# COLLISION FREE TRAJECTORY GENERATION USING ARTIFICIAL POTENTIAL FUNCTION AND STEREO VISION SYSTEM

By

RONGHUA YAO

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTOR OF SCIENCE

UNIVERSITY OF FLORIDA

2013

To my mother and father

ACKNOWLEDGMENTS

I would personally like to thank my supervisory committee chair (Dr. Norman Fitz-Coy) for giving me the opportunity to do this research. I would like to thank him for his patience and kindness in giving me advice and instructing me in finishing my master's thesis. I would like to thank the University of Florida and Mechanical and Aerospace Engineering Department for giving me the chance to study in the United States. I would also like to thank the professors, friends and people who have given me help while studying abroad. I would like to thank my supervisory committee members (Dr. Gloria Wiens and Warren Dixon) for their assistance in validating my master's research. Finally I would like to thank my parents and family, without their support I would not have had the opportunity study abroad and none of this would be possible.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| APF | Artificial Potential Function |
| DVL | Doppler Velocity Log |
| GPS | Global Positioning System |
| INS | Inertial Navigation System |
| NASA | National Aeronautic & Space Administration |
| SAD | Sum of Absolute Difference |
| SSD | Sum of Squared Difference |
| SVS | Stereo Vision System |
| TOF | Time of Flight |

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

# COLLISION FREE TRAJECTORY GENERATION USING ARTIFICIAL POTENTIAL FUNCTION AND STEREO VISION SYSTEM

By

Ronghua Yao

May 2013

Chair: Norman Fitz-Coy
Major: Mechanical Engineering

Mobile robot path planning is one of the most important research domains of robot technology. Most self-navigating autonomous robots work in an environment that is filled with complex and unpredictable obstacles. The robot must work in this environment and avoid collision with the obstacles. The sensors for environment perception and robot self-status perception are implemented on the robot to avoid the obstacles. Usually the sensors used for perception are divided into two parts: one part is the passive sensors like passive sonar and stereo vision system, and the other is the active sensors like time of flight (TOF) optical sensors. In this thesis, stereo vision system is used for environment perception. The environment model can be built based on the position and local information of obstacles which is provided by the stereo vision system.

In this research, artificial potential function (APF) is used as the path planning method. This is because the APF method has an advantage in real time obstacle avoidance and is convenient for practical use. However the APF method fails to take the specific robot dynamic model into consideration. Also there is the local minimal problem for this method. The improvement for the APF method is presented in this thesis. In this

research a collision free path is generated by combining the stereo vision system and the APF method. The normal stereo vision system is improved by adding the memory to remember the detected obstacles. In this way image processing methods can be used to improve the drawbacks of the APF theory. In most practical implementation of robots, the robot is desired to finish its mission with minimal cost of time and fuel. Due to this requirement, optimal time or optimal fuel trajectory should be generated for the robot to track when it moves in the workspace. Also, the kinematic profile corresponding to the optimal trajectory should be generated to guide the motion of the robot. For further research, two kinematic smoother trajectory generation methods are discussed and compared. Finally the improved stereo vision system and time optimal trajectory generation method are evaluated by simulation.

# CHAPTER 1
# INTRODUCTION

## Motivation

The use of robotics for transportation and operation in agriculture, mining, industry, commerce and home service has become more accessible with the development of robot technology. In these domains, compared to the work of people, robots can offer more efficient and precise performance, improve safety, reduce the cost for labor training and some other expenses. Mobile robotics is one branch of robotics that is commonly used in these areas. Usually the mission for mobile robots is to autonomously transit from a known launch site to a given target location. These robots always work in unknown environments and terrains, so they have to plan the moving path on time with the information provided by perception sensors. As the unknown working environment is complex and unpredictable, different sensors must be added to the robots for various tasks. Robots cannot directly use the global position system (GPS) for navigation in this kind of unknown environment. The self-location of mobile robots is provided by GPS or a self-generated guiding map. Furthermore, in autonomous robot service the cost of time and energy should be considered as greatly influencing in the robot performance. Therefore minimizing time and fuel costs during robots' service and ensuring robots can finish its tasks while avoiding collision with obstacles plays a crucial role in improving the efficiency of robots. This thesis provides the method for mobile robots to plan the path in an unknown working environment, the perception sensor in this thesis used for perception is the stereo vision system. Also, a lot of research institutes and companies in these different domains work on mobile

robots working in unknown environments. Following section gives the examples of the application of the mobile robot in different areas.

## Mission Examples

One primary application of the mobile robot is the indoor delivery robot. The work environment for this kind of robot is usually in a home, restaurant or laboratory. The situation and obstacles in these areas are complex and unpredictable, the delivery robot has to go across these areas and travel around people and certain places. Figure 1-1 A) shows the Adapt SPC-4200 indoor robot and the Adapt Lynx self-navigating indoor vehicle, these two robots are designed for delivering goods in challenging environments that may include narrow and crowed areas as well as dynamic and peopled locations [1-2]. They are both self-navigating autonomous robots that can generate paths and avoid collisions based on the sensors. Willow Garage has also developed the indoor delivery robot, TurtleBot, shown in Figure 1-1C). This robot is a stereo vision based self-navigating robot and can be improved as a platform for other tasks by combining its stereo vision system [3].


A B C

Figure 1-1. Examples of self-navigation robots. A) The Adapt SPC-4200 indoor robot, B) Adapt Lynx self-navigating indoor robot, C) Indoor robot TurtleBot produced by Willow Garage [1-3].

In extreme environments not accessible to people, self-navigating autonomous robots are designed to replace people in doing the work such as experiments, excavating, and exploring. Usually in this environment, there are unpredictable obstacles like rocks and pits, and a robot has to arrive at the destination to finish tasks without collision with these obstacles. Bluefin Robotics developed their Bluefin-21 underwater vehicle for underwater missions. The robot is relies on INS, DVL and SVS for navigation when working underwater [4]. FMC Technologies also developed their Ultra Heavy-Duty robotic system for underwater tasks using automatic navigation and positioning system [5]. The automatic robots working in this environment not accessible to people highly improved the safety and efficiency of the working environment. In 2007 the National Aeronautic & Space Administration (NASA) began to sponsor the Regolith Excavation Competition, which provides reference for further lunar mining robotics designs [6]. People cannot work openly on the lunar surface, and robots working there cannot use GPS for navigation and localization. For these reasons, self-navigating mining robots are needed to work in this environment. They should be able to find a safe path to navigate, avoiding collisions with potentially dangerous obstacles using sensors. In order for this to work, path planning and obstacle avoiding algorithms for this self-navigating robot should be developed to improve the work efficiency and safety.


Figure 1-2. Bluefin-21 underwater vehicle developed by Bluefin Robotics [4].

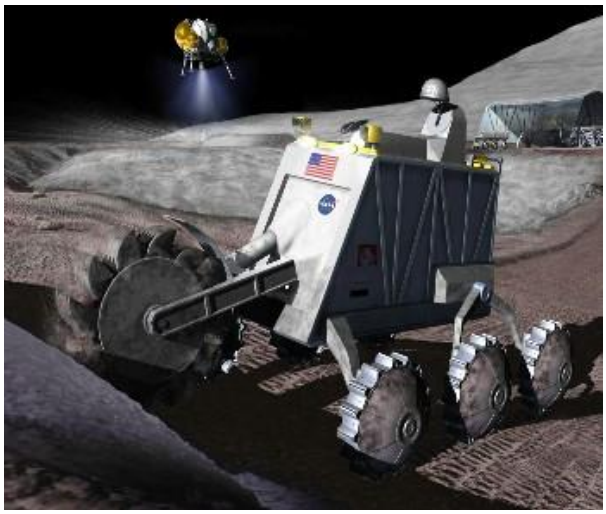


Figure 1-3. Regolith Excavation Competition sponsored by NASA [6].

**Review**

The APF method is commonly used for real time robot path planning. Khatib applied the APF method on robots to produce collision free path [7]. However, it is very easy for a robot to be trapped into a local minimum when using the normal APF method, so Kim, J. O. and Khosla used the harmonic function as the potential function for path planning [8]. This is because the harmonic function will not generate the local minimum in the potential field. Mabrouk and McInnes, Min gyu Park and Jae hyun Jeon also provided some other methods to solve the local minimal problem [9-10]. But the normal APF method is not perfect in avoiding collisions with moving obstacles, S.S. GE and Y.J. CUI improved the normal APF method for avoiding collisions with moving obstacles by taking the velocity into consideration [11]. This highly improved the robot‘s performance in tracking the moving destination and avoiding moving obstacles.

One of the most important components for robots to avoid collision with obstacles is the ability to detect the environment. The stereo vision system is one of the passive sensors for environment perception. The depth map generation algorithm of stereo

vision system is introduced in the work of Ali Kilic and Maryum F. Ahmed [18-22]. In autonomous robot localization and navigation, a map is the essential component. M. Hebert proposed the occupied grid map to describe the environment for robot self-navigation [23]. Now, the stereo vision system is widely implemented on different kinds of robots for obstacle detection and navigation, Allan Eisenman and Steven B. Goldberg implemented the stereo vision system on mars rover [16-17]. In the work of Don Murray, the stereo vision system is used to build a grid map for real time navigation [20]. The technology to use the stereo vision system for navigation is developing quickly and greatly promotes the development of robotics.

When a robot moves in the workspace, its movement is constrained by the limitation of velocity, acceleration and jerk. The path generated from the APF method is a series of discrete sets of points, the corresponding trajectory should be generated for the robot to track, the corresponding kinematic profile also should be generated to guide the robot. V. Munoz prompted the trajectory generation method to generate the trajectory constrained by velocity and acceleration [24]. K. Petrinec and Z. Kovacic provided the trajectory generation method with consideration of jerk continuity and jerk limitation [25]. In practical applications, robots are desired to finish the tasks in minimal time. Marko Lepetic developed the time optimal trajectory generation method [26]. Sonja Macfarlane [27] and Imran Waheed [28] provide two different methods for robot trajectory generation and implemented their method on specific robots for evaluation.

## Research Scope

This thesis develops the collision free trajectory generation method by combining the APF method and the improved stereo vision system. Chapter 2 explores the method that uses APF to avoid obstacles with specific contours. In Chapter 2 the obstacle

configuration is discretized to build the APF potential field, and simulation examples are provided to evaluate the performance of this method to avoid obstacle with specific contours. The drawbacks of the normal APF method in path planning are also discussed in Chapter 2, and these drawbacks can be solved by the improving stereo vision system which is discussed in later Chapter 4.

Chapter 3 details the kinematic profile generation method's performance in tracking the interpolated smooth collision free trajectory. In the first part of Chapter 3, the positive and negative aspects of two trajectory interpolation methods are discussed in generating the smooth velocity and acceleration profile, as well as the method to rescale kinematic profiles to meet constrains of robots' physical limitations. In the second part of Chapter 3, the time optimal trajectory generation method is developed. In this part the generation method of time optimal velocity and acceleration profiles is developed.

Chapter 4 develops the improved stereo vision system to improve the limitations of the APF method. In Chapter 4 the depth perception principle of the stereo vision system is discussed and evaluated by simulation. Furthermore a gridded map is developed in order to memorize the obstacle information. The solution to the local minimum problem of the APF method and other improvements of the APF method are discovered based on the processing of this grid occupied map.

In Chapter 5, a specific scenario is constructed to simulate the performance of this time optimal collision free trajectory generation method. The improved stereo vision system is used in the simulation for environment perception. Two smooth trajectory generation methods and the corresponding kinematic profiles generation methods are

compared by simulation to evaluate the time optimal trajectory generation method. Finally in Chapter 6 the conclusion of the research in this thesis is made and further research planning is prompted.

# CHAPTER 2
# PATH PLANNING WITH ARTIFICIAL POTENTIAL FUNCTION

## Theoretical Development

Artificial potential function (APF) is used to describe the working space for robots. In the APF method, the goal produces an attractive force on the robot and the obstacles produce repulsive forces on robot. With the force imposed by the environment, the robot can avoid collisions with obstacles and reach the goal from any initial position.

By using the APF method, the obstacles, goal and robot are simplified as some points. The robot's working space can be described as a potential field by APF, and each position in the working space has a corresponding value. The goal has the minimum value in the potential field and the obstacles have relatively high values. The gradient on each position of the potential field gives the fastest direction to arrive at the adjacent relatively small value position [14-15]. So in the potential field, robots can get to the destination from the initial position with the direction of the gradient. APF has two parts: the attractive potential function and the repulsive potential function. In this research the APF form proposed by Khatib [1] is used. The attractive potential function and attractive force are expressed as:

$$U_{attr} = \frac{1}{2} k\rho^2(X, X_g) \tag{2-1}$$

$$F_{attr} = -\nabla[U_{attr}(X)] = k\rho(X, X_g) \tag{2-2}$$

Where $\rho(X, X_g)$ is the relative distance between a certain position and the goal in the potential field. The attractive force is obtained by taking the derivative of the attractive potential function. The repulsive potential function and repulsive force are expressed as:

$$U_{rep}(X) = \begin{cases} 0.5\eta \left(\frac{1}{\rho(X,X_0)} - \frac{1}{\rho_0}\right)^2, \rho(X, X_0) \le \rho_0 \\ 0 \qquad\qquad\qquad\quad , \rho(X, X_0) > \rho \end{cases} \tag{2-3}$$

$$F_{rep}(X) = -\nabla\left[U_{rep}(X)\right] = \begin{cases} \eta\left(\frac{1}{\rho(X,X_0)} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2(X,X_0)} & ,\rho(X,X_0) \le \rho_0 \\ 0 & ,\rho(X,X_0) > \rho \end{cases} \quad (2\text{-}4)$$

Where $\rho(X, X_0)$ is the relative distance between a certain position and obstacles in the potential field, and $\rho$ is the maximum influence distance of the repulsive force. From Eq. (2-3) and Eq. (2-4), there is no repulsive force beyond the influence area. The repulsive force is obtained by taking the derivative of the repulsive potential function. The total force on the robot is expressed as:

$$F_{total} = F_{attr} + F_{rep} \quad (2\text{-}5)$$

The force imposed on the robot in the potential field is the resultant force of attractive force and repulsive force. With this resultant force the robot moves from its initial point to its destination.

## APF for Obstacles with Specific Configuration

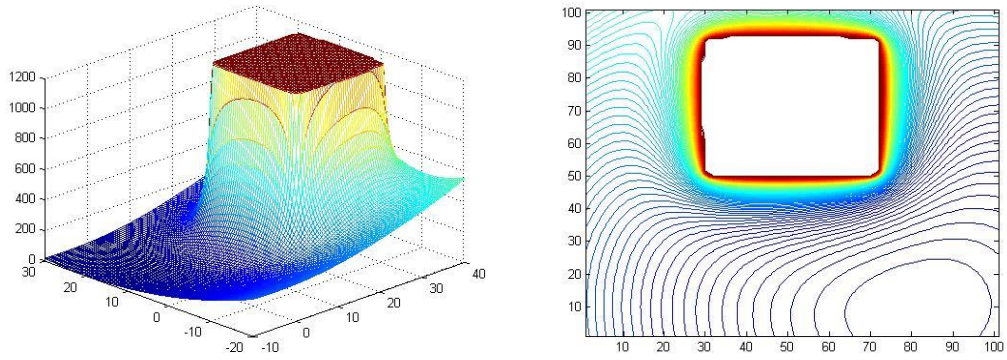
### Potential Field of Obstacles

In the APF method, the obstacles are simplified as a series of points, but in practical application the obstacles have specific contours. In some circumstances, by choosing appropriate attractive and repulsive coefficients the robots can avoid collision with obstacles and arrive at the destination. This may be appropriate in the condition in which the size of obstacles is very small compared to the relative distance [12]. But in most practical problems, there are obstacles with complex contours in the workspace of the robot. In this case robots have to work and move around these obstacles. In building the potential field of the robot's working space, these obstacles must be modeled.

Usually obstacle configuration is described as a combination of a series of splines, but these splines cannot be directly expressed in the APF, so it can be discretized into a series of points. Each of the discretized points will generate the

repulsive force in its area of influence. The final repulsive force on a certain position is the sum of repulsive forces from different discretized points.

The proper attractive and repulsive coefficient should be chosen to generate the potential field. Figure 2-1 A) shows the potential field of a rectangular obstacle, the minimum value stands for the position of the goal, the relatively high value is under the influence of the rectangular obstacle. There is a threshold value for the repulsive APF, so the area inside the contour of the rectangular obstacle has the some value in the potential field. Figure 2-1 B) gives the contour line map of the potential field. In this map, the gradient value, the position of the goal and obstacles are clearly showed.




A B

Figure 2-1. Example showing the obstacle using APF. A) Potential field of rectangular obstacle, B) The corresponding Contour line map of rectangular obstacle.

**Limitation of the APF Method**

The APF method is widely used because of its simple computation form. Compared to other methods for path planning, the APF method can generate a collision free trajectory with barely any computation burden. Therefore, APF can be used for real time obstacle avoidance in complex environment especially with moving obstacles. It is facilitated to use APF method for path planning [10].

However, there are also drawbacks of the normal APF method. The obstacles with a concave contour may have local minimum in its potential field. Figure.2-2 shows a case where there is a local minimum. The potential value around the local minimum point is greater, so when the robot moves in this area it will be trapped, the resultant force on robot cannot lead the robot to arrive the destination. Therefore, in building the potential field, the local minimum must be avoided. In 1992, Kim, J. O. and Khosla use the harmonic function to solve local minimum problem [8], this is because the harmonic function has no local minimal in the potential map. Paraskevas Dunias also details the Harmonic Function in his research to solve the local minimal problem [13]. In Chapter 5 the local minimum problem is solved by combining the APF method with stereo vision system.

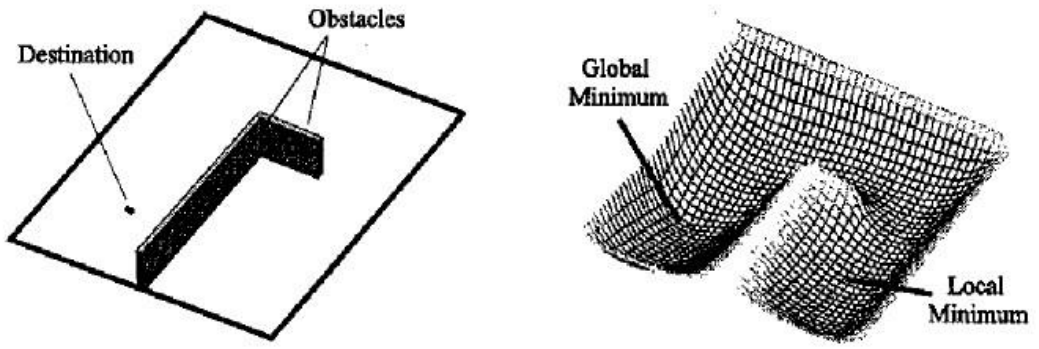



Figure 2-2. The local minimum problem in potential field [13]. A) The right angle concave obstacle, B) The corresponding potential field of the obstacle.

**Static and Moving Obstacle Avoidance**

The numerical simulation examples are presented to demonstrate the APF method for path planning and obstacle avoidance. In the examples, the behavior of the APF method for static obstacles and moving obstacles is evaluated. Two assumptions

are made for further research about the APF method. One is that the robot already knows the contours of the obstacles in its work space before begins, and the other is that the robot has no information about the obstacles in its work space, but it can detect obstacles with the stereo vision system when it moves around. The parameters of the APF method used in the simulation are presented in Table 2-1. Figure 2-3 presents the workspace situation in the simulation, the blue lines stand for static obstacles, the yellow lines stand for moving obstacles, and the robot is simplified as a green point and its detecting area is presented by the circular sector. In the simulation the robot can successfully reach the destination and avoid collision with the obstacles. The simulation results are presented as two parts, one is the simulation with previously known obstacles, the other is with previously unknown obstacles.

Table 2-1. APF method simulation parameters.

| Parameter | Value | Unit |
|---|---|---|
| Initial Position $P_o$ | [0 0] | m |
| Finial Position $P_f$ | [180 170] | m |
| Step Length $L$ | 0.1 | m |
| Attractive Coefficient $k$ | 0.05 | N/A |
| Repulsive Coefficient $\eta$ | 5000 | N/A |
| Detection Radius $R_d$ | 10 | m |
| Repulsive Force Influence Radius $R_f$ | 30 | m |

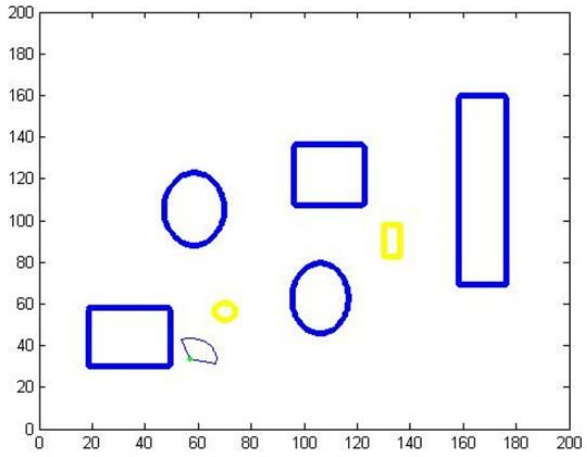



Figure 2-3. Simulation of obstacle avoidance with APF method.

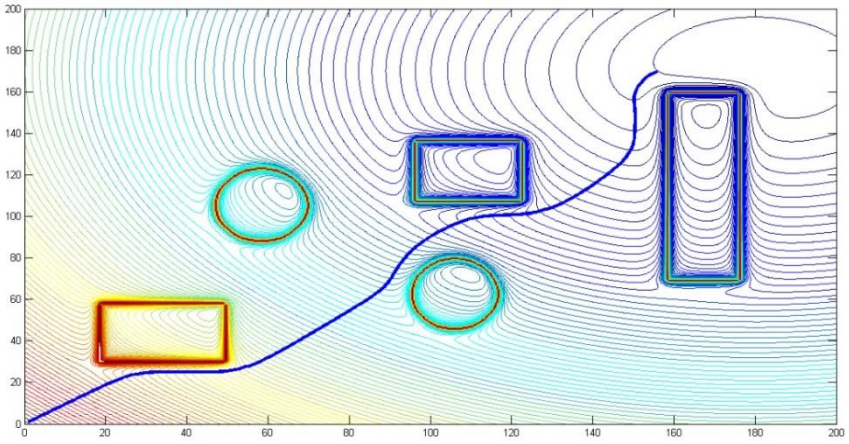



Figure 2-4. Path planning with APF method for previously known obstacles.

As shown in Figure 2-4, the robot knows the contour and location of the obstacles before it starts moving, and the generated path is presented as a blue line. In this situation, the generated path is smooth and collision free. In Figure 2-5, the robot does not know the information of the obstacles before it starts, it detects the environment with each step. With the detected information of the obstacles, the robot calculates the resultant force at that position, and then decides where to go next.

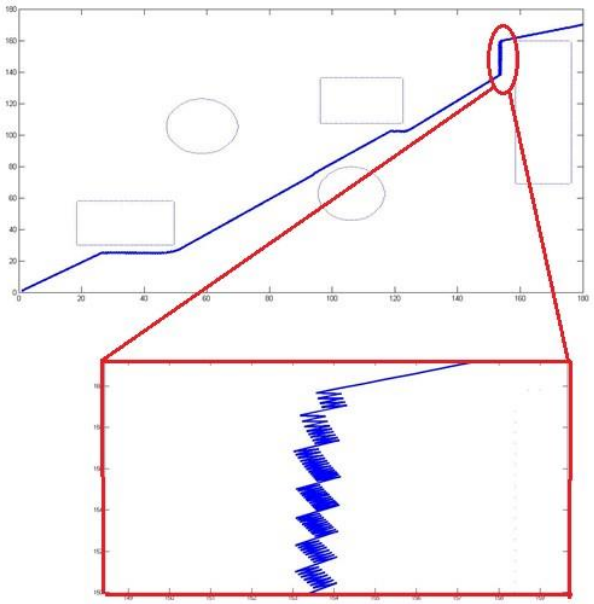


Figure 2-5. Path planning with APF method for previously unknown obstacles.

From the results in Figure 2-5, the generated path is collision free, but it's not as smooth as the situation with previously known obstacles. In this situation, the robot does not know the information of the obstacles previously. The robot calculates each move step by step, based on the information it gathers along the way. Each time when the robot plans the length and direction of the next step, it is only influenced by the repulsive forces from the obstacles which it has detected in that instant, but at that time the robot may still be located in an area influenced by previously detected obstacles. As the influence of the repulsive force on the robot does not change continuously, so the path from the APF method is in a zigzag shape. The zigzag path is more obvious when the robot moves near obstacles. Because of this, the robot consumes more time and fuel. In Chapter 4, the improved stereo vision system is introduced to reform this zigzag path by combining the APF method with the stereo vision system. In these simulations, the robot moves step by step with constant length, but in practical situations there must be a corresponding kinematic profile to make the robot move from one position to another. Furthermore, there is always a demand for minimum time and fuel consumption in practical situations. In Chapter 3 the smoother trajectory and time optimal trajectory generation method is presented.

# CHAPTER 3
# TRAJECTORY GENERATION WITH OBTAINED PATH

The path generated from the APF method is a set of discrete points, for each step the robot moves from one point to another in straight line. As the path is combined by a series of straight segments, there will be sharp turning corners on these discrete points. When the robot arrives at this turning point, the velocity in the moving direction has to decelerate to zero. This costs much time and fuel for a robot moving in this kind of path. The kinematic profiles corresponding to the APF path are also discrete which makes it difficult for the robot to track the generated APF path precisely.

The APF path is a collision free path, its drawback is not smooth. In this chapter, the interpolated smooth spline is introduced to approximate the original APF path. This approximated trajectory will not go across the obstacles and can generate corresponding smooth kinematic profiles. The interpolation control points are chosen from the original APF path. To guarantee precision of the approximation, the intervals between these control points and the intervals between the corresponding variables should be within reasonable values. This is also an important requirement to make the generated new trajectory not cross the obstacles. In order to make the acceleration profile along the trajectory continuous, the variable of the interpolation function on each segment must be at least three orders.

## Interpolation Theory

### Cubic Spline

2D Trajectory can be approximated by interpolating the X and Y coordinates of the original APF path separately with same interpolation variable. Cubic spline is described by the three order polynomial, so there exists the first-order derivative and the

second-order derivative. In other words, the smooth velocity and acceleration profile can be generated from cubic spline trajectory after interpolation with time. Figure 3-1 shows the principle of cubic spline interpolation method for the interpolation of the X coordinate. Fit the given X coordinate $(X_i, X_{i+1})$ of the waypoints $(P_i, P_{i+1})$ during the time interval $(t_i\ , t_{i+1})$ using a cubic polynomial $x_i(t)$.

$$x_i(t) = B_{i,1} + B_{i,2}(t - t_i) + B_{i,3}(t - t_i)^2 + B_{i,4}(t - t_i)^3 \tag{3-1}$$

Where $x_i(t)$ is the interpolation function in this interval and $B_{i,j}$ is the coefficient of the interpolation function. The cubic interpolation function in the left adjacent interval is defined as $x_{i-1}(t)$ and the right is $x_{i+1}(t)$.

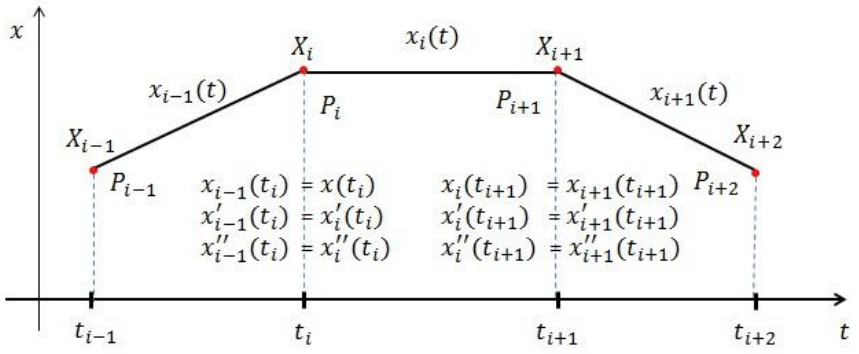



Figure 3-1. Cubic spline interpolation method illustration.

Constraints that make the generated trajectory pass through the waypoints from the original APF path are shown in Eq. (3-2) which makes the interpolated trajectory pass along the original APF path.

$$x_i(t_i) = X_i\ , x_i(t_{i+1}) = X_{i+1} \tag{3-2}$$

The smoothness in velocity and continuity in acceleration are regulated by two extra constraints that are imposed on the adjacent intervals.

$$\begin{cases} x'_{i-1}(t_i) = x'_i(t_i) \\ x''_{i-1}(t_i) = x''_i(t_i) \end{cases} \tag{3-3}$$

$$\begin{cases} x_i'(t_{i+1}) = x_{i+1}'(t_{i+1}) \\ x_i''(t_{i+1}) = x_{i+1}''(t_{i+1}) \end{cases} \quad (3\text{-}4)$$

In practical application of trajectory generation, the initial condition of position and velocity usually have specific demands in different conditions. So the Eq. (3-5) is used as the interpolation function for cubic spline interpolation, the property of this function is detailed in the work of Imran Waheed [28].

$$x_i(t) = B_{i,1}(t-1)^2(2t+1) + B_{i,2}t^2(3-2t) + B_{i,3}t(t-1)^2 - B_{i,4}t^2(1-t) \quad (3\text{-}5)$$

With this cubic polynomial the time variable for each interval is restricted to$(0,1)$. Thus, each parameter of the cubic polynomial stands for the initial position and velocity as well as the final position and velocity.

$$x_i(0) = B_{i,1} = X_{\mathrm{i}} \quad (3\text{-}6)$$

$$x_i(1) = B_{i,2} = X_{\mathrm{i+1}} \quad (3\text{-}7)$$

$$\frac{dx_i}{dt}(0) = B_{i,3} = v_{\mathrm{i}} \quad (3\text{-}8)$$

$$\frac{dx_i}{dt}(1) = B_{i,4} = v_{\mathrm{i+1}} \quad (3\text{-}9)$$

Furthermore, no matter how many segments need to be interpolated, the initial and final velocity condition of the interpolated trajectory can be directly defined as the desired values by using the Eq. (3-5).

**Fourth Order Spline**

Fourth order spline is one order more than the cubic spline, so using fourth order spline for waypoints interpolation can make the acceleration profile smooth and the jerk profile continuous. Jerk continuity is important in jerk bounding circumstance, as the jerk control is related with the control of torque rate. Thus the continuity of jerk will affect the original path tracking accuracy, the continuity of jerk and the smoothness of

acceleration will also affect the life of robotics. Fit the given X coordinate $(X_i, X_{i+1})$ of the waypoints $(P_i, P_{i+1})$ during the time interval $(t_i\,, t_{i+1})$ using a fourth order polynomial $x_i(t)$

$$x_i(t) = B_{i,1} + B_{i,2}(t - t_i) + B_{i,3}(t - t_i)^2 + B_{i,4}(t - t_i)^3 + B_{i,5}(t - t_i)^4 \qquad (3\text{-}10)$$

Where $x_i(t)$ is the interpolation function in this interval, and $B_{i,j}$ is the coefficient of the interpolation function. The cubic interpolation function in left adjacent interval is defined as $x_{i-1}(t)$ and the right is $x_{i+1}(t)$, the specific interpolation algorithm and simulation examples are detailed in [25].

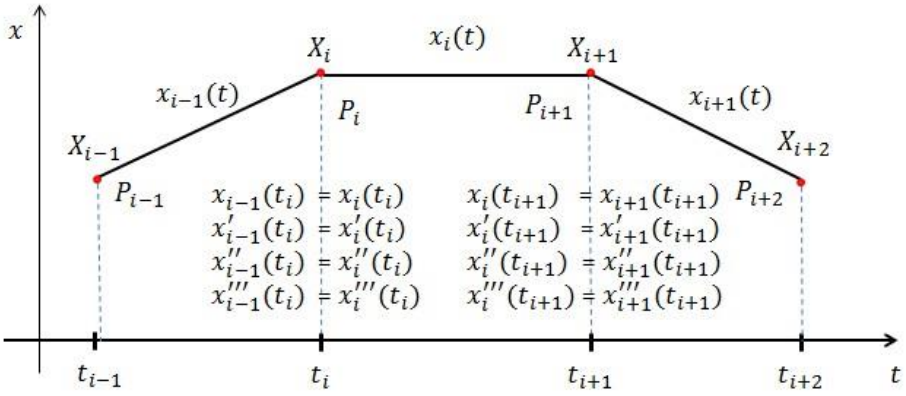



Figure 3-2. Fourth order spline interpolation method illustration.

Constraints that make the generated trajectory pass through the waypoints from the original APF path are shown in Eq. (3-11) which makes the interpolated trajectory pass along the original APF path.

$$x_i(t_i) = X_i\,, x_i(t_{i+1}) = X_{i+1} \qquad (3\text{-}11)$$

The smoothness in velocity, acceleration and continuity in jerk are regulated by three extra constraints that are imposed on the adjacent intervals.

$$\begin{cases} x'_{i-1}(t_i) = x'_i(t_i) \\ x''_{i-1}(t_i) = x''_i(t_i) \\ x'''_{i-1}(t_i) = x'''_i(t_i) \end{cases} \qquad (3\text{-}12)$$

$$\begin{cases} x_i'(t_{i+1}) = x_{i+1}'(t_{i+1}) \\ x_i''(t_{i+1}) = x_{i+1}''(t_{i+1}) \\ x_i'''(t_{i+1}) = x_{i+1}'''(t_{i+1}) \end{cases} \tag{3-13}$$

In most conditions, the robot starts moving from static statues and when it reaches its destination it should keep static at that point. So on the initial and final points of the trajectory, the velocity and acceleration should be zero. For this fourth order spline the initial condition is defined by the fourth order interpolation function discussed before. However, the interpolation function on the final segment should be one order more than the other segments to make the final velocity and acceleration reach the desired value at the final point of the trajectory. The method to generate the final segment interpolation function is presented in [25].

**Features of Cubic and Fourth Order Spline**

For the cubic and fourth order spline interpolation method, no matter which one is chosen to approximate the original path, the chosen waypoint is the only constraint to regulate the generated trajectory passing through the same region as the original path does. The generated trajectory must pass through these waypoints, as the coordinates of the waypoints are expressed in the interpolation functions as constraints. The smoothness of velocity profile is determined by the continuity of its corresponding acceleration profile, the smoothness of acceleration profile is determined by the continuity of its corresponding jerk profile.

As Fourth order spline is one order higher than the cubic spline, so the third order derivative of fourth order spline is continuous but the third order derivative of cubic spline is discrete. So the jerk profile of cubic spline is discrete and the jerk profile of fourth order spline is continuous. For this reason, the acceleration profile of fourth order

spline is smoother than that of the cubic spline. Also as the fourth order polynomial is one order higher than the third order polynomial, the graph of fourth order polynomial oscillates much more than the graph of third order polynomial. As cubic spline is interpolated by the third order polynomial and the fourth order spline is interpolated by the fourth order polynomial, so the trajectory generated by the fourth order method oscillates much more than that generated by cubic spline method. Figure 3-3 shows that the trajectory generated by the fourth order oscillates much more than the cubic spline trajectory and the kinematic profile corresponding to fourth order trajectory oscillates more than that from cubic trajectory. But the acceleration profile is smoother than the cubic spline method.

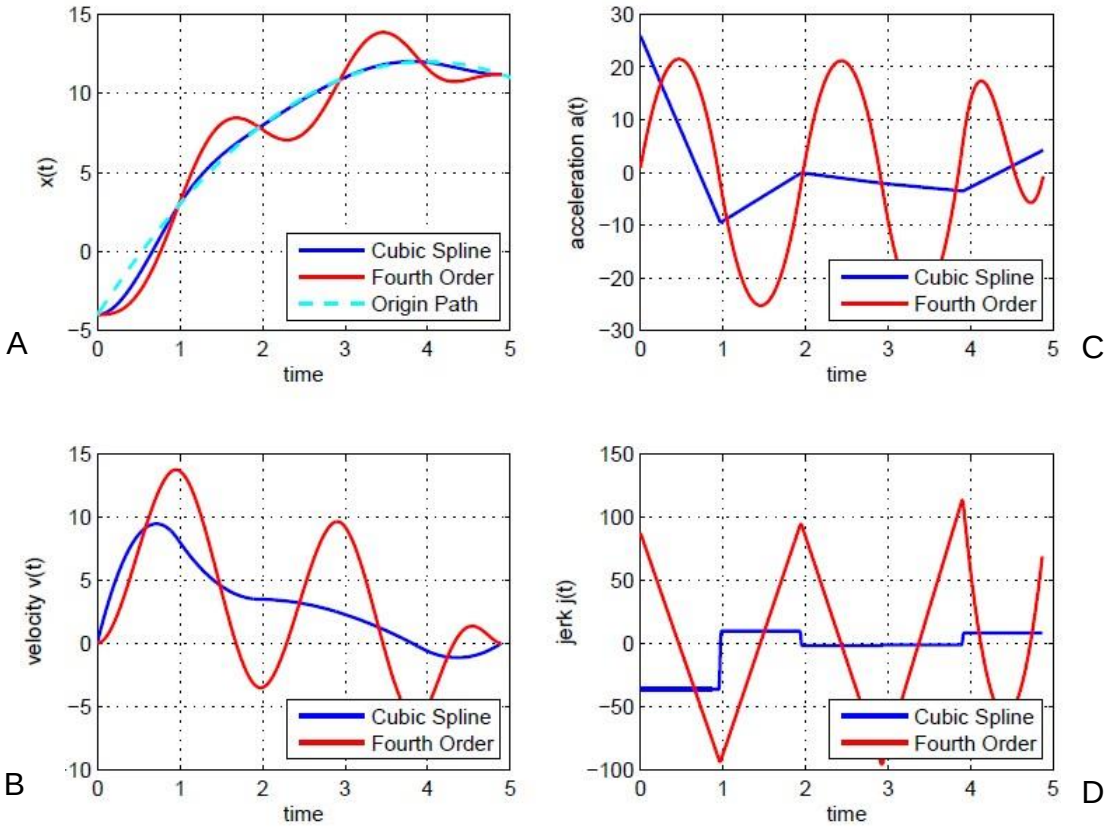



Figure 3-3. Trajectories and kinematic profiles generated by the cubic and fourth order interpolation methods, trajectories are interpolated with 5 waypoints. A) Trajectory, B) Velocity profile, C) Acceleration profile, D) Jerk profile.

The interpolation function for each segment between two waypoints is different with each other and the only input to calculate the interpolation function is the coordinate of the waypoint. In 2D trajectory generation the inputs are the X coordinate and its corresponding time or the Y coordinate and its corresponding time. So the changing of the time interval of each segment or the changing of chosen waypoints will have an effect on the property of the generated trajectory and the corresponding kinematic profile.

The following discusses the influence of waypoint choosing and time interval in trajectory generation. If more waypoints with closer distances are chosen from the original path, the generated trajectory is more similar to the original path. But in this way, the original path is divided into more segments, as the interpolation polynomial for each segment is different, when the number of polynomials used for interpolation increases, the oscillation times of the generated trajectory increase too. Also the total number of oscillation times of the kinematic profiles will increase. Usually the velocity and acceleration vibrate once in each segment. Sometime choosing more waypoints will connect the waypoints by high curvature curves. Due to the limited turning radius of robot, this will lead to the robot being unable to track the trajectory. In Figure 3-4, three times waypoints are chosen to interpolate the same original path compared to the simulation in Figure 3-3. By comparison, the absolute interpolation error decreases with more waypoints, but the curvature along the generated trajectory increases. Likewise, the oscillation times along the trajectory and its corresponding kinematic profile increase.

Oscillation of the trajectory can lead to the vibration of the velocity and acceleration. As the vibration in acceleration, the robot has to continually change the

states between acceleration and deceleration, which causes more fuel consumption. The vibration in velocity makes the robot unable to stay at a steady speed, so the robot can hardly stay in high speed for long time. This causes the robot to spend more time to reach the destination.

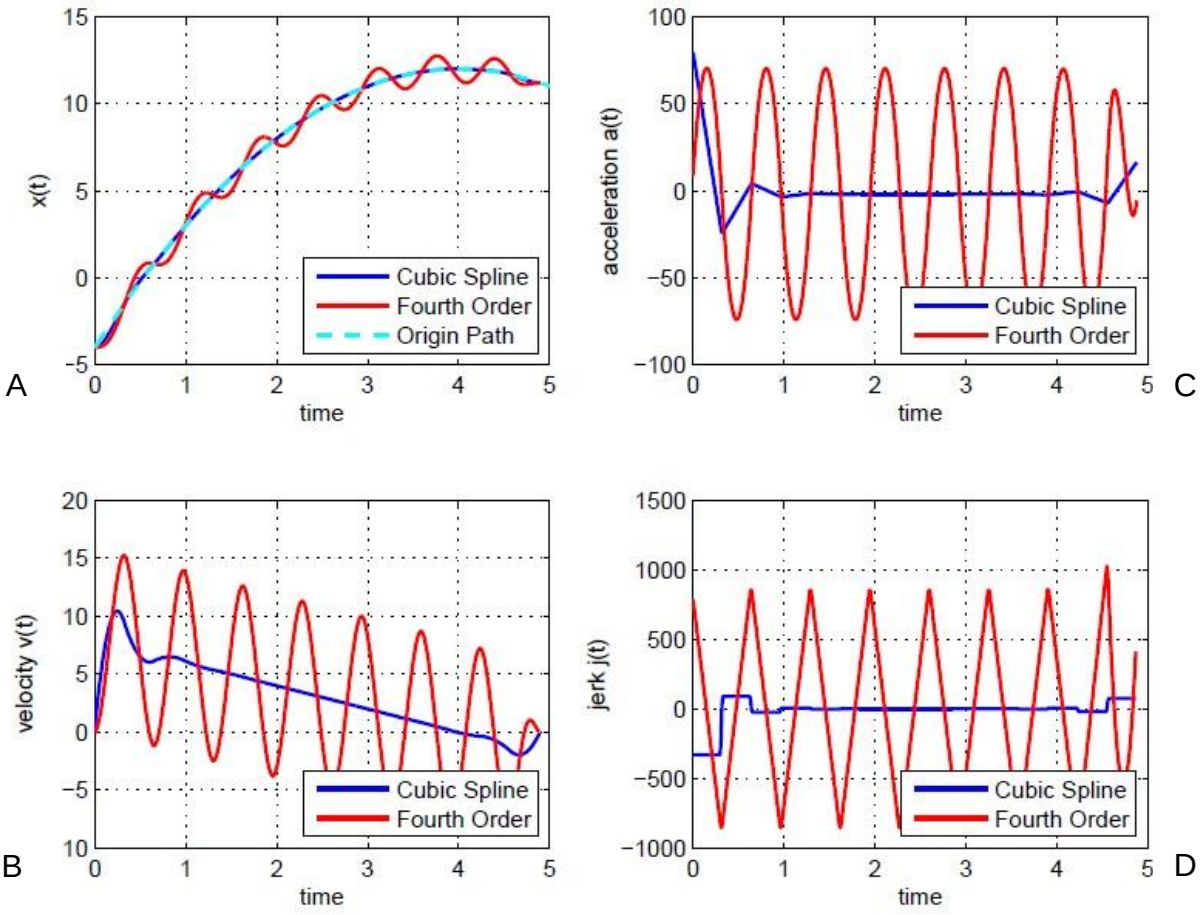



Figure 3-4. Trajectories and kinematic profiles generated by the cubic and fourth order interpolation methods, trajectories are interpolated with 15 waypoints. A) Trajectory, B) Velocity profile, C) Acceleration profile, D) Jerk profile.

In Figure 3-5 the same waypoints are chosen for the interpolation of the original path as it does in Figure 3-3, but in Figure 3-3 the time interval for each segment is the same while in Figure 3-5 the time interval is different with each other. When the time interval changes the magnitude of the oscillation will increase. So in order to generate the optimized trajectory, the distance between waypoints and time interval must be

properly defined. The desired trajectory should make the robot consume the least amount of fuel or cost the shortest time.

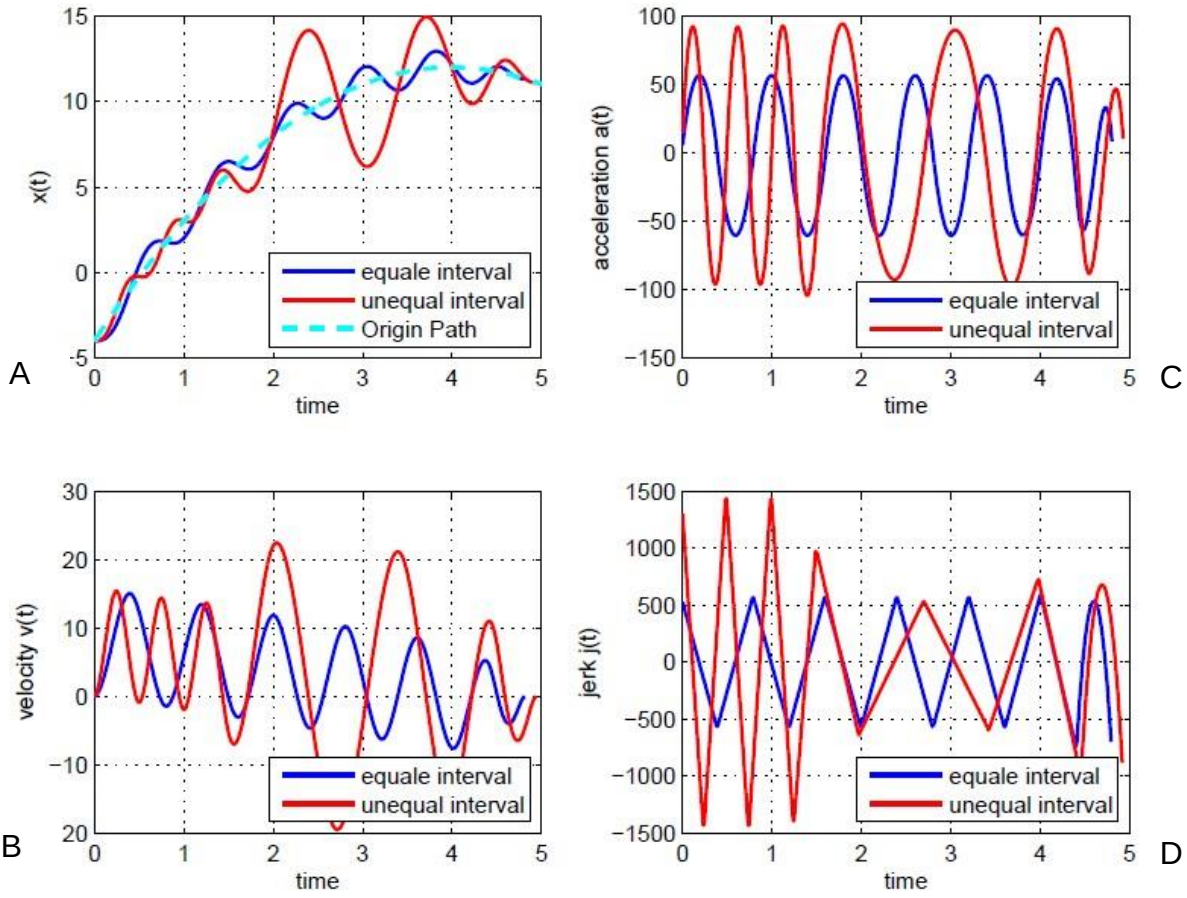



Figure 3-5. Trajectories and kinematic profiles generated by the cubic and fourth order interpolation methods, trajectories are interpolated with different time intervals. A) Trajectory, B) Velocity profile, C) Acceleration profile, D) Jerk profile.

## Trajectory Optimization with Limited Condition

### Physical Limitation in Trajectory Generation

Interpolating robot trajectory from the waypoints of original APF path can guarantee that the robot reaches its destination without collision with obstacles if the robot moves along it. A reasonable trajectory can ensure the robot tracks it perfectly, which means when the robot moves on this trajectory it will not slide. So the velocity, acceleration and jerk related to the trajectory must meet some specific limitations.

All the robot's movements like acceleration, deceleration and turning are caused by the friction force between wheels and the road. The tangential acceleration on the wheel makes the robot speed up and slow down along the tangential direction. The maximum frictional force is related to the friction coefficient and the mass of the robot. The radial acceleration ensures that the robot turns on the curve. The maximum non-sliding turning speed is restricted by both the turning curve radius and the radial friction force on the wheel. So in order to make the robot tracking the generated trajectory perfectly, these non-sliding constraints must be met [24]. Figure 3-6 shows the relation between radial acceleration and tangential acceleration, they should not be beyond the friction ellipse.

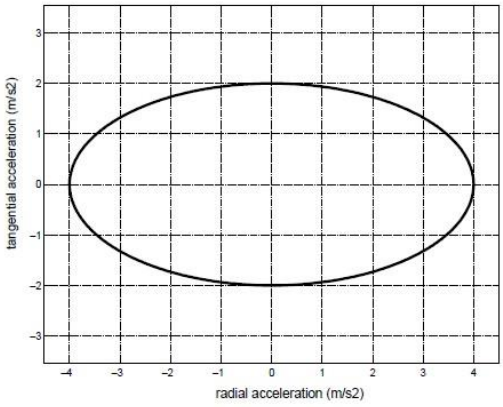



Figure 3-6. Radial and tangential friction ellipse [24].

When the robot moves in a complex obstacle figuration environment, the speed of robot cannot remain high in case of collision with suddenly appearing obstacles. It should leave enough response time for the robot to decelerate.

Usually the moving robot is controlled by several step motors or servo motors. The velocity and acceleration of the robot are related to the speed and torque of the motors, so there also must be some restriction of the velocity and acceleration from the

motors. Furthermore, the limited torque rate in a certain range can extend the life of the robot as well as improve the tracking precision of moving robots [25]. So the jerk must be limited and continuous, in this way the acceleration will become smoother.

**Optimization for Feasible Trajectory**

From the constraints discussed in the previous section, the maximum and minimum of the velocity, acceleration and jerk can be obtained. So the corresponding velocity, acceleration and jerk files of the trajectory must meet the limitations of these extreme values. For the interpolated trajectory, the shape of the trajectory is determined. To reform the restricted dynamic profile without changing the shape of the trajectory, the time interval during this segment must be rescaled. In this thesis the recalling method presented in V. Muiiozt's work is used [24].

Suppose $g_i(t)$ is one trajectory segment in the time interval$(t_i\ , t_{i+1})$, and $v(t)$,$a(t)$ and $j(t)$ are velocity, acceleration and jerk profile. The limitations of these variables are respectively defined as$v_{limit}(t)$, $a_{limit}(t)$ and$j_{limit}(t)$.

$$\lambda_v = \max[|v(t)|]\,/v_{limit}(t) \tag{3-14}$$

$$\lambda_a = \sqrt{\max[|a(t)|]\,/a_{limit}(t)} \tag{3-15}$$

$$\lambda_j = \sqrt[3]{\max[|j(t)|]\,/j_{limit}(t)} \tag{3-16}$$

Define $t_\lambda$ is the recalled time in the previous trajectory segment, so the new dynamic profiles will become $v(t_\lambda)$,$a(t_\lambda)$ and $j(t_\lambda)$. $t_\lambda$ can be obtained from Eq. (3-18)

$$\lambda = \max(\lambda_v, \lambda_a, \lambda_j) \tag{3-17}$$

$$t_\lambda = \lambda t \tag{3-18}$$

Then we can obtain the new velocity, acceleration and jerk profile in the new scaled time. Figure 3-7 gives the simulation example of velocity profile rescalling.

$$\begin{cases} v(t_\lambda) = v(t)/\lambda \\ a(t_\lambda) = a(t)/\lambda \\ j(t_\lambda) = j(t)/\lambda \end{cases} \tag{3-19}$$

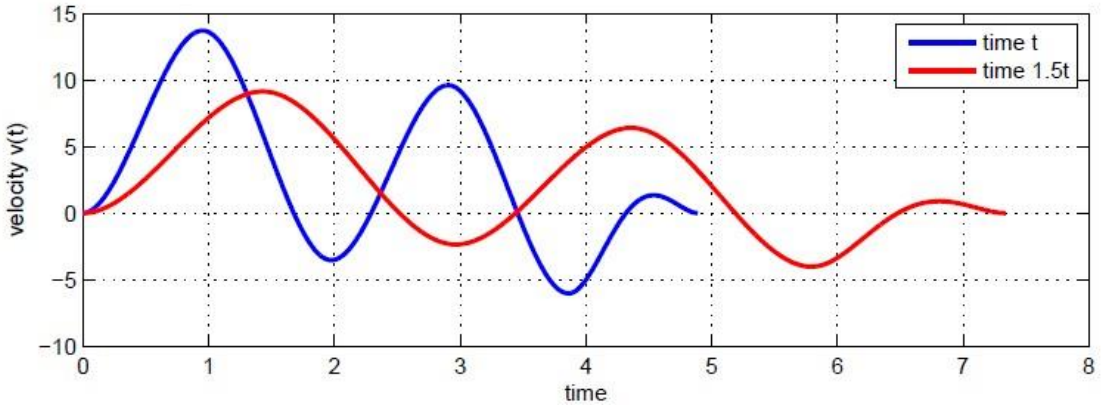



Figure 3-7. Velocity profile with different time scale.

**Time Optimal Trajectory Tracking**

After the path planning using the APF method, the generated path consists of a series of waypoints. By simply connecting these waypoints with straight lines, the robot can avoid collisions with the obstacles along the path. However, there will always be sharp corners on the waypoints. So each time the robot meets these corners, it has to decelerate to zero velocity at these points in order to track the path. Thus the path should have continuous curvature to improve tracking performance and reduce time cost. For the generated continuous curvature trajectory, in practical conditions, the robot should track this trajectory with minimal time or minimal fuel cost. In the following section, a method is introduced for obtaining the time optimal kinematic profile of the robot. Using this method, the robot can finish tracking certain given trajectories with optimal time cost.

As the fractional force between the robot's wheels and the road is limited, so the maximum central acceleration is also limited. Thus when the robot turns along the

trajectory, the maximum turning speed is directly related to the curvature value at the turning point. The curvature along the trajectory should be continuous and not oscillate too much, which will help to save the robot's time and energy of robot when it moves along the trajectory. From the equation for trajectory curvature shown in Eq. (3-22), the minimum polynomial order to interpolate the trajectory should be three. The interpolation order cannot be too high, when the order of interpolation function goes higher, the corresponding curvature will oscillate much more. Marko Lepeti proposed a time optimal trajectory generation method [26], the following time optimal velocity profile generation method is based on Marko Lepeti's work [26].

Cubic spline is used for trajectory interpolation. The X direction coordinate position and Y direction coordinate position are separately interpolated with variable parameter u. Then from this obtained continuous curvature trajectory, the time optimal kinematic profile can be generated for trajectory tracking.

$$v_x = \frac{dX}{du}\frac{du}{dt} \tag{3-20}$$

$$v_y = \frac{dY}{du}\frac{du}{dt} \tag{3-21}$$

When the certain generated trajectory, $\frac{dX}{du}$ and $\frac{dY}{du}$ are determined, the time optimal velocity profile $v_x$ and $v_y$ can be obtained by using time optimal $\frac{du}{dt}$. The curvature of the trajectory can be obtained from the interpolated function by using Eq. (3-22). As there are robot speed limitations from its mechanical parts and working environment, the robot cannot move beyond its highest speed. Also the central acceleration provided by the wheels is also limited. In order to guarantee that the robot does not slide on the road, robot cannot move with a velocity beyond the curvature

limitation of the road. The maximum velocity profile with variable u can be calculated out with Eq. (3-23). The permitted maximum velocity on each point of the road can be obtained in this way.

$$\kappa(u) = \frac{x'(u)y''(u) - y'(u)x''(u)}{(x'(u)^2 + y'(u)^2)^{3/2}} \tag{3-22}$$

$$\sqrt{\frac{a_{max}}{\kappa(u)}} = v_{max}(u) \tag{3-23}$$

Where $a_{max}$ is the maximum central acceleration the robot can provide when tracking the path and $v_{max}(u)$ is the max velocity on the corresponding parameter u to prevent sliding on path.

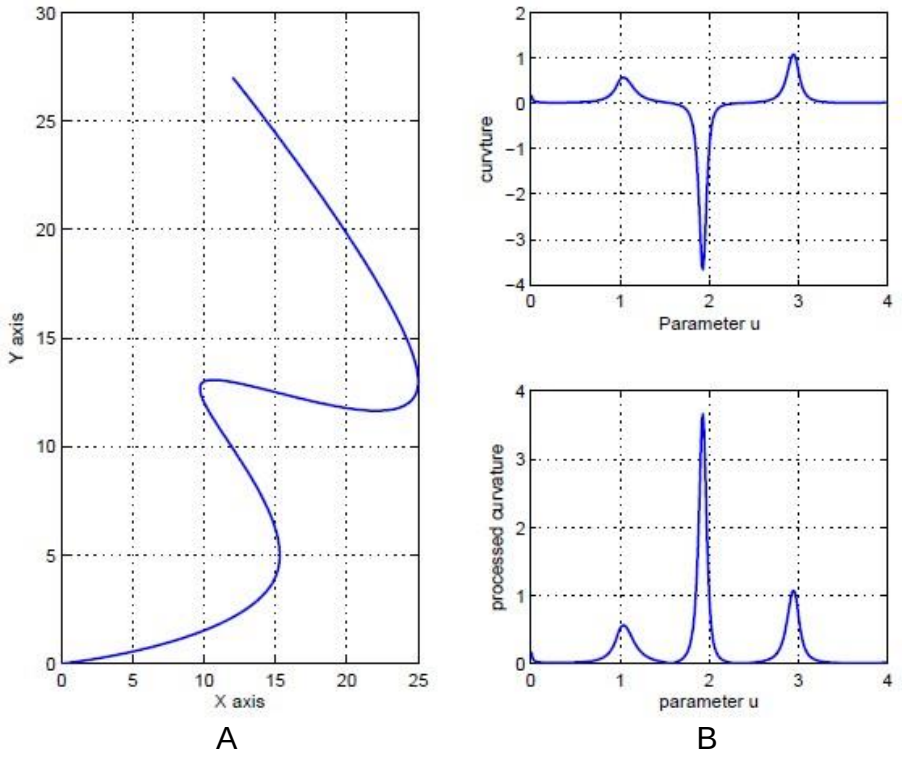



Figure 3-8. Trajectory example to generate the time optimal kinematic profile. A) Interpolated trajectory with variable u, B) The curvature corresponding to the interpolated trajectory.

As the central acceleration provided by the road is constant, so the maximum curvature value that can prevent robot sliding at maximum speed can be calculated. If

the robot will not slide on the highest speed permitted maximum curvature, it also will not slide on the trajectory with a curvature value less than this maximum curvature value. In Figure 3-8 A) the trajectory is presented, and in the top Figure 3-8 B) is the curvature corresponding to this trajectory. The bottom Figure 3-8 B) is the absolute curvature value with consideration of permitted maximum velocity, and the highest speed permitted maximum curvature is drawn instead of the curvature with smaller value.

The relation between trajectory length and the interpolation parameter u can be found through Eq. (3-24). The total cost of time for the trajectory is shown in Eq. (3-25). For certain interpolated trajectory, the distance cannot be changed, so in order to reduce the total time cost in tracking, the permitted highest velocity profile should be generated for the corresponding trajectory.

$$ds = \sqrt{d^2x + d^2y} = \sqrt{x'(u)^2 + y'(u)^2}du \tag{3-24}$$

$$t = \int_{initial}^{final} \frac{ds}{v(s)} = \int_{initial}^{final} \frac{\sqrt{x'(u)^2+y'(u)^2}}{v(u)}du \tag{3-25}$$

The maximum velocity profile related to parameter u can be obtained by using Eq. (3-23). In Figure 3-10 the blue line shows the maximum velocity profile corresponding to curvature. This line indicates the permitted maximum velocity on each point of the road related to parameter u. Since the robot needs to get the maximum velocity profile, it should accelerate with its maximum acceleration until the robot reaches the maximum velocity after which acceleration goes returns to zero. Each time the robot decelerates it should decelerate with maximum deceleration until it reaches the curvature permitted maximum velocity after which the acceleration returns to zero. For each extreme curvature, the corresponding permitted maximum velocity value on this extreme point

can be obtained. It means the velocity satisfies the turning demand at this extreme point. Before this extreme point the robot should decelerate with maximum deceleration and after this extreme point the robot should accelerate with the maximum acceleration. The generated velocity cannot be beyond the curvature permitted maximum velocity. When the velocity goes beyond the limitation it will go horizontally. The velocities before and after this extreme point are obtained with Eq. (3-26) and Eq. (2-27).

$$2a_{max}\Delta s = v_{next}^2 - v_{pre}^2 \tag{3-26}$$

$$\sqrt{2a_{max}\Delta s + v_{pre}^2} = v_{next} \tag{3-27}$$

.Suppose the maximum robot acceleration $a_{max}$ is $2m/s^2$, the maximum central acceleration provide by wheels is $2.5m/s^2$, and the highest speed of the robot is $8m/s$. Then using the method discussed above, the time optimal velocity profile can be generated.

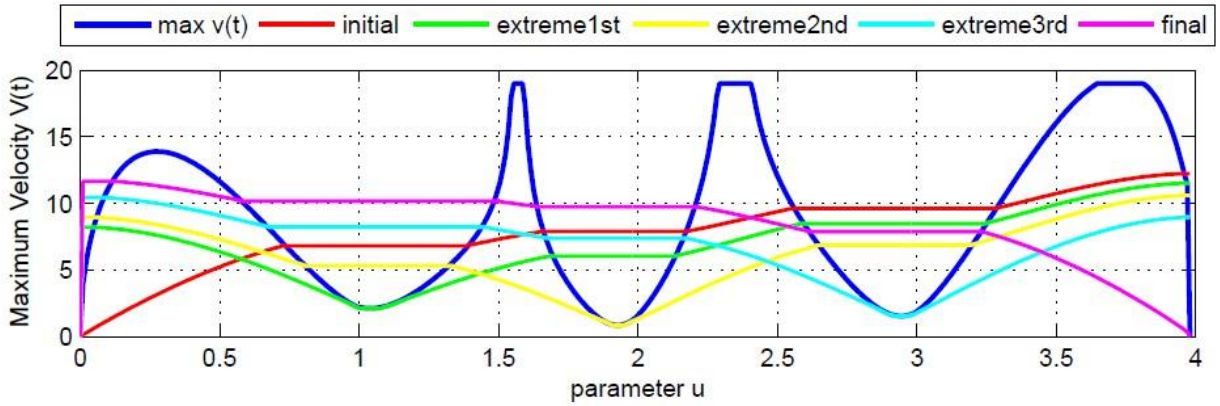



Figure 3-9. The corresponding time optimal velocity profile related with parameter u.

The simulated results are shown in Figure 3-9, The maximum velocity profile is finally generated by choosing the minimum velocity value related to the same parameter u. But this velocity profile is related to the parameter u, by using Eq. (3-24) and Eq. (3-25) the velocity profile related to time will be obtained. The velocity profile related to

time is shown in Figure 3-10. Although this velocity profile is continuous, it is not continuous in acceleration and jerk. This is because the time optimal velocity profile is generated based on the assumption that the robot accelerates with maximum acceleration and decelerates with maximum deceleration. However in reality, the robot cannot change the acceleration value directly from its minimum to maximum or change from the static status to its extreme acceleration status.

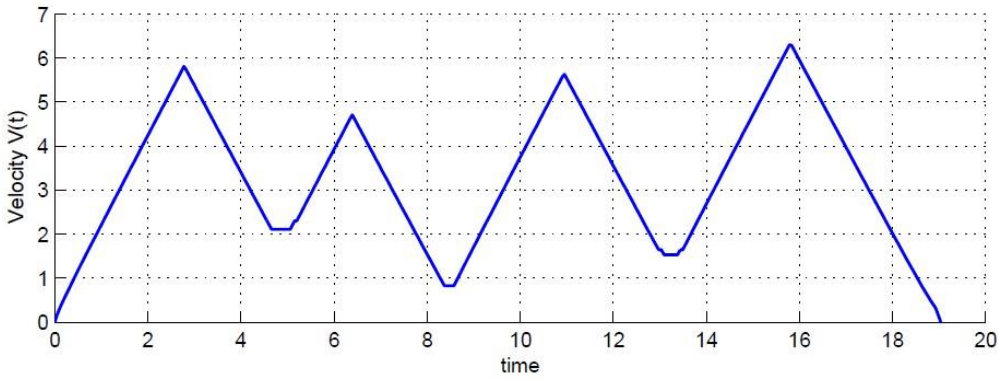



Figure 3-10. The corresponding time optimal velocity profile related with time.

In order to make the acceleration profile become smooth, the acceleration must change in certain algorithm but not change abruptly. So each time the acceleration status changes, the acceleration should gradually change in certain algorithm. Sonja Macfarlane uses the sine wave as the acceleration changing algorithm in her work [27]. The acceleration changing function is written as:

$$a(t) = \frac{a_{max}}{2} \sin\left(\frac{\pi}{dt_{max}} t - \frac{\pi}{2}\right) + \frac{a_{max}}{2} \tag{3-28}$$

Where $a_{max}$ is the robot maximum acceleration and $dt_{max}$ is the cost of time for the robot to change its acceleration from one status to another status. Eq. (3-28) shows the acceleration changing algorithm to connect the discrete acceleration segments. Figure 3-11 shows the behavior of distance, velocity, acceleration and jerk by using the sine

wave changing algorithm. When the robot changes the acceleration with this algorithm the corresponding velocity profile will become smooth. Because the sine wave has an infinite number of derivatives, the corresponding jerk profile is also continuous.

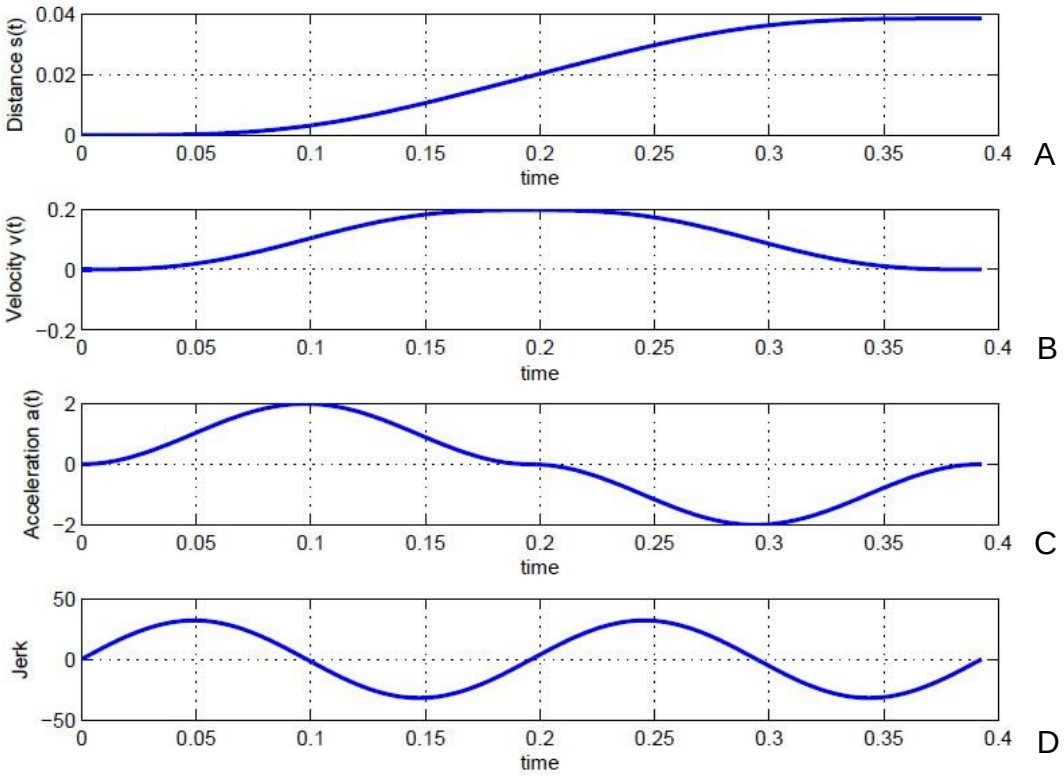



Figure 3-11. The kinematic profile used to connect different kinematic profile segments. A) Distance, B) Velocity profile, C) Acceleration profile, D) Jerk profile.

# CHAPTER 4
# STEREO VISION SYSTEM

In robot navigation, the APF method is used as the collision free navigation algorithm. The application of this algorithm requires the obstacle information and robot location. In this chapter, stereo vision system is used as the environment perception sensor. The stereo vision system is capable of offering the location of obstacles relative to the robot. By using the relative distance between robot and obstacles provided by the stereo vision system, the robot can build an occupancy grid map for collision free path planning. In a grid map, it shows the position of the robot and the instant relative position between the robot and obstacles.

## Depth Perception of Stereo Vision System

The stereo vision system simulates the perception algorithm of human eyes, it consists of two parallel cameras with a certain distance, just like the relative position of human eyes. In depth perception, the two cameras take an image of the same scene. Because of the different location of the two cameras, the image of each camera is different. The pixels stand for the same image element located in different positions on each image and by finding the pixels of one camera and the corresponding pixels of the other camera, the disparity can be generated. The relative distance of obstacles in the workspace can be calculated from this disparity map.

### Pre-Processing of Stereo Vision System

Image pre- processing is the step before stereo matching which includes color adjustment, camera calibration, stereo image rectification and image crop operation [18]. The raw images from the two cameras have differences in contrast and brightness, as well as noises and distortion in both of them. Through pre-processing the stereo

matching precision will be improved. The most important step before stereo matching is the image rectification. The stereo image is rectified by the epipolar constraint line. Epipoplar constraints makes the corresponding pixels of two images lies on the same horizontal line, this line is also called the epipolar line. Through the constraints of the epipolar line, the searching area for stereo matching is reduced. In stereo matching it just needs to search the pixels along the same epipolar line. This highly improves the stereo matching computation speed, thus the real time obstacle detecting robot can move with higher speed. Figure 4-1 shows the example of stereo image rectification and the corresponding epipolar line.


A B

Figure 4-1. Rectification of stereo image A) Left image, B) Right image [21].

**Depth Perception Algorithm**

After the pre-processing of two stereo images, the corresponding pixels lie on the same epipolar line. In order to get the disparity map, the disparity of a single pixel on one image with the corresponding pixel on the other image should be identified. Usually the area based algorithm is used for solving this single pixel stereo matching problem [18]. A block consisting of one middle pixel and its surrounding pixels is taken into consideration and the color values or intensive values of this block are calculated to find

the best matched block area in the other image. The most widely used methods for the calculation method for this correspondence problem are the Sum of Absolute Difference (SAD) and the Sum of Squared Difference (SSD) algorithms [18]. Eq. (4-1) shows the SSD method to find the best matched pixel.

$$SSD_x(x,y) = \sum_j \sum_i [I_l(x+i, y+j) - I_r(x+d_x+i, y+d_y+j)]^2 \qquad (4\text{-}1)$$

Where $I_l$ is the pixel value in the left image, $I_r$ is the pixel value in the right image, $x$ is the pixel coordinate in the X direction, $y$ is the pixel coordinate in the Y direction, $d_x$ is the disparity in the X direction, is the disparity in the Y direction, $i$ is the pixel enumerator in the X direction, $j$ is the pixel enumerator in the Y direction. With the disparity map and using the image forming principle and similar triangles the depth map can be generated. Figure 4-2 shows how to generate depth map from the disparity map, Eq. (4-1) presents also presents the principle.

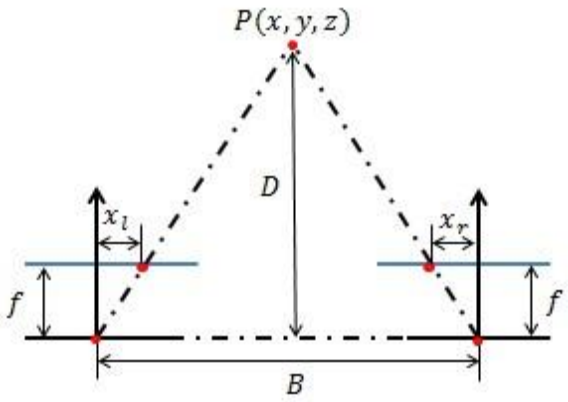



Figure 4-2. Stereo vision system for depth perception.

$$D = f\frac{B}{dx} \qquad (4\text{-}2)$$

Where D is the distance of the point from camera base line, $f$ is the focus length of camera, B is camera offset on the camera base line, and $dx$ is the corresponded pixel disparity of this point.

**Depth Perception Simulation**

Following is an example to simulate the stereo vision system for depth map generation. Table 4-1 provides the camera parameters and object position information in the simulation. The camera and the object monkey are built in 3D animation software Blender to simulate for stereo vision system. In this way the stereo image can be generated from each camera in the model.

Table 4-1. Simulation parameters of stereo vision system.

| Parameter | Value |
|---|---|
| Cameras | Canon 60D |
| Camera Resolution | X:1920 Y:1080 |
| Camera Sensor Size | Width:28.7mm Length:19mm |
| Left Camera Location | [-0.05 -6 0], unit: m |
| Right Camera Location | [-0.05 -6 0], unit: m |
| Object Location | [0 2 0], unit: m |
| Camera Focal Length | 35mm |
| Actual Size of Each Pixel | width=28.7/1920mm height=19/1080mm |

Figure 4-3 presents the model built in Blender, two offset parallel cameras are set as the stereo vision system. Two images are generated from this system and the monkey's location is different in the horizontal axis of the image coordinate.

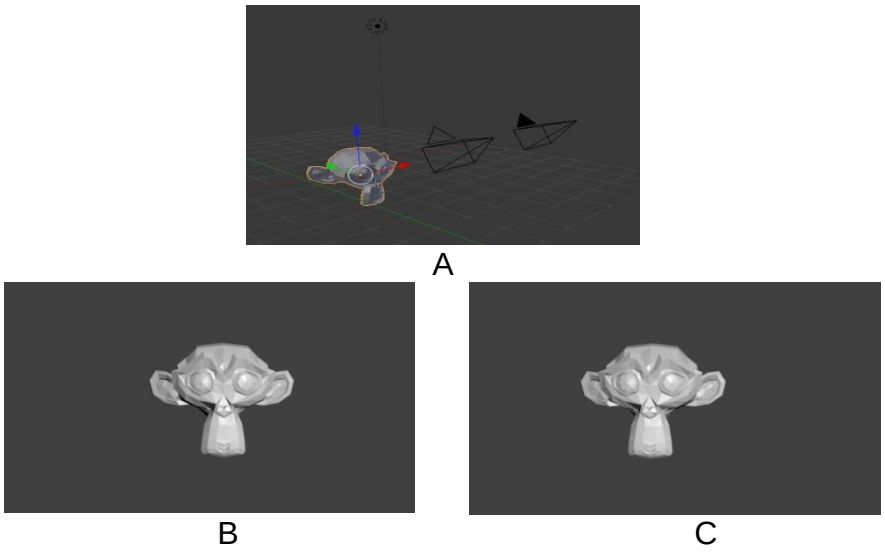



Figure 4-3. The simulation model and the stereo image of the two cameras. A) Object and stereo cameras model, B) Left image, C) Right image.

After pre-processing of the stereo images, the stereo matching is processed. Figure 4-4 shows the generated disparity map and the corresponding depth map. By comparing the results of the depth map with the former defined relative position between camera base line and object, the stereo vision system can provide information about the precise location of obstacles with this method.

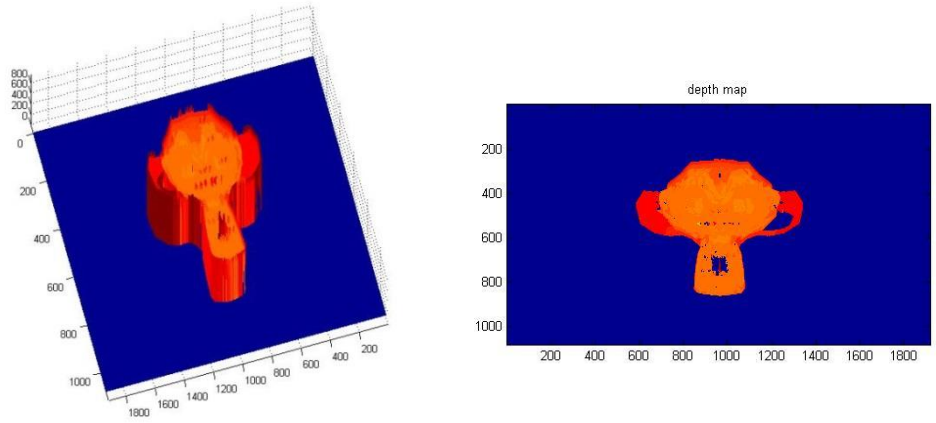



Figure 4-4. Simulation Results. A) The disparity map, B) The depth map.

## Improvement of Vision System

### Obstacle Location Memory with Cells Map

Stereo vision system is used to provide the robot with obstacle information. With the obstacle information the robot can generate the resultant APF force imposed on it. The stereo vision system provides obstacle information before the start of each step. If there are obstacles located in the detecting area, there will be a repulsive potential force imposed on the robot. If there are no obstacles located in the detecting area, the robot is only influenced by the attractive force from the destination. In the simulation presented in Figure 2-6 of Chapter 2, the generated path from the APF method with normal stereo vision system is zigzag shape, it will cost the robot much more fuel and

time when moving with this path. The reason it leads to this kind of zigzag path is that the robot lost the repulsive force generated from the obstacles detected in the previous step when it starts path planning for the next step. This means that the potential force imposed on the robot is calculated only with the obstacles in the detecting area at that instant, but the robot may still be located at the position which is under the influence of previously detected obstacles at that time. So the vision system with memory of the obstacles location is generated to reduce the oscillation of the path. In order to improve the APF method, the map for navigation is divided into grids to represent the workspace of the robot. The location of obstacles is represented by the occupied grids on the map. With this occupied grid map, which is also called the cells map, the location of obstacles is easily be remembered by the vision system. This is because this cells map can be translated into a matrix form, in this way, obstacle location can be easily remembered. Figure 4-5 shows the principle of cells map for location memory. From Figure 4-5 A) to Figure 4-5 B) robot moves one step toward the right angle obstacle.

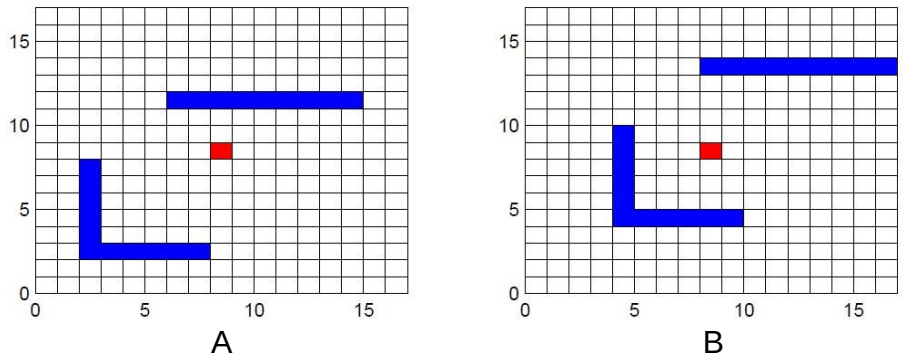



Figure 4-5. Representation of obstacles in cells map when the robot is moving, A) Cells map before the robot is moving, B) Robot moves toward left bottom with one step.

In a cells map, when the parts of an obstacle are located in the area represented by one grid, that grid will be occupied. The location of the robot is also represented in

this way, but the grid that represents the location of the robot is always located in the middle of the map. If the robot does not move out of the area represented by its location grid, the cells map does not change. If it moves beyond this area, the new location will be recalculated and the corresponded change in grid is obtained. The location of the robot is still at the middle of the map and the location of the obstacles change instead. For each movement of the robot, new obstacles move in the map and the pre-detected obstacles moves out. The obstacles move in opposite directions and the same length in the map compared to the movement of robot. Following in Figure 4-6 is the example to compare the performance of vision system in path planning between the normal vision system and the one with location memory. There is always a minimum turning radius limitation when the robot turns along the path, so when the robot plans for the collision free path the turning radius constraint should also be taken into consideration.

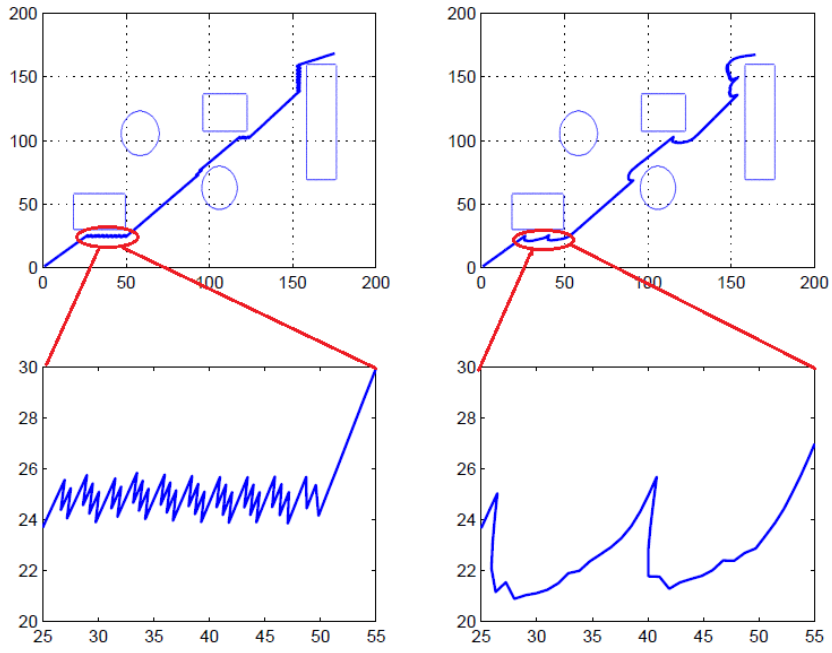



Figure 4-6. Comparing the APF path planning method with two vision systems, A) The path with normal stereo vision system, B) The path with improved stereo vision system

Because the improved stereo vision system can remember the obstacles information around the robot, so long as the robot is located in the influence area of previously detected obstacles, the repulsive force will be always imposed on the robot. In this way the robot gets a smoother path with less oscillation.

**Step Estimation in Path Planning**

In the APF method for path planning, the robot moves step by step. For each step, if the robot moves from static status and ends with static status, the robot will repeatedly accelerate and decelerate which costs much fuel and time. In order to improve the efficiency of path planning, the initial and final condition should be defined for each step including velocity, acceleration and step length. When the vision system detects no obstacles, the robot should keep moving with high speed and long step distance. When the obstacles are close enough, the robot should move with low speed.

The kinematic profile to guide the movement of the robot must be generated for each step based on the circumstances of the robot workspace. Based on the relative distance between obstacles and the robot, the final kinematic value after one step must ensure the robot has enough break distance to avoid collisions. To make the robot move smoothly, the final kinematic value of the pre step is the initial value of the next step. Also the trajectory function for each step should have at least fourth order. With the defined initial and final kinematic value, the trajectory for each step can be generated with Eq. (4-2) and Eq. (4-3) [27]. The trajectory is expressed in a fifth order polynomial and in this way the corresponding jerk profile of the robot is also smooth.

$$x_i(t) = B_{i,1} + B_{i,2}\nabla t + B_{i,3}\nabla t^2 + B_{i,4}\nabla t^3 + B_{i,5}\nabla t^4 + +B_{i,6}\nabla t^5 \quad (4\text{-}3)$$

$$\nabla t = t - t_1 \quad (4\text{-}4)$$

$$\begin{bmatrix} B_{i,1} \\ B_{i,2} \\ B_{i,3} \\ B_{i,4} \\ B_{i,5} \\ B_{i,6} \end{bmatrix} = \left( \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 & t_2^4 & t_2^5 \\ 0 & 1 & 2t_2 & 3t_2^2 & 4t_2^3 & 5t_2^4 \\ 0 & 0 & 2 & 6t_2 & 12t_2^2 & 20t_2^3 \end{bmatrix} \right)^{-1} \begin{bmatrix} p_{initial} \\ v_{initial} \\ a_{initial} \\ p_{\text{final}} \\ v_{\text{final}} \\ a_{\text{final}} \end{bmatrix} \quad (4\text{-}5)$$

Where $t_1$ is the time at the initial point, $t_2$ is the time at the final point, $p_{initial}$, $v_{initial}$ and $a_{initial}$ are the kinematic conditions at the initial point, $p_{\text{final}}$, $v_{\text{final}}$ and $a_{\text{final}}$ are the kinematic conditions at the final point, and $B_{i,\text{j}}$ is the coefficient of the trajectory function.

**Image Processing for Vision System**

The potential field of the workspace is generated based on the information provided by the stereo vision system. So when the obstacle information is changed, the corresponding potential field is also changed. In this way the APF path planning method can be improved by processing the stereo image of the stereo vision system.

**Local minimal avoidance**

When the robot moves in a workspace with complex obstacle contours, especially with the concave configuration, it is easy to generate local minimum in potential filed with the APF method. So when the robot gets trapped in the local minimum, it can go out by changing the shape of the obstacle. As the vision system memorizes the position and contour of the obstacles with cells map, the processing of stereo image is actually the processing of cells map. Figure 4-7 gives the example of local minimum avoidance by using stereo image processing, it shows the procedure of coming out of the local minimum with this method. In Figure 4-7 the obstacle is located in the middle, the blue line stands for the part of obstacle that is not detected by robot, and the red part is the detected part of the obstacle. The robot starts moving from the

initial point, as the obstacle has the concave contour, so the robot is trapped into the local minimum. When the robot is trapped, the vision system connects the endpoint of the concave obstacle part. After image processing, the robot recalculates the potential field with this new obstacle contour. But the robot is still trapped in a new local minimum, so the robot repeats the image processing method once more. In this way the robot finally goes out of the local minimum and reaches the destination. The three images of Figure 4-7 show the procedure of how the robot gets out of the local minimum. By combining the APF method and the stereo vision system, the theoretical defects can be overcome.

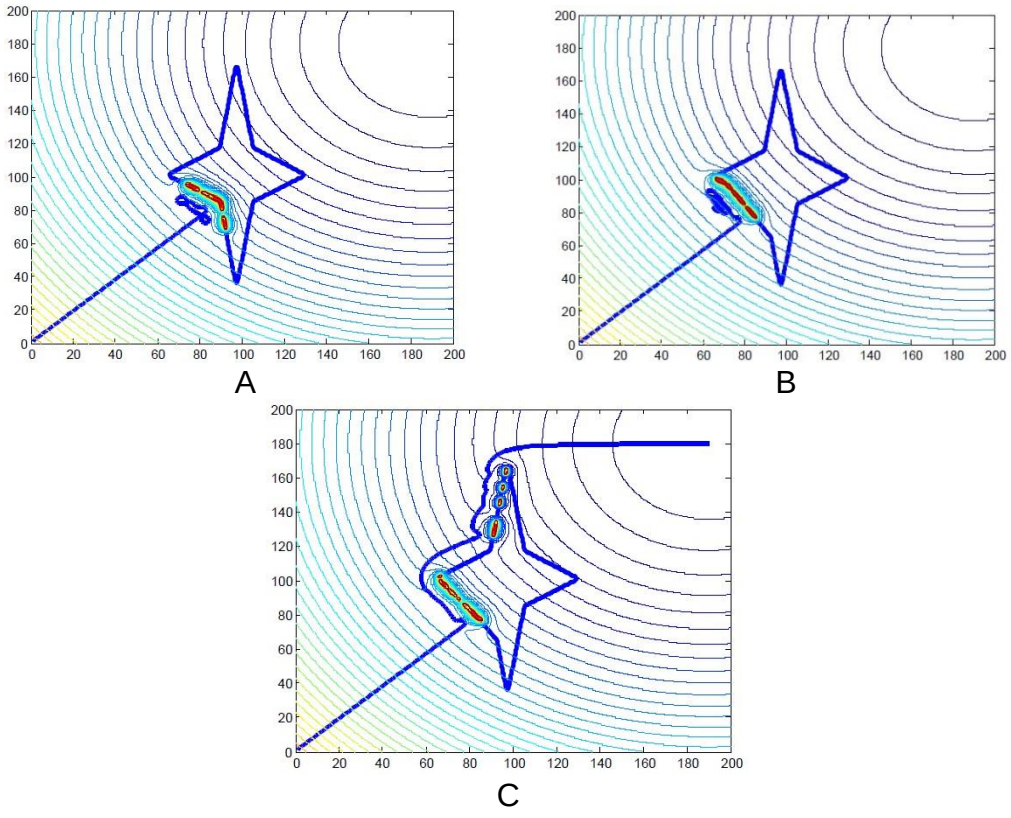



Figure 4-7. The procedure of local minimum avoidance of the APF method by using the stereo image processing. A) Robot is trapped in local minimal, B) Robot processed the detected obstacles using improved stereo vision system, C) Robot moves out from the local minimal.

**Obstacle segments connection**

In a realistic problem, the robot has mass, volume, size in width and height, so it cannot pass the areas beyond constrains of those physical limitations. So in path planning with the APF method, the actual size of the robot and obstacles should be taken into consideration. Through the stereo vision system, if some area between the detected obstacles is smaller than the size of robot, these areas should be occupied in the cells map. Through this method the repulsive force will be generated from these areas to keep the robot away from these areas, so the generated path cannot pass through these limited areas.

Furthermore, the contours of obstacles are usually not smooth in a real workspace. If the robot moves along the obstacle with an undulated contour, the generated path along this obstacle will be oscillated. So the stereo vision system can connect the convex of the detected contour to make the contour become smoother. In this way the oscillation of the generated robot moving path can be reduced.

# CHAPTER 5
# SIMULATION MODEL AND RESULTS

In practical application a self-navigating robot needs to travel back and forth in its workspace to finish tasks. When the robot travels between the base station and the destination, it always traverses an area with obstacles and unpermitted space which the robot must navigate without collision with these obstacles. Also the robot is desired to work in high efficiency. In other words, it should work with minimal cost of time or fuel. In this chapter the simulation model is built to verify the collision free trajectory generation methods which are presented in this thesis.

## Introduction of Simulation Model

When the robot traverses the area with obstacles, it can detect the environment with stereo vision system. The robot then avoids collision with these obstacles using the APF method. In Figure 5-1, it shows a scenario for simulation, the robot is represented by the blue rectangle, and the detecting area is represented by the blue circle sector. The robot under this situation can be used as the mining robot on the moon and may satisfy the requirement of Regolith Excavation Competition sponsored by NASA [6] as the robot gets the mineral from the goal and delivers it to the baseport.

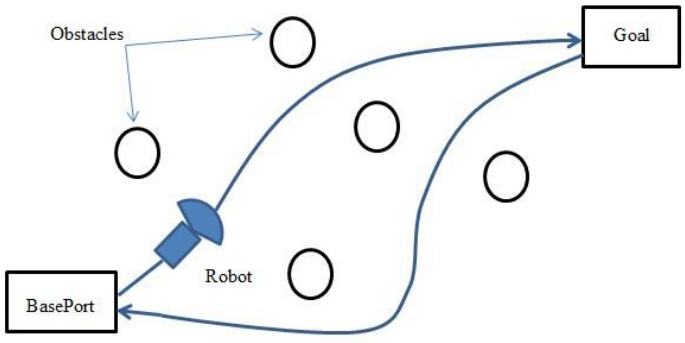



Figure 5-1. Scenario Description of Simulation Model.

The robot transports between the base station and the out port. During the traversing between the base station and out port, the robot must avoid collisions with the obstacles. Before the robot moves in the working area, it only knows the probable location and contours of the obstacles which are represented as some circle obstacles. The robot can detect its surrounding environment with its stereo vision system. Thus when robot moves in the working area it can get the specific configuration of the obstacles. The traveling area in this simulation is shown in Figure 5-2, the black patterns are obstacles, the base port is on the left bottom corner and the goal is on the up right corner.

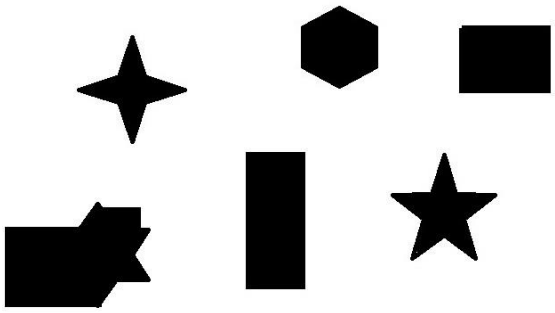


Figure 5-2. Obstacles which are located in travelling area in the simulation model.

Table 5-1. APF method simulation parameters

| Parameter | Value | Unit |
|---|---|---|
| Initial Position $P_o$ | [0 0] | m |
| Finial Position $P_f$ | [180 170] | m |
| Step Length $L$ | 0.1 | m |
| Attractive Coefficient $k$ | 0.05 | N/A |
| Repulsive Coefficient $\eta$ | 5000 | N/A |
| Detection Radius $R_d$ | 10 | m |
| Repulsive Force Influence Radius $R_f$ | 30 | m |

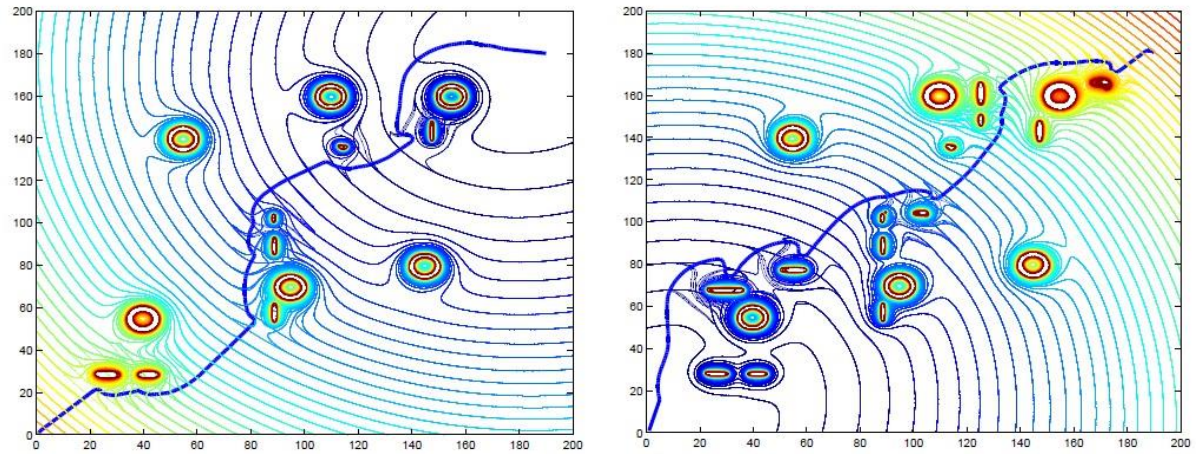
In the simulation, the robot uses the stereo vision system for depth perception when it moves back and forth. With the obstacle information provided by the stereo

vision system, the robot builds the cells map and uses the APF method to generate the potential field for navigation. The simulation parameters are listed in Table 5-1.

### Workspace Detection and Collision Free Path Generation

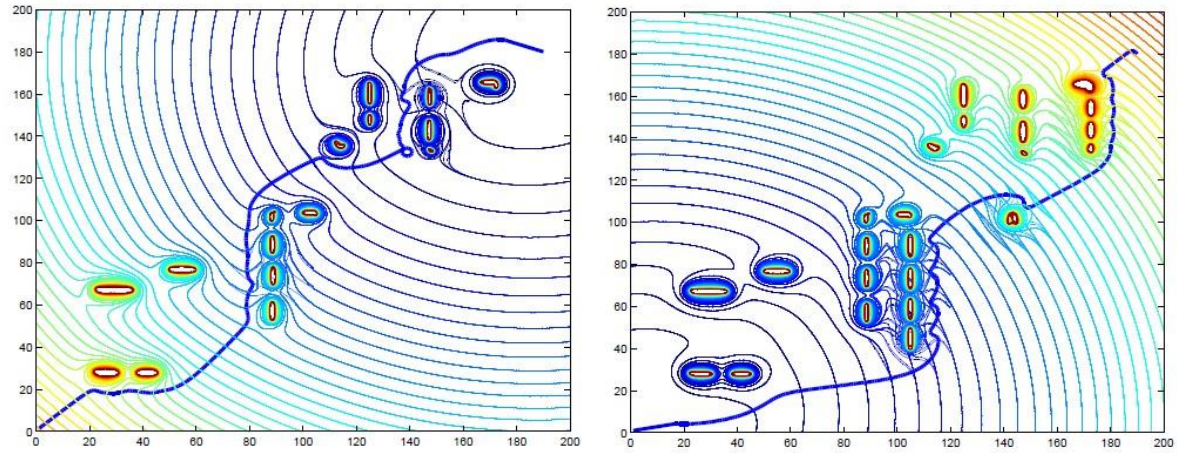
Before the moving starts, the robot knows the probable locations and contours of the obstacles, but it does not know the specific shapes and the locations of these obstacles. During movement, the robot uses the APF algorithm as the guidance law to make sure it can reach the destination and come back without a collision with any obstacles. During the transport between the base station and the out port, the robot detects the configuration of the obstacles encountered by it. But every time the robot can only detect limited parts of the obstacles in the detecting area, after new obstacles are detected the memory of the vision system can memorize the shape and location of these new obstacles. In this way after the robot goes back and forth several times, the location and the shape of the obstacles in travelling area can be gradually improved and the moving path of the robot is gradually refined. After certain travelling times, there will not be new obstacles appearing in the detecting area and the moving of path of the robot becomes constant as the potential field of the travelling area will not be changed. Before the first travelling round trip the robot only knows the probable location and size of obstacles. These obstacles are represented as circles in Figure 5-3 A), while the actual contours of the obstacles are shown in Figure 5-2. In Figure 5-3 A) during the movement, new obstacles were detected and memorized. The blue line is the APF path from the initial point to the goal. In Figure 5-3 B) when the robot arrives at the goal it has to come back to the initial point. At this time the new destination becomes the initial goal and the new potential field is recalculated. The source of the attractive force is changed to the initial point and the robot comes back with the estimated obstacles and the

detected obstacles. After the first round trip, the obstacles deleted along the moving path are stored in the memory of the stereo vision system, so in the next round trip in Figure 5-4 the robot can use this detected obstacle information instead of estimated information. In this way, after the fourth time travelling, there are no new obstacles detected, so the path generated in this potential field becomes constant. Figure 5-3 to Figure 5-6 shows the procedure of gradually refined potential field.



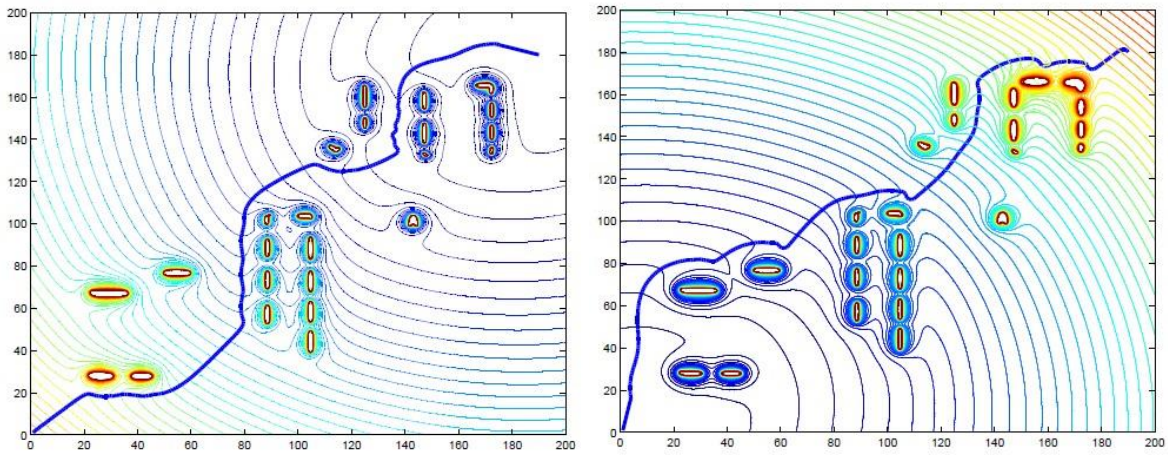
A B

Figure 5-3. Potential field and moving path in the first round trip, A) The path from initial point to out port, B) The back path from out port to initial point.



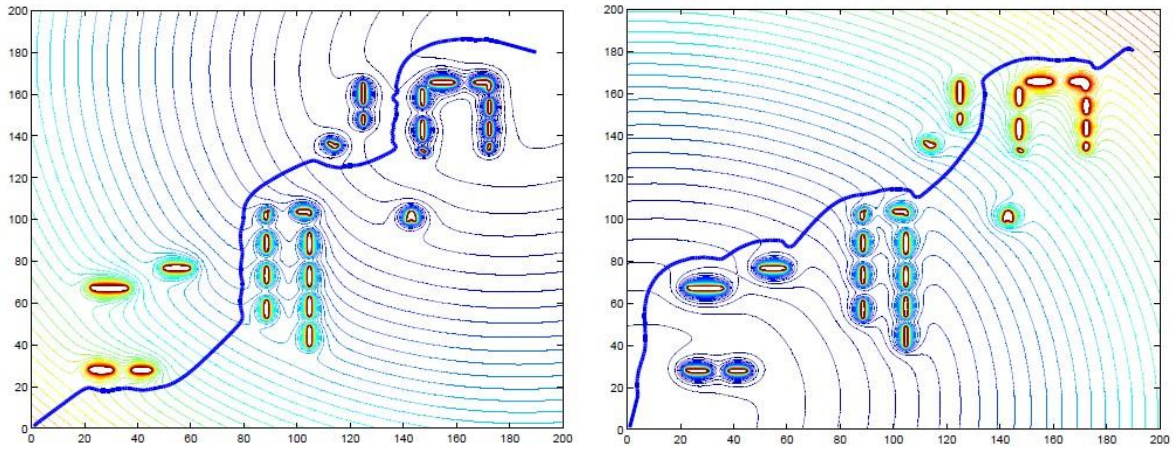
A B

Figure 5-4. Potential field and moving path in the second round trip, A) The path from initial point to out port, B) The back path from out port to initial point.



A B

Figure 5-5. Potential field and moving path in the third round trip, A) The path from initial point to out port, B) The back path from out port to initial point.



A B

Figure 5-6. Potential field and moving path in the fourth round trip, A) The path from initial point to out port, B) The back path from out port to initial point.

In the generation of this collision free path, the robot detects obstacles with the stereo vision system in each step. At the beginning of each step, the robot decides the location of the end of this step based on the detected obstacle information and the kinematic information at the starting point. After a total of four round trips, no new obstacles are detected and the path from APF method becomes constant. So in this way the robot always switches the status of acceleration and deceleration. Also the

generated path oscillates very much and it costs a lot of time and fuel. But as the generated moving path is collision free, the time optimal trajectory can be generated based on this path generated by the APF method.

## Optimal Trajectory Generation

As the collision free path generated by the APF method becomes constant after the fourth travelling trip, the robot can go along this generated path without detecting new obstacles for further tasks. But this generated APF path only provides the end point position of each moving step, so it is hard for the robot to track. Also there are sharp corners along the path, and the oscillation of the path highly reduces the robot efficiency. With the interpolation method introduced in Chapter 3, new trajectory should be generated for the robot to track in further tasks. New trajectory should be easy for tracking and the desired trajectory makes the robot costs minimal time and fuel. The new generated trajectory is constrained by a series of waypoint from the APF path. Under constraint of these waypoints, the new trajectory will not cross with the obstacles. Two trajectory generation methods are used to generate the easy tracking collision free trajectory for the robot. In following content, the simulation results are also provided.

### Spline Interpolation Method

For this spline interpolation method, the position profile in the X axis and the Y axis are interpolated separately with the variable of time. Then the velocity, acceleration and jerk profile in each axis can be generated directly by taking the derivation of the corresponding axis position profile.

$$v_x = \frac{1}{dt}\left(\frac{dX}{dt}\right) \tag{5-1}$$

$$v_y = \frac{1}{dt}\left(\frac{dY}{dt}\right) \tag{5-2}$$

The waypoints used for interpolation are extracted from the APF path with the same distance interval, and the cost of time for each interval is supposed to be 1s. Then the generated trajectory by using cubic spline and fourth order spline is shown in Figure 5-7. Two different interpolation methods are implemented and compared, one is the cubic spline method and the other is the fourth order interpolation method. Both methods are interpolated with time for the trajectory generation. Suppose the maximum robot acceleration $a_{max}$ is $2m/s^2$, the maximum central acceleration provide by wheels is $2.5m/s^2$, and the highest speed of robot is $8m/s$. The cost of time for each interpolated segment is 1s and the corresponding kinematic profile is shown in Figure 5-8. Group A is the kinematic profile of the X axis and group B is the kinematic profile of the Y axis.

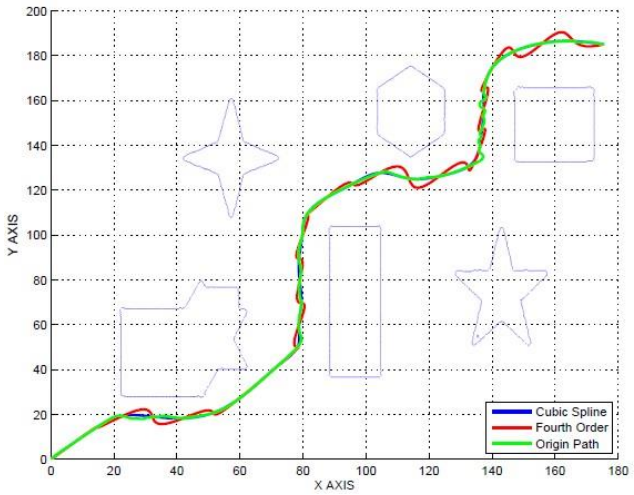



Figure 5-7. Trajectory generated by taking two spline interpolation methods.

As the robot trajectory is interpolated with time in two directions, so the kinematic profile shown in Figure 5-8 describes the robot motion in two directions. The resultant velocity, acceleration and jerk profile can be obtained by using the data in these two directions. Figure 5-9 A) shows the resultant kinematic profile of the cubic interpolation

method, Figure 5-10 A) shows the resultant kinematic profile of the fourth order interpolation method.

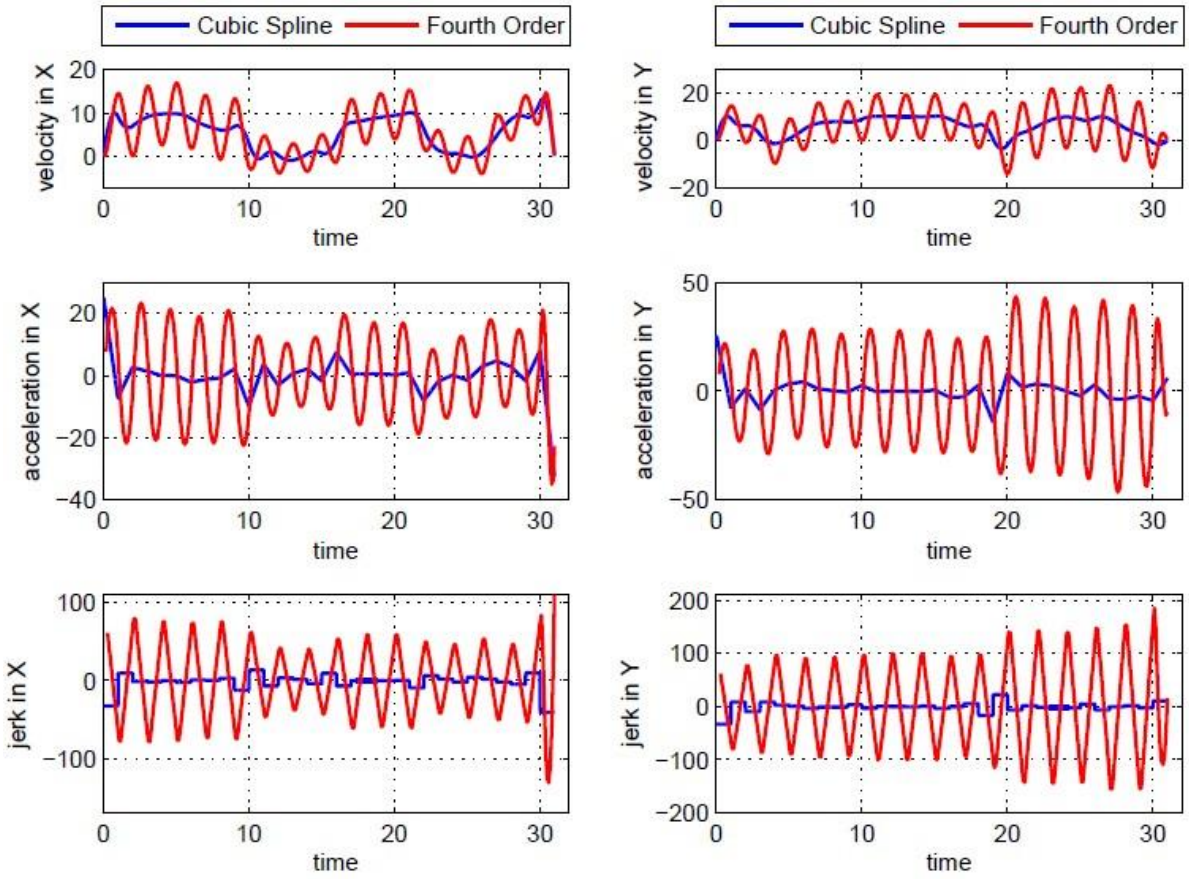



Figure 5-8. Kinematic profile of the two spline interpolation methods, A) The kinematic profile in X direction, B) The kinematic profile in Y direction.

In Figure 5-9 A) and Figure 5-10 A) the kinematic profile is generated by taking the derivative of the interpolated trajectory with time. The maximum kinematic profile values are beyond the robot constraints. This is because the time interval for the robot to move between the two adjacent waypoints is too short. Figure 5-9 B) and Figure 5-10 B) is the kinematic profile after rescaling of the kinematic profile in group Figure 5-9 A) and Figure 5-10 A), After time rescaling, both the velocity and acceleration profile is under the robot physical limitation, but the shape is still keeping the same. It takes

almost 130s using the cubic spline method and takes more than 150s using the fourth order spline method.

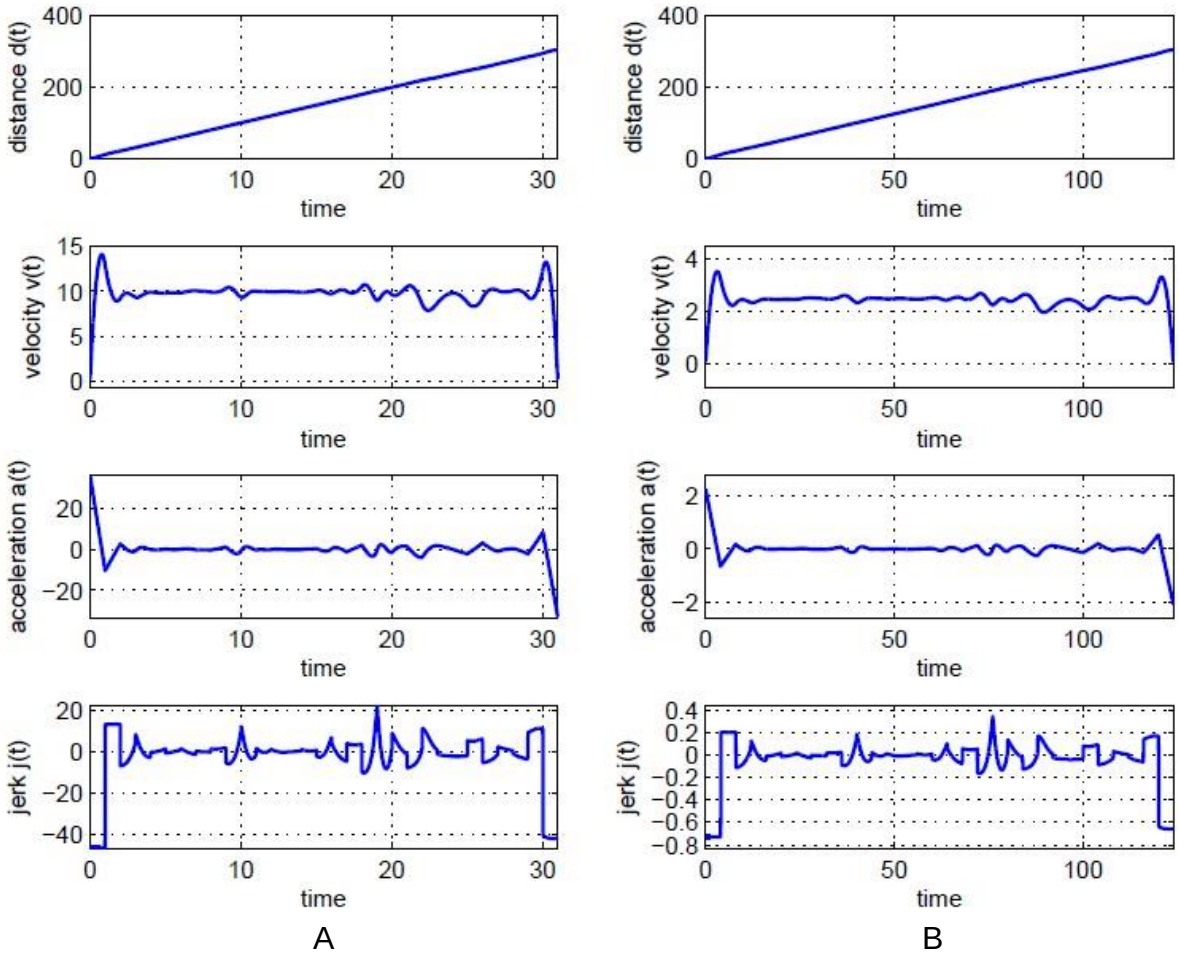



Figure 5-9. Kinematic profile from the cubic interpolation method, A) The original kinematic profile from interpolated trajectory, B) The rescaled kinematic profile.

The kinematic profile generated from the fourth order interpolation method oscillates much more than that from the cubic interpolation method. With the profile generated from the fourth order interpolation method, the robot can reach high velocity values. However, it has to always change the status between acceleration and deceleration and there will be more time and fuel consumption with this method. This is shown in Figure 5-9 and Figure 5-10. The fourth order interpolation method takes

almost 30 seconds more than the cubic interpolation method. The jerk profile obtained from the fourth order interpolation method is continuous but the amplitude is greater than the cubic interpolation method.

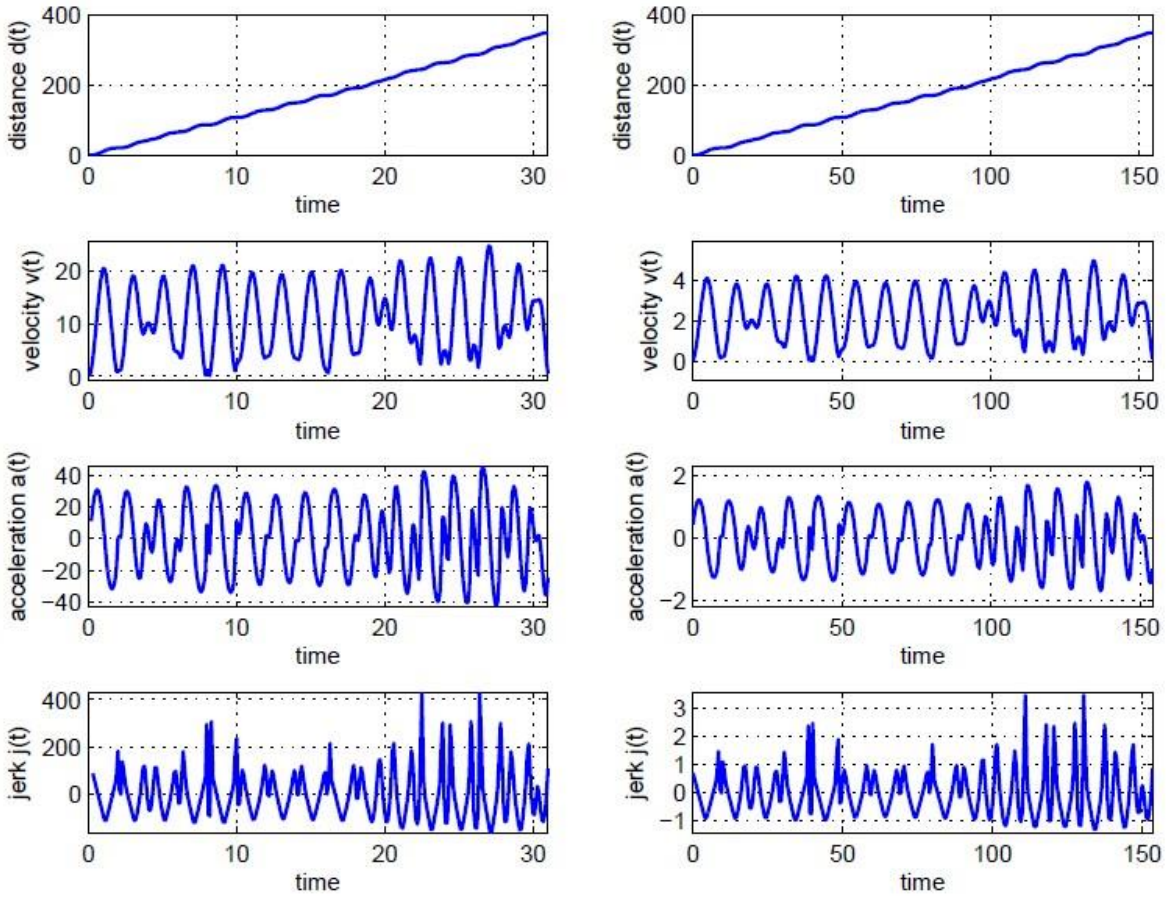



Figure 5-10. Kinematic profile from the fourth order interpolation method, A) The original kinematic profile from interpolated trajectory, B) The rescaled kinematic profile.

**Time Optimal Trajectory Generation Method**

It is convenient to generate the kinematic profile by using the spline interpolation method, but this method cannot ensure that the robot track the generated trajectory with minimum cost of time. In most practical problems the robot is desired to track the constant trajectory within the least amount of time, so the time optimal trajectory

generation method is used. Cubic spline is used for trajectory interpolation. The X direction coordinate position and the Y direction coordinate position are separately interpolated with variable parameter u. The generated trajectory is shown in Figure 5-11, this trajectory is the as same as the trajectory from the cubic interpolation spline with variable of time.

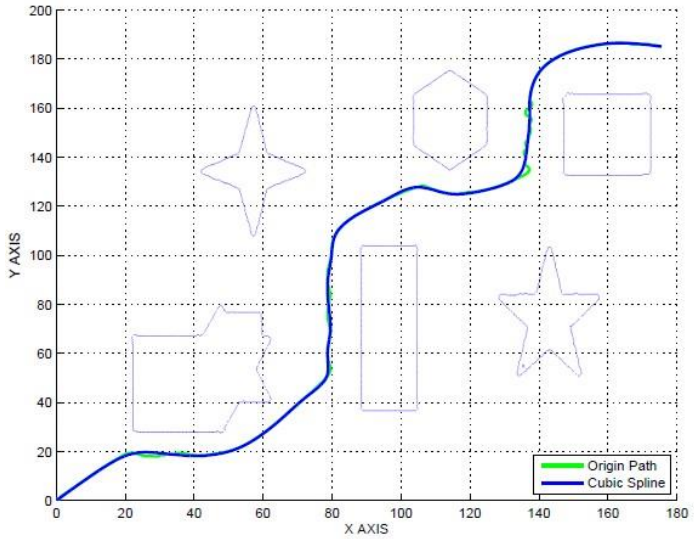



Figure 5-11. The trajectory used for time optimal trajectory generation method.

The kinematic profile corresponding to the time optimal trajectory is generated from this interpolated trajectory with variable u. Because the trajectory for tracking is chosen, $\frac{dX}{du}$ and $\frac{dY}{du}$ cannot be changed. The proper $\frac{du}{dt}$ should be chosen for the time optimal kinematic profile.

$$v_x = \frac{dX}{du}\frac{du}{dt} \tag{5-3}$$

$$v_y = \frac{dY}{du}\frac{du}{dt} \tag{5-4}$$

The curvature of the trajectory can be obtained from the trajectory interpolation function. Figure 5-12 shows the generated curvature. The red line is the curvature generated directly from the trajectory, and the blue line takes the maximum velocity

limitation into consideration. Because the robot maximum velocity is $8m/s$ and the maximum central acceleration provided by road is $2.5m/s^2$, even the robot moves at the highest speed. If the curvature of the trajectory is smaller than $0.04m^{-1}$, the robot will not slide. To get the maximum velocity profile permitted by the curvature of the trajectory, only the curvature greater than $0.04m^{-1}$ will be taken into consideration.

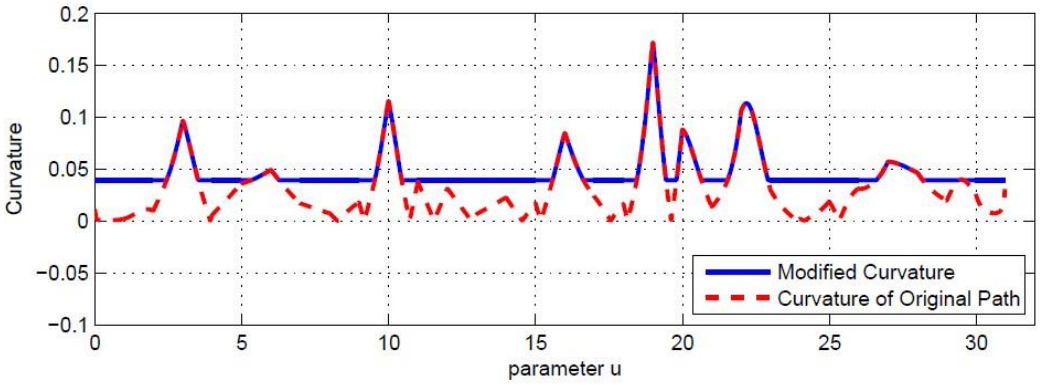



Figure 5-12. The curvature of the interpolated trajectory.

The maximum velocity profile can be generated based on the modified curvature of the interpolated trajectory. In Figure 5-13 the red line is the permitted maximum velocity on each position of the trajectory by the corresponding curvature at that position. The robot starts from static status at the initial point and keeps static when it arrives at the final point. If the robot tracks the generated trajectory with minimum cost of time it must accelerate with maximum acceleration and decelerate with maximum deceleration. The robot velocity is also limited by the extreme value of the curvature. When the robot moves on the point with the extreme curvature value, the velocity at this point must ensure the robot will not slide on the road. So before the robot arrives at this point it must decelerate and after this point the robot can accelerate in order to save time. In Figure 5-13 the different color lines, except the initial and final ones are the maximum velocity profile under different extreme curvature constraints. Before and after the

extreme curvature point, the robot accelerates and decelerates with maximum acceleration, when the speed goes beyond the limit of curvature it will become horizontal. There will be relatively smaller maximum velocity profile corresponding to this horizontal part.

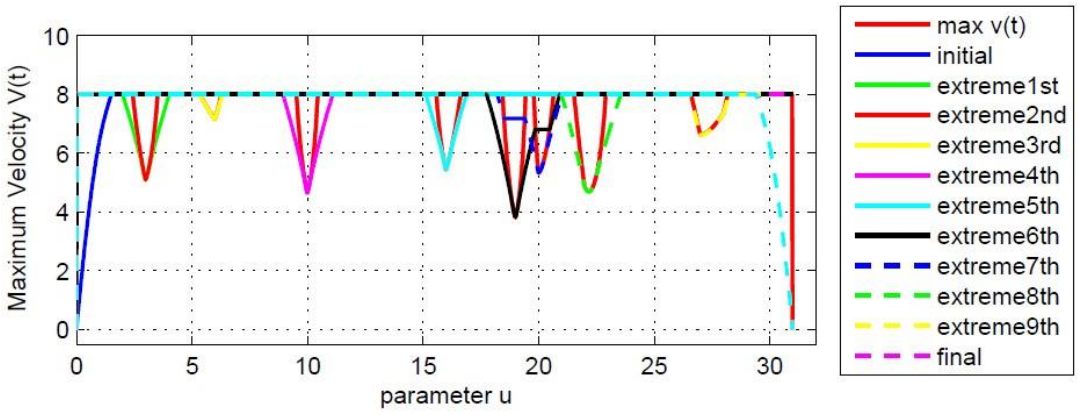



Figure 5-13. The maximum velocity profile with variable of parameter u.

Finally, the maximum velocity profile is obtained by choosing the minimum velocity value corresponding to a different parameter u. Then the velocity profile with variable of time will be obtained by taking the method in Chapter 3. The maximum velocity profile with variable of time is shown in Figure 5-14.

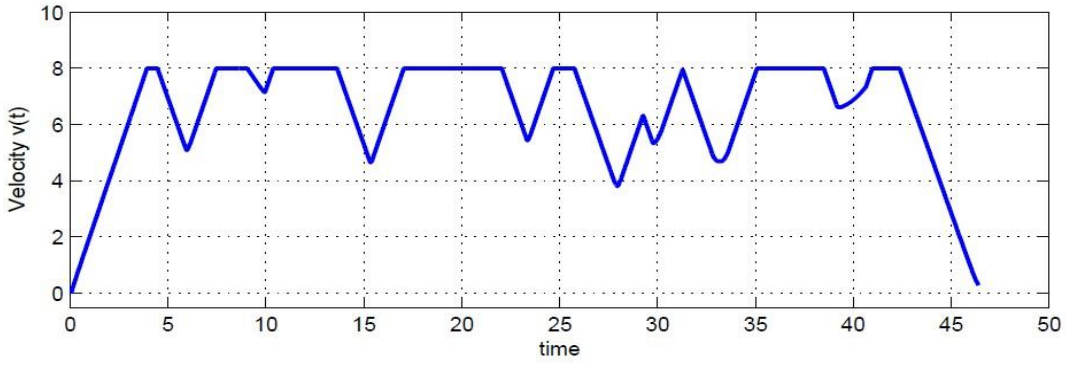



Figure 5-14. The maximum velocity profile with variable of time.

This generated maximum velocity limitation is based on the maximum acceleration and deceleration, but in real conditions acceleration cannot change sharply.

Suppose the acceleration is changed with the regulation presented in Chapter 3. The smoother velocity profile and the corresponding acceleration and jerk profiles are shown in Figure 5-15. With this method the robot starts acceleration from the initial point with maximum acceleration, every time the robot reaches the highest speed the acceleration will go to zero, then the robot will keep moving at the highest speed. Before reaching the position with limited curvature the robot decelerates with the previously mentioned maximum deceleration. Every time the robot changes the acceleration status it will cost little time, so the jerk changing magnitude is bigger than the interpolation method.

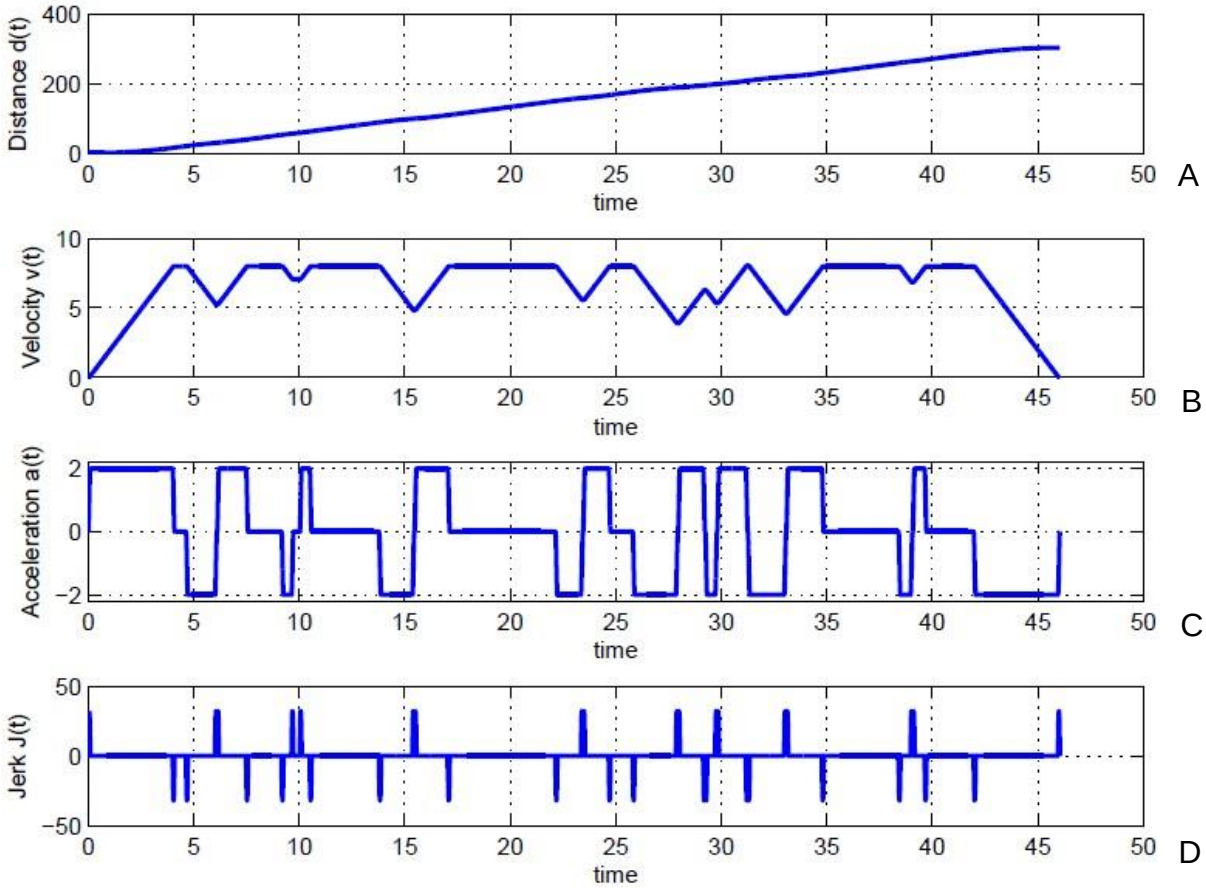



Figure 5-15. The time optimal kinematic profile of the interpolated trajectory. A) Distance, B) Velocity profile, C) Acceleration profile, D) Jerk profile.

## Simulation Results Analysis

By comparing the two methods for optimal trajectory generation, the spline interpolation method is more convenient to generate the corresponding kinematic profile. Because the trajectory is interpolated with variable of time in this method, the kinematic profile can be directly generated by taking the derivation of the trajectory with time. But this method cannot provide the time optimal kinematic profile for the generated trajectory. The time optimal trajectory generation method generates the kinematic profile by considering the limitation of curvature, the robot moves at the highest speed anytime under constraint of maximum velocity. But in this method, the work to make the velocity become smoother is difficult and needs more calculation.

The time optimal method takes a total of 46s for robot to get to the destination, the cubic interpolation methods takes 120s and the fourth order interpolation method takes 154s. The velocity profile from time optimal method can stay on the highest speed for most of the time. The velocity profile from cubic spline method can also stay in certain value for most of the time, but due to the limitation of acceleration, the rescaled velocity can only reach $4m/s$ as the highest speed. Because the kinematic profile from the fourth order interpolation method oscillates greatly, the robot always accelerates and decelerates, so this method costs the most of time.

The jerk profile from the time optimal method and the fourth order interpolation method are continuous, and the jerk profile from the cubic interpolation method is discrete. But the continuous jerk profiles have greater amplitude compared to the jerk profile from the cubic interpolation method. Especially in the time optimal method, the jerk can go up to $30m/s^3$ this will lead to the severe change of robot's movement. The

jerk profile from the cubic interpolation method is discrete, and this will influence the trajectory tracking precision.

# CHAPTER 6
# CONCLUSION AND FUTURE RESEARCH

This thesis discussed the path planning method for self-navigating robots in an unknown environment. The APF method is used to avoid the obstacles and the sensor for environment perception is the stereo vision system. By studying the depth perception algorithm of the stereo vision system and the principle of the APF method, an improved stereo vision system is presented in the thesis to solve the local minimum problem and to reduce the oscillation of the generated collision free path. The grid map is proposed for robot navigation. By combining the grid map and the depth map, an imaging processing method can be used to optimize the path generated from the normal APF method. The path from the APF method is connected by a series of straight line segments, this is hard for a robot to track and it will cost much time and energy. The corresponding trajectory generation method is proposed in this thesis to approximate the APF path. Two interpolation methods are compared and discussed to find the best method for trajectory interpolation. Also, the method to generate time optimal trajectory and the corresponding kinematic profile is discussed. Finally, in the constructed scenario all of these new methods are evaluated by simulation. The simulation results prove that the new stereo vision system and the time optimal trajectory are the most valid and efficient.

The method provided in this thesis is suitable to the environment with static obstacles. The APF method can be used to avoid moving obstacles, but if it is used directly the performance will not be very good. Further work should be done in the further to improve the APF method for avoiding moving obstacle. Also, the trajectory generation method is based on the constant APF path; if the robot encounters moving

obstacles the original APF must be changed. In future research, the collision free trajectory generation method with moving obstacles should be discussed. Furthermore, in this thesis, the time optimal trajectory and corresponding time optimal kinematic profile generation method is used to optimize the constant APF path, so before the APF path becomes constant each moving step of the robot is not time optimal. More work should be done in the future to optimize the each step of the robot when using the APF method.

LIST OF REFERENCES


[1] (2013, February) Adept Mobile Products. Adept. [Online]. Available: http://www.adept.com/products/mobile-robots/mobile-platforms/lynx/general

[2] (2013, February) Adept Mobile Products. Adept. [Online]. Available: http://www.adept.com/products/mobile-robots/mobile-transporters/spc-4200/general

[3] (2013, February) TurtleBot. Willow Garage. [Online]. Available: http://www.willowgarage.com/turtlebot

[4] (2013, February) Bluefin-12D. Bluefin Robotics. [Online]. Available: http://www.bluefinrobotics.com/products/bluefin-12d/

[5] (2013, February) Ultra Heavy-Duty Work-Class ROV System. FMC Technologies. [Online]. Available: http://www.schilling.com/products/ROVs/Pages/UHD.aspx

[6] (2013, February) NASA's Fourth Annual Lunabotics Mining Competition. NASA. [Online]. Available: http://www.nasa.gov/offices/education/centers/kennedy/technology/lunabotics.html#Rules

[7] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," IEEE International Conference on Robotics and Automation, Vol.2, pp.500–505, 1985.

[8] Jin-Oh Kim and Pradeep K. Khosla, "Real-Time Obstacle Avoidance Using Harmonic Potential Functions," IEEE Transactions on Robotics and Automation, Vol.8, No.3, pp.338–349, 1992.

[9] M. H. Mabrouk and C. R. McInnes, "Solving the Potential Field Local Minimum Problem Using Internal Agent States," Robotics and Autonomous Systems, Vol.56, No.12, pp.1050–1060, 2008.

[10] Min gyu Park, Jae hyun Jeon and Min cheol Lee, "Obstacle Avoidance for Mobile Robots Using Artificial Potential Field Approach with Simulated Annealing," IEEE International Symposium on Industrial Electronics,Vol.3, pp.1530-1535, 2001.

[11] S.S. GE and Y.J. CUI, "Dynamic Motion Planning for Mobile Robots Using Potential Field Method," Autonomous Robots, Vol.13, pp.207-222, 2002.

[12] Josue David Munoz, "Rapid Path-Planning Algorithm for Autonomous Proximity Operations of Satellites," PHD dissertation, University of Florida, USA, 2011.

[13] Paraskevas Dunias, "Autonomous Robots Using Artificial Potential Fields," PHD dissertation, Eindhoven University of Technology, Netherlands, 1996.

[14] Nicholas S. Martinson, “Obstacle Avoidance Guidance and Control for Autonomous Satellites,” PHD dissertation, University of Florida, USA, 2009.

[15] Andrew R Tatsch, “Artificial Potential Function Guidance for Autonomous In-Space Operations,” PHD dissertation, University of Florida, USA, 2006.

[16] Allan Eisenman, Carl Christian Liebe, Mark W. Maimone, Mark A. Schwochert and Reg G. Willson, “Mars Exploration Rover Engineering Cameras,” Jet Propulsion Laboratory, 2003.

[17] Steven B. Goldberg, “Stereo Vision and Rover Navigation Software for Planetary Exploration,” IEEE Aerospace Conference Proceedings, Vol.5, No.5, pp.2025-2036, 2002.

[18] Ali Kilic, “Navigation of Mobile Robot Using Stereo Vision,” Master thesis, University of Gaziantep, Turkey, 2010.

[19] Don Murray and Jim Little, “Using Real-time Stereo Vision for Mobile Robot Navigation,” Autonomous Robots, Vol.8, No.2, pp.161-171, 2000.

[20] Don Murray and Cullen James, “Stereo Vision Based Mapping and Navigation for Mobile Robots,” IEEE International Conference on Robotics and Automation, Vol.2, pp.1694-1699, 1997.

[21] (2013, February) Stereo Rectification. Fit.com. [Online]. Available: http://fit.com.ru/Projects/stereo_rectification.htm

[22] Maryum F. Ahmed, “Development of a Stereo Vision System for Outdoor Mobile Robots,” Master thesis, University of Florida, USA, 2006.

[23] M. Hebert, “Pixel-Based Range Processing for Autonomous Driving,” IEEE International Conference on Robotics and Automation, Vol.4, pp.3362-2267, 1994.

[24] V. Munoz, A. Ollero, M. Prado and A. Simon, “Mobile Robot Trajectory Planning with Dynamic and Kinematic Constraints,” IEEE International Conference on Robotics and Automation, Vol.4, pp.2802-2807, 1994.

[25] K. Petrinec and Z. Kovacic, “Trajectory Planning Algorithm Based on the Continuity of Jerk,” IEEE Mediterranean Conference on Control & Automation, pp.1-5. 2007.

[26] Marko Lepetic, Gregor Klancar, Igor Skrjanc, Drago Matko and Bostjan Potocnik “Time Optimal Path Planning Considering Acceleration Limits,” Robotics and Autonomous Systems, Vol.45, pp.199-210, 2003.

[27] Sonja Macfarlane, “On-Line Smooth Trajectory Planning for Manipulators,” Master thesis, The University of British Columbia, Canada, 2001.

[28] Imran Waheed, “Trajectory / Temporal Planning of a Wheeled Mobile Robot,” Master thesis, University of Saskatchewan, Canada, 2006.

# BIOGRAPHICAL SKETCH

Ronghua Yao was born in Youyu, Shanxi, China, in 1988. In September 2007 he was accepted into Beijing Institute of Technology's Department of Mechanical and Vehicle Engineering to purse his bachelor's degree in mechanical engineering. There he was honored excellent graduated student after graduation.

After completing his bachelor's degree in Jun 2011, he was accepted into the master's program in mechanical engineering at the University of Florida. While in his master's program his study and research focus on dynamic, control and system. His interest is focus on robotics, embedded system and mechatronics.