Visual Servoing-Based Dynamic Accuracy Enhancement of Industrial Robots by Using Photogrammetry Sensor

Tingting Shu

A Thesis

in

The Department

of

Mechanical, Industrial & Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

December 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mrs. Tingting Shu**

Entitled: **Visual Servoing-Based Dynamic Accuracy Enhancement of Industrial Robots by Using Photogrammetry Sensor**

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

Doctor of Philosophy (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. Weiping Zhu

_____ External Examiner
Dr. Fengfeng (Jeff) Xi

_____ Examiner
Dr. Brandon Gordon

_____ Examiner
Dr. Youmin Zhang

_____ Supervisor
Dr. Wenfang Xie

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

_____ 2023

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

Abstract

Visual Servoing-Based Dynamic Accuracy Enhancement of Industrial Robots by Using Photogrammetry Sensor

Tingting Shu, Ph.D.

Concordia University, 2024

Industrial robots are defined as robot systems for manufacturing, with feature programmability, certain automation, and capability to move on several axes. However, the insufficient accuracy has limited the industrial robots to many potential applications in aerospace manufacturing. Typical accuracy requirement in aerospace manufacturing, such as drilling and fastening, are $\pm 0.20mm$ or less. Unfortunately, the discrepancy between a virtual-model robot and the corresponding real robot can even reach around $8 \sim 15mm$ due to the deflection of the mechanical structure and tolerances. Therefore, accuracy enhancement is highly significant in order to expand industrial robots to more applications in aerospace manufacturing. Visual servoing is extensively applied to control industrial robots with the help of the visual information feedback, especially for unmodeled environment. In recent decades, using visual servoing to reach the desired pose precisely has attracted the attention of many researchers. In this thesis research, three visual servoing-based control schemes are proposed to target at the accuracy enhancement of positioning and path tracking.

The research work in this thesis includes four parts. First, an adaptive Kalman filter (AKF) is developed to estimate the pose information of the objects in Cartesian space from the measurements of the visual sensor with noises. In order to address the difficulty in obtaining precise process error covariance and measurement error covariance, an adaptive algorithm is proposed to tune the covariance matrices so that the Kalman filter can produce synchronous pose estimations even when the objects are moving at certain acceleration or high speed. In this research, a photogrammetry sensor, C-Track 780 is selected as the visual sensor. The original measurement data from C-Track 780 are contaminated with the noises. The proposed AKF algorithm is developed to process the

measurement data from C-Track 780 to obtain smooth pose information for visual servoing.

Second, an effective dynamic pose correction (DPC) scheme for industrial robots is proposed to enhance the pose reaching accuracy for satisfying both position and orientation precision requirement. By applying the DPC scheme, the end-effector of an industrial robot can approach the poses in its reachable workspace with high accuracy. Some experiments are implemented on an industrial robot, FANUC M20-iA, by using C-Track 780. The experimental results demonstrate high pose accuracy ($\pm 0.050mm$ for position and $\pm 0.050deg$ for orientation).

In the third part, a practical dynamic path tracking (DPT) scheme for industrial robots is elaborated for improving the path tracking accuracy. The proposed DPT scheme is designed to realize 3D dynamic path tracking by correcting the robot movement in real time. By using the proposed DPT scheme, the industrial robot can be controlled to follow the pre-planned path with high tracking accuracy. The dynamic stability for the robot system with the proposed DPT scheme is proved theoretically through Lyapunov function. Moreover, the effectiveness of the proposed DPT scheme is verified by the experiments on FANUC M20-iA with C-Track 780. The experimental results show the high path tracking accuracy ($\pm 0.20mm$ for position and $\pm 0.10deg$ for orientation) is achieved.

In the last part, adaptive iterative learning control (AILC) in parallel with the proposed DPT scheme is proposed to update the time-varying control parameters along iteration axis and calculate new compensation to adjust the control inputs produced by the DPT module at each time interval based on the memorized data information and current feedback. Three experiments in different situations (without path correction, with DPT control, and with AILC control) are carried out for the comparison. The pose accuracy can be stably confined to less than $0.10mm$ for position and $0.05deg$ for orientation. Moreover, the repetitive disturbances can be also overcome within certain iterations so that the vibrations can be significantly reduced. Therefore, the AILC algorithm proposed verified to be effective to further improve the DPT scheme.

The research work in this thesis explores various schemes to enhance the positioning and path tracking accuracies for 6-DOF industrial robots. The proposed schemes, DPC, DPT and AILC, are proved to be effective on some FANUC robots which can be representative in 6-DOF articulated industrial robots for manufacturing.

Acknowledgments

I would like to express my deepest gratitude to my supervisor Dr. Wenfang Xie for her continuous support during my research. Her help, patience and encouragement enabled me to complete this thesis. I benefited greatly from her guidance during the study in Concordia University. I wish to acknowledge the financial support funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by the Collaborative Research and Development (CRD) program, and in part by the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ). At the same time, huge thanks to our research partner Dr. Ilian A. Bonev, Dr. Ahmed Joubair from ETS (École de technologie supérieure) for providing lab equipments and significant suggestions. Moreover, I am deeply thankful to Dongdong Zheng, Pengcheng Li, Tao Zhou, Jianyu Tang, Sepehr Gharaaty, Xiaoyang Zhang, Xiaoming Zhang, for their collaboration in our research group. Additionally, I would like to thank all of my friends who have been supporting me go through the recent years. Last but not least, I sincerely thank my family for their support and understanding.

Contents

List of Figures

List of Tables

Nomenclature

$\mathcal{F}_B$	base frame of the robot
$\mathcal{F}_D$	dynamic referencing frame
$\mathcal{F}_E$	tool frame of the robot defined in $\mathcal{F}_{U_R}$
$\mathcal{F}_S$	sensor frame
$\mathcal{F}_{E'}$	tool frame of the robot defined in $\mathcal{F}_S$
$\mathcal{F}_{O_T}$	reference frame of the task object
$\mathcal{F}_{U_R}$	user frame of the robot
$\mathcal{F}_{U_T}$	user frame in task space
$\mathcal{F}_{U_{Eq}}$	equivalent user frame to $\mathcal{F}_{U_R}$ defined in $\mathcal{F}_S$
$\mathcal{W}_R$	workspace of the robot
$\mathcal{W}_T$	task space
${}^{S}_{E'}\mathbf{H}$	transformation matrix from $\mathcal{F}_S$ to $\mathcal{F}_{E'}$
${}^{U_R}_{E}\mathbf{H}$	transformation matrix from $\mathcal{F}_{U_R}$ to $\mathcal{F}_E$
${}^{U_T}_{E'}\mathbf{H}$	transformation matrix from $\mathcal{F}_{U_T}$ to $\mathcal{F}_{E'}$
${}^{U_T}_{E}\mathbf{R}$	rotation matrix from $\mathcal{F}_{U_T}$ to $\mathcal{F}_E$
${}^{U_T}_{O_T}\mathbf{H}$	transformation matrix from $\mathcal{F}_{U_T}$ to $\mathcal{F}_{O_T}$

${}^{U_T}_{S}\mathbf{H}$	transformation matrix from $\mathcal{F}_{U_T}$ to $\mathcal{F}_S$
${}^{E'}_{E}\mathbf{R}$	rotation matrix from $\mathcal{F}_{E'}$ to $\mathcal{F}_E$
AILC	adaptive iterative learning control
AKF	adaptive Kalman filter
CAD	computer-aided design
CMM	coordinate measure machine
DOF	degree of freedom
DPC	dynamic pose correction
DPT	dynamic path tracking
FOV	field of view
IBVS	image-based visual servoing
ILC	iterative learning control
PBVS	position-based visual servoing
TCP	tool center point

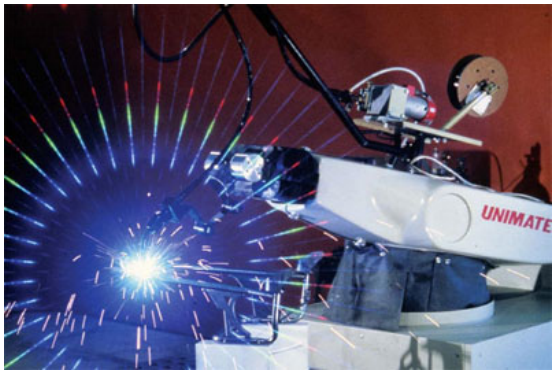
Chapter 1

Introduction

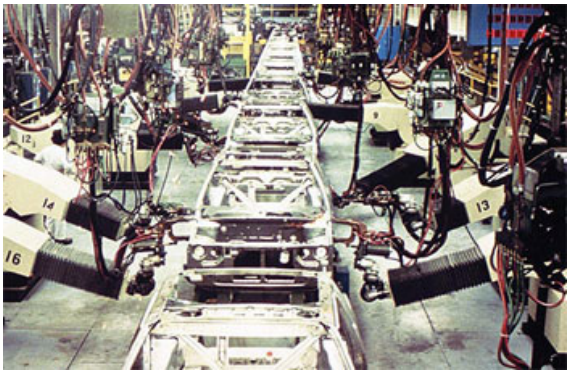
1.1 Background

According to the definition from ISO (the International Organization for Standardization), an industrial robot is a multipurpose manipulator with some actuators, featuring to be automatically controllable and programmable in three or more axes and be used for industrial automation applications [1]. The first industrial robot, Unimate in Fig.1.1a, was invented by George C. Devol in 1954 and patented in 1961 [2]. The typical application field of the industrial robots is automotive manufacturing, such as handling, welding, assembling, cleaning, dispensing, processing. With the help of Unimate spot welding robot shown in Fig.1.1b, the production capability of GE (General Motors) succeeded to reach 110 cars/hour, which was more than twice of the production speed of any other automotive company in 1969. Generally, industrial robots are considered as labor replacement of human to complete repetitive or dangerous tasks with high precision and endurance, for example, lifting very heavy objects, collecting/packaging radioactive waste, working in very high temperature, contaminated or dusty environments. As stated in the World Robotics 2021 Industrial Robots Report [3], the total of estimated 3 million industrial robots are operated in factories around the world with an increase of 10% even during the global pandemic in 2021. The benefits of industrial robots attract more and more manufacturers. The development and application of industrial robots in the most recent 10 years is astounding through the demonstration of Fig.1.2a and Fig.1.2b. The average global robot density is nearly 126 industrial robots per 10,000 employees in

the manufacturing industries according to the statistics in [3].

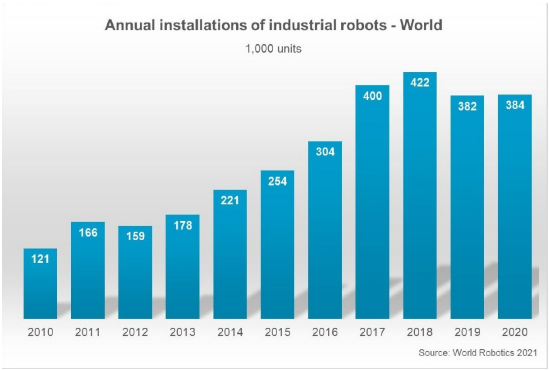


(a) The first industrial robot: Unimate

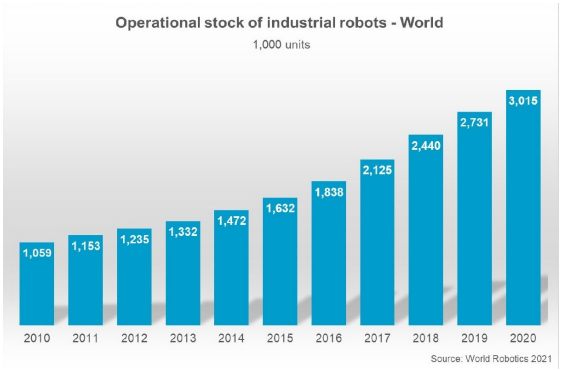


(b) Unimate spot welding robots in automated automotive manufacturing

Figure 1.1: The first industrial robots Unimate and its application in automotive manufacturing [4].



(a) Worldwide Annual Installations



(b) Worldwide Operational Stock

Figure 1.2: The development and application of industrial robots during 2010-2020 [3]

In recent years, the aerospace manufacturers, e.g. Boeing and Lockheed Martin, have been increasingly automating the manufacturing process by using industrial robots to replace human labor. The main applications in aerospace manufacturing are including drilling, fastening as shown in Fig.1.3a, welding, inspection as shown in Fig.1.3b, sealing, dispensing, and other rigid manufacturing. The fuselages of the modern large aircrafts are connected by using a lot of rivets since they are much tighter and stronger under vibrations and extreme temperature conditions comparing to the screws and welds. For example, there are nearly $60,000$ rivets on the fuselages of a Boeing 777 [5]. Each complete riveting process consists of drilling the hole, aligning, filling and fastening. If drilling a high-quality hole is performed by a skilled technician, several steps, including drilling a

pilot hole, drilling a full-size hole and reaming the hole, are required by using a complex heavy jig and powerful tools. It can be estimated how enormous amount of human labor needed to manually install all the rivets for a large aircraft. As a comparison, two industrial robots like Fig.1.3a can accomplish the drilling and riveting process with a higher precision in a fraction of the time that the skilled technicians spend, not to mention liberate the human from stress injuries.

(a) The industrial robots work inside and outside a 777 jet fuselage to set fasteners [6]

(b) Robot inspecting composites for irregularities, courtesy Spirit AeroSystems Inc. [7]

Figure 1.3: The aerospace applications of industrial robots

The benefits of automating the aerospace manufacturing with industrial robots have been extensively acknowledged. Meanwhile, there are still many potential applications in aerospace manufacturing fields, which are confined to be robotized due to the poor accuracy of the industrial robots. Generally, the manufacturer of the industrial robots only provide the repeatability as the performance criteria. Repeatability represents the ability of the industrial robot to consistently reach the same task point, and it is useful when the task point can be taught on-site. The repeatability offered by small or medium industrial robots varies in $0.03 \sim 0.10mm$ and can be bigger than $0.2mm$ for large ones [8]. However, teaching method becomes unfeasible for many applications in aerospace manufacturing because of some features of a large aircraft, such as large size, complex structure and high precision requirement. Moreover, it is very time-consuming to teaching a industrial robot to drill large numbers of holes on a large-scale aircraft. On the other hand, the advanced task planning methods based on offline simulation or computer aided design appears demanding, and the implementation of these methods will rely on the accuracy of the industrial robots. According to the standard process specifications in the aerospace industry, the desired accuracy of robot manipulation for manufacturing is around $\pm 0.20mm$ [9, 10]. Accordingly, a nominal accuracy of $\pm 0.10mm$

is selected to ensure robots to be able to complete these tasks [11]. Unfortunately, the discrepancy between a virtual-model robot and the corresponding real robot can even reach around $8 \sim 15mm$ due to the deflection of the mechanical structure and tolerances [12]. Therefore, enhancing the accuracy of industrial robots tends to be unavoidable as robotizing aerospace manufacturing has been extensively explored.

1.2 Motivation

Various complicated factors, as listed in Table.1.1, may affect the accuracy of industrial robots and can be classified into five categories: parametric, environmental, computational, measurement, and application [13]. In the past decades, many researchers have been trying to reduce or compensate the influence caused by these factors so as to improve the accuracy of robots.

Table 1.1: The factors to impair the accuracy of the industrial robots .

No.	Factor categories	Specific description
1	parametric	kinematic parameters deviation resulted from the assembly and manufacturing errors, influence of dynamic parameters, drive-train compliance, joint zero-reference offsets, friction, hysteresis, backlash and other nonlinearities
2	environmental	temperature changes, warm-up process
3	computational	computer round-off, steady-state servo errors
4	measurement	nonlinearity and resolution of joint position sensors
5	application	installation errors, workpiece position and geometry errors

The static calibration is a traditional method to enhance the accuracy of industrial robots by identifying the precise values of the kinematic parameters. Calibration is mostly based on kinematic models, or sometimes partly dynamic model involved as well, such as joint or link compliance, backlash [13, 14]. Since the static calibration considers only the static or quasi static factors, static pose accuracy can be increased to a limited extent by static calibration. Another calibration method proposed by Zhao et.al in [15] is to adjust the kinematic parameters for path tracking of industrial robots through iterative learning control method. However, the offsets of kinematic parameters will be variable with different paths and moving speeds. As a result, when dynamic positioning and path tracking accuracies of industrial robots are demanded, calibration can be only the first step to

achieve the reliable solution.

In recent years, majority of the robotic applications in aerospace industry take the direct method to install high-accuracy secondary encoders at the output of each joint axis [10, 16]. This method can improve the accuracy and stiffness to some extent through excluding the influences of the distortion and backlash from the reducers and gears. However, a complete set of high-accuracy secondary encoders for one industrial robot is very expensive. In addition, the encoders should be custom-implemented, and work only on a few large robot models. For instance, FANUC offers this option only for their R2000 and M900 robot models, at an extra cost of about $\$30,000$.

Visual servoing is known as a control method based on visual feedback information. Compared to the aforementioned two methods, visual servoing proves to be more economic, and effective to enhance the accuracy of an industrial robot through real-time complementary control to guide the end-effector with the help of a laser tracker or a close-range photogrammetry system [17, 18, 19, 20]. Visual servoing for industrial robots provides the possibility to correct the occurring errors for dynamic positioning and path tracking accuracy enhancement because it is a closed-loop real-time control method considering system dynamics. Although there have a lot of research work involving visual servoing, very limited existing works aimed to achieve the high-accuracy approaching of fixed points and the high-precision tracking of pre-planned trajectories for a complete 6-DOF industrial robot. Hence, the research on enhancing dynamic pose approaching or path tracking accuracy for industrial robots is very significant and promising. Accordingly, the main motivation of this thesis research work is to develop effective visual servoing based control methods to improve the accuracy of the industrial robots.

1.3 Problem Statement

In fact, a modern industrial robot is a complicated system, composed of at least five components: the manipulator, the end-effector, the robotic controller, feedback equipments, and the human-interface device. The majority of industrial robots currently available on the market are designed to work in conjunction with controllers specifically manufactured by the same company, ensuring seamless operation. The robotic controller can be considered as the ”brain” of an industrial

robot. All the motion commands to the actuators mounted on the axes of the manipulator and the end-effector are produced by the robotic controller based on the feedback information. The robotic controller provides access for the operators to control the movement of the end-effector. In another word, the manipulator can be commanded only through the robotic controller, which becomes a barrier when the accuracy of the industrial robot cannot not meet the task requirements and some complementary strategies need to be employed.

Generally, industrial robots are mass-produced with components that are less rigid and precise, and thus they cannot be as accurate as computer numerical control machine. As a result, the kinematic and dynamic models of an industrial robot that is programmed in the robotic controller can be an overly simplified version of reality. On the other hand, path tracking errors for high-speed or large-load robots are primarily induced by dynamic factors, including Coriolis force, centrifugal force, and dynamic coupling within the robot's joints [21, 22, 23]. Besides, there are many other factors as listed in Table.1.1 affecting the accuracy of industrial robots.

In fact, the accuracy of an industrial robot can be enhanced by minimizing the errors between the real robot and the mathematical representation of the virtual robot in the robotic controller. The goal of this thesis research work are to improve dynamic pose accuracy for approaching desired pose and dynamic tracking accuracy for tracking pre-planned path. One effective and affordable solution is to guide the end-effector with the feedback information of a close-range photogrammetry sensor based on visual servoing. In other words, an outer loop part, i.e., a visual servoing based controller, before the robotic controller will be designed for dynamic pose correcting (DPC) or dynamic path tracking (DPT).

1.4 Research Objectives and Scope

This thesis research work mainly aims to develop efficient DPC/DPT control algorithm to improve the pose reaching accuracy and dynamic path tracking accuracy [24, 25]. First, the DPC/DPT schemes are proposed and verified by practical experiments. Then, in order to improve the control process and further enhance the DPT accuracy, an adaptive iterative learning control scheme is designed to optimize the DPT scheme. At the same time, all the proposed control algorithms should

guarantee the stability of the system in all tasks. Additionally, the development of an adaptive Kalman filter (AKF) for noise filtering to obtain smooth pose estimation is significant.

The research work in this thesis is carried out in five main phases. First, relevant literature and the technical reports in the field are reviewed, which is summarized in the next chapter. Then, the development of the AKF is introduced and its effectiveness is verified. Next, the DPC scheme is presented to realize desired pose reaching by correcting the pose error of the end-effector in real time. The fourth phase is the development of the DPT scheme, which is presented to realize dynamic path tracking by on-line correcting the path-following movement of the end-effector. The DPT scheme includes path analyzer and path tracking control module. The path analyzer part computes the pose of the closest point on the desired task path according to the current TCP pose, and decides the next path step. The path tracking control module includes DPT control algorithm to produce the control input for robot controller based on the current tracking error. Last, an adaptive iterative learning control (AILC) is proposed to achieve higher accuracy and better transient performance based on the DPT scheme. The AILC scheme updates time-varing adaptive parameters and iteratively generates control inputs to offset the control inputs derived from DPT scheme. The AILC scheme is implemented as a memory-based controller, where the control signals and error signals of previous trials are utilized to update the current control signals.

1.5 Contributions

In this Ph.D research, accuracy enhancements for industrial robots to reach a given pose or track a given path are investigated. The contributions of this Ph.D research are listed as follows:

- A dynamic pose correction (DPC) control scheme is proposed which can guide a 6-DOF industrial robot to achieve high pose approaching accuracy. The desired pose can be static or moving. Some experiments are implemented on an industrial robot, FANUC LR Mate 200iC, by using C-Track 780. The experimental results demonstrates high pose accuracy ($\pm 0.050mm$ for position and $\pm 0.050deg$ for orientation).

- A dynamic path tracking (DPT) control scheme is developed to control the industrial robot to track the desired path in the workspace with high accuracy. The experimental results show

the high path tracking accuracy ($\pm 0.20mm$ for position and $\pm 0.10deg$ for orientation) is achieved when the end-effector of FANUC M20-iA is moving at speed $25mm/s$.

- An adaptive iterative learning (AILC) scheme based DPT control scheme is designed to further improve the control process of the proposed DPT control scheme. The AILC scheme is designed to update time-varying parameters along iteration axis without known system dynamics. Comparing with DPT control scheme, it is proved that better transient performance and higher path tracking accuracy can be achieved with the AILC based DPT scheme. The pose accuracy can be stably confined to less than $0.1mm$ for position and $0.05deg$ for orientation when the end-effector of FANUC M20-iA is moving at speed $50mm/s$. Moreover, the repetitive disturbances can be also overcome within certain iterations so that the vibrations can be significantly reduced. Therefore, the AILC algorithm is verified to be effective to further improve the DPT scheme.

- Seamlessly integrate the third control terminal with the industrial robot controller.

- A latest photogrammetry sensor, C-track 780 from Creaform, is used to provide the feedback measurements. It is proved to be an effective and economic coordinate measure machine (CMM) by the experimental results.

- In order to extract reliable pose information from the noise contaminated sampled data, an adaptive Kalman filter (AKF) is proposed. The AKF algorithm can be adaptive to variable velocity and acceleration for moving objects with irregular sampling interval.

1.6 Publications

The presented research work is documented in a number of journals and conference proceedings. The following is the list of author's publications.

Journal Publications:

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

- Gharaaty S., Shu T.T., Xie W.F., Joubair A. and Bonev I.A.,, I.A. (2018), **Online pose correction of an industrial robot using an optical coordinate measure machine system**, *International Journal of Advanced Robotic Systems, 15(4), 1729881418787915*. DOI: https://doi.org/10.11 77/1729881418787915.

- Li P.C., Shu T.T., Xie W.F., and Tian W., **Dynamic Visual Servoing of A 6-RSS Parallel Robot Based on Optical CMM**, *Journal of Intelligent & Robotic Systems*, 102, 40 (2021). https://doi.org/10.1007/s10846-021-01402-5.

- Zhang R.H., Wang Y.N., Xie W.F., Shu T.T., Tan H.R. and Jiang, Y.M., **Coordination control of the automated fiber placement system using photogrammetry-based leader–follower approach**, *Control Engineering Practice*, 141, 105691(2023), Elsevier, https://doi.org/10.1016/j.conengprac.2

- Shu T.T., Li P.C., Zhang R.H., and Xie W.F., **Adaptive Iterative Learning Control for Enhancing the Dynamic Path Tracking Accuracy of 6-DOF Industrial Robots**, submitted to *International Journal of Advanced Robotic Systems* in 2023.

Conference Publications:

- Shu T.T., Gharaaty S., Xie W.F., Joubair A. and Bonev I.A., **Dynamic path tracking of industrial robots with high accuracy by visual servoing**, *Proc. of 12th IEEE Conference on Industrial Electronics and Applications, June 18-21, Siem Reap, Cambodian*, 2017.

- Gharaaty S., Shu T.T., Xie W.F., Joubair A. and Bonev I.A., **Accuracy enhancement of industrial robots by on-line pose correction**, *Proc. of 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), pp. 214 – 220, June 16-18, Wuhan, China*, 2017.

- Tang J.Y., Zhou T., Zakeri E., Shu T.T and Xie, W.F., **Photogrammetry-based dynamic path tracking of industrial robots using adaptive neuro-PID control method and robust Kalman filter**, *IEEE–2023 9th International Conference on Automation, Robotics and Applications (ICARA 2023), Feb. 10-12, 2023, Abu Dhabi, United Arab Emirates.*

- Zhou T., Tang J.Y., Shu T.T and Xie, W.F., **Reinforcement Learning (RL)-Based Visual Servoing of an Industrial Robot**, *Proceedings of the Canadian Society for Mechanical*

Engineering International Congress, Computational Fluid Dynamics Canada International Congress CSME - CFD-SC2023, May 28-31, 2023, Sherbrooke, QC, Canada.

1.7 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, a comprehensive literature review including industrial robots, coordinate measure machine (CMM), Kalman Filter, visual servoing, and iterative learning control is presented. In Chapter 3, the development of AKF is introduced and pose estimation from the feedback measurements is verified. In Chapter 4, the DPC control scheme is proposed and tested on FANUC M20-iA. In Chapter 5, the DPT control scheme is developed and validated on FANUC M20-iA. Chapter 6 presents the AILC scheme in parallel with the DPT scheme and the comparison of experimental results is provided. In Chapter 7, the conclusions of the thesis and the future works based on current research are given.

Chapter 2

Literature Review

2.1 Introduction

The thesis research work involves industrial robots, coordinate measure machine, Kalman filter, visual servoing, and iterative learning control. In terms of strategies and other aspects, the following is a review of the current research status in related fields.

2.2 Industrial Robots

When the first robot manipulators were constructed in the 1960s, only carefully preplanned movements could be executed in structured environment [26]. In recent decades, with the development of artificial intelligence (AI), modern intelligent control theory, mechanics with versatile sensors, image precessing, computer processor and peripheral devices, and other interdisciplinary research, industrial robots have gradually dominated the manufacturing fields, such as semiconductors, automotive, metal and machinery, plastic and chemical products. Industrial robots are predominantly utilized for handling, assembling, welding, cleaning, processing and dispensing [3]. Undoubtedly, industrial robots have make significant contributions to the Fourth Industrial Revolution (FIR) on industrial automation [27].

The major market share of industrial robots is possessed by the top four manufacturers which

are FANUC, KUKA, ABB and Yaskawa Electric. All these four manufacturers have unique advantages to prevail over each other, and have been operating for over 40 years. The available industrial robots in the market can be classified seven types as described in Table.2.1 [28]. SCARA represents Selective Compliance Articulated Robot Arm or Selective Compliance Assembly Robot Arm. Rectangular, cylindrical, and spherical are three basic shapes for industrial robots. The common features of these three shapes are the symmetry of their structures such that their reachable workspace are symmetric.

Articulated robots are the most common structure of robotic manipulators and occupying over 50% of annual installations according to the demonstration in Fig.2.1 [29]. Normally, an articulated robot comprises of six serial connected links and joints. Each joint and link are capable of translating and rotating separately. The manipulation of articulated robots is very flexible and their movements imitate the human arm in some extent. Usually, joints 1-3 are like human arm, responsible for positioning the end-effector to a objective point in space, and joints 4-6 are like the wrist for orientating the end-effector. The typical sizes of available articulated robots in the market range among $0.5m \sim 3.5m$ [28].

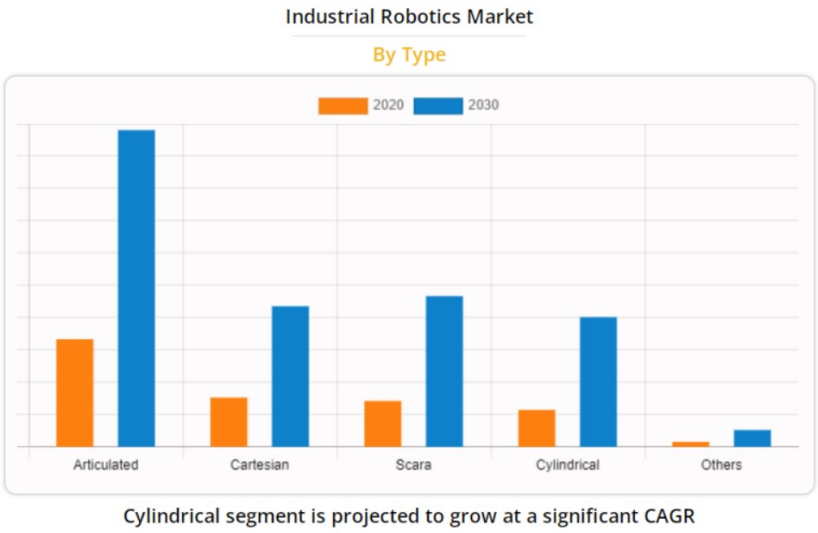


Figure 2.1: The industrial robotics market analysis by type in 2020 and forcast in 2030.

The important features, presented in subsection 2.2.1 are critical to an industrial robot and determine its capability and practical implementation.

Table 2.1: The types of the industrial robots and their features.

No.	types	stucture	movement features	applications
1	Cartisian robots	3 linear axes	3-DOF, move on 3-axis straight lines (in and out, up and down, side to side	CNC machines, 3D printing
2	SCARA Robots	pliable on X-Y axes and rigid on Z axis	4-DOF, move in X, Y, Z planes, rotate around Z axis	palletizing, assembly, bio-medical
3	articulated robots	mostly 4 or 6 axes	4-DOF or 6-DOF, rotating about the joints	welding, assembly, handling, packaging, machine tending
4	cylindrical Robots	1 rotary joint and 1 prismatic joint	2-DOF, rotating around the base and extending along another axis	coating, machine tending, simple assembly, in compact workspaces
5	parallel robots	3 joints and 3 arms on a same base	3-DOF, 3 arms control the 3 joints in a dome shape	pharmaceutical, electronic industries, fast pick and place food product
6	spherical robots	an arm with 2 rotary joints and 1 linear joint	3-DOF, move in spherical space	injection molding, welding, die casting, material handling
7	Cobots	many types	interact with humans in a shared space	palletizing, pick and place, machine tending, quality inspection

2.2.1 Repeatability and Accuracy

Repeatability and accuracy are two significant specifications for industrial robots. Fig.2.2 demonstrates the differences between the definitions of repeatability and accuracy.

Repeatability of an industrial robot refers to the repeated precision that the TCP (tool center point) of the end-effector return to the same location [30]. Repeatability is the result of statistics and represents the maximum radius of the sphere encompassing the set of the locations that the TCP of the end-effector arrives at starting from the same initial location with the identical configuration and payload. Repeatability is the capability to perform the repetitive tasks, such as machine loading

and assembly. Typical repeatability provided by the robot manufacturers varies from $0.005mm$ for precise micro-positioning manipulators to $1 \sim 2mm$ for large spot-welding manipulators [31].

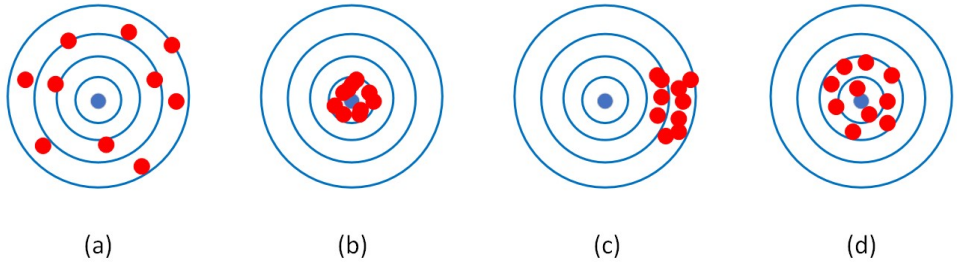


Figure 2.2: (a) low repeatability and accuracy, (b) high repeatability and accuracy, (c) high repeatability and low accuracy, (d) low repeatability and high accuracy.

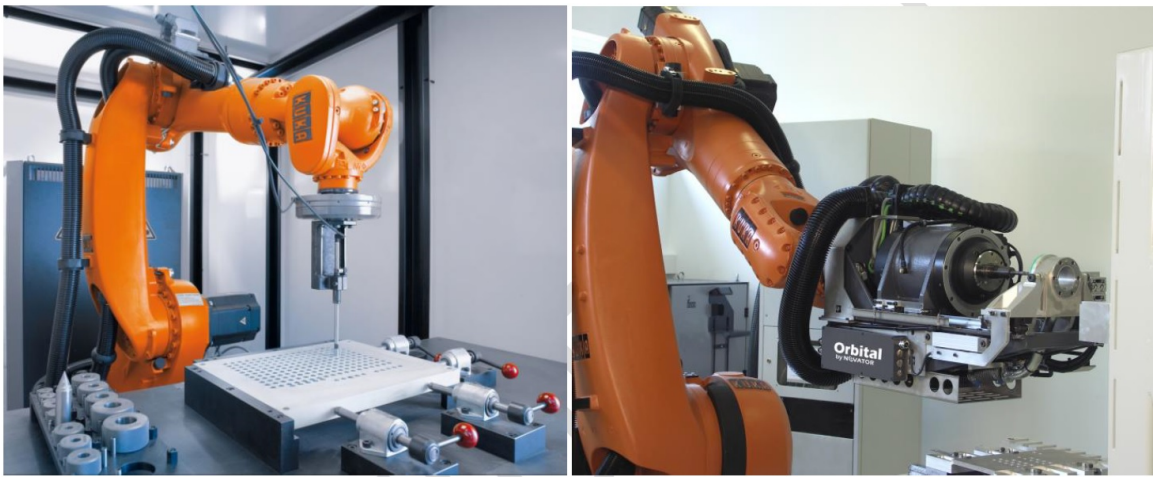


Figure 2.3: Examples of precision robotic applications .

For industrial robots, two types of accuracy, positioning accuracy and path tracking accuracy should be considered respectively. Positioning accuracy refers to the precision for the object to move to a pre-planned location, while path tracking accuracy represents the quality of tracking process when the TCP of the end-effector moves along the pre-planned path or trajectory. Many factors, such as inaccurate kinematic parameters (joint angels, joint or link deflections under payload, link lengths, etc.), friction, drive backlash), can impair the accuracy of industrial robots. Various precision applications in aerospace manufacturing industry, e.g., laser cutting, inspection, deburing and fastening, are increasingly robotized (Fig.2.3). The unidirectional position repeatability of a typical robot manipulator in manufacturing industry is about $0.02mm$, while the position accuracy of

such a robot, even after calibrated, can be only around $1mm$ [32]. The position accuracy of the industrial robots without calibration can be as bad as $\pm 10mm$ due to inaccurate kinematic models and assembly-line manufactured [31]. However, the majority of aerospace applications require tolerances of $0.20mm$ or less [32].

2.3 Coordinate Measure Machine (CMM)

A coordinate measuring machine (CMM) is defined as a type of sensor that can measure the information of the discrete points at the surface of the physical object so as to obtain the geometry (including position and orientation in the sensor frame) of the object [33]. In nowadays, no standardization of CMMs exists yet. There are tens of manufacturers, hundreds of different types of CMMs, and their corresponding unique software programs for running the machines [34]. The probe is one key component for all the CMMs in the market and can be non-contact, contact, and combined (multi-probes). Typically, for non-contact probes, laser signal or optical vision are used to attain the distance from the surface of the object to the probe. Contact probes are usually applied to measure the specific point or to obtain the profile of the object by scanning the surface of the object. Majority of the CMMs are able to provide the 6-DOF pose measurements in micrometer precision with the help of the synchronous data processing of the attached software.

Figure 2.4: Using laser tracker to calibrate the industrial robots.

CMMs have been extensively applied for measuring the manufacturing and assembly process in the fields of electronics, aerospace, machine tool, automobile, and etc. since the Ferranti Company

in Scotland developed the first CMM for their military intention in the 1950s [33]. CMMs with non-contact probes are perfectly suitable for high-precision, complex or easily deformable parts. A laser tracker is one type of portable non-contact CMM to utilize the laser beam travels to compute the accurate distance from the tracker to the target part. Comparing to the traditional CMM, laser tracker systems have absolute advantages on durability, flexibility, accuracy, and reliability for large-scale metrology applications. Since laser tracker systems first arose in the 1990s, they have dominated most of the high-precision applications, such as calibrations of the industrial robots in Fig.2.4, measuring and prototyping in aerospace manufacturing. The left picture in Fig.2.4 shows the setup at ÉTS (École de technologie supérieure) for calibrating a ABB robot using a FARO laser tracker. The right picture in Fig.2.4 demonstrates how KUKA calibrates its industrial robots using a FARO laser tracker as well. However, the cost of using laser tracker systems are generally over $\$100,000$ [35]. One most common spherically mounted retro-reflector (SMR) as the target device of a laser tracker costs at least $\$1000$. The measuring accuracy of a laser tracker can be guaranteed only in perfect laboratory conditions.

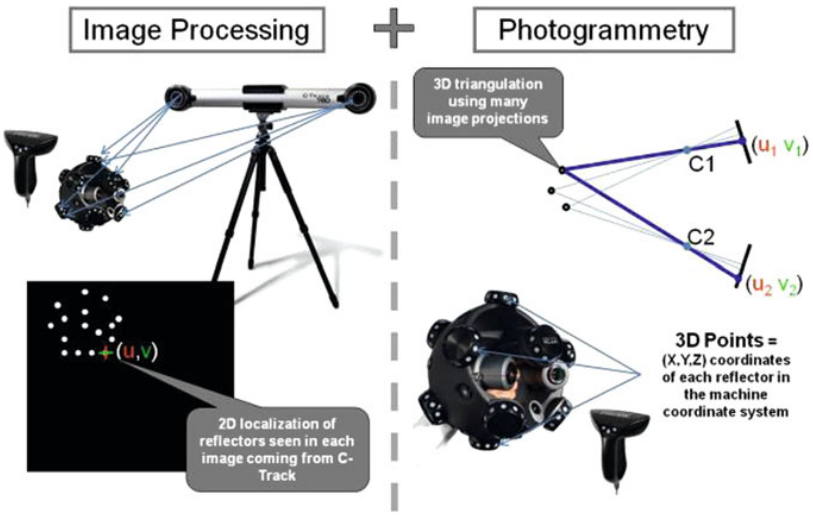


Figure 2.5: The image processing of a optical CMM [36].

Vision-based optical CMMs are also non-contact portable. Optical CMMs are regarded as CNC (Computer Numerical Control) image measuring machines, which rely on the technology advancement of the image processing to obtain automatic fast and accurate measurements. Fig.2.5 demonstrates an example of the complex principles for an optical CMM. Recent decades have seen that

the price of the optical CMMs becomes more competitive, and much better performance, e.g., faster measurements, excellent reliability and superior portability, are offered by optical CMMs [36]. Optical metrology has breakthrough on performing high-frequency, continuous inspections on all sizes, shapes, and materials. Correspondingly, optical CMMs will start playing a more important role in manufacturing industry.

2.4 Kalman Filter

The name of Kalman filter was originated from Rudolf E. Kalman since he completely present the theory of Kalman filter in 1960 [37]. According to the fundamental principle of Kalman filter, Kalman filter is also called as LQE (linear quadratic estimation) [38]. The algorithm of Kalman filter integrates the control and statistics theory, and it produces real-time optimal estimations recursively based on a series of observed measurements instead of a single measurement. The algorithm of Kalman filter does not require big memory space and complex computation consumption. Its unique advantages have attracted many researchers from different fields and become one of the most common data filtering and fusion methods. The three significant features of Kalman filter are summarized in [38] as in Table.2.2.

Table 2.2: The three significant features of Kalman filter.

No.	features	explanation
1	discrete	The measurement information is sampled at constant interval repeatedly.
2	recursive	Current estimations is based on the past prediction and current measurements, then update the prediction matrix for the future.
3	predictive	The prediction for the future is based on the current state, e.g., acceleration, velocity, position, etc.

2.4.1 Kalman Filter for Nonlinear System

In recent decades, Kalman filter has been widely applied in visual servoing for optimal or sub-optimal estimation [39, 40]. Normally, a linear dynamic process model is required and a priori

statistical information of the process and measurement noise are assumed to be known in Kalman filter algorithm. However, industrial robots mechanism including kinematics and dynamics is highly nonlinear and the pose estimation from image information captured by the eye-to-hand cameras is a nonlinear computation. In order to apply Kalman filter to estimate the end-effector pose of industrial robots, many researchers have utilized extended Kalman filter (EKF) by [17, 41]. The optimality of EKF depends on the accuracy of dynamic process model, which is hard to obtain. Moreover, the process and measurement noise are time varying as the robots move at different speeds, with various configurations, along different trajectories and in different situations. Recently, the adaptive Kalman filter (AKF) has attracted more and more attention of the researchers to overcome the problems and adapt to complicated situations without knowing certain system model and precise priori knowledge about the noise [39, 42, 43]. Therefore, this thesis research adopts the AKF to achieve the smooth pose estimation when the industrial robots are subjected to time-varying process and measurement noise.

2.4.2 Kalman Filter with Irregular Sampled Measurements

Ideally, the measurements for Kalman filter estimation are expected to be sampled at a constant time interval δt since δt is a constant variable in the algorithm of Kalman filter. However, irregular sampling rates and time-varying time delays tend to be unavoidable for obtaining the measurements.

Early in 1984, T. Glad and L. Ljung published a research about using Kalman filter to estimate velocity from the irregular measurements [44]. The research in [45] presents using EKF to process the infrequent and delayed measurements for estimating nonlinear state during chemical process. The computation of algorithm [45] is time-consuming and the accuracy has to be sacrificed to shorten the time consumption. The researchers [46] presents a modified Kalman filter with time-varying delays and irregular sampling for linear time varying system, not for nonlinear system. The researches in [47] and [48] introduce Kalman filter based sensor fusion to deal with the irregular sampling problem in slow-rate measurement for laboratory data analysis. To the best of our knowledge, few literature about Kalman filter with irregular sampled measurements for non-linear systems, especially for industiral robots. In this thesis research, the AKF's filtering frequency is set as a fraction of the sampling frequency. Its updating algorithm utilizes extrapolations of the

original measurements, incorporating accurate time stamps during the sampling interval to mitigate the impact of irregular sampling intervals.

2.5 Visual Servoing for Industrial Robots

Conventionally, the control for accuracy enhancement of industrial robots are confined to be performed under known structured environments with calibrated kinematic parameters. However, inevitable errors and uncertainties tends to exist even after complicated calibration at the expense of huge efforts and cost. Undoubtedly, the emergence of visual servoing, a closed-loop control scheme, present a dawn light in autonomous control of industrial robots. Visual servoing is an approach to control the movement of the robot by using the visual measurements as feedback. Visual servoing is a multi-disciplinary research topic involving computer vision, robotic kinematics, nonlinear dynamics, control theory, as well as real-time systems. The term *visual servoing* was first introduced by Hill and Park in 1979 [49]. Prior to the introduction of this term, the less specific term *visual feedback* was generally used. Shirai and Inoue [50] first used the term *visual feedback* in a application to control a robot. Researchers have made considerable progress in this field in the past forty years. The appearance of the high speed processors and cameras has made visual servoing possible for real time implementation and industrial applications. The revolutions brought by visual servoing for industrial robots includes the following [41]:

- Exact pose information of target objects is not required. Also, the target objects can be changed or moving.
- The requirement for Kinematic or dynamic model of industrial robots is relaxed. Therefore, the efforts for calibration and the expenses on mechanical stiffness of robot joints and arms can be largely reduced.
- Uncertain factors, such as gear backlash and elasticity of the links, are not completely fatal to robot accuracy.

In fact, most research around visual servoing is related to some specific task. For instance, the endpoint of the task is single target or multi-target and the optimal trajectory planning is required or

not. The target/targets is/are fixed/motionless or variable/moving. If the task is to approach single point or follow desired path, the objective of visual servoing is to pursue positioning accuracy or path tracking accuracy [24, 25]. After a task is specified, visual servoing will confront a series of decisions. First, feature points and parameters need to be selected [51]. Second, visual sensor and corresponding configuration should be determined. Visual sensor can be single camera, dual cameras or multi-camera. Also, the relation between the cameras and the end-effector of the industrial robot can be eye-to-hand, eye-in-hand or combination of both. Third, control scheme is necessary to be selected from position-based visual servoing (PBVS), image-based visual servoing (IBVS) and combination of both [52]. Last, specific control methods need to be specified to realize visual servoing, e.g., PID [25] or adaptive control [53].

2.5.1 Visual Servoing Categorization

According to the features extracted as feedback of controller, visual servoing can be classified into three categories, position or pose based visual servoing (PBVS) [17, 54, 55], image-based visual servoing (IBVS) [56, 57] and hybrid visual servoing [52]. In IBVS, the control input for the robots is computed with respect to the errors between the desired value and the current value of feature points on the image plane, which is weak in large rotations and tracking the path off-line planned in Cartesian space. IBVS is a 2D visual servoing scheme which converges in the vicinity of the desired image and often has the difficulty to estimate the depth [52]. In PBVS, the actual pose of the related object in Cartesian space is estimated according to the feature points on the image plane. Correspondingly, the control input for the robots is based on the errors between the current pose and desired pose of the end-effector [25, 55].

Considering the location of the vision system, the configuration of visual servoing is classified into eye-to-hand [19, 58], eye-in-hand [17, 57], cooperation of eye-in-hand and eye-to-hand. Eye-in-hand is referred to as the situation where the visual sensor is rigidly mounted on the end effector of the robot and eye-to-hand means the one where the visual sensor observes the robot within its workspace. By using the same visual sensor, the eye-in-hand type can obtain partial but precise sight of the scene whereas the eye-to-hand type has a less precise but global sight. The coordinate transformations between the pose of the visual sensor and the pose of the end-effector mounted on

the industrial robot are constant under the configuration of eye-in-hand while it is variable for the configuration of eye-to-hand as the end-effector is moving. For both configurations, the visual sensor is necessary to be calibrated periodically so as to fine tune the intrinsic parameters which may be changed due to operations and environments. However, there are still existing errors existing in the kinematic transformation even after calibration. As a result, pose accuracy is hard to be guaranteed under the configuration of eye-in-hand since only the target object is observed. Therefore, visual servoing with eye-in-hand visual sensor is also classified as endpoint open-loop [59, 60]. In contrast, visual servoing with eye-to-hand visual sensor is called as endpoint closed-loop because both the end effector of the robot and the target object are observed in the field of view (FOV) of the visual sensor synchronously [24, 25, 55, 61]. In order to integrate the advantages of eye-in-hand and eye-to-hand, some researchers attempt to integrate both of them in one system [62, 63, 64]. For example, eye-to-hand camera offers feedback information for the translation movement at the first phrase and the rotation movement at the second phrase is guided based on the measurements from eye-in-hand camera [62].

With respect to the number of the cameras included in the visual sensor system, visual servoing is categorized into monocular vision [56] and stereo vision [58, 65]. Binocular vision is the most usual in stereo vision. For binocular vision, two views of the scene are taken from known different viewpoints to resolve depth ambiguity existing in monocular vision. The location of feature points in one view must be matched with the location of the same feature points in the other view. The coordinates of an point in the FOV of the stereo vision system can be computed by using triangulation projection [66]. Stereo vision system is rarely used in an eye-in-hand configuration due to the limits in the camera baseline distance, which affects the accuracy of depth estimation.

In this thesis research, three proposed control schemes (DPC, DPT and AILC) can be all classified as PBVS since the desired pose or path and the pose feedback from visual sensor are given as Cartesian coordinate. Binocular visual sensor , C-Track 780, is used in the experiments for verifying three control schemes and it observes the end-effector or other objects in its FOV at certain distance. Therefore, three control schemes can also be classified as eye-to-hand and stereo vision visual servoing.

2.5.2 Visual Servoing Applications

For industrial robots, visual servoing is mainly applied to improve the performance of the industrial robots by guiding the end-effector to desired pose (position and orientation) which can be variable or fixed. With the help of the visual servoing, industrial robots seem to be equipped with "eyes" and "brain", and can be aware of where to go and how fast to move. More extensive and complex applications, e.g., dexterous assembly and adhesive dispensing in aerospace manufacturing, liquid handling in medical analyzing, machine and cutting in automatic manufacturing, spray painting, etc., become possible for industrial robots to liberate human from tedious and dangerous work, as shown in Fig.2.6. Beside the applications in industrial manufacturing, with the capability enhancement of the industrial robots, more new fields begin to be explored. For example, industrial robots are designed to inspecting the oil and natural gas pipelines [67]. In Canada, there are nearly 100,000 km of aging pipeline with oil and natural gas flowing through and some of these pipelines have never got inspected for corrosion, tear, wear and any other defects [68]. Undoubtedly, this type of applications will bring human uncountable benefits. Another example is the vision-based application of industrial robots in agriculture. It is well-known that there are high labor intensity and time sensitivity in agriculture. It couldn't be imagined how significant to the farmers if their work are mostly accomplished by intelligent robots.

2.6 Iterative Learning Control for Industrial Robots

There are many control methods to improve the accuracy of robots [10, 16, 59, 70]. However, due to the presence of the uncertainties in dynamic model or parameters, it has been difficult to achieve perfect tracking from the beginning of the trajectory. Moreover, many industrial tasks are operated repetitively over a finite time period. In such cases, the control performance can be significantly improved by taking advantage of the process repeatability. As a result, a new control theory, iterative learning control (ILC), was born in 1971 [71]. ILC is a recently developed control method, but is well-established in control fields. ILC can effectively improve the transient response and tracking performance for uncertain dynamic systems that work repetitively. ILC excels in achieving perfect tracking for repetitive systems that perform tasks within defined time intervals. Industrial

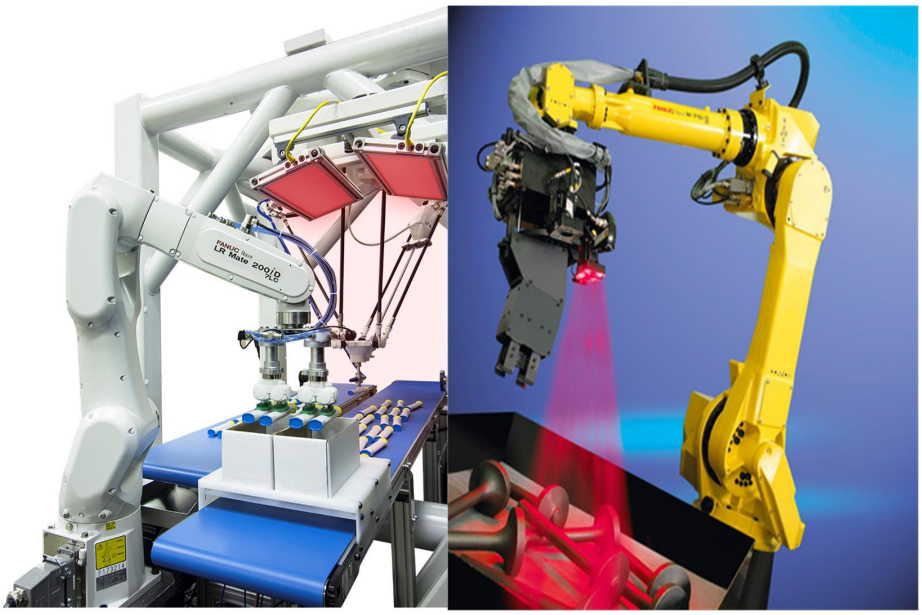


Figure 2.6: (a) FANUC robot 200iD pick and place with vision guiding (left); (b) FANUC Robot working with 3D vision (right) [69].

robots serve as a prime example of these repetitive systems. The basic configuration of iterative learning control (ILC) is illustrated in Fig.2.7. Over the past decades, ILC theory and application for accuracy enhancement of industrial robots has been explored fruitfully. One typical application environment is industrial robots for manufacturing industries. The first ILC research was to control the robotic manipulators by Arimoto et al. [72].

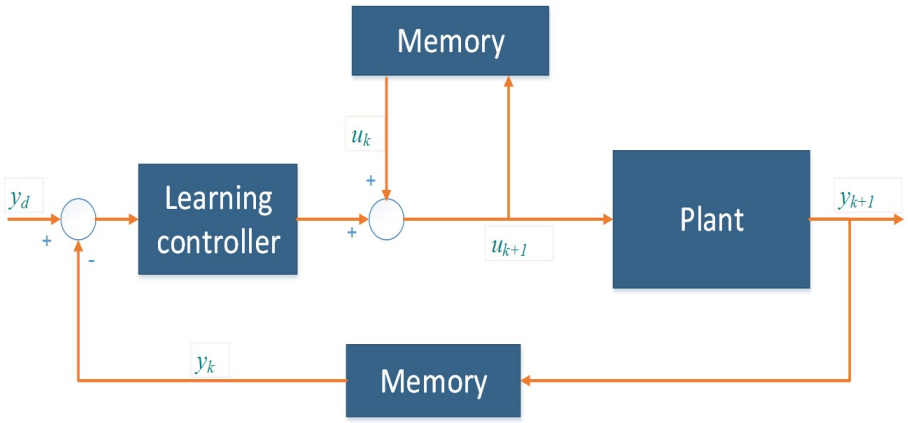


Figure 2.7: Basic configuration of Iterative learning control

It is broadly recognized that Uchiyama et al. gave the first ILC formulations [73], which was published in Japanese in 1978. Arimoto et al. present it in English in 1984 [72], which opened

further research on ILC. Thenceforth, various related issues on ILC theory, such as the structure of ILC, updating laws of ILC, optimization of ILC, robustness of ILC, convergence of ILC, adaptive ILC, etc, have been extensively investigated. A large amount of literature has been published, such as research monographs [74, 75, 76, 77], and survey papers [78, 79, 80, 81]. ILC is recognized as an important branch of intelligent control, with industrial robots serving as the primary focus of experimentation and application.

There are four desirable features that can explain why ILC becomes an attractive control technique [77]. The first feature of ILC is the simplicity of ILC structure. A simple controller is always preferable not only for the implementation cost but also for the control quality or reliability. The second feature of ILC is the fact that ILC can achieve a near perfect tracking both in the transient period and in steady state with repetitive learning. The third feature of ILC is its design and execution with the uncertainties of system model or parameters under a global Lipschitz condition. This feature is an important feature for real-time implementation, because practically it is tough and costly to build an accurate plant model. The fourth feature of ILC is its capability to acquire non-causal signals for control compensation. Since ILC is a memory-based controller, the control signals of previous iteration are utilized to update the current control signals. Consequently, it is easy to compensate the process or sampling-time delay inherently in the feedback loops. These features of ILC determine its unique place in control fields. The actual industrial robots are highly complex nonlinear systems with a multitude of uncertain nonlinearities while many industrial tasks are high-accuracy-required and repetitively operated. It is natural to apply ILC to industrial robot as well as other repeatable nonlinear systems [80, 82, 83, 84] which work repeatedly over a finite time interval. The only essential restriction for ILC on nonlinear system is to satisfy global Lipschitz condition (GLC).

The purpose of this section is to review the research and application of ILC for nonlinear systems and especially for industrial robots. Literature review on the existing ILC updating laws for nonlinear systems and industrial robots will be given.

2.6.1 ILC Design for Nonlinear Systems

According to the relation form $F(.)$, ILC updating law for nonlinear system can be categorized into two classes: classic ILC updating law and hybrid ILC updating law, which are reviewed in the following two subsections respectively.

Classic ILC Design for Nonlinear systems

Arimoto et al. provided the classic ILC updating law initially in 1986 as below [85]:

$$u_{k+1}(t) = u_k(t) + \phi e_k(t) + \Gamma \dot{e}_k(t) + \psi \int e_k(t)dt, \tag{2.1}$$

where as denoted in Fig.2.7, k is iteration number; $e_k(t) = y_d(t) - y_k(t)$ and $\dot{e}_k(t) = \dot{y}_d(t) - \dot{y}_k(t)$; ϕ , Γ and ψ are gain matrices; $u_k(t)$ and $u_{k+1}(t)$ are control inputs at iteration k and $k+1$ respectively. With respect to the tracking error $e_k(t)$, Eq.(2.1) is PID-type linear learning law because it includes a proportional, derivative and integral gain terms on $e_k(t)$. Before introducing PID-type learning law, Arimoto et al. proposed the first ILC updating law in 1984 [72], which was a D-type algorithm.

Unarguably, in recent years, the P-type, D-type, PD-type learning law are utilized extensively, particularly for nonlinear systems [72, 86, 87]. Since the learning gains depend on tuning like PID feedback control, the uncertainties of dynamic system model and parameters are allowed. On the other hand, because there is a natural integrator action along the iteration axis, the last term in Eq.(2.1), integrator term appears rarely in the learning algorithms.

Based on the classic ILC updating law Eq.(2.1), a few modifications have been made by recent researchers. For example, some researchers [88, 89, 90] used the current tracking error $e_{k+1}(t)$ as the proportional term instead of previous one $e_k(t)$. This method is called current-iteration, and in fact, the current tracking error is output feedback. Additionally, some researchers [75, 91, 88] advanced the high-order learning algorithms by using one more previous iterations of $u(t)$ and $e(t)$, for example $u_k(t),\ u_{k-1}(t)$ and $e_k(t),\ e_{k-1}(t)$, to obtain $u_{k+1}(t)$. This high-order algorithm provide more design parameters and achieve faster convergence rate than the first-order algorithm and the tracking error can go to arbitrarily small in some circumstances [92]. However, the theoretic

proof has not been well established so far. Besides, in order to reduce the influence of initial conditions variation, the simplest method is to introduce a forgetting factor to modify the ILC updating law [77]. Take a modified P-type ILC updating law as an example:

$$u_{k+1}(t) = \alpha u_k(t) + \phi e_k(t) \tag{2.2}$$

where $\alpha \in (0, 1)$ is a forgetting factor.

The classic ILC can reject the repeating disturbance and noise by learning, but it is weak to compensate non-repeatable disturbance. Therefore, many researchers combined feedback control concept with iterative learning control. In [93, 94], Longman et al. used ILC to modify the reference signal to compensate feedback controller to follow the desired trajectory. I [78], Bristow et al. introduced ILC controller to alter the control input together with feedback controller to feed in system plant.

Whatever learning algorithm is designed, it should be proved to be convergent along the iteration axis in order that the perfect tracking is achieved.

Hybrid ILC Design for Nonlinear Systems

Many researchers have attempted to combine classic ILC ideas with existing control methodologies. In order to solve some specific problems or issues, they integrate classic ILC with other control theories, such as robust control, optimization theory, adaptive control , fuzzy control, and neural network control.

Robust ILC Robustness is a common problem when a controller is designed for a nonlinear dynamic system with uncertainties. Robust ILC represents the ILC with robustness consideration. The researchers pay much attention on the robust ILC for nonlinear systems subjected to uncertain disturbances or time-varying reference or time-varying uncertainties [76, 82, 95, 92, 96, 97]. In [84, 98], Chen et al. introduced an ILC algorithm to realize output tracking for nonlinear stochastic systems with unknown dynamics and noise statistics. Stochastic ILC is defined as ILC for systems that contain stochastic signals with system noise, measurement noise, random packet losses, etc. A

latest survey by Shen et al. detailed the stochastic ILC techniques [81].

Optimal ILC Optimal ILC is one of main research areas for ILC theory and also well-established. Similar to robust ILC, optimization tends to be considered in generic ILC design. The idea of norm-optimal ILC was mentioned first in [99]. Amann et al. proposed a kind of norm-optimal ILC controller to obtain optimal control force at $k+1$ iteration by minimizing the cost function [100]. Hatzikos et al. applied the norm-optimal ILC to calculate the optimal learning gain by genetic algorithm (GA) for a class of nonlinear ILC problems [101]. Owens et al. developed optimality-based adaptive ILC algorithms [102, 103]. Based on quadratic cost form, Choi et al. minimized the performance index function by using the steepest gradient method [104]. Besides the above examples, the brief by Owens et al. elaborated the theory development, algorithms and design guidelines about norm-optimal ILC before 2013 [105].

Fuzzy ILC and Neural Network ILC Fuzzy ILC and neural network ILC are another two small branches, relating ILC to fuzzy logic and neural network theory respectively. The learning gains of ILC can be obtained by using neural network or fuzzy logic schemes [106]. Xu et al. took fuzzy control as the basic control part and ILC as the refinement part to design a fuzzy-PD ILC controller [107]. Precup et al. merged the advantages of fuzzy control and ILC to improve the performance of control system [108]. Chien et al. presented a fuzzy ILC for a class of non-lipschitz nonlinear system with a given task in [109] and proposed a new adaptive fuzzy ILC for the uncertain nonlinear systems to deal with various control tasks, nonzero initial state errors and non-repeatable disturbances in [110]. Chien also used ILC to tune all the weights of neurons of a feedforward neural network such that desired learning performance can be achieved in [111]. Chow et al. utilized two recurrent neural networks with time-varying weights in the ILC system to approximate the nonlinear system and achieve the desired system response [112]. Yang et al. used the neural network to estimate learning gain of ILC updating law to track different reference trajectories [113]. Similarly, Jiang et al. [114] and Yamakita et al. [115] used neural-network-based ILC to realize trajectory tracking for nonlinear system.

From the literature review on fuzzy ILC and neural network ILC, only a few researchers carried

out limited research although the fuzzy ILC and neural network ILC were proved to be effective under certain circumstances by their simulation results or analysis. On the other hand, it is very interesting that robust ILC, fuzzy ILC and neural network ILC often appear together with adaptive iterative learning control, defined as adaptive ILC, which is an important ILC research branch for nonlinear dynamic system and will be reviewed in the following content.

Adaptive ILC Before giving the review of adaptive ILC for nonlinear system, the energy function method as the fundamentals of adaptive ILC is introduced. For nonlinear dynamic systems, especially for nonlinear uncertain systems, Lyapunov stability theory has been broadly applied in the control design and stability analysis. Some ILC researchers succeeded in developing the energy function concept in both time domain and iteration domain with the enlightenment from Lyapunov stability theory [82, 93]. Thereafter, the ILC design and convergence analysis become more flexible and interesting. Typically, the adaptive ILC is developed to deal with nonlinear systems with uncertainties of model or parameters [77, 116, 117, 118, 119]. In recent years, the concept of the composite energy function(CEF) was proposed. Such composite energy function represents the system energy in both iteration domain and time domain [117, 120, 121].

Since adaptive techniques have obtained great achievement both in theory and application for tracking control of uncertain nonlinear systems, the adaptive ILC seems to attract more and more attention of the ILC researchers. There are mainly three types of adaptive ILC methods:

- *Type 1*: differential adaptive ILC [117]
- *Type 2*: pointwise adaptive ILC [120, 122, 123]
- *Type 3*: hybrid differential-difference adaptive ILC [124, 125]

In Type 1, the uncertain parameters are supposed to be time-invariant constants and are updated in the time domain in each iteration. Type 2 is applicable to the cases where the parameters are time-varying in time domain and invariant in the iteration domain. In Type 2, the control parameters are updated in the iteration domain. Compared with Type 1, Type 2's response is not so smooth because the parametric estimation in Type 1 is proportional to the integration of updating term, which is related to system states and tracking error, while in Type 2 the parametric estimation is directly proportional to updating term. Combining these two methods, Qu et al. [124] and Tayebi

et al. [125] introduced Type 3, so-called differential-difference adaptive learning law by choosing the weighting factor to decide the tradeoff between Type 1 and Type 2. Theoretically speaking, an appropriate value of the weighting factor in Type 3 can obtain both the smooth response and adaption to time varying parameters. However, very few researches have been done about Type 3 and only simulation results under some assumptions or limitations for simplified system are provided [125]. On the other hand, in practical applications for industrial robots, adaptive parameters, especially when the number of them are less than the DOFs of the robot, are impossible to be constant or slow-varying due to complex nonlinear dynamics and disturbances. Therefore, pointwise adaptive ILC (Type 2) by choosing time-varying adaptive parameters and updating them along iteration axis is adopted in this thesis research.

2.6.2 ILC Design for Industrial Robots

Industrial robots are usually used to replace human to perform some fixed tasks repetitively over a fixed time period. Therefore, it is natural to apply iterative learning control (ILC) to enhance the robot accuracy. Because the dynamic model of industrial robots is highly nonlinear and coupled, it attracts many researchers to attempt using ILC technique to approach perfect tracking. There are a lot of existing literature about ILC application research for industrial robots. Some benchmark works are reviewed as below chronologically. Since 1984, Arimoto et al.'s original work [72], PID-type ILC scheme, has been broadly recognized and has been guiding later research. The PID-type ILC scheme, convergence proof and analysis, forgetting factor, as well as application examples in robotic manipulators, have founded ILC theory research and been inspiring many recent researchers. In 1985, M Togai et al. proposed a discrete learning scheme to realize position and velocity tracking convergence and discussed the similarity between learning controller and state observer [126]. In 1986, Arimoto et al. analyzed how to use P-type, D-type, PD-type in tracking control for manipulators [85]. In 1988, Bondi et al. proposed high-gain feedback to ILC application for robotic manipulators and succeeded in improving the convergence rate and stability [127]. Also in 1988, Oh et al. proposed an adaptive ILC scheme without demand of initial conditions alignment and defined a novel Lyapunov-like function to prove the tracking error convergence [128]. In 1993, Horowitz et al. applied adaptive ILC with function identification, compared the method with betterment learning

schemes and other adaptive ILC with integral transforms, and gave the experimental results [87]. In 2000, Choi et al. proposed an adaptive ILC scheme to estimate uncertain parameters in time domain, to identify and compensate repetitive disturbances in iteration domain, provided simulation to confirm the validity of the scheme [129]. In 2002, Norrlöf et al. [130] proposed an estimation procedure with a Kalman filter based on adaptive ILC and an optimization of a quadratic criterion to realize disturbance rejection and trajectory tracking. In 2004, Tayebi et al. [131] advanced a practical adaptive ILC scheme and used a Lyapunov-like energy function to prove its convergence. From literature among the ILC application research for industrial robots, the number of the reported adaptive ILC application research is over one third, and the adaptive ILC research is almost synchronous from 1984 to now. Thus adaptive ILC scheme is an important and practical direction in ILC application for industrial robots. Finding Lyapunov-like energy function is common method to prove the validity and convergence of ILC algorithms. Tayebi et al. are very active on the adaptive ILC application for robotic manipulators [83, 96, 97, 131, 132, 125], whose works are relatively new but have been widely accepted in the area.

Classic ILC Design for Industrial Robots

The overall control law can be described as below:

$$\tau_k(t) = \tau_{fb}(t) + \tau_{kff}(t) \tag{2.3}$$

where $\tau_{fb}(t)$ is feedback control part computed on feedback signal and $\tau_{kff}(t)$ is feed-forward part computed by ILC algorithm.

The feedback control input $\tau_{fb}(t)$ can be obtained as below:

$$\tau_{fb}(t) = G(q) - K_p(q - q_d) - K_d(\dot{q} - \dot{q_d}) \tag{2.4}$$

$\tau_{fb}(t)$ is designed to stabilize the closed-loop system and keep the error bounded. The learning input $\tau_{kff}(t)$ is designed mainly to compensate uncertain disturbances and nonlinear dynamics when the robotic system repeats its operations to track the desired trajectory such that it can improve the

tracking performance along the iteration axis. Actually, the learning input is the output of ILC controller. Therefore, how to design ILC controller is the key problem in the application for industrial robots. Classic ILC is PID-type control along iteration axis, and consists of proportional, integral, and derivative terms on the error, while the P-term plays a stabilizer role and results in monotonic convergence, the I-term can increase the convergent rate and reject the effect of initial conditions variation, and D-term is able to reduce the effect of disturbance inputs. Since there is a natural integrator along the iteration axis, I-term in classic ILC is rarely used. As a result, for industrial robots, D-type ILC and PD-type ILC are most broadly used in classic ILC. Arimoto proved the convergence of D-type ILC algorithm in [72] based on contraction mapping theory while some prior information of the system dynamics is required. The convergence analysis of this learning scheme for industrial robots proved by Kawamura et al. in [133]. PD-type ILC also requires the prior knowledge about system dynamics, and global Lipschitz continuous (GLC) condition should be satisfied strictly.

Adaptive ILC Design for Industrial Robots

In tracking control application for industrial robots, adaptive control is well-established and extensively employed in scenarios where uncertain parameters remain constant, and control task extends over $[0, \infty)$. However, when dealing with time-varying uncertain parameters and a control task defined within a fixed time frame $[0, T]$, adaptive ILC becomes a more suitable approach. On the other hand, due to the high nonlinearities of system dynamics for robot manipulator, satisfying the GLC condition is almost impossible. Correspondingly, contraction mapping theory is not suitable to prove the convergence of control process again. Inspired by the contribution of Lyapunov functions in nonlinear system control, researchers came up with composite energy function (CEF), which was a combination of a standard Lyapunov function and a $L_2\text{-}norm$ of the parametric errors. CEF approach significantly broadens the ILC research and facilitates ILC design and convergence analysis. The general control strategy in adaptive ILC is similar as Eq.(2.3), a PD-feedback part for stability in parallel with ILC part for compensating the unknown parameters and disturbances. Three typical schemes of adaptive ILC are model-based multi-parameter [131], model-free multi-parameter [132, 134, 135]and one-parameter [131]. They can be classified as Type 2, pointwise adaptive ILC [80]. The convergence of model-based multi-parameter adaptive ILC was proved by

Tayebi et al. in [131]by using an suitable composite energy function(CEF) . The main difference between model-based multi-parameter and model-free multi-parameter is the use of system parameter matrix. The convergence of model-free multi-parameter adaptive ILC was proved by Islam et al. in [134] by using appropriate composite energy function(CEF). Compared to the former two adaptive ILC schemes, one-parameter adaptive ILC seems to be simplest. However, comparing the experimental results of model-free multi-parameter and one-parameter adaptive ILC on a 5-DOF CRS255 robotic manipulator provided in [132], the former demonstrates much faster convergence rate along iteration axis and higher tracking accuracy than the latter. Moreover, the proof of the convergence by using appropriate composite energy function (CEF) [131] needs to be assured with the knowledge of certain bounds of some system parameters. Therefore, model-free multi-parameter adaptive ILC is more appropriate for pursuing high tracking accuracy of industrial robots without any priori knowledge of the system dynamic models. In this thesis research, the adaptive ILC proposed in Chapter 6 is a model-free two-parameters adaptive ILC.

2.7 Summary

In this chapter, the developments, applications and important features of industrial robots and CMMs are presented in first two subsections. Then Kalman filter, as an effective real-time filter, is introduced, especially Kalman filter for nonlinear system and irregular sampled measurements is emphasized. Then, visual servoing, as the principal control strategy in this thesis research work, is reviewed. The extensive applications and existing categorization of visual servoing are presented. Last not the least, iterative learning control (ILC), as an important advanced control method in robotic fields, is explored. Since this thesis research work is focusing on the control of serial manipulators which are typical nonlinear system, the developments of ILC for nonlinear system are mainly involved.

Chapter 3

Adaptive Kalman Filter Based Pose Estimation with Irregular Sampling

3.1 Introduction

In this thesis research, the experiments for verifying dynamic pose correction (DPC) scheme, dynamic path tracking (DPT), and adaptive iterative learning control scheme (AILC) scheme are implemented on two FANUC robots (FANUC M20iA and FANUC LR Mate 200iC) by using an optimal CMM system, C-Track 780 from Creaform, to provide pose (position and orientation) information measurements. Since noises inevitably exist in the measurements of C-Track 780, Kalman filter as a type of real-time filter is introduced to remove the noises from the measurements of C-Track 780 and obtain the usable pose estimation. In this chapter, FANUC M20iA and FANUC LR Mate 200iC will be first introduced. Then, the important parameters for C-Track 780 from Creaform will be provided. Last but not the least, an adaptive Kalman filter will be presented.

3.2 Workspace Description and Pose Estimation

3.2.1 Workspace Description

In this thesis research, an intuitive scenario is an optical CMM, C-Track 780, watching and guiding the end-effector of an industrial robot, e.g., a FANUC robot, to complement some tasks

which may include some task objects. First, the relation in workspace among optical CMM, the industrial robot and other physical bodies needs be constructed. Consider a 6-DOF industrial robot with an end-effector in Fig.3.1, which illustrates the definition and relation of the coordinate frames. Generally, the default tool frame is defined at the center of the flange of Joint 6, $\mathcal{P}_0$. A tool frame $\mathcal{F}_E$ with origin at the TCP can be defined by offsetting the default tool frame to the TCP with the known relative position information. Assume that the end-effector can reach anywhere in the reachable workspace $\mathcal{W}_R$ of the robot. In the workspace, a user frame $\mathcal{F}_{U_R}$ for the robot can be defined with respect to base frame $\mathcal{F}_B$. The pose of the end-effector in $\mathcal{F}_{U_R}$ can be represented by the tool frame $\mathcal{F}_E$ in $\mathcal{F}_{U_R}$. Then, the homogeneous transformation matrix ${}^{U_R}_{E}\mathbf{H}$ from $\mathcal{F}_{U_R}$ to $\mathcal{F}_E$ can be obtained.

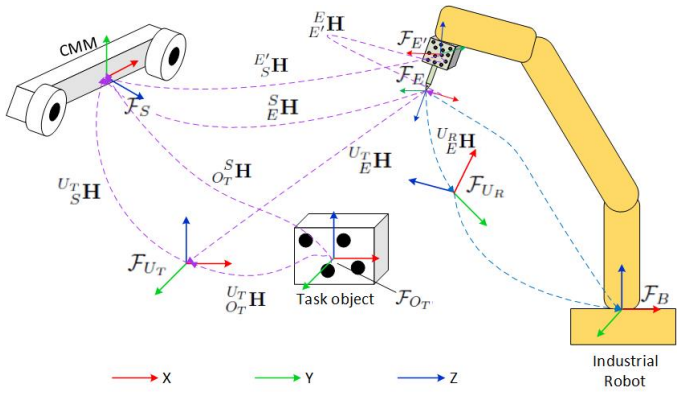


Figure 3.1: Definition and relation of the coordinate reference frames in the workspace.

On the other hand, when a task is given, the task space $\mathcal{W}_T$ ($\mathcal{W}_T \subset \mathcal{W}_R$) can be defined. In the task space, a user frame $\mathcal{F}_{U_T}$ for the task can be defined. Correspondingly, the task object with an defined reference frame $\mathcal{F}_{O_T}$ involved in this task can be represented in $\mathcal{F}_{U_T}$.

Moreover, in terms of the optical CMM, there is default sensor frame $\mathcal{F}_S$ whose origin is at the center of the CMM. User frame $\mathcal{F}_{U_T}$ can be defined by using the CMM to be represented in $\mathcal{F}_S$ by moving the end-effector along the axes of $\mathcal{F}_{U_R}$ so that $\mathcal{F}_{U_T}$ is parallel to $\mathcal{F}_{U_R}$. The end-effector with positioning targets is identified as a model with the reference frame $\mathcal{F}_{E'}$. In order to locate the end-effector, during any movement, the end-effector should be in the field-of-view (FOV) of the CMM. The pose of the end-effector in $\mathcal{F}_S$ is represented by the frame $\mathcal{F}_{E'}$ in $\mathcal{F}_S$. Therefore, the homogeneous transform matrix ${}^{U_T}_{S}\mathbf{H}$ from $\mathcal{F}_{U_T}$ to $\mathcal{F}_S$ and ${}^{S}_{E'}\mathbf{H}$ from $\mathcal{F}_S$ to $\mathcal{F}_{E'}$ at any time can

be obtained respectively. The homogeneous transformation matrix ${}^{U_T}_{E'}\mathbf{H}$ from $\mathcal{F}_{U_T}$ to $\mathcal{F}_{E'}$ can be obtained as follows

$$ {}^{U_T}_{E'}\mathbf{H} = {}^{U_T}_{S}\mathbf{H}\,{}^{S}_{E'}\mathbf{H}. \tag{3.1} $$

Similarly, the task object with positioning targets, as shown in Fig.3.1, can be represented by the reference frame $\mathcal{F}_{O_T}$ in $\mathcal{F}_S$. The homogeneous transformation matrix ${}^{U_T}_{O_T}\mathbf{H}$ from $\mathcal{F}_{U_T}$ to $\mathcal{F}_{O_T}$ can be obtained as ${}^{U_T}_{O_T}\mathbf{H} = {}^{U_T}_{S}\mathbf{H}\,{}^{S}_{O_T}\mathbf{H}$.

Since user frame $\mathcal{F}_{U_T}$ in the task space $\mathcal{W}_T$ is parallel to $\mathcal{F}_{U_R}$ memorized in robot controller, there is only translation and no rotation between $\mathcal{F}_{U_T}$ and $\mathcal{F}_{U_R}$ in workspace. The main difference is that $\mathcal{F}_{U_R}$ is represented in $\mathcal{F}_B$ while $\mathcal{F}_{U_T}$ in $\mathcal{F}_S$. As shown in Fig.3.1, the relation between $\mathcal{F}_E$ and $\mathcal{F}_{E'}$ is relatively constant. The rotation matrix ${}^{U_T}_{E}\mathbf{R}$ from $\mathcal{F}_{U_T}$ to $\mathcal{F}_E$ is same as ${}^{U_R}_{E}\mathbf{R}$ from $\mathcal{F}_{U_R}$ to $\mathcal{F}_E$. There, the rotation matrix ${}^{E'}_{E}\mathbf{R}$ from $\mathcal{F}_{E'}$ to $\mathcal{F}_E$ can be deducted as follows

$$ {}^{E'}_{E}\mathbf{R} = {}^{U_T}_{E'}\mathbf{R}^{-1}\,{}^{U_T}_{E}\mathbf{R}. \tag{3.2} $$

3.2.2 Pose Estimation

The pose estimation of any object in task space is similar to the pose estimation of the end-effector. The pose estimation of the end-effector is the localization of the frame $\mathcal{F}_{E'}$ with respect to the sensor frame $\mathcal{F}_S$ by first mapping $\mathcal{F}_{E'}$ in $\mathcal{F}_S$ and then projecting them onto the image plane. Since a dual-camera sensor C-Track 780 is employed as the visual measurement instrument, the pose estimation principle about binocular vision is presented in this section. C-Track 780 can provide continuous image acquisition and transmission in real time. Moreover, C-Track 780 can track the reference model, including a rigid set of reflectors, which work as the feature points. It is assumed that there are n feature points, $n > 3$, on the rigid end-effector, and the homogeneous coordinates of each feature point in sensor frame are denoted as ${}^{S}\mathbf{P}_i = (x_i, y_i, z_i, 1)$, $i = 1 \cdots n$, while the projection coordinates of each feature point on the image plane of the j^{th} Camera is represented as ${}^{C}\mathbf{P}_{ij} = (u_{ij}, v_{ij}, 1)$, $i = 1 \cdots n$ and $j = 1, 2$, where j is the number of dual cameras. ${}^{C}\mathbf{X}_j$, $j = 1, 2$, is the projection matrix of each camera. The perspective projection can be given as below

[136]:

$$
\begin{aligned}
{}^{C}\mathbf{P}_{ij} &= {}^{C}\mathbf{X}_{j}\,{}^{S}\mathbf{P}_{i}, \\
{}^{C}\mathbf{X}_{j} &= \mathbf{B}_{j}\,{}^{C}_{S}\mathbf{H}_{j},
\end{aligned}
\tag{3.3}
$$

where $j = 1, 2$; $\mathbf{B}_j$ is the camera matrix, integrating the intrinsic parameters of the j^{th} camera; ${}^{C}_{S}\mathbf{H}_j$ is the homogeneous transformation matrix from the sensor frame $\mathcal{F}_S$ to the j^{th} camera frame. When the dual-cameras sensor is calibrated, $\mathbf{B}_j$ and ${}^{C}_{S}\mathbf{H}_j$ are known. Therefore, when ${}^{C}\mathbf{P}_{ij}$ is obtained in the image plane, ${}^{S}\mathbf{P}_i$ can be computed.

Ideally, ${}^{S}\mathbf{P}_i$ computed from each camera should be the same. However, due to the distortion, calibration errors and other noises, there is difference between two results of ${}^{S}\mathbf{P}_i$ from dual cameras. The triangulation is the main way to balance the difference in the results [137]. In order to make a matching pair of points, ${}^{C}\mathbf{P}_{i1}$ and ${}^{C}\mathbf{P}_{i2}$, to meet in space, the following constraint should be satisfied:

$$
{}^{C}\mathbf{P}_{i1}^{T}\mathbf{G}\,{}^{C}\mathbf{P}_{i2} = 0,
\tag{3.4}
$$

where $\mathbf{G}$ is the fundamental matrix that can be computed when dual camera projection matrices, ${}^{C}\mathbf{M}_1$ and ${}^{C}\mathbf{M}_2$, are given. Due to the uncertainty of image processing, Eq.(3.4) may not be satisfied accurately. According to optimal correction principle of Kanatani [137], the objective function is

$$
\min_{{}^{C}\hat{\mathbf{P}}_{i1}{}^{T}\mathbf{G}\,{}^{C}\hat{\mathbf{P}}_{i2}=0} (d({}^{C}\mathbf{P}_{i1}, {}^{C}\hat{\mathbf{P}}_{i1}) + d({}^{C}\mathbf{P}_{i2}, {}^{C}\hat{\mathbf{P}}_{i2})),
\tag{3.5}
$$

where $\min(.)$ represents the minimization function subject to the constraint ${}^{C}\hat{\mathbf{P}}_{i1}{}^{T}\mathbf{G}\,{}^{C}\hat{\mathbf{P}}_{i2} = 0$, $d(*,*)$ denotes Euclidean distance, ${}^{C}\hat{\mathbf{P}}_{i1}$ and ${}^{C}\hat{\mathbf{P}}_{i2}$ are the estimated points of ${}^{C}\mathbf{P}_{i1}$ and ${}^{C}\mathbf{P}_{i2}$ respectively. As a result, ${}^{C}\hat{\mathbf{P}}_{i1}$ and ${}^{C}\hat{\mathbf{P}}_{i2}$ can be obtained by the following formulas:

$$
\begin{aligned}
{}^{C}\hat{\mathbf{P}}_{i1} &= {}^{C}\mathbf{P}_{i1} - \frac{({}^{C}\mathbf{P}_{i1}, \mathbf{G}\,{}^{C}\mathbf{P}_{i2})\mathbf{V}\mathbf{G}\,{}^{C}\mathbf{P}_{i2}}{\boldsymbol{\omega}}, \\
{}^{C}\hat{\mathbf{P}}_{i2} &= {}^{C}\mathbf{P}_{i2} - \frac{({}^{C}\mathbf{P}_{i1}, \mathbf{G}\,{}^{C}\mathbf{P}_{i2})\mathbf{V}\mathbf{G}^{T}\,{}^{C}\mathbf{P}_{i1}}{\boldsymbol{\omega}}, \\
\boldsymbol{\omega} &= (\mathbf{G}\,{}^{C}\mathbf{P}_{i2}, \mathbf{V}\mathbf{G}\,{}^{C}\mathbf{P}_{i2}) + (\mathbf{G}^{T}\,{}^{C}\mathbf{P}_{i1}, \mathbf{V}\mathbf{G}^{T}\,{}^{C}\mathbf{P}_{i1}),
\end{aligned}
\tag{3.6}
$$

where the inner product of two vectors a and b is denoted as (a, b); projection matrix $\mathbf{V} \equiv diag(1, 1, 0)$.

Then, by Eq.(3.3), the coordinate of i^{th} feature point in sensor frame, ${}^{S}\mathbf{P}_i$ can be obtained.

After the position information of all the feature points on the end-effector is prepared, the pose estimation of the end-effector can be developed. Suppose n feature points on the rigid end-effector are fixed and known from the definition of the frame $\mathcal{F}_{E'}$, whose homogeneous coordinates are denoted as ${}^{E'}\mathbf{P}_i = ({}^{E'}x_i, {}^{E'}y_i, {}^{E'}z_i, 1)$. It is assumed the current pose of $\mathcal{F}_{E'}$ in $\mathcal{F}_S$ is denoted as $(x_c, y_c, z_c, \gamma_c, \beta_c, \alpha_c)$, where (x_c, y_c, z_c) represents the origin position of $\mathcal{F}_{E'}$ in $\mathcal{F}_S$, while the orientation $(\gamma_c, \beta_c, \alpha_c)$ represents the Euler-angle rotation from $\mathcal{F}_{E'}$ in $\mathcal{F}_S$. With the pose $(x_c, y_c, z_c, \gamma_c, \beta_c, \alpha_c)$ of $\mathcal{F}_{E'}$ in $\mathcal{F}_S$, the homogeneous transformation matrix ${}^{S}_{E'}\mathbf{H}$ from $\mathcal{F}_{E'}$ to $\mathcal{F}_S$ can be formulated as below:

$$ {}^{S}_{E'}\mathbf{H} = \begin{bmatrix} R(\gamma_c, \beta_c, \alpha_c) & (x_c, y_c, z_c)^T \\ 0\,0\,0 & 1 \end{bmatrix}, \tag{3.7} $$

where $R(\gamma_c, \beta_c, \alpha_c)$ is rotation matrix from $\mathcal{F}_E$ to $\mathcal{F}_S$. Correspondingly, the transformation equation of i^{th} feature point can be written as:

$$ {}^{S}\mathbf{P}_i = {}^{S}_{E'}\mathbf{H}\, {}^{E'}\mathbf{P}_i, \tag{3.8} $$

which can be unfolded into three nonlinear equations with six unknown variables $(x_c, y_c, z_c, \gamma_c, \beta_c, \alpha_c)$ as below:

$$ \begin{aligned} x_i &= x_c + C_{\gamma_c} C_{\beta_c}\, {}^{E'}x_i + (C_{\gamma_c} S_{\beta_c} S_{\alpha_c} - S_{\gamma_c} C_{\alpha_c})\, {}^{E'}y_i \\ &\quad + (C_{\gamma_c} S_{\beta_c} C_{\alpha_c} + S_{\gamma_c} S_{\alpha_c})\, {}^{E'}z_i, \\ y_i &= y_c + S_{\gamma_c} C_{\beta_c}\, {}^{E'}x_i + (S_{\gamma_c} S_{\beta_c} S_{\alpha_c} + S_{\gamma_c} C_{\alpha_c})\, {}^{E'}y_i \\ &\quad + (S_{\gamma_c} S_{\beta_c} C_{\alpha_c} - C_{\gamma_c} S_{\alpha_c})\, {}^{E'}z_i, \\ z_i &= z_c - S_{\beta_c}\, {}^{E'}x_i + C_{\beta_c} S_{\alpha_c}\, {}^{E'}y_i + C_{\beta_c} C_{\alpha_c}\, {}^{E'}z_i, \end{aligned} \tag{3.9} $$

where $C_a = cos(a)$ and $S_a = sin(a)$. In order to solve Eq.(3.9) for $(x_c, y_c, z_c, \gamma_c, \beta_c, \alpha_c)$, at least three non-collinear feature points are required [17]. However, as indicated in [138], at least four coplanar feature points are necessary for an unique solution while additional non-coplanar feature points can be used to improve the estimation accuracy with measurement noise. Since the number

of the feature points on the end-effector n is more than 3, $(x_c, y_c, z_c, \gamma_c, \beta_c, \alpha_c)$ can be determined uniquely. By using the proprietary software VXelements provided by Creaform, the end effector is defined as the tracking model which is built based on the selected reflectors on the surface of the end effector. The positional and rotational information of the tracking model with respect to sensor frame can be acquired, recorded or displayed simultaneously. Therefore, the computation to obtain the pose of the end effector is carried out by VXelements.

3.3 Adaptive Kalman Filter for Smoothing Estimated Pose

The presence of noise is inevitable in the image information from the optical CMMs. Moreover, the movement of the end-effector is highly likely to cause vibration, blur and distortion to the images. Therefore, in this section, an adaptive Kalman filter is presented to smooth the estimated pose data of the end-effector. The sampling interval of the CMM is denoted as T_s. Ideally, T_s can be regarded as a constant. However, T_s tends to be variable. Assuming that T_f represents a fixed filter interval. $T_f \approx T_s/n, n \geq 2$ is a strategy to reduce the influence of time delay resulted from the irregular sampling interval. For convenience, k instead of kT_f represents the current time instant and $k-1$ is the previous time instant. Suppose that the current state vector of the industrial robot is represented as follows:

$$\begin{aligned} \boldsymbol{\rho}_{k,k} = (&x(k), y(k), z(k), \gamma(k), \beta(k), \alpha(k), \\ &\dot{x}(k), \dot{y}(k), \dot{z}(k), \dot{\gamma}(k), \dot{\beta}(k), \dot{\alpha}(k))^T, \end{aligned} \tag{3.10}$$

i.e. the pose and velocity of the end-effector at the current time instant. Precise time stamp t_c is marked with the current measurement estimation $\boldsymbol{o}_c = (x_c, y_c, z_c, \gamma_c, \beta_c, \alpha_c)^T$.

Then, an adaptive Kalman filter method is given as the following recursive equations.

First, the prediction equations are given as below:

$$\begin{aligned} \hat{\boldsymbol{\rho}}_{k,k-1} &= \mathbf{A}\hat{\boldsymbol{\rho}}_{k-1,k-1}, \\ \mathbf{W}_{k,k-1} &= \mathbf{A}\mathbf{W}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{k-1}, \end{aligned} \tag{3.11}$$

where $\mathbf{A} \in R^{12} \times R^{12}$ is a state transition matrix which is applied to the previous state $\hat{\boldsymbol{\rho}}_{k-1,k-1}$

to obtain the current predicted state $\hat{\boldsymbol{\rho}}_{k,k-1}$; $\mathbf{W}_{k,k-1} \in R^{12} \times R^{12}$ is the current prediction of the error covariance matrix which is a measure of the accuracy of the state estimate while $\mathbf{W}_{k-1,k-1} \in R^{12} \times R^{12}$ are the previous error covariance matrix; $\mathbf{Q}_{k-1} \in R^{12} \times R^{12}$ is the process noise covariance computed from the information at time instant $k-1$. All the diagonal elements of $\mathbf{A}$ are 1 and $\mathbf{A}_{i,i+6}(i = 1 \cdots 6)$ are equal to the sampling interval T_s. $\hat{\boldsymbol{\rho}}_{0,0}$ and $\mathbf{W}_{0,0}$ are initialized by using the static samples before path tracking movement. Second, Kalman filter gain $\mathbf{D}_k$ is computed as below:

$$\mathbf{D}_k = \mathbf{W}_{k,k-1}(\mathbf{W}_{k,k-1} + \mathbf{\Omega}_{k-1})^{-1}, \tag{3.12}$$

where $\mathbf{\Omega}_{k-1}$ is the previous measurement noise covariance. Third, the estimation updating is formulated as below:

$$\begin{aligned} \hat{\boldsymbol{\rho}}_{k,k} &= \hat{\boldsymbol{\rho}}_{k,k-1} + \mathbf{D}_k((\boldsymbol{o}_k{}^T, O_6)^T - \hat{\boldsymbol{\rho}}_{k,k-1}), \\ \mathbf{W}_{k,k} &= \mathbf{W}_{k,k-1} - \mathbf{D}_k\mathbf{W}_{k,k-1}, \end{aligned} \tag{3.13}$$

where $\hat{\boldsymbol{\rho}}_{k,k}$ is produced as the optimal pose of the end-effector at time instant k; $O_6 = (0, 0, 0, 0, 0, 0)$. The extrapolation algorithm for computing the current measurement vector $\boldsymbol{o}_k = (x_c(k), y_c(k), z_c(k), \gamma_c(k), \beta_c(k), \alpha_c(k))^T$ is as follows. If $\boldsymbol{o}_c$ is not updated in current filter interval, $\boldsymbol{o}_k$ can be updated as:

$$\begin{aligned} x_c(k) &= \hat{x}(k-1) + \dot{\hat{x}}(k-1)T_f, \quad y_c(k) = \hat{y}(k-1) + \dot{\hat{y}}(k-1)T_f, \\ z_c(k) &= \hat{z}(k-1) + \dot{\hat{z}}(k-1)T_f, \quad \gamma_c(k) = \hat{\gamma}(k-1) + \dot{\hat{\gamma}}(k-1)T_f, \\ \beta_c(k) &= \hat{\beta}(k-1) + \dot{\hat{\beta}}(k-1)T_f, \quad \alpha_c(k) = \hat{\alpha}(k-1) + \dot{\hat{\alpha}}(k-1)T_f. \end{aligned} \tag{3.14}$$

If $\boldsymbol{o}_c$ is updated during former filter interval, $\boldsymbol{o}_k$ can be updated as:

$$\begin{aligned} x_c(k) &= x_c + \dot{\hat{x}}(k-1)(t-t_c), \quad y_c(k) = y_c + \dot{\hat{y}}(k-1)(t-t_c), \\ z_c(k) &= z_c + \dot{\hat{z}}(k-1)(t-t_c), \quad \gamma_c(k) = \gamma_c + \dot{\hat{\gamma}}(k-1)(t-t_c), \\ \beta_c(k) &= \beta_c + \dot{\hat{\beta}}(k-1)(t-t_c), \quad \alpha_c(k) = \alpha_c + \dot{\hat{\alpha}}(k-1)(t-t_c), \end{aligned} \tag{3.15}$$

where t is current system time.

$\mathbf{Q}_k$ and $\mathbf{\Omega}_k$ are symmetric positive definite matrices. The proper selection or updating of $\mathbf{Q}_k$ and $\mathbf{\Omega}_k$ are critical for the accuracy of filtered pose. When the robot is static, $\mathbf{\Omega}_k$ can be easily obtained by using the root mean square error of the static measurements from VMI. However, $\mathbf{\Omega}_k$

is a time-varying matrix when the robot is moving because the higher velocity will make the delay, blur, vibration and other uncertain factors worse.

A new effective method is proposed to update matrix $\mathbf{\Omega}_k$ based on the velocity of the TCP as follows

$$\begin{aligned} \Delta\mathbf{\Omega}_k &= diag[\mu_1\dot{x}(k), \mu_2\dot{y}(k), \mu_3\dot{z}(k), \mu_4\dot{\gamma}(k), \mu_5\dot{\beta}(k), \mu_6\dot{\alpha}(k)], \\ \mathbf{\Omega}_k &= \mathbf{\Omega}_0 + \Delta\mathbf{\Omega}_k, \end{aligned} \tag{3.16}$$

where $diag[\ ]$ represents a diagonal matrix whose elements are shown in the bracket; $\mathbf{\Omega}_0$ is the measurement noise covariance matrix when the TCP is static before starting from the start point; $\mu_i,\ i = 1 \cdots 6$, are the constant weights determining the influence of variable velocity, which is estimated by analyzing the static measurements of C-Track in workspace.

An adaptive recursive method is presented to optimize $\mathbf{Q}_k$ to compensate the prediction error and uncertain dynamic disturbance [43]. The predictor error $\hat{\boldsymbol{\epsilon}}_k$ can be estimated adaptively as below:

$$\hat{\boldsymbol{\epsilon}}_k = \hat{\boldsymbol{\rho}}_{k,k} - \mathbf{A}\hat{\boldsymbol{\rho}}_{k-1,k-1}, \tag{3.17}$$

Additionally, let $\Delta\mathbf{W}_k = \mathbf{A}\mathbf{W}_{k-1,k-1}\mathbf{A}^T - \mathbf{W}_{k,k}$. Then, $\mathbf{Q}_k$ can be computed recursively as below:

$$\begin{aligned} \bar{\boldsymbol{\epsilon}}_k &= \bar{\boldsymbol{\epsilon}}_{k-1} + \frac{1}{N}(\hat{\boldsymbol{\epsilon}}_k - \hat{\boldsymbol{\epsilon}}_{k-N}), \\ \mathbf{Q}_k &= \mathbf{Q}_{k-1} + \frac{1}{N-1}((\hat{\boldsymbol{\epsilon}}_k - \bar{\boldsymbol{\epsilon}}_k)(\hat{\boldsymbol{\epsilon}}_k - \bar{\boldsymbol{\epsilon}}_k)^T \\ &\quad - (\hat{\boldsymbol{\epsilon}}_{k-N} - \bar{\boldsymbol{\epsilon}}_k)(\hat{\boldsymbol{\epsilon}}_{k-N} - \bar{\boldsymbol{\epsilon}}_k)^T) \\ &\quad + \frac{1}{N(N-1)}((\hat{\boldsymbol{\epsilon}}_k - \hat{\boldsymbol{\epsilon}}_{k-N})(\hat{\boldsymbol{\epsilon}}_k - \hat{\boldsymbol{\epsilon}}_{k-N})^T) \\ &\quad + \frac{1}{N}(\Delta\mathbf{W}_{k-N} - \Delta\mathbf{W}_k), \end{aligned} \tag{3.18}$$

where N is the length of the past measurements memory for updating $\mathbf{Q}_k$. Moreover, in order to ensure that $\mathbf{Q}_k$ is positive definite, the diagonal elements need to be reset to their absolute values.

3.3.1 Adaptive Kalman Filter Initialization

In this thesis research, C-Track 780 from Creaform is selected as visual measurement instrument for all the experiments. Its maximum updating frequency for measuring the pose of a object with

VXelements is $29hz$. Accordingly, the filter interval T_f of AKF is initialized to be $12ms$ which is approximately close to $\frac{1}{29*3}s$. Based on the testing on C-Track 780, the sampling interval with VXelements is not strictly $\frac{1}{29}s$ and the time difference between two continuous sampling intervals can be in $\pm 3ms$.

Some variables and parameters involved in adaptive Kalman filter should be initialized properly with respect to $Eq.(3.10) - Eq.(3.18)$. First, the current pose measurements can be used to initialize the pose part of the initial estimation $\hat{\boldsymbol{\rho}}_{0,0}$ and the derivative part, velocity estimation, can be set to zeros. Then, the initial covariance matrix $\mathbf{W}_{0,0}$ can be initialized as a 12×12 identity matrix, which affects the transient convergent speed of the Kalman filter. The diagonal elements of $\mathbf{A}$ are 1 and $\mathbf{A}_{i,i+6}(i = 1 \cdots 6)$ is taken as $T_c/10^3$. Ideally, the current pose of the end-effector is equal to the former six elements of the current state vector $\hat{\boldsymbol{\rho}}_{k,k}$. $\boldsymbol{\Omega}_0$ is $0.001diag(0.15, 0.15, 0.8, 0.158, 0.143, 0.0219)$, which are obtained by calculating the root mean square error of static measurements for fixed point in $\mathcal{W}_R$. The constant weights $\mu_i,\ i = 1 \cdots 6$ for updating $\boldsymbol{\Omega}_k$ with velocity changes are set as $1.5e-6$. Compared to $\boldsymbol{\Omega}_0$, $\mathbf{Q}_k$ is initialized as $(1e-6)diag(1, 1, 10, 1, 1, 1, 1, 1, 10, 1, 1, 2)$ and the length N in Eq.(3.18) is 20.

3.4 Analyzing Pose Measurements without/with filtering by Kalman filter

Fig.3.2 and 3.3 demonstrate the original position and orientation measurements of the end-effector at a static point. As shown in Fig.3.2 and 3.3, the original pose measurements obtained from C-Track 780 are noisy. In order to analyze the accuracy and reliability of the pose measurements from C-Track 780, the poses of the end-effector at 8 different static points are measured and the first 500 measurements are used for calculating the RMS (root mean square) error, which is the RMS deviation from the mean of the measurements. The RMS errors of the pose measurements of the end-effector at 8 static points are presented in Table.3.1. At each point, the pose measurements are taken in two repeated tests. For instance, $\text{Test}1-1$ and $\text{Test}1-2$ are carried out at point 1. The distribution definition of 8 points in the FOV of C-Track 780 is according to Fig.A.6. In Table.3.1, 'Center' refers to the points in or close to $y-z$ plane in sensor frame, 'Right' refers to the points at

the right side of $y-z$ plane in sensor frame with positive x coordinate, and 'Left' refers to the points at the left side of $y-z$ plane in sensor frame with negative x coordinate. According to Table.3.1, all the RMS errors are less than 0.01 which represents the accuracy of the measurements from C-Track. Also, it can be observed in Table.3.1 that the RMS errors in center area of FOV are smaller than in right and left area. Therefore, the workspace for experiments can be considered to fit around the center area of the FOV. Moreover, the difference of the RMS errors obtained from the repeated tests at the same point are very small and can be neglected, which prove the high repeatability of C-Track 780. In fact, the high repeatability of C-Track 780 is especially critical factor for achieving high pose accuracy in Chapter $4-6$.

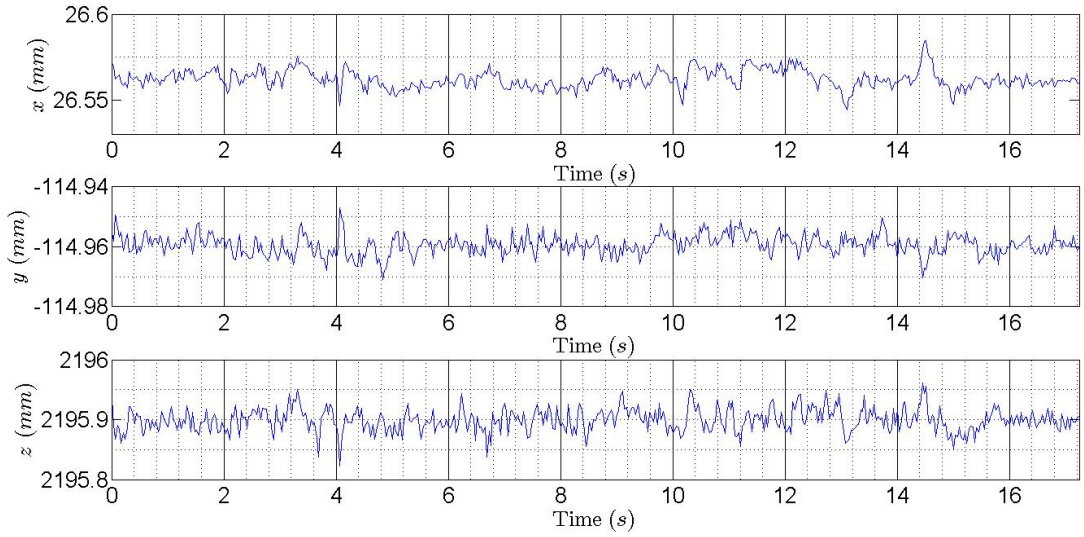


Figure 3.2: Static position measurements of the end-effector in sensor frame $\mathcal{F}_S$ by using C-Track 780 .

On the other hand, appropriate filtering is certainly necessary for extracting the accurate pose information from the noisy measurements of C-Track 780. Fig.3.4 and 3.5 demonstrate real-time pose measurements and filtered pose information by using the AKF for the end-effector at a static point. Fig.3.6 and 3.7 show real-time pose measurements and filtered pose information by using the AKF for the end-effector moving along a line at speed $25mm/s$. As observed in Fig.3.4$\sim$ 3.7, the pose estimations through the filtering of AKF can follow real-time pose measurements simultaneously. Also, the pose estimations through the filtering of AKF are much smoother than

real-time pose measurements so that they can be used as feedback information for close-loop control of the robot without causing vibration.

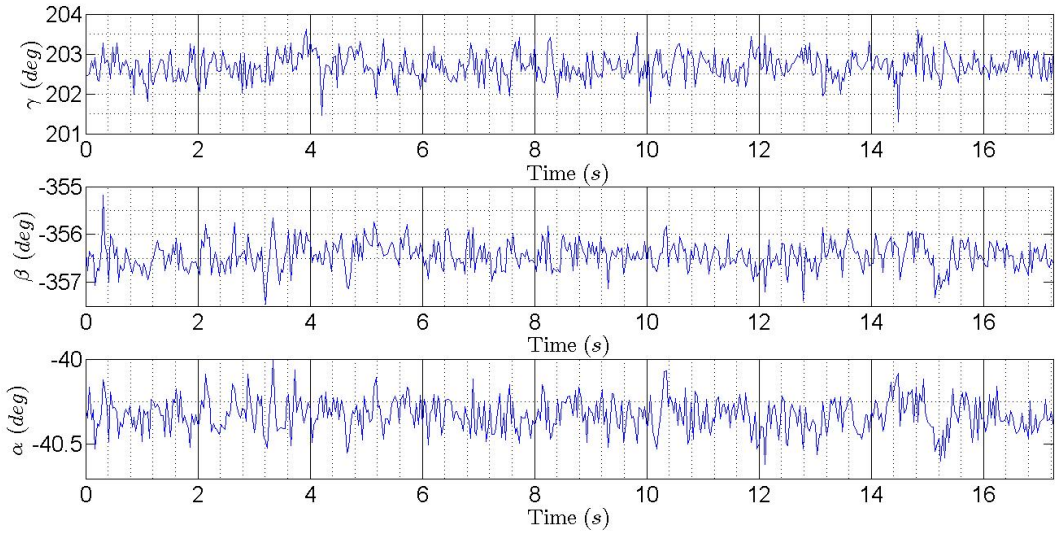


Figure 3.3: Static orientation measurements of the end-effector in sensor frame $\mathcal{F}_S$ by using C-Track 780 .

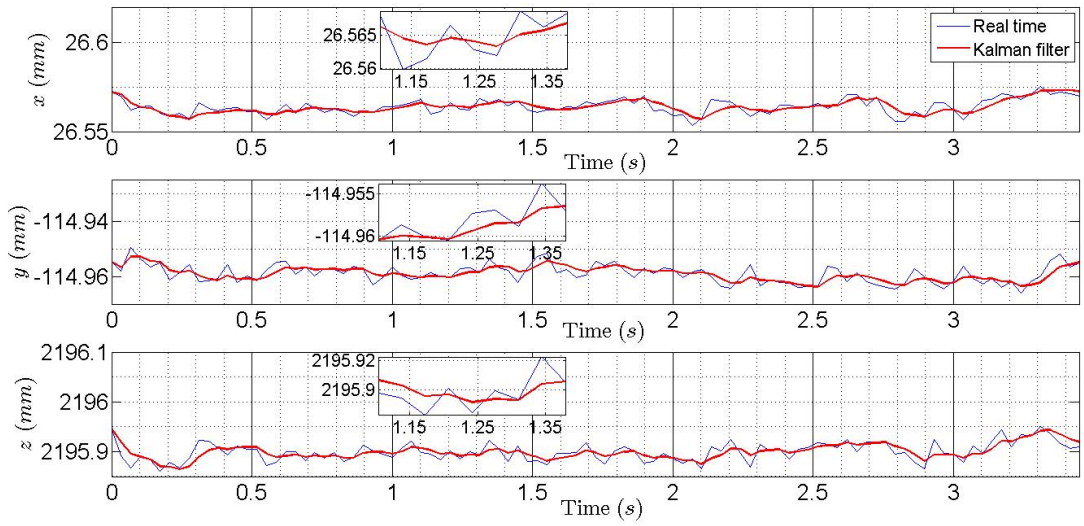


Figure 3.4: Position measurements of the end-effector at a fixed point in sensor frame $\mathcal{F}_S$ without/with filtering by AKF.

Table 3.1: RMS error of pose measurements by using C-Track 780 to measure the pose of the end-effector at fixed points.

RMS error / Test No.	Stationary Pose Measurement Error						Distribution
	$x(mm)$	$y(mm)$	$z(mm)$	$\gamma(deg)$	$\beta(deg)$	$\alpha(deg)$	
Test1-1	0.00437	0.00278	0.01621	0.00461	0.00442	0.00159	Center
Test1-2	0.00438	0.00269	0.01608	0.00460	0.00453	0.00157	Center
Test2-1	0.00465	0.00299	0.01591	0.00462	0.00405	0.00164	Center
Test2-2	0.00471	0.00281	0.01682	0.00467	0.00475	0.00169	Center
Test 3-1	0.00633	0.00520	0.02394	0.00611	0.00477	0.00152	Right
Test 3-2	0.00628	0.00512	0.02390	0.00609	0.00470	0.00155	Right
Test 4-1	0.00663	0.00413	0.02164	0.00582	0.00529	0.00172	Right
Test 4-2	0.00623	0.00368	0.02167	0.00576	0.00544	0.00171	Right
Test 5-1	0.00512	0.00349	0.01965	0.00512	0.00484	0.00176	Right
Test 5-2	0.00523	0.00827	0.01859	0.00479	0.00462	0.00168	Right
Test 6-1	0.00427	0.00256	0.01658	0.00487	0.00497	0.00167	Left
Test 6-2	0.00426	0.00268	0.01582	0.00487	0.00485	0.00165	Left
Test 7-1	0.00516	0.00348	0.01642	0.00525	0.00513	0.00138	Left
Test 7-2	0.00559	0.00367	0.01738	0.00529	0.00527	0.00148	Left
Test 8-1	0.00569	0.00565	0.01555	0.00475	0.00451	0.00119	Left
Test 8-2	0.00539	0.00558	0.01539	0.00497	0.00441	0.00119	Left

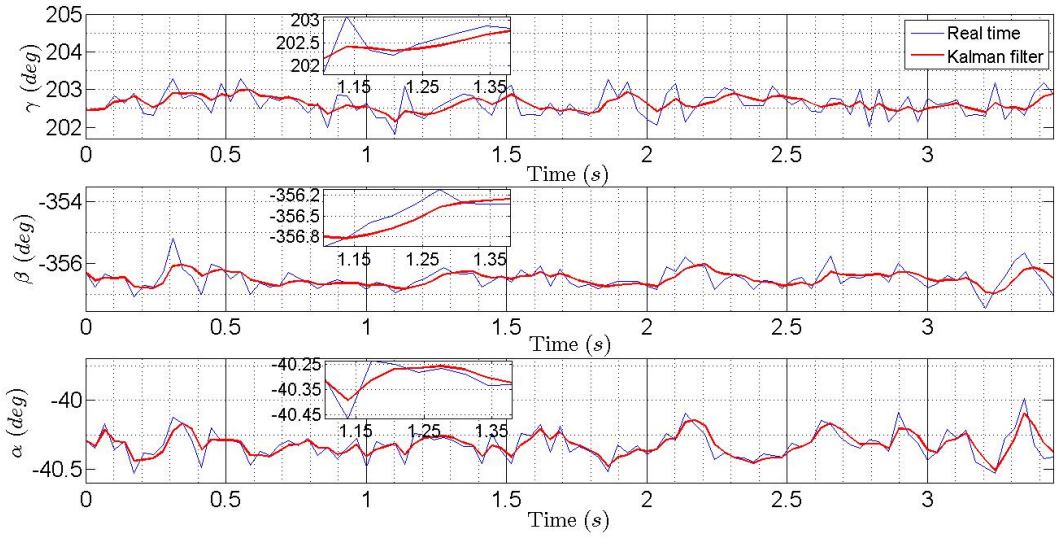


Figure 3.5: Orientation measurements of the end-effector at a fixed point in sensor frame $\mathcal{F}_S$ without/with filtering by AKF.

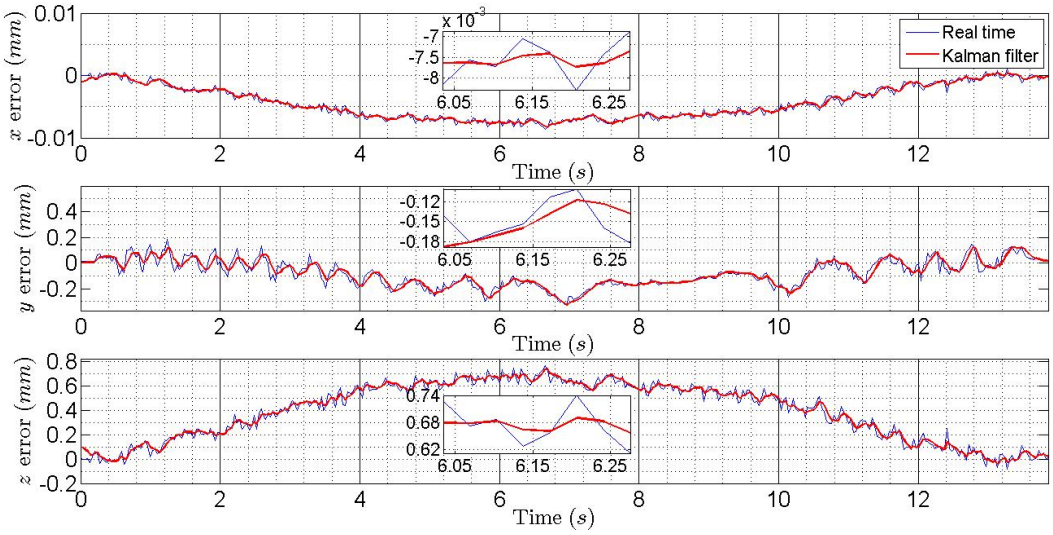


Figure 3.6: Position measurements of the end-effector in sensor frame $\mathcal{F}_S$ without/with filtering by AKF for the end-effector moving along a line at speed $25mm/s$.

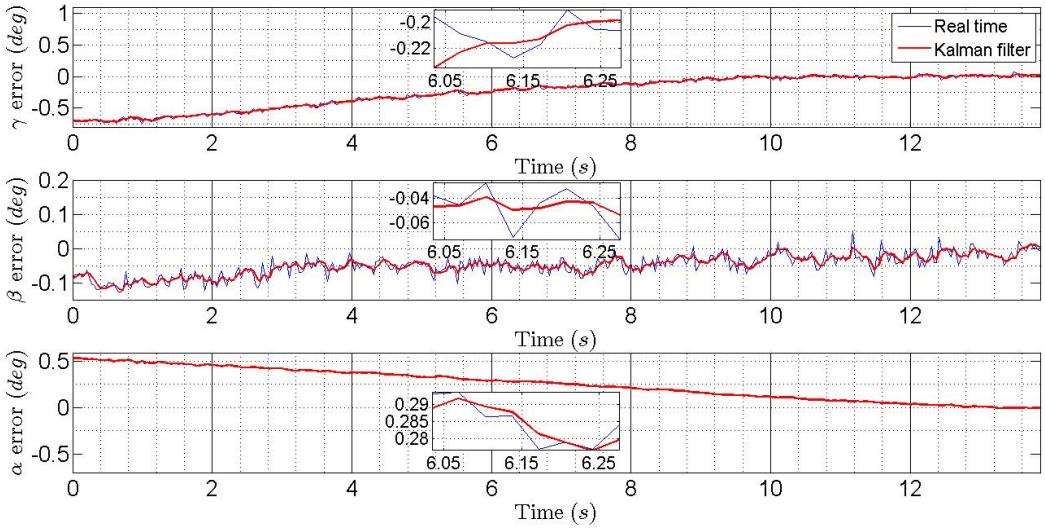


Figure 3.7: Orientation measurements of the end-effector in sensor frame $\mathcal{F}_S$ without/with filtering by AKF for the end-effector moving along a straight line at speed $25mm/s$.

3.5 Summary

This chapter provides some necessary preparation for the subsequent chapters. In Section 3.2, the relation of all the physical bodies in workspace is constructed by defining attached reference coordinate frames and identifying their transformation matrix. Also, the triangulation method for

pose estimation from dual cameras is presented. Moreover, an adaptive Kalman filter is developed for smoothing the pose estimation from the noisy measurements. Last, the pose measurements by using C-Track 780 are analyzed, the proposed AKF is applied for filtering the real-time pose measurements and the AKF is proved to be effective and practicable.

Chapter 4

Dynamic Pose Correction of Industrial Robots with High Accuracy through a Photogrammetry Sensor

4.1 Introduction

There are varieties of complicated factors, as listed in Table.1.1, that affect robot accuracy. According to the performance criteria description about industrial robots in ISO 9283 standard [139], there are two types of robot accuracy, pose accuracy for pose approaching performance and path accuracy for path tracking performance. Pose accuracy is required in the applications such as spot welding/soldering, drilling, inspecting, riveting and fastening, while path accuracy is critical for the applications such as arc welding, painting, spraying, cutting, deburring and polishing. Actually, pose accuracy can assure accurate initial pose for achieving high path accuracy during path tracking. Nowadays, off-line task planning in Cartesian coordinate space has become the main trend as the complexity of the application environments are increased. Accordingly, high robot accuracy is especially important for accomplishing the specific task. As concluded in Chapter 1, visual servoing is the most preferred method for industrial robots because dynamic disturbance and errors during the moving of the end-effector can be tackled based the visual feedback information. Peter Corke

has presented the major issues of visual servoing for industrial robots in early 1990s [140]. In this chapter, the research is to investigate the efficient position based visual servoing (PBVS) methods for improving the pose accuracy. One advantage of PBVS is no transformation from the desired pose in Cartesian coordinate space to image space of the visual system. The other advantage of PBVS is better convergence and execution rate than that of IBVS when the visual system can be calibrated beforehand [141]. Also the direct pose error in PBVS represented by Cartesian coordinate is more convenient for observing and analyzing the dynamic performance. One dynamic pose correction (DPC) scheme including PBVS control strategy for enhancing the dynamic pose accuracy will be presented in this chapter. It is well known that industrial robots tends to be applied to perform repetitive tasks. In the first round, DPC module works as the complementary controller to the robot controller, and the tool center point (TCP) of the end-effector can be controlled to the desired pose gradually with high pose accuracy no matter it is fixed or moving, single pose or multiple poses, with visual measurements from a photogrammetry sensor. As the TCP of the end-effector reaches the desired pose with satisfied accuracy, DPC module can enable the robot controller to record the location in its position register. Then, in the next round, the TCP of the end-effector can be fast directed to desired pose by the robot controller independently. The pose accuracy achieved in the first round plus high repeatability of the industrial robot can accommodate most of the precision requirements in aerospace manufacturing. When high pose accuracy is always strictly demanded, only minor adjustment through DPC module is supplemented after the first round. Therefore, the DPC scheme can be regarded as a learning based control method. The first round can be considered as a learning procedure. On the other hand, the traditional manually teaching method by teach pendant can be completely replaced by the DPC scheme so that more complex large-size tasks can be robotized.

In this thesis research, an portable optical CMM, C-Track 780, is selected as the photogrammetry sensor in the experiments for implementing the DPC scheme. Also, thanks to TRUaccuracy technology [142], a group of fixed reflectors close to the task space can be identified as dynamic referencing model and offering stable reference frame for extracting the pose information from visual measurments. Moreover, based on existing close range photogrammetry methodology, the

reflectors can be regarded as the feature points in the two views and need to be pairing and synchronized. On the other hand, VXelements is a commercial software platform provided by creaform, and it can communicate with the controller of C-Track 780. The model of an observed object with attached space distributed reflectors can be represented as a coordinate frame with some feature points defined in the dynamic frame through the interface of VXelements. Also, some significant functions, such as visual measurement data sampling, analyzing and demonstrating, are integrated in VXelements.

In order to avoid the disturbance from the measurement noises, the feedback information feeding in DPC module is using the pose estimation from the proposed adaptive Kalman filter in Chapter 3. In traditional robot operation practice, the fixed pose is taught manually by moving the TCP of the end-effector in the workspace and recording the pose of finite intermediate points. Moreover, the low accuracy and repeatability of the robot can not ensure the robot TCP to reach the points with high accuracy. Apparently, when the task is complex and requires high-precision, such method is hard to meet the strict high-precision requirements. In this chapter, the off-line planned pose is imported to robot controller. DPC can work in the control computer, which is connected with robot controller and visual sensor, $i.e.$, C-Track 780, by Ethernet. DPC is considered to compensate the calibration error, vibration error, uncertain model error as well as the repeatability error. The implementation of the DPC scheme in this paper is more flexible and convenient with the guarantee of high accuracy compared with the current robot controller. The task points can be produced by optimal task planning algorithm in advance and saved in the computer which can communicate with the robot controller.

Some experiments are implemented to verify the proposed DPC scheme on both Fanuc M20-iA and FANUC LR Mate 200iC. The successful application of the proposed scheme on Fanuc M20-iA and FANUC LR Mate 200iC with C-track 780, from Creaform, as visual measurement sensor exhibits the effectiveness of the DPC scheme.

The rest of this chapter is organized as the following. Problem statement is provided in Section 4.2 and Section 4.3 introduces the control configuration of the DPC scheme, equivalent pose error computation, DPC control law design. The experimental results on FANUC LR Mate 200iC with C-track 780 as visual sensor are presented to verify the effectiveness of the DPC scheme in Section

4.4. Finally, the concluding summaries and future works are given in Section 4.5.

4.2 Problem Statement

Many potential robotic applications in aerospace manufacturing are hindered due to insufficient robot accuracy although robotic automation has been thriving in this field for recent decades. Positioning accuracy within $\pm 0.25mm$ or less is expected for majority of the on-part operations in aerospace manufacturing [10]. However, the robot accuracy of standard industrial robot can reach within $0.3 \sim 0.5mm$ even after appropriate calibration [32, 143]. Therefore, robot accuracy deficiency is urgent to be resolved. In practice, positioning accuracy for many specific applications of industrial robots, such as drilling and fastening, is necessary to include both position accuracy and orientation accuracy of the end-effector, i.e., pose accuracy of the end-effector. This chapter will focus on enhancing pose accuracy of the end-effector.

Since the optical CMM, C-Track 780, is deployed in the following experiments as the photogrammetry sensor and dynamic referencing is definitely applied, dynamic referencing frame $\mathcal{F}_D$ can be defined in sensor frame $\mathcal{F}_S$ through VXelements by selecting certain space distributed positioning targets in workspace of the industrial robot. Then all the transformation related to $\mathcal{F}_S$ in Fig.5.1 need to be replaced by $\mathcal{F}_D$ as Fig.4.1. Through the 3D view of VXelements, the setup location of C-Track 780 can be adjusted to have the task space in the center of its FOV such that optimum quality of images can be obtained. Since there are unavoidable noises emerging in the measurements data from the photogrammetry sensor, it is essential to process the noisy data and obtain the effective estimation through the adaptive Kalman filter introduced in Chapter 3 before the feedback information is needed by the control part.

Traditionally, the destination of the end-effector is taught in robot user frame $\mathcal{F}_{U_R}$ by using teach-pendant. In this thesis research, the desired pose P_d of the end-effector can be off-line pre-planned and on-line modified in task space $\mathcal{W}_T$ according to the task requirement. The desired pose $\mathbf{p}_d$ of the end-effector is defined as $(x_d, y_d, z_d, \gamma_d, \beta_d, \alpha_d)$ in task user frame $\mathcal{F}_{U_T}$, where x_d, y_d, z_d and $\gamma_d, \beta_d, \alpha_d$ are coordinates and orientations of $\mathcal{F}_E$ in $\mathcal{F}_{U_T}$ respectively. It is assumed that $\mathbf{P}_0 = (x_0, y_0, z_0, \gamma_0, \beta_0, \alpha_0)$ is the start point. Then, the objective of DPC scheme is to move

the end-effector from $\mathbf{P}_0$ to $\mathbf{P}_d$ to satisfy the accuracy requirement.

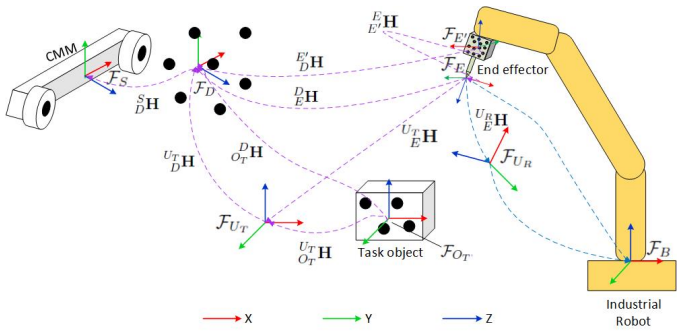


Figure 4.1: Definition and relation of the coordinate reference frames in the workspace with dynamic referencing frame.

4.3 Dynamic Pose Correction Control

Since the pre-planned destination is given in $\mathcal{F}_{U_T}$ instead of $\mathcal{F}_{U_R}$ or $\mathcal{F}_B$, it is not compatible in robot controller. As a result, there are coordinate frame calibration errors in addition to the low accuracy and repeatability of the industrial robot. Therefore, the robot controller can not guarantee to control the TCP of the end-effector to the desired point with high pose accuracy independently. A DPC control scheme is proposed in this section to improve the pose accuracy of the industrial robot to approach given points with specific poses.

4.3.1 Control Structure

The control configuration of DPC control scheme is demonstrated as Fig.4.2, which can be divided into three parts. The first part consists of robot controller and robot. Normally, each industrial robot comes with its proprietary robot controller. Only through robot controller, the robot parameters can be accessed and robot movement command can be realized. The second part is visual measurement part, which includes visual measurement instrument optical CMM and signal filtering that estimates the pose feedback information. The third part is the DPC control module, which is developed as the complementary control part cascaded on the robot controller. The control configuration consists of shown in Fig.4.2. In Fig.4.2, all the functions in the left dashed-line box are implemented as software modules running on the computer which is connected with the physical

devices in the right dashed-line box.

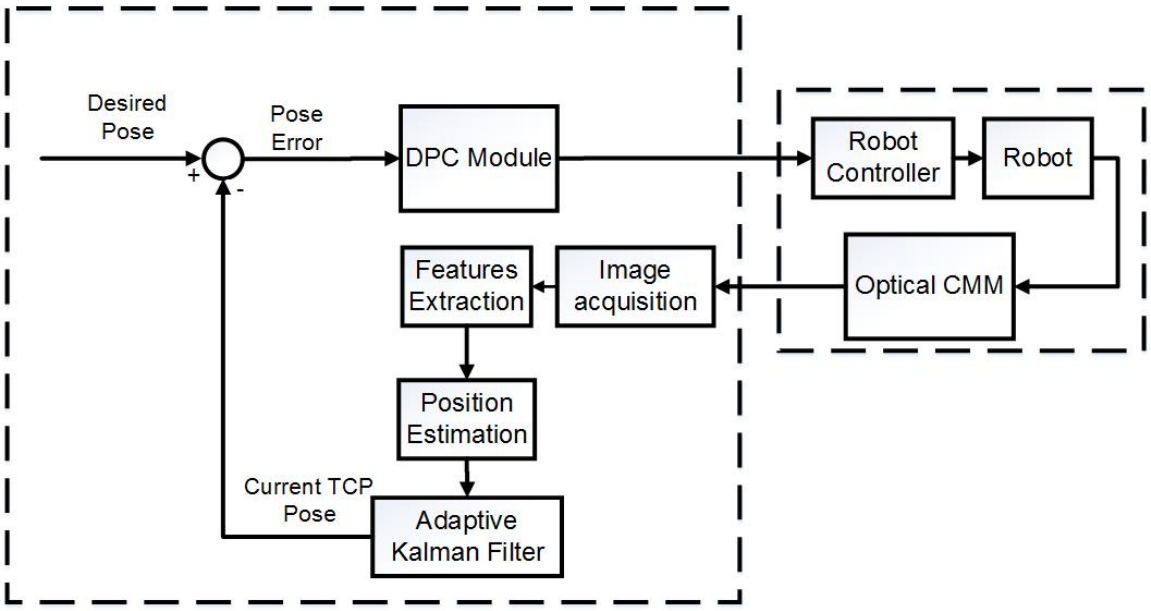


Figure 4.2: Control configuration of dynamic pose correction (DPC) control.

4.3.2 Pose Error Computation in Equivalent User Frame

Control interval and discrete time instances are denoted as T_c and k. At the end of each T_c, the current pose and velocity estimation of the end-effector, denoted as $\hat{\boldsymbol{\rho}}_{k,k}$, is obtained from the output of adaptive Kalman filter in Fig.4.2. $\mathbf{p}_c(k)$ is used to denote the pose part of $\hat{\boldsymbol{\rho}}_{k,k}$. Some conditions are assumed to be satisfied for any pose $\mathbf{p}_c(k) = (x(k), y(k), z(k), \gamma(k), \beta(k), \alpha(k))$ of the moving end-effector as follows:

$$\begin{aligned} &x(k) \in (x_{min} + \epsilon, x_{max} - \epsilon), \quad y(k) \in (y_{min} + \epsilon, y_{max} - \epsilon), \quad z(k) \in (z_{min} + \epsilon, z_{max} - \epsilon), \\ &\gamma(k) \in (\gamma_{min}, \gamma_{max}), \quad \beta(k) \in (\beta_{min}, \beta_{max}), \quad \alpha(k) \in (\alpha_{min}, \alpha_{max}), \end{aligned} \tag{4.1}$$

where (x_{min}, x_{max}), (y_{min}, y_{max}), and (z_{min}, z_{max}) are the 3D range of the workspace along three axis of $\mathcal{F}_{U_T}$; ϵ is a positive constant for isolating close proximity around the boundaries of the workspace so that big jerks or singularities can be avoided; $(\gamma_{min}, \gamma_{max})$, $(\beta_{min}, \beta_{max})$ and $(\alpha_{min}, \alpha_{max})$ are the orientation range of the end-effector to assure most of the targets attached on the end-effector being observed by the photogrammetry sensor. For the sake of simplicity, a

cuboid space as the workspace for the specific task is taken from the reachable workspace of the industrial robot. Therefore, the pose error $\boldsymbol{\nu}(k) = (\nu_x(k), \nu_y(k), \nu_z(k), \nu_\gamma(k), \nu_\beta(k), \nu_\alpha(k))$ can be computed as follows:

$$\begin{aligned} &\boldsymbol{\nu}(k) = \mathbf{p}_d - \mathbf{p}_c(k), \\ &\nu_x(k) = x_d - x(k), \quad \nu_y(k) = y_d - y(k), \quad \nu_z(k) = z_d - z(k), \\ &\nu_\gamma(k) = \gamma_d - \gamma(k), \quad \nu_\beta(k) = \beta_d - \beta(k), \quad \nu_\alpha(k)) = \alpha_d - \alpha(k). \end{aligned} \tag{4.2}$$

Since the pose error $\boldsymbol{\nu}(k)$ is represented in $\mathcal{F}_{U_T}$ in task space, which is completely different from the robot user frame $\mathcal{F}_{U_R}$. Both $\mathcal{F}_{U_T}$ and $\mathcal{F}_{U_{Eq}}$ are defined in $\mathcal{F}_S$. If $\boldsymbol{\nu}(k)$ is transformed into $\mathcal{F}_{U_{Eq}}$ as a vector $\boldsymbol{\varsigma}(k) = (\varsigma_x(k), \varsigma_y(k), \varsigma_z(k), \varsigma_\gamma(k), \varsigma_\beta(k), \varsigma_\alpha(k))$, the translational and rotational errors in $\mathcal{F}_{U_{Eq}}$ are the same as those in $\mathcal{F}_{U_R}$. Suppose the homogeneous transformation matrix of $\mathcal{F}_{U_T}$ and $\mathcal{F}_{U_{Eq}}$ in $\mathcal{F}_S$ is denoted as ${}^{S}_{U_T}\mathbf{H}$ and ${}^{S}_{U_{Eq}}\mathbf{H}$ respectively. The homogeneous transformation matrix of $\mathcal{F}_{U_T}$ in $\mathcal{F}_{U_{Eq}}$ is ${}^{U_{Eq}}_{U_T}\mathbf{H}$. The translational part of $\boldsymbol{\varsigma}(k)$ can be computed from $\boldsymbol{\nu}(k)$ as follows

$$\begin{aligned} {}^{U_{Eq}}_{U_T}\mathbf{H} &= {}^{S}_{U_{Eq}}\mathbf{H}^{-1}\, {}^{S}_{U_T}\mathbf{H}, \\ \begin{bmatrix} \varsigma_x(k) & \varsigma_y(k) & \varsigma_z(k) & 1 \end{bmatrix}^T &= {}^{U_{Eq}}_{U_T}\mathbf{H} \begin{bmatrix} \nu_x(k) & \nu_y(k) & \nu_z(k) & 1 \end{bmatrix}^T. \end{aligned} \tag{4.3}$$

For the rotational part of $\boldsymbol{\varsigma}(k)$, the equivalent angle-axis method in [144] is used. Generally, the rotational change is slower than translational change. The corresponding homogeneous matrix for the desired pose of the end-effector $\mathbf{p}_d$ in $\mathcal{F}_{U_T}$ is ${}^{U_T}_{d}\mathbf{H}$ while homogeneous matrix ${}^{U_T}_{r}\mathbf{H}$ represents the current TCP pose $\mathbf{p}_c(k)$ in $\mathcal{F}_{U_T}$. Therefore, the homogeneous matrix ${}^{r}_{d}\mathbf{H}$ that represents the transformation from current pose of the end-effector to the desired pose can be obtained as below

$${}^{r}_{d}\mathbf{H} = {}^{U_T}_{r}\mathbf{H}^{-1}\, {}^{U_T}_{d}\mathbf{H}, \tag{4.4}$$

where ${}^{r}_{d}\mathbf{R}$ (3×3 matrix) is the left upper part of ${}^{r}_{d}\mathbf{H}$. Based on ${}^{r}_{d}\mathbf{R}$, the equivalent angle ϕ and axis

${}^{r}\vec{K}$ can be derived. The equivalent axis ${}^{U_{Eq}}\vec{K}$ defined in $\mathcal{F}_{U_{Eq}}$ can be obtained as below:

$$\begin{aligned} {}^{U_{Eq}}_{r}\mathbf{H} &= {}^{U_{Eq}}_{U_T}\mathbf{H}\,{}^{U_T}_{r}\mathbf{H}, \\ {}^{U_{Eq}}\vec{K} &= {}^{U_{Eq}}_{r}\mathbf{R}\,{}^{r}\vec{K}, \end{aligned} \tag{4.5}$$

where 3×3 rotation matrix ${}^{U_{Eq}}_{r}\mathbf{R}$ is the left upper part of homogeneous matrix ${}^{U_{Eq}}_{r}\mathbf{H}$.

Corresponding to rotation operation ${}^{r}_{d}\mathbf{R}$, the equivalent angle in $\mathcal{F}_{U_{Eq}}$ is the same as ϕ. Therefore, rotation matrix ${}^{U_{Eq}}({}^{r}_{d}\mathbf{R})$ in $\mathcal{F}_{U_{Eq}}$ can be derived. On the other hand, rotation operation ${}^{U_{Eq}}({}^{r}_{d}\mathbf{R})$ can be realized by fixed angles method [144] as follows

$${}^{U_{Eq}}({}^{r}_{d}\mathbf{R}) = Rot_z(\varsigma_\alpha(k))Rot_y(\varsigma_\beta(k))Rot_x(\varsigma_\gamma(k)), \tag{4.6}$$

where Rot_x, Rot_y and Rot_z are the rotation matrix around x-axis, y-axis and z-axis of $\mathcal{F}_{U_{Eq}}$ with fixed angles $\varsigma_\gamma(k)$, $\varsigma_\beta(k)$ and $\varsigma_\alpha(k)$ respectively [144]. Since ${}^{U_{Eq}}({}^{r}_{d}\mathbf{R})$ is known, $\varsigma_\gamma(k)$, $\varsigma_\beta(k)$ and $\varsigma_\alpha(k)$ can be computed through Eq.(4.6). $\varsigma_\gamma(k)$, $\varsigma_\beta(k)$ and $\varsigma_\alpha(k)$ are the rotational part of $\boldsymbol{\varsigma}(k)$.

4.3.3 DPC Control Law Design

The general dynamic model of 6-DOF manipulators [145] can be described by

$$\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) + \mathbf{G}(\mathbf{q}(t)) = \boldsymbol{\tau}(t), \tag{4.7}$$

where $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, and $\ddot{\mathbf{q}}(t) \in \mathbf{R}^6$ represent the robot joint angles, velocity and acceleration; $\mathbf{M}(\mathbf{q}(t)) \in \mathbf{R}^{6\times 6}$ is a positive-definite, symmetric inertia matrix; $\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) \in \mathbf{R}^6$ denotes centrifugal and Coriolis torques; $\mathbf{G}(\mathbf{q}(t)) \in \mathbf{R}^6$ is the gravitational torques; $\boldsymbol{\tau}(t) \in \mathbf{R}^6$ denotes control torques.

Assume that $t = kT_c$. By using forward finite-difference for discretization, substitute $\dot{q}_i(t) = (q_i(k) - q_i(k-1))/T_c$ and $\ddot{q}_i(t) = (q_i(k+1) - 2q_i(k) + q_i(k-1))/T_c^2$ $(i = 1 \cdots 6)$ into Eq.(4.7), the discrete state equation is formulated as

$$\begin{bmatrix} \mathbf{q}(k) \\ \mathbf{q}(k+1) \end{bmatrix} = \begin{bmatrix} 0_{6\times6} & \mathbf{I}_{6\times6} \\ \boldsymbol{\delta}(k) & \boldsymbol{\xi}(k) \end{bmatrix} \begin{bmatrix} \mathbf{q}(k-1) \\ \mathbf{q}(k) \end{bmatrix} + T_c \begin{bmatrix} 0_{6\times6} \\ \boldsymbol{\zeta}(\mathbf{q}(k)) \end{bmatrix} \boldsymbol{\tau}(k) + \begin{bmatrix} 0_{6\times6} \\ -\boldsymbol{\zeta}(\mathbf{q}(k)) \end{bmatrix} \mathbf{g}(\mathbf{q}(k)). \tag{4.8}$$

where

$$\begin{aligned} \boldsymbol{\delta}(k) &= T_c \mathbf{M}^{-1}(\mathbf{q}(k)) \mathbf{C}(\mathbf{q}(k), \mathbf{q}(k-1)) - \mathbf{I}, \\ \boldsymbol{\xi}(k) &= 2\mathbf{I} - \mathbf{M}^{-1}(\mathbf{q}(k)) \mathbf{C}(\mathbf{q}(k), \mathbf{q}(k-1)), \\ \boldsymbol{\zeta}(k) &= T_c^2 \mathbf{M}^{-1}(\mathbf{q}(k)), \end{aligned} \tag{4.9}$$

$\mathbf{I}$ is 6×6 identity matrix.

The extraction of pose error $\boldsymbol{\varsigma}(k)$ is prepared for dynamic pose correction (DPC) module in Fig.4.2. DPC module is developed to generate control input for the robot controller so that six elements of $\boldsymbol{\varsigma}(k)$ converge to meet certain boundaries as follows

$$|\varsigma_x(k)|, |\varsigma_y(k)|, |\varsigma_z(k)| \leq \lambda_p, \quad |\varsigma_\gamma(k)|, |\varsigma_\beta(k)|, |\varsigma_\alpha(k)| \leq \lambda_o \tag{4.10}$$

where λ_p and λ_o are positive constants as expected accuracy for position and orientation respectively. It is required that the movement of the end-effector from $\mathbf{P}_0$ to $\mathbf{P}_d$ is continuous and smooth, and the control input is compatible with robot controller and satisfies the saturation conditions.

According to [144], Jacobian matrix $\boldsymbol{J}(\mathbf{q}(k))$ can be used to relate the velocities of the end-effector $\dot{\mathbf{p}}(k)$ and robot joints $\dot{\mathbf{q}}(k)$ as below:

$$\dot{\mathbf{p}}(k) = \boldsymbol{J}(\mathbf{q}(k)) \dot{\mathbf{q}}(k), \tag{4.11}$$

where $\dot{\mathbf{p}}(k)$ is represented in $\mathcal{F}_{U_R}$, and $\mathbf{q}(k)$ is the current joint position. When the changes of the joint angles are very small, based on Eq.(4.11), the following equations can be deduced as below:

$$\Delta\mathbf{p}(k) = \boldsymbol{J}(\mathbf{q}(k)) \Delta\mathbf{q}(k), \tag{4.12}$$

where $\Delta\mathbf{p}(k)$ denotes the small change of the pose of the end-effector in $\mathcal{F}_{U_R}$, and $\Delta\mathbf{q}(k)$ is the

corresponding change of the joint position. In this chapter, singularities are not considered. Consequently, the current joint error $\mathbf{e}(k) = \mathbf{q}_d(k) - \mathbf{q}(k)$, where $\mathbf{q}_d(k)$ is the desired joint position, can be obtained by

$$\mathbf{e}(k) = \boldsymbol{J}^{-1}(\mathbf{q}(k))\boldsymbol{\varsigma}(k). \tag{4.13}$$

$\mathbf{G} \in \mathbf{R}^6$ is the gravitational torques with desired pose $\mathbf{q}_d$. In the context of classic control methods for robot manipulators [146], a PD (Proportional-Derivative) control law incorporating a constant gravity torque is given as

$$\boldsymbol{\tau}(k) = \boldsymbol{K}_p\mathbf{e}(k) + \boldsymbol{K}_e(\mathbf{e}(k) - \mathbf{e}(k-1)) + \mathbf{G}, \tag{4.14}$$

where control gains $\boldsymbol{K}_p, \boldsymbol{K}_e \in R^{6\times 6}$ are positive definite diagonal matrices. Therefore, according to Eq.(4.14) and Eq.(4.14), the error dynamics can be obtained as below:

$$\begin{aligned}
\mathbf{e}(k+1) &= \mathbf{q}_d - \mathbf{q}(k+1), \\
&= \mathbf{q}_d - \boldsymbol{\delta}(k)\mathbf{q}(k-1) - \boldsymbol{\xi}(k)\mathbf{q}(k) - T_c\boldsymbol{\zeta}(\mathbf{q}(k))\boldsymbol{\tau}(k) + \boldsymbol{\zeta}(\mathbf{q}(k))\mathbf{g}(\mathbf{q}(k)) \\
&= (1 - \boldsymbol{\delta}(k) - \boldsymbol{\xi}(k))\mathbf{q}_d + \boldsymbol{\delta}(k)\mathbf{e}(k-1) + \boldsymbol{\xi}(k)\mathbf{e}(k) \\
&\quad - T_c\boldsymbol{\zeta}(\mathbf{q}(k))[\boldsymbol{K}_p\mathbf{e}(k) + \boldsymbol{K}_e(\mathbf{e}(k) - \mathbf{e}(k-1)) + \mathbf{G}] + \boldsymbol{\zeta}(\mathbf{q}(k))\mathbf{g}(\mathbf{q}(k)) \\
&= \boldsymbol{\eta}_1\mathbf{e}(k) + \boldsymbol{\eta}_2\mathbf{e}(k-1) + \boldsymbol{\eta}_3
\end{aligned} \tag{4.15}$$

where

$$\begin{aligned}
\boldsymbol{\eta}_1 &= \boldsymbol{\xi}(k) - T_c\boldsymbol{\zeta}(\boldsymbol{K}_p + \boldsymbol{K}_e) \\
\boldsymbol{\eta}_2 &= (\boldsymbol{\delta}(k) + T_c\boldsymbol{\zeta}\boldsymbol{K}_e)) \\
\boldsymbol{\eta}_3 &= (1 - \boldsymbol{\delta}(k) - \boldsymbol{\xi}(k))\mathbf{q}_d - T_c\boldsymbol{\zeta}\mathbf{G} + \boldsymbol{\zeta}(\mathbf{q}(k))\mathbf{g}(\mathbf{q}(k))
\end{aligned} \tag{4.16}$$

Define $f_1(k) = \mathbf{e}(k-1)$, $f_2(k) = \mathbf{e}(k)$, and $\mathbf{F}(k) = [f_1(k),\ f_2(k)]^T$, then one has

$$\mathbf{F}(k+1) = \mathbf{Z}\mathbf{F}(k) + \begin{bmatrix} 0 \\ \boldsymbol{\eta}_3 \end{bmatrix},\ \mathbf{Z} = \begin{bmatrix} 0 & 1 \\ \boldsymbol{\eta}_1 & \boldsymbol{\eta}_2. \end{bmatrix} \tag{4.17}$$

Therefore, if only $\boldsymbol{K}_p$, $\boldsymbol{K}_e$ are selected so that the eigenvalues of $\mathbf{Z}$ are within the unit circle.

Correspondingly, the error dynamics Eq.(4.15) is stable, and $\mathbf{F}(k)$ is bounded.

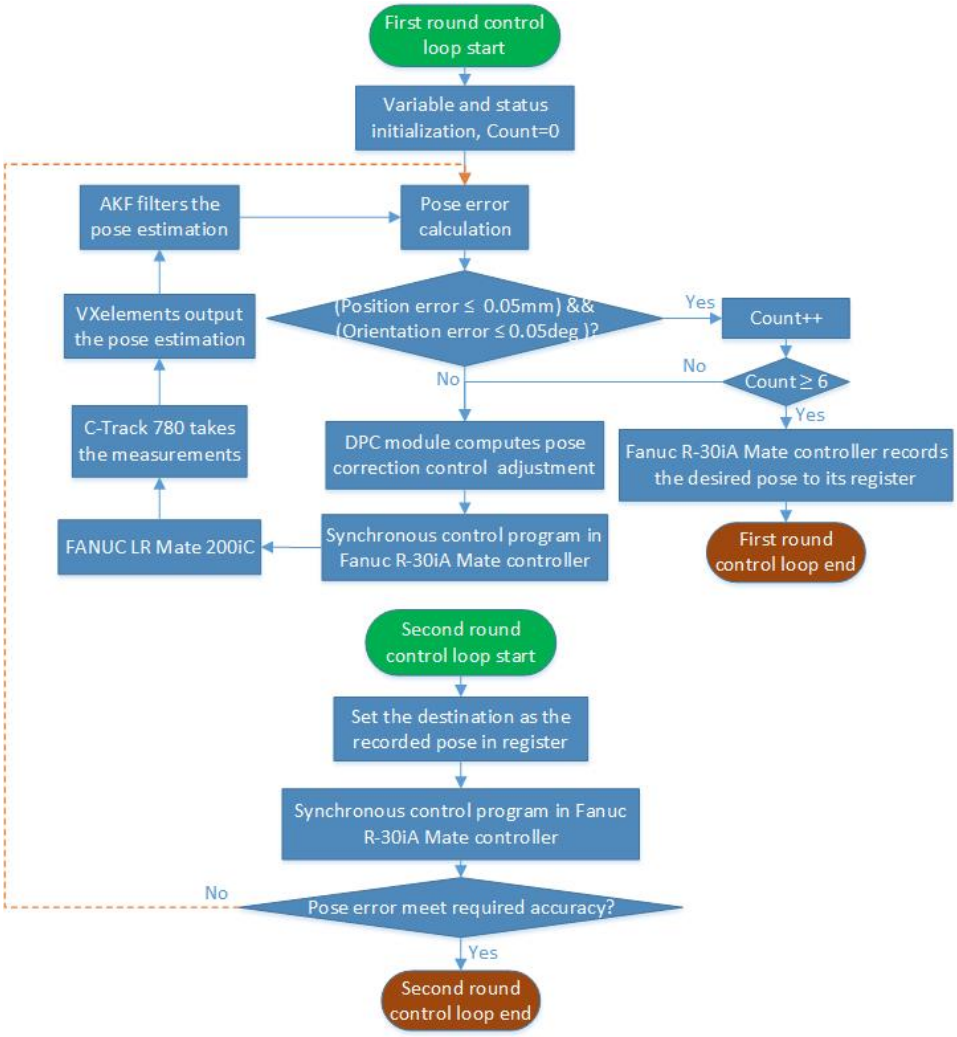


Figure 4.3: Control flowchart for implementing DPC Scheme.

4.4 Experiments for Implementing DPC Scheme

In this section, the implementation and verification of the DPC scheme are carried out on the 6-DOF manipulator, FANUC LR Mate 200iC by using the C-Track 780 as dynamic pose measuring instrument. The control flowchart of the implementation is demonstrated as Fig.4.3. At the first round, when the absolute value of pose error between current pose and desired pose has been completely meeting the preset thresholds, i.e., expected accuracy, for at least 6 times, it is regarded that

the end-effector arrives at the desired pose and the current pose is recorded in the position register by synchronous control program in Fanuc R-30iA Mate controller. Then at the second round, the control program in Fanuc R-30iA Mate controller can directly guide the end-effector to the recorded destination with high repeatability. All the late rounds will just repeat the second round. The required accuracy of most tasks can be definitely satisfied. Even if rigorous high accuracy is required, the procedures of DPC module following the dash-arrow line in Fig.4.3 can be run shortly to realize fine tuning.

4.4.1 Experimental Preparation

In Fig.4.4, there are FANUC LR Mate 200iC with an end-effector, a cube with 3D distributed reflectors, C-Track 780 on the tripod and a Trical mounted on the worktable. FANUC LR Mate 200iC comes with the Fanuc R-30iA Mate controller and is a 6-axis hollow wrist robot which has six revolute joints. The end-effector, as shown in Fig.4.5(a), is specially customized and composed of a plate with a head, a needle inserted through the hole of the head, a ball tool attached to the tip of the needle. A few of reflectors are spread on the plate and the head. The reflectors on the cube and on the base of FANUC LR Mate 200iC are selected to construct dynamic reference frame. C-Track 780 can observe the end-effector through all the reflectors in its FOV. The Trical was first introduced by Gaudreault et al. in 2016 [61]. The Trical, as shown in Fig.4.5(c), consists of three Mitutoyo digital indicators that are orthogonally-arranged and can be applied to measure the position of a precision ball. The TriCal can provide high-accuracy position measurements oweing to the high resolution of Mitutoyo digital indicator, which is $0.001mm$. In this research, the Trical is used to measure 3D position of the ball attached to the end-effector, as shown in Fig.4.5(b), and verify the position accuracy relative to the desired pose of the destination.

The C-Track 780, shown in Fig.4.4, is a dual-camera optical CMM system which can dynamically track the reference models including a rigid set of reflectors, and provide continuous image acquisition (3D measurements) transmitted to the control system in real time. The reflectors work as the feature points in the images. The repeatability and volumetric accuracy of the C-Track 780 is $0.0025mm$ and $0.065mm$ respectively. The repeatability of the C-Track 780 is the main factor for guiding the end-effector to the pre-planned destination.

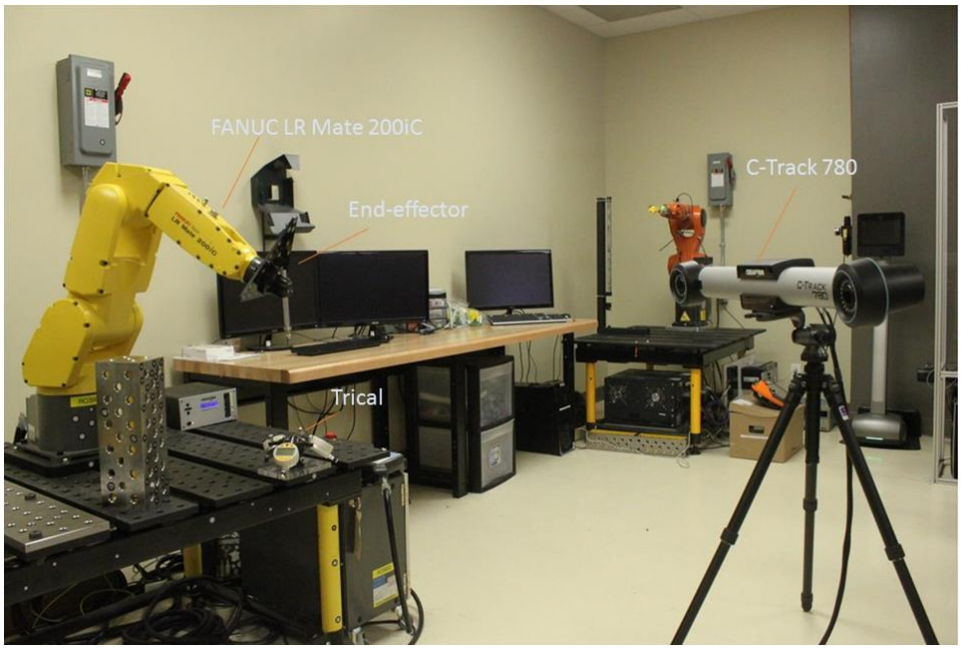


Figure 4.4: Experimental setup for implementing DPC Scheme.

In the experiment, the artificial end-effector is mounted at the center of flange on Joint 6 of FANUC LR Mate 200iC. A built-in 6-points method is provided by Fanuc R-30iA Mate controller to define a new tool frame on the tip of the end-effector, which is regarded as the TCP. Three points are used to identify TCP position, and three more points are used to define the orientation of the new tool frame $\mathcal{F}_E$ with respect to the original tool frame $\mathcal{F}_T$ can be opted as the current tool frame and recognized by the robot controller. Then, new user frame $\mathcal{F}_{U_R}$ can be also defined in robot base frame $\mathcal{F}_B$ through 3-points method provided by Fanuc R-30iA Mate based on the positions of $\mathcal{F}_E$.

On the other hand, $\mathcal{F}_E$ in sensor frame $\mathcal{F}_S$ of C-Track 780 can be defined in VXelements, and it must be identical to the one in $\mathcal{F}_B$ so that the TCP pose can be obtained from C-Track 780. As shown in Fig.4.5(a), more than 4 non-collinear reflectors stick on the end-effector. The end-effector model $Model_E$ is created by choosing these reflectors together in VXelements. Then, TCP is set as the center of $Model_E$ and $\mathcal{F}_E$ is attached to $Model_E$. Therefore, when C-Track is tracking $Model_E$, the pose of the end-effector is represented as the pose of $\mathcal{F}_E$ at TCP with respect to $\mathcal{F}_{U_T}$ in VXelements. In such case, the representation of TCP is identical to the definition of TCP in the robot controller. Moreover, the equivalent user frame $\mathcal{F}_{U_{Eq}}$ is necessary to be defined in VXelements so that when the current pose correction is defined in $\mathcal{F}_{U_{Eq}}$, it is a compatible adjustment for dynamic

pose correction of FANUC LR Mate 200iC in $\mathcal{F}_{U_R}$. Also, the pre-planned initial pose $\mathbf{P}_0$ and desired pose $\mathbf{p}_d$ of the end-effector are given by off-line method.

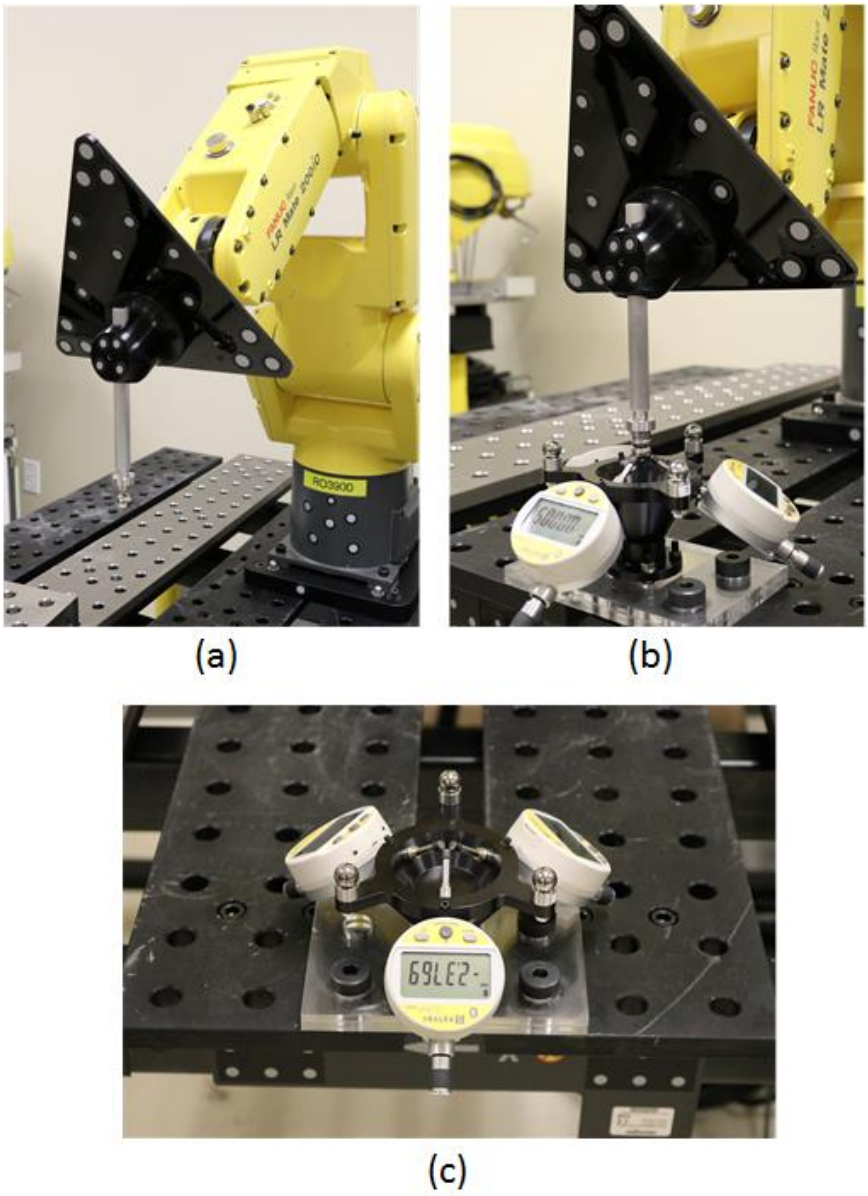


Figure 4.5: (a) FANUC LR Mate 200iC with the end-effector, (b) The ball tool on the needle tip touches three ball tips of the Trical, (c) The Trical.

In the experiment, DPC scheme is implemented as a software executive module integrating Funuc PCDK and VXelements to build the communication between Fanuc R-30iA Mate controller and the controller of C-Track 780. The control interval T_c should be larger than $34.48ms$ since the

maximum data updating frequency of C-Track 780 is $29Hz$. The interpolation interval (ITP) of FANUC LR Mate 200iC is $8ms$, the time cycle at which the destination is computed and sent to the motion environment. In the experiment, DPC module is executed on a high-performance computer with Intel Xeon Processor E5-1650 v3 3.5GHz and NVIDIA Quadro K2200 professional graphics board.

4.4.2 Control Parameters Initialization

The maximum updating frequency of C-Track 780, $29Hz$, is designated to obtain rapidest feedback measurements. Thus, the sampling period T_s for processing the measurements from C-Track 780 is $1/29$ second. Once the image information from C-Track 780 is updated, the pose estimation and AKF will work sequentially to obtain the pose information with minimum delay. Therefore, the pose updating rate is equal to $29Hz$. On the other hand, the control period T_c for DPT module is determined by considering T_s, ITP of Fanuc M20-iA and maximum control step. For the experiments in this research, T_c is set as $3T_s$. $\boldsymbol{K}_p$ is set as $[diag(0.12, 0.12, 0.12, 0.1, 0.1, 0.1), \mathbf{0}_{6\times6}]$, and $\boldsymbol{K}_e$ is set as $[diag(0.02, 0.02, 0.01, 0.01, 0.01, 0.01), \mathbf{0}_{6\times6}]$. The expected position accuracy λ_p and orientation accuracy λ_o are preset as $0.05mm$ and $0.05deg$ respectively.

4.4.3 Experimental results

In this subsection, the experimental results are recorded when the TCP of the end-effector is moving from $\mathcal{P}_0$ to $\mathcal{P}_d$ with the complementary control of DPC module. some data analyses are conducted on the saved experimental results. In order to better evaluate the position accuracy, distance error is defined as $Distance(k) = sqrt((\nu_x(k))^2, (\nu_y(k))^2$. Take the results of one typical experiment as an example, distance error, position errors and orientation errors during the movement of the end-effector from $\mathcal{P}_0$ to $\mathcal{P}_d$ are demonstrated by Fig.4.6$\sim$4.8 at speed $25mm/s$ respectively. The control interval T_c is set The time duration of this round is $3.83s$. The initial error is computed as $\mathcal{P}_d - \mathcal{P}_0$, and the final error is obtained as $\mathcal{P}_d$ minus the final pose where pose accuracy is satisfied and the TCP of the end-effector stops moving. The initial error and final error of this experiment are shown in Table.4.1. It can be observed that the positional and oriental part of the final error are both much less than the expected accuracy, $0.05mm$ and $0.05deg$, and even the final distance error

is just $0.023mm$. In fact, according to Fig.4.7 and 4.8, position errors and orientation errors are continuously convergent during the re-checking of 6 times after the first time when the decreasing errors meet the expected accuracy. On the other hand, the Trical is used for verifying the accuracy of this experiment at the same time. Before this experiment is started, the end-effector is moved so that the ball tool on the needle tip of the end-effector touches three ball tips of the Trical and the pose of the end-effector is saved as the desired pose by VXelements. Then, three Trical indicators are reset for initializing the data display as the current status, shown in Fig.4.9 (a1) and (a2). Fig.4.9 (b1) and (b2) demonstrate the data display of three Trical indicators at the first round and Fig.4.9 (c1) and (c2) are for the second round. The specific data displayed on the three Trical indicators for initialization and two rounds are presented in Table.4.2 and the maximum value is $0.007mm$. Moreover, the corresponding distance errors for two rounds are less than $0.008mm$. Therefore, the position accuracy of this experiment with DPC control is validated through the measurements of the high-revolution Trical.

In order to investigate the reliability and repeatability of the DPC scheme, the first round of this experiment in Fig.4.3 has been repeated many times for data analysis. Table.4.4 presents the pose accuracy and time duration of 6 tests. The maximum position error and orientation error are less than $0.03mm$ and $0.015deg$ respectively. Moreover, the time duration of each test is very stable and it is around $3.8s$. The first round of this experiment with different T_c has been tested as well. Table.4.4 demonstrates the durations of 9 tests with 3 types of T_c, such as $2T_s$, $3T_s$ and $4T_s$. Relatively speaking, the control process with control interval $4T_s$ is more steady and smooth, andthe control process with control interval $3T_s$ converges faster. However, for control interval $2T_s$ there are overshoots and vibration during the process.

There are some important progresses in the research work presented in this chapter compared to the research work presented in [25]. First, adaptive Kalman filter is more effective filtering method than RMS average method used in [25] on data smoothness and real-time performance. RMS average method is impossible to provide the real-time pose information. Also its corresponding control is not continuous due to intermittent feedback pose information from RMS average. There are always vibrations especially when the pose error is large. The second progress is the control strategy design of DPC including specific configuration limits and derivative status which are not considered

in [25]. Last, better control performance and results in this research are obtained, such as faster and smoother control process, even better pose accuracy.

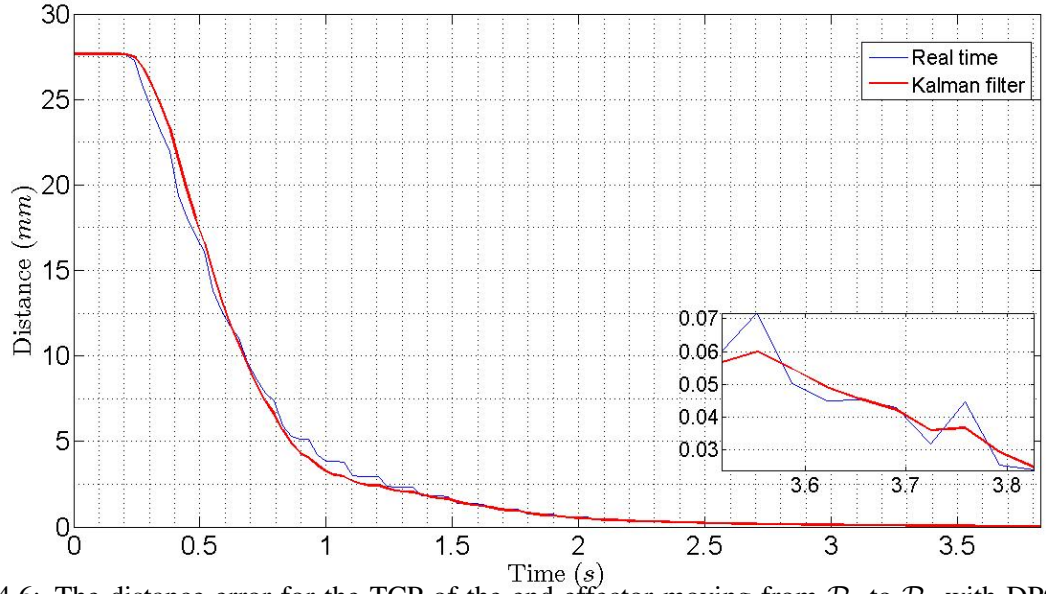


Figure 4.6: The distance error for the TCP of the end-effector moving from $\mathcal{P}_0$ to $\mathcal{P}_d$ with DPC control.

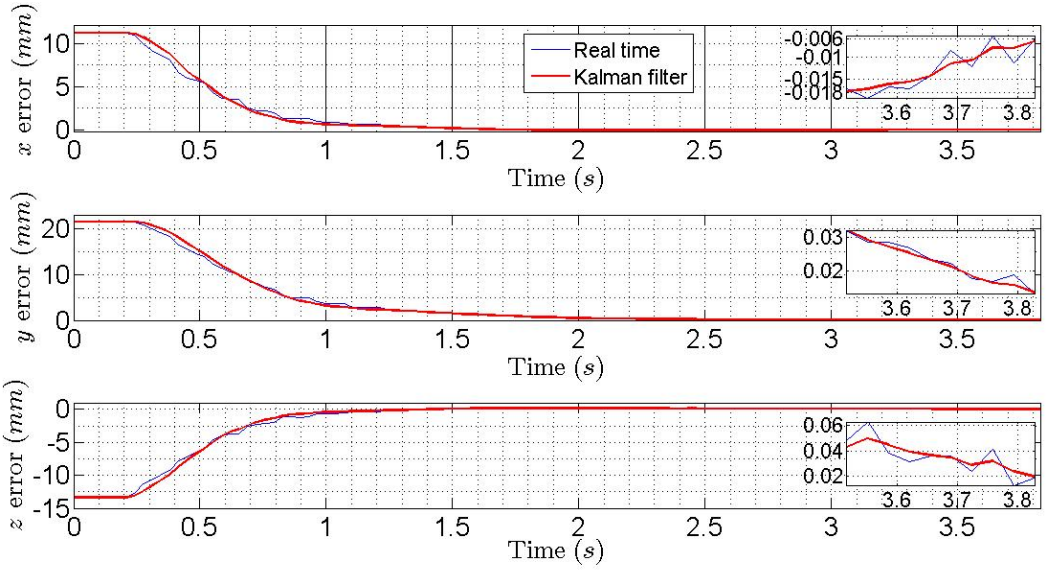


Figure 4.7: The position error of $\mathcal{F}_E$ in $\mathcal{F}_{U_T}$ for the TCP of the end-effector moving to the desired pose with DPC control.

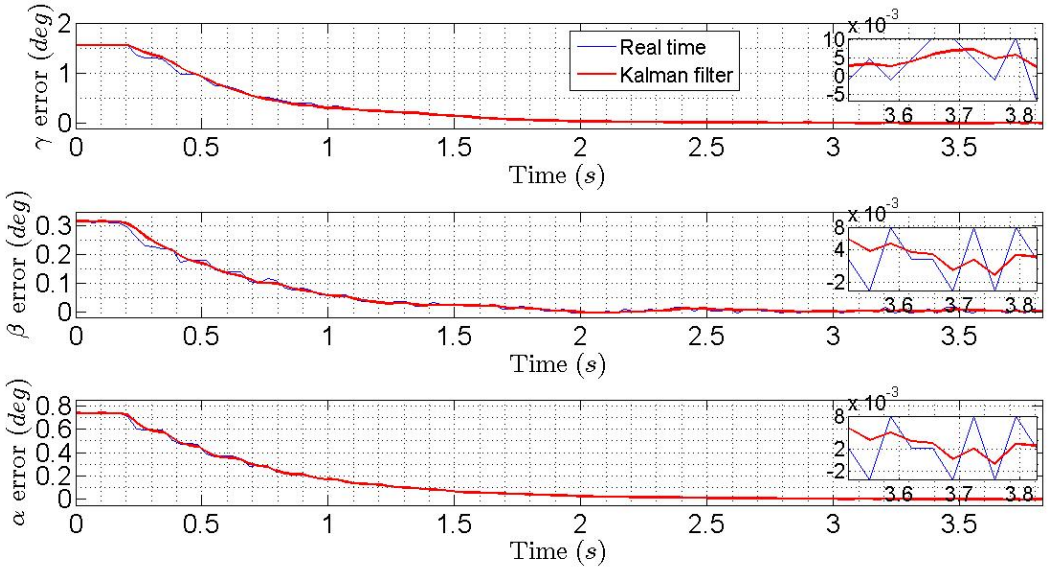


Figure 4.8: The orientation error of $\mathcal{F}_E$ in $\mathcal{F}_{U_T}$ for the TCP of the end-effector moving from $\mathcal{P}_0$ to $\mathcal{P}_d$ with DPC control.

Table 4.1: The comparison of initial error and final error.

Item / Type	Pose error					
	$x(mm)$	$y(mm)$	$z(mm)$	$\gamma(deg)$	$\beta(deg)$	$\alpha(deg)$
Initial error	11.19	21.45	-13.38	1.55	0.32	0.73
Final error	-0.006	0.019	0.013	-0.007	0.002	0.002

Table 4.2: Data display of three Trical indicators.

Round No.	Data display (mm)			Distance error (mm)
	Indicator 1	*Indicator* 2	*Indicator* 3	
Round 0 (initial)	0.004	0.002	0.000	0.0045
Round 1	0.007	-0.003	0.002	0.0079
Round 2	0.005	-0.005	0.004	0.0073

4.5 Conclusion

A practical learning based DPC scheme is proposed in this paper. Compared to the conventional way, the proposed DPC scheme becomes less dependent on the robot controller and more efficient.

The robot controller only works as the movement command channel between the control computer and the robot. Any pre-planned 6D task point in task space can be recorded by VXelements and reached accurately by the TCP of the end-effector with the pose feedback from visual measurement instrument. Moreover, a adaptive Kalman filter is proposed to process the pose estimation of the end-effector. The experimental results demonstrate that the proposed DPC scheme can improved the pose accuracy to be less than $\pm 0.05mm$ and $\pm 0.05deg$. At the end of the first round, the desired pose can be saved as the pose of the tool frame in $\mathcal{F}_{U_R}$. For the second round and late rounds, the task program can run in the robot controller independently or obtain fast adjustment to satisfy the required accuracy. The learning based DPC scheme can also work as on-line teaching to transfer the off-line planned task to the information recorded in the position register of robot controller. In the furture, the proposed DPC scheme will be tested on the other types of industrial robots such as KUKA and ABB.

Table 4.3: Pose accuracy and duration of 6 tests for the TCP of the end-effector moving from $\mathcal{P}_0$ to $\mathcal{P}_d$ with DPC control.

Performance / Test No.	Pose Accuracy						Duration
	$x(mm)$	$y(mm)$	$z(mm)$	$\gamma(deg)$	$\beta(deg)$	$\alpha(deg)$	$t(s)$
1	-0.006	0.019	0.013	-0.007	0.002	0.002	3.828
2	-0.007	0.013	0.012	0.001	0.003	0.004	3.724
3	-0.005	0.010	0.024	0.001	0.002	-0.001	3.759
4	-0.011	0.020	0.027	0.003	0.003	0.004	3.724
5	-0.013	0.021	0.015	0.014	-0.004	-0.001	3.793
6	-0.010	0.017	0.025	0.005	-0.002	-0.001	3.828

Table 4.4: Time durations for the TCP of the end-effector moving from $\mathcal{P}_0$ to $\mathcal{P}_d$ with DPC control at different T_c .

T_c / t (s)	Control interval								
	$2T_s$	$2T_s$	$2T_s$	$3T_s$	$3T_s$	$3T_s$	$4T_s$	$4T_s$	$4T_s$
Duration	2.52	3.0	3.0	2.65	2.96	2.86	3.76	3.83	3.83

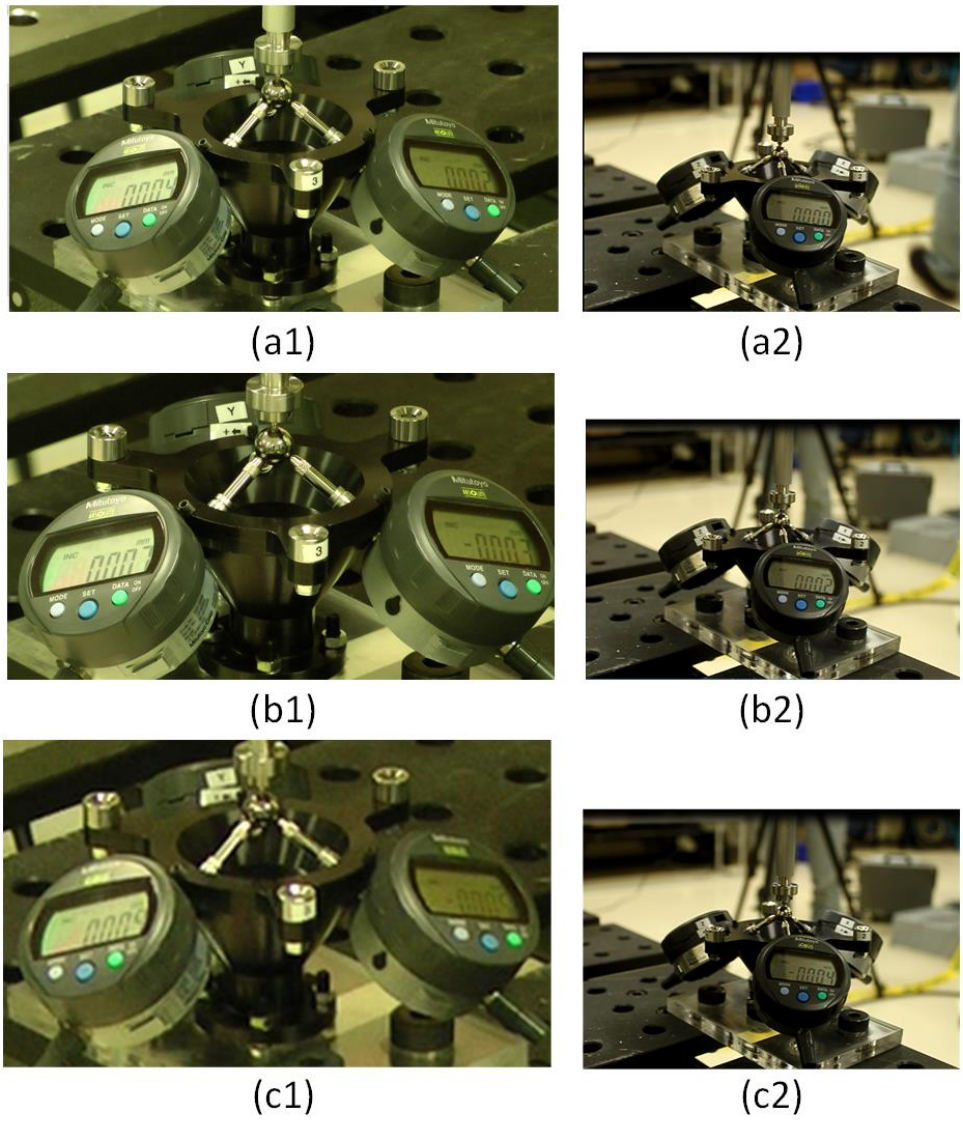


Figure 4.9: (a1) and (a2) Data display of three Trical indicators after initialization; (b1) and (b2) Final data display of three Trical indicators at the first round; (c1) and (c2) Final data display of three Trical indicators at the second round.

Chapter 5

Dynamic Path Tracking of Industrial Robots With High Accuracy Using Photogrammetry Sensor

5.1 Introduction

In industrial manufacturing fields, many tasks, such as cutting, milling, lathing, are expected to use robots to implement operation automatically. According to the standard process specifications in aerospace industry (e.g, Airbus [9]), the desired accuracy of robot manipulation for manufacturing is normally around $\pm 0.20mm$. However, due to the mechanical tolerances and deflection in the robot structure, the typical difference of a virtual robot in simulation and a real robot can be $8 - 15mm$ [147], which is difficult to meet the precision requirement of many potential applications. Therefore, the relatively low accuracy of current robots is the main problem for the industrial manufacturing applications. Especially, it poses a critical obstacle to use advanced task planning techniques which integrate off-line simulation and CAD-based methods.

In Chapter 4, an effective dynamic pose correction (DPC) scheme has been developed to enhance pose reaching accuracy for satisfying both position and orientation precision requirement for industrial robots. In a robot application where a high-precision tracking of various trajectories is

demanded, dynamic continuous strategies for enhancing real-time tracking accuracy become necessary, while an eye-to-hand PBVS scheme is the main approach. In this chapter, a dynamic path tracking (DPT) aims at correcting the end-effector's pose in the industrial applications by using a photogrammetry sensor.

In PBVS, the pose estimation of the robot is a challenging task and needs a good camera calibration and object model. Also, the pose measurements are computed from the current image information of the feature points, which tend to include distortion, blur and other uncertain noise. In Chapter 3, pose estimation by using AKF for pose correction is presented. In this chapter, the 3D pose is extracted from model-based image information measured by C-Track 780 deployed in eye-to-hand configuration and used for tracking 3D path in workspace. The purpose is to control the tool center point (TCP) of the end-effector on industrial robots to follow the path off-line preplanned in Cartesian space with respect to specific task accurately. In the current robot operation practice, the task path is taught manually by moving the TCP in the workspace and recording the position of finite intermediate points. Moreover, the low accuracy and repeatability of the robot can not ensure the TCP to track the task path with high accuracy. Apparently, when the task path is complex and requires high-precision, such method is hard to meet the strict high-precision requirements. In this chapter, a general DPT scheme is proposed by using the low cost photogrammetry sensor, C-Track 780 from Creaform to improve the accuracy of industrial robots. DPT can work in the control computer, which is connected with robot controller and visual sensor, $i.e.$ C-Track 780 by Ethernet. DPT is considered to compensate the calibration error, vibration error, uncertain model error as well as the repeatability error. The implementation of the DPT scheme is more flexible and convenient with the guaranteed high accuracy compared with the other strategies such as retrofitting with high-end encoders or using high cost laser tracker in static calibration. The task path can be produced by optimal path planning algorithm in advance and be saved in the computer which can communicate with the robot controller. The successful applications on FANUC M20-iA and FANUC LR Mate 200iC with C-Track 780 as visual measurement sensor demonstrate the effectiveness of the proposed DPT scheme.

The rest of this chapter is organized as the following. In Section 5.2, the workspace description and problem statement are provided. Section 5.3 introduces the control configuration of the DPT

scheme, tracking control law design and stability analysis. The simulation results on Puma 560 are provided to prove the robustness of the DPT scheme. The experimental results on FANUC M20-iA with C-track 780 as pose measurement instrument are presented to confirm the effectiveness of the DPT scheme in Section 5.4. Finally, the concluding remarks and future works are summarized in Section 5.5.

5.2 Workspace Description and Problem Statement

5.2.1 Workspace Description

A pre-planned task path in workspace is demonstrated as Fig.5.1. In order to set up a relation between $\mathcal{F}_{U_R}$ and $\mathcal{F}_{U_T}$, another equivalent user frame $\mathcal{F}_{U_{Eq}}$ is introduced. Ideally, there is only translation and no rotation between $\mathcal{F}_{U_{Eq}}$ and $\mathcal{F}_{U_R}$ in workspace. The main difference is that $\mathcal{F}_{U_R}$ is represented in $\mathcal{F}_B$ while $\mathcal{F}_{U_{Eq}}$ in $\mathcal{F}_S$. Frame $\mathcal{F}_{U_{Eq}}$ is defined by selecting one point in $\mathcal{W}_T$ as the origin of $\mathcal{F}_{U_{Eq}}$ and taking measurements of two points by visual measurement instrument when moving the end effector along X and Y axis of $\mathcal{F}_{U_R}$. Then, $\mathcal{F}_{U_{Eq}}$ in $\mathcal{F}_S$ can be computed by using these three points. The definition and relation of the above-mentioned coordinate frames are illustrated in Fig.5.1.

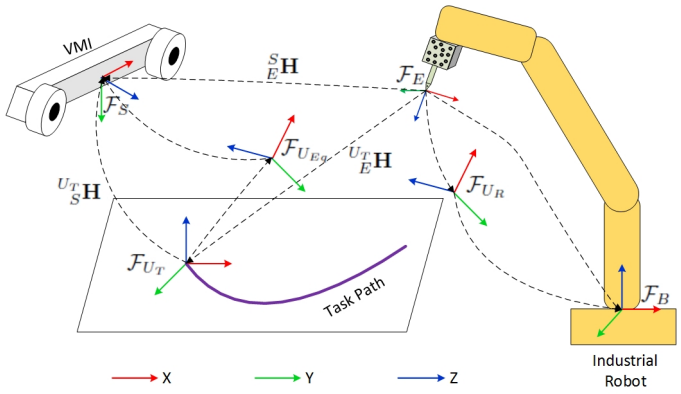


Figure 5.1: Definition and relation of the coordinate reference frames in the workspace.

5.2.2 Problem Statement

When the industrial robots are designated to do cutting, painting, and fiber placement in aerospace industry, the corresponding finite task path with start point and end point should be planned in advance. The robots are expected to move from the start point to end point strictly following the task path with high accuracy.

Traditionally, the task path is planned in robot user frame $\mathcal{F}_{U_R}$ by using teach-pendant to record the point nodes or to input the point nodes manually. Between the position nodes, the path is produced by using certain interpolation of several existing interpolation methods in robot controller. The task path defined in task user frame $\mathcal{F}_{U_T}$ is continuously differentiable and independent of the robot controller so that the task path can be pre-planned in task space $\mathcal{W}_T$. Any task path $\mathcal{P}$ in $\mathcal{F}_{U_T}$ can be described as follows

$$\begin{aligned} p_x &= x(\eta), \quad p_y = y(\eta), \quad p_z = z(\eta), \\ p_\gamma &= \gamma(\eta), \quad p_\beta = \beta(\eta), \quad p_\alpha = \alpha(\eta), \end{aligned} \tag{5.1}$$

where η is a normalized variable, and $\eta \in [0, 1]$. $\eta = 0$ corresponds to the start point on the task path $\mathcal{P}$ while $\eta = 1$ corresponds to the end point on $\mathcal{P}$. During $0 < \eta < 1$, continuous position (p_x, p_y, p_z) forms the geometry profile of $\mathcal{P}$ in $\mathcal{F}_{U_T}$ while $(p_\gamma, p_\beta, p_\alpha)$ is the orientation of $\mathcal{F}_E$ in $\mathcal{F}_{U_T}$. The task path $\mathcal{P}$ defined in Eq.(5.1) is served as the desired path. In order to carry out the task, the control objective is to control the TCP to track (p_x, p_y, p_z) along the desired path with $(p_\gamma, p_\beta, p_\alpha)$ satisfying the error tolerance. The pose of the end-effector can be estimated from the measurements of visual measurement instrument as the algorithm presented in Subsection 3.2.2 of Chapter 3. Moreover, the adaptive Kalman filter proposed in Section.3.3 of Chapter 3 can be applied to remove the noise and obtain the precise pose and velocity information from the above pose estimation.

5.3 Dynamic Path Tracking Control

Since the task path $\mathcal{P}$ is not planned in $\mathcal{F}_{U_R}$ or $\mathcal{F}_B$, it is not compatible in robot controller. Besides, only finite separate points, not consecutive geometry curve, can be imported into robot

controller. Most importantly, the low accuracy and repeatability of the robot can not guarantee the TCP of the end-effector to track the task path accurately with the robot controller.

In this chapter, PBVS-based dynamic path tracking (DPT) scheme is presented to realize the control configuration consists of four parts shown in Fig.5.2, which is similar to Fig.4.2. One difference is path analyzer, which computes the pose of the closest point on the desired task path according to the current TCP pose, and decides the next path step. The other is the path tracking control, which includes control algorithm to produce the control input for robot controller based on the current pose error. The algorithm of path tracking control is fundamentally identical to the algorithm of the DPC module in Chapter 4. As for filtering part, adaptive Kalman filter is augmented with acceleration estimation to rapidly adapt to the varying desired pose during the path tracking movement of the end-effector.

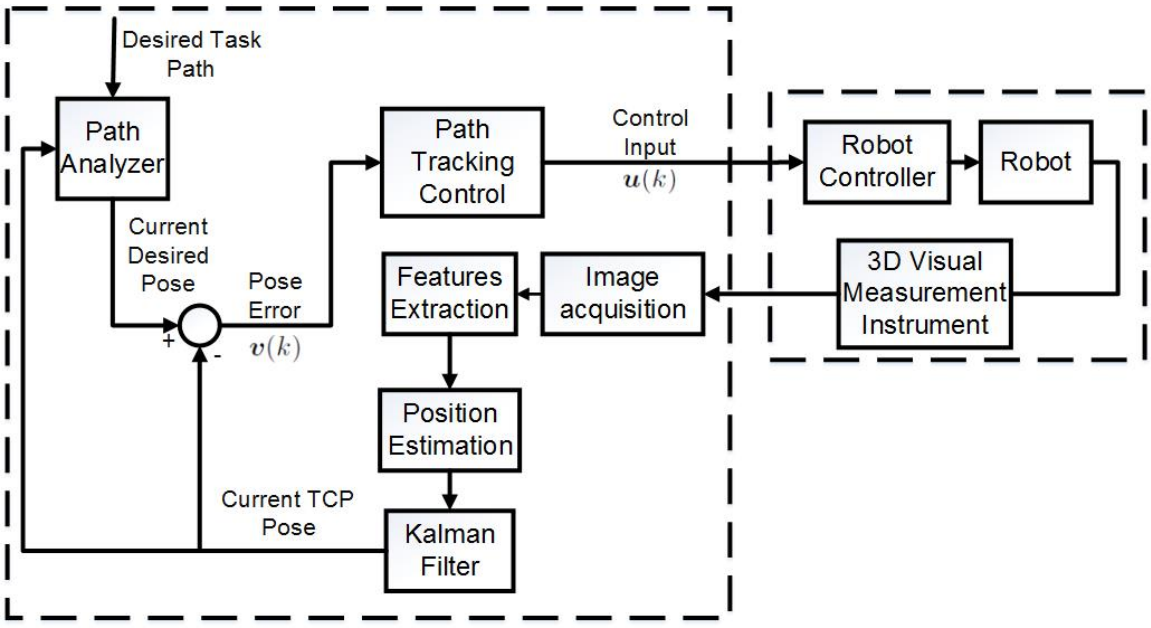


Figure 5.2: Control configuration of dynamic path tracking control.

5.3.1 Pose Error Computation in Equivalent User Frame

When the end-effector is in the FOV of the viusal measurement instrument, the current pose information of the end-effector in $\mathcal{F}_S$ can be obtained as where $\mathcal{F}_E$ is in $\mathcal{F}_S$. Assume η_1 and η_2 are both in $(0, 1)$ and $\eta_2 > \eta_1$. if $\Delta\eta = \eta_2 - \eta_1$ is small enough, the curve segment on the task path

$\mathcal{P}$, described in $Eq.(5.1)$, can be regarded as a straight line segment approximately. During each path segment $\Delta\eta$, the pose information $(x(\eta_2), y(\eta_2), z(\eta_2), \gamma(\eta_2), \beta(\eta_2), \alpha(\eta_2))$ of $\mathcal{P}$ at η_2 is the current desired pose. A path analyzing method shown in Fig.5.3 can calculate the pose error and decide compensation input of the next step. In this chapter, a fixed control interval is used, which is denoted as T_c and $T_c = \varrho\Delta\eta$, where ϱ is positive and relevant to the tracking speed. At the end of each T_c, the current TCP pose and velocity estimation, denoted as $\hat{\boldsymbol{\rho}}_{k,k}$, is obtained from the output of Kalman filter in Fig.5.2.

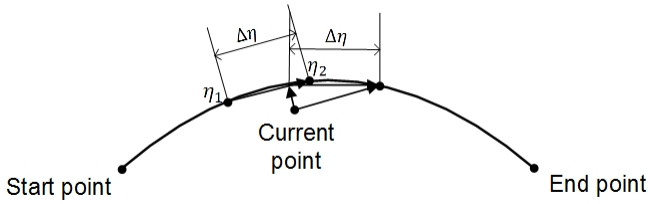


Figure 5.3: Path analyzing method of dynamic path tracking control.

Given the segment line between (η_1, η_2) and the current pose, it is simple to obtain the pose $\mathbf{p}_d(k)$ of the closest point from current point to the line. Moreover, $\mathbf{p}_c(k)$ is used to denote the pose part of $\hat{\boldsymbol{\rho}}_{k,k}$. Therefore, the pose error $\boldsymbol{\nu}(k) = (\nu_x(k), \nu_y(k), \nu_z(k), \nu_\gamma(k), \nu_\beta(k), \nu_\alpha(k))$ can be computed as follows:

$$\boldsymbol{\nu}(k) = \mathbf{p}_d(k) - \mathbf{p}_c(k). \tag{5.2}$$

Then, the pose error $\boldsymbol{\nu}(k)$ represented in $\mathcal{F}_{U_T}$ needs to be transfered pose error $\boldsymbol{\varsigma}(k)$ in $\mathcal{F}_{U_{Eq}}$ which axes parallel to the robot user frame $\mathcal{F}_{U_R}$. The procedures to obtain the positional part and rotational part of $\boldsymbol{\varsigma}(k)$ is in the same way introduced in subsection 4.3.2 of Chapter 4.

5.3.2 Tracking Control Law Design

Let $\mathbf{s}(k) = [\mathbf{q}(k-1)^T, \mathbf{q}(k)^T]^T$. Then according to Eq.(4.8), the state equation can be written as

$$\mathbf{s}(k+1) = \mathbf{F}(k)\mathbf{s}(k) - \mathbf{Y}(k) + \mathbf{O}(k)\boldsymbol{\tau}(k), \tag{5.3}$$

where

$$\mathbf{F}(k) = \begin{bmatrix} 0_{6\times6} & \mathbf{I}_{6\times6} \\ \boldsymbol{\delta}(k) & \boldsymbol{\xi}(k) \end{bmatrix}, \ \mathbf{Y}(k) = \begin{bmatrix} 0_{6\times6} \\ \boldsymbol{\zeta}(\mathbf{q}(k)) \end{bmatrix} \mathbf{g}(\mathbf{q}(k)),$$
$$\mathbf{O}(k) = T_c \begin{bmatrix} 0_{6\times6} \\ \boldsymbol{\zeta}(\mathbf{q}(k)) \end{bmatrix}. \tag{5.4}$$

Both the translational and rotational part of pose error $\boldsymbol{\varsigma}(k)$ are prepared for path tracking control block in Fig.5.2. The objective of path tracking control block is to produce control input for the robot controller so that all the six elements of $\boldsymbol{\varsigma}(k)$ converge to a certain boundary to meet the precision requirement. It is required that the TCP movement along the task path $\mathcal{P}$ is smooth, and the control input is compatible with robot controller and satisfies the saturation conditions.

According to Eq.(4.12), the current joint error $\tilde{\mathbf{q}}(k) = \mathbf{q}_d(k) - \mathbf{q}(k)$, where $\mathbf{q}_d(k)$ is the desired joint position, can be obtained by

$$\tilde{\mathbf{q}}(k) = \boldsymbol{J}^{-1}(\mathbf{q}(k))\boldsymbol{\varsigma}(k). \tag{5.5}$$

$\mathbf{s}_d(k) = [\mathbf{q}_d(k-1)^T, \mathbf{q}_d(k)^T]^T$ denotes the desired state at time instant k. Since the task path $\mathcal{P}$ is continuously differentiable, an assumption can be given as

$$\mathbf{s}_d(k+1) = \mathbf{s}_d(k) + \mathbf{f}(k), \tag{5.6}$$

where $\mathbf{f}(k) \in R^6$ and each element of it is a bounded function. $\mathbf{e}(k)$ denotes the state error at time k as $\mathbf{e}(k) = \mathbf{s}_d(k) - \mathbf{s}(k) = [\tilde{\mathbf{q}}(k-1)^T, \tilde{\mathbf{q}}(k)^T]^T$. According to Eq.(5.5), $\mathbf{e}(k)$ can be derived as:

$$\mathbf{e}(k) = \mathbf{Z}(k)[\boldsymbol{\varsigma}(k-1)^T, \boldsymbol{\varsigma}(k)^T]^T,$$
$$\mathbf{Z}(k) = \begin{bmatrix} \boldsymbol{J}^{-1}(\mathbf{q}(k-1)) & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \boldsymbol{J}^{-1}(\mathbf{q}(k)) \end{bmatrix}. \tag{5.7}$$

A control law is designed to stabilize closed-loop system as follows

$$\boldsymbol{\tau}(k) = \mathbf{h}(k) + \boldsymbol{\Gamma}\mathbf{e}(k) + \boldsymbol{\Lambda}(\mathbf{e}(k) - \mathbf{e}(k-1)), \tag{5.8}$$

where $\boldsymbol{\Lambda} = [diag[\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4, \Lambda_5, \Lambda_6], \mathbf{0}_{6\times6}]$ and $\boldsymbol{\Gamma} = [diag[\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5, \Gamma_6], \mathbf{0}_{6\times6}]$ are control gain matrices; $\mathbf{O}(k)\mathbf{h}(k) = \mathbf{Y}(k)$. Λ_i and Γ_i, $i = 1 \cdots 6$, are positive constants. Therefore, according to Eq.(5.3) , Eq.(5.6), and Eq.(5.8), $\mathbf{e}(k+1)$ can be obtained as below:

$$\begin{aligned}
\mathbf{e}(k+1) &= \mathbf{s}_d(k+1) - \mathbf{s}(k+1), \\
&= \mathbf{s}_d(k) + \mathbf{f}(k) - \mathbf{F}(k)\mathbf{s}(k) + \mathbf{Y}(k) - \mathbf{O}(k)\boldsymbol{\tau}(k), \\
&= \mathbf{s}_d(k) + \mathbf{f}(k) - \mathbf{F}(k)\mathbf{s}(k) - \mathbf{O}(k)\boldsymbol{\Gamma}\mathbf{e}(k) \\
&\quad - \mathbf{O}(k)\boldsymbol{\Lambda}(\mathbf{e}(k) - \mathbf{e}(k-1)), \\
&= (\mathbf{I} - \mathbf{F}(k))\mathbf{s}_d(k) + \mathbf{f}(k) + (\mathbf{F}(k) - \mathbf{O}(k)\boldsymbol{\Gamma} \\
&\quad - \mathbf{O}(k)\boldsymbol{\Lambda})\mathbf{e}(k) + \mathbf{O}(k)\boldsymbol{\Lambda}\mathbf{e}(k-1), \\
&= \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2\mathbf{e}(k),
\end{aligned} \tag{5.9}$$

where

$$\begin{aligned}
\boldsymbol{\xi}_1 &= (\mathbf{I} - \mathbf{F}(k))\mathbf{s}_d(k) + \mathbf{f}(k) + \mathbf{O}(k)\boldsymbol{\Lambda}\mathbf{e}(k-1), \\
\boldsymbol{\xi}_2 &= \mathbf{F}(k) - \mathbf{O}(k)\boldsymbol{\Gamma} - \mathbf{O}(k)\boldsymbol{\Lambda}.
\end{aligned} \tag{5.10}$$

Assume that at previous time instant $k-1$, the state error $\mathbf{e}(k-1)$ is bounded. Moreover, $\mathbf{s}_d(k)$ and $\mathbf{f}(k)$ are known to be bounded. Therefore, $\boldsymbol{\xi}_1$ is bounded. $\boldsymbol{\xi}_2$ is mainly related to control matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$.

5.3.3 Stability Analysis

In this section, the stability of the proposed control law is analyzed by using Lyapunov funcion. It is assumed the task path $\mathcal{P}$ is reachable for the robot and visible in the FOV of the VMI. Since only 6-DOF industrial robot is considered in this chapter, the robot is kinematically non-redundant so that the Jacobian matrix $\boldsymbol{J}(\mathbf{q}(k))$ can satisfy non-singularity along $\mathcal{P}$.

The Lyapunov function candidate is defined as

$$V(k) = \mathbf{e}(k)^T\mathbf{e}(k), \tag{5.11}$$

where $V(k) \geq 0$ and $V(k) = 0$ only if $\mathbf{e}(k) = 0$.

The time difference of Eq.(5.11) can be formed as

$$\begin{aligned}
&\Delta V(k+1) = V(k+1) - V(k) \\
&= \mathbf{e}(k+1)^T\mathbf{e}(k+1) - \mathbf{e}(k)^T\mathbf{e}(k) \\
&= (\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2\mathbf{e}(k))^T(\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2\mathbf{e}(k)) - \mathbf{e}(k)^T\mathbf{e}(k) \\
&\leq 2\boldsymbol{\xi}_1^T\boldsymbol{\xi}_1 + 2(\boldsymbol{\xi}_2\mathbf{e}(k))^T(\boldsymbol{\xi}_2\mathbf{e}(k)) - \mathbf{e}(k)^T\mathbf{e}(k) \\
&= 2\boldsymbol{\xi}_1^T\boldsymbol{\xi}_1 + \mathbf{e}(k)^T(2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2 - \mathbf{I})\mathbf{e}(k),
\end{aligned} \tag{5.12}$$

where by tuning the parameters of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$, $2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2 - \mathbf{I}$ can be negative definite. Therefore, $\Delta V(k+1) \leq 0$ is satisfied as long as the following inequality is satisfied:

$$2\boldsymbol{\xi}_1^T\boldsymbol{\xi}_1 \leq \mathbf{e}(k)^T(\mathbf{I} - 2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2)\mathbf{e}(k) \leq \|\mathbf{I} - 2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2\|\|\mathbf{e}(k)\|^2. \tag{5.13}$$

Correspondingly,

$$\|\mathbf{e}(k)\|^2 \geq \frac{2\boldsymbol{\xi}_1^T\boldsymbol{\xi}_1}{\|\mathbf{I} - 2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2\|} = \frac{2\|\boldsymbol{\xi}_1\|^2}{\|\mathbf{I} - 2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2\|}. \tag{5.14}$$

Then,

$$\|\mathbf{e}(k)\| \geq \frac{\sqrt{2}\|\boldsymbol{\xi}_1\|}{\sqrt{\|\mathbf{I} - 2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2\|}}. \tag{5.15}$$

From Eq.(5.15), it is clear that $\|\mathbf{e}(k)\|$ is bounded by $\frac{\sqrt{2}\|\boldsymbol{\xi}_1\|}{\sqrt{\|\mathbf{I}-2\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2\|}}$. Also, by choosing suitable control parameters of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$ such that $\boldsymbol{\xi}_2^T\boldsymbol{\xi}_2$ is small, then $\mathbf{e}(k)$ can be minimized.

5.4 Simulation and Experimental Results

In this section, the simulation of the DPT scheme based on Puma 560 is carried out. Then, the DPT scheme is implemented on the 6-DOF industrial manipulator, FANUC M20-iA by using the C-track 780 as dynamic pose measuring instrument.

5.4.1 Simulation Results

Puma 560 is the first industrial robot, whose parameters are well known and very similar to those of most modern 6-axis industrial robots [148]. The developed DPT scheme has been implemented on Puma 560 in Matlab and with the help of the Robotic Toolbox provided by Peter Corke [148]. Small deviations, e.g., 5% error, are imposed on Jacobian matrix $\boldsymbol{J}(\mathbf{q}(k))$. The simulation results on the line tracking without path correction and with the proposed DPT scheme are presented in Table.5.1. It is noticed that after applying DPT scheme, the tracking errors resulted from the inaccurate $\boldsymbol{J}(\mathbf{q}(k))$ can be reduced to $\pm 0.05mm$ for position and $\pm 0.05deg$ for orientation. Therefore, the simulation results demonstrate that the DPT scheme is robust to some deviations on $\boldsymbol{J}(\mathbf{q}(k))$.

Table 5.1: Simulation Results of DPT scheme on Puma 560 with some deviations on $\boldsymbol{J}(\mathbf{q}(k))$.

Steady state	Without path correction	With path correction
x error $[mm]$	8.3	± 0.05
y error $[mm]$	33.9	± 0.0007
z error $[mm]$	60.0	± 0.0138
γ error $[deg]$	5.93	± 0.05
β error $[deg]$	2.76	± 0.05
α error $[deg]$	7.04	± 0.05

5.4.2 Experimental Preparation

The experimental setup for implementing the DPT scheme as shown in Fig.5.4 includes FANUC M20-iA with the end-effector attached some reflectors, C-Track 780 on the tripod, and an aluminum shelf attached dozens of distributed reflectors. FANUC M20-iA in Fig.5.4 is a 6-axis hollow wrist robot which has six revolute joints. FANUC M20-iA comes with the FANUC R30iB controller, which provides dynamic path modification function through Ethernet connection. FANUC company provide software package, PCDK toolkit to support the connected computer to communicate with the robot controller by Ethernet. C-Track 780 is a photogrammetry sensor with dual cameras. In this experiment, it provides fast visual measurements periodically by simultaneously sampling the reflectors in its FOV. The repeatability and volumetric accuracy of the C-Track 780 is $0.0025mm$ and $0.065mm$ respectively. The distributed reflectors on the aluminum shelf in the workspace are

used to construct as a reference model attached with a dynamic reference frame.

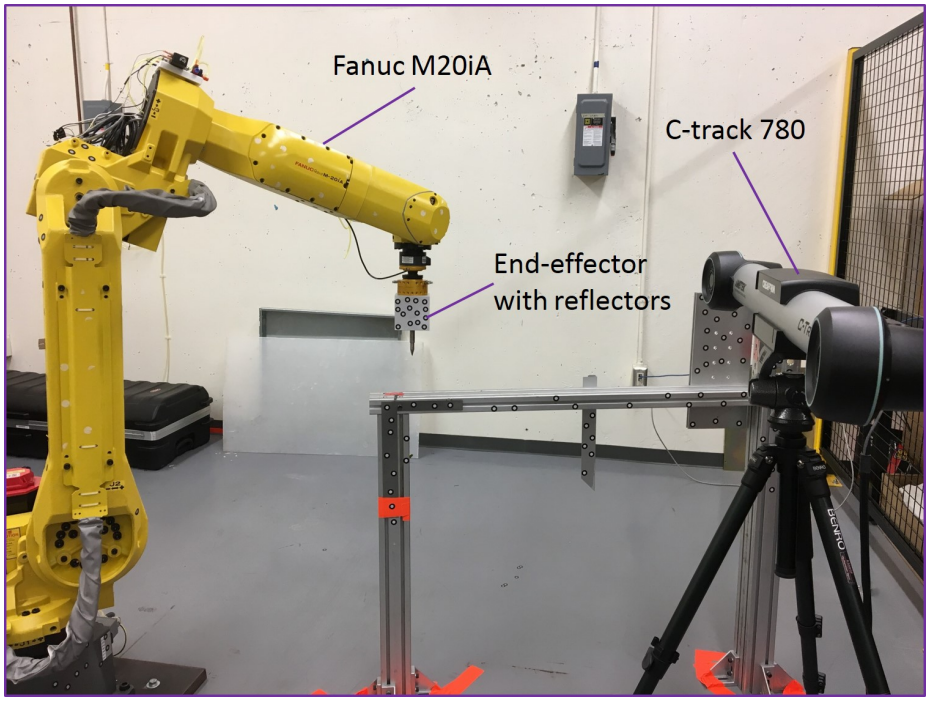


Figure 5.4: Experimental setup.

The end-effector of FANUC M20-iA in Fig.5.4 is artificially customized. The cube of the end-effector is symmetric and designed for holding the reflectors. The normal axis of the needle on the end-effector is along the z axis of the original tool frame defined on the flange center. The needle tip works as tool center point (TCP) and the new tool frame $\mathcal{F}_E$ is originated at the needle tip so as to be observed conveniently. As for the movement control of the end-effector, it is equivalent to translate the TCP and rotate $\mathcal{F}_E$ around the TCP relative to the robot user frame $\mathcal{F}_(U_R)$. FANUC R30iB provides a built-in 6-points method to define the new tool frame $\mathcal{F}_E$ on the needle tip. Thus, $\mathcal{F}_E$ at the TCP can be activated as the current tool frame and recognized by the robot controller. Then, one 3-points method provided by FANUC R30iB is used to define the user frame $\mathcal{F}_{U_R}$ based on the position that $\mathcal{F}_E$ is represented in base frame $\mathcal{F}_B$.

On the other hand, $\mathcal{F}_E$ needs to be defined in VXelements so that the TCP pose can be estimated from the measurements of C-Track 780. The non-collinear reflectors sticked on the cube of the end-effector can grouped as the end-effector model $Model_E$ in VXelements. At the same time, $\mathcal{F}_E$ originated at the TCP is attached to $Model_E$. Thus, the pose of the end-effector can be represented

as the pose of $\mathcal{F}_E$ with respect to $\mathcal{F}_{U_T}$ in VXelements so that the definition of $\mathcal{F}_E$ is identical to the one defined in the robot controller. Also, the equivalent user frame $\mathcal{F}_{U_{Eq}}$ is defined in VXelements so that current pose correction in $\mathcal{F}_{U_{Eq}}$ is a compatible adjustment for the path tracking of FANUC M20-iA in $\mathcal{F}_{U_R}$. Moreover, the task path $\mathcal{P}$ can be pre-planned through off-line tools according to the task description.

The DPT scheme is implemented by the software of DPT module. FANUC PCDK and VXelements are software packages provided by FANUC and Creaform as dynamic link library (DLL). Based on these DLL, DPT module can communicate with FANUC R30iB controller and the controller of C-Track 780 respectively. The control interval T_c should be bigger than $34.48ms(1/29Hz)$ as a result that the maximum updating frequency of C-Track 780 is $29Hz$. The interpolation interval (ITP) of FANUC M20-iA is $8ms$. The destination can be computed and sent to the motion environment during the cycle. In the experiments, DPT module is running on the high-performance computer with Intel Xeon Processor E5-1650 v3 3.5GHz and NVIDIA Quadro K2200 professional graphics board. Simultaneously, one teach pendant (TP) program in robot controller is running for actuating the robot to move along the tool path P.

5.4.3 Error analysis for FANUC M20-iA

Before implementing DPT, the error analysis for FANUC M20-iA moving along straight line without using DPT for path correction has been conducted. Three tests are carried out to compare the distance error between the current position and the desired line $\mathcal{P}_{d1}$ when the TCP is moving along $\mathcal{P}_{d1}$, described in Table.5.2. The forward speed and the backward speed are $25mm/s$ and $15mm/s$ respectively.

Table 5.2: Pose information of $\mathcal{P}_{d1}$ for the end-effector for experiments.

	Start point	End point
$x\ [mm]$	238.04	-454.41
$y\ [mm]$	16.96	166.1
$z\ [mm]$	2009.69	2597.18
$\gamma\ [deg]$	-90.11	-90.21
$\beta\ [deg]$	12.98	12.67
$\alpha\ [deg]$	0.93	0.98

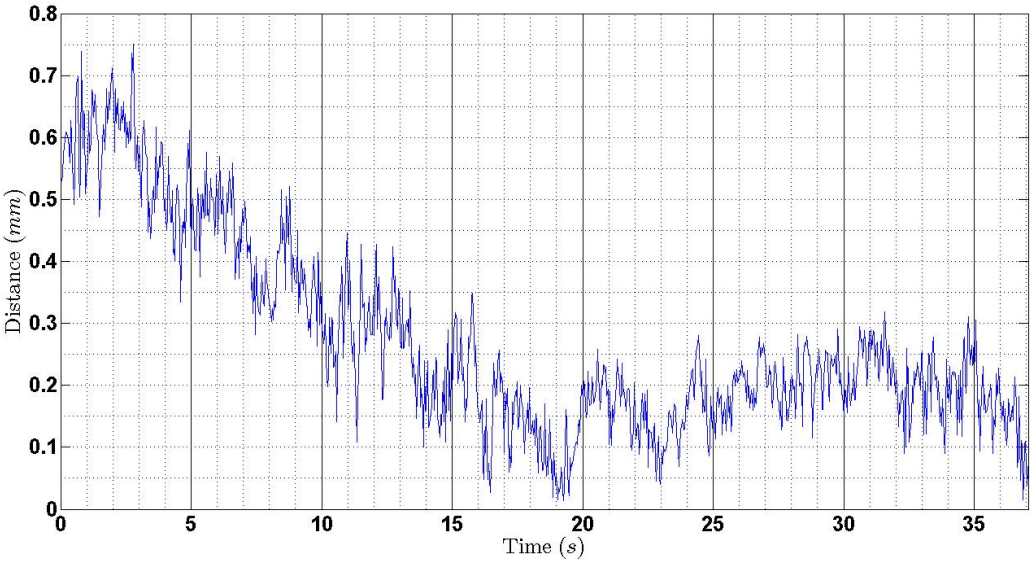


Figure 5.5: Distance error for FANUC M20-iA moving forward along $\mathcal{P}_{d1}$ at speed $25mm/s$ without path correction.

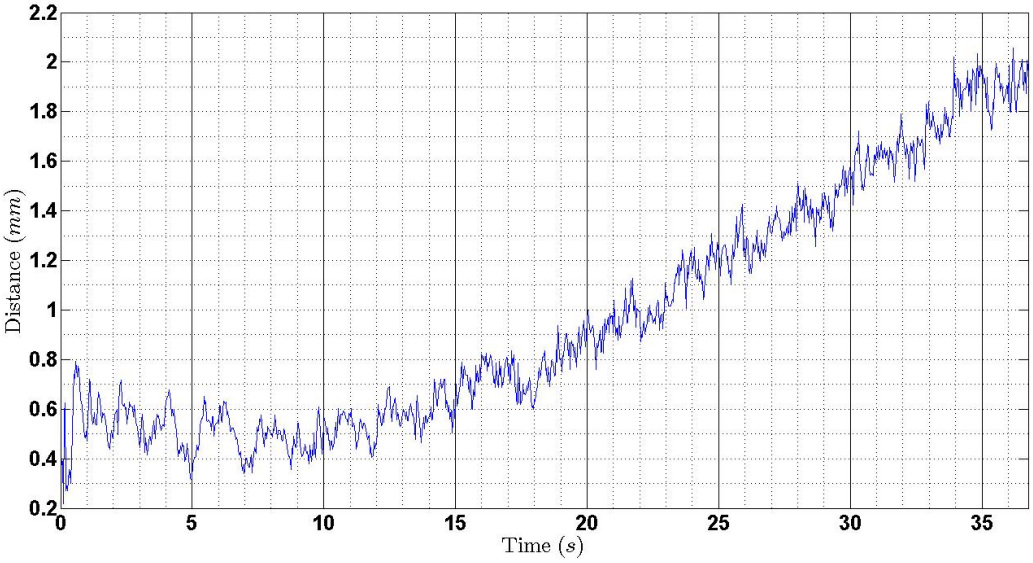


Figure 5.6: Distance error for FANUC M20-iA moving backward along $\mathcal{P}_{d1}$ at speed $25mm/s$ without path correction.

For saving space, only the distance errors with moving speed as $25mm/s$ are shown Figs.5.5 and 5.6. By analyzing the results, the following phenomenons are observed:

- Different error patterns as Tracking along the same path with different direction.

- Tracking different paths exhibits different error patterns.
- The distance error increases with higher tracking speed.
- Low frequency oscillations exist as the robot is following a straight path.
- Larger oscillation occurs with faster tracking movement for the robot.

Besides, high-frequency noise caused by the noisy measurements from C-Track appears on the distance error, which means that the measurements from C-Track cannot be directly used as the feedback in dynamic path tracking control. The distance error is caused by three major factors: robot nonlinearity, low-frequency vibration of robot movement and high-frequency noise of C-Track measurement. The AKF aims at removing the high frequency noise and the PBVS aims at reducing the distance error. Therefore, in order to enhance the accuracy of path tracking of FANUC M20-iA, the proposed control scheme with a proper noise filtering should address all above-mentioned factors.

5.4.4 Kalman Filter Initialization

According to Eq.(3.10)-Eq.(3.18) proposed in Chapter 3, in order to start the Kalman filter, the following variables and parameters should be initialized properly.

The initial estimation $\hat{\boldsymbol{\rho}}_{0,0}$ is initialized by using the current pose estimation and setting velocity zero. The initial covariance matrix $\mathbf{W}_{0,0}$ is set as a 12×12 identity matrix, which affects the transient convergent speed of the Kalman filter.

The diagonal elements of $\mathbf{A}$ are 1 and $\mathbf{A}_{i,i+6}(i = 1 \cdots 6)$ is taken as $T_c/10^3$. Ideally, the current pose of the end-effector is equal to the former six elements of the current state vector $\hat{\boldsymbol{\rho}}_{k,k}$. $\boldsymbol{\Omega}_0$ is $0.001diag[0.15, 0.15, 0.8, 0.158, 0.143, 0.0219]$, which are obtained by calculating the root mean square error of static measurements for fixed point in $\mathcal{W}_R$. The constant weights $\mu_i,\ i = 1 \cdots 6$ for updating $\boldsymbol{\Omega}_k$ with velocity changes are set as $1.5e-6$. Compared to $\boldsymbol{\Omega}_0$, $\mathbf{Q}_k$ is initialized as $(1e-6)diag[1, 1, 10, 1, 1, 1, 1, 1, 10, 1, 1, 2]$ and the length N in Eq.(3.18) is 20.

5.4.5 Control Parameters Initialization

The sampling time T_s for C-Track is set to be $1/29$ second, while $29Hz$ is the maximum frequency of C-Track 780. Once the image information from C-Track 780 is updated, the pose estimation and AKF will work sequentially to obtain the pose information with minimum delay. Therefore, the pose updating rate is equal to $29Hz$. On the other hand, the control period T_c for DPT module is determined by considering T_s, ITP of FANUC M20-iA and maximum control step. For the experiments in this research, T_c is set as $300ms$. $\boldsymbol{\Gamma}$ is set as $[diag[[0.3, 0.3, 0.3, 0.15, 0.15, 0.15], \mathbf{0}_{6\times6}]$, and $\boldsymbol{\Lambda}$ is set as $[diag[0.04, 0.04, 0.04, 0.02, 0.02, 0.0], \mathbf{0}_{6\times6}]$.

5.4.6 Experiments for Implementing DPT scheme

Line Tracking

Two experiments are carried out to obtain the results by using DPT and AKF when FANUC robot is moving forward and backward along $\mathcal{P}_{d1}$ at speed $25mm/s$ respectively. The pose updating rate of AKF is equal to the sampling rate of C-Track. Figs.5.5–5.6 show the results when FANUC M20-iA is moving forward and backward along $\mathcal{P}_{d1}$ at speed $25mm/s$ without path correction, while Figs.5.7–5.9 and Figs.5.10–5.12 show the results by using DPT for path correction. For tracking $\mathcal{P}_{d1}$, the TCP of FANUC M20-iA starts at the starting point and stops at the end point. Correspondingly, the TCP accelerates at the beginning segment of the path and decelerates at the final segment of the path. Due to the influence of the acceleration, AKF can converge and follow the measured data after around $3sec$. Then after around $6sec$, the distance error will keep less than $\pm0.2mm$. Table.5.3 and Table.5.4 show the comparison results after $9sec$. According to Table.5.3 and Table.5.4, the line tracking accuracy is improved obviously by using DPT. Besides, another two experiments are conducted to analyze the results for FANUC M20-iA moving forward and backward along $\mathcal{P}_{d1}$ at speed $15mm/s$. Compared to the tracking results at speed $25mm/s$, the results at speed $15mm/s$ show that similar accuracy improvement can be achieved. For brevity, only the distance error comparison is provided in Table.5.5. Therefore, the results demonstrate that DPT scheme can improve the line tracking accuracy up to $\pm0.1mm$ for position and $\pm0.05deg$ for rotation respectively.

Circle Tracking

Two groups of experiments are carried out to compare the tracking accuracy when the TCP of the FANUC M20-iA is moving along two half circles, $\mathcal{P}_{d2}$ and $\mathcal{P}_{d3}$, which have the common circle center in Table.5.6 and different radii, $300mm$ and $400mm$ respectively, at speed $25mm/s$. For brevity, the results about $\mathcal{P}_{d3}$ are not provided by figures. Fig.5.13 shows the distance errors when FANUC robot is moving along $\mathcal{P}_{d2}$ at speed $25mm/s$ without path correction, while Figs.5.14–5.15 show the results by using DPT for path correction. The comparison of the results is summarized in Table.5.7 where the distance error is less than $\pm 0.2mm$ with path correction. During the circle tracking movement, the orientation of the end-effector is controlled with respect to the circle center and the rotational accuracy is up to $\pm 0.1deg$. From the experimental results of both line and circle tracking, one can see that the developed DPT scheme can guide the TCP to follow the pre-planned path with high accuracy up to $\pm 0.2mm$ for position and $\pm 0.1deg$ for orientation. Compared with the current robot controller without path correction, the path tracking accuracy has been significantly improved in terms of distance error and pose error between the TCP pose and desired one.

5.5 Conclusion

In this chapter, a practical DPT scheme is proposed to command the robot to follow the desired path through the robot controller. The DPT scheme adopts PBVS strategy and eye-to-hand configuration where a photogrammetry sensor, C-track, can measure the pose of the end-effector in real time. Moreover, in order to reduce the influence from the image noise, blur and distortion, an adaptive Kalman filter is proposed to process the pose estimation of the end-effector. The stability of the DPT scheme is proved by using Lyapunov stability theory. The experimental tests demonstrate that the proposed DPT scheme can improve the line tracking accuracy to $\pm 0.1mm$ for position and $\pm 0.05deg$ for orientation, and the circle tracking accuracy to $\pm 0.2mm$ for position and $\pm 0.1deg$ for orientation. The developed DPT scheme renders the industrial robots high-end ones without retrofitting the robot with expensive encoders. The future work includes applying the DPT scheme to the other types of path tracking such as S-shape curve, square etc. and testing on the other types of industrial robots such as ABB and KUKA.

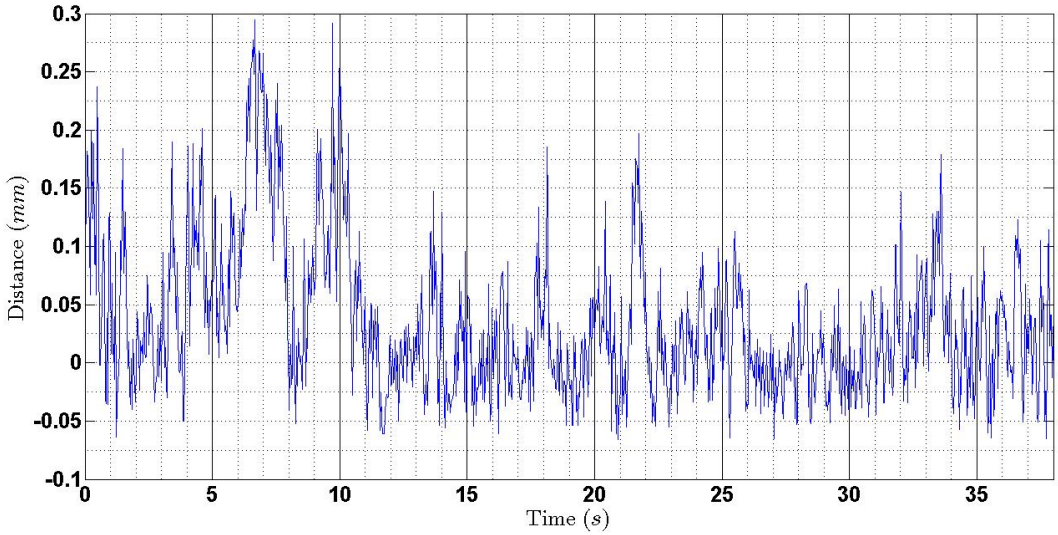


Figure 5.7: The distance error for FANUC M20-iA moving forward along $\mathcal{P}_{d1}$ at speed $25mm/s$ with path correction.

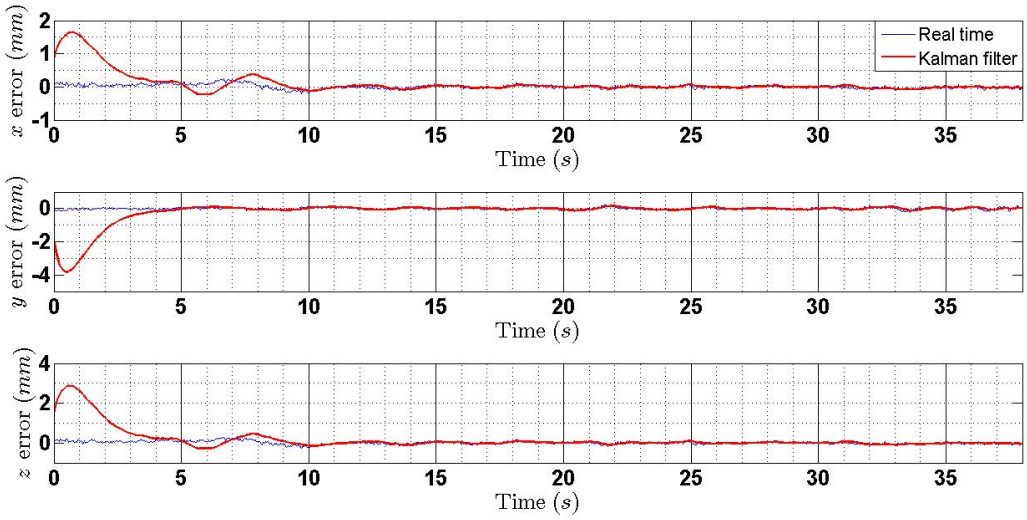


Figure 5.8: The position error of the TCP for FANUC M20-iA moving forward along $\mathcal{P}_{d1}$ at speed $25mm/s$ with path correction.

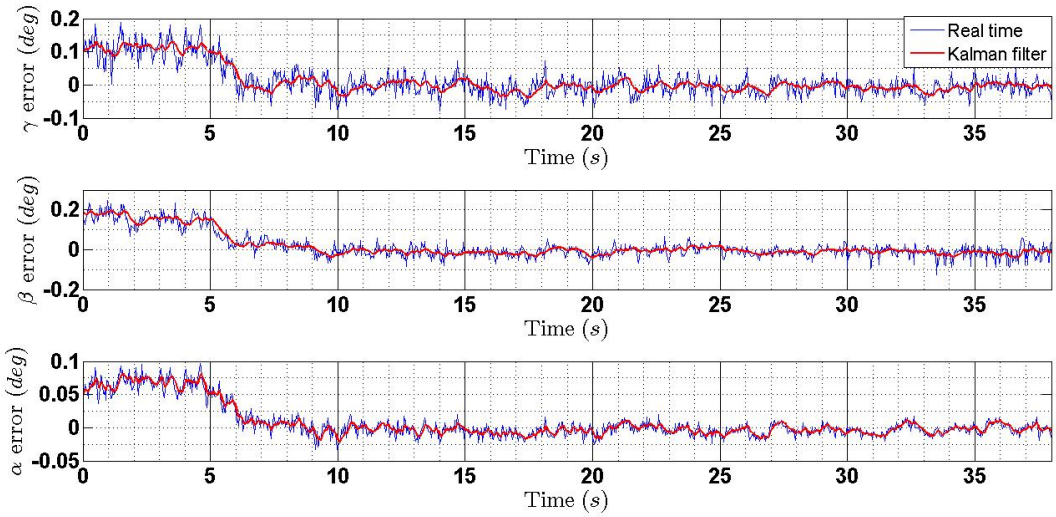


Figure 5.9: The orientation error of the TCP for FANUC M20-iA moving forward along $\mathcal{P}_{d1}$ at speed $25mm/s$ with path correction.

Table 5.3: Results comparison for FANUC M20-iA moving along $\mathcal{P}_{d1}$ forward at speed $25mm/s$.

Steady state	Without path correction	With path correction after $9s$	Percent improvement
Distance error $[mm]$	≤ 0.45	≤ 0.25	55.5%
x error $[mm]$	± 0.25	± 0.10	60.0%
y error $[mm]$	± 0.25	± 0.10	60.0%
z error $[mm]$	± 0.25	± 0.10	60.0%
γ error $[deg]$	± 0.20	± 0.05	75.0%
β error $[deg]$	± 0.15	± 0.05	66.7%
α error $[deg]$	± 0.08	± 0.05	37.5%

Table 5.4: Results comparison for FANUC M20-iA moving along $\mathcal{P}_{d1}$ backward at speed $25mm/s$.

Steady state	Without path correction	With path correction after $9s$	Percent improvement
Distance error $[mm]$	≤ 2.20	≤ 0.20	90.9%
x error $[mm]$	± 1.30	± 0.10	92.3%
y error $[mm]$	± 0.60	± 0.10	83.3%
z error $[mm]$	± 1.60	± 0.10	93.8%
γ error $[deg]$	± 0.15	± 0.05	66.7%
β error $[deg]$	± 0.30	± 0.05	83.3%
α error $[deg]$	± 0.05	± 0.05	0.0%

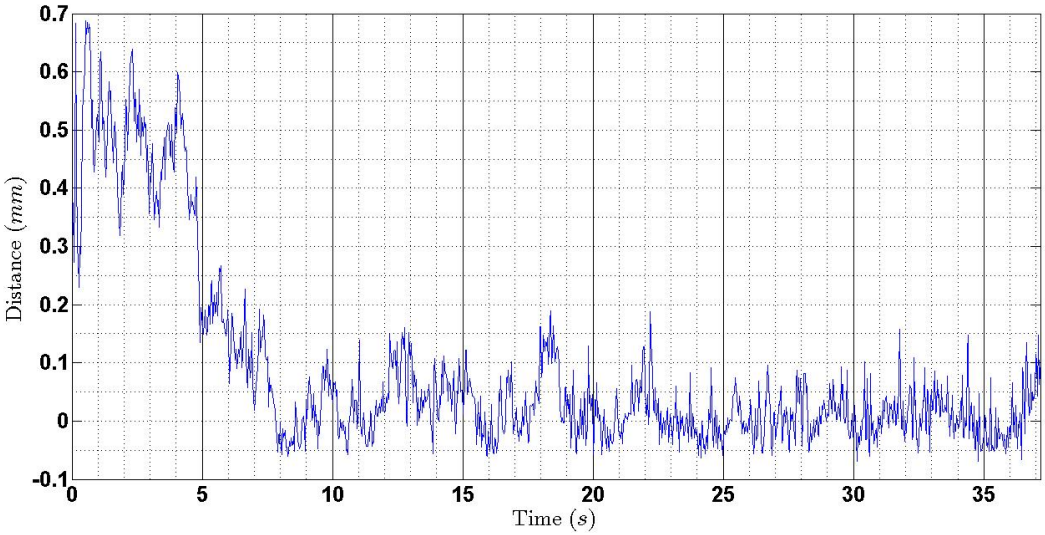


Figure 5.10: The distance error for FANUC M20-iA moving along $\mathcal{P}_{d1}$ backward at speed $25mm/s$ with path correction.

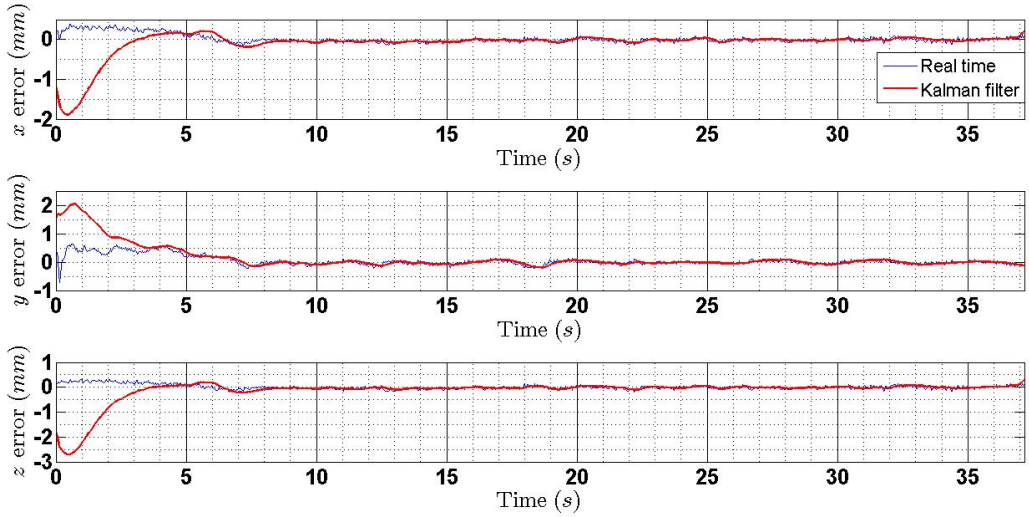


Figure 5.11: The position error of the TCP for FANUC M20-iA moving along $\mathcal{P}_{d1}$ backward at speed $25mm/s$ with path correction.

Table 5.5: Results comparison for FANUC M20-iA moving forward and backward along $\mathcal{P}_{d1}$ at speed $15mm/s$.

Distance error	Without path correction	With path correction after $9s$	Percent improvement
Forward $[mm]$	≤ 0.50	≤ 0.20	60.0%
Backward $[mm]$	≤ 2.10	≤ 0.20	90.5%

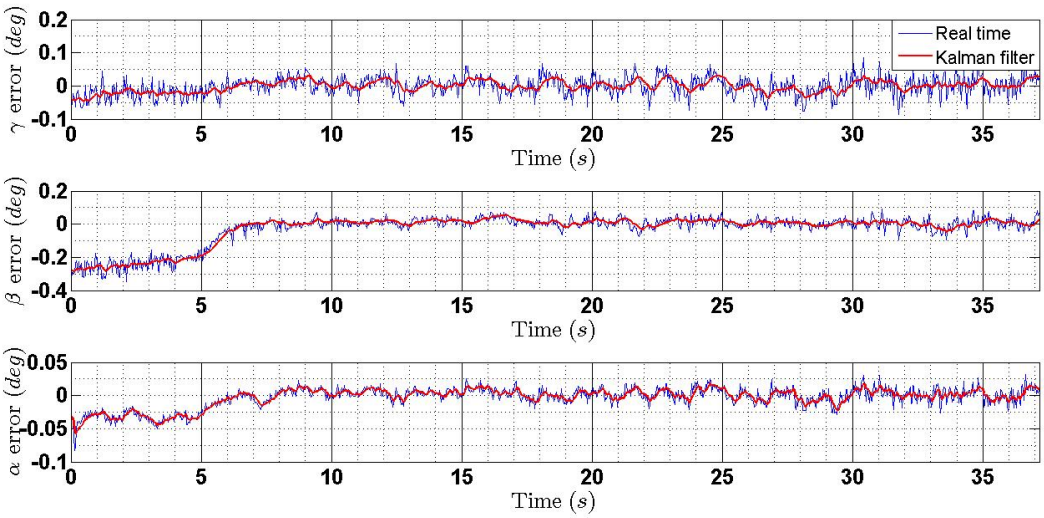


Figure 5.12: The orientation error of the TCP for FANUC M20-iA moving along $\mathcal{P}_{d1}$ backward at speed $25mm/s$ with path correction.

Table 5.6: Circle center for experiments.

Circle center			
$x\ [mm]$	-98.01	$\gamma\ [deg]$	-5.62
$y\ [mm]$	142.34	$\beta\ [deg]$	8.10
$z\ [mm]$	2405.64	$\alpha\ [deg]$	-1.58

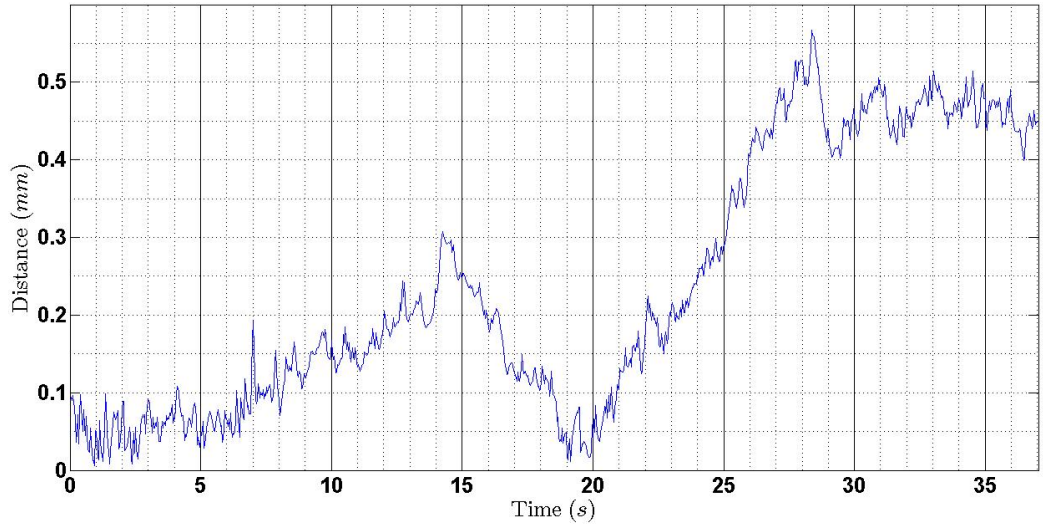


Figure 5.13: The distance error for FANUC M20-iA moving along $\mathcal{P}_{d2}$ at speed $25mm/s$ without path correction.

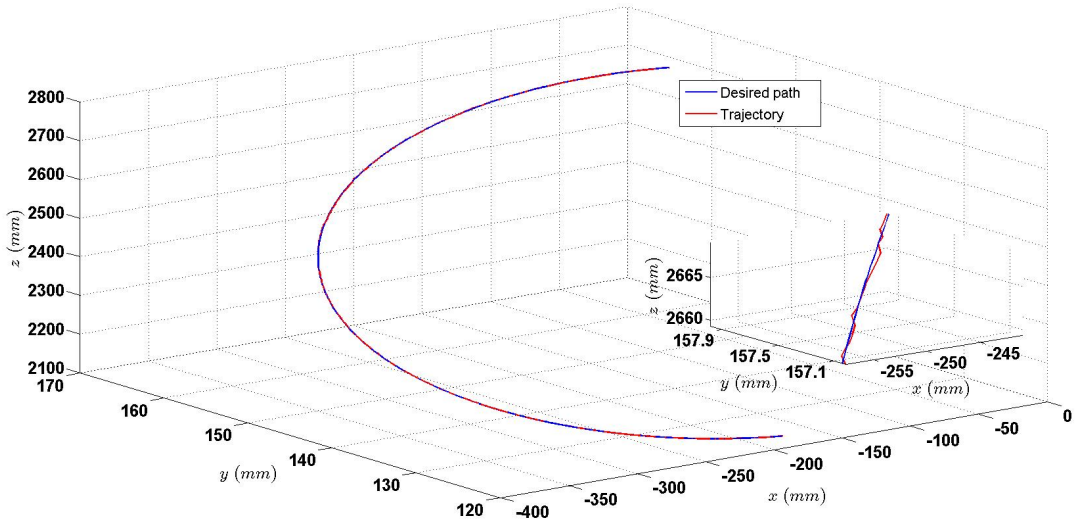


Figure 5.14: The trajectory of the TCP for FANUC M20-iA moving along $\mathcal{P}_{d2}$ at speed $25mm/s$ with path correction.

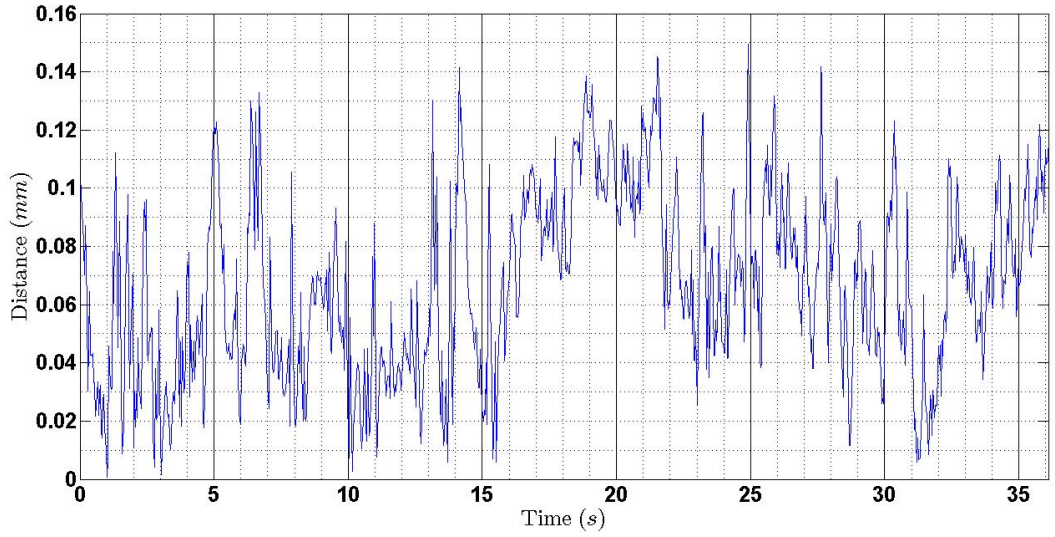


Figure 5.15: The distance error for FANUC M20-iA moving along $\mathcal{P}_{d2}$ at speed $25mm/s$ with path correction.

Table 5.7: Results comparison for FANUC M20-iA moving along $\mathcal{P}_{d2}$ and $\mathcal{P}_{d3}$ at speed $25mm/s$.

Distance error	Without path correction	With path correction	Percent improvement
$\mathcal{P}_{d2}$ $[mm]$	≤ 0.57	≤ 0.17	70.2%
$\mathcal{P}_{d3}$ $[mm]$	≤ 0.85	≤ 0.19	77.6%

Chapter 6

Adaptive Iterative Learning Control for Dynamic Path Tracking of Industrial Robots

6.1 Introduction

In aerospace manufacturing, robot manipulators are extensively employed to repeat program-planned tasks and the task paths for them tend to be repetitive. It is the scenario for the researchers to apply iterative learning control (ILC) strategy. Many early research works on ILC, such as [85, 149, 133], require some priori information of the system and rely on the 2-D contraction mapping theory. The principle of iterative learning control (ILC) is to acquire better performance through iteratively adjusting control input to the plant based on the status error and control information of previous iterations in [78]. The performance of ILC is improved as the states' errors converge to zero asymptotically along the iteration axis. Both transient response and tracking performance can be improved by ILC especially for repeating disturbances and system model uncertainty. In comparison to a feedback controller, ILC can avoid the lag in the transient phase since the previous information enables ILC to anticipate the control adjustment. ILC is similar but superior to a feedforeward control for overcoming the unknown repetitive disturbances. Actually, ILC can be regarded as a

feedback control in the iteration domain, as a result it can be robust to the system model uncertainty. However, ILC cannot excel in non-repeating noise or disturbances comparing to a feedback control. Therefore, ILC combined with a feedback control can be appropriate for both repeating and non-repeating disturbances.

Inspired by the researches through Lyapunov and Lyapunov-like methods in control fields, adaptive iterative learning control (AILC) has been proposed to estimate the uncertain parameters for robot manipulators from iteration to iteration by some researchers in recent decades , e.g., [131, 134, 135]. In theses research work, the unknown parameters can be updated in time domain during each run and the repeated disturbances can be overcome along the iteration axis. However, the same issue for the AILC in these research work as that for the traditional adaptive control method is the requirement on the constant unknown parameters. To address this issue, AILC method with time-varying parameters through iterations for nonlinear systems is proposed by Xu et al. in [80]. However, the preliminary conditions given in [80], such as Lipschitz continuous condition, is not applicable for a complex nonlinear system as 6-DOF industrial robots. Moreover, the simulation results demonstrated in [80] present oscillating convergence along the iteration axis. Currently, there are no specific experiments conducted on 6-DOF industrial robots, with the exception of Tayebi et al. is work [132], which presents the experimental results for a a 5-DOF CRS255 robotic manipulator and demonstrates a slow convergence rate and increasing high-frequency vibrations on joint movements over several dozen iterations.

The DPT control presented in Chapter 5 is essentially a feedback control. The transient performance of the DPT control is not ideal due to startup acceleration. Also, unexpected vibrations appear for the DPT control when the end-effector of the industrial robot is moving with higher speed. In this thesis research, an AILC for dynamic path tracking is proposed. An adaptive updating law for a parameter vector including time-varying parameters through iterations is designed. These adaptive parameters are not confined to be constants k_c and k_g as used in [132]. The updating time-varying parameters may implicitly contain the information from the system dynamics and repeated disturbances. The experimental results on a 5-DOF CRS255 robotic manipulator at speed $1.41mm/s$ presented in [132] show high frequency oscillations existing in the control inputs on $25th$ iteration. The oscillations are attributed to the noise accumulation and low velocity

approximation in the measurement. In this thesis research, an adaptive Kalman filter is applied to provide well-filtered real time estimations of the poses and velocities of tool center point (TCP) of the end-effector. In addition, an AILC is in parallel with DPT control to reject both repeating and non-repeating disturbances.

One common problem in the ILC is that the initial conditions of state errors $e_i(0)$ are random which cannot be set as identical in each iteration. The worst case of 5 types for initial conditions mentioned in [80] is that the initial errors are randomly variable but subject to constant limits. In this thesis research, initial conditions $e_i(0)$ are random and bounded while initial errors are generated by the first round of DPT module. DPT module has been proved to be effective for suppressing the path tracking errors under certain small values. The performance of DPT module is verified to be repeatable to a great degree. Therefore, although identical initial conditions cannot be satisfied, the initial errors can be initialized to be less than the max errors achieved by DPT module. The initial errors produced by DPT module are not apparently aligned and repeated as the same values. The AILC law can be robust to the initial shifts. The stability of the AILC law is analyzed in this thesis research.

The DPT control scheme proposed in Chapter 5 has achieved path tracking accuracy to $\pm 0.1mm$ for position and $\pm 0.05deg$ for orientation, and the circle tracking accuracy to $\pm 0.2mm$ for position and $\pm 0.1deg$ for orientation when the moving speed of the end effector is less than $25mm/s$. The objective of AILC algorithm in this thesis research is to compensate the DPT control scheme for obtaining better transient performance and path tracking accuracy from iteration to iteration. According to the analysis of the experimental data, the distance error for path tracking of the end-effector with AILC control can be less than $0.1mm$ at speed $50mm/s$ due to the improvement of the transient performance. Moreover, the path tracking accuracy for both position and orientation can be decreasing along iteration axis.

The rest of this chapter is organized as the following. In Section 6.2, the problem statement includes the dynamic state equations of 6-DOF manipulators, relevant properties and some assumptions. Section 6.3 introduces the control configuration of the AILC in parallel with DPT scheme, AILC algorithm and stability analysis. The experiments on Fanuc M20-iA by using C-track 780

to provide visual measurements are implemented and the demonstration of the experimental results prove the effectiveness of the AILC in parallel with DPT scheme in Section 6.4. Finally, the concluding remarks and future works are given in Section 6.5.

6.2 Problem Statement

The general dynamic model of 6-DOF manipulators Eq.(4.7) has the following common properties [145]:

- **Property 1** $\mathbf{M}(\mathbf{q}(t))$ is positive-definite, symmetric and bounded as $\mu_1 \leq \|\mathbf{M}(\mathbf{q}(t))\| \leq \mu_2$. μ_1 and μ_2 are existing positive constants.
- **Property 2** $\mathbf{M}(\mathbf{q}(t)) - 2\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ is skew-symmetric, and $\|\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\| \leq \lambda_a \|\dot{\mathbf{q}}(t)\|$. λ_a is a positive constant.
- **Property 3** $\|\mathbf{G}(\mathbf{q}(t))\| \leq \lambda_b$. λ_b is positive constant.
- **Property 4** The left part of Eq.(4.7) can be written in linear form to the system parameters as: $\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) + \mathbf{G}(\mathbf{q}(t)) \equiv \mathbf{\Psi}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t))\mathbf{\Theta}$, $\mathbf{\Psi}(\mathbf{q}(t), \dot{\mathbf{q}}(t).\ddot{\mathbf{q}}(t)$ is $6 \times h$ regressor matrix can be obtained when $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ and $\ddot{\mathbf{q}}(t)$ are available. $\mathbf{\Theta}$ is the vector of h system parameters.

Define $\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}_d(t)$, $\mathbf{q}_d(t)$ is the desired reference. Then, the error dynamics can be written as

$$\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{e}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{e}}(t) + \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}_d(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}_d(t)\mathbf{G}(\mathbf{q}(t)) = \boldsymbol{\tau}(t). \quad (6.1)$$

Assume that $t = kT_c$. The discrete form of Eq.(6.1) can be written as

$$\mathbf{M}\frac{\mathbf{e}(k+1) - 2\mathbf{e}(k) + \mathbf{e}(k-1)}{{T_c}^2} + \mathbf{C}\frac{\mathbf{e}(k) - \mathbf{e}(k-1)}{T_c} + \mathbf{C}\dot{\mathbf{q}}_d + \mathbf{f} = \boldsymbol{\tau}(k), \quad (6.2)$$

where for convenience, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{G}$ denote $\mathbf{M}(\mathbf{q}(k))$, $\mathbf{C}(\mathbf{q}(k), \dot{\mathbf{q}}(k))$, and $\mathbf{G}(\mathbf{q}(k))$ respectively; $\mathbf{f} = \mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{G}$ which is bounded according to **Property 1** and **3**.

Let $\tilde{\mathbf{e}}(k) = \mathbf{e}(k) - \mathbf{e}(k-1)$. Eq.(6.2) can be written as

$$\mathbf{M}(\mathbf{e}(k+1) - \mathbf{e}(k) - \tilde{\mathbf{e}}(k)) + T_c\mathbf{C}\tilde{\mathbf{e}}(k) + {T_c}^2\mathbf{C}\dot{\mathbf{q}}_d + {T_c}^2\mathbf{f} = {T_c}^2\boldsymbol{\tau}(k), \tag{6.3}$$

Then,

$$\begin{aligned}\mathbf{e}(k+1) =& \mathbf{e}(k) + \tilde{\mathbf{e}}(k) - \mathbf{M}^{-1}T_c\mathbf{C}\tilde{\mathbf{e}}(k) - \mathbf{M}^{-1}{T_c}^2\mathbf{C}\dot{\mathbf{q}}_d \\ & - \mathbf{M}^{-1}{T_c}^2\mathbf{f} + \mathbf{B}_0\boldsymbol{\tau}(k) + \Delta\mathbf{B}\boldsymbol{\tau}(k),\end{aligned} \tag{6.4}$$

where $\mathbf{B}_0 = \lambda_{min}(\mathbf{M}){T_c}^2$, with $\lambda_{min}(\mathbf{M})$ being the minimum eigenvalue of $\mathbf{M}$; $\Delta\mathbf{B} = \mathbf{M}^{-1}{T_c}^2 - \mathbf{B}_0$.

The aim of the AILC scheme is to reduce $\mathbf{e}(k)$ through iterations. Accordingly, path tracking accuracy can be further improved by integrating AILC scheme in parallel with the DPT scheme in Chapter 5. In this chapter, System parameters vector $\boldsymbol{\Theta}$ is assumed to be unknown. The description of the task path is same as defined in Section 5.2.

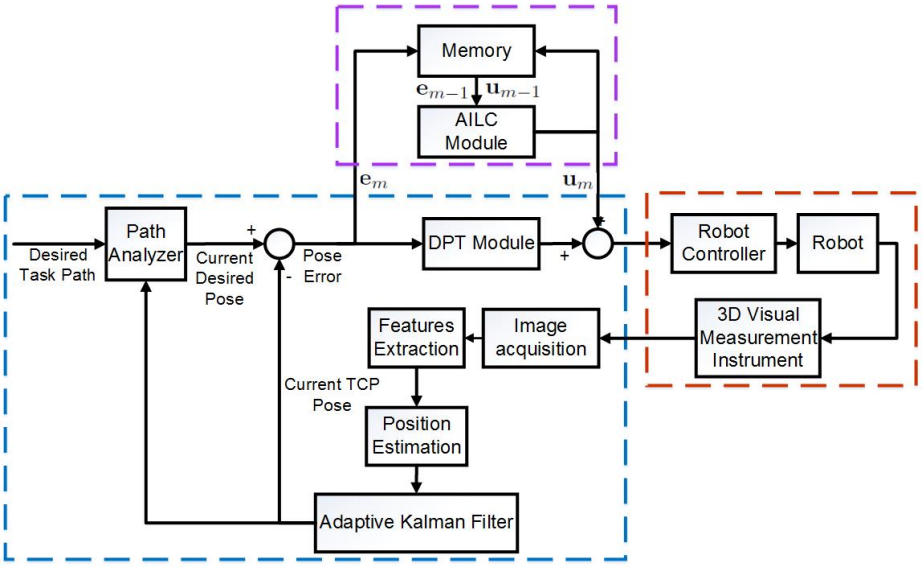


Figure 6.1: Control configuration of dynamic path tracking by using AILC scheme in parallel with DPT scheme.

6.3 Adaptive Iterative Learning Control for Dynamic Path Tracking

Fig.6.1 demonstrates the control configuration of dynamic path tracking by using AILC control to compensate DPT scheme proposed in Chapter 5. Comparing to Fig.5.2, the significant difference is the upper block including the AILC algorithm module and learning memory. Based on the saved pose error information and control input of former iteration, the AILC algorithm module produces control input of current iteration to in parallel with the control input of DPT module. The adaptive vector of the AILC algorithm in this thesis research is designed to include time-varying parameters which can be updated along iteration axis and reflect the system dynamics and repeated disturbances without knowing the system parameters. The AILC algorithm can be classified into pointwise adaptice ILC (Type 2) and model-free multi-parameter one at the same time.

6.3.1 Adaptive Iterative Learning Control algorithm

The state error $\mathbf{e}_m(k)$ at time k for iteration m is defined as $\mathbf{e}_m(k) = \mathbf{q}_m(k) - \mathbf{q}_d(k)$. Then, the control objective of the AILC algorithm is $\mathbf{e}_m(k) \to 0$ when $m \to \infty$. Similar to Eq.(5.7), $\mathbf{e}_m(k)$ can be obtained as:

$$\mathbf{e}_m(k) = \boldsymbol{J}^{-1}(\mathbf{q}_m(k))[\varsigma_m(k)^T - \varsigma_d(k)^T]^T, \tag{6.5}$$

The error dynamic equation Eq.(6.4) for joint i, $i = 1 \cdots 6$, in iteration can be written as

$$\mathbf{e}_m^i(k+1) = \alpha_1 \mathbf{e}_m^i(k) + \alpha_2 \tilde{\mathbf{e}}_m^i(k) + \alpha_3 + \mathbf{B}_0 \boldsymbol{\tau}_m^i(k), \tag{6.6}$$

where $m \in \mathbf{Z}$ denotes the iteration number and $\alpha_1,\ \alpha_2,\ \alpha_3 \in R$.

Then, according to Eq.(6.6), the ideal controller for joint i can be designed as

$$\boldsymbol{\tau}_m^i(k) = -\ \Lambda_1 \mathbf{e}_m^i(k) - \Lambda_2 \tilde{\mathbf{e}}_m^i(k) - \Phi_m \theta_m, \tag{6.7}$$

$$\theta_m = [\frac{\alpha_1}{\mathbf{B}_0} - \Lambda_1,\ \frac{\alpha_2}{\mathbf{B}_0} - \Lambda_2,\ \frac{\alpha_3}{\mathbf{B}_0}]^T, \tag{6.8}$$

$$\Phi_m = [\mathbf{e}_m^i(k),\ \tilde{\mathbf{e}}_m^i(k),\ 1], \tag{6.9}$$

where $\Lambda_1, \Lambda_2 \in R+$ are control gains. θ_m related to system parameters is assumed to be unknown. An adaptive iterative learning control (AILC) law is designed to compensate the control input of DPT module as follows

$$\boldsymbol{\tau}_m^i(k) = -\Lambda_1 \mathbf{e}_m^i(k) - \Lambda_2 \tilde{\mathbf{e}}_m^i(k) - \Phi_m \hat{\theta}_m, \tag{6.10}$$

$$\hat{\theta}_m(k) = \begin{cases} \hat{\theta}_{m-1}(k) + \Gamma \Phi_{m-1}^T \mathbf{e}_{m-1}^i(k+1), \text{ when } m > 1 \\ 0, \text{ otherwise} \end{cases}, \tag{6.11}$$

where $\hat{\theta}_m$ is the estimation of θ_m and $\Gamma \in R+$.

6.3.2 Stability Analysis

The task path $\mathcal{P}_{d2}$ is assumed to be reachable for the TCP of the end-effector and visible in the FOV of the VMI. Moreover, the Jacobian matrix $\boldsymbol{J}(\mathbf{q}(k))$ can satisfy non-singularity along $\mathcal{P}_{d2}$ since a 6-DOF rigid industrial robot is kinetically non-redundant. The stability proof of the proposed control algorithm can be based on positive definite Lyapunov-like energy function (LEF) by ensuring monotonic decreasing of LEF under some conditions which has been introduced by Xu. in [150] and Tayebi. in [131].

First, it is assumed the finite time span T for each iteration is constant. Let $\tilde{\theta}_m(k) = \theta_m(k) - \hat{\theta}_m(k)$. The Lyapunov-like function candidate for each iteration between $[0, T]$ can be defined as

$$\boldsymbol{W}_m(k) = \frac{\mathbf{B}_0 \tilde{\theta}_m^T(k) \tilde{\theta}_m(k)}{2\Gamma}. \tag{6.12}$$

Submitting Eq.(6.10) into Eq.(6.6), $\mathbf{e}_m^i(k+1)$ can be obtained as

$$\mathbf{e}_m^i(k+1) = \mathbf{B}_0 \Phi_m \tilde{\theta}_m(k). \tag{6.13}$$

Moreover, when $m > 1$, using Eq.(6.11), $\tilde{\theta}_m(k)$ can be updated as

$$\tilde{\theta}_m(k) = \tilde{\theta}_{m-1}(k) - \Gamma \Phi_{m-1}^T \mathbf{e}_{m-1}^i(k+1). \tag{6.14}$$

According to Eq.(6.12), Eq.(6.13), and Eq.(6.14), the iteration-decreasing trend of $\boldsymbol{W}_m(k)$ along iteration axis can be proved as

$$\begin{aligned}
\Delta\boldsymbol{W}_m(k) =& \boldsymbol{W}_m(k) - \boldsymbol{W}_{m-1}(k) \\
=& \frac{\mathbf{B}_0\tilde{\theta}_m^T(k)\tilde{\theta}_m(k)}{2\Gamma} - \frac{\mathbf{B}_0\tilde{\theta}_{m-1}^T(k)\tilde{\theta}_{m-1}(k)}{2\Gamma} \\
=& \frac{\mathbf{B}_0(\tilde{\theta}_{m-1}(k) - \Gamma\Phi_{m-1}^T\mathbf{e}_{m-1}^i(k+1))^T(\tilde{\theta}_{m-1}(k) - \Gamma\Phi_{m-1}^T\mathbf{e}_{m-1}^i(k+1))}{2\Gamma} \\
& - \frac{\mathbf{B}_0\tilde{\theta}_{m-1}^T(k)\tilde{\theta}_{m-1}(k)}{2\Gamma} \\
=& \frac{\mathbf{B}_0(-\Gamma\tilde{\theta}_{m-1}^T(k)\Phi_{m-1}^T\mathbf{e}_{m-1}^i(k+1) - \Gamma\Phi_{m-1}\tilde{\theta}_{m-1}(k)\mathbf{e}_{m-1}^i(k+1)}{2\Gamma} \\
& + \frac{\Gamma^2\Phi_{m-1}\Phi_{m-1}^T(\mathbf{e}_{m-1}^i(k+1))^2)}{2\Gamma} \\
=& \frac{\mathbf{B}_0(-2\Phi_{m-1}\tilde{\theta}_{m-1}(k)\mathbf{e}_{m-1}^i(k+1) + \Gamma\Phi_{m-1}\Phi_{m-1}^T(\mathbf{e}_{m-1}^i(k+1))^2)}{2} \\
=& -\mathbf{B}_0\Phi_{m-1}\tilde{\theta}_{m-1}(k)\mathbf{e}_{m-1}^i(k+1) + \frac{\mathbf{B}_0\Gamma\Phi_{m-1}\Phi_{m-1}^T(\mathbf{e}_{m-1}^i(k+1))^2}{2} \\
=& -(\mathbf{e}_{m-1}^i(k+1))^2 + \frac{\mathbf{B}_0\Gamma\Phi_{m-1}\Phi_{m-1}^T(\mathbf{e}_{m-1}^i(k+1))^2}{2} \\
=& -(1 - \frac{\mathbf{B}_0\Gamma\Phi_{m-1}\Phi_{m-1}^T}{2})(\mathbf{e}_{m-1}^i(k+1))^2,
\end{aligned} \tag{6.15}$$

which is less than 0 if only $\Gamma < \frac{2}{\mathbf{B}_0\Phi_{m-1}\Phi_{m-1}^T}$. Therefore, $\tilde{\theta}_m(k) \to 0$ when $m \to \infty$. According to Eq.(6.13), when $\tilde{\theta}_m(k) \to 0$, it is clear that $e_m^i(k+1) \to 0$.

On the other hand, when $m = 1$, the control law Eq.(6.10) becomes

$$\boldsymbol{\tau}_m^i(k) = -\Lambda_1\mathbf{e}_m^i(k) - \Lambda_2\tilde{\mathbf{e}}_m^i(k). \tag{6.16}$$

Using Eq.(6.6), the closed loop dynamics becomes

$$\begin{aligned}
e_m^i(k+1) =& (\alpha_1 - \mathbf{B}_0\Lambda_1)\mathbf{e}_m^i(k) + (\alpha_2 - \mathbf{B}_0\Lambda_2)\tilde{\mathbf{e}}_m^i(k) + \alpha_3 \\
=& (\alpha_1 - \mathbf{B}_0\Lambda_1 + \alpha_2 - \mathbf{B}_0\Lambda_2)\mathbf{e}_m^i(k) - (\alpha_2 - \mathbf{B}_0\Lambda_2)\mathbf{e}_m^i(k-1) + \alpha_3.
\end{aligned} \tag{6.17}$$

Define $x_1(k) = \mathbf{e}_m^i(k-1)$, $x_2(k) = \mathbf{e}_m^i(k)$, and $\mathbf{x}(k) = [x_1(k),\ x_2(k)]^T$, then one has

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + \begin{bmatrix} 0 \\ \alpha_3 \end{bmatrix},\ A = \begin{bmatrix} 0 & 1 \\ -(\alpha_2 - \mathbf{B}_0\Lambda_2) & \alpha_1 - \mathbf{B}_0\Lambda_1 + \alpha_2 - \mathbf{B}_0\Lambda_2. \end{bmatrix} \tag{6.18}$$

Apparently, if only Λ_1, Λ_2 are selected such that the eigenvalues of A are within the unit circle, then Eq.(6.17) is stable, and $\mathbf{x}(k)$ is bounded.

6.4 Experimental Results

In order to verify the efficiency of the proposed AILC scheme, some experiments have been carried out for controlling the TCP of the end-effector to move along a typical desired task path in the workspace of Fanuc M20-iA. The desired task path $\mathcal{P}_{d2}$ is described in Table.6.1. The experimental results are presented in three types when the TCP of the end-effector is tracking $\mathcal{P}_{d2}$ in three different situations such as without dynamic path correction, with DPT control and with AILC control. The control flowchart for implementing AILC scheme is shown in Fig.6.2. The experimental setup is the same as shown in Fig.5.4.

Table 6.1: Pose information of $\mathcal{P}_{d2}$ for the end-effector in user frame $\mathcal{F}_{U_T}$ for the experiment.

	$x(mm)$	$y(mm)$	$z(mm)$	$\gamma(deg)$	$\beta(deg)$	$\alpha(deg)$
Start point	-758.95	-485.75	731.92	175.16	-6.84	-15.07
End point	-1319.36	-317.03	864.83	178.44	12.18	-18.69

6.4.1 Parameters Initialization

In the experiments of this research, the initial errors $e_m(0)$ are assumed to be bounded. Initial errors are generated by the first round of the DPT module. The DPT module has been proved to be effective for suppressing the path tracking errors under certain small values . The performance of DPT module is verified to be repeatable to a great degree. Therefore, although identical initial conditions cannot be satisfied, the initial errors can be initialized to be less than the max errors

achieved by DPT module. The worst case of 5 types for initial conditions mentioned in [80] is that the initial errors are randomly variable but under constant small values. The initial errors produced by DPT module are not apparently aligned and repeated as the same values. However, the AILC algorithm can be robust to the initial shifts.

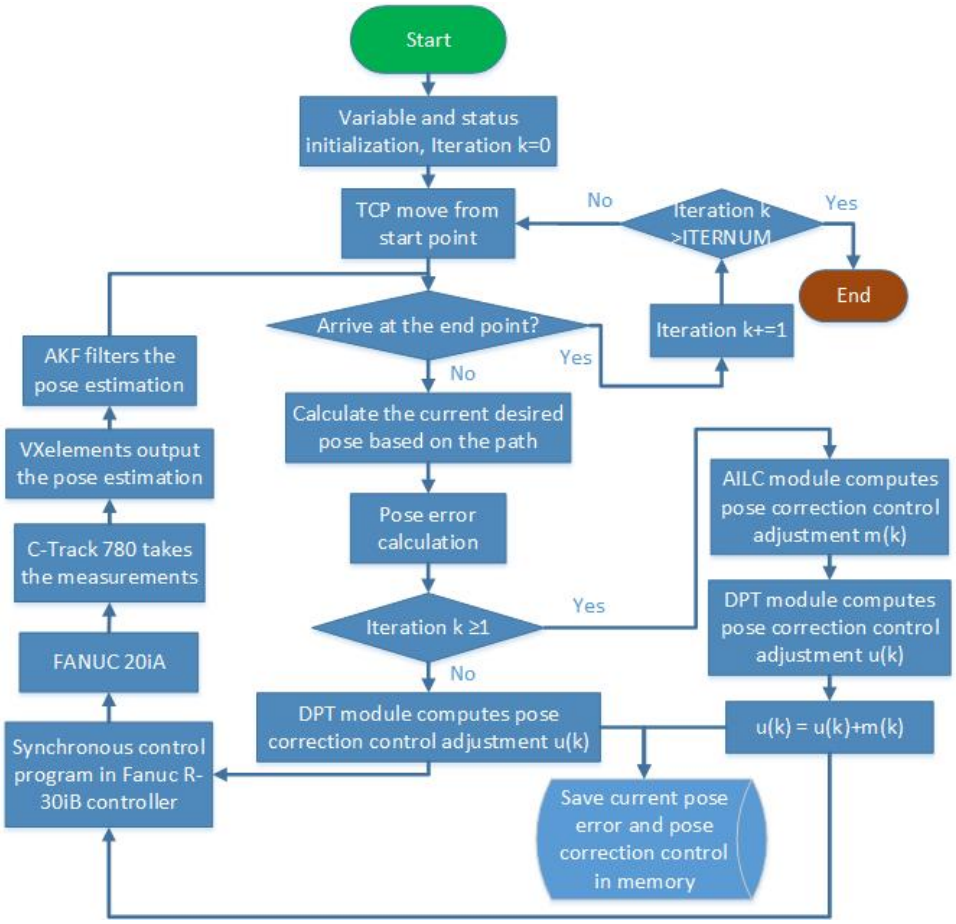


Figure 6.2: Control flowchart for implementing AILC scheme.

The initialization of Kalman filter for the experiments of AILC is the same as that in Chapter 5. Compared to the control interval T_c configured in Chapter 5, T_c for the experiments to implement AILC is set as 3 times of the sampling time T_s in order to generate rapid control signal to accommodate the higher moving speed of the end-effector. The moving speed of the end-effector applied in this research is $50mm/s$, while the moving speed of the end-effector for the experiments to verify the DPT scheme in Chapter 5 is $25mm/s$. Control gains Λ_1 and Λ_2 for 6 axis are set as $[0.1, 0.1, 0.1, 0.07, 0.0.07, 0.0.07]$ and $[0.015, 0.015, 0.015, 0.01, 0.01, 0.01]$ respectively. Adaptive

learning parameters are initialized as 0 and positive learning gain Γ is selected as 1.5. Greater values of the learning gain parameters may accelerate the convergence rate. However, the influence of the noises appearing through iterations can be magnified and cause vibrations for the movements of the joints. The worst case is the divergence of the learning process.

6.4.2 The Analysis of Experimental Results

Path Tracking without Dynamic Path Correction

The first experiment is conducted for the TCP of the end-effector moving along $\mathcal{P}_{d2}$ at speed $50mm/s$ without dynamic path correction and the error analysis of the experimental Results are demonstrated in Figs.6.3-6.5. According to Fig.6.3, the maximum distance error at speed $50mm/s$ without dynamic path correction is over $2.5mm$, which is greater than the maximum distance errors at speed $25mm/s$ shown in Fig.5.5 and 5.6. The divergence patterns of position error along x and z in user frame are similar. However, divergence range of position error along z axis is up to $2.541mm$ which is much greater than the maximum error $0.628mm$ along z axis. Moreover, there are obvious vibrations shown on the position error along y axis in Fig.6.4 although the maximum error along y axis is less than $0.2mm$. The maximum divergence range of orientation error without dynamic path correction is up to $0.254deg$ which is from the rotation angle γ around axis x of $\mathcal{F}_{U_T}$ despite of the fact that the desired orientation is unvarying along $\mathcal{P}_{d2}$. On the other hand, the experimental Results without dynamic path correction validate the observations in Chapter 5 that tracking different paths exhibits different error patterns and the distance error increases with higher tracking speed.

Path Tracking with DPT Control

Then, the second experiment for the TCP of the end-effector to track $\mathcal{P}_{d2}$ at speed $50mm/s$ with DPT control is carried out and Figs.6.6-6.8 demonstrate distance error, position error and orientation error respectively. Compared to the experimental results of the first experiment without dynamic path correction, the maximum distance error with DPT control has been significantly reduced to less than $0.3mm$. Also, position error and orientation error for each axis are improved to a great extent with dynamic path correction of DPT module and the convergence time is less than $5s$. The

vibrations on position error along y are apparently suppressed. However, the expected accuracy of robot manipulation for aerospace manufacturing is less than $0.20mm$ according to the standard process specifications in the aerospace industry [9, 10]. As the moving speed of the end-effector for path tracking in this experiment is twice of the speed preset in the experiments of Chapter 5, the desired accuracy $0.20mm$ is difficult to be achieved with the only help of DPT control.

Path Tracking with AILC Control

The objective of the third experiment is to implement AILC control to verify the efficiency of the AILC algorithm. As shown in Fig.6.1, AILC module is in parallel with DPT module. The first iteration of the third experiment is executed to initialize the parameters of AILC algorithm automatically and it is the same as the second experiment. The iterative learning process start from the second iteration. The definition of average error of each iteration is as below:

$$A = (\sum_{1}^{n} |e(k)|)/n, \tag{6.19}$$

where A denotes the average error; k and n are discrete time index and maximum time index for each iteration respectively. According to Eq.(6.19), average distance error, average position error and average orientation error can be derived for each iteration. Figs.6.9-6.11 present the gradual converging process of average distance error, average position error and average orientation error along iteration axis. The total iterations demonstrated in these figures are 100. As shown in Fig.6.9, apparent decreasing trend of average distance error can be observed from around $0.14mm$ to less than $0.08mm$. The average position error, as shown in Fig.6.6 , is gradually reduced to be less than $0.025mm$ for x error, $0.05mm$ for y error and z error along iteration axis respectively. At the same time, the average orientation error is obtained further continuous reduction while average γ error and average α error are less than $0.015deg$ and average β error is less than $0.01deg$ at iteration 100. Figs.6.12-6.14 demonstrate distance error, position error and orientation error at the last iteration, i.e., iteration 100. Compared Fig.6.6 with Fig.6.12, the distance error is stablely decreased along the entire time axis with AILC control. Moreover, the maximum distance error with AILC control is less than $0.18mm$ and the distance error after $5s$ is keeping less than $0.11mm$. Similarly, the

maximum values of position error and orientation error with AILC are all lowered and they are controlled steadily to smaller value after $5s$ comparing with the experimental results of the second experiment. Especially, the vibrations on both position error and orientation error with AILC control are efficiently reduced so that the control process is observed to be smooth.

Comparison of three experiments

Through the comparison of some features in three aforementioned experiments, the advantages of the AILC control algorithm for enhancing the accuracy of Fanuc M20-iA can be recognized. First, the maximum pose error and distance error of the end-effector during the entire path tracking process,i.e., when Fanuc M20-iA move from the start point to end point of $\mathcal{P}_{d2}$ at speed $50mm/s$, are identified and presented in Table.6.2. It can be observed that the maximum errors of each item in each column of Table.6.2 are decreased greatly. Moreover, as shown in Figs.6.3-6.8 and Figs.6.12-6.14, the maximum errors definitely occur in the initial $5s$. Therefore, the comparison of the maximum errors verify that the transient performance can be significantly improved using AILC module to compensate DPT control scheme. The RMS errors of pose error and distance error from three experiments are listed in Table.6.3. Similar to the maximum errors in Table.6.2, the RMS errors for both position errors and orientation errors can be significantly reduced. Table.6.4 gives the percentage accuracy improvement of each item according the results demonstrated in Table.6.3. Most of the position errors and orientation errors can be decreased over 50% except that the accuracy of α, orientation around z axis of use frame, can be improved as much as 40.3%, while the overall distance error can be reduced up to 53.0%.

6.5 Conclusion

In this chapter, an AILC scheme in parallel with the DPT scheme proposed in Chapter 5 is presented. The AILC algorithm aims to improve both transient and steady performances of pose tracking by updating the time-varying parameters adaptively along iteration axis. The control signal generated from the AILC algorithm adjust the control inputs produced by the DPT module at each time interval based on the memorized data information and current feedback. Three experiments in

different situations (without path correction, with DPT control, and with AILC control) are carried out. Through the demonstration of data figures and tables, it is evident to recognize the enhancement of developed AILC strategy compared to the DPT control module itself. First, the transient control performance is improved with the AILC module. After certain limited iterations, high path tracking accuracy can be achieved even during the beginning $5s$ and the maximum distance error is less than $0.2mm$ when the TCP of the end-effector is following the desired path at speed $50mm/s$, which is twice of the speed applied in Chapter 5. Second, the pose accuracy can be stably confined to less than $0.1mm$ for position and $0.05deg$ for orientation through iterative learning. Moreover, the repetitive disturbances can be also overcome within certain iterations so that the vibrations can be significantly reduced. Therefore, the AILC algorithm proposed in this chapter is verified to be very effective to further improve the DPT scheme. The future work includes applying the combined control scheme to the other types of industrial robots such as ABB and KUKA and practical industrial applications.

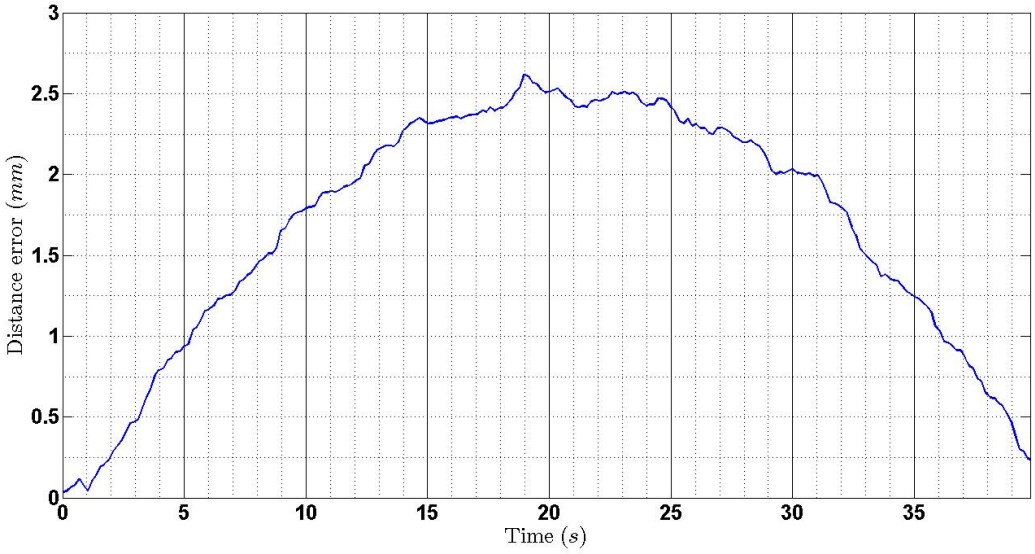


Figure 6.3: Distance error for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ without path correction.

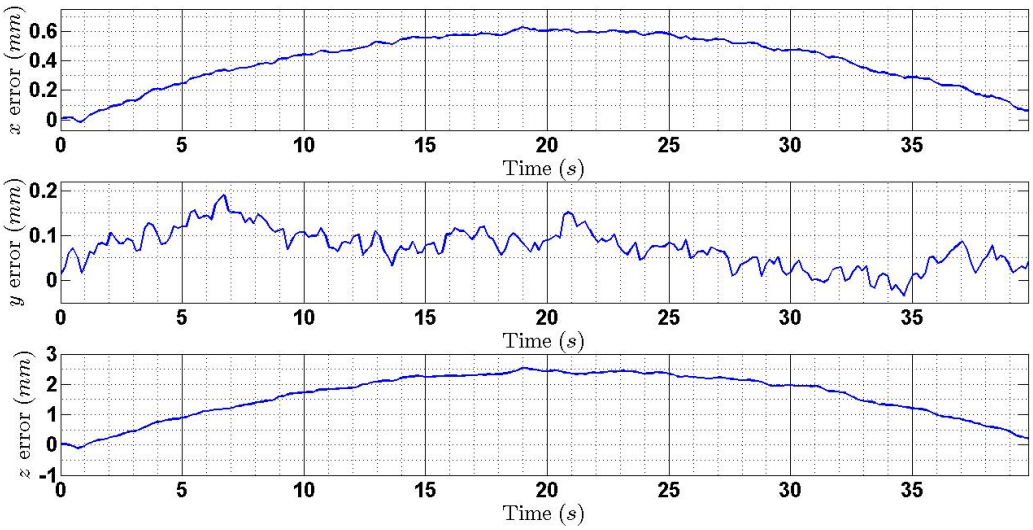


Figure 6.4: Position error for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ without path correction.

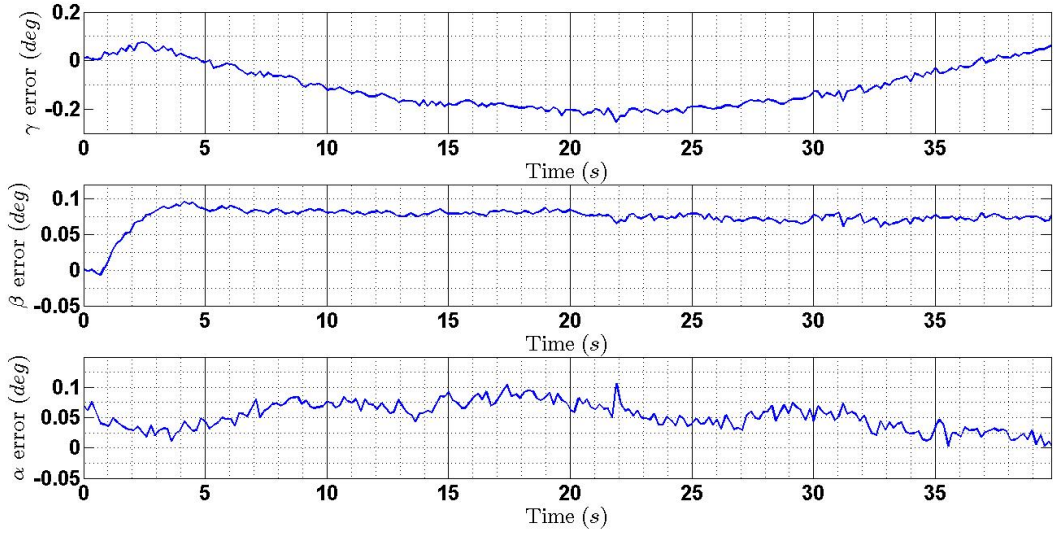


Figure 6.5: Orientation error for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ without path correction.

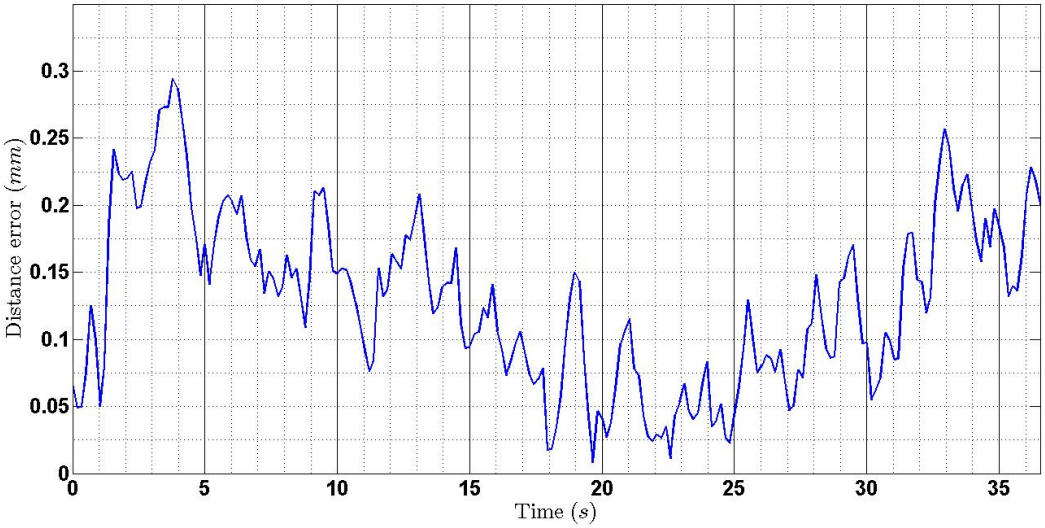


Figure 6.6: Distance error for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with DPT control.

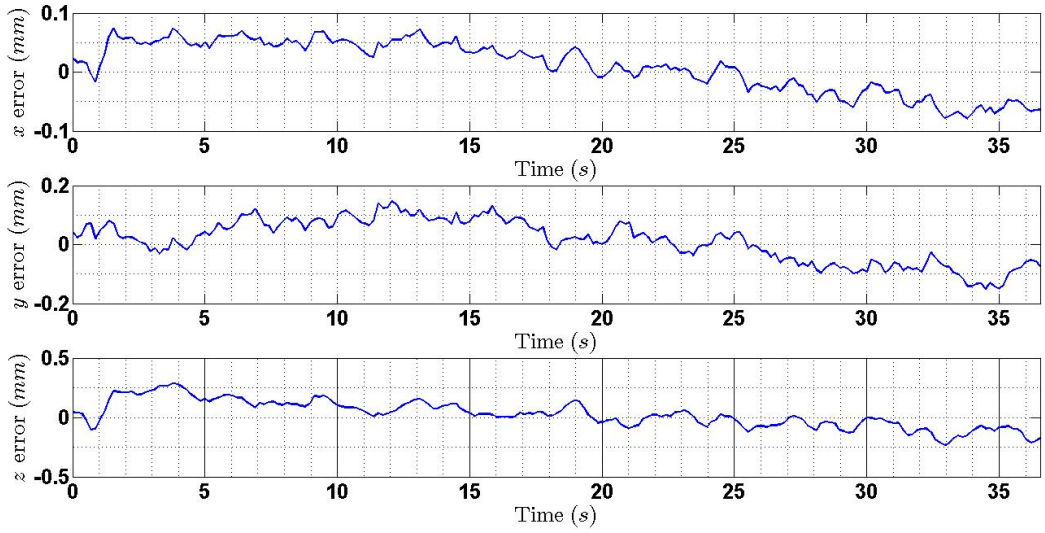


Figure 6.7: Position error for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with DPT control.

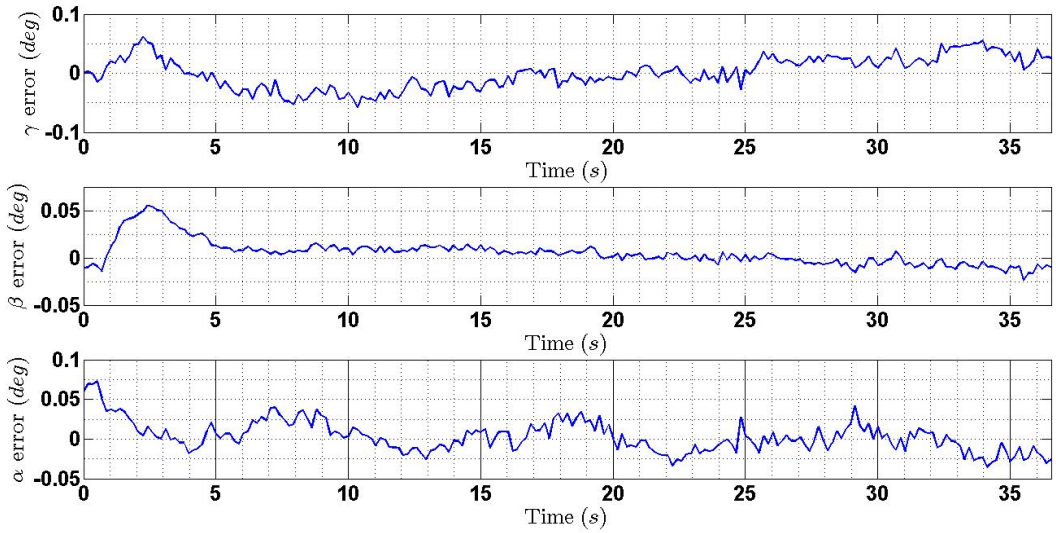


Figure 6.8: Orientation error for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with DPT control.

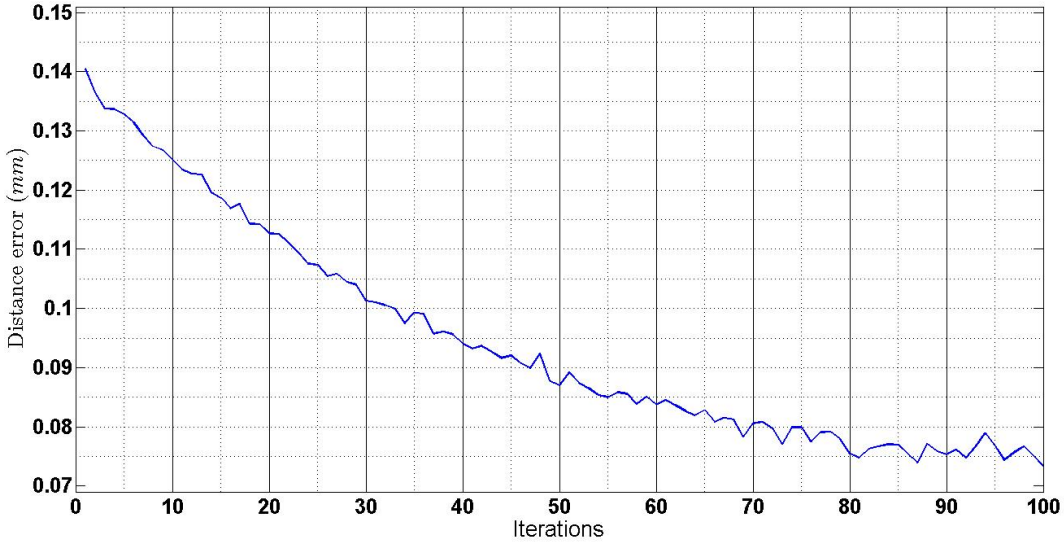


Figure 6.9: Average distance error of each iteration for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with AILC control.

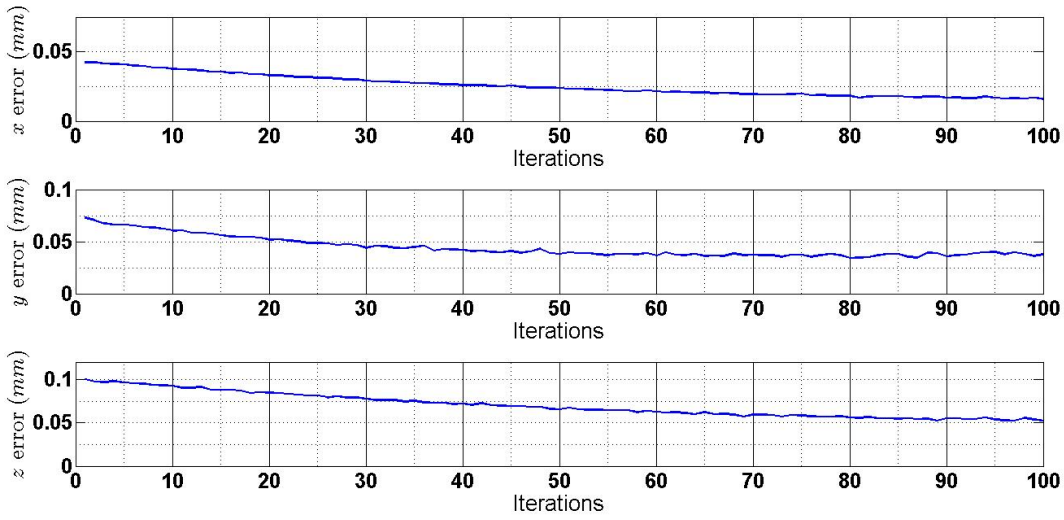


Figure 6.10: Average position error of each iteration for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with AILC control.

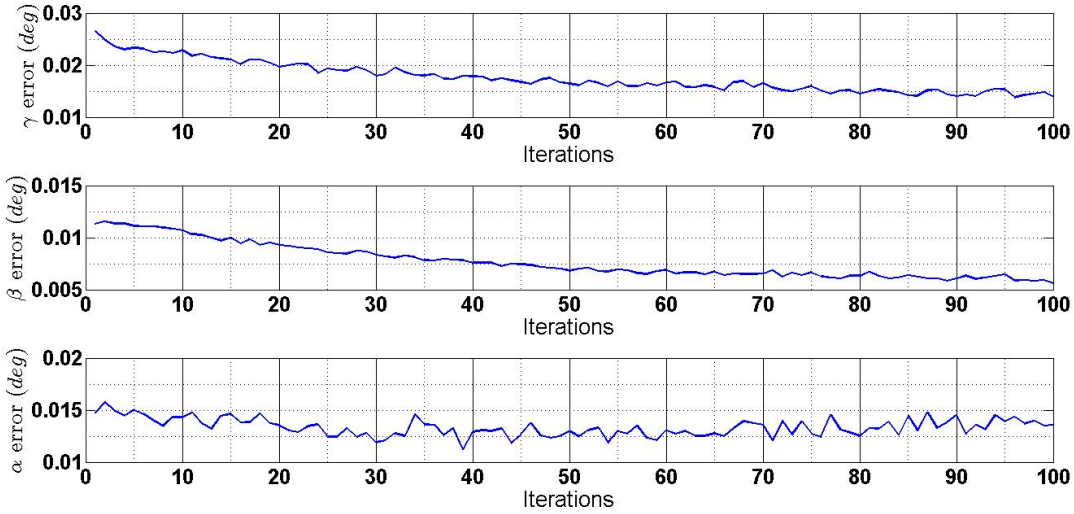


Figure 6.11: Average orientation error of each iteration for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with AILC control.

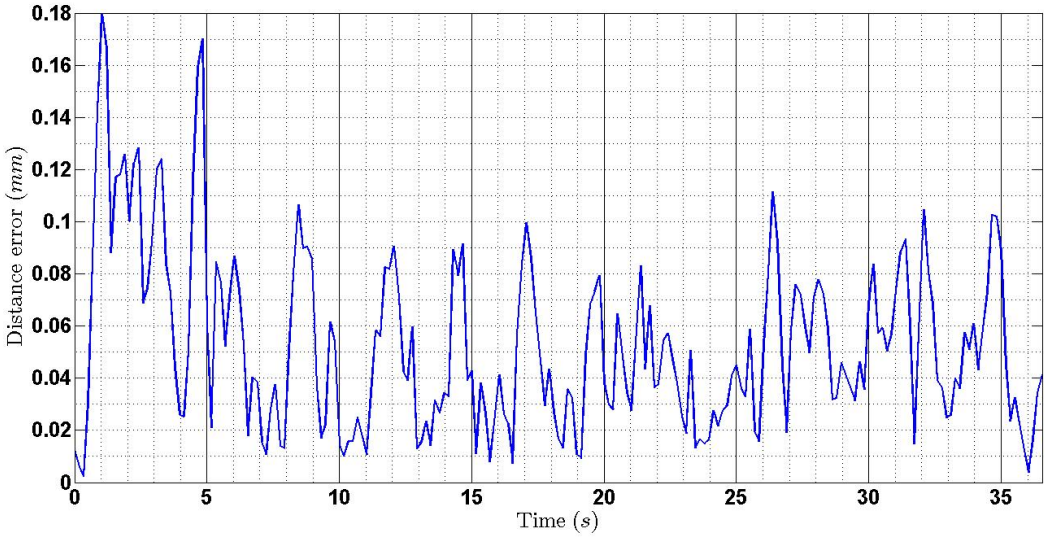


Figure 6.12: Distance error of last iteration for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with AILC control.

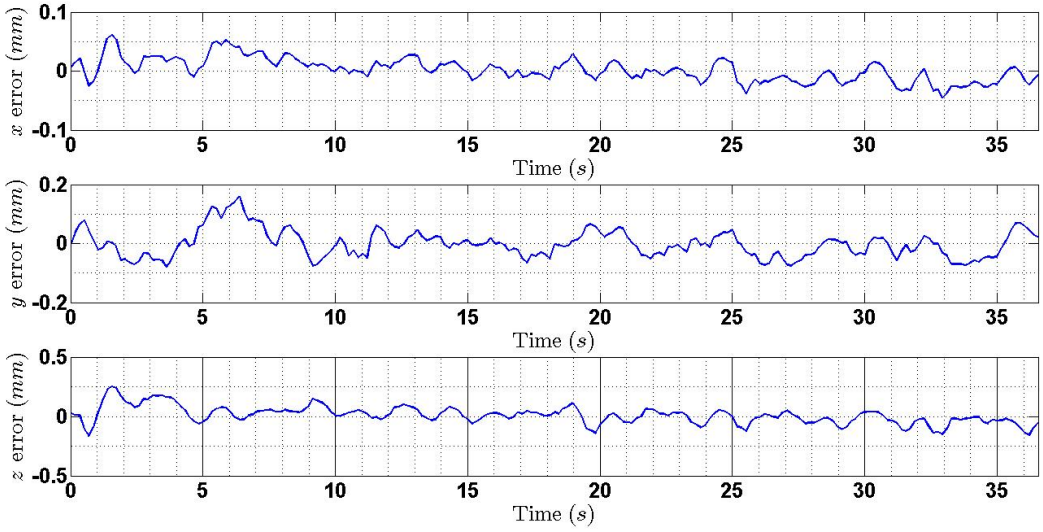


Figure 6.13: Position error of last iteration for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with AILC control.

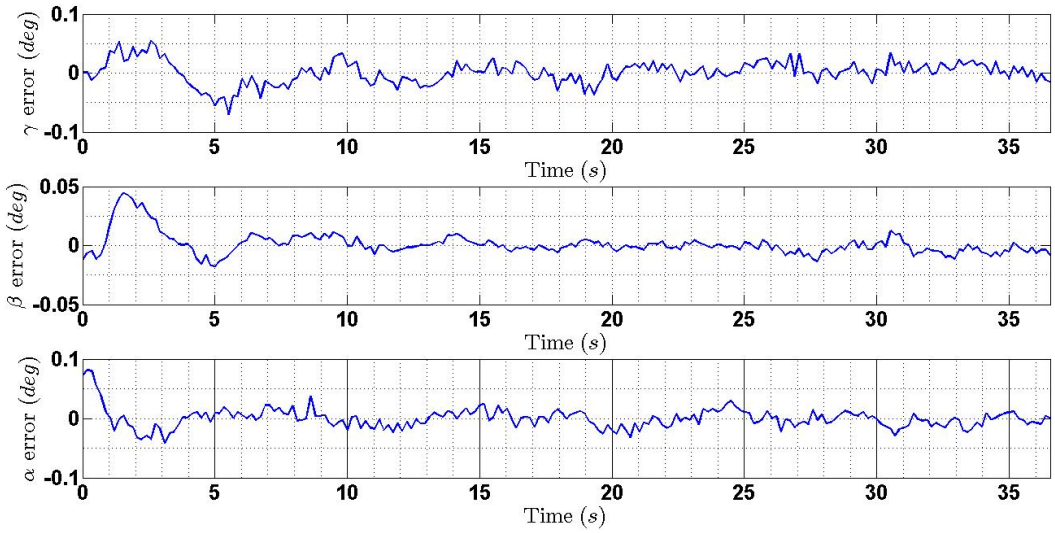


Figure 6.14: Orientation error of last iteration for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$ with AILC control.

Table 6.2: The comparison of maximum errors for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$.

Maximum error / Control Type	Pose error						Distance error
	$x(mm)$	$y(mm)$	$z(mm)$	$\gamma(deg)$	$\beta(deg)$	$\alpha(deg)$	
Without path correction	0.628	0.190	2.541	-0.254	0.096	0.107	2.619
With DPT control	-0.096	-0.168	0.284	0.076	-0.061	0.073	0.295
With ILC control	-0.046	-0.076	-0.159	-0.037	-0.013	0.038	0.168

Table 6.3: The comparison of RMS errors for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$.

RMS error / Control Type	Pose error						Distance error
	$x(mm)$	$y(mm)$	$z(mm)$	$\gamma(deg)$	$\beta(deg)$	$\alpha(deg)$	
Without path correction	0.426	0.081	1.715	0.135	0.074	0.056	1.769
With DPT control	0.048	0.079	0.122	0.029	0.018	0.021	0.153
With ILC control	0.016	0.037	0.060	0.014	0.005	0.012	0.072

Table 6.4: The accuracy improvement with AILC control superior to DPT control for Fanuc M20-iA moving forward along $\mathcal{P}_{d2}$ at speed $50mm/s$.

Percentage accuracy improvement						
x	y	z	γ	β	α	Distance error
66.5%	53.7%	51.0%	50.7%	71.8%	40.1%	53.0%

Chapter 7

Summary and Future Works

7.1 Summary of the Thesis

6-DOF serial robots as typical industrial robots have been extensively applied in industrial manufacturing and liberating human from tedious, dangerous works. However, there are still many potential applications in aerospace manufacturing due to their high accuracy requirements. Traditional calibration methods can only improve the static accuracy of industrial robots to a limited extent. However, the dynamic accuracy for approaching desired points and tracking desired paths is more significant for industrial robots in most applications. A lot of research work have been carried out for enhancing the dynamic accuracy of industrial robots in recent decades. Visual servoing based control methods have been intensively investigated due to their feasibility and effectiveness. In this thesis research, the accuracy enhancement of industrial robots for both desired points approaching and desired path tracking is explored. Kalman filter is selected to process the pose feedback information from VMI. An adaptive Kalman filter is introduced with velocity estimation and acceleration estimation. Then, different control schemes for accuracy enhancement are proposed and some experiments are implemented to verify the proposed schemes on FANUC robots with C-Track 780 providing visual measurements.

(1) Dynamic pose correction (DPC) scheme for enhancing the dynamic pose accuracy of industrial robots

The DPC scheme is a position based visual servoing (PBVS) method and designed to guide the

TCP of the end-effector to reach designated points with specific pose accurately. The designated points can be static or moving. Any pre-planned task point in task space can be set as current objective pose and it can be on-line updated. The DPC module is supposed to be working on the third computer which is independent to the robot controller. The DPC module includes real-time control algorithm and it can compute the movement commands to the robot controller in order to control the TCP of the end-effector moving toward the desired pose according to the pose feedback from visual measurement instrument. Moreover, an adaptive Kalman filter (AKF) is employed to reduce the influence from the image noise, blur and distortion and obtain the pose estimation of the end-effector. The DPC scheme is implemented on on FANUC LR Mate 200iC with C-Track 780 providing the visual measurements. The experimental results demonstrate that the pose accuracy can be improved to be less than $\pm 0.05mm$ for position and $\pm 0.05deg$ for orientation. The desired pose can be saved as the pose of the tool frame in $\mathcal{F}_{U_R}$ at the end of the first round. The task program can run in the robot controller independently or obtain fast adjustment to satisfy the required accuracy for the second round and late rounds. The learning based DPC scheme can also work as on-line teaching to transfer the off-line planned task to the information recorded in the position register of robot controller.

(2) Dynamic path tracking (DPT) scheme for achieving high path-tracking accuracy of industrial robots.

The DPT scheme is also a PBVS strategy and it is proposed to control the TCP of the end-effector attached on the tool flange of the industrial robot to keep tracking the desired path with high accuracy. The stability of the robot system with the compensation of the DPT control algorithm is proved by using Lyapunov stability theorem. In order to verify the proposed DPT scheme, some experiments are implemented on FANUC M20-iA with C-Track 780 as the photogrammetry sensor. The visual measurements of C-Track 780 are processed by the AKF to obtain current pose of the objects. The moving speed of the end-effector in these experiments is set as $25mm/s$. According to the data analysis of the experimental results, the path tracking accuracy can be enhanced to $\pm 0.10mm$ for position and $\pm 0.05deg$ for orientation, and the circle tracking accuracy can be improved to $\pm 0.20mm$ for position and $\pm 0.10deg$ for orientation. The high tracking accuracy achieved through the proposed DPT scheme can satisfy the precision requirements of some

applications, such as arc welding and sealing, in aerospace manufacturing.

(3) Adaptive iterative learning control (AILC) scheme for further improving the dynamic tracking performance of industrial robots.

Iterative learning control (ILC) is a tracking control method for repeating task systems and the tracking accuracy of the systems can be enhanced through learning process from iteration to iteration. As typical task-repetitive systems, industrial robots have been extensively applied to the researches of ILC in recent decades. In this thesis research, an AILC scheme in parallel with the DPT scheme is proposed to obtain further superior control performance. The AILC algorithm is to update the time-varying parameters adaptively along iteration axis. At the same time, new compensations are produced by the AILC algorithm to adjust the control input computed by the DPT module at each time interval based on the memorized data information and current feedback. The stability proof of the dynamic system with AILC scheme is proved through a positive definite Lyapunov-like energy function (LEF) by ensuring monotonic decreasing of LEF under some conditions along both time and iteration axis. The AILC scheme has been implemented successfully on FANUC M20-iA with C-Track 780 as VMI. Through data analysis and comparison of experimental results, it is proved that the AILC control module in parallel with the DPT module can outperform the DPT module alone. First, the transient control performance is improved with the AILC module. After certain limited iterations, high path tracking accuracy can be achieved even during the beginning $5s$ and the maximum distance error is less than $0.20mm$ when the TCP of the end-effector is following the desired path at speed $50mm/s$, which is twice of the speed applied in Chapter 5. Second, the pose accuracy can be confined to less than $0.10mm$ for position and $0.05deg$ for orientation through iterative learning. Moreover, the repetitive disturbances can be also overcome within certain iterations so that the vibrations can be significantly reduced. Therefore, the AILC scheme proposed in Chapter 6 is verified to be very effective to further improve the path tracking performance comparing with the DPT scheme.

7.2 Future Works

Although this thesis research has achieved some remarkable results, there are still many problems that needs further investigation:

- It is necessary to test the proposed control schemes on the other types of industrial robots, such as ABB and KUKA.
- The proposed control schemes can be designed as a control black box to be compatible with different industrial robots through initial learning and configurations.
- It is worth verifying the proposed DPT and AILC schemes to other types of path tracking, such as S-shape curve, square, etc..
- User-friendly human-machine interface can be developed so that non-professional operators can customize their own task by updating several specific parameters.
- In future, other artificial intelligence techniques, e.g., reinforcement learning, will be explored to improve the accuracy of industrial robots.

Appendix A

My Appendix

A.1 FANUC robots

FANUC is the abbreviation of ”Fuji Automatic Numerical Control”. FANUC Ltd. was established in 1972 as part of Fujitsu to develop servo system and early numerical control [151]. In this thesis research, two typical industrial robots, FANUC M20iA and FANUC LR Mate 200iC, are employed as experimental subjects. FANUC M20iA is the lab property of Concordia University and FANUC LR Mate 200iC is installed in the lab of ÉTS (École de technologie supérieure). They both are 6-DOF serial manipulators.

A.1.1 FANUC LR Mate 200iC

FANUC LR Mate 200iC, shown in Fig.A.1 (a), is a tabletop-size and light-weight industrial robot. It can be used for picking and packing, material handling, machine tending, part washing, testing and sampling, dispensing, material removal, assembly, education and entertainment. Its primary features are listed in Table.A.1 [152]. The work envelope of FANUC LR Mate 200iC is shown in Fig.A.2. There are fail-safe mechanical brakes on its 6 joints. Its unique software options provide real-time collision protection, singularity avoidance and internet connectivity. FANUC LR Mate 200iC works as a serial actuator with 6 joints and all the movements of the joints are commanded through Fanuc R-30iA Mate Controller, shown in Fig.A.1 (b). Fanuc R-30iA Mate Controller is the compact version of the standard Fanuc R-30iA controller and has much more advantages, e.g.,

shorter start-up time (less than 1 minute), fast restart (hot-start from the memorized movements and check collisions or exceptions safely), longer uptime, fewer maintenance requirements, and etc..

A.1.2 FANUC M20iA

FANUC M20iA as shown in Fig.A.3 (a) is a versatile 6-axes industrial robot with high inertia performance. It can be applied in many solutions, such as material handling and removal, assembly, water-jet/laser cutting, dispensing, picking and packing, machine load/unload, machine tending, testing and sampling, and other applications in industrial manufacturing. The primary features of FANUC M20iA are listed in Table.A.2 [153]. The work envelope of FANUC M20iA is shown in Fig.A.4. Comparing to FANUC LR Mate 200iC, FANUC M20iA has larger size, payload, work envelope and lower repeatability. It can accommodate several medium-payload applications together within its workspace. Moreover, the hollow upper arm&wrist and the shelf at the back of the upper arm provide enough space for wiring and mounting peripheral devices. Also, 2D vision cable from joint 1 to joint 3 is built in Fanuc M20iA as standard option. FANUC M20iA in Concordia is controlled through FANUC R-30iB controller, shown in Fig.A.3 (b). The communication between FANUC M20iA and FANUC R-30iB controller is using high-speed Ethernet connection to ensure fast efficient real-time control. The programs can be developed on teach pendant or on the third computer.

A.2 Optical CMM from Creaform

Comparing to most of high-precision optical CMMs in the market, e.g., FARO laser trackers and Nikon Metrology's optical CMM, C-Track 780 is much more cost-effective and user-friendly. C-Track 780, shown in Fig.A.5 (a), is an optical CMM with dual-camera manufactured by Creaform. C-Track controller as Fig.A.5 (c) is the connection bridge between C-Track 780 and a third computer through high speed firewire cable and ethernet cable respectively. Also, C-Track controller centralizes all incoming information from different equipments, such as C-Track 780, HandyProbe (shown in Fig.A.5 (d)). Table.A.3 demonstrates the technical specifications of C-Ttrack 780 from

the technical manual provided by Creaform. The measurement volume of C-Ttrack 780 is indicated in Fig.A.6. However, better accuracy can be obtained in the area excluding the borders and Z-dimension. The repeatability, the 3rd item in Table.A.3, is taking the maximum value from the measurements of the $500mm$-long standard artifact between 1.8m and 2m only along X or Y axis (dashed-line area in Fig.A.6) by using HandyProbe under dynamic referential mode. The artifact should be in a X-Y plane according to the sensor frame of C-Track. The single point repeatability, the 4th item in Table.A.3, is a distance deviation by putting the tip of HandyProbe in the calibration cone, shown in Fig.A.5 (e), with multiple orientations. The measurement speed of C-Track 780, the 6th item in Table.A.3, is the acquisition rate of the images per camera per second with two cameras measuring up to 100 optical reflectors at every image simultaneously.

C-Track 780 must be calibrated periodically (at most 20 days) by using the calibration bar as shown in Fig.A.5 (b) so as to have a premium performance. Calibration should be carried out in the actual measurement conditions, such as in the same stable temperature environment. All the targets on the calibration bar should be visible to the dual cameras of C-track 780. When calibrating C-Track 780, two calibration volumes, $3.8m^3$ and $7.8m^3$, can be opted. $7.8m^3$ is the maximum volume that C-Track 780 can take the measurement. If $3.8m^3$ is selected for calibrating, C-Track 780 is downgraded to the measurement volume of C-Track 380. The accuracy in $3.8m^3$ is better than in $7.8m^3$ according to the performance values in Table.A.3.

HandyProbe is an arm-free CMM and can be integrated with C-Track 780 for dynamic measurement and probing inspection. HandyProbe is useful for some static measuring, such as identifying specific point, line or plane, defining the coordinate frame attached on the objects. Also, HandyProbe can be calibrated by using the calibration cone.

The dual cameras of C-Track 780 acquire all the positioning targets in its measurement volume simultaneously. The positioning targets can be adhesive or magnetic reflectors as shown in Fig.A.5 (g) and (f). The adhesive reflectors are low-coat, ultralight and passive. A rigid body, as a geometry model, can be represented by a series of positioning targets which are liked rigidly with a reference frame. The VXtrack module is the software module provided by creaform for interpreting the visual measurements of C-Track 780 to high-precision dynamic tracking pose information of the measured point or object [142]. First, the coordinates of the targets in sensor frame are estimated

from the measurements of the two cameras based on the basic photogrammetry triangulation algorithm. Then, the position and orientation of the reference frame attached to the object, as a rigid body, can be deducted through 3D localization [154]. Combined with VXtrack module, C-Track 780 can be applied to monitor complex manufacturing process, robot guidance and calibration, inspecting the deformations, etc..

Figure A.1: (a) FANUC LR Mate 200iC, (b) Fanuc R-30iA Mate Controller [155].

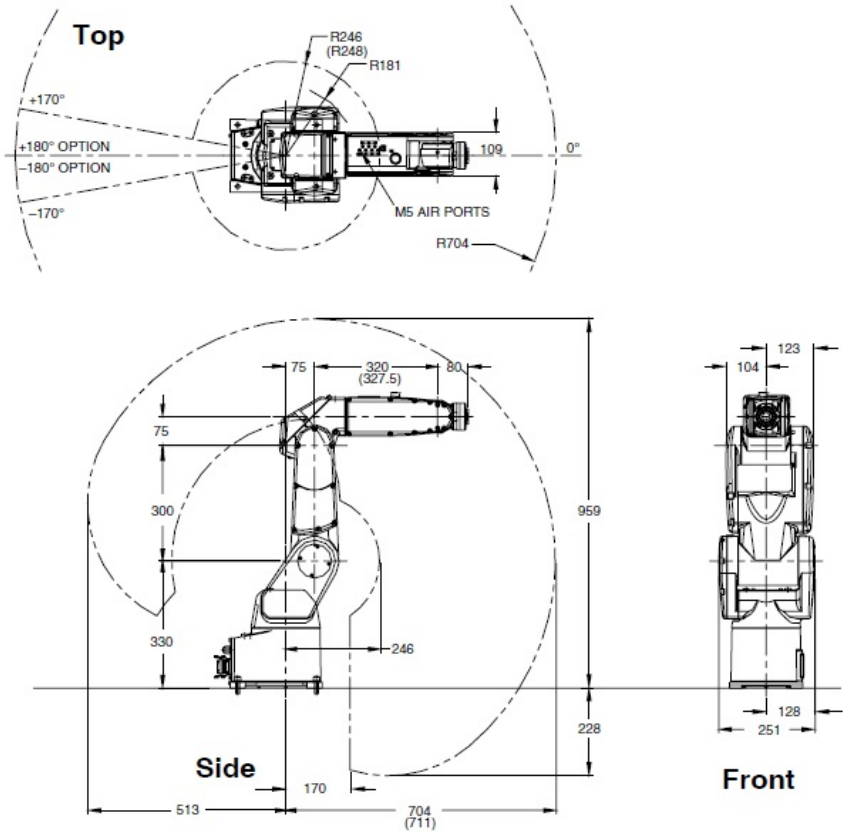


Figure A.2: The work envelope of FANUC LR Mate 200iC [152]. The unit for dimension is millimeter.

Figure A.3: (a) FANUC M20iA, (b) Fanuc R-30iB Controller [156].

Table A.1: FANUC LR Mate 200iC robot specification.

Item		Features
Axes		6
Payload at wrist (kg)		5
Reach (mm)		704
Repeatability (mm)		± 0.02
Interference Radius (mm)		181
Motion range (degrees)	$J1$	340
	$J2$	200
	$J3$	388
	$J4$	380
	$J5$	240
	$J6$	720
Motion speed (degrees/s)	$J1$	350
	$J2$	350
	$J3$	400
	$J4$	450
	$J5$	450
	$J6$	720
Mechanical brakes		6 axes
Mechanical weight (kg)		250
Vibration (m/s^2)		0.5 or less

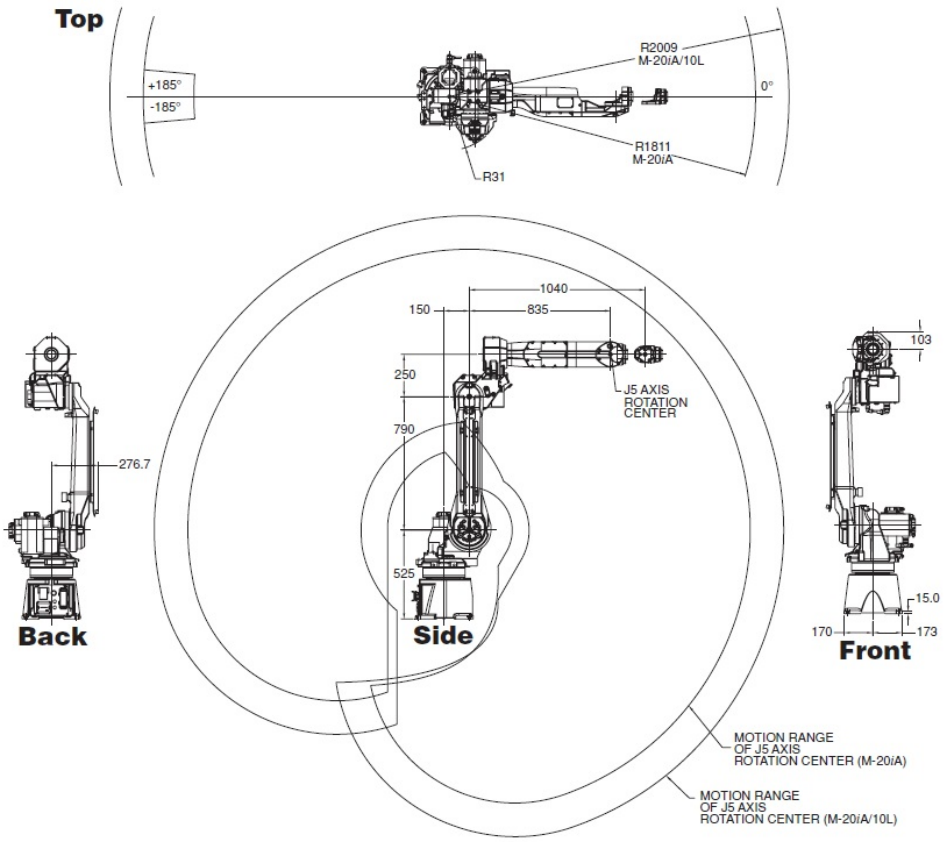


Figure A.4: The work envelope of FANUC M20iA [153]. The unit for dimension is millimeter.

Table A.2: FANUC M20iA robot specification.

Items		Features
Axes		6
Payload at wrist (kg)		20 at wrist
Reach (mm)		1811
Repeatability (mm)		± 0.08
Interference Radius (mm)		321
Motion range (degrees)	$J1$	$340/370$
	$J2$	260
	$J3$	458
	$J4$	400
	$J5$	3600
	$J6$	900
Motion speed (degrees/s)	$J1$	195
	$J2$	175
	$J3$	180
	$J4$	360
	$J5$	360
	$J6$	550
Mechanical brakes		6 axes
Mechanical weight (kg)		27
Vibration (m/s^2)		4.9 or less

Table A.3: The technical specifications of C-Track 780.

No.	Items		Performance
1	Measurement rate for 100 reflectors (reflectors/s)		up to 3000
2	Measurement rate with oversampling for 100 reflectors (reflectors/s)		up to $24,000$
3	Repeatability (RMS value in mm)		up to 0.025
4	Single point repeatability (RMS value in mm)	$3.8m^3$	up to 0.05
		$7.8m^3$	up to 0.055
5	Volumetric accuracy (RMS value in mm)	$3.8m^3$	up to 0.06
		$7.8m^3$	up to 0.065
6	Measurement speed (hz)		30
7	Temperature range (°C)		$-15 \sim 40$

Figure A.5: (a) C-Track 780 on a tripod, (b) Calibration bar, (c) Controller, (d) HandyProbe, (e) Calibration cone, (f) Adhesive reflectors, (g) magnetic reflectors.

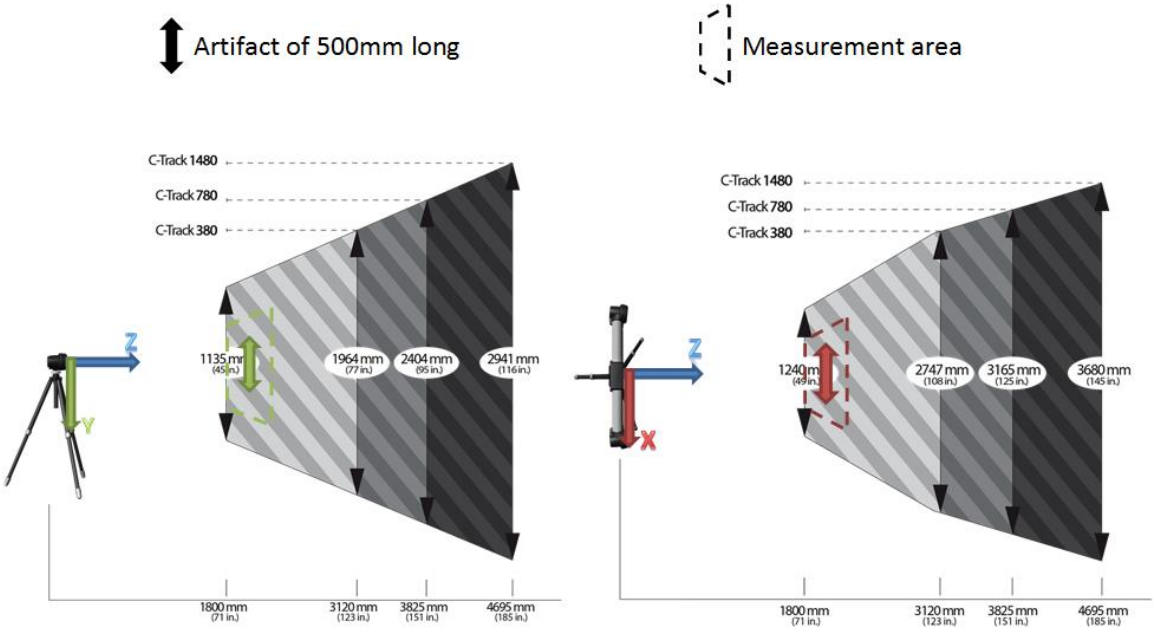


Figure A.6: The measurement volume of C-Track serials [157].

Bibliography

[1] ISO. Robots and robotic devices vocabulary, 2012.

[2] Jr George C Devol. programmed article transfer, U.S. Patent US2988237A, June 13, 1961.

[3] IFR. IFR presents World Robotics 2021 reports, 2021.

[4] Automate. UNIMATE//The First Industrial Robot vocabulary, 2022.

[5] Thomas Drake Miyano. Moon-based planetary defense campaign. *Journal of Space Safety Engineering*, 5(2):85–105, 2018.

[6] ALAN BOYLE. FAUB on the job: Watch robots work with mechanics to put together Boeing 777 jets vocabulary, 2017.

[7] Automate. Robots in Aerospace Applications vocabulary, 2022.

[8] Adel Olabi, Mohamed Damak, Richard Bearee, Olivier Gibaru, and Stephane Leleu. Improving the accuracy of industrial robots by offline compensation of joints errors. In *2012 IEEE international conference on industrial technology*, pages 492–497. IEEE, 2012.

[9] Nick Holt. Automotive robots reach for the sky. http://www.automotivelogisticsmagazine.com/uncategorized/automotive-robots-reach-for-the-sky, 2007 (accessed on Oct. 8, 2023).

[10] Russell Devlieg. High-accuracy robotic drilling/milling of 737 inboard flaps. *SAE International Journal of Aerospace*, 4(2011-01-2733):1373–1379, 2011.

[11] Mark Summers. Robot capability test and development of industrial robot positioning system for the aerospace industry. *SAE transactions*, 114(1):1108–1118, 2005.

[12] ABB. Absolute Accuracy Industrial robot option vocabulary, 2011.

[13] Albert Nubiola and Ilian A Bonev. Absolute calibration of an abb irb 1600 robot using a laser tracker. *Robotics and Computer-Integrated Manufacturing*, 29(1):236–245, 2013.

[14] Branko Karan and Miomir Vukobratović. Calibration and accuracy of manipulation robot models—an overview. *Mechanism and Machine Theory*, 29(3):479–500, 1994.

[15] Yi Min Zhao, Yu Lin, Fengfeng Xi, and Shuai Guo. Calibration-based iterative learning control for path tracking of industrial robots. *IEEE Transactions on industrial electronics*, 62(5):2921–2929, 2014.

[16] Russell C DeVlieg. Robotic manufacturing system with accurate control, March 24 2015. US Patent 8,989,898.

[17] William J Wilson, Carol C Williams Hulls, and Graham S Bell. Relative end-effector control using cartesian position based visual servoing. *Robotics and Automation, IEEE Transactions on*, 12(5):684–696, 1996.

[18] Farrokh Janabi-Sharifi, Lingfeng Deng, and William J Wilson. Comparison of basic visual servoing methods. *Mechatronics, IEEE/ASME Transactions on*, 16(5):967–983, 2011.

[19] Fujie Wang, Lulu Song, and Zhi Liu. Image-based visual servoing control for robot manipulator with actuator backlash. In *Informative and Cybernetics for Computational Social Systems (ICCSS), 2016 3rd International Conference on*, pages 272–276. IEEE, 2016.

[20] Mohammad Keshmiri and Wen-Fang Xie. Image-based visual servoing using a trajectory optimization technique. *IEEE/ASME Transactions on Mechatronics*, 2016.

[21] Cheng Li, Yuanqing Wu, Harald Löwe, and Zexiang Li. Poe-based robot kinematic calibration using axis configuration space and the adjoint error model. *IEEE Transactions on Robotics*, 32(5):1264–1279, 2016.

[22] Qingxuan Jia, Shiwei Wang, Gang Chen, Lei Wang, and Hanxu Sun. A novel optimal design of measurement configurations in robot calibration. *Mathematical Problems in Engineering*, 2018, 2018.

[23] Le Ma, Patrick Bazzoli, Patrick M Sammons, Robert G Landers, and Douglas A Bristow. Modeling and calibration of high-order joint-dependent kinematic errors for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 50:153–167, 2018.

[24] Tingting Shu, Sepehr Gharaaty, WenFang Xie, Ahmed Joubair, and Ilian A Bonev. Dynamic path accuracy using photogrammetry sensor. *IEEE/ASME Transactions on Mechatronics*, 23(3):1159–1170, 2018.

[25] Sepehr Gharaaty, Tingting Shu, Ahmed Joubair, Wen Fang Xie, and Ilian A Bonev. Online pose correction of an industrial robot using an optical coordinate measure machine system. *International Journal of Advanced Robotic Systems*, 15(4):1729881418787915, 2018.

[26] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.

[27] Klaus Schwab. *The fourth industrial revolution*. Currency, 2017.

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

[29] Pawar Digvijay. *Industrial robotics market: global Opportunity analysis and industry forcast, 2021-2030*. Allied Market Research, 2022.

[30] Low Kin Huat. *Industrial robotics: programming, simulation and applications*. Pro Literatur Verlag, Germany/ARS, Austria, 2006.

[31] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

[32] Carl Landau. High accuracy assembly of large aircraft components using coordinated arm robots. Technical report, SAE Technical Paper, 2016.

[33] Wikipedia. Coordinate measuring machine vocabulary, 2022.

[34] Connectus. Bringing Accuracy to Industrial Robots vocabulary, 2017.

[35] Substance. 6 Advantages and Disadvantages of Coordinate Measuring Machine vocabulary, 2022.

[36] Jean-Francois Larue, Daniel Brown, and Marc Viala. How optical cmms and 3d scanning will revolutionize the 3d metrology world. In *Integrated Imaging and Vision Techniques for Industrial Inspection*, pages 141–176. Springer, 2015.

[37] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[38] Bashar Alsadik. kalman filter. *Adjustment Models in 3D Geomatics and Computational Geophysics*, 4:299–326.

[39] SY Chen. Kalman filter for robot vision: a survey. *Industrial Electronics, IEEE Transactions on*, 59(11):4409–4420, 2012.

[40] Erik Cuevas, Daniel Zaldivar, and Raul Rojas. Kalman filter for vision tracking. *Measurement*, 33, 01 2005.

[41] Farrokh Janabi-Sharifi. Visual servoing: theory and applications. *Opto-Mechatronic Systems Handbook*, pages 15–1, 2002.

[42] Maurizio Ficocelli and Farrokh Janabi-Sharifi. Adaptive filtering for pose estimation in visual servoing. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 19–24. IEEE, 2001.

[43] Farrokh Janabi-Sharifi and Mohammed Marey. A kalman-filter-based method for pose estimation in visual servoing. *Robotics, IEEE Transactions on*, 26(5):939–947, 2010.

[44] Torkel Glad and Lennart Ljung. Velocity estimation from irregular, noisy position measurements. *IFAC Proceedings Volumes*, 17(2):1069–1073, 1984.

[45] Ajit Gopalakrishnan, Niket S Kaisare, and Shankar Narasimhan. Incorporating delayed and infrequent measurements in extended kalman filter based nonlinear state estimation. *Journal of Process Control*, 21(1):119–129, 2011.

[46] Ignacio Peñarrocha, Roberto Sanchis, and Julio Ariel Romero. State estimator for multi-sensor systems with irregular sampling and time-varying delays. *International Journal of Systems Science*, 43(8):1441–1453, 2012.

[47] Alireza Fatehi and Biao Huang. Kalman filtering approach to multi-rate information fusion in the presence of irregular sampling rate and variable measurement delay. *Journal of Process Control*, 53:15–25, 2017.

[48] Jingyi Wang, Yousef Alipouri, and Biao Huang. Multirate sensor fusion in the presence of irregular measurements and time-varying time delays using synchronized, neural, extended kalman filters. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2021.

[49] J.Hill and W.T. Park. *Real time control of a robot with a mobile camera*. SRI International, 1979.

[50] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108, 1973.

[51] Farrokh Janabi-Sharifi and William J Wilson. Automatic selection of image features for visual servoing. *IEEE Transactions on Robotics and Automation*, 13(6):890–903, 1997.

[52] Lingfeng Deng, Farrokh Janabi-Sharifi, and William J Wilson. Hybrid motion control and planning strategies for visual servoing. *Industrial Electronics, IEEE Transactions on*, 52(4):1024–1040, 2005.

[53] Chien-Chern Cheah, Chao Liu, and Jean-Jacques E Slotine. Adaptive tracking control for robots with unknown kinematic and dynamic properties. *The International Journal of Robotics Research*, 25(3):283–296, 2006.

[54] Do-Hwan Park, Jeong-Hoon Kwon, and In-Joong Ha. Novel position-based visual servoing approach to robust global stability under field-of-view constraint. *Industrial Electronics, IEEE Transactions on*, 59(12):4735–4752, 2012.

[55] Yi Min Zhao, Yu Lin, Fengfeng Xi, Shuai Guo, and Puren Ouyang. Switch-based sliding mode control for position-based visual servoing of robotic riveting system. *Journal of Manufacturing Science and Engineering*, 139(4):041010, 2017.

[56] Mohammad Keshmiri, Wen-Fang Xie, and Abolfazl Mohebbi. Augmented image-based visual servoing of a manipulator using acceleration command. *IEEE Transactions on Industrial Electronics*, 61(10):5444–5452, 2014.

[57] Amir Hajiloo, Mohammad Keshmiri, Wen-Fang Xie, and Ting-Ting Wang. Robust online model predictive control for a constrained image-based visual servoing. *IEEE Transactions on Industrial Electronics*, 63(4):2242–2250, 2016.

[58] J Fuentes-Pacheco, J Ruiz-Ascencio, and JM Rendón-Mancha. Binocular visual tracking and grasping of a moving object with a 3d trajectory predictor. *Journal of applied research and technology*, 7(3):259–273, 2009.

[59] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[60] Asnor Juraiza Ishak and Sarmad Nozad Mahmood. Eye in hand robot arm based automated object grasping system. *Periodicals of Engineering and Natural Sciences*, 7(2):555–566, 2019.

[61] Martin Gaudreault, Ahmed Joubair, and Ilian A Bonev. Local and closed-loop calibration of an industrial serial robot using a new low-cost 3d measuring device. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4312–4319. IEEE, 2016.

[62] Gregory Flandin, François Chaumette, and Eric 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), volume 3, pages 2741–2746. IEEE, 2000.

[63] Abdul Muis and Kouhei Ohnishi. Eye-to-hand approach on eye-in-hand configuration within real-time visual servoing. *IEEE/ASME transactions on Mechatronics*, 10(4):404–410, 2005.

[64] Ali AlBeladi, Evan Ripperger, Seth Hutchinson, and Girish Krishnan. Hybrid eye-in-hand/eye-to-hand image based visual servoing for soft continuum arms. *IEEE Robotics and Automation Letters*, 7(4):11298–11305, 2022.

[65] Wen-Chung Chang. Binocular vision-based 3-d trajectory following for autonomous robotic manipulation. *Robotica*, 25(5):615–626, 2007.

[66] François Chaumette, Seth Hutchinson, and Peter Corke. Visual servoing. *Springer handbook of robotics*, pages 841–866, 2016.

[67] ROB NIKOLEWSKI. This robot can check oil and gas pipelines to help prevent spills. https://www.latimes.com/business/la-fi-pipelines-robots-20170605-story.html, 2017 (accessed on Oct. 8, 2023).

[68] Jordan Pearson. Robots are wandering through oil pipelines that have never been inspected. https://www.vice.com/en/article/4x397w/robots-are-wandering-through-oil-pipelines-that-have-never-been-inspected, 2015 (accessed on Oct. 8, 2023).

[69] Jordan Pearson. What is machine vision. https://www.roboticstomorrow.com/article/2019/12/what-is-machine-vision/14548, 2019 (accessed on Oct. 8, 2023).

[70] Tarek A. Khaled, Ouassima Akhrif, and Ilian A. Bonev. Dynamic path correction of an industrial robot using a distance sensor and an adrc controller. *IEEE/ASME Transactions on Mechatronics*, 26(3):1646–1656, 2021.

[71] Murray Garden. Learning control of actuators in control systems, January 12 1971. US Patent 3,555,252.

[72] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic systems*, 1(2):123–140, 1984.

[73] M. Uchiyama. Formulation of high speed motion pattern of mechanical arm by trial. *Trans. Soc. Instrument Contr. Engineers*, 14(6):706–712, 1978.

[74] Yangquan Chen and Changyun Wen. *Iterative learning control: convergence, robustness and applications*. Springer-Verlag, 1999.

[75] Jian-Xin Xu and Ying Tan. *Linear and nonlinear iterative learning control*, volume 291. Springer Berlin, 2003.

[76] Hyo-Sung Ahn, Kevin L Moore, and YangQuan Chen. *Iterative learning control: robustness and monotonic convergence for interval systems*. Springer Science & Business Media, 2007.

[77] J.X. Xu, S.K. Panda, and T.H. Lee. *Real-time Iterative Learning Control: Design and Applications*. Advances in Industrial Control. Springer London, 2008.

[78] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. A survey of iterative learning control. *Control Systems, IEEE*, 26(3):96–114, 2006.

[79] Hyo-Sung Ahn, YangQuan Chen, and Kevin L Moore. Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 37(6):1099, 2007.

[80] Jian-Xin Xu. A survey on iterative learning control for nonlinear systems. *International Journal of Control*, 84(7):1275–1294, 2011.

[81] Dong Shen and Youqing Wang. Survey on stochastic iterative learning control. *Journal of Process Control*, 24(12):64–77, 2014.

[82] Jian-Xin Xu and Zhihua Qu. Robust iterative learning control for a class of nonlinear systems. *Automatica*, 34(8):983–988, 1998.

[83] Abdelhamid Tayebi and Marek B Zaremba. Iterative learning control for non-linear systems described by a blended multiple model representation. *International Journal of control*, 75(16-17):1376–1384, 2002.

[84] Han-Fu Chen and Hai-Tao Fang. Output tracking for nonlinear stochastic systems by iterative learning control. *Automatic Control, IEEE Transactions on*, 49(4):583–588, 2004.

[85] Suguru Arimoto. Mathematical theory of learning with applications to robot control. In *Adaptive and Learning Systems*, pages 379–388. Springer, 1986.

[86] Samer S Saab. Stochastic p-type/d-type iterative learning control algorithms. *International Journal of Control*, 76(2):139–148, 2003.

[87] Roberto Horowitz. Learning control of robot manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 115(2B):402–411, 1993.

[88] Yangquan Chen, Changyun Wen, and Mingxuan Sun. A robust high-order p-type iterative learning controller using current iteration tracking error. *International Journal of Control*, 68(2):331–342, 1997.

[89] Jian-Xin Xu, Xiao-Wei Wang, and Lee Tong Heng. Analysis of continuous iterative learning control systems using current cycle feedback. In *American Control Conference, Proceedings of the 1995*, volume 6, pages 4221–4225. IEEE, 1995.

[90] Notker Amann, David H Owens, Eric Rogers, and Anja Wahl. An h∞ approach to linear iterative learning control design. *International Journal of Adaptive Control and Signal Processing*, 10(6):767–781, 1996.

[91] Yang Quan Chen, Changyun Wen, Jian-Xin Xu, and Mingxuan Sun. High-order iterative learning identification of projectile's aerodynamic drag coefficient curve from radar measured velocity data. *Control Systems Technology, IEEE Transactions on*, 6(4):563–570, 1998.

[92] Tao Liu and Furong Gao. Robust two-dimensional iterative learning control for batch processes with state delay and time-varying uncertainties. *Chemical Engineering Science*, 65(23):6134–6144, 2010.

[93] Richard W Longman. Designing iterative learning and repetitive controllers. In *Iterative learning control*, pages 107–146. Springer, 1998.

[94] Richard W Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10):930–954, 2000.

[95] Chiang-Ju Chien and Jing-Sin Liu. A p-type iterative learning controller for robust output tracking of nonlinear time-varying systems. *International Journal of Control*, 64(2):319–334, 1996.

[96] A Tayebi, S Abdul, MB Zaremba, and Y Ye. Robust iterative learning control design: application to a robot manipulator. *Mechatronics, IEEE/ASME Transactions on*, 13(5):608–613, 2008.

[97] Abdelhamid Tayebi and Jian-Xin Xu. Observer-based iterative learning control for a class of time-varying nonlinear systems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 50(3):452–455, 2003.

[98] Hanfu Chen. Almost sure convergence of iterative learning control for stochastic systems. *Science in China Series F*, 46(1):67–79, 2003.

[99] Toshiharu Sugie and Toshiro Ono. An iterative learning control law for dynamical systems. *Automatica*, 27(4):729–732, 1991.

[100] Notker Amann, David H Owens, and Eric Rogers. Predictive optimal iterative learning control. *International Journal of Control*, 69(2):203–226, 1998.

[101] V Hatzikos, J Hätönen, and DH Owens. Genetic algorithms in norm-optimal linear and non-linear iterative learning control. *International Journal of control*, 77(2):188–197, 2004.

[102] David H Owens and Jari Hätönen. Iterative learning control—an optimization paradigm. *Annual Reviews in Control*, 29(1):57–70, 2005.

[103] David H Owens and JJ Hatonen. A new optimality based adaptive ilc-algorithm. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, volume 3, pages 1496–1501. IEEE, 2002.

[104] Chong-Ho Choi and Tae-Jeong Jang. Iterative learning control in feedback systems based on an objective function. *Asian Journal of Control*, 2(2):101–110, 2000.

[105] David H Owens, Christopher T Freeman, and T Dinh Van. Norm optimal iterative learning control with intermediate point weighting: theory, algorithms and experimental evaluation. *IEEE Transactions on Control Systems Technology*, 21(3):999–1007, 2013.

[106] Won G Seo, BH Park, and Jin S Lee. Adaptive fuzzy learning control for a class of nonlinear dynamic systems. *International journal of intelligent systems*, 15(12):1157–1175, 2000.

[107] Jian-Xin Xu and Jing Xu. A new fuzzy logic learning control scheme for repetitive trajectory tracking problems. *Fuzzy sets and systems*, 133(1):57–75, 2003.

[108] R Precup, Stefan Preitl, József K Tar, Marius-Lucian Tomescu, Márta Takács, Péter Korondi, and Péter Baranyi. Fuzzy control system performance enhancement by iterative learning control. *Industrial Electronics, IEEE Transactions on*, 55(9):3461–3475, 2008.

[109] Chiang-Ju Chien, Chun-Te Hsu, and Chia-Yu Yao. Fuzzy system-based adaptive iterative learning control for nonlinear plants with initial state errors. *Fuzzy Systems, IEEE Transactions on*, 12(5):724–732, 2004.

[110] Chiang-Ju Chien. A combined adaptive law for fuzzy iterative learning control of nonlinear systems with varying control tasks. *Fuzzy Systems, IEEE Transactions on*, 16(1):40–51, 2008.

[111] Chiang-Ju Chien and Li-Chen Fu. An iterative learning control of nonlinear systems using neural network design. *Asian Journal of Control*, 4(1):21–29, 2002.

[112] Tommy WS Chow, Xiao-Dong Li, and Yong Fang. A real-time learning control approach for nonlinear continuous-time system using recurrent neural networks. *Industrial Electronics, IEEE Transactions on*, 47(2):478–486, 2000.

[113] Jin Young Choi and Hyun Joo Park. Use of neural networks in iterative learning control systems. *International Journal of Systems Science*, 31(10):1227–1239, 2000.

[114] Ping Jiang and Rolf Unbehauen. Iterative learning neural network control for nonlinear system trajectory tracking. *Neurocomputing*, 48(1):141–153, 2002.

[115] Masaki Yamakita, Masashi Ueno, and Teruyoshi Sadahiro. Trajectory tracking control by an adaptive iterative learning control with artificial neural networks. In *American Control Conference, 2001. Proceedings of the 2001*, volume 2, pages 1253–1255. IEEE, 2001.

[116] Qing-Yuan Xu, Yun-Shan Wei, Jing Cheng, and Kai Wan. Adaptive ilc design for nonlinear discrete-time systems with randomly varying trail lengths and uncertain control directions. *International Journal of Control, Automation and Systems*, 21(9):2810–2820, 2023.

[117] Mark French and Eric Rogers. Non-linear iterative learning by an adaptive lyapunov technique. *International Journal of Control*, 73(10):840–850, 2000.

[118] William Messner, Roberto Horowitz, W-W Kao, and Michael Boals. A new adaptive learning rule. *Automatic Control, IEEE Transactions on*, 36(2):188–197, 1991.

[119] Dong Sun and F Xiaolun Shi Yunhui Liu. Adaptive learning control for cooperation of two robots manipulating a rigid object with model uncertainties. *Robotica*, 14(04):365–373, 1996.

[120] Jian-Xin Xu and Ying Tan. Composite energy function based learning control. *Linear and Nonlinear Iterative Learning Control*, pages 95–114, 2003.

[121] Zhihua Qu and Jianxin Xu. Asymptotic learning control for a class of cascaded nonlinear uncertain systems. *Automatic Control, IEEE Transactions on*, 47(8):1369–1376, 2002.

[122] Jian-Xin Xu and Zenn Z Bien. *The frontiers of iterative learning control*. Springer, 1998.

[123] Jian-Xin Xu and Ying Tan. A composite energy function-based learning control approach for nonlinear systems with time-varying parametric uncertainties. *Automatic Control, IEEE Transactions on*, 47(11):1940–1945, 2002.

[124] Zhihua Qu and Jian-Xin Xu. Model-based learning controls and their comparisons using lyapunov direct method. *Asian Journal of Control*, 4(1):99–110, 2002.

[125] Abdelhamid Tayebi and Chiang-Ju Chien. A unified adaptive iterative learning control framework for uncertain nonlinear systems. *Automatic Control, IEEE Transactions on*, 52(10):1907–1913, 2007.

[126] Masaki Togai and Osamu Yamano. Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach. In *Decision and Control, 1985 24th IEEE Conference on*, pages 1399–1404. IEEE, 1985.

[127] Paola Bondi, Giuseppe Casalino, and Lucia Gambardella. On the iterative learning control theory for robotic manipulators. *Robotics and Automation, IEEE Journal of*, 4(1):14–22, 1988.

[128] Sang-Rok Oh, Zeungnam Bien, and Il Hong Suh. An iterative learning control method with application to robot manipulators. *Robotics and Automation, IEEE Journal of*, 4(5):508–514, 1988.

[129] JY Choi and JS Lee. Adaptive iterative learning control of uncertain robotic systems. *IEE Proceedings-Control Theory and Applications*, 147(2):217–223, 2000.

[130] Mikael Norrlöf. An adaptive iterative learning control algorithm with experiments on an industrial robot. 2002.

[131] Abdelhamid Tayebi. Adaptive iterative learning control for robot manipulators. *Automatica*, 40(7):1195–1203, 2004.

[132] A Tayebi and S Islam. Adaptive iterative learning control for robot manipulators: Experimental results. *Control Engineering Practice*, 14(7):843–851, 2006.

[133] St Kawamura, F Miyazaki, and S Arimoto. Applications of learning method for dynamic control of robot manipulators. In *Decision and Control, 1985 24th IEEE Conference on*, pages 1381–1386. IEEE, 1985.

[134] S Islam and A Tayebi. New adaptive iterative learning control (ailc) for uncertain robot manipulators. In *Electrical and Computer Engineering, 2004. Canadian Conference on*, volume 3, pages 1645–1651. IEEE, 2004.

[135] Chiang-Ju Chien and Abdelhamid Tayebi. Further results on adaptive iterative learning control of robot manipulators. *Automatica*, 44(3):830–837, 2008.

[136] Farrokh Janabi-Sharifi and Mohammed Marey. *A kalman-filter-based method for pose estimation in visual servoing*, volume 26. IEEE, 2010.

[137] Kenichi Kanatani, Yasuyuki Sugaya, and Hirotaka Niitsuma. Triangulation from two views revisited: Hartley-sturm vs. optimal correction. *In practice*, 4:5, 2008.

[138] Joseph SC Yuan. A general photogrammetric method for determining object position and orientation. *Robotics and Automation, IEEE Transactions on*, 5(2):129–142, 1989.

[139] ISO 9283:1998(en). Manipulating industrial robots — Performance criteria and related test methods vocabulary, 2022.

[140] Peter I Corke. Visual control of robot manipulators-a review. *Visual servoing*, 7:1–31, 1993.

[141] Yuan-Chih Peng, Devavrat Jivani, Richard J Radke, and John Wen. Comparing position- and image-based visual servoing for robotic assembly of large structures. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1608–1613. IEEE, 2020.

[142] Creaform. Dynamic Tracking: VXtrack vocabulary, 2022.

[143] Tomas Kubela, Ales Pochyly, and Vladislav Singule. Assessment of industrial robots accuracy in relation to accuracy improvement in machining processes. In *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*, pages 720–725. IEEE, 2016.

[144] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.

[145] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[146] Rafael Kelly, Victor Santibáñez Davila, and Julio Antonio Loría Perez. *Control of robot manipulators in joint space*. Springer Science & Business Media, 2005.

[147] Abb Robotics. Absolute accuracy industrial robot option. https://library.e.abb.com/public/e69d8dd25cd7d36bc125794400374679/AbsAccPR10072EN_R6.pdf, 2011 (accessed on Oct. 8, 2023).

[148] Peter I Corke and Oussama Khatib. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer, 2011.

[149] Zeungnam Bien and Kyung M Huh. Higher-order iterative learning control algorithm. In *IEE Proceedings D (Control Theory and Applications)*, volume 136, pages 105–112. IET, 1989.

[150] Jian-Xin Xu. The frontiers of iterative learning control-ii. *Systems, control and information*, 46(5):233–243, 2002.

[151] Wikipedia. FANUC vocabulary, 2022.

[152] FANUC Robotics. LR Mate 200iC Series vocabulary, 2022.

[153] FANUC Robotics1. FANUC M10iA/M20iA vocabulary, 2022.

[154] David W Eggert, Adele Lorusso, and Robert B Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine vision and applications*, 9(5):272–290, 1997.

[155] Surplus Record. R30ia mate controller vocabulary, 2022.

[156] FANUC Robotics2. FANUC R30iB vocabulary, 2022.

[157] Creaform. Technical Specifications: Dynamic Tracking vocabulary, 2022.