Accuracy Improvement of Industrial Robot Using PID Controller Based on Back-propagation Neural Networks

Jianyu Tang

A Thesis

in

The Department

of

Mechanical, Industrial & Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

August 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Jianyu Tang**

Entitled: **Accuracy Improvement of Industrial Robot Using PID Controller Based on Back-propagation Neural Networks**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Brandon W. Gordon

_____ External Examiner
Dr. Chunyan Lai

_____ Examiner
Dr. Brandon W. Gordon

_____ Supervisor
Dr. Wen-Fang Xie

Approved by _____
Martin D. Pugh, Chair
Department of Mechanical, Industrial & Aerospace Engineering

_____ 2022

_____
Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Accuracy Improvement of Industrial Robot Using PID Controller Based on Back-propagation Neural Networks

Jianyu Tang

Nowadays, industrial robots are widely used in many fields. With the wide range of applications, the accuracy of robots has become an issue of concern. In some path following tasks, the accuracy of existing robots does not yet meet the industry standard. In this research, the accuracy of the robot is enhanced by using position-based visual servoing.

The purpose of this research is to propose a novel dynamic path tracking (DPT) method to solve the aforementioned accuracy problem. This method uses the C-Track from Creaform to measure the end-effector of the robot M-20iA from Fanuc and to use the visual information to guide the robot to follow the desired path.

First, a stereo binocular camera C-Track provides the real-time pose information of the end-effector. To remove the noise transmitted from the camera, the research uses a robust Kalman filter (RKF) to improve the performance of the standard Kalman filter under disturbance by correcting the state variable error covariance (P).

Then, the Fanuc robot M-20iA uses position-based visual servoing (PBVS) strategy to correct the position and orientation of the end-effector, which is implemented through dynamic path modification (DPM) function, in conjunction with real-time data acquired by C-Track.

Next, adaptive neuro-PID (ANPID) control is developed as the PBVS scheme for DPM correction. Such control strategy has a strong adaptive and self-learning capability, which enables online tuning of the PID controller parameters, resulting in better performance in robot control.

Finally, extensive experiment tests have been carried out and the results show that the the accuracy of path following reaches ±0.08mm and ±0.04deg, compared with the accuracy ±0.2mm and ±0.1deg achieved by conventional PID controller [1].

Acknowledgments

Three-year master's career is coming to an end. Every step I have taken in these three years could not have been achieved without the support and encouragement of my supervisors, friends and family.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Wen-Fang Xie for her continuous support, invaluable advice, and tutelage during my postgraduate study. Her profound knowledge and rigorous work attitude inspire me in all time of my academic research and daily life.

I would also like to offer my special thanks to Mr. Ehsan Zakeri. He is very helpful in every step of my research and taught me a lot of knowledge and skills. I sincerely wish him all the best in his future research.

In addition, I would like to thank my partner Mr. Tao Zhou. In the process of research, we overcame difficulties together. It is his kind help that have made my study and life in the Canada a wonderful time. In daily life, I would also like to appreciate my roommate Mr. Xingyu Shen. He also gives a lot of care and advice, and I wish him well during his PhD career. Heartly thanks also go to my colleagues and friends who accompanied me along these years: Dr. Henghua Shen, Mrs. Tingting Shu, Mr. Ronghua Zhang, Mr. Long Chen, Mr. Ningyu Zhu, Mr. Runze Wang, Mr. Haibo Feng, Mr. Linghao Meng, Mr. Hao Zhang and Miss. Tianyue Zhong.

Last but not least, I am deeply grateful to my parents for their unwavering support and belief in me. Without their tremendous understanding and encouragement in the past few years, I would not be where I am today.

To my beloved parents

Bingtao Tang** and **Li Zhou

Contents

List of Figures

List of Tables

Nomenclature

AKF Adaptive Kalman filter

ANN Artificial Neural Network

ANPID Adaptive neuro-PID

AOKF Adaptive optimal Kalman filter

ART Adaptive resonance theory

BP Backpropagation

CNN Convolutional Neural Network

DOF Degrees of freedom

DPM Dynamic path modification

DPT Dynamic Path Tracking

EIH Eye-in-hand

EKF Extended Kalman filter

ETH Eye-to-hand

FNN Fuzzy neural network

HNN Hopfield Neural Network

HVS Hybrid visual servoing

IBVS Image based visual servoing

KF Kalman filter

MLP Multi-layer perceptron

MSE Mean square error

PBVS Position based visual servoing

PDP Parallel distributed processing

PID Proportional-integral-derivative

RBF Radial basis function

RKF Robust Kalman filter

RMS Root mean square

SD Standard deviation

SKF Standard Kalman filter

TP Teach pendant

UKF Unscented Kalman filter

UT Unscented transformation

Chapter 1

Introduction

In recent years, the tendency towards the employment of industrial robots for high precision manufacturing tasks, such as welding, cutting, drilling, riveting, and milling, has significantly increased [2]. These tasks are characterized by high repeatability with high accuracy, so it is very appropriate to employ industrial robots to perform such tasks. However, the current absolute position accuracy of most industrial robots is around 1mm [3], which cannot meet the industry standard process specifications in aerospace industry, i.e., ±0.2mm [4].

One choice for improving the accuracy of industrial robots is calibration [5]. Robot calibration is a process of identifying the true geometric parameters in the kinematic structure of industrial robots. Currently, there is a proliferation of calibration techniques that can be applied to calibrate the geometric models of industrial robots using different modeling, and identification approaches [6–8]. However, in these techniques, the effect of the motion on the robot is not taken into accent. To cope with this issue, dynamic calibration by using visual servoing technique has been suggested [9, 10]. The visual servoing technique is referred to as the control strategy that uses real time visual information to control the robot manipulators.

This thesis aims mainly to develop a novel visual servoing technique, named as dynamic path tracking (DPT) method for industrial robots based on photogrammetry sensors and adaptive neuro-PID (ANPID) method. To implement the above mentioned method requires the preparation of a controller programmed in Microsoft Visual studio with C# .net language, a Creaform's C-track for receiving and transmitting pose information, and a FANUC robot for testing the feasibility of the

algorithm.

1.1 Industrial Robots

An industrial robot is a machine device that can be programmed to perform tasks automatically, replacing people in hazardous environments to perform a variety of complex tasks. Depending on the mechanical structure, industrial robots can be divided into the following five categories: Cartesian, SCARA, Cylindrical, Parallel and Articulated.

Cartesian robots, which can also be called linear robots, can operate on three orthogonal axes (X, Y, Z). They are widely used in 3D printing and CNC machine tools because of its high repeatability, and positioning accuracy [11]. Selective Compliance Assembly Robot Arm (SCARA) robot, a special type of industrial robot with cylindrical coordinate type, has 3 rotary joints with axes parallel to each other for positioning and orientation in the plane. Another joint is a moving joint, which is used to complete the movement of the end effector in the direction perpendicular to the plane. Compared to the Cartesian robot, SCARA robot has a circular working range, so it is more flexible.

Cylindrical robot is usually applied to spot welding, casting, molding, and handling. It has at least two joints, a rotating joint at the base and a prismatic joint that can be moved [12]. It has a larger working envelop than the Cartesian robot but requires a more sophisticated control system. The above three robots are shown in Figure 1.1 [12].

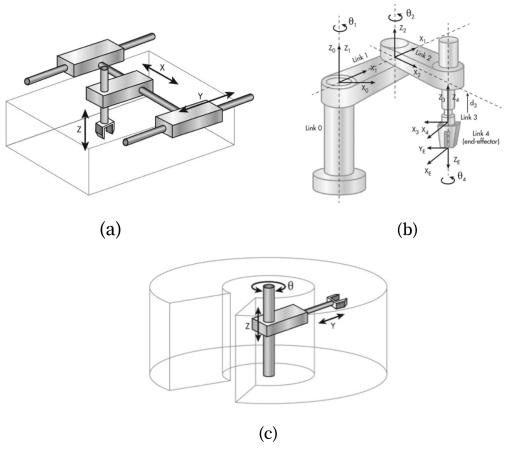


Figure 1.1: Different industrial robot [12] (a) Cartesian robot (b) SCARA robot (c) Cylindrical robot

A parallel robot is composed of two platforms and several parallel kinematic chains. The two platforms are the mobile platform and the fixed platform, where the mobile platform is fixed with its end-effector [13]. Figure 1.2 presents Delta Robot FANUC M-3iA, which is the famous parallel robot. It has three carbon fiber lightweight arms which are attaching to a small tool plate that can be attached to the end effector. The movement of the joints allows the tool plate to create a cylindrical work envelop [14]. Its low inertia allows it to have very fast speed and acceleration, and it has a wide range of uses in the food, pharmaceutical, and electronic industries.

Figure 1.2: Delta Robot FANUC M-3iA

The last but most common type of industrial robot is the articulated robot [15]. It is shaped like a human arm and has two or more axes. It is made of rigid rods connected by joints. This type of robot is widely used in the manufacturing industries. Figure 1.3 [16] shows the articulated robot FANUC Arc Mate 120iC, which is a 6-axis robot and is often used for welding due to its good repeatability.

In this project, the articulated robot FANUC M-20iA is used to test the algorithm. Like FANUC Arc Mate 120iC, it is also a 6-axis industrial robot and has 6 degrees of freedom (DOF). Figure 1.4 shows the robot in the Concordia University Advanced Mechatronics and Robotics Lab, whose maximum payload to carry is 20kg and its workspace can reach up to 1811 mm. It has a repeatability of 0.08mm, but often fails to achieve this accuracy under load, when it is subject to environmental

influence and some uncertain disturbance.

Figure 1.3: Articulated Robot FANUC Arc Mate 120iC

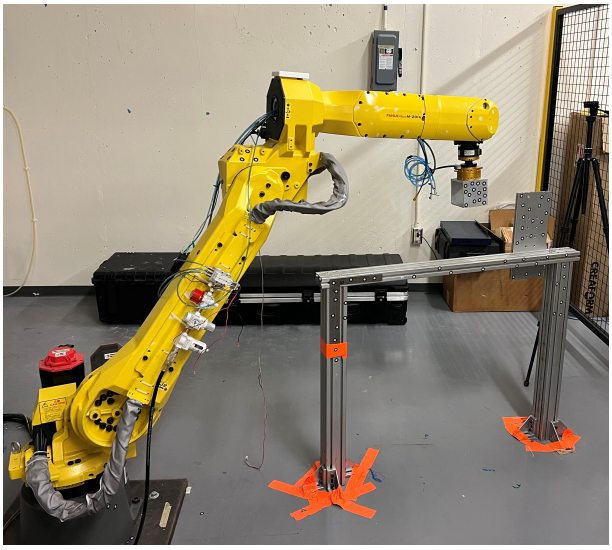


Figure 1.4: FANUC M-20iA in Concordia University Advanced Mechatronics and Robotics Lab

1.1.1 FANUC Robot Programming

FANUC robots can be programmed in two ways. The first is online programming on the Teach Pendant (TP). The second is offline programming via other simulation software, such as ROBOGUIDE for FANUC robots. The essence of online programming is to record and read the position and

orientation of the end effector directly by using the teach pendant. Because the robot is jogged in manual mode when recording and reading, simple paths such as straight lines, circles and polygons are efficient using online programming. However, for irregular paths or path planning, offline programming is required.

Offline programming can be written separately in the simulation software ROBOGUIDE so that there is no impact on the robot. This software is based on a nominal kinematic model in which the behavior of the robot can be evaluated to check singularities and possible collisions. The offline program can be uploaded to the robot if the simulation results are good. However, due to the difference between the real environment and the simulation environment, the program may not perform as expected.

Whether programming online or offline, the path is generated by connecting a series of points. Between each two points, the basic motion formats of FANUC robots are divided into the following: Joint motion (J), Linear motion (L), Circular motion (C), and Circle arc motion (A). Joint motion is the basic way of FANUC robot motion. When the robot is running in joint motion mode, all joints will do the minimum movement. The robot also accelerates along all axes at the same time. After moving at a defined speed, it decelerates and stops at the same time. Therefore, its trajectory is usually non-linear and the attitude of the end-effector is arbitrary during the movement. Linear motion, as the name implies, is the movement of a robot along a straight line from a starting point to a target point. It ensures that the motion of the end-effector is linear by actuating all joints. Both Circular motion and Circle arc motion move in a circular arc from the start point to the target point, but between the two points, another via point needs to be inserted. The difference between them is in the program writing, the Circle arc motion needs to teach two points in one line, while the Circular motion only teaches one position in one line. Figure 1.5 [17] shows the trajectory and TP program of the four motion commands.

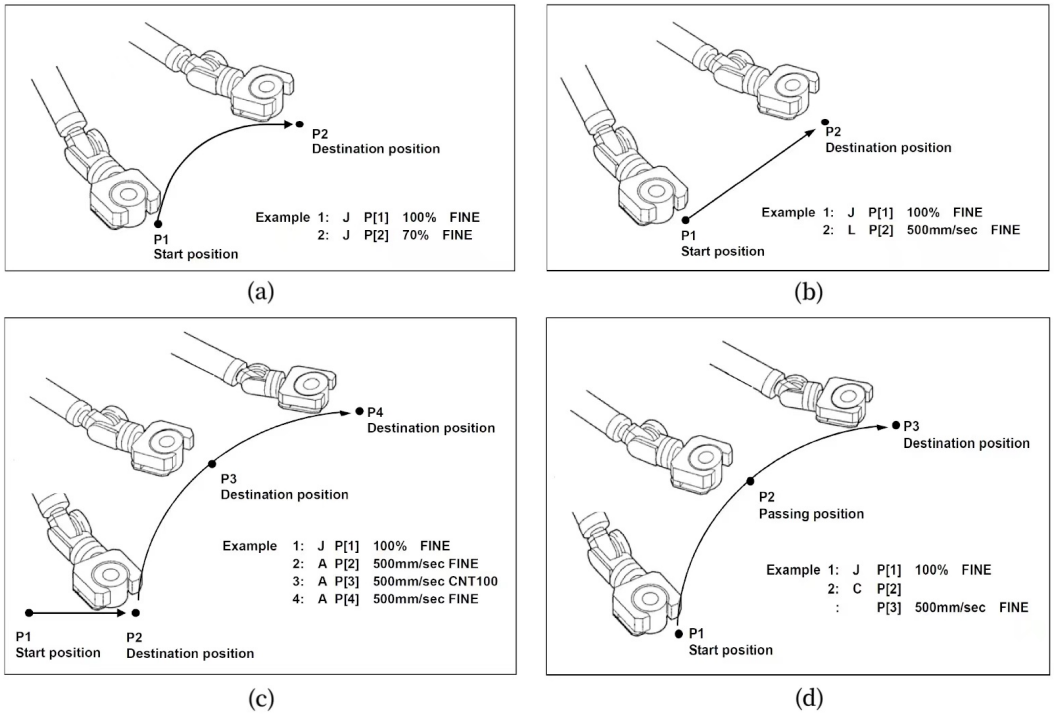


Figure 1.5: Four motion formats (a) Joint motion (b) Linear motion (c) Circular motion (d) Circle arc motion

1.1.2 DPM Introduction

Dynamic Path Modification (DPM) is a practical function on the FANUC M-20iA robot that allows real-time modification of the path through a third-party interface during robot operation. DPM can be divided into Modal DPM and Inline DPM depending on whether the modified object is the whole path or a destination [17]. The differences between these two are shown on Figure 1.6.

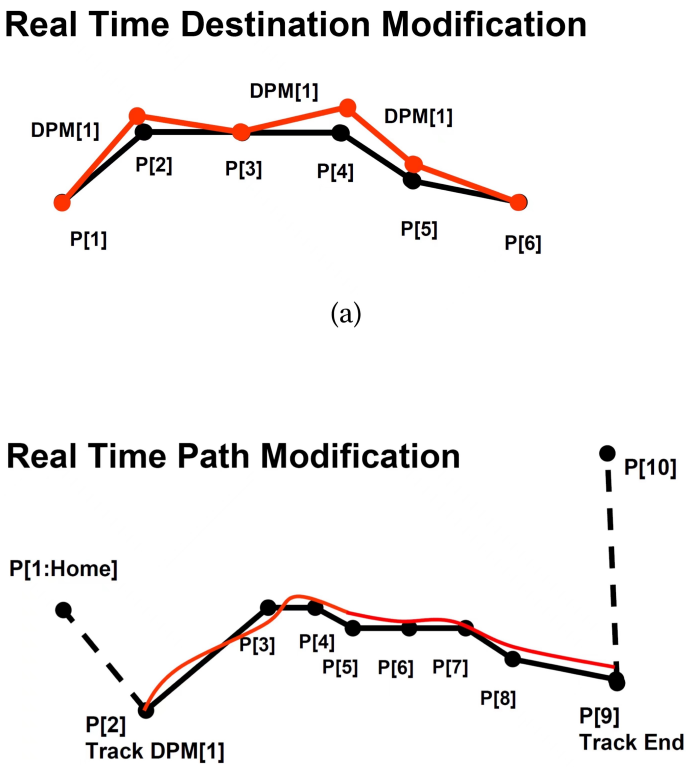


Figure 1.6: Two types of DPM (a) Modal DPM (b) Inline DPM

In DPM, the robot will set the offset continuously under the DPM motion program until the end of the DPM motion. Figure 1.7 illustrates this logic flow. In Modal DPM mode, the robot can also perform stationary tracking. In this motion, the robot does not make path modifications on the path until it reaches the target position. After reaching to the target position, the robot executes DPM commands to make corrections to the position. When the specified synchronization digital input signal shows TRUE, the robot will continue with the next motion. Figure 1.8 shows the two different TP programs of the Modal DPM mode.

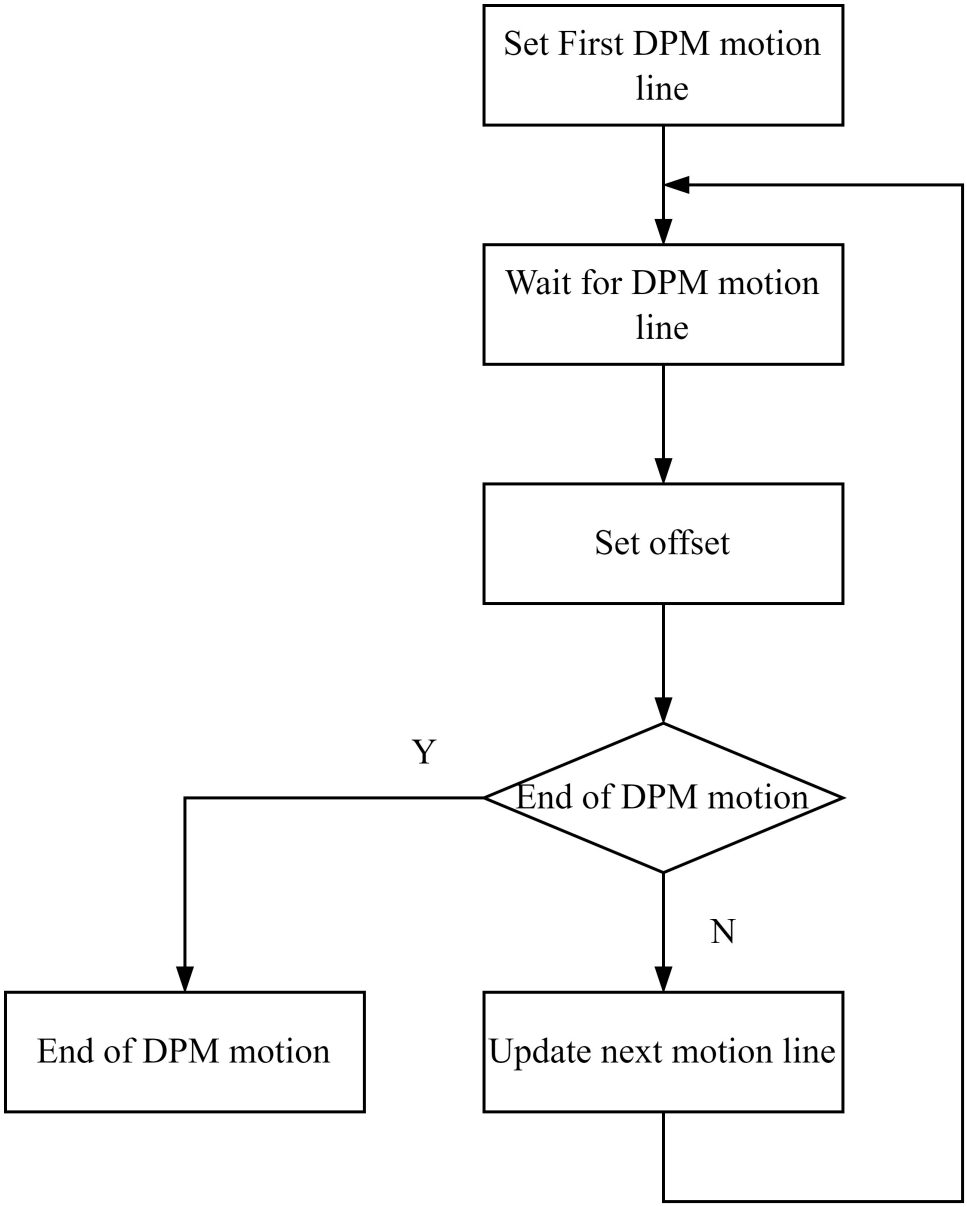
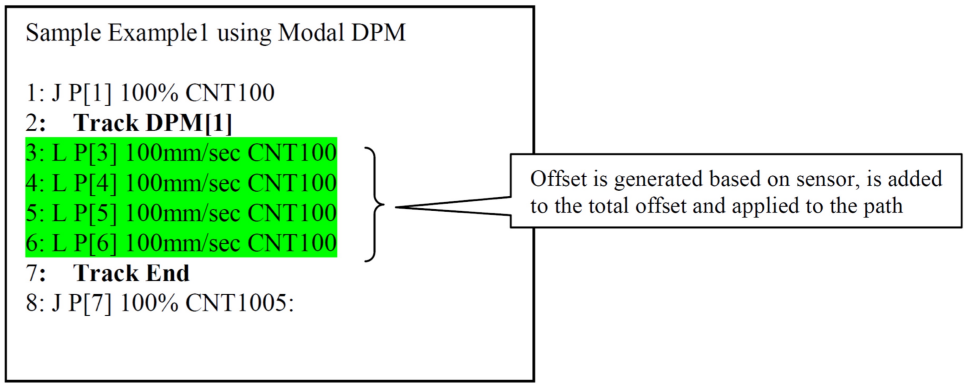
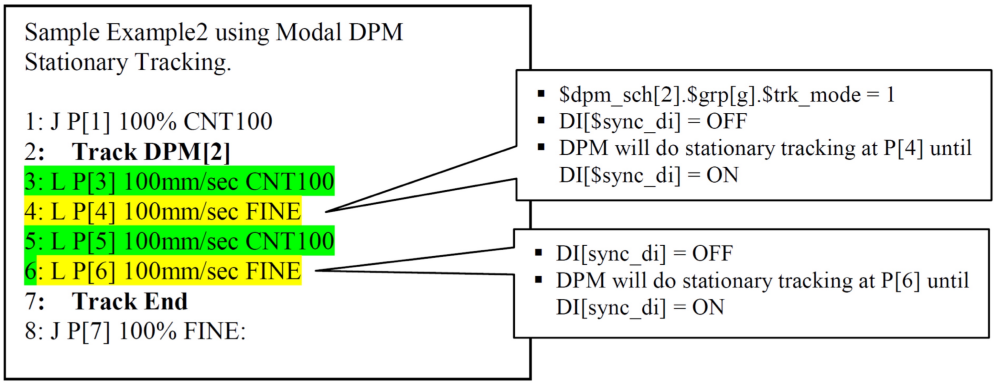


Figure 1.7: Block diagram of the DPM



(a)



(b)

Figure 1.8: (a) TP program example (b) TP program example for stationary tracking

1.2 Motivation

One of the most important ways to improve robot accuracy is static calibration, but traditional calibration methods are time consuming. Moreover, their accuracy is affected because of changing workload and other uncertainties, so a method of dynamic path tracking assisted by photographic sensors is proposed [1], which can improve the accuracy to 0.2 mm. The previous method uses PD control, which has the problem of not being robust when the robot is subject to uncertainties and disturbances. Thus, in this research, a PID control based on neural network, which can tune the

coefficients of PID control adaptively, is proposed, thus enhancing the robustness and accuracy of the robot.

1.3 Contribution

In this research, a novel dynamic path tracking technique is proposed, which uses the photogrammetry sensors to detect the pose of end effector and employ the neuro-PID to tune the coefficients online. The pose information is processed by robust Kalman filter (RKF) to filter out the contaminated noise. The neuro-PID controller learns and tunes the PID parameters through the learning ability of neural network, and the adaptive nature of the neural network also enables the PID parameters to be tuned in real time according to the environment to achieve the optimal control effect. Using this method, the pose accuracy is improved to ±0.08mm.

This method, which does not depend on kinematic chain parameters kinematics of the robot, can be employed for dynamic calibration task of many industrial robots. It is more convenient than traditional methods and can be run directly on the developed software, thus reducing the calibration time.

In this method, all the control and adaptation process are in real time, which increases the practicality of this method.

1.4 Outline

This thesis has six main chapters. Chapter 1 is the introduction of the industrial robots. It also consists of the programming method and DPM function of the FANUC M-20iA. Then it shows the motivation and contribution of this research. Chapter 2 gives a literature review on the visual servoing, Kalman filter and neural networks. In Chapter 3, the robust Kalman filter is proposed to remove the noise from the signals which contains the pose information. Chapter 4 focuses on PID controller and the design of the novel PID control based on neural networks. The experimental setup and results are presented in Chapter 5. Finally, Chapter 6 concludes the whole thesis and shows the future works.

Chapter 2

Literature Review

In this chapter, a comprehensive literature review on visual servoing for industrial robots, Kalman filter handling the signal noise and neural network control is given.

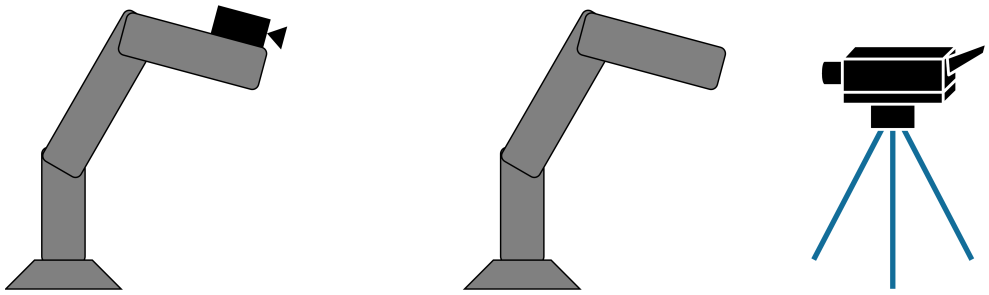
2.1 Visual Servoing

Visual servoing is the technique of machine vision and robotics control. Machine vision, as a machine bionic system similar to the human eye, uses an optical device, such as camera, to acquire information about real objects and then process and execute the related information.

In the 1960s, due to the development of robotics and computer technology, people started to study robots with vision function. The vision system of a robot obtains the target pose through image processing and then calculates the pose for robot motion based on the target pose. In the early 1970s, Shirai and Inoue applied vision systems to control systems in assembly tasks. This method improve the accuracy by feeding the calculated relative errors back to the manipulator [18]. This process was called “Visual Feedback” during this period. It was not until 1979 that Hill and Park [19] first introduced the concept of ”Visual Servoing”. Obviously, the meaning of “Visual Feedback” is only to extract the feedback signal from visual information, while “Visual Servoing” includes the whole process from visual signal processing to robotics control. So visual servoing can reflect the relevant research related to robotics vision and control more comprehensively than visual feedback.

Visual servoing can be used with industrial robots in industrial applications such as assembly, spot welding, and painting [20]. The visual servoing is also widely used in the medical application [21] because it is equipped with a camera and can perform some surgeries with the real-time picture from the camera.

Depending on the camera installation configurations, visual servoing is also classified as eye-in-hand (EIH) and eye-to-hand (ETH) [22]. When the camera is mounted on the robot arm and the position of end effector is fixed relative to it, this configuration is called EIH. In ETH, the position of the camera is fixed relative to the robots base frame [23, 24]. Figure 2.1 shows these two different configurations.



(a) (b)

Figure 2.1: Camera installation configuration (a) Eye-in-hand (b) Eye-to-hand

In vision servoing system, according to the number of cameras, it can be divided into monocular vision system, binocular vision system, or stereo vision system. A single camera minimizes the processing time required to extract visual information, but the lack of depth information of the object imposes limitations on the servo operation. In this case, if the object model in unknown, the stereo vision system can provide the complete 3D information with higher accuracy [25].

Depending on the type of feedback signal, there are three main types of visual servoing: Position Based Visual Servoing (PBVS), Image Based Visual Servoing (IBVS) and Hybrid Visual Servoing (HVS) [26].

In PBVS, as shown in Figure 2.2, its initial given information and feedback information are presented in the form of Cartesian space. In this space, the acquired image signal is used to obtain the current pose information of the robot, and then the difference between the current and desired pose is transmitted to the controller, thus forming closed-loop feedback. IBVS, shown in Figure 2.3,

is to define both the given information and the feedback information in the image feature space. By comparing the relevant features of the current image and the desired image, the error relationship between the two images is found and then used as a control signal.

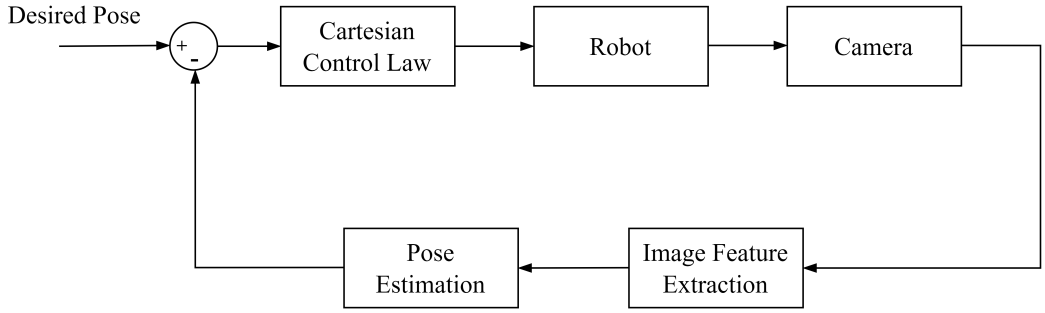


Figure 2.2: Position-based visual servoing

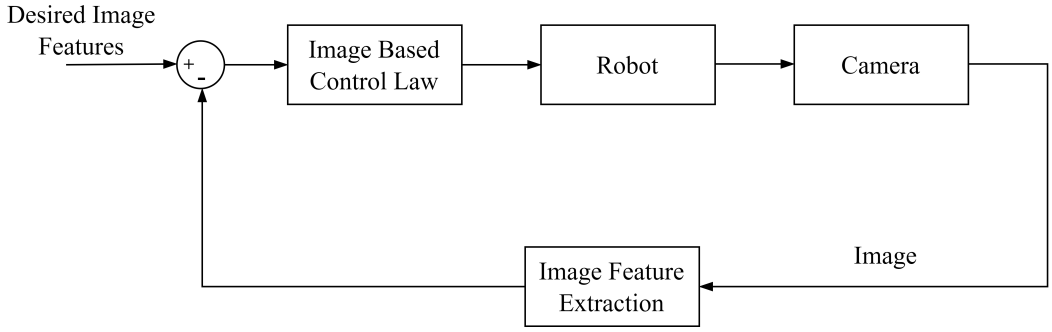


Figure 2.3: Image-based visual servoing

Both PBVS and IBVS have their own advantages and disadvantages. Therefore, Malis et al. [26] proposed a hybrid visual servoing approach. The block diagram is presented in Figure 2.4. Compared with IBVS, HVS, allows for strict convergence in the task space. Compared with PBVS, HVS does not require 3D model information of the object. In HVS, the desired pose information of the robot needs to be known in advance. The corresponding homography matrix can be obtained from the relationship between the current pose of the robot and the desired pose, and then the robot is controlled by this matrix for rotation operation, followed by the translation information obtained from the change of image features, and thus the robot is controlled for translation operation. This method achieves a partial decoupling of the translational and rotational motions of the robot, which can also be referred to as visual servoing control based on homography matrix [26].

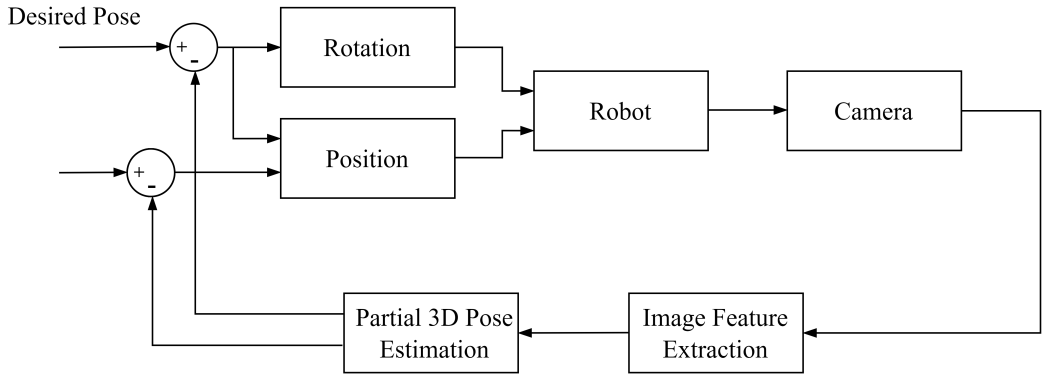


Figure 2.4: Hybrid visual servoing

In IBVS, the robot motion in image space, which is robust compared to camera calibration errors. However, the singularities of the image Jacobian matrix are unavoidable in this method and there is also the problem of local minima. Compared with IBVS, PBVS allows the design of feasible controllers for robotic systems in Cartesian space. This is crucial for applications requiring high accuracy robot path following. HVS, although combining the advantages of both methods, is more sensitive to image noise and increases the complexity of the method. Therefore, PBVS is chosen in this research.

2.2 Kalman Filter

In visual servoing, since the transmitted image signal has noise, an algorithm is needed to figure out this noise, i.e., Kalman filter (KF). KF is essentially an optimal recursive data processing algorithm. It has been used for more than 30 years in a wide range of applications, including robot navigation, control, sensor data fusion, and even in military applications such as radar systems and missile tracking.

Rudolf E. Kalman [27] proposed an algorithm which is used to estimate unobservable state variables based on observable variables that may have some measurement error. This algorithm is usually applied to a linear stochastic system, and it estimates the optimal system state from the measured output containing random noise. This optimality is established in the sense that the mean squared errors (MSE) is minimized; therefore, it is known as linear least-mean-squares estimation. In terms of practical applications, this observer is more widely used in non-stationary processes.

Since it can follow some optimal criteria to restore the signal, it has a function equivalent to that of a filter: noise reduction. This is the reason why it is called the Kalman filter.

Due to the above characteristics of the Kalman filter, it is appropriate to apply it to pose estimation. It has the following advantages in the pose estimation. First of all, it is able to fuse data from different sensors. Secondly, it can improve dynamic estimation because it implements temporal filtering [28]. Thirdly, since the Kalman filter can predict the state at the next time, it can also be used as a windowing technique, which is crucial in image processing. Last but not least, the Kalman filter plays an important role in the design of control systems because it can estimate the pose velocity. However, if the relative position estimation problem is to be solved using the Kalman filter, the dynamics model and output equations must be defined in advance [29].

When designing the model, the dynamics model is linear because of the characteristics of the standard Kalman filter (SKF) used for linear systems. However, when the dynamics model under observation is nonlinear, it will lead to inaccurate estimation results. The extended Kalman filter (EKF) as a nonlinear version of the Kalman filter can be a good solution for problems in nonlinear systems. It does not use a predetermined linear dynamics model, but linearizes the nonlinear dynamics model before completing the state estimation. Hence, the estimation of EKF is more accurate than that of SKF in the nonlinear system. In some research, the EKF has been used for robot pose detection [30, 31]. In addition to EKF, there is a similar algorithm called UKF, which uses unscented transformation (UT) to estimate the covariance matrix; therefore, UKF has a lower error in the estimation results. It was used in [32] for the pose estimation. Whether SKF for linear systems or EKF and UKF for nonlinear systems, the quality of the filtering still depends on the process noise covariance Q and the measurement noise covariance R. Therefore, choosing the appropriate Q and R remains a problem in designing the Kalman filter. To solve this problem, an algorithm called adaptive Kalman filter (AKF) was proposed in [33]. There are many methods that can adjust Q and R. The most commonly used one is the covariance estimation correction based on forgetting factor [34, 35]. Another AKF is used to deal with the estimation error of state error covariance P due to the uncertainty of the dynamic model [36]. In this method, the correction coefficient is calculated by comparing the theoretical estimated P with the estimated P, thus correcting P affected by system uncertainty to the real P. Therefore, it is also called the adaptive optimal Kalman filter (AOKF) [36].

The limitation of Kalman filter is the requirement of precisely known system model parameters and noise statistics [37]. In many practical application problems, measurement system errors arise due to the influence of the surrounding environment, errors caused by the measurement equipment itself, and improper selection of models and parameters. Therefore, it is difficult to eliminate such system errors using the SKF. Moreover, these system errors can lead to errors in the estimation of the state error covariance P. Therefore, to cope with this issue, this thesis proposes a Robust Kalman filter (RKF) which adaptively corrects for the estimated P by comparing the theoretical and estimated P.

2.3 Neural Network

Artificial neural network (ANN) are artificial intelligent (AI) methods that simulates the function of human brain. It mainly draws on the process of information processing in the nervous system of human brain. It is based on the theoretical foundation of mathematical network topology, and is characterized by large-scale parallelism, high fault tolerance and functions of self-adaptation, self-learning and self-organization, integrating information processing and storage. Therefore, it has a wide range of application prospects. The research of ANN involves computer science, cybernetics, information science, microelectronics, and other disciplines. As a branch of intelligent control, ANN has attracted much attention from the control community for its unique non-traditional expression and learning capability.

Neural networks have been developed for more than half a century. In 1943, McCulloch and Pitts [38] collaborated to build the first mathematical model (MP model) of a neuron. In 1949, Hebb proposed the first learning rule in [39], which explains how nerve cells in the brain change and adapt during the learning process. In 1958, Rosenblatt [40] generalized the MP model by adding a learning mechanism based on it. At the same time, he introduced the concept of perceptron, which was the first time to put the theory of neural networks into engineering practice. The neural network boom continued until 1969, when Minsky and Papert [41] pointed out the limitations of perceptron in their book Perceptron, which led researchers to turn their attention away from the study of neural networks. Nevertheless, some scientists persisted in the field of neural networks and achieved some

important results. In 1976, Grossbery [42, 43] proposed adaptive resonance theory (ART). Three years later, in 1980, Fukushima [44] proposed the neocognitron, which is the original concept of Convolutional Neural Networks (CNN). Since then, a variety of new breakthroughs and studies have emerged. In 1982, Hopfield [45] proposed a new neural network model, the Hopfield neural network (HNN). In this model, he introduced the concept of energy function for the first time and gave the basis for determining the convergence of the network. In 1985, Ackley et al. [46] introduced the simulated annealing algorithm to the neural network and proposed the network model of Boltzmann Machine. This model algorithm provides an effective method for neural network optimization calculation. In 1986, Rumelhart and McClelland et al. [47] proposed parallel distributed processing (PDP) to rediscover and improve the backpropagation (BP) neural networks algorithm, which is currently the most common network and is widely used for solving practical problems.

Nowadays, many fields are permeated with neural networks, which achieve good results in image processing, pattern recognition, robot vision, etc. How to use neural network theory more effectively in the field of industrial intelligent control and better combine it with conventional PID control to achieve more ideal control effect has become a hot topic for many researchers.

In the field of control, neural networks can be combined with control techniques to design the controllers because of their nonlinear system identification ability. In [48], Yanagawa and Miki proposed a PID controller based on a single neuron, which was applied to a DC servomotor system. They illustrate the good tuning performance of this approach. Although single neuron has a simple structure, it has low mapping ability for nonlinear objects, so multi-layer perceptron (MLP) is proposed to solve nonlinear problems. MLP belongs to feedforward neural network. Radial basis function neural network (RBF) is a single hidden layer neural network. The paper [49] proposes a parameter self-tuning control method combining RBF neural network and PID control for the time-delay and parameter time-varying characteristics in temperature control systems. In addition, control algorithm based on neural network plays an important role in other fields as well. In the biomedical field, the combination of neural networks and PID can be well applied to nonlinear control system of the lower extremities of the exoskeleton and correct errors due to parameter variations during the movement of the lower extremities of the exoskeleton [50]. In the field of aviation, variable configuration spacecraft can generate large uncertainties and disturbances due to configuration

changes. Ran et al. [51] propose an adaptive fuzzy neural network (FNN) based on RBF neural network to solve this problem, where the parameters of the PID are tuned by the FNN. The results show that this controller has better performance than the conventional PID controller in terms of response speed and robustness. In addition to the above nonlinear control systems, an industrial robot system is also a complex multi-modal nonlinear system. In [52], Jiyue et al. authors, while studying the trajectory tracking control algorithm for industrial robots, proposed a novel PID control algorithm based on the core idea of fuzzy neural network algorithm in order to solve the impact on robot trajectory tracking accuracy due to acceleration and velocity. From the simulation experiment result, the stability and good control accuracy of the method are proved. In addition to the RBF neural network, another feedforward neural network is the BP neural network. BP algorithm is simpler than RBF in solving problems with the same accuracy requirements. When the number of training samples increases, the number of hidden layer neurons of RBF network is much higher than that of BP neural network, which makes the structure of RBF network will be too large. In [53], Fatma et al. authors developed an adaptive feedforward PID controller based on BP neural network to control the joint position of the robot. Yu and Hui [54] use two BP neural networks for prediction of the output of the nonlinear process and for automatic adjustment of the PID parameters, respectively. In this research, inspired by [53–56], a BP neural network-based PID controller is proposed. Combining BP neural networks and control systems can make the PID parameters change in real time according to the environment to achieve a desired control effect by the adaptive nature of the neural network. Thus, the accuracy of path following for industrial robot can be improved.

2.4 Summary

In this chapter, the basic concepts and categories of visual servoing are introduced. Then the Kalman filter for processing image signal noise is introduced, and the different application scenarios of different kinds of Kalman filters are analyzed. Finally, the development of neural networks is presented and a novel trend in the field of control is introduced, which is the combination of neural networks with PID control.

Chapter 3

Pose Estimation

In this chapter, a robust Kalman filter (RKF) for pose detection of industrial robots is presented. This method is based on the correction of the state error covariance P which is a measure of the estimated accuracy of the state estimate. A larger covariance P indicates that the predicted result has a larger deviation from the true value, namely, the estimate is less reliable; a smaller covariance P indicates that the predicted result is closer to the true value, namely, the estimate is more reliable. Because uncertainty can make P inaccurate, although P is updated with a new value after each prediction and update progress, a more accurate P will make this iterative process shorter and make the results converge to the true value faster. The RKF proposed in this chapter can solve the filtering problem for systems with uncertainties and disturbances.

3.1 Robust Kalman Filter

3.1.1 Standard Kalman Filter for Pose Detection

The filtering process is divided into two phases: prediction and correction. In the prediction phase, the SKF uses the error covariance calculated from the current position to estimate the new position of the target. In the correction phase, SKF records the target location and calculates the posteriori estimate for the next cycle. The time update equations are as follows,

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \tag{1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2}$$

where $\hat{x}$ is defined as estimated state vector, Q is the process noise covariance matrix and A is 12×12 state transition matrix and the diagonal elements are 1. Then the $A_{i,i+6} = T$ where T is sampling interval and i is from 1 to 6. B is called control-input matrix. After obtaining P_k^- which is called priori estimate error covariance, the measurement update equations are as follows,

$$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1} \tag{3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(Z_k - H\hat{x}_k^-\right) \tag{4}$$

$$P_k = \left(I - K_k H\right) P_k^- \tag{5}$$

where R is the measurement noise covariance, Z_k is measurement vector and H is measurement matrix and equals to $[I_{6\times6}\ O_{6\times6}]$. The measurement update equations use the P_k^- to calculate the Kalman gain K_k, which can be applied to update the estimation and error covariance.

3.1.2 Robust Correction of the State Error Covariance

The uncertainties and random disturbance in system lead to the incorrect estimation of P. To solve this problem, the method which P can be corrected automatically is suggested. After comparing the estimate P and theoretical evaluated $\bar{P}$, the correction factor α is added to compensates the inaccuracy of P. The result is called $\hat{P}$, which is as follows,

$$\hat{P}_k = \frac{P_k}{\hat{\alpha}_k} \tag{6}$$

where the $\hat{\alpha}$ is defined as follows,

$$\hat{\alpha}_k = \beta\hat{\alpha}_{k-1} + (1-\beta)\alpha_k \tag{7}$$

where the β is the forgetting factor coefficient which is from 0 to 1.

$$\alpha_k = \begin{cases} 1 & \text{tr}\left(\bar{P}_k\right) < \text{tr}\left(P_k\right) \\ \frac{\text{tr}(P_k)}{\text{tr}\left(\bar{P}_k\right)} & \text{otherwise} \end{cases} \tag{8}$$

where $\bar{P}$ is obtained as follows,

$$\bar{P}_k = \omega\left(d_{k-1}d_{k-1}^T\right) + (1-\omega)\left(d_k d_k^T\right) \tag{9}$$

$$d_k = O_k - H\hat{x}_k \tag{10}$$

where ω is the filter coefficient which is valued between 0 and 1, O_k is the vector of noisy signals of the robot end-effector pose, acquired from vision measurement system.

3.2 Robustness

A feedback system is robust in the sense that the system is stable under a particular type of uncertainty, with asymptotic regulation and dynamic properties that remain constant. In practical control problems, uncertainty is often bounded. According to the range bound of uncertainty change, the worst-case control system design within this range is the basic idea of robust control system design. If the system is robust in the worst case, then it must be robust in other cases as well.

In general, optimal estimation strategies are derived from specific signal models and noise environments, such as the linear model and Gaussian white noise, while in practice such assumptions are not precise. In this case, an outlier can greatly affect the accuracy of the parameter estimates (e.g., mean, variance). For the Kalman filter, the Gaussian assumption is a guarantee of its optimality. Although it is optimal in the case of linear and Gaussian white noise, it has no resistance to outlier. Moreover, the Kalman filter is updated online, and an outlier will not only affect the current estimate, but also contaminate many future states in time, thus making it impossible to track the signal well. The robustness of the Kalman filter is such that the filter not only gives optimal results when the assumptions hold (i.e., linear model and Gaussian white noise), but the results are still acceptable when the observations deviate from the assumptions.

3.3 Results and Discussion

In this section, the RKF and SKF are compared by experiment. In the experimental test, the robot follows a desired path for linear motion, and PBVS and PID control are used to ensure accuracy. Also, an acceleration and deceleration were added during the process to test the performance of the RKF. The parameters of SKF and RKF are shown in Table 3.1 and Table 3.2.

Table 3.1: Parameters of the SKF

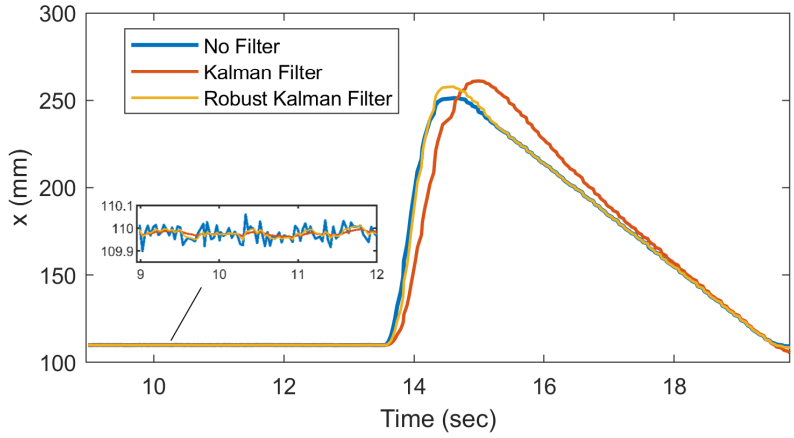
Parameter	**Value**
Q	diag[0.001 0.001 0.001 0.0002 0.0002 0.0002 0.001 0.001 0.001 0.0001 0.0001 0.0001]
R	diag[0.003 0.003 0.003 0.0006 0.0006 0.0006]

Table 3.2: Parameters of the RKF

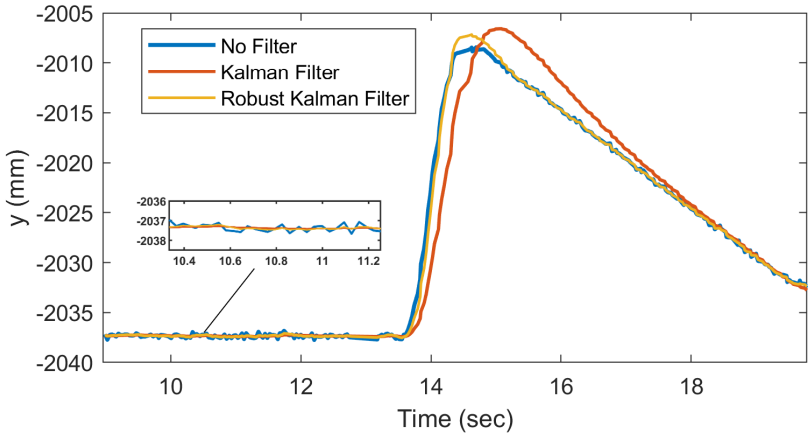
Parameter	**Value**
Q	diag[0.001 0.001 0.001 0.0002 0.0002 0.0002 0.001 0.001 0.001 0.0001 0.0001 0.0001]
R	diag[0.003 0.003 0.003 0.0006 0.0006 0.0006]
α	0.9
β	0.995
ω	0.9

3.3.1 Experimental Results

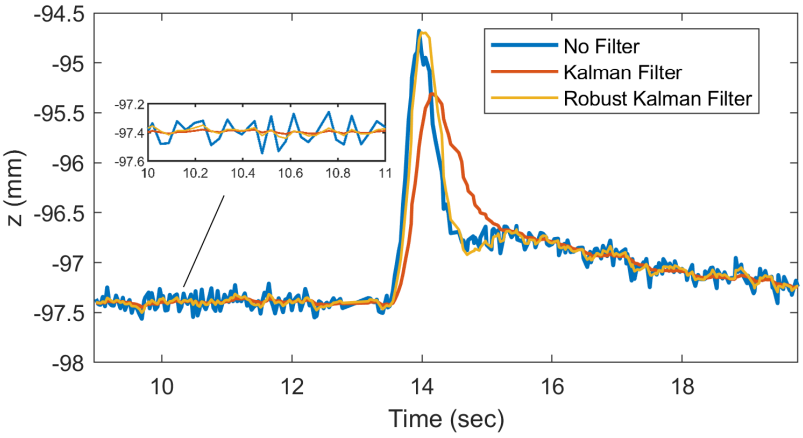
In the experimental test, C-Track continuously acquires pose information of end-effector at 29 frames per second while the robot is following the desired signal. The signal is then filtered through the mentioned KFs, and the experimental results are shown in Figure 3.1.



(a)



(b)



(c)

Figure 3.1: Comparison result of the filters

In this figure, it is clear that the RKF is closer to the true signal than the SKF, and SKF can not track the signal correctly. SKF has significant errors in the prediction of all pose variables for the end-effector. When the rate changes between 14s and 16s, the lag of SKF is obvious and converges for a long time. However, the RKF tracks the signal well, which reflects the fact that the RKF is more robust than the SKF in case of abrupt changes. In addition to a figure, the numerical indicators of these two filters are shown in Table 3.3 and Table 3.4. According to this table, the RKF has lower root mean square (RMS) and standard deviation (SD) values than the SKF for all variables, which indicates that the RKF is smoother than the SKF and the filtering error is lower.

Table 3.3: RMS for the filters

Filter	**x**	**y**	**z**
SKF	8.5520	1.7195	0.2067
RKF	1.7305	0.3648	0.0900

Table 3.4: SD for the filters

Filter	**x**	**y**	**z**
SKF	8.3094	1.6714	0.2018
RKF	1.6836	0.3558	0.0886

3.4 Summary

In this chapter, a robust Kalman filter is proposed. It can improve the filter performance by correcting the estimation P in the presence of disturbance. Experimental tests show that the proposed RKF can effectively reduce the effect of noise on the pose estimation signal and outperforms the SKF in terms of smoothness and filtering error. With regards to lag and convergence, compared to SKF, the RKF has smaller lag and shorter convergence time, especially when the rate changes. In the next chapter, the pose information after RKF will be used as the feedback to the dynamic path tracking controller.

Chapter 4

Designing Adaptive-neuro PID controller

Once the pose information is obtained from RKF, it is used as the feedback signal for the PBVS to fulfill the dynamic path tracking. In order to obtain the high accurate dynamic path tracking, a PID control which is a non-model based control is used to eliminate the difference between the desired signal and the output value. In the presence of disturbances in the system, the control effect is not ideal because parameters of conventional PID control cannot be tuned in real time. To solve this problem, this chapter proposes a new PID control algorithm combining neural network and conventional PID control, i.e., neuro-PID controller, which can adjust the parameters in real time to achieve better control performance.

4.1 Conventional PID Control

The PID control is a mature and effective algorithm capable of linearly combining system errors and then feeding them back to the controlled object. With the development of control theory, PID control frequently appears in various industries. The closed-loop system in this research consists of a PID controller, a servo system and robust Kalman filter, as shown in Figure 4.1.

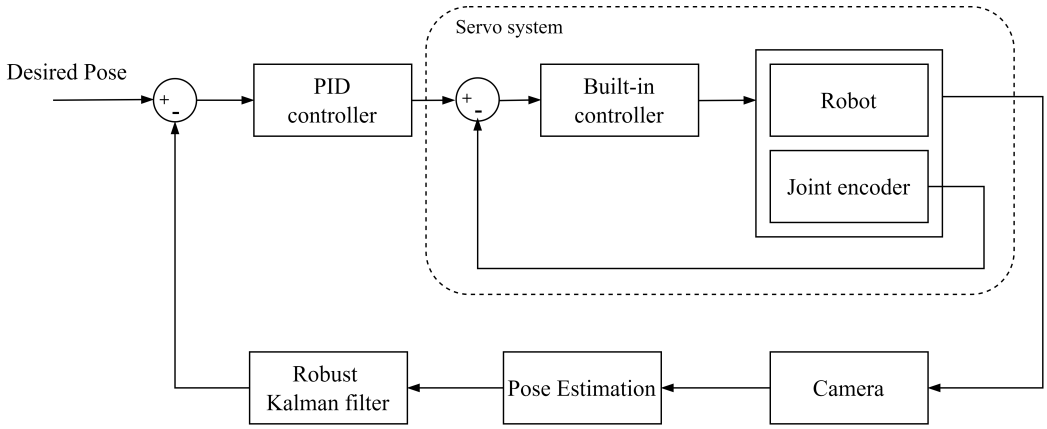


Figure 4.1: A typical PBVS with PID control block diagram

4.1.1 Conventional PID control introduction

The working principle of PID controller is to calculate the error e between the desired value $r(t)$ and the real output value $y(t)$, and then feed it into the three modules of proportional, integral and differential. The combined output is the closed-loop control signal $u(t)$. Then it is fed into the controlled object to make the output $y(t)$ close to $r(t)$, and finally realize the control of the controlled object. The mathematical expression of this process is as follows.

$$u(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t)dt + T_D \frac{de(t)}{dt} \right] \tag{11}$$

where K_p, T_I and T_D are the proportional gain, integral time constant, and derivative time constant respectively.

In the PID control process, the proportional part can calculate the system error value proportionally and feed it back to the controlled object immediately, but it is prone to steady-state error. The integral part integrates the error value and then feeds it back to the controlled object to reduce the steady-state error. The differential part is able to predict the trend of error based on its derivative to shape the system response.

4.1.2 Parameter Tuning

The proportional element is able to reflect proportionally the error signal of this control system. When the K_p increases, the system response will be faster and the steady-state error of the system

will be reduced, which improves the control accuracy of the whole system. However, if K_p becomes too large, a large overshoot will be generated. If K_p becomes too small, the system overshoot will decrease and the steady-state margin will increase, but then the regulation accuracy of the system will be reduced.

The integrating element is mainly used to eliminate steady-state errors. However, it has a hysteresis phenomenon, which makes the system response slower and the overshoot larger. When the integration time constant T_I is too small, the system integration effect will be too large, which will intensify the overshoot of the system and even generate oscillations.

The differentiation element reflects the trend of the error signal. Before the error becomes large, the system generates an effective correction signal, which effectively speeds up the system response time and thus reduces the regulation time. The disadvantage of the differentiation element is that it is less resistant to infection. When the differential time constant T_D is too large, the system brakes early, thus extending the regulation time. When the T_D is too small, the system brakes laggingly and the overshoot increases, corresponding to a slower system response.

The adjustment of the three parameters K_p, T_I and T_D is the core of the PID control. In the development of PID control, there are many methods for three parameter tuning, such as the Ziegler-Nichols method [57], the Kappa-Tau method [58] and Internal Model Control method [59].

Most of these parameter adjustment methods need to obtain the characteristic parameters of the controlled object in advance, and then calculate three parameters according to the known formula, which is more suitable for manual off-line adjustment. However, their process is more complicated and less efficient, and it is difficult to achieve the best effect of control. Since there are some uncertainties in modern control systems, the ability of conventional PID controllers to adjust the three PID control gains online must be addressed in order to solve the problem of these uncertainties.

4.2 Digital PID Control

Eq. 11 describes the continuous PID control algorithm, which can be implemented with the help of integral circuits, differential circuits and amplifiers of analog circuits. With the development of computer technology, PID control have been realized with software. In control engineering, the use

of computer PID control algorithms to implement digital PID controller can be flexible to change the PID parameters, which is not possible with analog PID controllers. But computer control is a kind of sampling control, need to sample the error value to do further calculations. Therefore, it is necessary to transform Eq. 11 from the continuous domain to the discrete domain, i.e., to convert analog continuous PID control to digital PID control. Digital PID control algorithms are usually divided into position PID and incremental PID.

4.2.1 Position PID Control

The specific steps to carry out the discretization process are: First, replace the continuous time t with the discrete sampling time point kT. (See Eq. 12). Then, use sums instead of integrals. (See Eq. 13). and replace the differential operation with first-order backward differencing (See Eq. 14).

$$t \approx kT \quad k = 0, 1, 2 \cdots \tag{12}$$

$$\int_0^t e(t)dt \approx T \sum_{j=0}^{k} e(jT) \tag{13}$$

$$\frac{de(t)}{dt} \approx \frac{e(kT) - e[(k-1)T]}{T} \tag{14}$$

In the above discretization process, the sampling period T must be short enough to ensure sufficient accuracy. For the convenience of writing, T is omitted, and then Eq. 12, Eq. 13, and Eq. 14 are substituted into Eq. 11 to obtain the discretized PID expression,

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_I} \sum_{j=0}^{k} e(j) + \frac{T_D}{T}[e(k) - e(k-1)] \right\} \tag{15}$$

or

$$u(k) = K_p e(k) + K_i \sum_{j=0}^{k} e(j) + K_d[e(k) - e(k-1)] \tag{16}$$

where K_i is the integral gain which equals to $K_p \frac{T}{T_I}$ and K_d is the derivative gain which equals

to $K_p \frac{T_D}{T}$.

The PID control law presented in Eq. 15 or Eq. 16 is called position PID control. The disadvantage of the positional PID control algorithm is that each output is related to the past state and the calculation is cumulative for $e(k)$, which leads to a large computer computing workload. Moreover, because the computer output of $u(k)$ corresponds to the actual position of the controlled object, if the computer fails, a large change in $u(k)$ will cause a large change in the position of the controlled object, which is not allowed in production practice. To avoid possible production accidents, incremental PID control is proposed.

4.2.2 Incremental PID Control

The incremental PID means that the output of the digital controller is simply the amount of change in the control quantity $\Delta u(k)$. According to Eq. 16, Eq. 17 can be obtained recursively.

$$u(k-1) = K_p e(k-1) + K_i \sum_{j=0}^{k-1} e(j) + K_d[e(k-1) - e(k-2)] \tag{17}$$

Subtract Eq. 17 from 16 to obtain Eq. 18.

$$\Delta u(k) = K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)] \tag{18}$$

Eq. 18 is incremental PID control algorithm formula.

4.3 ANPID Control Design

In order to achieve the desired performance for PID controller, adjusting its gains is necessary to form a relationship of both mutual cooperation and mutual constraints. This relationship is not only a simple linear combination, but also is from the infinitely variable non-linear combination to find the best. The arbitrary nonlinear expression ability of the neural network can realize the PID control with the best combination through the learning of the system performance. In this research, motived by [60], the adaptive neuro-PID (ANPID) is proposed, whose gains are tuned adaptively using a BP methods. The system structure of ANPID control is shown in Figure 4.2.

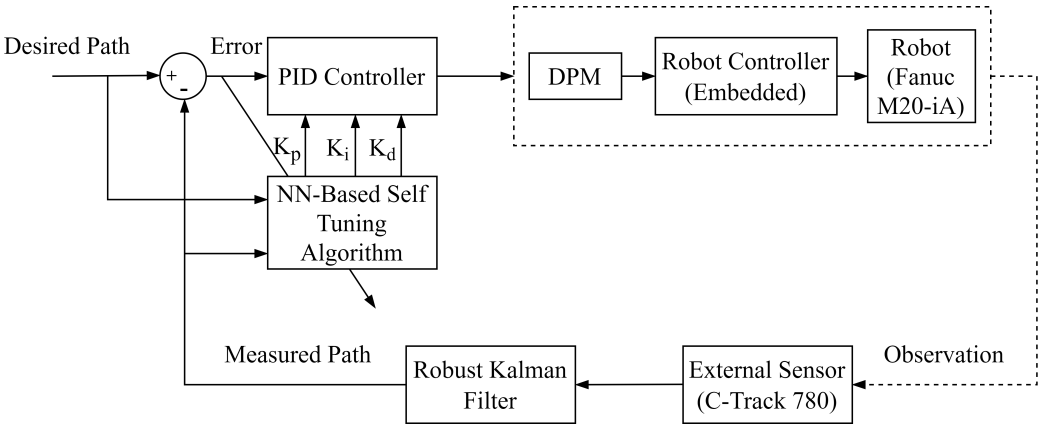


Figure 4.2: System structure of ANPID control

The controller consists of two parts. The first part is the conventional PID controller. It directly performs closed-loop control of the controlled object and the three parameters K_p, K_i and K_d for online adjustment mode. The second part is the MLP-NN. It adjusts the parameters of the PID controller according to the operating state of the system to achieve some optimization of the performance index.

Since the computer signal is discrete, incremental PID is used instead of conventional PID control, and the control algorithm formula is the same as Eq. 18.

The neural network uses three layers, which are the input layer, hidden layer and output layer. The structure is shown in Figure 4.3.

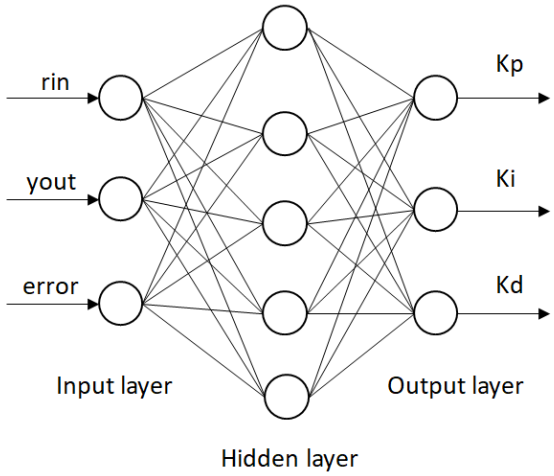


Figure 4.3: Three layer MLPNN structure

The design steps of the neural network are as follows.

(1) Determine the MLP-NN structure. The number of nodes M in the input layer, the number of nodes Q in the hidden layer and the number of layers in the hidden layer should be determined in the smallest possible way, which can reduce the system size, system complexity and learning time. The number of neuron nodes in the input and output layers is related to the actual problem. In the PID controller, the output nodes are set to three, corresponding to the three parameters of PID, K_p, K_i and K_d. The hidden layer has the function of extracting features from the input signal. In PID control, one or two hidden layers are sufficient to meet the requirements.

(2) Select the initial values of the connection weights for the hidden layer as well as the output layer. Take the commonly used activation function Sigmoid function as an example, at both ends of its curve, the function value tends to be stable, so the gradient value is close to 0 and the modification of the connection weights will be very small, which makes the network learning very slow. When initializing the connection weights $w_{ij}^{(2)}(0)$ and $w_{li}^{(3)}(0)$, a small random number is generally chosen.

(3) Select the activation function. The activation function is chosen as hyperbolic tangent function, which is also a Sigmoid function.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{19}$$

The derivative is:

$$f'(x) = 1 - \tanh^2(x) = 1 - f^2(x) \tag{20}$$

Due to the K_p, K_i, K_d is positive, we choose non-negative hyperbolic tangent function as output layer activation function:

$$g(x) = \frac{1}{2}[1 + \tanh(x)] = \frac{e^x}{e^x + e^{-x}} \tag{21}$$

(4) Sample the system input value $r(k)$ and the actual output value $y(k)$ at the current k moments, the following values are calculated in turn.

The inputs to the input layer are:

$$O_j^{(1)}(k) = x(j) \quad j = 1, 2, \ldots, M \tag{22}$$

The inputs to the hidden layer are:

$$net_i^{(2)}(k) = \sum_{j=1}^{M} w_{ij}^{(2)} O_j^{(1)}(k) \tag{23}$$

The outputs to the hidden layer are:

$$O_i^{(2)}(k) = f\left(net_i^{(2)}(k)\right) \quad i = 1, 2, \ldots, Q \tag{24}$$

The inputs to the output layer are:

$$net_l^{(3)}(k) = \sum_{i=1}^{Q} w_{li}^{(3)} O_i^{(2)}(k) \tag{25}$$

The outputs to the output layer are:

$$O_l^{(3)}(k) = g\left(net_l^{(3)}(k)\right) \quad l = 1, 2, 3 \tag{26}$$

where, $w_{ij}^{(2)}$ is the weighting factor of hidden layer. The superscripts (1), (2), and (3) represent the input layer, hidden layer, and output layer, respectively.

(5) Select the loss function $E(k)$.

$$E(k) = \frac{1}{2}[r(k) - y(k)]^2 \tag{27}$$

(6) According to the negative gradient rule, the adjustment of each weight from hidden layer to output layer can be expressed as follows.

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} \tag{28}$$

where η is the learning rate, and $\frac{\partial E(k)}{\partial w_{li}^{(3)}}$ is:

$$\frac{\partial E(k)}{\partial w_{li}^{(3)}} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial net_l^{(3)}(k)} \cdot \frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}(k)} \tag{29}$$

From the previous equation, we can get the following result.

$$\begin{cases} \frac{\partial u(k)}{\partial O_1^{(3)}(k)} = e(k) - e(k-1) \\ \frac{\partial u(k)}{\partial O_2^{(3)}(k)} = e(k) \\ \frac{\partial u(k)}{\partial O_3^{(3)}(k)} = e(k) - 2e(k-1) + e(k-2) \end{cases} \tag{30}$$

Since the $\frac{\partial y(k)}{\partial u(k)}$ is unknown, it can be approximated by a sign function $\operatorname{sgn} \frac{\partial y(k)}{\partial u(k)}$. But the value of the sign function depends on whether the inner function is positive or negative, so this result can also be replaced by $\operatorname{sgn}\left(\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right)$. The inaccuracy effect of the approximation result can be compensated by adjusting the learning rate.

By simplification and approximation, the final amount of change in the connection weights of the output layer after learning is obtained.

$$\Delta w_{li}^{(3)}(k) = \eta \cdot e(k) \cdot \operatorname{sgn}\left(\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right) \cdot g'\left(net_l^{(3)}(k)\right) \cdot O_i^{(2)}(k) \tag{31}$$

Similarly, the amount of change in the connection weights of the hidden layer after learning is:

$$\Delta w_{ij}^{(2)}(k) = \eta \cdot f'\left(net_i^{(2)}(k)\right) \cdot \left(\sum_{l=1}^{3} \delta w_{li}^{(3)}(k)\right) \cdot O_j^{(1)}(k) \tag{32}$$

$$\delta = e(k) \cdot \operatorname{sgn}\left(\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right) \cdot g'\left(net_l^{(3)}(k)\right) \tag{33}$$

In order to avoid "local minima", the inertia variable α can be introduced, which serves to remember the direction of change of the connection weights at the previous moment. In this way, the "inertia effect" can be used to suppress possible oscillations in the network training and provide a buffering and smoothing effect. Ultimately, the two changes are shown below.

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \alpha \Delta w_{li}^{(3)}(k-1) \tag{34}$$

$$\Delta w_{ij}^{(2)}(k) = -\eta \frac{\partial E(k)}{\partial w_{ij}^{(2)}} + \alpha \Delta w_{ij}^{(2)}(k-1) \tag{35}$$

(7) When the loss function $E(k)$ is less than the set value, it means that the three parameters of the network output have satisfied the PID controller requirements, and the learning is finished. Conversely, let $k = k + 1$ and return to (4) to continue learning.

4.4 Summary

This chapter first analyzes the basic principle of the conventional PID controller, the role of the three PID parameters and their existence in the control process of parameter rectification. Then two digital PID control algorithms are introduced, namely, the position PID control and the incremental PID control. Finally, the design of the PID controller based on BP neural network is studied. The structure of the neural network is determined, including the determination of the number of neurons in the three layers, the determination of the initial values of the weights and the selection of the activate function. On this basis, inertia coefficients are introduced to avoid local optimization. In the next chapter, the ANPID designed in this chapter is applied to experiments, and the advantages of ANPID are demonstrated by experimental results.

Chapter 5

Experiment

In the previous chapters, we have introduced pose estimation and PID controller. In this chapter, the effectiveness of the designed PID is evaluated by experimental results. The first part of this chapter focuses on the experimental equipment used to do this experiment, which consists of three parts: Fanuc M-20iA robot, Creaform C-Track 780 and a computer with external control software. The remainder of this chapter presents the line tracking experiments and analyzes the results of the obtained experiments.

5.1 Experimental Setup

5.1.1 Robot

In this research, the definition and relationship of the above coordinate system is shown in Figure 5.1. The C-Track corresponds to frame S and the Fanuc end-effector corresponds to frame E. The pose in the sensor frame is represented by the homogeneous transformation ${}^{S}_{E}H$. The homogeneous transform matrix ${}^{B}_{S}H$ from robot frame B to C-Track frame S can be obtained. In order to use real-time pose correction, a frame W parallel to the robot frame B needs to be set up. Therefore, to obtain the pose ${}^{B}_{E}H$ in the robot frame, the following equation is used.

$$ {}^{B}_{E}H = {}^{B}_{S}H \times {}^{S}_{E}H \tag{36} $$

Because the frame W does not need to coincide with the frame B, only the rotation matrix is considered.

$$ {}^{B}_{E}R = {}^{B}_{S}R \times {}^{S}_{E}R \tag{37} $$

In this case, the ${}^{B}_{S}R$ is obtained by following steps:

(1) Select a point in the robot work space as the initial point ${}^{S}A_0$ which is measured in the C-Track frame S,

(2) Move the end-effector in the x-axis direction in the robot frame B to a point ${}^{S}A_1$ which is also measured in the frame S,

(3) Move the end-effector in the y-axis direction in the robot frame B to a point ${}^{S}A_2$,

(4) Calculate the unit vectors of the robot frame:

$$ {}^{S}\hat{x} = \frac{{}^{S}A_1 - {}^{S}A_0}{\|{}^{S}A_1 - {}^{S}A_0\|} \tag{38} $$

$$ {}^{S}\hat{y} = \frac{{}^{S}A_2 - {}^{S}A_1}{\|{}^{S}A_2 - {}^{S}A_1\|} \tag{39} $$

$$ {}^{S}\hat{z} = \frac{{}^{S}\hat{x} \times {}^{S}\hat{y}}{\|{}^{S}\hat{x} \times {}^{S}\hat{y}\|} \tag{40} $$

(5) Get the rotation matrix from frame S to frame B:

$$ {}^{B}_{S}R = \begin{bmatrix} {}^{S}\hat{x} & {}^{S}\hat{y} & {}^{S}\hat{z} \end{bmatrix}^{T} \tag{41} $$

(6) Obtain the ${}^{B}_{E}R$ by measuring the end-effector pose from robot sensor.

Fanuc M-20iA is a 6-DOF serial robot as shown in Figure 5.1. The whole system consists of robot, R-30iB controller and teach pendant. The role of the teach pendant is to program online. The role of the controller is to control the robot operation and provide API interface to implement the DPM functions mentioned in Chapter1. For safety reasons, the lab robot is placed in a safety fence and experimental programs can only be run when the gate to the fence is closed [61].

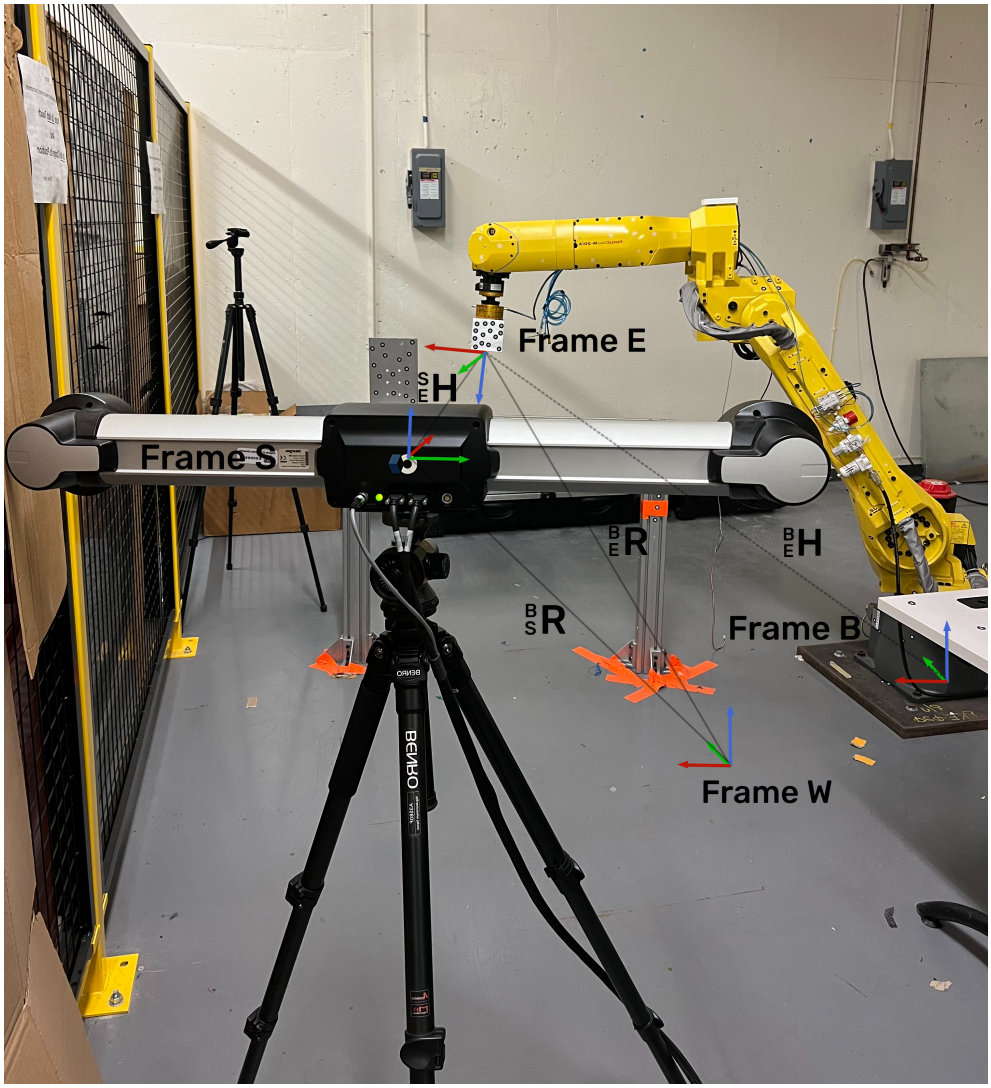


Figure 5.1: Definition and relation of the coordinate reference frames in the workspace

5.1.2 C-Track

Creaform C-Track 780, shown in Figure 5.2, is a dual camera that can provide dynamic tracking capabilities. It can measure the position and orientation of the object with reflective stickers. In addition to the basic tracking function, C-Track can also perform model inspecting and scanning, corresponding to the VXinspect and VXscan modules, respectively. All these functional modules, including VXtrack, are integrated in a software called VXelements developed by Creaform [62]. The software also provides an API for communication with other software.

Figure 5.2: C-Track 780

As a portable stereo camera and optical Coordinate Measuring Machine (CMM), C-Track can achieve 0.065mm volumetric accuracy and 0.0025mm repeatability. Since it is a precision equipment, C-Track needs to be used in strict accordance with the following steps:

(1) Calibration of the accuracy of C-Track with the attached calibration bar. Since the bar also comes with reflective stickers, people should avoid covering the stickers during the calibration process,

(2) Apply the reflection sticker to the object to be measured. This experiment requires reflection stickers to be placed on the end-effector of the robot and on the obejet,

(3) Place the C-Track at a proper distance from the target where all stickers are within a 3.8 m^3 volume,

(4) Define the tracking model and reference target in VXtrack software. In this experiment, the tracking model is the end-effector and the reference target is the aluminum alloy frame.

(5) Start tracking.

5.1.3 Software General Setup

To enable intercommunication between the robot, sensors and external controllers, it is necessary to develop a software that can collect and process data in real time. This software is designed to realize the following functions: remote control the robot so that it can automatically run the programs stored in the system, remote control C-Track and collect pose information from it in real time. The offset of the DPM is calculated by using the control algorithm proposed in Chapter 4. All of the above devices communicate using Ethernet cables and the TCP/IP protocol. The software was developed by the .NET framework and the main program was written in C#. The interface is shown in Figure 5.3. As seen in the figure, it is divided into three modules, which are Robot, C-Track and Real Time Control.

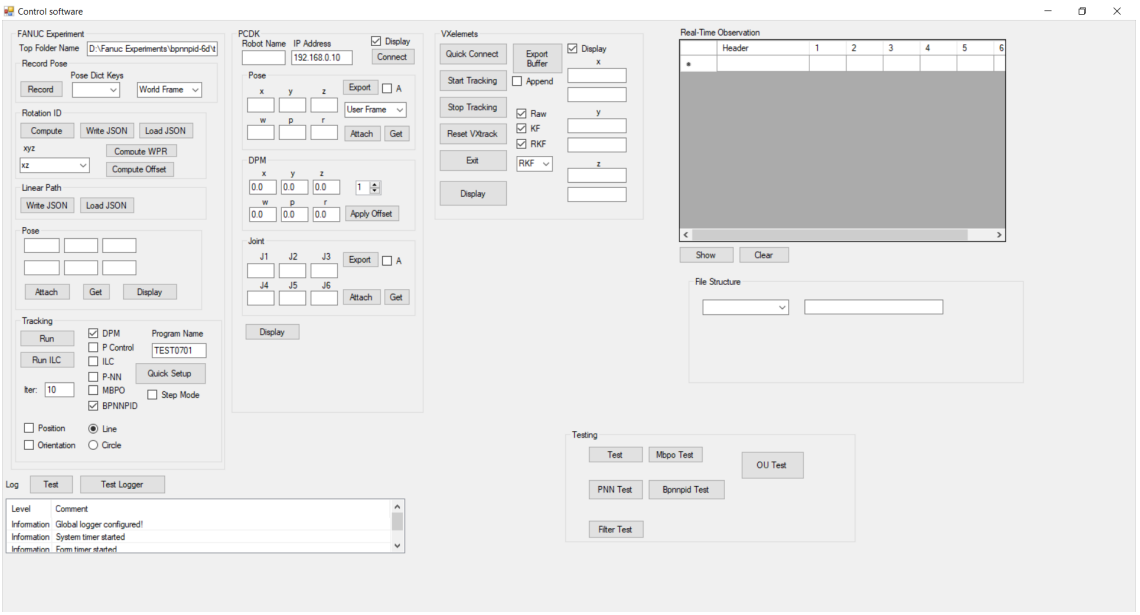


Figure 5.3: Interface of control software

The functions of the Robot module are to connect to the robot, get the pose information and joint position information in real time, run the TP program and send DPM offsets.

In the C-Track module, the software can connect to VXelements API. After extracting the tracking information of the model, it can process the data and save it. The button “Quick Connect” includes following steps.

(1) Connect to VXelements,

(2) Import reference targets,

(3) Import model targets.

When tracking begins, C-Track will start collecting data at a sampling rate of 29Hz. At the same time, the software also performs robust Kalman filtering in order to eliminate the noise from the measurement signal.

The real-time control module contains the following functions: calculate the transformation between robot frame and sensor frame, calculate the DPM offset using the designed control algorithm, and transmit the offset to the robot at fixed time intervals. According to Nyquist-Shannon theorem [63], the original continuous signal can be completely reconstructed from the sampled samples only if the sampling frequency is higher than twice the signal bandwidth. Therefore, the time interval of 80ms is appropriately chosen, which is more than twice the signal sampling time of C-Track.

5.1.4 RKF and Controller Parameters Setting

Before the experiment, the parameters of the RKF and the initial values of the ANPID controller variables need to be determined. Like Chapter 3, the parameters of RKF is the same with Table 3.2.

As for ANPID, the number of output neurons is chosen to be three, denoted by K_p, K_i and K_d. The input layer has four neurons corresponding to the reference input $r(t)$, the actual output $y(t)$, the error variable $e(t)$ and the bias term. In order to reduce the system complexity and shorten the system learning time, the number of hidden layer neurons is chosen to be five.

In the case where the activation function is sigmoid function, the initial weights are usually set to a random number between -1 and 1. However, if the initial weights are too high, the net input to the neuron may be large. When the weights are adjusted during online adaptation, the values of the weights becomes small, which can make the training time of the network longer. If the net input of each neuron is around the zero value, then the initial learning of the network will be fast regardless of the input. Therefore, the initial values of the weights are chosen as random numbers between -0.5 and 0.5.

The learning rate in the experiment is chosen to be 0.07, because too large value will cause oscillations and too small one will make the convergence time longer. Meanwhile, in order to improve

the convergence speed of BP algorithm in learning and to obtain better dynamic performance, the inertia factor is introduced and 0.04 is chosen as its value. The above parameters are shown in Table 5.1.

Table 5.1: ANPID parameters Setting

Parameters	**Value**
Number of input layer neurons	4
Number of hidden layer neurons	5
Number of output layer neurons	3
Learning rate	0.07
Inertia factor	0.04
Range of Initial weights	[-0.5, 0.5]
Range of Proportional gain (K_p)	[0, 0.01]
Range of Integral gain (K_i)	[0, 0.01]
Range of Derivative gain (K_d)	[0, 0.001]

5.2 Experiment: Line Following

After introducing the preparation of the experiment, the following describes the experiment. Figure 5.4 shows the layout of the experiment. As seen in the figure, the C-Track is at a certain distance from the robot so that it can cover the entire target in its field of view. Before the experiments, the preparation is divided into the following steps:

(1) Calibrate the C-Track with calibration bar,

(2) Configure VXtrack in the Vxelements,

(3) Generate TP program by online programming,

(4) Modify the parameters of ANPID control,

(5) Create a folder which can store the data and be read by control software.

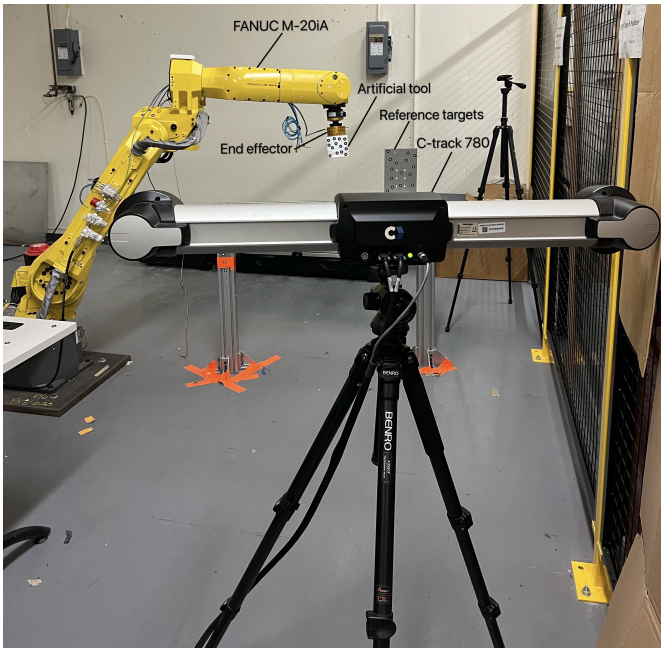


Figure 5.4: Layout of the experiment

When the setup is ready, the experiment begins. The experiment is divided into three main parts: the initialization part, the TP program part, the control loop part and the termination part. The experimental procedure is shown in Figure 5.5.

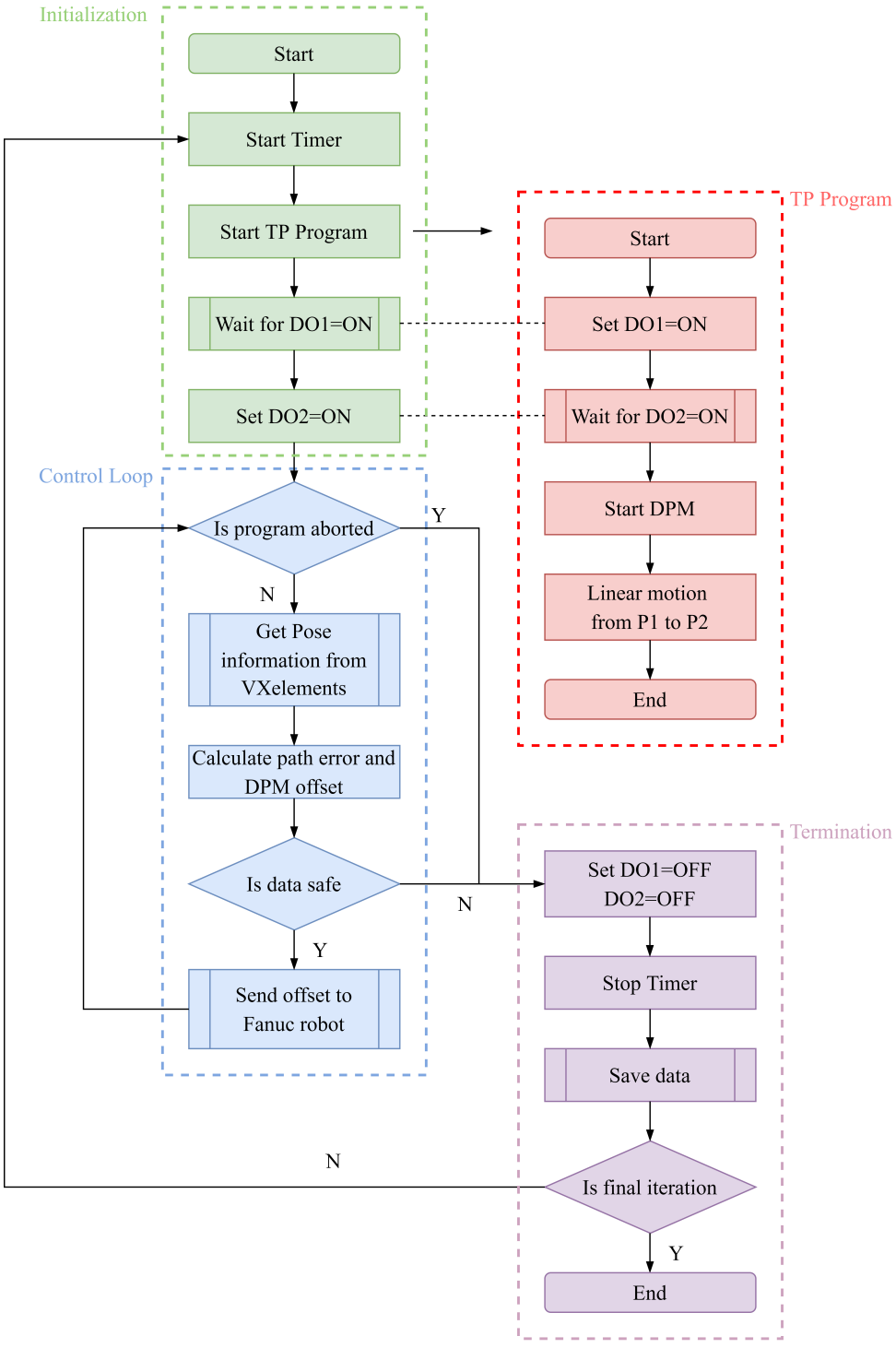


Figure 5.5: Experimental procedure

5.2.1 Experimental Results

The line following experiment is to define two points by TP program, and then the end-effector move in a straight line along the pre-defined two points, during which it maintains a constant orientation. Therefore, in the first experiment only the translation is considered, and the orientation is neglected. The definition of the coordinates of the two points and the setting of the parameters in the experiment come from the TP program and the control software, respectively.

In the TP program, the start point and end point are set in the teach mode, and then the motion rate is set as 25% which is means the speed is 25mm/s. The set parameters are shown in Table 5.2.

Table 5.2: Pose parameters of the TP program (Translation only)

TP Parameter	**Value**
Start point	[-371mm, 836mm, 715mm]
End point	[-476mm, 1386mm, 724mm]

In the control software, there are two controllers that need to be set. The first one is the baseline PID, which has the parameters shown in Table 5.3. Then the parameters of the designed ANPID are shown in Table 5.1.

Table 5.3: Baseline PID parameters (Translation only)

Coefficient	**Value**
Proportional gain (K_p)	diag(0.2, 0.2, 0.1)
Integral gain (K_i)	diag(0.005, 0.005, 0.005)
Derivative gain (K_d)	diag(0.05, 0.05, 0.05)

The second line following experiment is also defined two points, but its orientation changes with a constant rate. For comparison with the first translation-only experiment, the start point, end point and the speed are keep the same. The parameters are presented in Table 5.4. The parameters of baseline PID is also changed, which is shown in Table 5.5.

Table 5.4: Pose parameters of the TP program (Translation and Orientation)

TP Parameter	**Value**
Start point	[-371mm, 836mm, 715mm, $179°$, $-3.93°$, $89.2°$]
End point	[-476mm, 1386mm, 724mm, $171°$, $-12.8°$, $87.4°$]

Table 5.5: Baseline PID parameters (Translation and Orientation)

Coefficient	Value
Proportional gain (K_p)	diag(0.4, 0.3, 0.4, 17.19, 8.59, 25.78)
Integral gain (K_i)	diag(0.005, 0.005, 0.005, 1.15, 1.15, 1.15)
Derivative gain (K_d)	diag(0.0005, 0.0005, 0.0005, 0.2865, 0.2865, 0.2865)

Each experiment will be run twice, the first time to evaluate the performance of the baseline PID controller. To avoid chance, this run has 20 experimental episodes. The second run is to evaluate the performance of the ANPID controller, and the number of episodes is chosen to be 10, which is based on the results of the first experiment set evaluation.

5.2.2 Control Algorithm Performance Evaluation

The indexes based on the errors show the performance of the controller. Hence, the performance of the controller can be evaluated in two ways, one is the root mean square error (RMSE) and the other is the maximum value of the error. Based on the above evaluation methodology, there are three control configurations need to be assessed. The first group is the blank group, which means that there is no external controller, and the robot is only allowed to follow the TP program. The second group is the baseline PID controller only. The third group is ANPID control.

The first control experiment is to compare the first group with the second group. The results of the comparison of errors from x, y, z directions and 3D distances are shown in Figures 5.6. The comparison of orientation errors are shown in Figure 5.7. Referring to Figure 5.6, in the absence of an external controller, the path error becomes larger first, reaching a peak of 0.25 mm in the middle of the run. While in the presence of a baseline controller, the error maximum is around 0.1 mm.

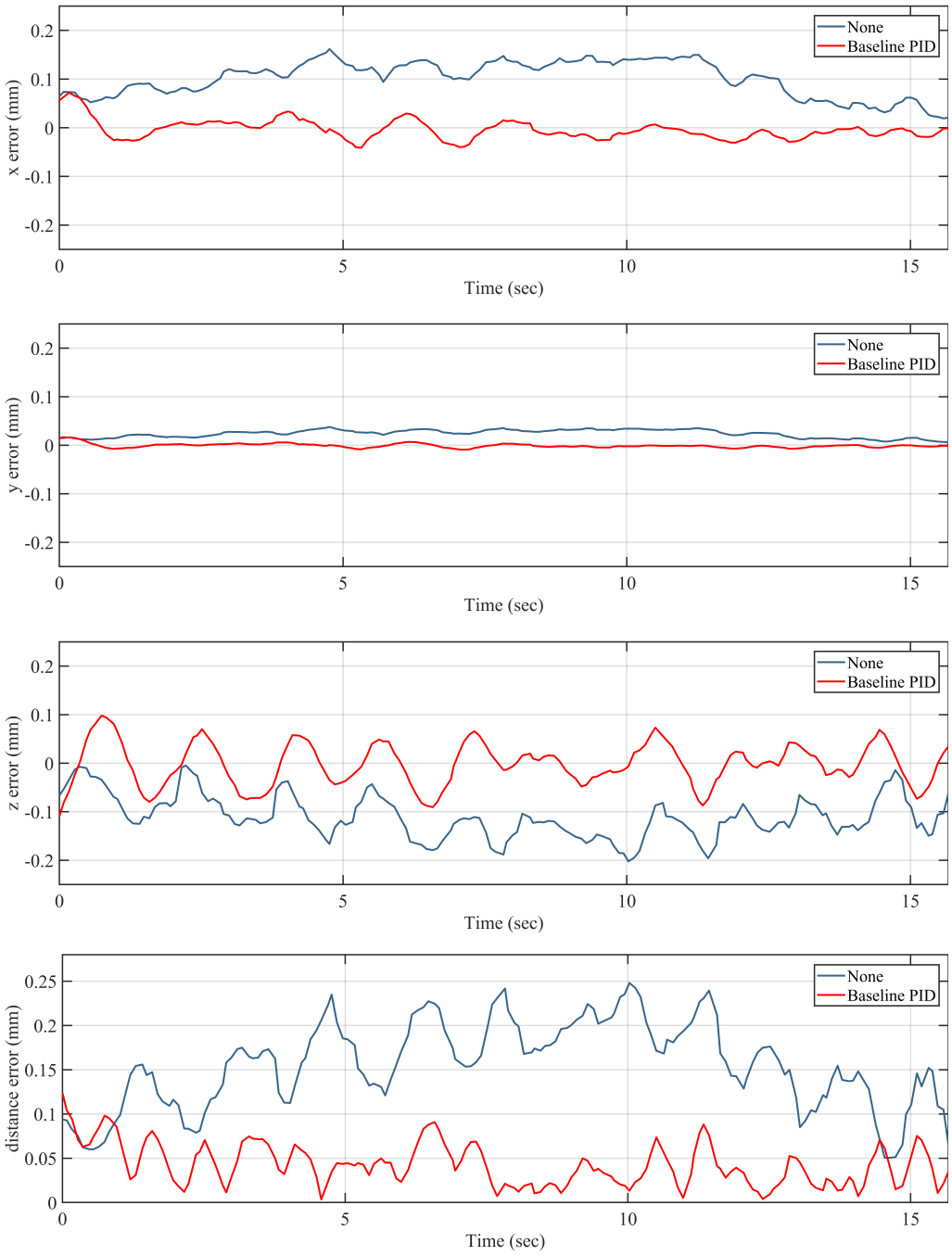
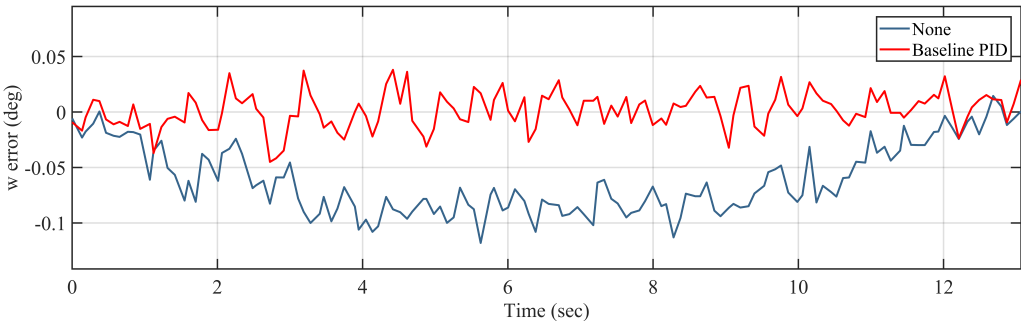
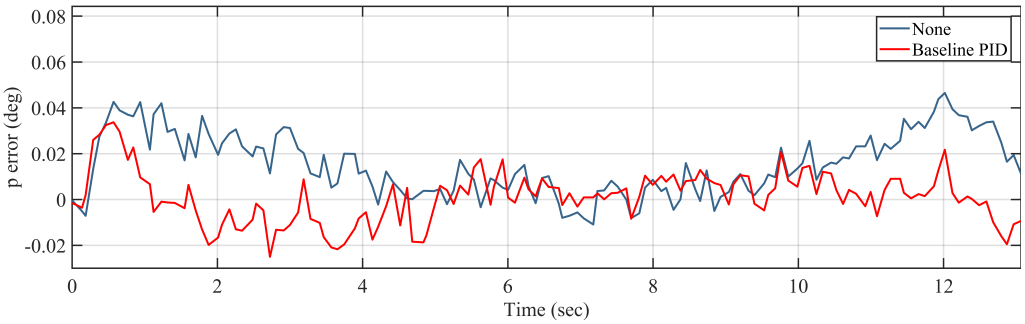


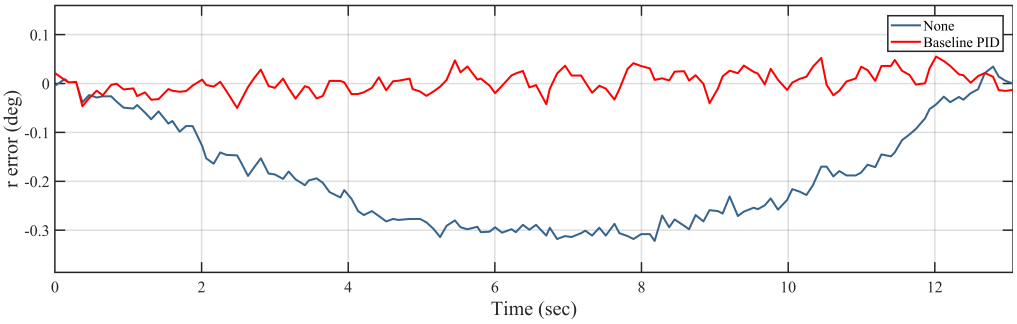
Figure 5.6: Comparison result of the path errors with and without baseline PID



(a)



(b)



(c)

Figure 5.7: Comparison result of the orientation errors with and without baseline PID (a) x-rotation (b) y-rotation (c) z-rotation

Figure 5.8 shows the maximum value of errors. This result is in agreement with that in the research of [1]. In Figure 5.7, the z-rotation errors reaches a peak of -0.3 deg in the middle of the run. In the presence of a baseline controller, the Figure 5.9 presents the maximum orientation errors are 0.04 deg, 0.03 deg and 0.06 deg, respectively.

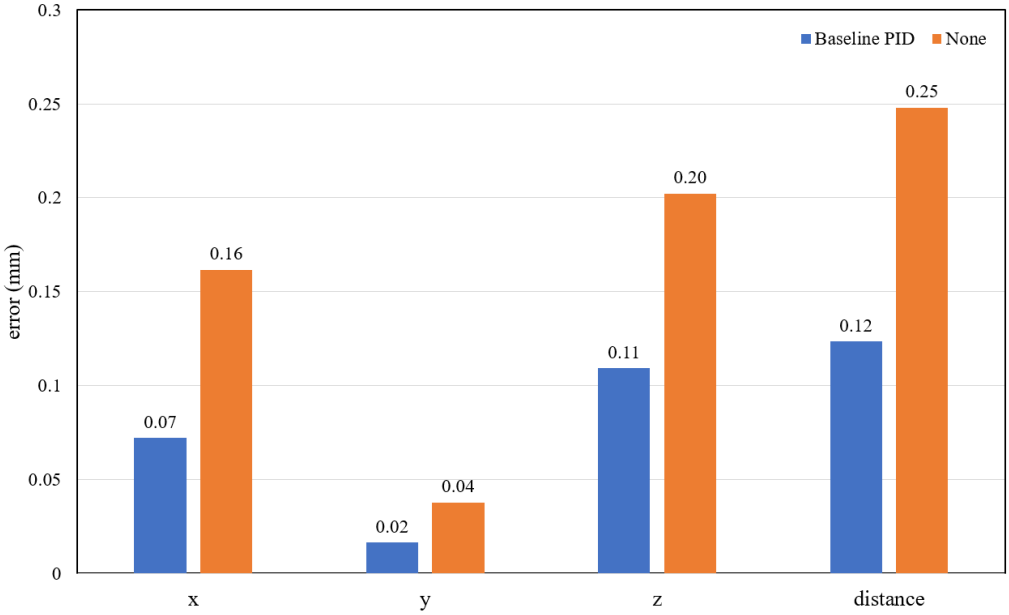


Figure 5.8: Comparison result of the maximum path errors with and without baseline PID

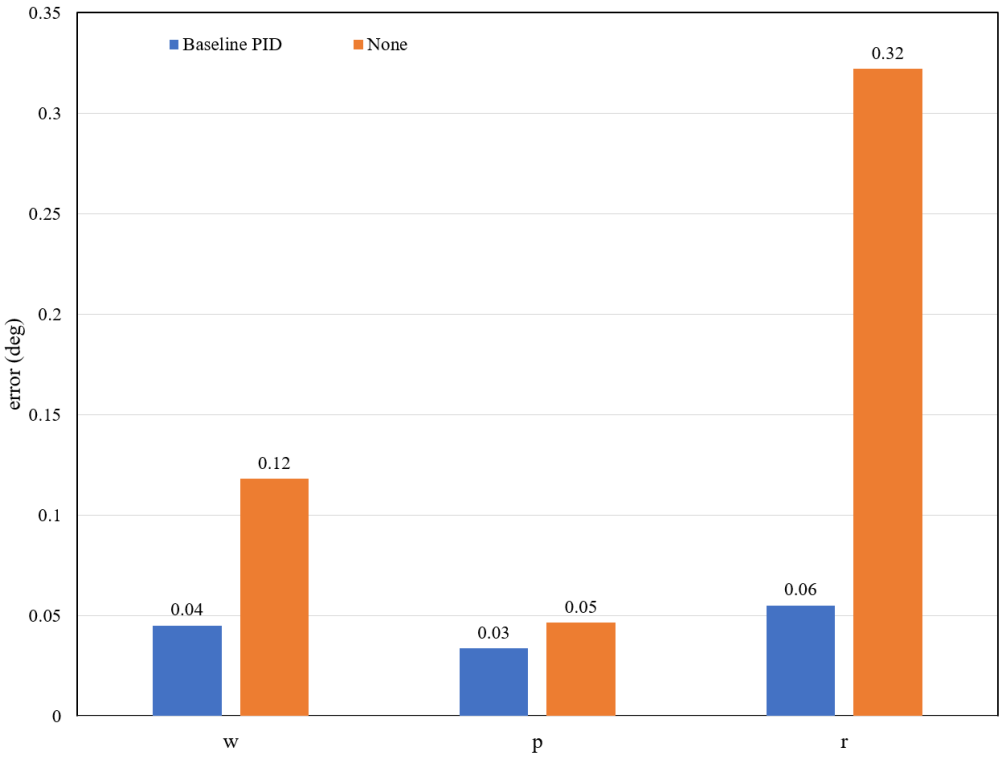


Figure 5.9: Comparison result of the maximum orientation errors with and without baseline PID

The second control experiment is designed to show the effect of ANPID. Figure 5.10 and Figure 5.11 show the error comparison results for this control group.

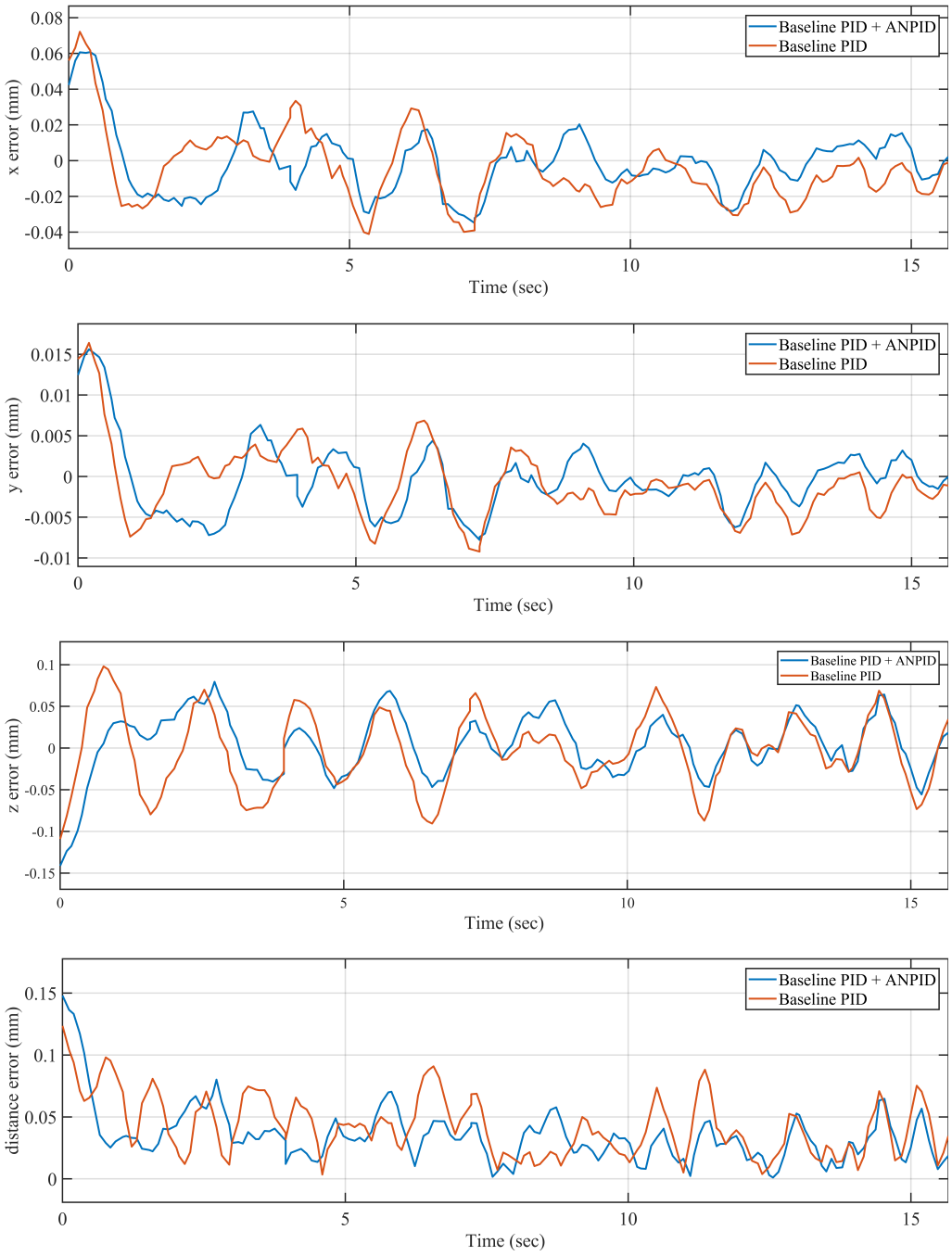
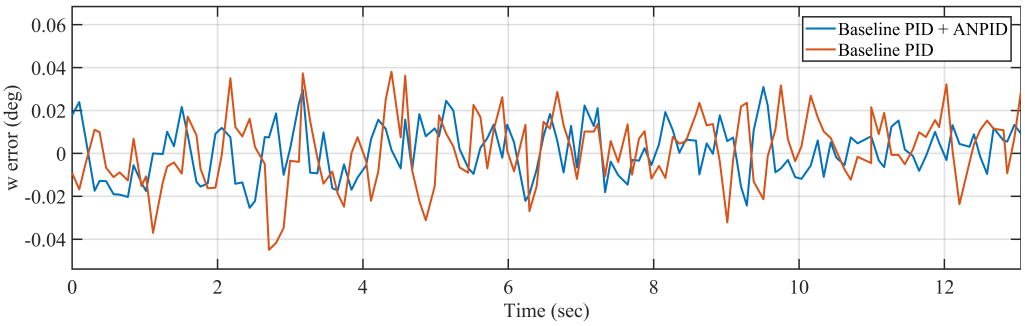
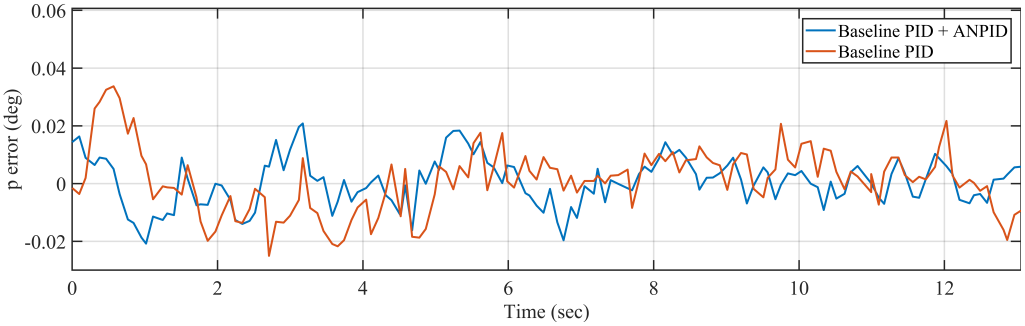


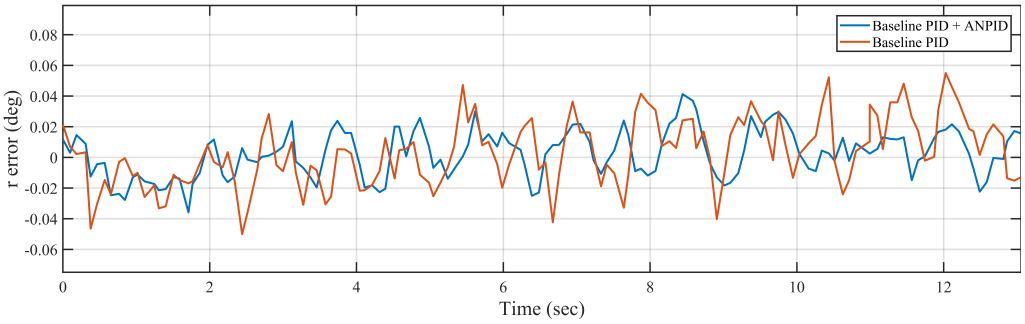
Figure 5.10: Comparison result of the path errors with and without ANPID



(a)



(b)



(c)

Figure 5.11: Comparison result of the orientation errors with and without ANPID (a) x-rotation (b) y-rotation (c) z-rotation

The errors are large until 5s because the robust filter has not yet converged at the beginning and the ANPID is still being adjusted. However, after 5s, the error of this group of paths with ANPID is significantly smaller than that of only baseline PID, and the maximum value of the path error is around 0.07mm. The maximum value of orientation errors are 0.031deg, 0.021deg and 0.041deg.

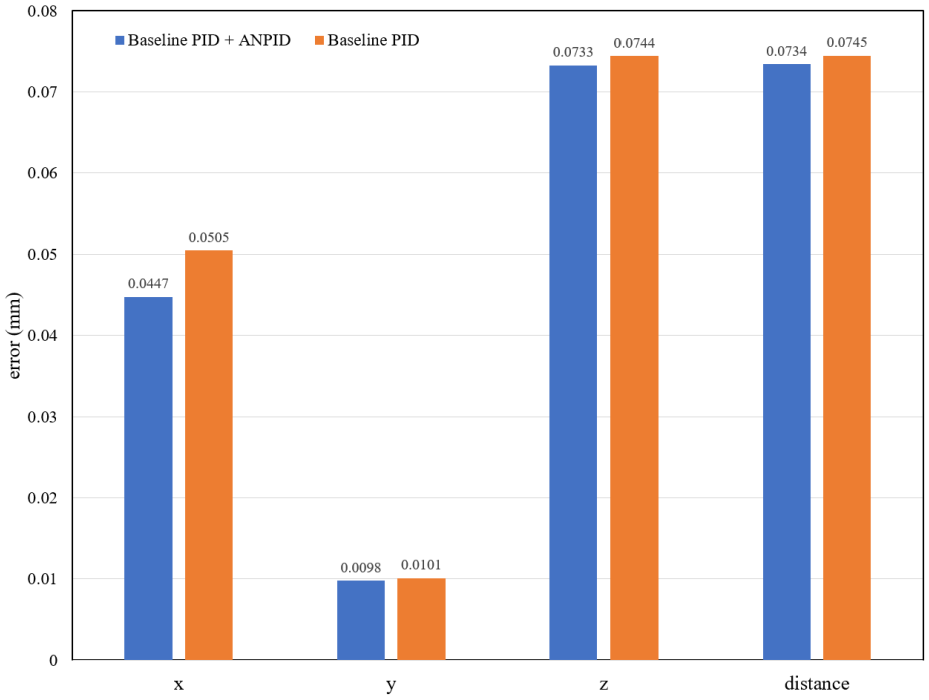


Figure 5.12: Comparison result of the maximum path errors with and without ANPID

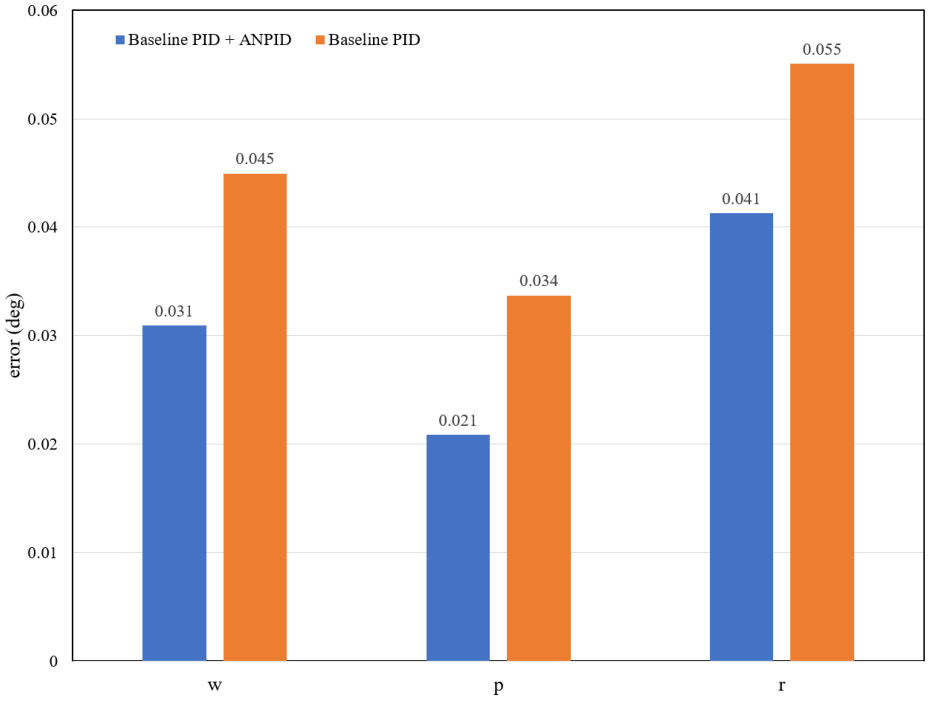


Figure 5.13: Comparison result of the maximum orientation errors with and without ANPID

Figure 5.12 and Figure 5.13 show the comparison of the maximum value of the errors after 5s. The group with ANPID has smaller errors in all six directions. The errors of ANPID are reduced by 11.37% in the x-direction, 2.97% in the y-direction, and 1.50% in the z-direction. Thus, in the 3D direction there is a 1.42% reduction. Also, the orientation errors are decreased by 31.17% in the x-rotation, 38.18% in the y-rotation and 25.04% in the z-rotation. This accuracy has been superior to the results of [1]. The contribution of this experiment is not only to demonstrate that ANPID is feasible, but also to optimize the previously studied control method. To further verify the enhancement brought by ANPID, Figure 5.14 and Figure 5.15 show the RMSE of this set of experiments. RMSE can reflect the precision of the measurement well because it is sensitive to the outliers of the data. As seen in Figures 5.14, the RMSE value for this set of data using ANPID is greatly reduced, by 13.42 %, 13.57 % and 6.79 % in the three directions, respectively. Ultimately, the RMSE is decreased by 8.35% in 3D distance, and in Figure 5.15, the RMSE is decreased by 23.15%, 25.21% and 29.30% in x-rotation, y-rotation and z-rotation, respectively.

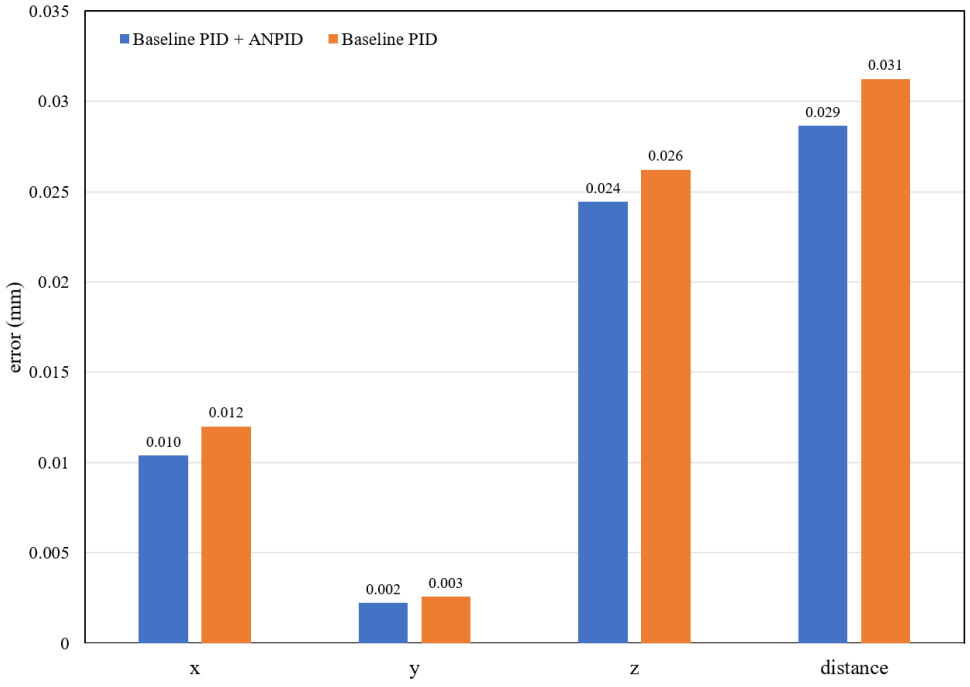


Figure 5.14: Comparison result of the path RMSE with and without ANPID

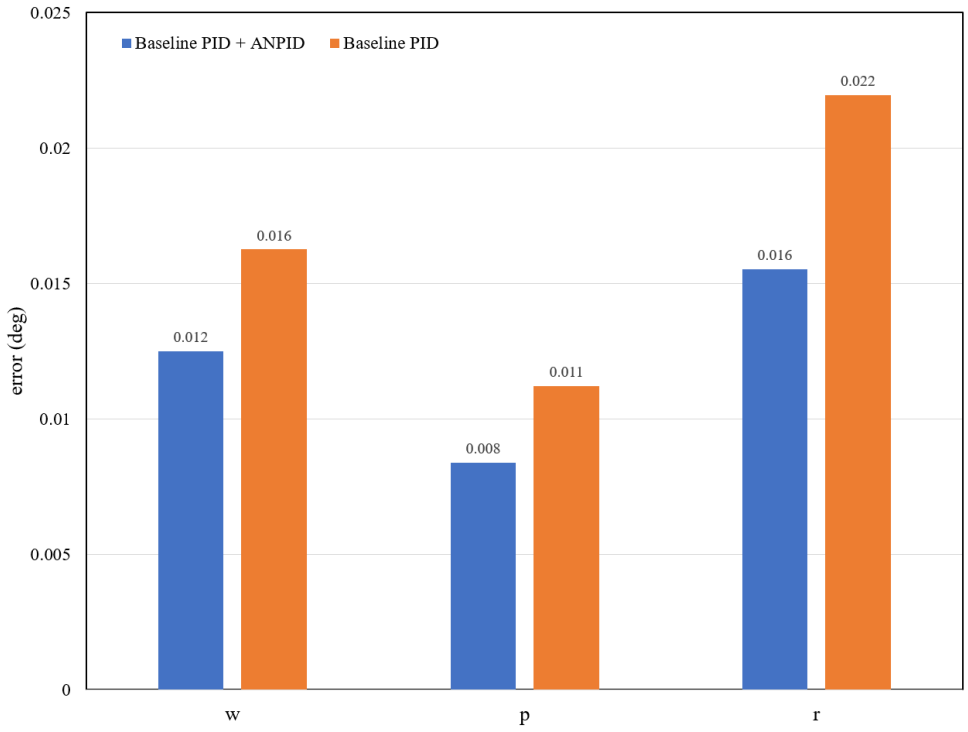


Figure 5.15: Comparison result of the orientation RMSE with and without ANPID

5.3 Summary

In this chapter, the configuration of the experiment is introduced. Then the steps of the experiment are shown in a flowchart, followed by the setting of the parameters required for the experiment. Finally, the performance of the controller is evaluated by two methods of evaluating the errors. Both evaluation methods proved that the performance of ANPID is the best among three group of controllers. Also, the industrial robot path following accuracy is improved to $\pm$0.08mm and $\pm$0.04deg by this method.

Chapter 6

Conclusion and Future Works

This chapter summarizes the research and the experimental results. In addition, improvements for future work are discussed.

6.1 Research Summary

This research presents a novel accurate dynamic path tracking method for Fanuc industrial robot (M-20iA) based on photogrammetry sensor (Creaform C-Track) and adaptive neuro-PID control method. In the experiment, the internal control system is not allowed to be modified by the user. Therefore, the motion path correction of the robot is required to use the API provided by the specified robot manufacturer. Also, a model-free control approach is also chosen in the design of the control algorithm, which reduces the control torque problem to a path error stabilization task.

When comparing the control algorithms, a baseline PID control and a complementary control based on it were used in the experiments. The complementary control is the ANPID. Baseline control was used to reduce most of the experimental error, a method that has also been demonstrated to work in previous research. The complementary ANPID control was used to further improve the dynamic accuracy.

The main contribution of the research are listed as follows:

- Using the photogrammetry sensor (C-Track) with RKF for a robust and highly accurate pose estimation,

- Developing a novel adaptive neuro-PID (ANPID) control method for accurate path following of industrial robot (Fanuc),
- Achieving a high accuracy which is up to ±0.08mm and ±0.04deg for the position for industrial robot (Fanuc).

6.2 Future Works

Combining neural networks with control algorithms is a promising direction. This research is an attempt, and the experimental results show that it is feasible to apply the neural network-based control algorithm to practice. Systems in real environments have many uncertainties, so adding more layers and neurons to the neural network is a direction of improvement. In addition, the filter algorithm can be improved. Adaptive features can be added to the RKF to increase the performance of the filter. Apart from line path following, more path following tasks such as circle, other curve will be tested on the experimental setup. Also, the developed control strategy will be tested on the other industrial robots such as ABB, KUKA and Denso etc.

Bibliography

[1] T.-T. Shu, S. Gharaaty, W.-F. Xie, A. Joubair, and I. A. Bonev, "Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1159–1170, 2018.

[2] C. Wong, C. Mineo, E. Yang, X.-T. Yan, and D. Gu, "A novel clustering-based algorithm for solving spatially constrained robotic task sequencing problems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2294–2305, 2020.

[3] M. Abderrahim, A. Khamis, S. Garrido, L. Moreno, and L. K. Huat, "Accuracy and calibration issues of industrial manipulators, industrial robotics: Programming, simulation and application," *InTech*, 2006.

[4] N. Holt, "Automotive robots reach for the sky," 2007. [Online]. Available: `https://www.automotivemanufacturingsolutions.com/automotive-robots-reach-for-the-sky/6495.article`.

[5] Z. Roth, B. Mooring, and B. Ravani, "An overview of robot calibration," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 1987.

[6] M. Ikits and J. M. Hollerbach, "Kinematic calibration using a plane constraint," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, pp. 3191–3196, IEEE, 1997.

[7] W. Veitschegger and C.-h. Wu, "A method for calibrating and compensating robot kinematic errors," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 39–44, IEEE, 1987.

[8] H. Hage, P. Bidaud, and N. Jardin, "Practical consideration on the identification of the kinematic parameters of the stäubli tx90 robot," in *Proceedings of the 13th World Congress in Mechanism and Machine Science, Guanajuato, Mexique*, p. 43, 2011.

[9] D.-H. Park, J.-H. Kwon, and I.-J. Ha, "Novel position-based visual servoing approach to robust global stability under field-of-view constraint," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4735–4752, 2011.

[10] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[11] W. Rui, Q. Huawei, S. Yuyang, and L. Jianliang, "Calibration of cartesian robot based on machine vision," in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1103–1108, IEEE, 2017.

[12] C. Gonzalez, "*What's the Difference Between Industrial Robots?*," 2016. [Online]. Available: `https://www.machinedesign.com/markets/robotics/article/21835000/whats-the-difference-between-industrial-robots`.

[13] W. Khalil and E. Dombre, *Modeling identification and control of robots*. CRC Press, 2002.

[14] Fanuc, "*What is a Delta Robot?*," 2022. [Online]. Available: `https://robotsdoneright.com/Articles/what-is-a-delta-robot.html`.

[15] M. Wilson, *Implementation of robot systems: an introduction to robotics, automation, and successful systems integration in manufacturing*. Butterworth-Heinemann, 2014.

[16] Fanuc, "*Industrial Robots*," 2022. [Online]. Available: `https://robotsdoneright.com/Industrial-robots.html`.

[17] Fanuc, "*Fanuc America corporation*," 2022. [Online]. Available: `http://www.fanucamerica.com`.

[18] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks," *Pattern recognition*, vol. 5, no. 2, pp. 99–108, 1973.

[19] J. Hill, "Real time control of a robot with a mobile camera," in *9th Int. Symp. on Industrial Robots, 1979*, pp. 233–246, 1979.

[20] P. I. Corke *et al.*, *Visual Control of Robots: high-performance visual servoing*. Research Studies Press Taunton, UK, 1996.

[21] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 1, pp. 40–45, 1997.

[22] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3, pp. 2741–2746, IEEE, 2000.

[23] B. Ahmadi, W.-F. Xie, and E. Zakeri, "Robust cascade vision/force control of industrial robots utilizing continuous integral sliding-mode control method," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 524–536, 2021.

[24] Y. Zhang, C. Hua, and J. Qian, "Adaptive robust visual servoing/force control for robot manipulator with dead-zone input," *IEEE Access*, vol. 7, pp. 129627–129636, 2019.

[25] E. Cervera, F. Berry, and P. Martinet, "Is 3d useful in stereo visual control?," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2, pp. 1630–1635, IEEE, 2002.

[26] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[27] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[28] D. Fontanelli, D. Macii, and D. Petri, "Dynamic synchrophasor estimation using smoothed kalman filtering," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 1–6, IEEE, 2016.

[29] J. Khodaparast, "A review of dynamic phasor estimation by non-linear kalman filters," *IEEE Access*, 2022.

[30] E. Ivanjko and I. Petrovic, "Extended kalman filter based mobile robot pose tracking using occupancy grid maps," in *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No. 04CH37521)*, vol. 1, pp. 311–314, IEEE, 2004.

[31] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, 2017.

[32] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[33] M. Ficocelli and F. Janabi-Sharifi, "Adaptive filtering for pose estimation in visual servoing," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1, pp. 19–24, IEEE, 2001.

[34] E. Zakeri, S. A. Moezi, and M. Eghtesad, "Optimal interval type-2 fuzzy fractional order super twisting algorithm: A second order sliding mode controller for fully-actuated and under-actuated nonlinear systems," *ISA transactions*, vol. 85, pp. 13–32, 2019.

[35] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *2017 IEEE power & energy society general meeting*, pp. 1–5, IEEE, 2017.

[36] Y. Yang and W. Gao, "An optimal adaptive kalman filter," *Journal of Geodesy*, vol. 80, no. 4, pp. 177–183, 2006.

[37] X.-J. Sun, Y. Gao, Z.-L. Deng, C. Li, and J.-W. Wang, "Multi-model information fusion kalman filtering and white noise deconvolution," *Information Fusion*, vol. 11, no. 2, pp. 163–173, 2010.

[38] W. S. McClulloch and W. Pitts, "A logical calculus of the ideas immanent in neurons activity," *Bulletin of mathematical biophysics*, vol. 5, no. 115-133, p. 10, 1943.

[39] D. Hebb, "The organization of behavior. a neuropsychological theory," 1949.

[40] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[41] M. Minsky and S. Papert, "An introduction to computational geometry," *Cambridge tiass., HIT*, vol. 479, p. 480, 1969.

[42] S. Grossberg, "Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors," *Biological cybernetics*, vol. 23, no. 3, pp. 121–134, 1976.

[43] S. Grossberg, "Adaptive pattern classification and universal recoding: Ii. feedback, expectation, olfaction, illusions," *Biological cybernetics*, vol. 23, no. 4, pp. 187–202, 1976.

[44] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[45] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[46] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[47] J. L. McClelland, D. E. Rumelhart, P. R. Group, *et al.*, *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*, vol. 2. 1987.

[48] S. Yanagawa and I. Miki, "Pid auto-tuning controller using a single neuron for dc servomotor," in *[1992] Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 277–280, IEEE, 1992.

[49] Y. Zhang, C. Yingliu, X. Song, and Z. Yan, "Application of rbf neural network pid controller in the rectification column temperature control system," in *2013 Sixth International Symposium on Computational Intelligence and Design*, vol. 2, pp. 72–75, IEEE, 2013.

[50] M. P. Belov, D. D. Truong, and P. Van Tuan, "Self-tuning pid controller using a neural network for nonlinear exoskeleton system," in *2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT)*, pp. 6–9, IEEE, 2021.

[51] R. Wang, Z. Zhou, and G. Qu, "Fuzzy neural network pid control based on rbf neural network for variable configuration spacecraft," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 1203–1207, IEEE, 2018.

[52] J. Wang, Y. Zhu, R. Qi, X. Zheng, and W. Li, "Adaptive pid control of multi-dof industrial robot based on neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 12, pp. 6249–6260, 2020.

[53] F. Abdelhedi, Y. Bouteraa, A. Chemori, and N. Derbel, "Nonlinear pid and feedforward control of robotic manipulators," in *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 349–354, IEEE, 2014.

[54] Y. Sun and H. Lin, "The research and application of adaptive pid controller based on neural network predictive model," in *2009 Fifth International Conference on Natural Computation*, vol. 2, pp. 454–459, IEEE, 2009.

[55] D. Zhao, C. Sun, Q. Wang, and W. Yang, "Neural network based pid control for quadrotor aircraft," in *International Conference on Intelligent Science and Big Data Engineering*, pp. 287–297, Springer, 2015.

[56] G. Pei, M. Yu, Y. Xu, C. Ma, H. Lai, F. Chen, and H. Lin, "An improved pid controller for the compliant constant-force actuator based on bp neural network and smith predictor," *Applied Sciences*, vol. 11, no. 6, p. 2685, 2021.

[57] J. G. Ziegler, "Optimum settings for automatic controllers," *trans. ASME*, vol. 65, pp. 433–444, 1943.

[58] K. J. Åström and T. Hägglund, "New tuning methods for pid controllers," in *European Control Conference, 1995*, 1995.

[59] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: Pid controller design," *Industrial & engineering chemistry process design and development*, vol. 25, no. 1, pp. 252–265, 1986.

[60] H. Wang, B. Chen, and C. Lin, "Adaptive neural control for strict-feedback stochastic nonlinear systems with time-delay," *Neurocomputing*, vol. 77, no. 1, pp. 267–274, 2012.

[61] J.-Y. Tang and W.-F. Xie, "Safe operating procedure for fanuc robot operation," 2022.

[62] Creaform, "*Creaform Inc.*," 2022. [Online]. Available: `https://www.creaform3d.com/en`.

[63] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.