filters, approximate grid-based filters, and particle filters, approximate the optimal Bayesian solution.

Particle filtering is based on sequential Monte Carlo (MC) method for implementing a recursive Bayesian filter by Monte Carlo simulations [151]. The main idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, the MC characterization becomes an equivalent representation to the usual functional description of the posterior pdf.

To describe the algorithm, let $\{X^i_{0:k}, w^i_k\}^{N_s}_{i=1}$ denote a random measure that characterizes the posterior pdf $p(X_{0:k}|Z_{1:k})$, where $\{X^i_{0:k}, i = 0, ..., N_s\}$ is a set of

support points with associated weights $\{w_k^i, i = 1, ..., N_s\}$ and $X_{0:k} = \{X_j, j = 0, ..., k\}$ is the set of all states up to time k. The weights are normalized such that $\sum_i w_k^i = 1$. Hence, the posterior density at k can be approximated as,

$$p(X_{0:k}|Z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(X_{0:k} - X_{0:k}^i). \tag{6.6}$$

The weights are chosen using the principle of importance sampling as following. Suppose $p(x) \propto \pi(x)$ is probability density for which $\pi(x)$ can be evaluated. Let $x^i \sim q(x), i = 1, ..., N_s$ be samples that are easily generated from a proposal $q(\cdot)$ called an importance density. So a weighted approximation to the density $p(\cdot)$ is given by,

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i), \tag{6.7}$$

where,

$$w^i \propto \frac{\pi(x^i)}{q(x^i)}, \tag{6.8}$$

is the normalized weight of the ith particle.

Hence, if the samples $X_{0:k}^i$ are drawn from an importance density $q(X_{0:k}|Z_{1:k})$, the weights in (6.6) as defined by (6.8) becomes,

$$w_k^i \propto \frac{p(X_{0:k}^i|Z_{1:k})}{q(X_{0:k}^i|Z_{1:k})}. \tag{6.9}$$

At each iteration, we have samples constituting an approximation to $p(X_{0:k-1}|Z_{1:k-1})$ and approximating $p(X_{0:k}|Z_{1:k})$ with a new set of samples.

Returning to the sequential case, at each iteration, we could have samples constituting an approximation to $p(X_{0:k-1}|Z_{1:k-1})$ and want to approximate $p(X_{0:k}|Z_{1:k})$ with a new set of samples. If the importance density is chosen as,

$$q(X_{0:k}|Z_{1:k}) = q(X_k|X_{0:k-1}, Z_{1:k})q(X_{0:k-1}|Z_{1:k-1}) \tag{6.10}$$

then we can obtain samples $X_{0:k}^i \sim q(X_{0:k}|Z_{1:k-1})$ by augmenting each of the existing samples by $X_{0:k-1}^i \sim q(X_{0:k-1}|Z_{1:k-1})$ with the new state $X_k^i \sim q(X_k|X_{0:k-1}, Z_{1:k})$.

To derive the weight update equation, $p(X_{0:k}|Z_{1:k})$ is first expressed in terms of $p(X_{0:k-1}|Z_{1:k-1})$, $p(Z_k|X_k)$ and $p(X_k|X_{k-1})$. Note that

$$\begin{aligned} p(X_{0:k}|Z_{1:k}) &= \frac{p(Z_k|X_{0:k}|Z_{1:k-1})p(X_{0:k}|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} && (6.11)\\ &= \frac{p(Z_k|X_{0:k}|Z_{1:k-1})p(X_k|X_{0:k-1}|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} \times p(X_{0:k-1}|Z_{1:k-1}) && (6.12)\\ &= \frac{p(Z_k|X_k)p(X_k|X_{k-1})}{p(Z_k|Z_{1:k-1})}p(X_{0:k-1}|Z_{1:k-1}) && (6.13)\\ &\propto p(Z_k|X_k)p(X_k|X_{k-1})p(X_{0:k-1}|Z_{1:k-1}) && (6.14) \end{aligned}$$

Substituting (6.11) into (6.9), the weight update equation can then be:

$$w_k^i \propto \frac{p(Z_k|X_k^i)p(X_k^i|X_{k-1}^i)p(X_{0:k-1}^i|Z_{1:k-1})}{q(X_k^i|X_{0:k-1}^i, Z_{1:k})q(X_{0:k-1}^i|Z_{1:k-1})} = w_{k-1}^i \frac{p(Z_k|X_k^i)p(X_k^i|X_{k-1}^i)}{q(X_k^i|X_{0:k-1}^i, Z_{1:k})}. \quad (6.15)$$

Furthermore, if $q(X_k|X_{0:k-1}, Z_{1:k}) = q(X_k|X_{k-1}, Z_k)$, the importance density becomes only dependent on X_{k-1} and Z_k. This is particularly useful in a common situation when only a filter estimate of $p(X_k|Z_{1:K})$ is required at each time step. Assume such a case except when explicitly stated otherwise. Hence only X_k^i need be stored, we can discard the path $X_{0:k-1}^i$ and history of observations $Z_{1:k-1}$. The weights is modified as,

$$w_k^i \propto w_{k-1}^i \frac{p(Z_k|X_k^i)p(X_k^i|X_{k-1}^i)}{q(X_k^i|X_{k-1}^i, Z_k)}, \quad (6.16)$$

and the posterior filtered density $p(X_k|Z_k)$ can be approximated as,

$$p(X_k|Z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(X_k - X_k^i). \quad (6.17)$$

The weights are defined in (6.16). As $N_s \to \infty$, the approximation (6.17) approaches the true posterior density $p(X_k|Z_{1:k})$.

A common problem with the particle filter is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. It is because that the variance of the importance weights only increase over time so it is impossible to avoid the degeneracy phenomenon. Degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $p(X_k|Z_{1:k})$ is almost zero. A measure of degeneracy of the algorithm

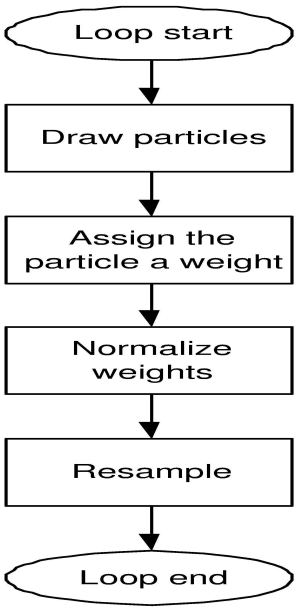


Figure 6.1: Generic particle filter procedure illustration

is the effective sample size N_{eff} [152], defined as,

$$N_{eff} = \frac{N_s}{1 + Var(w_k^{*i})}, \tag{6.18}$$

where $w_k^{*i} = \frac{p(X_k^i|Z_{1:k})}{q(X_k^i|X_{k-1}^i, Z_k)}$ is referred to as the true weight. The estimate $\widehat{N_{eff}}$ of N_{eff} can be obtained by,

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s}((w)_k^i)^2}. \tag{6.19}$$

To overcome the degeneracy problem, we need good choice of importance density or usage of resampling method.

The most common choice of importance density is,

$$q(X_k|X_{k-1}^i, Z_k) = p(X_k|X_{k-1}^i). \tag{6.20}$$

Substitute (6.20) into (6.16) yields,

$$w_k^i \propto w_{k-1}^i p(Z_k|X_k^i). \tag{6.21}$$

Among a cluster of other densities that can be used, the optimal importance density function that minimizes the variance of the true weights w_k^{*i} conditioned on X_{k-1}^i

and Z_k as shown in [153] is

$$q(X_k|X_{k-1}^i, Z_k)_{opt} = p(X_k|X_{k-1}^i, Z_k) \quad (6.22)$$

$$= \frac{p(Z_k|X_k|X_{k-1}^i p(X_k|X_{k-1}^i)}{p(Z_k|X_{k-1}^i)}. \quad (6.23)$$

Substituting into (6.16) yields,

$$w_k^i \propto w_{k-1}^i p(Z_k|X_k^i) \quad (6.24)$$

$$= w_{k-1}^i \int p(Z_k|X_k')p(X_k'|X_{k-1}^i)dX_k'. \quad (6.25)$$

This choice of importance density is optimal since for a given X_{k-1}^i, w_k^i takes the same value, whatever sample is drawn from $q(X_k|X_{k-1}^i, Z_k)_{opt}$.

Resampling method also reduces the effects of degeneracy. The base of resampling is to eliminate particles that have small weights and to concentrate on particles with large weights. The resampling involves generating a new set $\{X_k^{i*}\}_{i=1}^{N_s}$ by resampling (with replacement) N_s times from an approximate discrete representation of $p(X_k|Z_{1:k})$ given by,

$$p(X_k|Z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(X_k - X_k^i), \quad (6.26)$$

so that $P_r(X_k^{i*} = X_k^j) = w_k^j$. The resultant sample is an independent and identically distributed sample from the discrete density (6.26), hence the weights are now reset to $w_k^i = 1/N_s$.

In this project, a randomly moving object's position and velocity are predicted by particle filter algorithm. As the robot moves fast, the velocity and acceleration vary quickly and are hard to predict. To avoid sample impoverishment which makes all particles collapse to a single point, a random resampling algorithm [38] is applied to tackle this issue. In this work, the new particles are sampled from the neighborhoods of the focused particles based on the uniform distribution, $U(x_k^{i:l} - \alpha|x_k^{i:l}|, x_k^{i:l} + \alpha|x_k^{i:l}|)$. $x_k^{i:l}$ is the lth state variable of the state vector x_k^i. α determines the size of sampling region, which adapts to the effective number of particles N_{eff} and the variance $\sigma^2_{x_k^{i:l}}$ of the state variable.

$$\alpha \propto \frac{\sigma^2_{x_k^{i:l}}}{N_{eff}} \quad (6.27)$$

When N_{eff} decreases, the sampling region is expanded to cover diverse particles. Variance σ^2 determines the tracking precision of the state variable, larger σ^2 means a wider searching area and sampling region is expanded at the same time, and vice verse. In the improved resampling algorithm, two thresholds about effective sample size are set, one for degeneracy and the other for sample improvement. With the degeneracy threshold, the particles with small weights are eliminated and those with large weights ar retained and duplicated. Then the diversity of particles decreases rapidly and their effective size is less than the second threshold, new particles are sampled from the neighborhoods of previous samples based on uniform distribution. The size of the sampling space is determined by α, which adapts to the effective number of particles and variance of the state variable. So the distribution space of new samples is expanded while the continuity of the state space remains.

In practical situations, the predicted position would be limited by the physical constraints, a window is applied to keep the anticipated position within the reasonable area.

6.3 Trajectory Prediction

The robot soccer system is utilized to implement the particle filter algorithm in predicting the position and velocity of a mobile robot. The prediction problem can be formulated as follows: Given an object moving along some arbitrary path, it is required to predict its trajectory in real time and with endurable error. For the 2-D space of robot soccer system, the predicted position (x, y) is calculated based on past values of the object coordinates obtained through vision processing by camera. To limit the complexity of the vision process, the objects are represented by their mass points. It is assumed that the object velocity and acceleration are within a specified range due to the physical constraints. The predictor outputs are the anticipated values of the target position and velocity at the next sampling period, which are used by the path planning to calculate control input.

There are four stages associated with the generic particle filter method used for trajectory prediction:

1. Initialization: Sample X_0^i from the initial posterior distribution $p(X_0)$ and set the weights w_0^i to $\frac{1}{N}$, $i = 1, ..., N$. The target initial position and velocity are assigned at the start of the particle filter module.

2. Prediction: Each sample is passed through the system model to obtain the predicted samples:

$$\widehat{X}_k^i = f(X_{k-1}^i, w_{k-1}^i) \tag{6.28}$$

   where w_{k-1}^i is a sample drawn from the probability density function of the system noise $p_\omega(\omega)$. The predictions of the target are calculated.

3. Update: Once the position data Z_k is updated, evaluate the importance weight of each predicted sample in (6.29) and normalize the weights in (6.30).

$$w_k^i = \widetilde{w}_{k-1}^i \frac{p(Z_k|\widehat{X}_k^i)p(\widehat{X}_k|X_{k-1}^i)}{q(\widehat{X}_k|X_{k-1}^i, Z_k)} \tag{6.29}$$

$$\widetilde{w}_k^i = \frac{w_{k-1}^i}{\sum_{i=1}^N w_{k-1}^i} \tag{6.30}$$

   Define a discrete distribution $\{\widetilde{w}_k^i : i = 1, ..., N\}$ over $\{\widetilde{X}_k^i : i = 1, ..., N\}$, with probability mass $\widetilde{w}_k^i$ associated with element $\widetilde{X}_k^i$ at time k.

4. Resample: Calculate the effective sample size:

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\widehat{w}_k^i)^2} \tag{6.31}$$

An iteration of the recursive algorithm consists of the above four steps. In this work, the prediction function is implemented in an object module of robot soccer system control software and the game management enables the prediction according to situations. The robot behaviors include tracking and blocking the moving object to defeat an opponent team.

6.4 Experimental Results

The prediction module based on particle filter framework is tested in Robot Soccer System (Chapter 2). The system perceives the environment using an overhead camera at a refresh frequency of $40ms$. The Improved particle filter is implemented as an object in the software, with a Gaussion pdf, a sample number of 200, and an adaptive noise error of 0.01-0.05m.

Firstly, the prediction performance is tested for a random motion of the ball. In Figure 6.2, the ball's initial position is at (0.75, 0.65)m of the RSS coordination system. The predicted trajectory error is below 5cm, the max errors occurs around the sharp turning where the velocity is too large.

The prediction of motion could apply on either target position prediction or obstacle position prediction. The prediction is incorporated with the robot behaviors when it locks a target ball or opponent robot.

The experiments show the influence of the algorithm by comparing the performance of with and without the prediction. In Figure 6.4, from the same initial situation and the same target situation, the moving obstacle Obstacle1 moves from the the left lower side of the ground (0.15, 0.4)m to the upper right area, while Robot1 starts from (0.3, 0.2)m to get the ball on the upper part without collision with the Obstacle1. In Figure 6.4 (a), with the prediction algorithm, Robot1 could avoid the moving obstacle and took 2.5 seconds to reach the target ball. While in (b), Robot1 took longer time, 3 seconds to reach the ball. With the prediction of obstacle position, the robot reached the target earlier, which is an improvement in the competition ability during the match.

The robot behaviors associated with the robot soccer system are shown in Figure 6.3. The prediction function is ready to be called by the further behavior functions.

In Figure 6.6 Robot1 and Robot2 is assigned to pursue and surpass OppRobot1 and OppRobot2 respectively to protect goal area, with collision free function by the

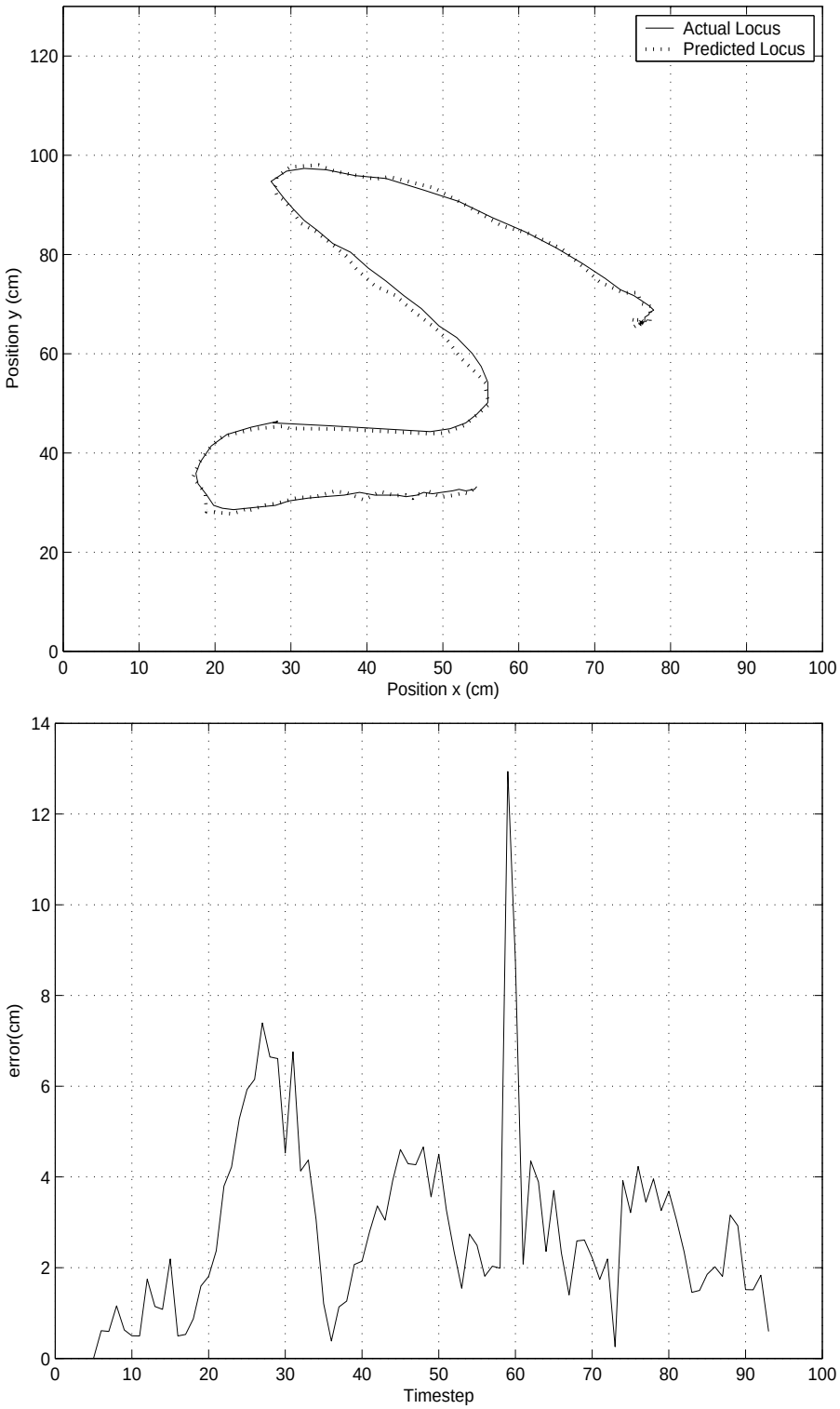


Figure 6.2: Random moving object trajectory prediction

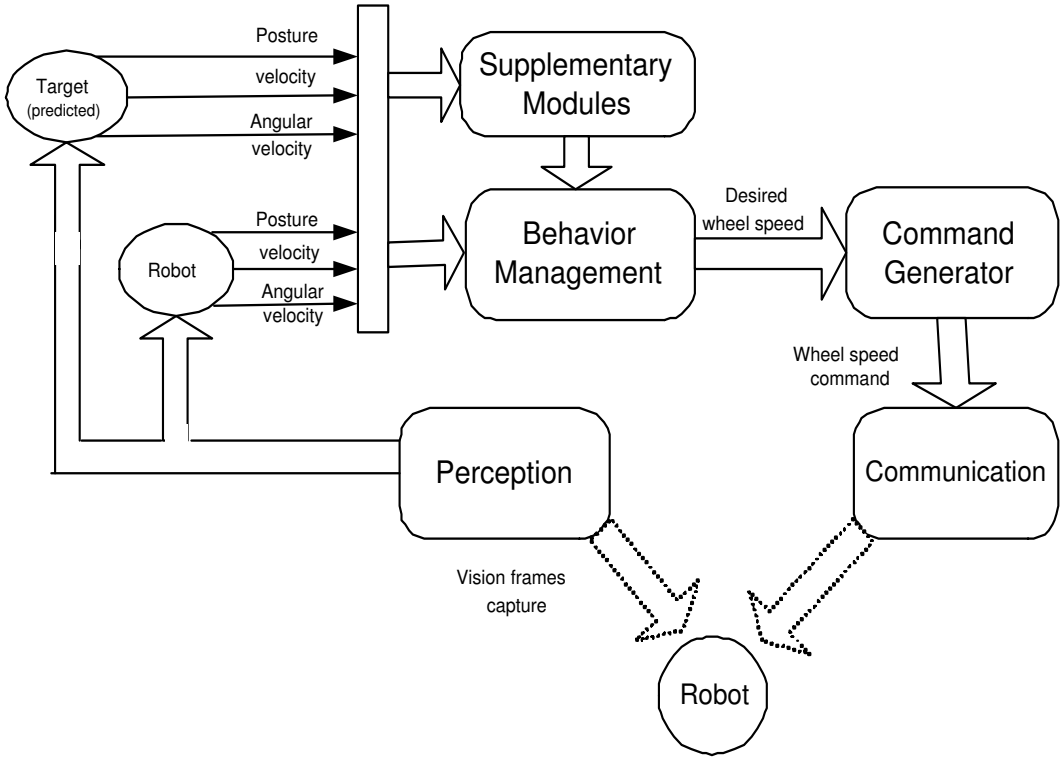
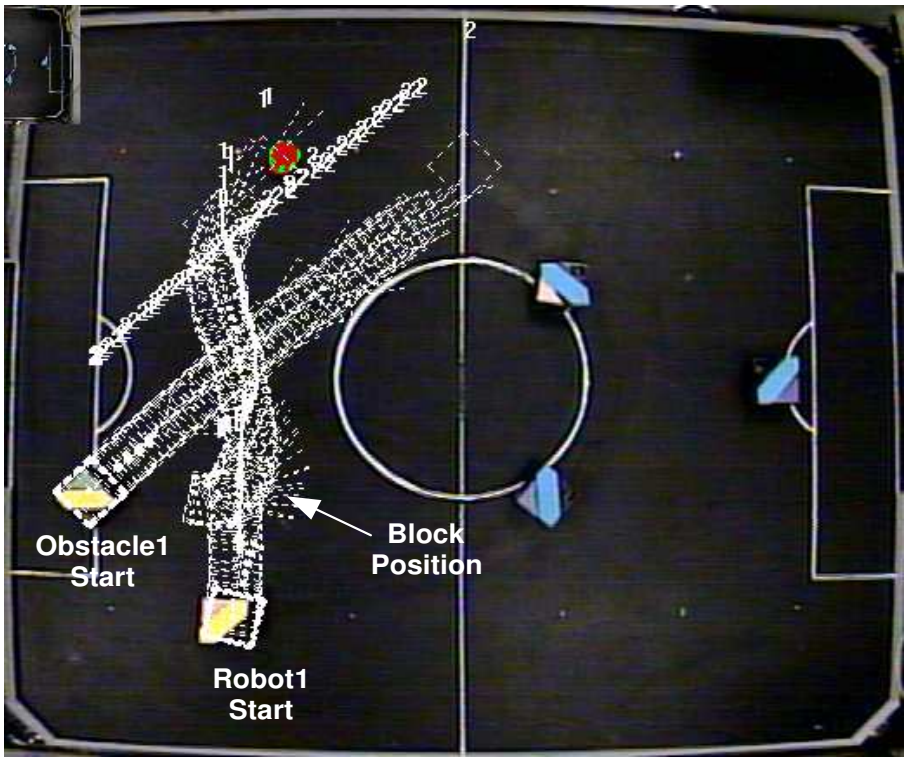


Figure 6.3: Behavior decision procedures

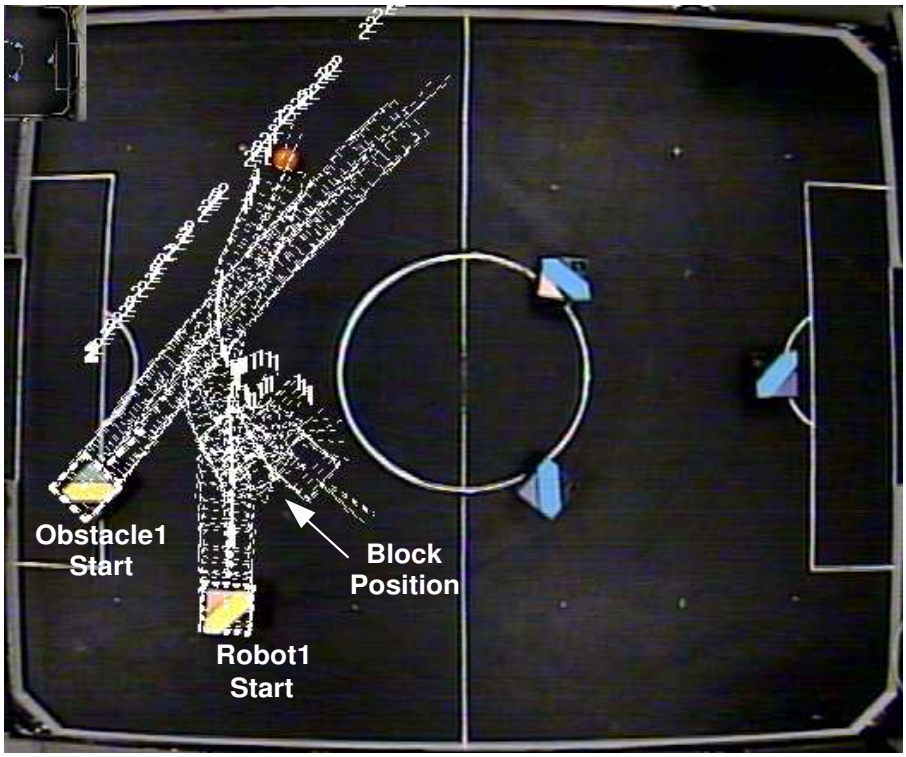
EAPF algorithm to avoid team robots. With the prediction as in Figure 6.6 (a), Robot1 pushed OppRobot1 away from the goal area, Robot2 successfully homed OppRobot2 in time and run to support the goalie, while as results in Figure 6.6 (b), the defending robots Robot1 and Robot2 run into jam with the opponents (ellipse area) and fail to assistant the goalie.

6.5 Discussions

In this chapter, a prediction method based on particle filter is constructed and verified successfully in a mobile robot system. For a competitive environment, the robots become more capable with the prediction algorithm. This would give the robot sufficient time and space to perform the necessary actions to avoid obstacles and change its route towards its destination position.



(a) robot motion with prediction



(b) robot motion without prediction

Figure 6.4: Robot motion comparison

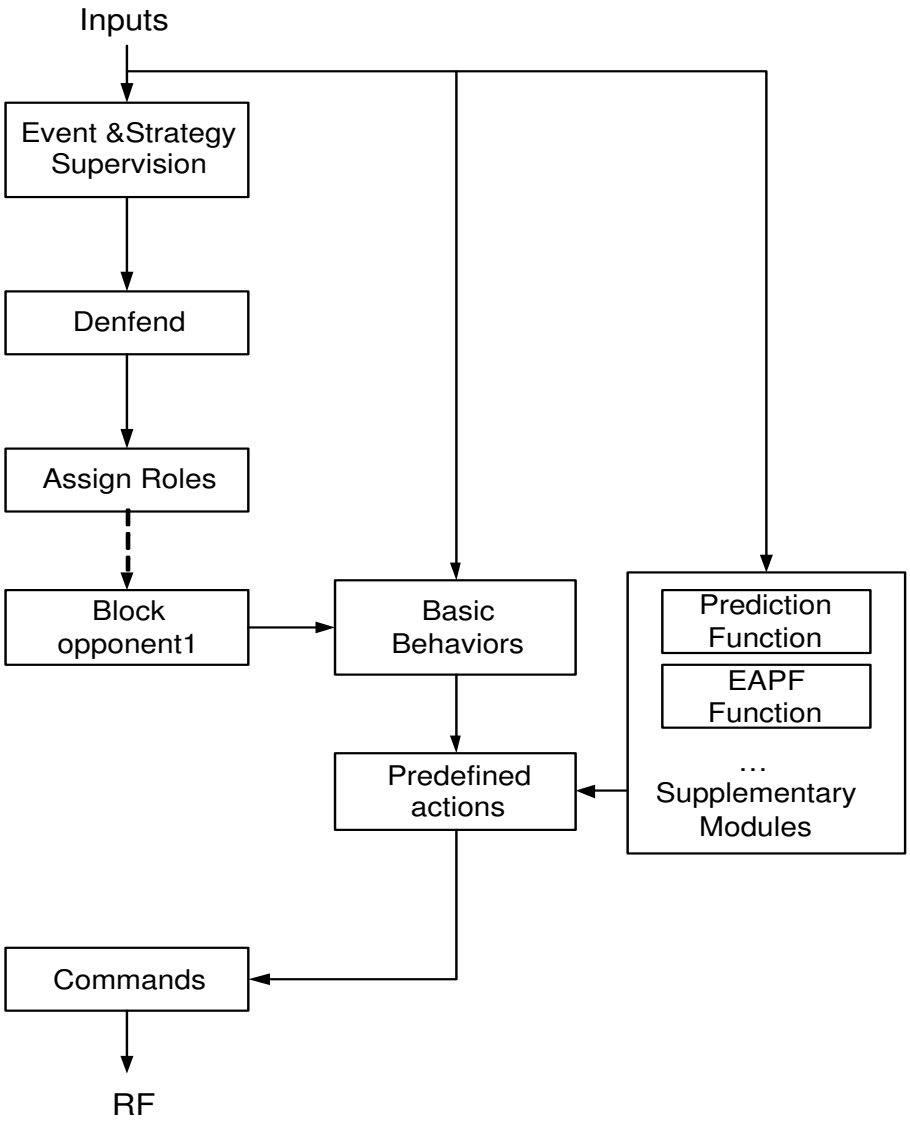
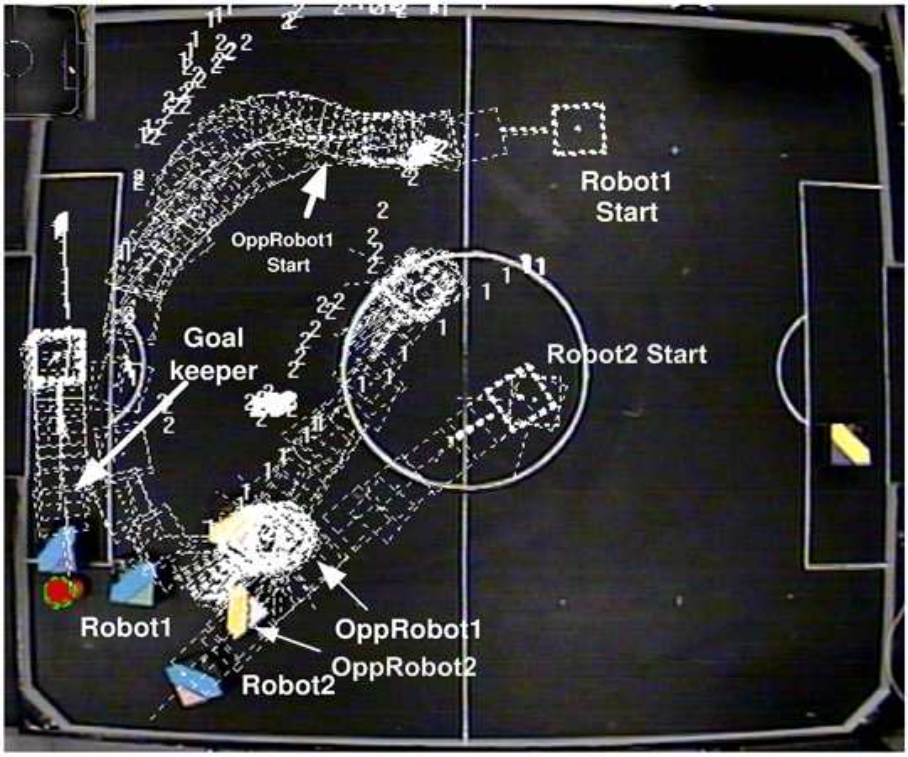
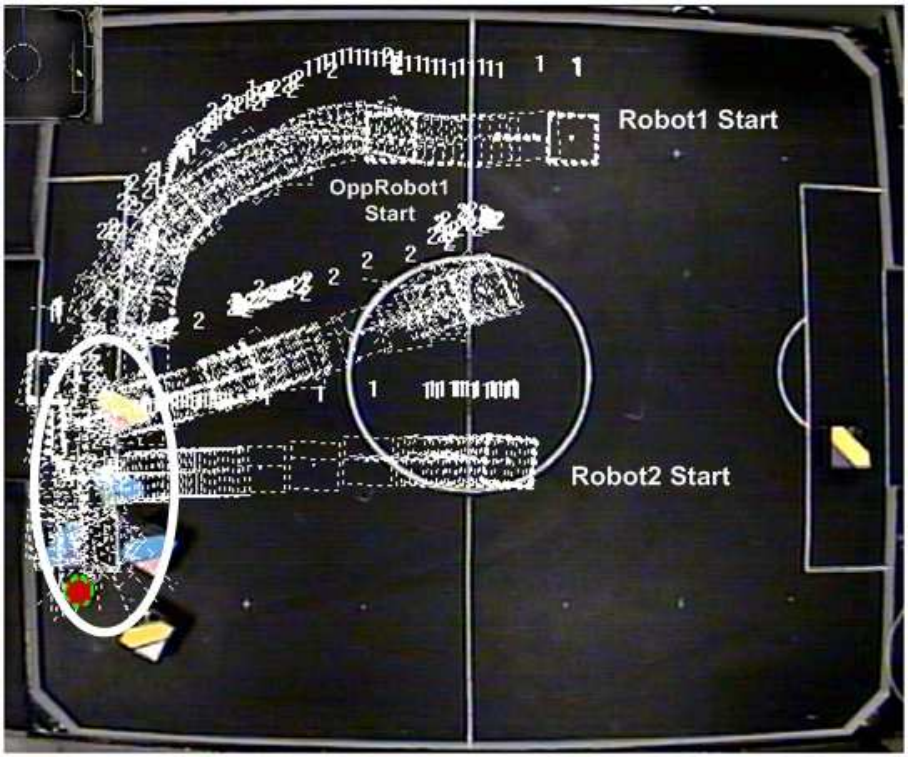


Figure 6.5: System processing with prediction



(a) robot motion with prediction



(b) robot motion without prediction

Figure 6.6: Robot motion comparison

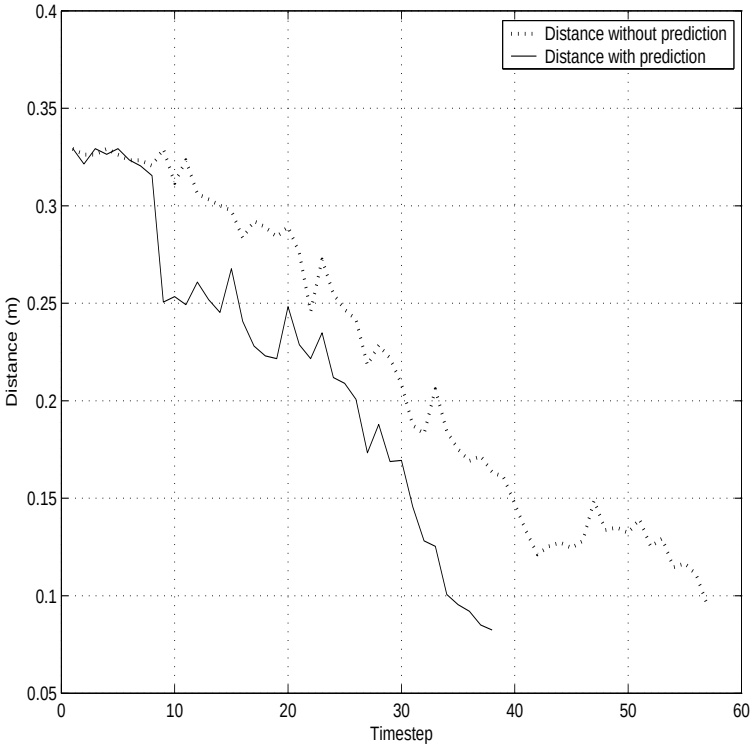


Figure 6.7: Distance between robot and target with & without prediction

Chapter 7

Conclusions

In the thesis, the multiple mobile robot system and dynamic path planning methods are studied with experiments on wheeled mobile robot.

Firstly, a multiple mobile robot system, Robot Soccer System is studied along hardware and game management architectures (Chapter 2). A top-down behavior management system is set up in C++. The individual robot model is presented in Chapter 3. A tracking controller based on the models is validated through simulation and experimentation.

An Electrostatic Potential Field based path planning method is presented in Chapter 4, it is observed that AW-EPF with adaptive window can generate shorter and quicker paths for mobile robots.

In Chapter 5, an Evolutionary Artificial Potential Field approach is proposed. For different values of the EAPF parameters, the potential fields and corresponding robot motion are simulated and tested. In the experiments, EAPF path planner navigated the robots effectively to avoid obstacles and reach the target. The EAPF functions proposed are tested in different scenarios for ball tracking and kicking. The proposed approach could be applied for real time obstacle avoidance in dynamic environments.

The two navigation approaches, based on EPF and EAPF respectively, are

inspired by the physical potential field distribution and have different features. EAPF approach is more flexible due to its construction flexibility and more agile in reacting to the changes in environment, while EPF is more suitable for global path planning. The drawback of EPF is the exponential increase in computing time with the increase in resolution of the grids.

In Chapter 6, a trajectory prediction method based on particle filter is presented. The results show that the prediction can improvement the performance of the robots in dynamic environments.

In the proposed approaches and experiments, each robot's path planning is based on data of the whole multiple mobile systems. The data of each robot are collected at each iteration and processed to navigate the single robot through stationary or moving robots in the system in real time, and the other team robots are labeled at the start of the robot's itself path planning. Therefore the whole process is based on multiple robot systems in real time.

Further research would be focusing on the combination of various path planning methods and utilizing more computational intelligence techniques.

Bibliography

[1] Wito Jack, editor. *Intelligence Robotics System.* Kluwer Academic/Plenum Publishers, N.Y, USA, 1999.

[2] Andrew Garoogian. *Robotics, 1960-1983 : an annotated bibliography.* Brooklyn, N.Y. : CompuBibs, 1984.

[3] Kamal Gupta, editor. *Practical Motion Planning in Robotics.* John Wiley and Sons, 3 edition, January 1998.

[4] Robin R. Murphy. *Introduction to AI robotics.* MIT Press, Cambridge, Mass., 2000.

[5] Gibilisco Stan, editor. *The McGraw-Hill Illustrated Encyclopedia of Robotics and Artificial Intelligence.* McGraw-Hill Inc., 1994.

[6] Darren M. Dawson; Michael M. Bridges; Zhihua Qu. *Nonlinear control of Robotic Systems for Environmental waste and Restoration*, volume 1. Prentic Hall PTR, Englewood Cliffs, NJ 07632, 1995.

[7] Ronald C. Arkin. *Behavior-based Robotics.* The MIT Press, Cambridge, MA, 1998.

[8] S.X. Hao Li; Yang. A behavior-based mobile robot with a visual landmark-recognition system. *Mechatronics, IEEE/ASME Transactions on*, 8(3):390–400, Sept 2003.

[9] Luo R.C.; Tse Min Chen. Development of a multi-behavior based mobile robot for remote supervisory control through the internet. *Mechatronics, IEEE/ASME Transactions on*, 5(4):376–385, Dec 2000.

[10] Byeong-Soon Ryu; Hyun Seung Yang. Integration of reactive behaviors and enhanced topological map for robust mobile robot navigation. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 29(5):474–485, Sept 1999.

[11] P. Vadakkepat; Ooi Chia Miin; Xiao Peng; Tong Heng Lee. Fuzzy behavior-based control of mobile robots. *Fuzzy Systems, IEEE Transactions on*, 12(4):559–565, Aug 2004.

[12] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers Group, Norwell, Massachusetts 02061 USA, 1991.

[13] D. Miller. A spatial representation system for mobile robots. *IEEE International Conference on Robotics and Automation*, pages 122–127, 1985.

[14] S.X. Willms, A.R.; Yang. An efficient dynamic system for real-time robot-path planning. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 36(4):755 – 766, August 2006.

[15] D. Leven; M. Sharir. An efficient and simple motion planning algorithms for a ladder moving in two-dimensional space amidst polygonal barriers. *Proc. 1st ACM Symp. Computational Geometry*, 50:1208–1213, 1997. Nice,France.

[16] O.Khatib. Commands dynamique dans i'espace erational des robots manipulateurs en esence d'obstacles. *Ph.D. dissertation, Ecole nationale Supeeieure de I'Aeeronatique et de I'Espace, France*, 1980.

[17] O.Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, Spring 1986.

[18] Caselli S. ; Reggiani M. ; Rocchi R. Heuristic methods for randomized path planning in potential fields. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA2001*, pages 426 –431, 2001.

[19] Park M.G.; Lee M.C. Experimental evaluation of robot path planning by artificial potential field approach with simulated annealing. *Proceedings of the 41st SICE Annual Conference*, 4, August 2002.

[20] P. Vadakkepat; K.C. Tan; M.L.Wang. Evolutionary artificial potential fields and their application in real time robot path planning. *Congress on Evolutionary Computation, Proceedings of the*, 1.1(256-263), 2000.

[21] S.S.Ge; Y.J. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5):615–620, 2000.

[22] A.R.; Pereira G.A.S.; Mesquita R.C.; Silva E.J.; Caminhas W.M.; Campos M.F.M. Pimenta, L.C.A.; Fonseca. Robot navigation based on electrostatic field computation. *Magnetics, IEEE Transactions on*, 42(4):1459 – 1462, April 2006.

[23] G.Dozier; A.Homaifar; S.Bryson; L.Moore. Artificial potential field based robot navigation, dynamic constrained optimization and simple genetic hill-climbing. *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence.*, pages 189 –194, 1998.

[24] Hassoun M.; Demazeau Y.; Laugier C. Motion control for a car-like robot: potential field and multiagent approaches. *Proc. of the Int. Workshop on Intelligent Robots and Systems. IEEE. Raleigh, NC (USA) . July*, 1992.

[25] S.K. Pathak, K.; Agrawal. An integrated path-planning and control approach for nonholonomic unicycles using switched local potentials. *Robotics, IEEE Transactions on*, 21(6):1201–1208, Dec. 2005.

[26] Gong Cheng; Jason Gu; Tao Bai; Osama Majdalawieh. A new efficient control algrithm using potential field: Extension to robot path tracking. *CCECE 2004- CCGEI 2004, Niagara Falls, May/mai 2004*, 2004.

[27] Sung Jin Yoo; Yoon Ho Choi; Jin Bae Park. Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(6):1381–1394, June 2006.

[28] Dongbing Gu; Huosheng Hu. Receding horizon tracking control of wheeled mobile robots. *Control Systems Technology, IEEE Transactions on*, 14(4):743 – 749, July 2006.

[29] Das T.; Kar I.N. Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. *Control Systems Technology, IEEE Transactions on*, 14(3):501–510, May 2006.

[30] Lin S.; Huang C.L.; Chuang M.K. Hierarchical fuzzy control for autonomous navigatio of heeled robots. *Control Theory and Applications, IEE Proceedings-*, 152(5):598 – 606, 9 Sept 2005.

[31] R.Fierro; F.L.Lewis. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, 9(4):589–600, July 1998.

[32] Jong-Min Yang; Jong-Hwan Kim. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *Robotics and Automation, IEEE Transactions on*, 15(3):578 – 587, June 1999.

[33] Dixon W.E.; de Queiroz M.S.; Dawson D.M.; Flynn T.J. Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity. *Control Systems Technology, IEEE Transactions on*, 12(1):138 – 147, 2004.

[34] F.L.Lewis; C.T.Abdallah; D.M.Dawson. *Control of Robot Manipulators*. MacMillan, New York, March 1993.

[35] Ilya Kolmanovaky; N. Harris McClamroch. Developments in nonholonomic control problems. *IEEE Control Systems*, pages 20–36, 1995.

[36] Pathak K.; Franch J.; Agrawal S.K. Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *Robotics, IEEE Transactions on*, 21(3):505–513, June 2005.

[37] Dongkyoung Chwa. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *Control Systems Technology, IEEE Transactions on*, 12(4):637–644, July 2004.

[38] Liu Jing; Prahlad Vadakkepat. Improved particle filter in sensor fusion for tracking random moving object. *IMTC2004 Instrumentation and Measurement Technology Conference*, May 2004.

[39] Ackerman C.; Itti L. Robot steering with spectral image information. *Robotics,IEEE Transactions on*, 21(2):247–251, April 2005.

[40] Herman Bruyninckx; Dominiek Reynaerts. Path planning for mobile and hyper-redundant robots using pythagorean hodograph curves. *International Conference on Advanced Robotics (ICAR'97)*, 50:595–600, July 1997. Monterey, CA.

[41] Hao Ying. *Fuzzy Control and Modeling: Analytical Foundations and Applications.* New York, 2000.

[42] George J. Klir; Tina A. Folger. *Fuzzy sets, Uncertainty, and Information.* Prentice Hall, 1988.

[43] Stefano Nolfi. Evolutionary robotics: Exploiting the full power of self-organization. *Self-Learning Robots II: Bio-robotics (Digest No. 1998/248)*, pages 3/1 –3/7, 1998.

[44] Meyer J.M.; Phil Husbands; Inman Harvey. Evolutionary robotics: A survey of applications and problems. *EVOROBOT'98, FIRST EUROPEAN WORKSHOP ON EVOLUTIONARY ROBOTICS*, pages 1–21, 1998.

[45] K. S. Narendra; K. Parthasarathy. Identification and control of dynamical systems using neural network. *Neural Networks, IEEE Transactions on*, 1(1):4–27, Jan. 1990.

[46] Fogel D.B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1995.

[47] Abdollah Homaifar; Daryl Battle; Edward Tunstel. Soft computing-based design and control for mobile robot path tracking. *Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pages 35–40, 1999.

[48] Y.Kanayama; Y.Kimura; F.Miyazaki; and T. Noguchi. A stable tracking control method for an autonomous mobile robot. *in Proc.IEEE Int. Conf. Robot.Automat.*, 1990.

[49] G. Loy; L. Fletcher; N. Apostoloff; A. Zelinsky. An adaptive fusion architecture for target tracking. *Proc. 5th IEEE Int. Conf. Autom. Face and Gesture Recog.*, pages 248–253, May 2002.

[50] Lee T.H.; Lam H.K.; Leung F.H.F.; Tam P.K.S.;. A practical fuzzy logic controller for the path tracking of wheeled mobile robots. *IEEE Control Systems Magazine*, pages 60–65, April 2003.

[51] M. Arulampalam; S. Maskell; N. Gordon; T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Process.*, 50(2):174C188, Feb 2002.

[52] Horling Bryan; Lesser Victor; Vincent Regis; Wagner Thomas. The soft real-time agent control architecture. *Autonomous Agents and Multi-Agent Systems*, 12(1), 2006.

[53] M. Sanjeev Arulampalam; Simon Maskell; Neil Gordon; Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processinig*, 50(2):174–188, Feb 2002.

[54] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/ai repository/ai/html/faqs/ai/genetic/part2/faq.html.

[55] Yuval Davidor. *Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization*, volume 1. World Scientific, Singapore, 1991.

[56] P. Lucidarme. An evolutionary algorithm for multi-robot unsupervised learning. *Evolutionary Computation, 2004. CEC2004. Congress on*, 2:2210 – 2215, 19-23 June 2004.

[57] Zu D.; Han J.D.; Campbell M. Artificial potential guided evolutionary path plan for multi-vehicle multi-target pursuit. *Robotics and Biomimetics, IEEE International Conference on*, pages 855 – 861, 2004. 22-26 Aug.

[58] Pishkenari H.N.; Mahboobi S.H.; Meghdari A. On the optimum design of fuzzy logic controller for trajectory tracking using evolutionary algorithms. *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, 1:660 – 665, Dec. 2004.

[59] Hornby G.S.; Takamura S.; Yamamoto T.; Fujita M. Autonomous evolution of dynamic gaits with two quadruped robots. *Robotics, IEEE Transactions on*, 21(3):402–410, June 2005.

[60] Carlos M. Fonseca; Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[61] Hishashi Tamaki; Hajime Kita; and Shigenobu Kobayashi. Multi-objective optimization by genetic algorithms : A review. *Proceedings of the 1996 International Conference on Evolutionary Computation, IEEE*, pages 517–522, 1996. Nagoya, Japan.

[62] R. Thomson; T. Arslan. An evolutionary algorithm for the multi-objective optimisation of vlsi primitive operator filters. *Congress on Evolutionary Computation (CEC'2002)*, 1:37–42, 2002.

[63] K.C. Tan; Tong H. Lee; E.F. Khor. Evolutionary algorithm with goal and priority information for multi-objective optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, 1:106–113, 1999. Washington D.C.

[64] Qin yong-fa; Zhao ming yang. Research on a new multiobjective combinatorial optimization algorithm. *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pages 187 – 191, Aug. 2004.

[65] Michael Erdmann; Tomas Lozano-Perez. On multiple moving objects. 1986.

[66] evin Dixon; John Dolan. Rave: A real and virtual environment for multiple mobile robot. *Systems K.*

[67] Christopher Clark, Stephen M. Rock, and Jean-Claude Latombe. Motion planning for multiple mobile robot systems using dynamic networks.

[68] Y. Guo; L. Parker. A distributed and optimal motion planning approach for multiple mobile robots, 2002.

[69] Maren Bennewitz; Wolfram Burgard. A probabilistic method for planning collision-free trajectories of multiple mobile robots,. *14 th European Conference on Artificial Intelligence*, 50(2):174–188, Feb 2000.

[70] Nishi T.; Ando M.; Konishi M.; Robotics. Distributed route planning for multiple mobile robots using an augmented lagrangian decomposition and coordination technique. *Robotics and Automation, IEEE Transactions on*, 21(6):1191–1200, 2005.

[71] et al. K. Madhava Krishna. Reactive navigation of multiple moving agents by collaborative resolution of conflicts. 2005.

[72] K.Kant; S.W.Zucker. Towards efficient trajectory planning: the path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.

[73] Y. Guo; L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2612–2619, 2002.

[74] T. Balch; R. C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Trans. Robott. Automat.*, 14:926–939, 1998.

[75] M. Egerstedt; H. Xiaoming. Formation constrained multi-agent control. *IEEE Trans. Robot. Automat.*, 17:947–951, 2001.

[76] R.Fierro; A.K.Das; V.Kumar; J.P.Ostrowski; J.Spletzer; J.Taylor. A vision-based formation control framework. *IEEE Trans. Robot. Automat.*, 18:813–25, 2002.

[77] J. M. Esposito; V. Kumar. Closed loop motion plans for mobile robots. *in Proc. IEEE Int. Conf. Robot. Automat. (ICRA00)*, 3:2777–2782, 2000. San Francisco, CA.

[78] Kar-Han Tan; Lewis M.A.;. Virtual structures for high-precision cooperative mobile robotic control. *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, 1:132–139, 1996.

[79] J. P. Desay; V. Kumar; P. Ostrowski. Control of change in formation for a team of mobile robots. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA99)*, 2:1556–1561, 1999. Detroit, MI.

[80] J. P. Desay; J. P. Ostrowski; V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Trans. Robot. Automat.*, 17:905–908, 2001.

[81] Alan K. Mackworth. On seeing robots. Technical Report TR-93-05, 1993.

[82] Robot World Cup Initiative. www.robocup.org.

[83] *http://www.fira.net/.*

[84] Y. J. Kim K. T. Seow J. H. Kim, D. H. Kim. *Soccer Robotics (Springer Tracts in Advanced Robotics)*. Sep. 2004.

[85] J.-H. Kim K.-H Park, Y.-J. Kim. Modular q-learning based multi-agent cooperation for robot soccer. *Robotics and Autonomous Systems*, 35(2):109–122, May 2001.

[86] Prahlad Vadakkepat J.-H. Kim. Multi-agent systems: A survey from the robot-soccer perspective. *Intelligent Automation and Soft Computing*, 6(1):3–18, Jan 2000.

[87] Lynne E. Parker. Current state of the art in distributed autonomous mobile robotics. In George Bekey Lynne E. Parker and Jacob Barhen, editors, *Distributed Autonomous Robotic System 4*, pages 3–12. Springer-Verlag, Tokyo, October 2000. ISBN: 4-431-70295-4.

[88] Peter Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, Massachusetts, 2000.

[89] Y.Neimark; N.A.Fufaev. Dynamics of nonholonomic systems. *American Mathematical Society Translations*, 33, 1973.

[90] R.M.Murray; Z.Li; S.S.Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, March 1994.

[91] A.Bloch; M.Reyhanoglu; N.H.McClamroch. Control and stabilization of nonholonomic dynamic systems. *Automatic Control, IEEE Transactions on*, 37(11):1746–1757, 1992.

[92] O.J.Sordalen. Conversion of the kinematics of a car with n trailers into a chinaed form. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 382–387, 1993.

[93] D.Tilbury. Exterior differential system and nonholonomic motion planning. *Memorandum No. UCB/ERL M94/90, Electronics Research Laboratory, University of California*, 1994.

[94] P.Muir; C.Neuman. Pulse-width modulation control of brushless dc motors for robotic applications. *Proceedings of the 27th Midwest Symposium on Circuits and Systems*, June 1984.

[95] J.Ganssle; M.Barr, editor. *Embedded System Distionary.* CMP Books, 2003.

[96] Kimon P. Valavanis; Timothy Hebert; Ramesh Kolluru; Nikos Tsourveloudis. Mobile robot navigation in 2-d dynamic environments using an electrostatic potential field. *IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(2):187–196, March 2000.

[97] Elon Rimon; Daniel E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans on Robotics and Automation*, 8(5):501–518, 1992.

[98] Nikos C. Tsourveloudis; Kimon P.Valavanis; Timothy Hevert. Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic. *IEEE Transaction on Robotics and Automation*, 17(4), 2001.

[99] Paul A. Tipler. *Physics for Scientists and Engineers: Volume 2 — Fourth edition.* W.H. Freeman/Worth Publishers, New York, 1999.

[100] William H. Hayt; Jack E. Kemmerly. *Engineering circuit analysis 5th Edition.* McGraw-Hill, New York, 1993.

[101] J.-H. Kim D.-H. Kim. A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. *Robotics and Autonomous Systems*, 42(1):17–30, Jan 2003.

[102] D.-S. Kim Y.-J. Kim, J.-H. Kim. Evolutionary programming-based univector field navigation method for fast. *IEEE Trans.on Systems Man and Cybernetics- Part B - Cybernetics*, 31(3):45–458, Jun 2001.

[103] K.-C. Kim J.-H. Kim P. Vadakkepat D.-H. Kim, Y.-J. Kim. Vector field based path planning and petri-net based role selection mechanism with q-leanring

for the soccer robot system. *Intelligent Automation and Soft Computing*, 6(1):75–88, Jan 2000.

[104] Steven Ratering; Maria Gini. Robot navigation in a known environment with unknown moving obstacles. *Autonomous Robots*, 1(2), 1995.

[105] CW Lim; SY Lim; MH Ang Jr. Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment. *IEEE International Symposium on Intelligent Control*.

[106] Dan O. Popa; Chad Helm; Harry E.Stephanou; Arthur C. Sanderson. Robotic deployment of sensor networks using potential fields. *Proceedings of the 2004 IEEE, ICRA*.

[107] Y.Wang ; G.S.Chirikjian. A new potential field method for robot path planning. *Proceedings of the 2000 IEEE Int. Conference on Robotics & Automation*, 2:977 –982, April 2000. San Francisco, CA.

[108] A. Poty; P. Melchior; A. Oustaloup. Dynamic path planning for mobile robots using fractional potential field. 2004.

[109] Hiroshi Igarashi; Masayoshi Kakikura. Path and posture planning for walking robots by aritificial potential field method. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA*, April 2004.

[110] Frank E. Schneider; Dennis Wildermuth. A potential field based approach to multi robot formation navigation. *Proceedings of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*.

[111] Atsushi Yamashita; Tamio Arai; Jun Ota; Hajime Asama. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, (2), April 2003.

[112] Biliang Zhong; Qi Zhang; Yimin Yang. Real time reactive strategies based on potential fields for robot soccer. *Proceedings of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing.*

[113] Jing Ren; Kenneth A. Mclsaac. A hybrid-system approach to potential field navigation for a multi-robot team. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, September 2003.

[114] J.Barraquand; J.C.Latombe. A monte-carlo algorithm for path planning with many degrees of freedom. *Proceeding of the IEEE International Conference on Robotics and Automation*, 3:1712–1717, 1990.

[115] P. Tournassoud. A strategy for obstacle avoidence and its application to multi-robot systems. *Proceeding of the IEEE International Conference on Robotics and Automation*, 3:1224–1229, 1986.

[116] Jin-Oh Kim; Pradeep K. Khosla. Real-time obstacle avoidance using harmonic potential funtions. *IEEE Trans. on Robotics and Automation*, 8(3):338–349, 1992.

[117] Ahmad A. Masoud. Integrating directional constraints in motion planning using nonlinear, anisotropic, harmonic potential fields. *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, pages 14–17, 1998 Gaithersburg, MD.

[118] Liu Chengqing; Marcelo H Ang Jr; Hariharan Krishnan; Lim Ser Yong. Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 983–988, April 2000.

[119] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, Oxford, 1996.

[120] S. Nolfi. Evolutionary robotics: Exploiting the full power of selforganization. 1998.

[121] K.-Y Im; S.-Y. Oh; S.-J. Han. Evolving a modular neural network-based behavioral fusion using extended vff and environment classification for mobile robot navigation. *IEEE Transactions on Evolutionary Computation*, 6(4):413–419, 2002.

[122] Gerry V. Dozier, Shaun McCullough, Abdollah Homaifar, and Loretta Moore. Multiobjective Evolutionary Path Planning via Fuzzy Tournament Selection. In *IEEE International Conference on Evolutionary Computation (ICEC'98)*, pages 684–689, Piscataway, New Jersey, May 1998. IEEE Press.

[123] J.Andersson. A survey of multiobjective optimization in engineering design. *Technical Report No. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linkoping University*, 2000.

[124] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 93–100, 1985. Hillsdale, New Jersey.

[125] C.M. Fonseca; P.J.Fleming. An overview of evolutionary algorithms in multiobjective optimizaion. *Evolutionary Computation*, 3(1):1–16, 1995.

[126] D.A.V. Veldhuizen; G.B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.

[127] C.A.C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, 1999.

[128] Kanta Tachibana; Takeshi Furuhashi. A structure identification method of submodels for hierarchical fuzzy modelling using the multiple objective genetic algorithm. *International Journal of Intelligent Systems*, 17(5):496–513, 2002.

[129] P.Vadakkepat; T.H. Lee; X. Liu. Application of evolutionary artificial potential field in robot soccer system. *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pages 2781–2785, 2001.

[130] Carlos A. Coello Coello. An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–144, June 2000.

[131] Ritzel B.J.; Eheart J.W. An investigation of niche and species formation in genetic function optimization. pages 42–50, 1989.

[132] Kursawe F. A variant of evolution strategies for vector optimization. *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, pages 193–197, 1991. New York.

[133] Goldberg D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Professional, 1989. Reading, MA.

[134] Carlos M. Fonseca; Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[135] David A. van Veldhuizen; Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.

[136] Fonseca C.M.; Fleming P.J.D. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms and Their Applications*, pages 141–153, 1993. New York.

[137] Kalyanmoy Deb; David E. Goldberg. An investigation of niche and species formation in genetic function optimization. pages 42–50, 1989.

[138] Horn J.; Nafpliotis N. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, Urbana, Illinois, USA, 1993.

[139] K. Srinivas N.; Deb. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[140] Tan K.C.; Lee T.H. Moea toolbox for computer aided multi-objective optimization. 2000.

[141] E.R.Timothy; R. McCartney. A cost term in an evolutionary robotics fitness function. *Congress on Evolutionary Computation. Proceedings of*, 1.1:125–132, 2000.

[142] A.S.Rana; A.M.S.Zalzala. An evolutionary algorithm for collision free motion planning of multi-arm robots. *Genetic Algorithms in Engineering Systems: Innovations and Applications First International Conference on*, pages 123 –130, 1995.

[143] Kar-Han Tan; Lewis M.A.;. Virtual structures for high-precision cooperative mobile robotic control. *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, 1:132–139, 1996.

[144] E. Beadle; P. Ujuric. A fast-weighted bayesian bootstrap filter far nonlinear model state estimation. *IEEE Tmnsodionr on Aemrpace and Electmnii: Systems, vol. 33, pp. 338 -343, 1997.*

[145] S. Zhou; V. Kmeger; R. Chellappa. Face recognition from video: a condensation approach. *Fifth IEEE lntrrnarionul Conference on Automatic Fuce and Cesrure Recognition, pp. 212 -21 7, May 2002.*

[146] W. R. Gilks and C. Bemini. Fallowing a moving target - monte earlo inference for dynamic bayesian models. *Journal of Royal Statistical Society*, 63:127–146, 2001.

[147] R. Aykmyd. Bayesian estimation for hamogencous and inhomogeneous gaussian random gelds. *IEEE Tranractions on Pattern Analysis and Machine Intelligence*, 20:533–539, 1998.

[148] B. Kwolek. Person following and mobile camera localization using particle filters. *Proc. 4th Int. Workshop Robot Motion and Control*, page 265C270, 2004.

[149] C.Hue; L.C.J.-P; P.Perez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50:309–325, 2002.

[150] C.Hue; L.C.J.-P; P.Perez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic System*, 38:791–812, 2002.

[151] P. Clifford J. Carpenter and P. Fearnhead. Improved particle filter for non-linear problems. *Proc. Inst. Elect. Eng., Radar, Sonar, Navig.,*, 1999.

[152] J. S. Liu and R. Chen. Sequential monte carlo methods for dynamical systems. *J. Amer. Statist. Assoc.,*, 93:1032–1044, 1998.

[153] A.Doucet. On sequential monte carlo methods for bayesian filtering. *Submitted for publication. Available as Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering*, 1998.