PATH-INVARIANT AND TIME-OPTIMAL MOTION CONTROL FOR INDUSTRIAL ROBOTS

by

Joonyoung Kim

B.Eng., Chung-Ang University (Seoul, Korea), 2002

M.A.Sc., University of Toronto, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2017

Abstract

This thesis presents practical methods for planning and control to improve the motion performance of industrial robots. Particular attention is given to the commercial six degrees-of-freedom articulated robot with a low-cost generic controller. A comparative study of motion control methods demonstrated that both smooth trajectory planning and filtering techniques, when combined with a traditional Proportional-Derivative control, are limited in achievable performance due to reduced accelerations (smooth trajectory) or large path-distortions (filtering technique). Instead, faster and more accurate motion is achieved with a low-order trajectory, namely, trapezoidal velocity profile, with feedforward control design based on an elastic model. The key component that makes the latter approach more appealing is the delay-free dynamic input shaper embedded in the feedforward control. Following the results from the comparative study, two innovations are proposed to satisfy the path-invariant and time-optimal motion. First, an online time-optimal trapezoidal velocity profile planned along multiple path segments is presented. The trajectory can be planned for arbitrary boundary conditions and path curvatures with only four system-dynamics computations per path segment. Next, a novel control method based on the flexible joint dynamic model is proposed to achieve high tracking performance for the proposed trajectory. The proposed nonlinear multivariable control can place the closed-loop poles arbitrarily with only position and velocity feedback. Real-world experiments with commercial industrial robots are carried out to validate the presented methods.

Preface

All of the work presented henceforth was conducted in the Collaborative Advanced Robotics and Intelligent Systems Laboratory at the University of British Columbia, Point Grey campus.

A version of Chapter 5 has been accepted for publication. [Kim, J.] and Croft, E. A. Co-designed Trajectory Planning and Feedforward Control for Industrial Robots. IEEE Transactions on Robotics.

J. Kim was the lead investigator for the project located in Chapter 5, where J. Kim was responsible for all major areas of concept formation, data collection and analysis, as well as the majority of manuscript composition. E. A. Croft was the supervisory author on these projects and was involved throughout the projects in concept formation and manuscript edits. The experiments in the thesis were performed at Hyundai Robotics Co., Ltd., Yongin, South Korea. The experimental results were disclosed upon the permission of Hyundai Robotics Co., Ltd.

Table of Contents

List of Tables

List of Figures

List of Symbols

Superscripts

c	Controlled
d	Desired reference

Subscripts

0	Initial
d	Derivative
f	Final
m	Motor
r	Link
p	Position
v	Velocity
ff	Feedforward
fb	Feedback
$flex$	Flexible part
max	Maximum
min	Minimum
$rigid$	Rigid part

Chapter 3

$\Re$	Real number
S	Position in task space
T_v	Transfer function of the inner-loop of cascaded PD control
n	The number of dimensions
s	Laplace variable or path parameter
t	Time
ω	Frequency
$\mathbf{q}_m$, $\mathbf{q}_r$	Generalized coordinates
$\mathbf{M}_m$, $\mathbf{M}_r$ (J_m, J_r)	Inertia (scalar)
$\mathbf{C}_r$	Corolis-Centrifugal force
$\mathbf{G}$	Gravity force
$\mathbf{K}$ (k)	Joint stiffness (scalar)
$\mathbf{r}$ (r)	Reduction ratio (scalar)
$\boldsymbol{\tau}_{rigid}$, $\boldsymbol{\tau}_{flex}$, $\boldsymbol{\tau}_r$	Joint torque
$\mathbf{u}$, $\mathbf{u}_{ff}$, $\mathbf{u}_{fb}$ (u, u_{ff}, u_{fb})	Controller input (scalar)
$\mathbf{e}_m$, $\mathbf{e}_r$	Tracking error
$\mathbf{K}_p$, $\mathbf{K}_v$	Gain of cascaded PD control: outer-loop gain, inner-loop gain
$\bar{\mathbf{K}}_p$	$\mathbf{K}_p\mathbf{K}_v$
$\boldsymbol{\psi}$	Perturbation
$\mathbf{f}$, $\mathbf{f}^d$, $\mathbf{f}^c$	Path, user-defined path, controlled path

Chapter 4

J	Jerk
A	Acceleration
γ	Unit step function
δ	Unit impulse function
T_j	Jerk period
T_a	Acceleration period
f_n	Natural frequency
h	Impulse response
α	Coefficient of the input shaping filter
T_z , T_m	Time delay: input shaping filter, simple moving average filter
ζ	Damping ratio
$\mathbf{K}_s$, $\mathbf{K}_r$, $\mathbf{K}_I$	Full-state control gain: deflection, link velocity, integral
$\mathbf{x}$	State vector
$\mathbf{R}$, $\mathbf{Y}$, $\mathbf{G}$	Frequency response model: input, output, transfer function

Chapter 5

λDLC	Lower bound of Dynamic Limit Curve
μDLC	Upper bound of Dynamic Limit Curve
$SE(3)$	Special Euclidean Group
T_k	Motion time
R_v, R_τ, R_p	Performance measures: peak velocity, peak torque, actuator utilization
Υ	Admissible region
k	Test index
t_a, t_c, t_f	Time along a trapezoidal velocity profile
$\mathbf{C}$	Coriolis-Centrifugal effect
$\mathbf{M}$	Inertia (link and motor)
$\mathbf{J}$	Manipulator Jacobian
$\mathbf{m}$, $\mathbf{b}$, $\mathbf{d}$, $\mathbf{e}$	Dynamic terms mapped into a parameterized path
$\mathbf{f}_m$, $\mathbf{f}_v$, $\mathbf{f}_c$	Friction: motor, viscous, Coulomb
$\boldsymbol{\tau}$	Motor torque
$\mathbf{p}$, $\mathbf{q}$	Position: in task space, in joint space
$\mathbf{u}_c$, $\mathbf{u}_k$, $\mathbf{u}_{k+1}$	Unit direction vectors
$\boldsymbol{\lambda}(\lambda)$	Weight of the path velocity and acceleration (scalar)

Chapter 6

C	Column space
ε	Singular perturbation parameter
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$	Matrices defined in the singular perturbation model
$\mathbf{J}$	Motor inertia
$\mathbf{K}_s$	Gain for slow control
$\mathbf{K}_p$, $\mathbf{K}_d$	PD gain for the slow control
$\bar{\mathbf{K}}$	Normalized joint stiffness
$\mathbf{L}$	Gain for fast control
$\mathbf{L}_p$, $\mathbf{L}_d$	PD gain for the fast control
$\mathbf{v}$	Total input of singular perturbation control
$\mathbf{v}_f$	Fast input of singular perturbation control
$\mathbf{v}_s$	Slow input of singular perturbation control
$\mathbf{z}$	Fast variable
$\boldsymbol{\eta}$	Boundary layer
$\boldsymbol{\varphi}$	Manifold

List of Abbreviations

ASME	The American Society of Mechanical Engineers
CNC	Computer Numerical Control
DOF	Degrees Of Freedom
DLC	Dynamic Limit Curve
EI	Extra Insensitive
FRF	Frequency Response Function
HIL	Hardware In the Loop
IEEE	The Institute of Electrical and Electronics Engineers
IST	Input Shaping Technique
KLC	Kinematic Limit Curve
LTI	Linear Time Invariant
ODE	Ordinary Differential Equation
PD	Position-Derivative
PITOM	Path-Invariant and Time-Optimal Motion
RNEA	Recursive Newton-Euler Algorithm
SMA	Simple Moving Average
SPC	Singular Perturbation Control
TCP	Tool Center Point
TVP	Trapezoidal Velocity Profile
ZV	Zero Vibration

Acknowledgements

I would first of all like to thank my supervisor Professor Elizabeth Croft for her guidance and support during my study at the University of British Columbia. I am very grateful for her comments and suggestions in preparing our publication work. I would also like to thank Professor Yusuf Altintas and Professor Ryozo Nagamune in the Department of Mechanical Engineering for their advices on my thesis work.

I am very thankful to Professor Alessandro De Luca in the University of Rome for evaluating this thesis. His review provided excellent critiques and valuable comments, ensuring that important papers were not omitted in this thesis.

This work was supported by Hyundai Robotics Co., Ltd., S. Korea. I would like to thank the Director of the Robot Research Institute and the leaders of the Control Research Team, Dong-Hyeok Kim, Hyun-Kyu Lim, and Tae-Sun Kang for allowing me to pursue my Ph.D. I would like to thank my colleagues at the company, specially Soo-Jong Kim, Won-Hyuk Choi, Dr. Jong-Kyu Oh, Hyeon-Ho Shin, Dr. Dae-Sik Kim, Si-Hyun Ryu, Minjeong Kim, and Dr. Yong-Kuk Lee, for providing a joyful atmosphere and helping me set up for experiments.

Most of all, I would like to thank my family including my wife Deborah Y. Moon, my daughter Yeseo (Isabelle), and my son Hankyum (Aaron) for their love, support, and tolerance for my absence from home during this work.

Lastly, I would like to acknowledge all the researchers and developers of industrial robots who envisioned a better world by creating better robots. Their past and present struggles to solve a number of daily real-world problems are greatly appreciated.

Chapter 1: Introduction

Advancements in industrial robotics research continue to provide significant benefits by reducing the need for manual labor that is repetitive, dull, and hazardous to humans. As robots become more autonomous and accurate, a larger set of tasks can be roboticized, freeing workers to focus on more challenging and cognitively demanding tasks. The roboticization of workspace has increased dramatically in recent years. World Robotics Report 2016 (http://www.ifr.org/industrial-robots/statistics) states that "By 2019, the number of industrial robots deployed worldwide is expected to increase to around 2.6 million units. That is about one million units more than in the year of 2015." Industrial robots are ubiquitous in traditional manufacturing processes such as welding, handling, cutting, painting, and are expected to play a major role in Industry 4.0 (linking online simulation with real-time manufacturing), as well as global retooling for new materials, shorter product cycles, increased variety of product mix and collaborative manufacturing.

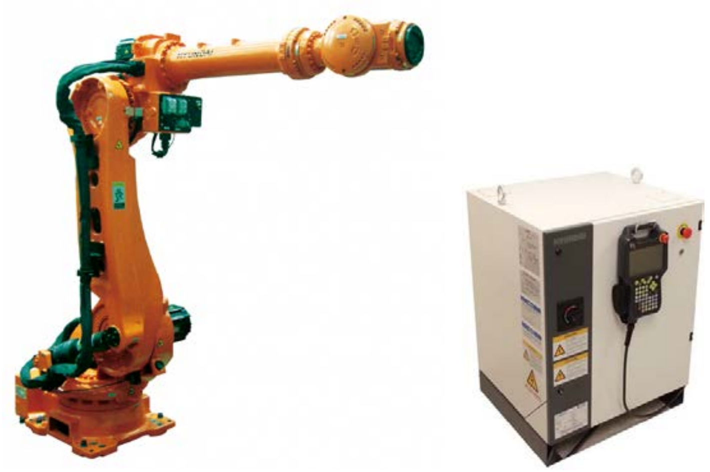


Figure 1.1. HS180 (left, 6 axis, payload: 180 kg, major application: spot welding and handling) and Hi5a controller and teach pendant (right, commercialized in 2008), used with permission of Hyundai Robotics Co., Ltd.

The most common configuration for an industrial robot includes a serial linkage (mechanical arm) and a control system as shown in Figure 1.1. Each joint of the mechanical arm is actuated by an electric motor and power is transmitted to each link via a high ratio gearbox. Thus, industrial robots can have high payloads with relatively small motors. The controller is connected to a teach pendant that provides a user interface so that robot motion can be programmed conveniently. This design allows an industrial robot to be a multi-purpose machine. That is, one industrial robot can be used for a wide range of applications in various industries. For instance, a typical industrial robot can be used from point-to-point tasks, e.g., spot welding and material handling, to line following tasks, e.g., arc welding, painting, and deburring. In most of these applications, the motion performance of the robot is critical to the overall system efficiency as cycle time is often limited by the speed and required accuracy of the robot trajectory. Therefore, research to improve the motion performance has attracted much attention from the robotics community. For 'high' robot motion performance, i.e., fast motion with a small tracking error, designing a 'good' control strategy is crucial and is the main topic of this thesis.

1.1 Motivation and Problem Statement

While it is important to obtain high performance for industrial robots, obtaining such a goal is a challenging task. First of all, the articulated manipulator such as the one shown in Figure 1.1 is a complex mechanical system. The links connected by six (typical) to eight joints are coupled and therefore the inertia changes depending upon the robot's configuration. In addition, the physical space where the robots and the user directly interact with environment, referred to as the *task space*, is not the same as the space where each motor is controlled, referred to as the *joint* *space*. The relationship between these two physical spaces is also highly nonlinear. Due to the complexity of the dynamics and geometry, the architecture of the motion control system for industrial robots is generally hierarchical, that is, the trajectory-planning module (or simply the planning module - the higher control level) and the servoing module (the lower control level). By adopting a hierarchical strategy, the problem of motion control is simplified because each module can be designed and implemented individually.

In parallel with the architecture of motion control systems, research on motion performance of industrial robots has typically been pursued independently in two groups, namely, planning and control. The first group, referred to as *the planning approach* in this thesis, focuses on improving the reference trajectory. In particular, the problem of minimum-time trajectory planning based on a rigid model has received much attention in this category (Bobrow *et al*., 1985; Shin and McKay, 1985). Other topics that are often mentioned include smooth trajectory planning (Macfarlane and Croft, 2003) and the filtering techniques (Singer and Seering, 1990). These works aim to generate smooth reference trajectories such that a simple Proportional-Derivative (PD) control can achieve good tracking performance. In the meantime, the second group, or *the control approach*, focuses on improving the control loop such that it can guarantee global asymptotic stability for arbitrary reference trajectories. The main focus of the control approach is modeling and control synthesis for the nonlinear multivariable dynamics of a robot with elastic joints. A number of control methods have been proposed based on the flexible joint dynamic model. Some of well-known works include feedback linearization (Spong, 1987), feedforward control (De Luca, 2000), passivity-based control (Brogliato *et al*., 1995), and the singular perturbation method (Spong *et al*., 1985).

Although many important results have been achieved by both approaches, the methods proposed in the literature are not widely used in industry. Some advanced model-based methods are used for only certain industrial robots such as QuickMove^{TM} and TrueMove^{TM} of ABB robots and the torque controller of the KUKA lightweight robot. However, details of the methods are rarely reported in the literature due to company confidentiality policies (Moberg, 2010). Many of commercial industrial robots today still use a simple planning and control schemes such as trapezoidal velocity profiles (TVP) and PD control. However, there are clear reasons why more advanced motion control methods are not broadly adopted in industry. For example, the classical minimum-time trajectory (Bobrow *et al*., 1985; Shin and McKay, 1985) requires solving dynamics along the entire path in advance, which is impractical to implement in a low-cost controller in spite of the recent advancements in computer technology. Even if the method can be implemented with increased computing power, tracking control is another problem as a simple PD control cannot obtain good tracking performance for such a bang-bang/high frequency trajectory. Some researchers have shown that increasing the smoothness of the trajectory can result in improved tracking performance. However, this approach is not always a viable solution in robotic applications where a high demand for cycle time is required, e.g., spot welding. Lastly, many proposed controllers are based on assumptions that cannot be applied to a typical industrial robot. That is, most of the tracking controllers proposed in the literature require accurate estimation of the link states including acceleration and jerk. However, the measurement of a typical industrial robot is available only on the motor side. In principle, the link acceleration and jerk can be computed based on the measured position and velocity and the dynamic model. In that case, measurement of higher order derivatives would not be required. However, in practice, the estimation of these derivatives is based on an inaccurate model, and the estimation error increases with the order of differentiation. This is, perhaps, the reason why a successful implementation of a model-based control based on high order derivatives is rarely reported in the literature. In contrast, it is reported that a control method based on lower order derivatives can be implemented successfully (Albu-Schäffer *et al.*, 2007).

In summary, there exists a sizable gap between the advanced motion control methods proposed by academic researchers and the simple methods widely used in industry. This gap must be bridged in order for both academia and industry to move forward together for the advancement of motion control technology of industrial robots.

1.2 Objective

It is the intention of this thesis to bridge the gap between academia and industry in the field of motion control technology of industrial robots. The effort starts with a hypothesis that there exist significant theoretical benefits in academic research work and at the same time there are clear rationales why industry sticks to obsolete techniques. Therefore, this thesis tries to maintain a practical standpoint with a focus on implementable solutions built upon the strict industrial constraints. To do so, a thorough review of the existing methods available in the literature is performed. The review includes works done by both the planning and control approaches to find the optimal combination of the planning and control strategies that can lead to high motion performance. It is also necessary to identify the constraints of the motion control system of a typical industrial robot for practical implementations. Based on the theoretical review and identified industrial constraints, this thesis aims to find the answers to the following questions.

- What is the best motion performance that can be obtained solely by improving the reference trajectory assuming a generic robot controller?

- What is a practical strategy for generating minimum-time trajectories under strict industrial constraints such as bounded computation time and online applications?
- How should the control loop be designed with available measurements to obtain good tracking performance for a minimum-time trajectory?

1.3 Contributions

These main contributions of this thesis are as follows:

1. A general motion control strategy that encompasses both planning and control to achieve high motion performance for industrial robots with elastic joints.
2. The concept of path-invariant and time-optimal motion (PITOM) that specifies how each planning and control must be designed to guarantee both high speed and accurate tracking simultaneously.
3. A comparison between the planning approach and control approach in a common framework. The methods that are compared include S-curve profiling, the input shaping technique (IST), feedforward control, and full-state control. The optimal S-curve profiling is proposed such that the jerk period is minimized to achieve vibration free responses under a high gain PD control. The reason why the control approach provides more opportunity for performance improvement than the planning approach is clearly demonstrated.
4. A solution to the online minimum-time trajectory planning along multiple path segments with minimum computational load. The proposed method obtains the minimum-time trajectory with only four system-dynamics computations per path segment, which enables the robot to quickly respond to sensor inputs with limited computation sources.

5. A method for the multivariable nonlinear control of an industrial robot with elastic joints without estimating high order states on the link side. A singular perturbation method is employed to find a unique set of gains such that the closed-loop poles are placed at the desired locations.

1.4 Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2: The related work in the literature on motion control methods for industrial robots is reviewed, focusing on the theoretical benefits and implementation issues. Both planning and control approaches are overviewed.

Chapter 3: In this chapter, the motion control system of a typical industrial robot is introduced. The constraints that must be satisfied for practical implementation in a low-cost robot controller are also discussed. Then, a concept for motion control, referred to as the PITOM is introduced. The concept specifies how trajectory and control must be designed to guarantee high motion performance.

Chapter 4: The purpose of this chapter is to compare some of the well-known motion control methods in order to investigate whether they satisfy the concept of the PITOM. Four methods, S-curve profiling, the input shaping technique (IST), feedforward control, and full-state control are analyzed and tested by experiment. The first two methods represent the planning approach and the latter two methods are for the control approach. The comparative study reveals that: (i) the optimal S-curve trajectory is too slow especially in short-pitch motions due to the added jerk period at every switching point; (ii) the IST will generate poor path accuracy due to significant path distortions; (iii) an equally good or better performance can be achieved by the control approach with a non-smooth trajectory input. The key component that makes the control approach more appealing than the planning approach is the delay-free dynamic input shaper that is embedded in the feedforward control. The result from this chapter has been submitted for publication in *IEEE/ASME Transactions on Mechatronics* (Kim and Croft, submitted in October 2016).

Chapter 5: This chapter presents the generation of a time-optimal, low-order trajectory that allows for replanning on the fly and planning for continuous motions to meet the online requirement. The well-known minimum-time trajectory (Bobrow *et al*., 1985; Shin and McKay, 1985) is approximated by the TVP trajectory to achieve time-optimality. It is shown that it is sufficient to solve dynamics only at four points along a path segment to achieve time-optimality as long as the path and dynamic terms can be locally linearized. Simulation and experiment are performed to verify the proposed trajectory. The work in this chapter has been accepted for publication in *IEEE Transactions on Robotics* (Kim and Croft, accepted in October 2016).

Chapter 6: In this chapter, a novel nonlinear control strategy is proposed to obtain good tracking performance for the proposed time-optimal trajectory. The control method is a combination of feedforward control and a singular perturbation method for feedback control. The feedforward control plays a dominant role in the accurate tracking of the time-optimal trajectory. The singular perturbation method is applied to suppress the residual vibrations at steady state and to guarantee stability. A method of selecting the feedback gain such that an arbitrary closed-loop performance can be achieved is also discussed. The proposed control method is simulated and experimentally demonstrated with the proposed time-optimal trajectories. The control method has been submitted for publication in *IEEE Transactions on Control Systems Technology* (Kim and Croft, submitted in February 2017).

Chapter 7: This chapter summarizes the main contributions of the thesis followed by concluding remarks and future work on the proposed motion control methods.

Chapter 2: Literature Review

In this chapter, the existing work for motion control methods of industrial robots is overviewed. Section 2.1 reviews the planning approach, which focuses on improving the motion performance either by time-optimal trajectories or smooth trajectories. Section 2.2 reviews the control approaches with emphasis on multivariable nonlinear control for the flexible joint dynamic model.

2.1 Planning Approach

There are a vast number of research works regarding trajectory planning for industrial robots. Since the main topic of the thesis is related to motion performance in terms of speed and accuracy, the review is limited to the related work such as minimum-time trajectory planning, smooth trajectory planning, and IST. A review on generation of motion trajectories and their frequency analysis can be found in Biagiotti and Melchiorri (2008).

2.1.1 Minimum-Time Trajectory Planning

2.1.1.1 Offline Trajectory Planning

The pioneering work for minimum-time trajectory planning is to plan a trajectory such that the robot travels along a specified path in minimum time given the maximum torque constraint. The path is represented with a single path parameter, s, typically $0 \le s \le 1$, and then the rigid model is represented with the path parameter, s . This simplifies the problem of finding the maximum acceleration and velocity because these maximum values can be calculated on the two dimensional phase plane, $(s, \dot{s})$, which is much simpler than finding them in the n –

dimensional vector space of the original dynamic model. In this method, the equation of motion is calculated along a specified path forward and backward in time to find the switching points. Due to its large computational burden, the method is sometimes referred to as *the off-line method*. Shin and McKay (1985), Bobrow *et al.* (1985), and Pfeiffer and Johanni (1987) first introduce the concept of minimum-time trajectory planning along specified paths. Shiller and Lu (1992) improve the initial work by considering the singular point, where it is shown that the smooth transition near a singular point can be obtained by sliding along the tangential direction of the velocity limit curve. The time-optimal trajectory along specified path is C^1- continuous and therefore generates discontinuity at every switching instance. Since the time-optimal trajectory of this early work contains high frequency content, the controlled path generates a large tracking error if a simple control loop, e.g., PD control, is used for tracking control. Therefore, the original method has been improved in various ways. Dahl (1992) shows that the method can be extended to a flexible joint robot, where the maximum path velocity is found based on the maximum/minimum jerk along a parameterized path. Constantinescu and Croft (2000) consider torque rate as an additional constraint in order to generate a trajectory that is C^2- continuous. They report that considering the torque rate significantly complicates the problem because the phase plane moves from two to three dimensions. Hollerbach (1984) shows that the time-optimality can be obtained by uniformly scaling the trajectory in the time domain. De Luca (2002) extends Hollerbach's work (1984) for a dynamic model with elastic joints. Verscheure *et al.* (2009) and Reynoso-Mora *et al.* (2013) show that the minimum-time trajectory along specified paths can be formulated as a convex optimization problem. In recent years, research on minimizing energy by scaling the time-optimal trajectory has received an increasing attention (Wigstrom *et al.*, 2013). Pham (2014) provides a complete solution for time-optimal trajectory along a specified path. The concept of minimum-time trajectory planning along specified path has also been applied to machine tools (Altintas and Erkorkmaz, 2003; Dong *et al*., 2007).

In summary, these off-line methods provide a useful framework to design a time-optimal trajectory as they allow one to view the relationship between the trajectory and the robot dynamics in a transparent way via a simple two dimensional phase plane. However, the large computational burden (typically 100 to 1000 times of dynamics computation per path segment) makes it difficult to implement the method in a typical robot controller due to their limited computational power. In particular, the requirement that the dynamics is solved along the entire path makes it extremely difficult to modify the planned trajectory on the fly. This capacity is often required in robot applications to meet demanded changes in operating speeds, implement collision avoidance trajectories, and move to new waypoints modified by sensor-events.

2.1.1.2 Online Trajectory Planning

To address the need for adaptive robot motion, online trajectory planning is essential. A light computational burden is the key requirement in online trajectory planning as a new trajectory should be generated in one or several control cycles. In addition, for a simple control loop, the online trajectory needs to be smooth to achieve a good tracking performance. Lloyd and Hayward (1993) propose a blending technique to generate a smooth transition between two path segments. Macfarlane and Croft (2003) propose a fifth order trajectory with a sine wave template for online applications considering constant kinematic bounds. The transition between two path segments is obtained by utilizing the work in Lloyd and Hayward (1993). The experimental results show that the proposed trajectory obtains a smaller tracking error and faster settling time compared with a traditional TVP trajectory under PD control. Lambrechts *et al*. (2005) propose a fourth-order trajectory for feedforward control assuming that the controlled system can be approximated with a double-mass model. It is assumed that the maximum kinematic bounds up to the snap, the derivative of the jerk, are known and the boundary conditions are zeros. Kröger and Wahl (2010) optimize S-curve trajectories (3rd order) such that the traveling time is minimized under the maximum velocity, acceleration, and jerk. They propose a detailed map that can be used to quickly determine the types of trajectory for different motion states under constraints. Broquère *et al.* (2008) propose a method to transit between two S-curve trajectories so that a smooth and continuous trajectory can be obtained at every waypoint. Katzschmann *et al.*, (2013) optimizes the acceleration for a third order trajectory by solving dynamics at every sampling time but this method suffers from instability. Nguyen *et al*. (2008) show that the TVP-like trajectories that consist of a ramp-up, a cruise with constant velocity, and a ramp-down can be calculated by a single numerical algorithm. However, their method is limited to zero boundary conditions. Ezair *et al*. (2014) extend the work in Nguyen *et al*., (2008) for arbitrary boundary conditions.

One common limitation of the online trajectories proposed in the literature is that the maximum kinematic bounds are somehow assumed to be known and it is not generally discussed how these bounds are chosen in an optimal sense. Therefore, some researchers focused on generating online time-optimal trajectories. Antonelli *et al.* (2007) and Kim *et al*. (2010) calculate the minimum-time TVP trajectory under the maximum torque/velocity constraints and proposed a method to generate continuous motions on the fly. Antonelli *et al.* (2009) improve the work in Antonelli *et al.* (2007) by considering jerk limits and also apply the work in Macfarlane and Croft (2003) for continuous motions in Cartesian space. In Kim *et al.* (2010), it is shown that the cycle time of commercial industrial robots could be improved by up to 30 % compared with the TVP designed with constant maximum kinematic bounds. However, the TVP trajectories is only 2nd order and it contains high frequency content that may excite the flexible modes of the robot when it is directly used as the reference input to the control loop. However, discussion on how to design the control loop for a low order trajectory is not mentioned in these methods. Another challenge for these online trajectories is that it is not guaranteed that the resulting path is invariant under different kinematic bounds especially during the transition between two path segments. For instance, the transition path is often generated by connecting the boundary conditions of each path segment by a new trajectory (Antonelli et al., 2007; Broquère *et al*., 2008; Kim *et al.*, 2010). In their methods, the geometric shape of the transition path is dependent upon the boundary conditions specified by trajectories along the adjacent path segments. This may lead to a large deviation of the path from a desirable one as the geometry of the path is affected by the boundary conditions at the beginning and the end of a transition. The method in Macfarlane and Croft (2003) does not possess such problem as the transition is generated by a geometric blending technique (Lloyd and Hayward, 1993). However, it requires that the boundary accelerations are enforced to be zero, which can cause an unnecessary delay in cycle times. In summary, this review did not uncover any online strategies for trajectory planning in the literature that are path-invariant (a path unaffected by its time-law) and time-optimal.

2.1.2 Input Shaping Technique

It is well known that a better tracking performance can be achieved as the order of the trajectory increases. However, obtaining a high order trajectory is in fact a challenging task. First of all, a closed-form solution of a third or higher order trajectory with arbitrary boundary conditions such that it follows a TVP-like trajectory, namely, ramp-cruise-ramp, is not available in the literature (Haschke *et al*., 2008; Ezair *et al*., 2014). One must resort to an iterative numerical method to calculate such high order trajectories. As it will be shown later in Chapter 4, the order of trajectory is inversly proportional to the magnitude of the frequency response of the trajectory. In other words, the effect of increasing the order of the trajectory is equivalent to lowering the magnitude of the frequency response of the trajectory, thus reducing chances to excite the resonant frequencies of the robot.

One can achieve the same goal by utilzing a low pass filter as long as the frequency content is concerned, which is, perhaps, an easier approach than planning with a high order trajectory. Among exisiting filters, the IST has received much attention for controlling mechanical systems due to its simple design with a small delay (Singhose, 2009). The idea of the IST is that it applies a series of impulse reponses to the input trajectory such that the filtered trajectory contains a periodic signal. If the system's pole is known, then the IST can be designed such that the impulse response can have the fundamental frequency that is equal to the pole of the plant. This is equialevant to adding a zero to the reference trajectory, which can cancel the plant pole when the filtered trajectory is applied to the plant (Hillsley and Yurkovich, 1993). The IST was originally proposed for an single degree-of-freedom (DOF) linear time invariant (LTI) system (Singer and Seering, 1990). It has been shown that the IST is effective to obtain a vibration-free response to simple dynamic systems such as cranes (Parker *et al*., 1999) and more complex, yet relatively linear, systems such as machine tools (Altintas and Khoshdarregi, 2012). The IST was also applied to a time-varying system with multiple modes (Khorrami, 1995) and industrial robot (Chang and Park, 2005; Park *et al*., 2006). However, it is challenging to implement a practical IST for a time-varying system with multiple modes such as industrial robots. In Khorrami (1995), an adaptive scheme for IST is proposed to control a two-link flexible manipulator. In this method, the difficulty of dealing with configuration-dependent flexible modes is relaxed by applying feedback linearization in the control loop to cancel nonlinearity and estimating the plant poles by measuring the tip vibrations. Chang and Park (2005) and Park *et al*. (2006) apply the IST to industrial robots, where the adaptive IST is applied to the first joint while a fixed IST is applied to the second and third joints. One interesting result from Chang and Park (2005) is that a simple low pass filter that is tuned at a constant bandwidth achieves almost the same residual vibrations as the adaptive IST. In Chang and Park (2005), the adaptive IST exhibits some chattering phenomina which is not found in the low pass filter. In fact, a recent work indicates that the estimating flexible modes of a typical industrial robot is more complex than it is generally known in the literature (Moberg, 2010). In this work, it is suggested that there could be several unknown flexible modes at high frequency that cannot be estimated by a commonly known method. The IST requires exact knowlge of the flexible modes to obtain vibration-free responses. Since the IST is not a low pass filter, it is fundamentaly incapable of handling unknown flexible modes at high frequencies. In addition, as discussed in more detail in Chapter 4, the IST and many of filtering techniques can distort the geometric shape of the originally planned trajectory (Altintas and Khoshdarregi, 2012).

2.1.3 Summary

In this section, methods to improve the motion performance for industrial robots have been reviewed. The classical time-optimal trajectory along specified paths is difficult to implement due to the heavy computational load. However, it provides a clear insight into the relation between the dynamics along specified paths. The research on online trajectory planning has pursued computationally efficient trajectories that can be planned for arbitrary motion states.

However, the constant kinematic bounds can underutilize the capability of the robot's achievable performance. In addition, the high order trajectories may suffer from increased computational load due to a numerical apporach. Another problem of the online trajectories is the variation of the path depending on the boundary conditions. The IST is effective to eliminate unwanted vibrations for a simple dynamic system but it is difficult to apply to industrial robots due to the complex dynamics and the phenominon of path-distortion.

2.2 Control Approach

Minimum-time trajectories can only be realized if they can be tracked accurately by the control loop. There are a large number of research papers dealing with tracking control for industrial robots. Therefore, this review focuses on work that is most relevant to model-based nonlinear multivariable tracking control. Adaptive control and robust control are also important topics but are not discussed in this thesis.

2.2.1 Rigid Model-Based Control

Research on tracking control for robot manipulators goes back to the early 80's. The key concepts utilized in this research are *feedback linearization* and *feedforward control*, based on nonlinear control, assuming that the robot is perfectly rigid (Slotine and Li, 1991; Craig, 2005). Feedback linearization for rigid models is sometimes referred to as the *computed torque control*. The main idea of this method is to cancel all nonlinear and configuration-dependent terms such as inertial force, Coriolis-centrifugal effect, and gravity force in the inner control loop so that the outer control loop becomes a unit mass system, which is a linear, decoupled, and configuration-independent system. In order for this control method to be realized, an efficient calculation of robot dynamics is essential. Luh *et al*. (1980) propose an efficient way to calculate the inverse dynamics, referred to as the recursive Newton-Euler algorithm (RNEA). The RNEA calculates the joint torque recursively utilizing the tree-structure of the robot manipulator, resulting in an $O(n)$ algorithm. Park *et al*., (1995) formulate the RNEA algorithm based on Lie group, which results in a more elegant and simpler algorithm than the original one. Asada and Kanade (1983) and An *et al*. (1998) apply the computed torque control to a direct-drive robot manipulator assuming that their systems can be approximated to ideal rigid-body dynamic models. They show that the tracking error could be decreased significantly by the computed torque control. However, they both report that the methods could not obtain good position accuracy for an arbitrary trajectory, for example, a trajectory with infinite jerks. The main idea of the feedforward control is similar to computed torque control. It pre-calculates all the necessary joint torque based on the motion command and feeds this term forward to the control loop. One major difference is that the dynamics is computed outside the loop in feedforward control whereas it is computed inside the loop in the computed torque method. This difference is an important consideration for implementation. In Craig (2005), it is argued that feedforward scheme is a preferred choice since the performance is less sensitive to the sampling frequency of the control loop. Leahy and Saridis (1987) verify the effectiveness of the feedforward control by experiment with a commercial industrial robot.

The nonlinear control based on a rigid-body model has been improved by several researchers to cope with minimum-time trajectories proposed in Bobrow *et al*. (1985) and Shin and McKay (1985). In Dahl and Nielson (1990) and Dahl (1994), it is argued that the nominal C^1 time-optimal trajectory proposed in the minimum-time trajectories cannot be tracked accurately because at least one joint is always saturated during motions and there is no room for torque to compensate disturbances. They propose, *the path-velocity controller* that can modify the nominal trajectory in case when it is saturated. The path-velocity controller is expressed in terms of the path parameter with the knowledge on the controller model and is designed outside the primary control loop, thus not affecting the stability and control performance. The experimental results for a commercial industrial robot show an improved tracking performance compared with a direct use of the time-optimal trajectory as the reference input (Dahl, 1994). Kieffer *et al*. (1997) proposes a method similar to Dahl and Nielson (1990) but the tracking performance can meet a prescribed tolerance. Bianco and Gerelli (2011) extend the original work of Dahl (1994) by including additional kinematic constraints. One significant challenge when implementing the path-velocity controller is that the controller must be parameterized by the path parameter. As it is shown in Chapter 5, the scheme is not suitable for online trajectories where different path parameters must be assigned for different path segments. In this case, it is not clear how to parameterize the controller at arbitrary motion states for different path parameters.

2.2.2 Flexible Joint Model-Based Control

2.2.2.1 High-Order State Feedback Control

In the mid-1980s, a research paper was published from industry. The paper argues that the rigid model is not accurate enough to describe a typical industrial robot (Sweet and Good, 1984; Good *et al*., 1985). The main reason is that the joint elasticity associated with the drive systems cannot be ignored and it must be considered for control design to achieve an acceptable control performance. Since then, a vast amount of research work have been performed on modeling and control for flexible robots with elastic joints. However, the dynamic model with elastic joints is much more complex compared with the rigid model. De Luca and Lucibello (1998) show that the flexible joint dynamic model can only be linearized by dynamic state feedback. The control input has its own dynamics that cannot be expressed algebraically in terms of state variable. Nicosia and Tomei (1998) discuss the necessary and sufficient conditions such that the flexible joint model can be feedback linearizable. Spong (1987) proposed *the simplified flexible joint model*, where the inertial coupling between the motor and the link is ignored by assuming that the kinetic energy of the rotor is due to its own rotation. It is argued that such assumption is correct for a high gear ratio and the model can be linearizable by static state feedback, as opposed to general flexible joint model. Since the simplified flexible joint model is much easier for control design compared with the general flexible joint model, the former has been studied extensively. Amongst others, feedback linearization, which is the counterpart of the computed torque for rigid model, is most often cited in the literature. Moberg and Hanssen (2008) discuss the feedback linearization from an industrial point of view. They claim that the exact linearization of the nonlinearity is difficult to achieve in reality due to the limitation of the achievable sampling rate and measurement requirements. The difficulty could be partially related to the fact that the feedback linearization requires feedback up to jerk on the link side, which is problematic for a typical industrial robot where only measurement on the motor side is available. In fact, real-world experimental results of feedback linearization are hardly reported in the literature. Albu-Schäffer and Hirzinger (2001) report that the feedback linearization could not be implemented with the available computing power of the given controller with sampling frequency of 1 kHz. There are other control methods that have been derived from the simplified flexible joint model. Brogliato *et al*. (1995) compare three nonlinear controllers that are derived from different theoretical backgrounds. The control methods that are compared include decoupling-based control (Seibert and Suarez, 1990), back-stepping based control (Nicosia and Tomei, 1998), and passivity-based control (Lozano and Brogliato, 1992). They show that the three control methods are globally stable with respect to the reference trajectory but they all require feedback up to jerk, similar to feedback linearization. Their simulation results indicate that the passivity-based control performs relatively better compared with other methods in terms of smoothness of the torque. The decoupling-based scheme requires a high gain on the motor side as the performance on the link side heavily depends on the error on the motor side. The back-stepping controller is beneficial in terms of the systemic derivation of the control law but it can exhibit input chattering as the stiffness increases. In Brogliato *et al*. (1998), the control methods discussed in Brogliato *et al*. (1995), are compared experimentally with additional control methods, PD controller, Slotine and Li controller (Slotine and Li, 1988), and singular perturbation approach (Spong *et al*., 1987). Interestingly, the Slotine and Li controller which is effectively the same as the computed torque control obtains the best performance overall. The nonlinear controller based on the flexible model, however, exhibits very poor performance, generating a large tracking error at steady state or even instability. The authors report that the most critical problem of the control methods derived from the flexible joint dynamic model is the input chattering which is presumably induced by the inaccurate estimation of the high order states such as acceleration and jerk. In fact, the major difference between the flexible model-based controller in Brogliato *et al*. (1995) and the Slotine and Li controller is that the former requires feedback up to jerk whereas only position and velocity feedback are necessary for the latter. In general, the high order states such as acceleration and jerk can only be estimated by numerical differentiation from measurements or by computing based on the dynamic model. In either case, it is unavoidable that the estimates on acceleration and jerk contain significant error due to measurement noise or model uncertainty.

2.2.2.2 Low-Order State Feedback Control

Due to the difficulty of estimating high-order states, some researchers focused on developing nonlinear controllers that only require position and velocity feedback. Tomei (1991) proves that the PD control that utilizes only motor position and velocity can obtain global stability for the flexible joint model. Kelly *et al*. (1994) shows that the method proposed in Tomei (1991) can be improved with filtered velocity feedback. De Luca (2000) proposes a feedforward control that utilizes only motor position and velocity for feedback control. The control structure is very attractive from an industrial point of view because the method can be easily combined with a PD controller implemented on the motor side, which is the standard controller for many industrial robots. Some researchers show that PD control can be combined with online gravity compensation with measured motor positions as opposed to the well-known constant compensation scheme with desired link positions (De Luca *et al*., 2005; Ott *et al*., 2008).

However, the performance on the link side is less satisfactory compared with full-state control because the feedback control utilizes only motor states, i.e., partial-state feedback (De Luca et al., 2007). The control method using the singular perturbation method, shortly *the singular perturbation control* (*SPC*) in this thesis, has received much attention because it provides a theoretical justification for designing a nonlinear controller based on a reduced-model. In this method, the flexible joint dynamic model is split into two time-scales. One is the fast system governed by the deflection between the motor and the link, or the ‘fast’ variable. The other is the rigid model that is ‘slow’ assuming that the joint stiffness is sufficiently high. The composite control law is then applied to each subsystem, typically velocity control on the fast system and a PD control on the slow system. Spong *et al*. (1985) applies a manifold-based control for the rigid model. Spong and Khorasani (1987) propose a corrective control, namely, a power series of the slow control, to improve the accuracy of the reduced model. It is suggested that a PD control on the fast system be used instead of a velocity control, especially for a demanding trajectory (Hung, 1991; Wilson and Irwin, 1993). Readman and Belanger (1992) show that the SPC can be applied to a direct-drive robot where the inertial coupling between the motor and the link cannot be neglected. Yuan and Stepanenko (1993) propose a full-state controller that is built upon the SPC, requiring feedback up to acceleration on the link side. Ghorbel and Spong (2000) develop a condition of the perturbation parameter to guarantee stability of the SPC. Taghirad and Khosravi (2012) propose a corrective PID control for the slow system such that it achieves a global stability. Some real-world experiments are carried out to show that the composite control is more effective than the rigid control, e.g., computed torque control (Gandhi and Ghorbel, 2005; Salmasi *et al*., 2010). However, a failed experiment with the SPC is also reported in Lessard *et al*. (2012). In Brogliato *et al*. (1998), it is reported that the SPC can exhibit a poor tracking performance for a demanding trajectory. The SPC is also applied to robots with flexible links (Siciliano and Book, 1998; Vakil *et al*., 2011).

In spite of this extensive research on the SPC, there are some limitations. First and most importantly, no simple way to select the gain such that it achieves the desired closed-loop performance is available in the literature. Selecting the gain based on each subsystem often leads to instability. Although one can examine whether the joint stiffness is large enough for the designed controller to guarantee stability by checking some complex conditions (Ghorbel and Spong, 2000), designing a new stiffness is difficult in reality when such conditions are not met. Second, successful simulation and experimental results have been achieved by using relatively smooth reference trajectories. This is perhaps due to the fact that the separation in two-time scales assumes a high stiffness value, or a small perturbation parameter. Therefore, if the joint stiffness is not sufficiently high, then the fast varying trajectory may excite some of the closed-loop poles that are not considered in the SPC model. However, the reference trajectory for commercial industrial robots is commonly designed in a time-optimal manner in order to minimize the cycle time (Antonelli *et al*., 2007; Björkman *et al*., 2008; Kim *et al*., 2010). As implied in Lessard *et al*., (2012), it is unclear how to design the SPC for a demanding trajectory. Lastly, the most commonly used technique to compensate the modeling error in the SPC is to use the corrective control for the slow system, that is, by adding corrective terms to the rigid control. However, the corrective terms require derivatives of the rigid control. Because the rigid control is typically designed as a PD control on the link side, the corrective control must contain higher order states than the velocity, i.e., link acceleration and jerk. Therefore, with the corrective control, SPC effectively becomes a full-state controller on the link side, similar to feedback linearization, defeating the original purpose of avoiding dependence on estimating high order states.

2.3 Summary

In this section, the research on the control approach for industrial robots has been reviewed. The nonlinear control methods based on rigid model cannot obtain a good tracking performance due to the lightly damped poles coming from the joint elasticity. Therefore, one must use more accurate dynamic model that takes account the joint elasticity. The general flexible joint dynamic model is difficult to design control laws as it cannot be feedback linearizable by static state feedback (Thümmel *et al*., 2001). The simplified flexible joint model is a good candidate for control design because it can be linearizable by static state feedback while capturing the effect of the joint elasticity. The full model-based control such as feedback linearization and other methods mentioned in Brogliato *et al*. (1995) are difficult to implement because it requires accurate estimation of high order states. The reduced model-based control can avoid using the high order states for feedback control but it is unclear how the desired closed-loop performance should be achieved for arbitrary trajectories.

Chapter 3: Motion Control System

In this Chapter, the motion control system of a typical industrial robot, namely, *the dynamic model of an articulated robot with elastic joints with a low-cost generic controller* is overviewed. For the dynamic model, only a brief introduction is provided as the relevant topic is well addressed in the literature. Since the control structure of a commercial robot controller is not easily available in the literature, the control structure is discussed in more detail with the Hi5a controller as an example. With respect to motion control, the Hi5a controller follows many aspects of the traditional robot controller such as hierarchical structure, decentralized PD control, and a low-cost design. Based on the analysis of the control structure of the Hi5a controller, some fundamental limitations of a traditional robot controller are discussed in terms of the relationship between the control structure and the achievable motion performance. Then, the concept of PITOM is introduced to specify the highest level of motion performance that is pursued by this thesis. Finally, a general approach is proposed such that the planning and control modules together achieve the PITOM under constraints given by a generic robot controller.

3.1 Dynamic Model

It is well-known that a rigid model cannot represent a typical industrial robot because the joint elasicity cannot be ignored in the modeling and control design (Good *et al*., 1985). Therefore, the flexible joint dynamic model (Spong, 1987) is used throughout this thesis and it is denoted by

$$\mathbf{M}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) + \mathbf{G}(\mathbf{q}_r) + \mathbf{K}(\mathbf{q}_r - \mathbf{r}^{-1}\mathbf{q}_m) = \mathbf{0} \in \Re^n, \tag{3.1a}$$

$$\mathbf{M}_m\ddot{\mathbf{q}}_m + \mathbf{f}_m(\dot{\mathbf{q}}_m) + \mathbf{r}^{-1}\mathbf{K}(\mathbf{r}^{-1}\mathbf{q}_m - \mathbf{q}_r) = \mathbf{u} \in \Re^n , \tag{3.1b}$$

where the link position and the motor position are denoted by $\mathbf{q}_r \in \Re^n$ and $\mathbf{q}_m \in \Re^n$, respectively. $\mathbf{M}_r(\mathbf{q}_r) \in \Re^{n\times n}$ is the position-dependent inertia matrix. $\mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) \in \Re^n$ is the Coriolis-centrifugal effect. $\mathbf{G}(\mathbf{q}_r) \in \Re^n$ represents gravity. $\mathbf{K} \in \Re^{n\times n}$, $\mathbf{r} \in \Re^{n\times n}$, and $\mathbf{M}_m \in \Re^{n\times n}$ are constant diagonal matrices, representing the joint stiffness, reduction ratio, and motor inertia, respectively. $\mathbf{f}_m \in \Re^n$ is the motor friction. $\mathbf{u} \in \Re^n$ is the input or the commanded motor torque. In Figure 3.1, the dynamic model (3.1) for a 1DOF case is shown.

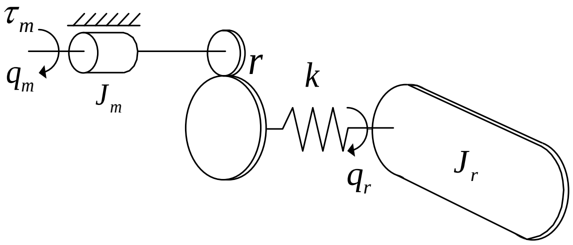


Figure 3.1. Flexible joint dynamic model for a 1DOF robot (J_r : link inertia, J_m : motor inertia, k : joint stiffness).

This set of two 2nd order nonlinear ordinary differential equations (ODEs) can be combined by first differentiating (3.1a) with respect to time twice then inserting the result into (3.1b), resulting in a single 4th order nonlinear ODE such that

$$\mathbf{u} = \boldsymbol{\tau}_{rigid} + \boldsymbol{\tau}_{flex} \in \Re^n , \tag{3.2}$$

where $\boldsymbol{\tau}_{rigid} \triangleq \mathbf{r}^{-1}\left[\left(\mathbf{M}_r(\mathbf{q}_r) + \mathbf{r}^2\mathbf{M}_m\right)\ddot{\mathbf{q}}_r + \mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) + \mathbf{G}(\mathbf{q}_r)\right] + \mathbf{f}_m(\dot{\mathbf{q}}_m)$, (3.3)

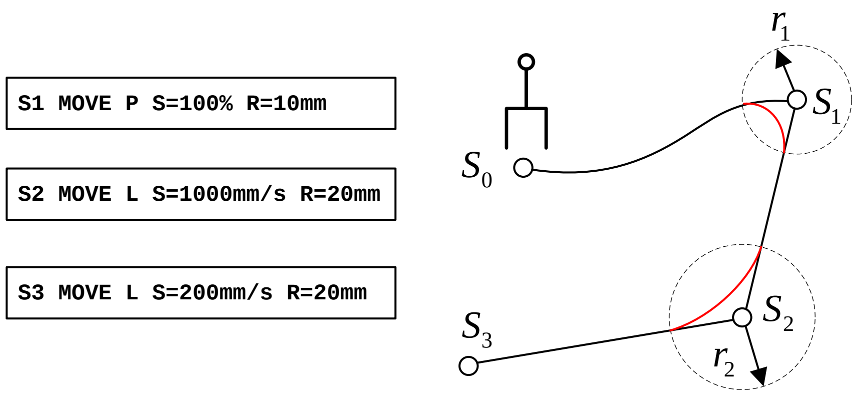
$$\boldsymbol{\tau}_{flex} \triangleq \mathbf{M}_m \mathbf{r} \mathbf{K}^{-1} \ddot{\boldsymbol{\tau}}_r , \tag{3.4}$$

$$\boldsymbol{\tau}_r \triangleq \mathbf{M}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{C}_r(\mathbf{q}_r,\dot{\mathbf{q}}_r) + \mathbf{G}(\mathbf{q}_r)\,. \tag{3.5}$$

The model described by (3.1) is widely used for controller design as it reflects several important aspects of robot dynamics such as inertial coupling, nonlinear velocity-dependent forces, and joint elasticity, while still being linearizable by static state feedback. It should be noted that the model (3.1) assumes that the inertial coupling between the motor and the link can be ignored for a high gear ratio. If the gear ratio is not relatively large, then a different model should be used. In this case, however, the model can be linearizable by dynamic state feedback (De Luca, 1998). Since the thesis concerns a typical industrial robot, where the gear ratio is relatively high, typically ranging from 50 to 200, the model (3.1) is assumed to be sufficiently accurate.

3.2 Control Architecture

The architecture of the control system of an industrial robot is typically divided into three subgroups; namely, a programming module (the highest control level), a trajectory planning module, and a servoing module (the lowest control level). The programming module is accessed by a user interface, typically a teach pendant (see Figure 1.1). Through this module, the user can teach the path and speed of the robot by specifying a set of waypoints, desired speed (often as a percentage of the robot's maximum speed or speed of the tool center point (TCP)), and the type of paths between the waypoints such as straight-lines, circular paths, or splines (see Figure 3.2). The user can also specify how closely the robot bypasses a waypoint as a radius, r, an approach widely used in industrial robots to optimize the cycle time. The trajectory planning module, or simply trajectory planner, interprets the robot program and generates both the geometric path *and* the time-history of motion commands for each joint such that the resulting robot motion produces the programmed motion. The planned trajectory is then sent to the servoing module or *control loop* by generating the trajectory in real-time. The control loop must obtain a good tracking performance for the desired trajectory to finally realize the motion intended by the user.

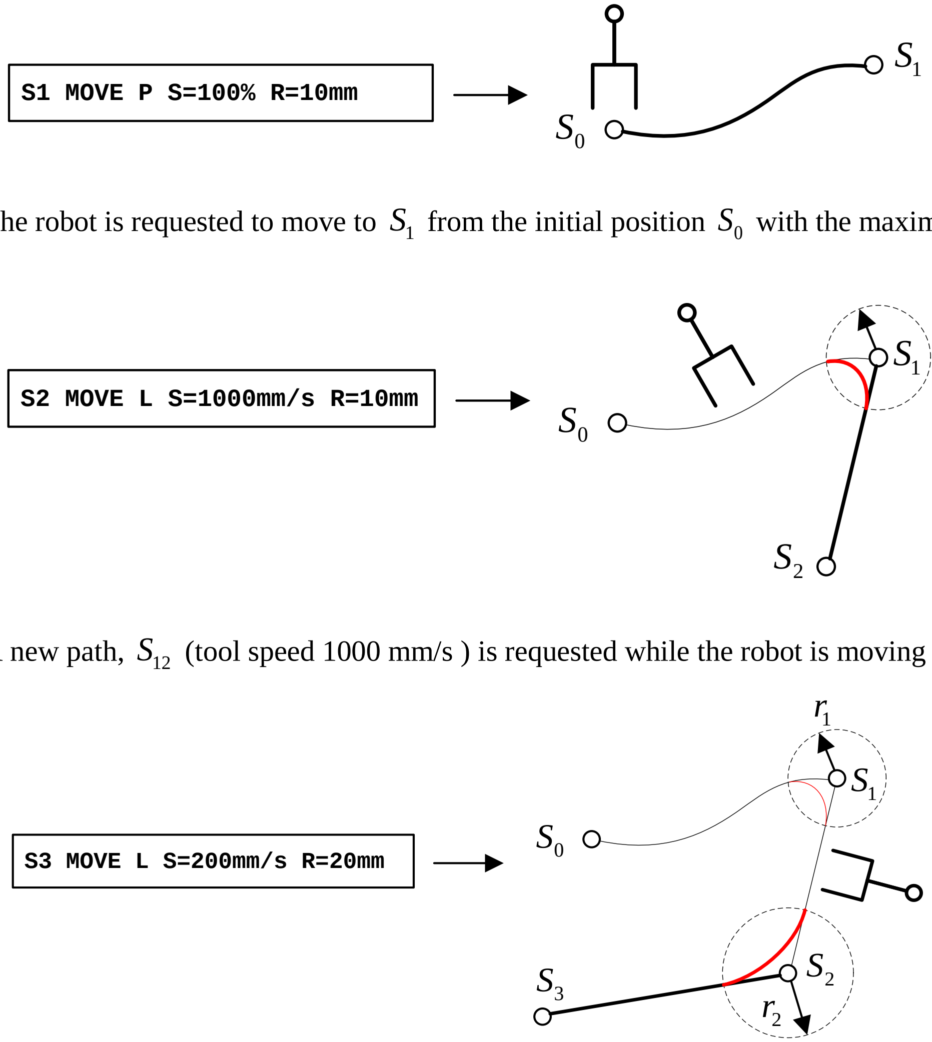


```
S1 MOVE P S=100% R=10mm
S2 MOVE L S=1000mm/s R=20mm
S3 MOVE L S=200mm/s R=20mm
```

Figure 3.2. An exemplar robot program and the desired tool path (P: straight line in joint space, L: straight line in Cartesian space).

3.2.1 The Execution of the Robot Program

The robot program is executed online sequentially, which means that the next instruction (motion command) can be executed only after the current instruction is complete. Sequential execution is required because the robot's motion can often be modified by sensor inputs. In Figure 3.3, the robot motion with sequential execution is described. In the figure, it is assumed that the robot is initially positioned at S_0. When the first instruction is entered, the trajectory planner must immediately generate the motion according to the given instruction. Then, the robot starts to move towards S_1 with the maximum speed (Figure 3.3(a)). While the robot is moving along the path S_{01} (the path between S_0 and S_1), the next instruction is entered (Figure 3.3(b)). According to this instruction, the robot should not stop at S_1. Instead, it should bypass S_1 at a given radius of 10 mm and then move at 1000 mm/s along a straight-line to S_2. This process continues whenever a new instruction is entered (Figure 3.3(c)). In this thesis, it is assumed that the programing module is fully specified and the main focus of the study is the optimal design of the planning module and the servoing module.



```
S1 MOVE P S=100% R=10mm
S2 MOVE L S=1000mm/s R=10mm
S3 MOVE L S=200mm/s R=20mm
```

(a) The robot is requested to move to S_1 from the initial position S_0 with the maximum speed.

(b) A new path, S_{12} (tool speed 1000 mm/s) is requested while the robot is moving along S_{01}.

(c) A new path, S_{23} (tool speed 200 mm/s) is requested while the robot is moving along S_{12}.

Figure 3.3. Online execution of a sequential robot program.

Now consider the requirements of the planning module such that the robot motion specified by the programming module is successfully executed. First, the final velocity of the trajectory for each path segment must be zero in case no further path segments appear. Second, the planner should be able to calculate the trajectories with non-zero boundary conditions. For instance, the transition path segments shown in red in Figure 3.3 must be planned while the robot is moving along the path towards the transition, after the next waypoint is specified. The initial and final velocities and accelerations of the transition path segments are not zero in general. Third, a unique transition path must be planned regardless of the operating speeds. Path repeatability independent of operating speed is required to meet requirements for collision checking and safety regardless of the size of the waypoint radius, which itself may not guarantee collision free motion. Finally, the trajectory planner must consider dynamics to permit the robot to quickly change accelerations while moving along multiple paths at high speeds.

Therefore, the requirements of the planning module can be summarized as: (i) the trajectory is calculated with arbitrary boundary conditions; (ii) the path is uniquely generated given waypoints and bypass radii; and (iii) the dynamics must be considered.

In the meantime, the dynamic model (3.1) is a 4th order nonlinear ODE. If this model is used for trajectory planning, then the trajectory must also be of the 4th order, meaning that the trajectory planner should generate continuous position, velocity, acceleration, and jerk. Unfortunately, a closed-form solution for the 3rd or higher order TVP-like trajectory with arbitrary boundary conditions is not available in the literature. Therefore, some of the boundary conditions must be forced to be zero in order to find a closed-form solution to ensure bounded computation time. However, such a resolution may result in a significant increase in cycle-time.

A numerical approach is also not desirable because the trajectory should then be computed iteratively with the dynamic model, generating a large computational load.

3.2.2 The Servoing Module

Assuming that the trajectory is successfully planned, then the user's intended motion can be realized only if the servoing module tracks the reference trajectory with small tracking error. In Figure 3.4, the block diagram for the motion control architecture of a Hi5a controller is shown. The reference trajectory for each joint is computed by inverse kinematics and then its unit is converted so that it can be used as the reference on the motor side, typically by static conversion, i.e., multiplied by the gear ratio. The motor reference is then controlled *independently* at each joint on the motor side by measuring the position and velocity obtained from its own digital encoder, namely a *decentralized control*. The servoing module is usually implemented in a separate hardware system from the mainboard where the programming module and planning module are implemented. The communication between the servoing module and the planning module is generally carried out through Ethernet or shared RAM. Since the servoing module controls both the position and the motor torque, the sampling frequency is generally higher than the mainboard. For Hi5a controller in Figure 1.1, the sampling frequencies of the planning and servoing module are 500 Hz and 2.5 KHz, respectively. Since the servoing module needs a high sampling rate, the computing power of each sample must be kept at a minimal cost. Thus, a decentralized PD controller is widely adopted for tracking control so that the computation required at each sampling period is limited to only several additions and multiplications.

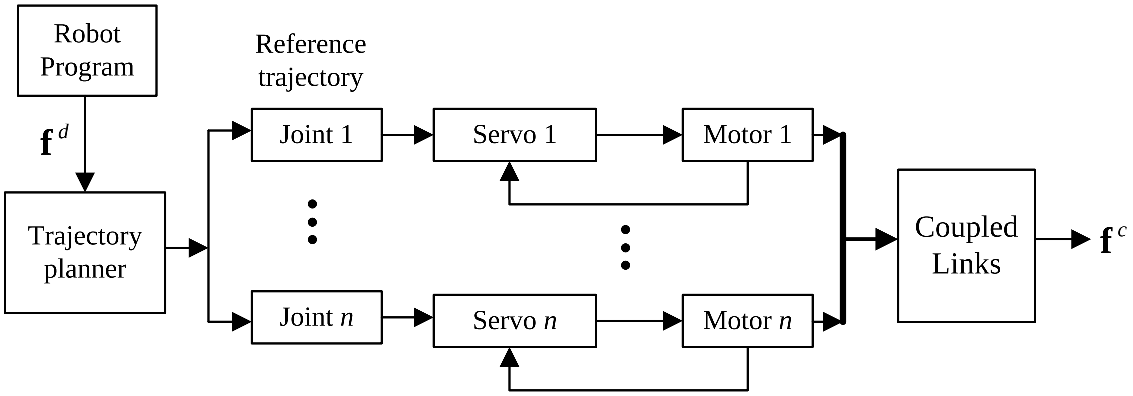


Figure. 3.4 Control structure with decentralized control ($\mathbf{f}^d$: user-specified path, $\mathbf{f}^c$: controlled path).

In Figure 3.5, PD control implemented in the Hi5a controller (Figure 1.1) is shown. It is a cascaded PD controller that is widely used for industrial robots (Miyazaki *et al*., 2003; Moberg, 2010). The cascaded PD control is more advantageous than a single loop PD control in terms of tuning control gains and robustness against input disturbances. The cascaded PD control shown in Figure 3.5 is implemented on the motor side. The feedforward control based on the rigid model is typically employed for gravity compensation and better tracking performance. Such a control scheme, *the PD control plus feedforward control based on a rigid model*, is perhaps the most common control method for industrial robots (Santibañez and Kelly, 2001; Craig, 2005). Note that the steady-state error is typically compensated by adding an integral control in the inner-loop. In this thesis, the integral control is not discussed for simplicity.

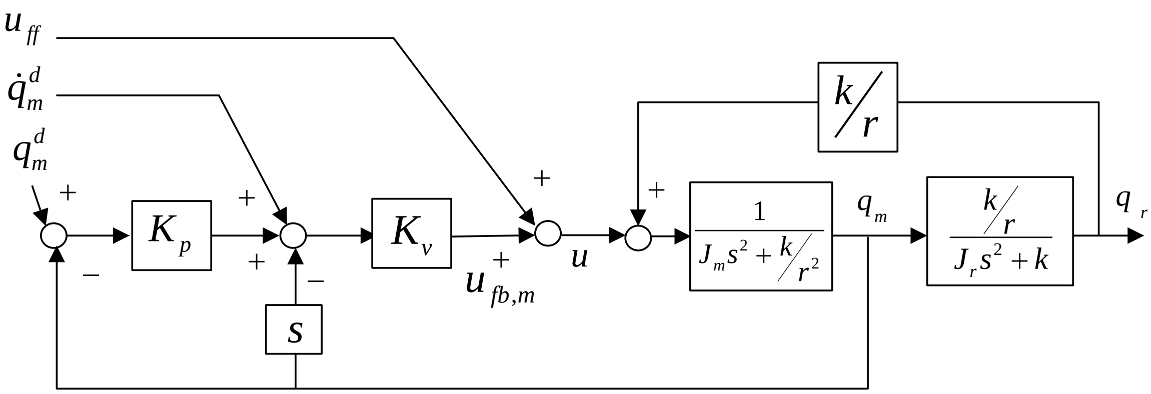


Figure. 3.5. Block diagram of the cascaded PD control for Hi5a controller.

Let us discuss the cascaded PD control mentioned above in more detail. The control law can be expressed as

$$\mathbf{u} = \mathbf{u}_{ff,rigid} + \mathbf{u}_{fb,m} \in \Re^n, \tag{3.6}$$

where $\mathbf{u}_{ff,rigid} = \mathbf{r}^{-1}\left[\left(\mathbf{r}^2\mathbf{M}_m + \mathbf{M}_r^d\right)\ddot{\mathbf{q}}_r^d + \mathbf{C}_r^d + \mathbf{G}^d\right] + \mathbf{f}_m(\mathbf{r}\dot{\mathbf{q}}_r^d) \in \Re^n$,

$\mathbf{M}_r^d \triangleq \mathbf{M}_r(\mathbf{q}_r^d)$, $\mathbf{C}_r^d \triangleq \mathbf{C}_r(\mathbf{q}_r^d, \dot{\mathbf{q}}_r^d)$, $\mathbf{G}^d \triangleq \mathbf{G}(q_r^d)$, and $\mathbf{q}_r^d$ is the desired link position. $\mathbf{u}_{fb,m}$ is the input generated by PD control with motor feedback. One can note that it is assumed that the elastic model is not available in the above feedforward control, i.e., $\mathbf{K} \to \infty$. Applying (3.6) to (3.2), we obtain

$$\mathbf{r}\mathbf{M}_m\mathbf{K}^{-1}\mathbf{M}_r\mathbf{q}_r^{(4)} + \left(\mathbf{r}\mathbf{M}_m + \mathbf{r}^{-1}\mathbf{M}_r\right)\ddot{\mathbf{e}}_r = \mathbf{u}_{fb,m}, \tag{3.7}$$

where $\mathbf{e}_r \triangleq \mathbf{q}_r^d - \mathbf{q}_r$ is the tracking error on the link side. In (3.7), it is assumed that the dynamic model is quite accurate, namely,

$$\mathbf{M}_r^d \approx \mathbf{M}_r,\ \ \mathbf{C}_r^d \approx \mathbf{C}_r, \text{ and } \mathbf{G}^d \approx \mathbf{G} \tag{3.8}$$

Note that the derivatives of $\mathbf{M}_r$, $\mathbf{C}_r$, and $\mathbf{G}$ are ignored in (3.7) as they are not as significant as other terms. Now let us assume the feedback control, $\mathbf{u}_{fb,m}$, to be a cascaded PD control as shown in Figure 3.5, namely,

$$\mathbf{u}_{fb,m} = \bar{\mathbf{K}}_p \mathbf{e}_m + \mathbf{K}_v \dot{\mathbf{e}}_m \in \Re^n , \tag{3.9}$$

where $\mathbf{e}_m \triangleq \mathbf{q}_m^d - \mathbf{q}_m$ is the tracking error on the motor side. $\mathbf{K}_p$ and $\mathbf{K}_v$ are the gain (diagonal) of the outer-loop and the inner-loop, respectively, and $\bar{\mathbf{K}}_p \triangleq \mathbf{K}_p \mathbf{K}_v$.

In general, the cascaded PD control is tuned in the following way. First, the velocity gain, K_v is chosen with as large a value as possible provided it satisfies the noise level and the torque constraints. To see this rationale, consider the transfer function of the inner loop with respect to motor input and motor velocity.

$$T_v(s) = \frac{K_V \left(J_r s^2 + k \right)}{J_m \, J_r \; s^3 + J_r K_V \, s^2 + k \left(J_m + \dfrac{J_r}{r^2} \right) s + K_V k} \tag{3.10}$$

The Routh-Hurwitz criterion reveals that (3.10) is stable for $K_v > 0$. Furthermore, it satisfies

$$\left| T_v(\omega) \right| \le 1 \;\; \text{for } \forall \omega . \tag{3.11}$$

For a typical industrial robot, the velocity gain is tuned such that the bandwidth of the inner-loop is between 80 Hz and 120 Hz, which is much higher than the lowest natural frequency of industrial robot, typically 10 Hz or less (Craig, 2005; Moberg, 2010). The closed-loop system on the motor side is obtained as

$$\dot{\mathbf{e}}_m + \mathbf{K}_p \mathbf{e}_m = \mathbf{K}_v^{-1} \left[\mathbf{r} \mathbf{M}_m \mathbf{K}^{-1} \mathbf{M}_r \mathbf{q}_r^{(4)} + \left(\mathbf{r} \mathbf{M}_m + \mathbf{r}^{-1} \mathbf{M}_r \right) \ddot{\mathbf{e}}_r \right] . \tag{3.12}$$

Assuming that the velocity gain is high and the joint stiffness is sufficiently large, then the system (3.12) is stable for any positive $\mathbf{K}_p$ and the tracking error on the motor side is decreased as the position gain increases. And the effect of the link tracking error is suppressed as the velocity gain increases. That is, good tracking performance and stability can be achieved simultaneously by high gain feedback on the motor side. However, the error dynamics on the link side is

$$\dot{\mathbf{e}}_r + \mathbf{K}_p \mathbf{e}_r \approx \boldsymbol{\psi} \ , \tag{3.13}$$

where $\boldsymbol{\psi} \triangleq \mathbf{K}^{-1}\left(\mathbf{K}_v^{-1}\mathbf{M}_m\mathbf{M}_r\mathbf{q}_r^{(4)} + \mathbf{M}_r\mathbf{q}_r^{(3)} + \mathbf{K}_p\mathbf{M}_r\ddot{\mathbf{q}}_r\right)$ and it is assumed that $\mathbf{q}_m^d = \mathbf{r}\mathbf{q}_r^d$.

Eq. (3.13) clearly explains the difficulty in choosing the position gain. Notice that a large $\mathbf{K}_p$ also implies a large perturbation or the effect of motion on the link side. Meanwhile, a small $\mathbf{K}_p$ will result in a slow response. Therefore, the rule-of-thumb is to choose $\mathbf{K}_p$ such that the bandwidth of the outer-loop is slightly less than the natural frequency of the link side. The result in (3.13) provides a further insight into how to design the mechanical components and the reference trajectory with the control method (3.9). That is, the tracking performance on the link side can be improved as the joint stiffness becomes larger and the inertia becomes smaller. If such modification is not possible, then the only possible choice is to reduce the effect of the derivatives of $\mathbf{q}_r$ by smoothing the reference trajectory.

3.3 Fundamental Limitations

According to the discussion in Sections 3.1 and 3.2, the motion control system of a typical industrial robot can be summarized as follows:

- **The dynamic model**: The flexible joint dynamic model is a 4th order nonlinear ODE. The model can be approximated by a 2nd order nonlinear ODE if the joint elasticity is ignored.
- **Control architecture:** To allow online execution, the trajectory must be planned online with arbitrary boundary conditions. For a closed-form solution, the order of the trajectory cannot be larger than two. The decentralized PD controller can obtain good tracking performance on the motor side by high gain feedback. Unless the trajectory is sufficiently smooth, the tracking performance on the link side is very poor.

Therefore, the design of the trajectory in a low-cost robot controller faces several contradictory requirements. At the planning module level, it is difficult to plan a time-optimal smooth trajectory due to complexity and a high computational load. In addition, a 3rd or higher TVP-like trajectory cannot be solved analytically for arbitrary boundary conditions. This means that it is almost impossible to plan a time-optimal yet smooth (order of three or higher) trajectory in a low-cost controller. However, the smoothness of the trajectory is imperative for the servoing module for good tracking performance on the link side with the decentralized PD control.

In industry, the time-optimality is a crucial requirement for a commercialized controller as the demand for short cycle times in common applications such as spot welding and pick-and-place material handling is very high. Therefore, the time-optimal TVP trajectory based on a rigid model is widely used in industry as they can be easily implemented in a low-cost controller (Antonelli *et al*. 2009; Kim *et al*. 2010). However, the TVP trajectory will generate poor tracking performance with the decentralized PD control as the infinite jerk of the TVP trajectory will excite the unmodeled flexibility inside the control bandwidth. If the servoing module cannot be modified, then the only feasible remedy for such a TVP trajectory to achieve good tracking performance with PD control is to apply a low-pass filter to the planned trajectory. This explains why many industrial robots use a low-pass filter between the planning module and the servoing module. However, as it will be discussed in more detail in Chapter 4, adding a low-pass filter brings another serious problem associated with path accuracy, named, *path-distortion* in this thesis.

3.4 Motion Control Strategy

Based on the above discussion, a new strategy for motion control is now presented. The new strategy pursues the *Path-Invariant and Time-Optimal Motion*. The PITOM is a general guideline of motion control for industrial robots such that it achieves both fast motions and small tracking errors simultaneously. The conceptual diagram is shown in Figure 3.6. The PITOM states that the user’s intended path should be travelled with the maximum capacity of the robot and it should be exactly followed by the controlled path. In other words, the trajectory must be time-optimal at all times and the resulting geometric path should be unique regardless of different operating speeds. In this way, the time-optimality is guaranteed whenever the robot is requested to move at the maximum operating speed and the path deviation does not occur during trajectory planning. Since the path-invariance and time-optimality is assured in the planning module, the only requirement of the servoing module is a good tracking performance for time-optimal trajectories. However, for typical industrial robots the servoing module must achieve this with motor measurements only.

There are two major differences between the PITOM compared with the motion control approach commonly addressed in the literature. First of all, the PITOM specifies the path accuracy in the planning module as well as the servoing module whereas the servoing module is assumed to take the most responsibility for path accuracy in the literature. However, as mentioned in Chapter 2, a large deviation from the desired path can occur during trajectory planning; high path accuracy cannot be achieved by the servoing module alone. Second, in the proposed scheme high tracking performance must be obtained with motor measurement only. Many control methods assume that measurements are available both on the motor side and on the link side. Without these link side measurements, it is difficult to obtain a desired performance specified by the PITOM. A novel control method that does not rely on accurate estimation of the link states is needed in the proposed scheme.

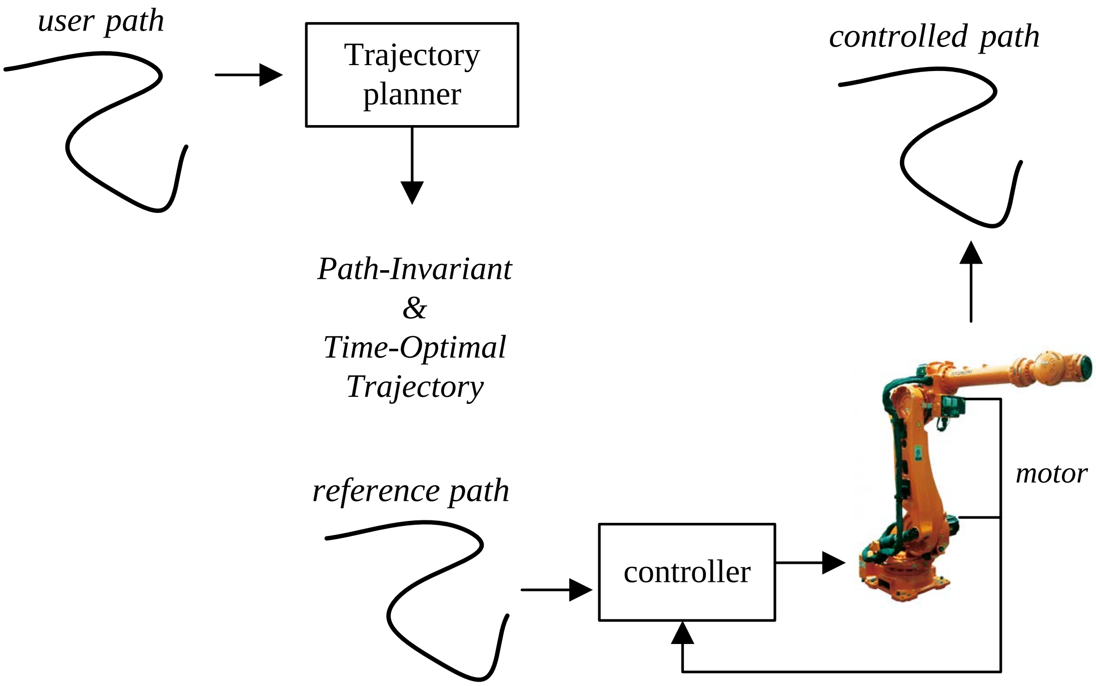
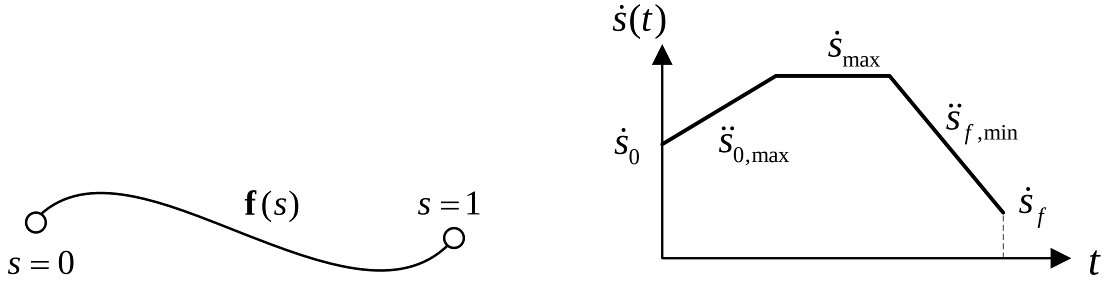


Figure 3.6. The schematic diagram of the PITOM.

3.4.1 Planning Module

The trajectory that is chosen in this thesis is the TVP trajectory (2nd order) in order to obtain a closed-form solution for arbitrary boundary conditions. The well-known rigid model, which is second order, suffices for computing the maximum speed and acceleration that are the kinematic parameters necessary for planning a TVP trajectory. As noted, the trajectory must also satisfy the condition of the path-invariance according to the PITOM. This means that the reference path generated by different planned trajectories must be unique. It is well-known that such conditions can be met by planning the trajectory, namely, the time law along the path. As shown in Figure 3.7(a), the path $\mathbf{f}(s)$ is independent of the trajectory, $s(t)$. Therefore, the necessary conditions of the path should be; (i) the path should be expressed in terms of a single path parameter, s and; (ii) the continuity of the path must be at least C^2 for feasible acceleration (2nd order trajectory). An example of the proposed trajectory is shown in Figure 3.7(b). The maximum kinematic values, $\dot{s}_{\max}$, $\ddot{s}_{0,\max}$, and $\ddot{s}_{f,\min}$ can be obtained by utilizing a rigid model.



(a) Parameterized path

(b) TVP along a parameterized path

Figure 3.7 Trajectory planning along a parameterized path.

3.4.2 Servoing Module

Since the proposed trajectory is a time-optimal TVP trajectory, the conventional PD controller cannot be used due to the lack of smoothness of the trajectory. Note that the proposed TVP trajectory is generated by neglecting the joint elasticity, i.e., using a second order model. Therefore, the proposed trajectory will excite the flexible modes of the robot when it is directly used as the reference input. It has been shown that a feedforward control based on the flexible joint dynamic model can be used with PD control on the motor side (De Luca, 2000). The method cancels the flexible modes by modifying the motor reference, i.e., a dynamic conversion of the motor references as opposed to the static conversion. However, since this method is purely a feedforward approach, the performance depends on the accuracy of the model. In addition, the feedback system is still realized only on the motor side (partial-state feedback). Therefore, the disturbance generated on the link side cannot be suppressed effectively. To achieve a good tracking performance on the link side, it is necessary to estimate the state on the link side. However, according to Figure 3.6, the measurement is available only on the motor side for control. This requires that the servoing module include a state observer to estimate the states on the link side. As the performance on the link side is heavily dependent upon the accuracy of the observer, the full model (3.1) must be used. Thus, a nonlinear multivariable controller with a state observer should be used in the servoing module. This means that the servoing module should be redesigned to allow a centralized control as shown in Figure 3.8. In the centralized control, the dynamic model (3.1) should be implemented and be computed at the servoing rate. Therefore, a major modification of the servoing module is inevitable if the existing servoing module is based on the decentralized scheme as shown in Figure 3.4.

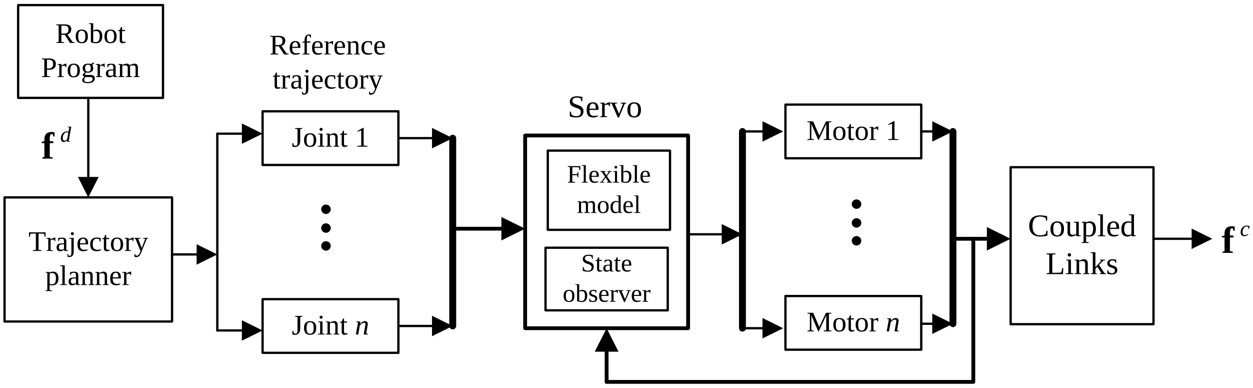


Figure 3.8. Control structure with centralized control.

3.5 Summary

In this chapter, the motion control system of a typical industrial robot has been overviewed. The analysis of the dynamic model and control architecture has revealed that obtaining high motion performance with a low-cost controller is a challenging task due to several constraints that must be met for practical implementations such as online programming, limited computational power, and available measurements for feedback control. Meanwhile, the PITOM has been proposed to specify how planning and servoing modules should be designed for high motion performance while satisfying the strict industrial constraints. Finally, a 2nd order time-optimal trajectory with model-based control (centralized control) has been proposed as a general motion control strategy that can satisfy the PITOM. In the next chapter, a comparative study is carried out to validate the proposed motion control strategy.

Chapter 4: Comparison of Motion Control Methods[1]

This chapter evaluates several motion control methods that are widely used for flexible mechanical systems such as the model (3.1). In Chapter 3, a time-optimal TVP with model-based control is suggested for high motion performance. Since the time-optimal TVP has high bandwidth, it is not certain whether good tracking performance can be solely achieved by control loop (the control approach). Further, without a head to head comparision, one could argue that better performance can be achieved by adopting a smooth trajectory (the planning approach). Therefore, in this chapter, the proposed scheme in Chapter 3, namely, *a low-order trajectory with model-based control*, is validated by comparing it against the planning approach.

In Kim and Croft (2015), an initial study is performed by simulation to show that the control approach may obtain better performance than the planning approach. Herein, the work of Kim and Croft (2015) is further enhanced with a more detailed analysis and experimental results. For comparison, this chapter first focuses on improving the planning approach such that the performance is optimized. To do so, a practical method to find the optimal jerk for S-curve profiling is presented such that the residual vibrations are minimized with PD control. It turns out that finding the optimal jerk does not require the complete dynamic model including the controller but the knowledge of the link inertia and joint stiffness will suffice. Therefore, one no longer needs to find some "optimal" jerk by trial-and-error, as previously described in the literature (Macfarlane and Croft, 2003; Tsay and Lin, 2005; Li *et al*., 2007; Nguyen *et al*., 2008;

[1] A version of this chapter has been submitted for publication in *IEEE/ASME Transactions on Mechatronics* (Kim and Croft. A comparative study on motion control methods for industrial robots with elastic joints).

Kröger and Wahl, 2010; Ezair *et al.*, 2014). The optimal S-curve with the TVP in the frequency domain is also compared to clearly illustrate the reason why the S-curve generates smaller vibrations than the TVP under the same maximum acceleration constraints. In Meckl (1998), a method to optimally select the ratio between the ramp-up time and the time to peak velocity is proposed. The difference between the proposed method herein and the method in Meckl (1998) is that the former is more general; it can be applied to multiple joints, and measurement of the natural frequency is not required.

The analysis of the input shaping technique (IST) reveals that a successful application of the traditional IST is extremely difficult due to the complexity of dynamics such as the configuration-dependent multiple vibration modes and unknown high frequency modes (Moberg, 2010). Therefore, a simple scheme that combines the zero-vibration (ZV) shaper and the simple moving average (SMA) filter is proposed. Such a combination results in a very robust filter that can suppress not only a wide range of frequency but also unknown high frequency modes. Furthermore, an important phenomenon when using the IST, namely, *path-distortion*, is discussed. It is shown that path-distortion is an inherent problem of any filtering technique that utilizes the previous reference values.

The proposed planning approaches are then compared with the control approach. The feedforward control (De Luca, 2000) and the full-state feedback control (Miyazaki *et al.*, 2003; Tungpataratanawong *et al.*, 2005) are selected for comparison as these controllers are widely used in industry and can be implemented on the available test platform without significant modification of the control structure. It is shown that the feedforward control can be regarded as a combination of a delay-free dynamic input shaper and input for the desired trajectory. The dynamic input shaper embedded in the feedforward control fundamentally differs from the traditional IST in that the former has no phase lag, thereby generating zero path-distortion while requiring the same knowledge of the model parameters, i.e., the link inertia and joint stiffness. The full-state control is the combination of the feedforward control and active damping on the link side. Therefore, it can further reduce the residual vibrations, which comes at the expense of more complicated control structure.

This chapter is organized as follows. First, the optimal S-curve profiling and a robust design of the IST is proposed. Then, two nonlinear controllers that represent the control approach are introduced with emphasis on the dynamic input shaper in the feedforward control. Finally, a real-world experiment is carried out to validate our theoretical results.

4.1 Planning Approach

4.1.1 S-curve Profiling

In this section, a method is presented to find the optimal jerk for S-curve profiling such that it generates no residual vibrations at the end of motion under the cascaded PD controller (Figure 3.5) with high gain feedback in the inner loop.

Eq. (3.12) suggests that the following condition may hold for a large $\mathbf{K}_v$ and a large stiffness value, which may be true for a typical industrial robot.

$$\mathbf{q}_m^d \approx \mathbf{q}_m \ . \tag{4.1}$$

Consider the static conversion for the motor reference, $\mathbf{q}_m^d = \mathbf{r}\mathbf{q}_r^d$. By inserting (4.1) into (3.1a), we obtain

$$\mathbf{M}_r \ddot{\mathbf{q}}_r + \mathbf{K}\,\mathbf{q}_r = \mathbf{K}\,\mathbf{q}_r^d \,, \tag{4.2}$$

where it is assumed that the nonlinear term, gravity, and friction is compensated by the feedforward control, $\mathbf{u}_{ff,rigid}$.

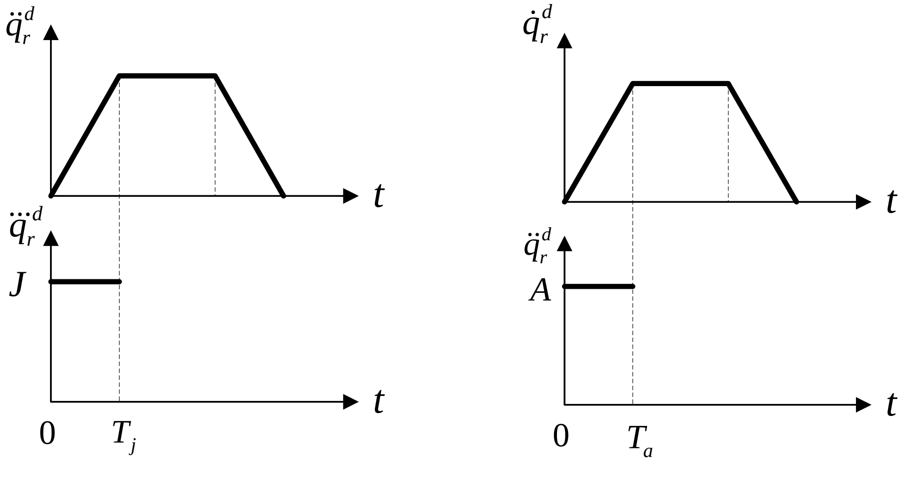
Let us consider the steady state, when it is possible to linearize the robot dynamics about a stationary point. For instance, suppose an instance when the robot starts to move from zero initial velocity or at the end of the reference trajectory, $\mathbf{q}_r^d(t)$. At steady state, Eq. (4.2) can be expressed in the frequency domain because the inertia matrix can be assumed to be constant, or

$$\mathbf{q}_r(j\omega) = \left(\mathbf{K} - \omega^2\mathbf{M}_r\right)^{-1}\mathbf{K}\,\mathbf{q}_r^d(j\omega)\,. \tag{4.3}$$

Now assume that the reference trajectory is designed as the S-curve (Figure 4.1(a)). The S-curve trajectory on the interval $[0 \quad T_j]$ can be represented as

$$\ddot{q}_r^d(t) = J\left[\,\gamma(t) - \gamma(t - T_j)\right], \tag{4.4}$$

where $\gamma(t)$ is the unit step function.



(a) S-curve (b) TVP

Figure 4.1. Comparison between S-curve and TVP.

Apply the Fourier Transform to (4.4) to obtain, similar to Biagiotti and Melchiorri (2008),

$$q_r^d(j\omega) = J\omega^{-4}\left(1-e^{-jT_j\omega}\right). \tag{4.5}$$

Therefore, the S-curve can be interpreted as a low-pass filter with the fundamental period, T_j^{-1} $[Hz]$ in the frequency domain. At NT_j^{-1}, where N is an integer, the amplitude of (4.5) becomes zero. Let us assume that the jerk period for all joints is the same as T_j for a coordinate motion, then

$$\mathbf{q}_r^d(j\omega)_{S-curve} = \omega^{-4}\left(1-e^{-jT_j\omega}\right)\mathbf{J}, \tag{4.6}$$

where $\mathbf{J} \triangleq \begin{bmatrix} J_1 & \cdots & J_n \end{bmatrix}^T$. Hence, the frequency response of the link position to the S-curve trajectory (4.6) at steady state under a high gain feedback control on the motor side is obtained as

$$\mathbf{q}_r(j\omega)_{S-curve} = \omega^{-4}\left(1-e^{-jT_j\omega}\right)\left(\mathbf{K}-\omega^2\mathbf{M}_r\right)^{-1}\mathbf{K}\,\mathbf{J}. \tag{4.7}$$

From (4.7), some useful idea can be obtained regarding how to design the jerk. The term $\mathbf{K}-\omega^2\mathbf{M}_r$ is the denominator of (4.7) for $\omega > 0$. Clearly, the poles are the natural frequencies of the link dynamics of (4.2). The natural frequencies can be calculated easily by solving the eigenvalue problem,

$$\mathbf{K}^{-1}\mathbf{M}_r\mathbf{x} = \omega^{-2}\mathbf{x}. \tag{4.8}$$

Then, the optimal value for T_j should be

$$T_j = f_{n,1}{}^{-1}, \tag{4.9}$$

where f_1 $[Hz]$ is the root square of the largest eigenvalue of (4.8), or the lowest natural frequency. By the choice of (4.9), the first mode of the system (4.2) cannot be excited. In addition, the higher modes can be suppressed quickly because the magnitude of $\mathbf{q}_r(j\omega)$ is attenuated by the term, ω^{-4}.

It is tempting to apply the above procedure to the TVP trajectory (Figure 4.1(b)) to see why the S-curve generates smaller vibrations than the TVP. Following the same procedure, the frequency response of the TVP trajectory is obtained as

$$\mathbf{q}_r^d(j\omega)_{TVP} = \omega^{-3}\left(1-e^{-jT_a\omega}\right)\mathbf{A}, \tag{4.10}$$

where $\mathbf{A} \triangleq \begin{bmatrix} A_1 & \cdots & A_n \end{bmatrix}^T$. Therefore, a similar conclusion can be obtained for the TVP trajectory as well. That is, the first mode of the system cannot be excited if the acceleration period is tuned at the first mode of the link dynamics

$$T_a = f_{n,1}^{-1}. \tag{4.11}$$

However, the result (4.11) cannot be applied to a typical industrial robot, where the maximum acceleration is constrained due to a time-optimal design (Antonelli *et al.*, 2007; Kim *et al.*, 2010). Nevertheless, it provides a clear reason why the S-curve generates smaller vibrations than the TVP. More specifically, consider the magnitudes of the two trajectories in the frequency domain.

$$\left|q_r^d(j\omega)_{S-curve}\right| = \sqrt{2}J\omega^{-4}\sqrt{1-\cos(T_j\omega)} \tag{4.12a}$$

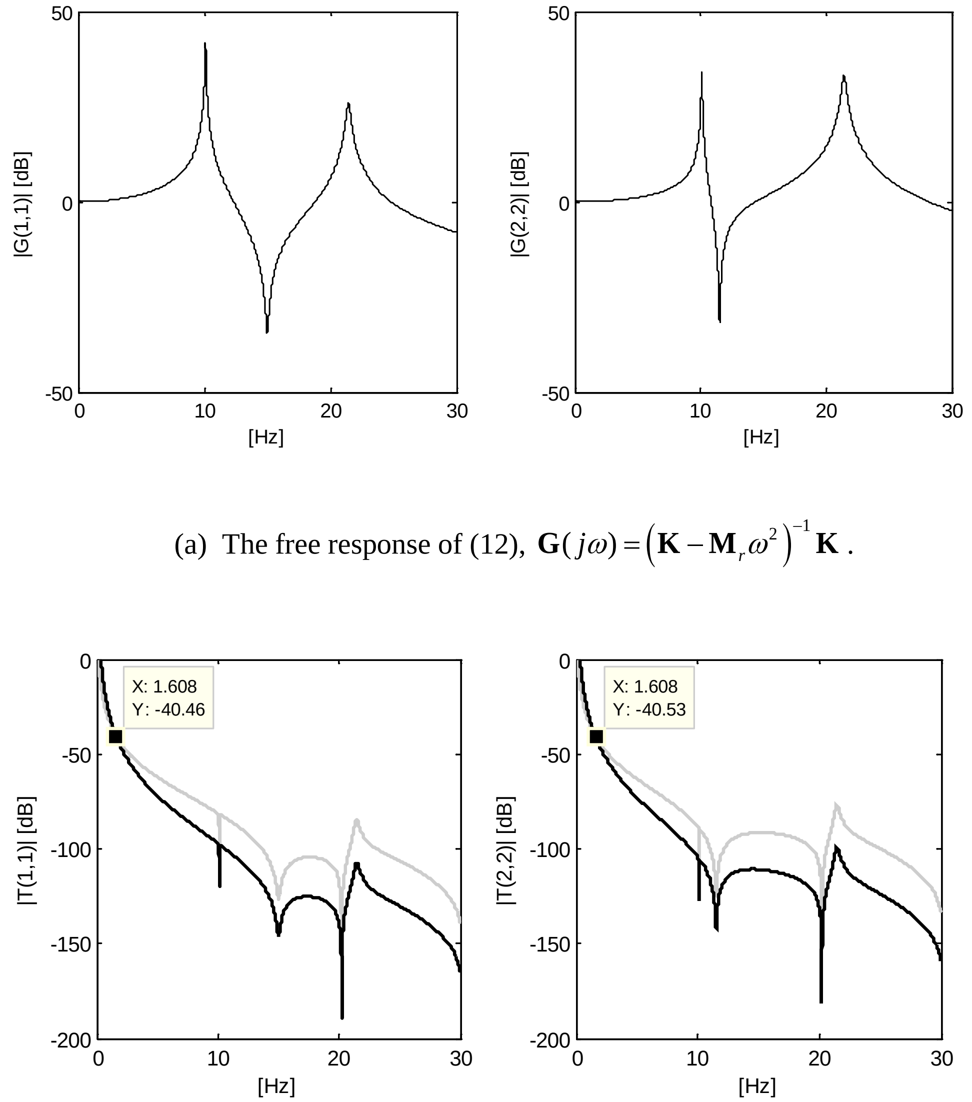
$$\left|q_r^d(j\omega)_{TVP}\right| = \sqrt{2}A\omega^{-3}\sqrt{1-\cos(T_a\omega)} \tag{4.12b}$$

From the optimal choices, (4-9) and (4-11), and assuming that the maximum accelerations are equal in both trajectories, i.e., $J = AT_j^{-1}$, then we obtain the following relation.

$$\left|\mathbf{q}_{r,i}^{d}(j\omega)_{S-curve}\right| < \left|\mathbf{q}_{r,i}^{d}(j\omega)_{TVP}\right| \text{ for } \omega > T_j^{-1}, \tag{4.13}$$

where $i = 1, \dots, n$. In Figure 4.2, an example of the above results is shown for a 2DOF robot. It can be observed that the peak magnitudes at the first mode are cancelled. Even though there are peaks at the second mode in the response (Figure 4.2(b)), the magnitudes are much reduced compared with the free response (Figure 4.2(a)). As predicted in (4.13), the magnitude of the response to the S-curve trajectory is less than the TVP for $f > \left(2\pi T_j\right)^{-1} = 1.608\ [Hz]$.

The above result suggests a suitable strategy for how the S-curve should be generated in an optimal fashion for industrial robots. That is, the acceleration and speed can still be chosen in a time-optimal manner by utilizing the rigid-body dynamic model, while the jerk period is selected as the period of the first mode of the robot. In this way, a high acceleration and speed can be achieved while suppressing the vibration modes of the robot. Notice that the above conclusion assumes that the system is linear around a stationary point. Therefore, such tuned S-curves may not work well if such assumption is not satisfied, e.g., when the robot is operating at a high speed and/or when the inertia changes dramatically during a short period of time as the robot traverses the workspace.



(a) The free response of (12), $\mathbf{G}(j\omega) = \left(\mathbf{K} - \mathbf{M}_r\omega^2\right)^{-1}\mathbf{K}$.

(b) Response to the trajectory, $\mathbf{Y}(j\omega) = \mathbf{G}(j\omega)\mathbf{R}(j\omega)$ (gray: TVP, black: S-curve).

Figure 4.2. Comparison between the TVP and the S-curve in the frequency domain. $\mathbf{A} = diag\left[10 \quad 10\right][rad/s^2]$, $f_n = \left[10.1 \quad 21.4\right][Hz]$, and $T_j = T_a = f_{n,1}^{-1}$. The model parameters are obtained by the CAD data and measurement for the 2nd and 3rd links of the robot shown in Figure 4.7. And a small damping is assumed in $\mathbf{G}(j\omega)$.

4.1.2 Input Shaping Technique

This section discusses another technique that can achieve minimum residual vibrations under a high gain PD control. The IST is a filtering method that was originally developed for cancelling residual vibrations in finite-time rest-to-rest motion tasks. It can also be applied to trajectory following tasks by ‘shaping’ the input trajectory such that the shaped trajectory has zero in magnitude at the natural frequency of the system, similar to the optimal S-curve presented in the previous section. The IST with minimum time delay, known as the ZV (Zero Vibration), is denoted by

$$h(t)_{ZV} = \alpha_0 \delta(t) + \alpha_1 \delta(t - T_z), \tag{4.14}$$

where $\alpha_0 = \dfrac{1}{1 + e^{\frac{-\pi\varsigma}{\sqrt{1-\varsigma^2}}}}$, $\alpha_1 = \dfrac{e^{\frac{-\pi\varsigma}{\sqrt{1-\varsigma^2}}}}{1 + e^{\frac{-\pi\varsigma}{\sqrt{1-\varsigma^2}}}}$, and $\delta(t)$ is the unit impulse function. For a lightly damped system, $\zeta \approx 0$, such as the flexible joint dynamic model (3.1), the ZV can be simplified as

$$h(t)_{ZV} = \frac{1}{2}\left[\delta(t) + \delta(t - T_z)\right]. \tag{4.15}$$

In the frequency domain,

$$H(j\omega)_{ZV} = \frac{1}{2}\left[1 + e^{-j\omega T_z}\right] . \tag{4.16}$$

Therefore, the reference trajectory filtered by the ZV becomes

$$\mathbf{q}_{r,i}^{d}(t)_{ZV} = \mathbf{q}_{r,i}^{d}(t) * h(t)_{ZV}, \tag{4.17}$$

Or,

$$\mathbf{q}_{r,i}^{d}(j\omega)_{ZV} = \frac{1}{2}\left[1+e^{-j\omega T_z}\right]\mathbf{q}_{r,i}^{d}(j\omega)\,. \tag{4.18}$$

As indicated in (4.18), the ZV can be interpreted as a notch-filter, adding a zero at the location, $e^{-j\omega T_z}$ in the complex plane. Applying (4.18) to (4.3), we obtain

$$\mathbf{q}_{r}^{d}(j\omega)_{ZV} = \frac{1}{2}\left[1+e^{-j\omega T_z}\right]\left(\mathbf{K}-\omega^2\mathbf{M}_r\right)^{-1}\mathbf{K}\mathbf{q}_{r}^{d}(j\omega)\,. \tag{4.19}$$

Thus, the ZV can cancel one pole of the system. However, unlike the S-curve trajectory, the ZV is not a low-pass filter. To cancel all the modes, the ZV can be designed as

$$\mathbf{q}_{r}^{d}(t)_{ZV} = h(t)_{ZV,1} * \cdots * h(t)_{ZV,n} * \mathbf{q}_{r}^{d}(t)\,. \tag{4.20}$$

In the frequency domain,

$$\mathbf{q}_{r}^{d}(j\omega)_{ZV} = \frac{1}{2^n}\prod_{i=1}^{n}\left[1+e^{-j\omega T_{z,i}}\right]\left(\mathbf{K}-\omega^2\mathbf{M}_r\right)^{-1}\mathbf{K}\mathbf{q}_{r}^{d}(j\omega)\,. \tag{4.21}$$

If we choose,

$$T_{z,i} = \left(2f_{n,i}\right)^{-1}, \tag{4.22}$$

then, all the modes cannot be excited. However, as discussed earlier, realization of (4.22) is extremely difficult in practice for industrial robots due to their complex dynamics. The modes vary significantly depending upon the robot's configurations. This means that the ZV must be designed as a time-varying filter, making it more complex for implementation. Lastly, the model (3.1) is only an approximation. There may exist unknown modes that are not considered in the model especially at high frequency. Because the ZV is not a low-pass filter, it is inherently limited to cope with uncertainties at high frequency. However, one important benefit of the ZV compared with the S-curve profiling is the 50% smaller delay (compare (4.11) with (4.22)) assuming a single DOF case. Another benefit is that the ZV can be used for arbitrary reference trajectories, allowing one to design a simpler trajectory than the S-curve trajectory. That is, one can use a time-optimal TVP trajectory and then apply the ZV. Therefore, a simple filter that can be used for industrial robots is proposed to address the time-varying multiple modes. Let us consider the combination of the ZV and the SMA (Simple Moving Average). The SMA can be represented as

$$h(t)_{SMA} = T_m^{-1}\left[\gamma(t) - \gamma(t - T_m)\right], \tag{4.23}$$

Or

$$H(j\omega)_{SMA} = j\omega^{-1}T_m^{-1}\left[1 - e^{-jT_m\omega}\right]. \tag{4.24}$$

If we combine the ZV with the SMA,

$$H(j\omega)_{ZV+SMA} = \frac{j}{2T_m\omega}\left(1 + e^{-jT_z\omega}\right)\left(1 - e^{-jT_m\omega}\right). \tag{4.25}$$

Let us choose

$$T_m = f_{n,1}^{\;-1} \text{ and } T_z = \left(3 f_{n,1}\right)^{-1}. \tag{4.26}$$

In Figure 4.3, the proposed filter (4.25) is shown assuming that the first and second mode varies in between 10 Hz and 20 Hz. A benefit of the proposed filter is that it can suppress a wide-range of frequency between $f_{n,1}$ and 2 $f_{n,1}$. For a typical industrial robot, the dominant modes are the first and the second mode. These modes are usually located close together when the robot is extending its ‘arm’ out from the shoulder because the first two modes are typically associated with the first two joints, whose inertia increases/decreases together with the robot's outstretched/retracted position. Because the filter is a low-pass filter, the unknown high frequencies can be suppressed. In Figure 4.3, the proposed filter is shown and is compared with another robust ZV, known as extra-insensitive (EI) shaper (Singhose, 2010). The delay of the proposed filter, $4\left(3f_{n,1}\right)^{-1}$, is twice that of the EI shaper, $2\left(3f_{n,1}\right)^{-1}$, but the magnitude near the target frequency (10 ~ 20 Hz) is much smaller in the proposed filter than for the EI shaper. Note that the proposed filter (ZV + SMA) is a low-pass filter whereas this is not the case for the EI shaper. The magnitudes of the two filters are shown in Figure 4.3. The peak magnitude for the proposed filter near the target frequency is only about 15 % and 20 % compared with the EI shaper and the single SMA filter, respectively.

From the perspective of residual vibrations only, the filtering technique is more attractive than S-curve profiling because the filtering technique allows one to plan with low order trajectories with smaller time delay as pointed out in Singhose (2010). Note that the proposed optimal S-curve profiling requires the addition of a jerk period for every transitions of acceleration whereas the delay of filtering method is just the filter time. For instance, let us consider a point-to-point motion with a small distance. In this case, the delay of the optimal S-curve trajectory is at least $3f_{n,1}^{-1}$ (jerk period inserted at stop to ramp-up, ramp-up to ramp-down, and ramp-down to stop) whereas the delay for the proposed filter is only $4\left(3f_{n,1}\right)^{-1}$.

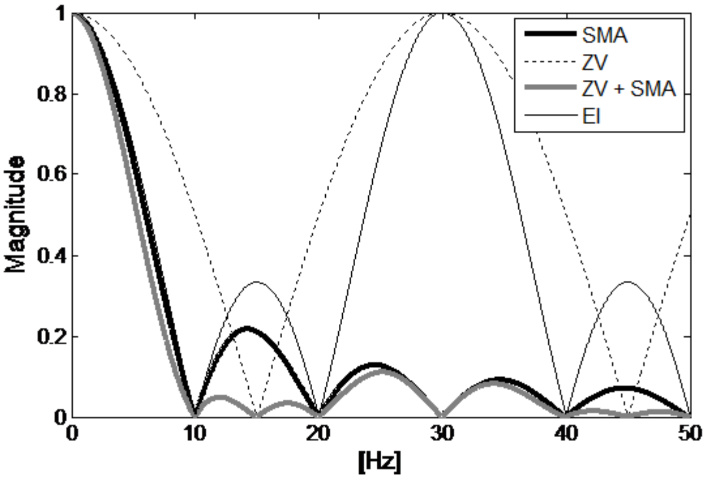


Figure 4.3. Frequency response of IST + SMA when target frequencies are in between 10 Hz and 20 Hz.

Unfortunately, filtering methods have a significant drawback as they distort the originally planned trajectory (Khoshdarregi *et al.*, 2014). The filter "shapes" the original reference trajectory so that a certain range of frequencies is cut-off. Suppose that the original reference trajectory is planned such that the end-effector follows some specific geometric path. Because the filter only addresses the frequency content of the reference trajectory, and not the geometric shape of the path, there is no way to guarantee that the geometric shape of the filtered trajectory will be preserved. This problem can be best described by illustration. In Figure 4.4, the geometric distortion by the ZV is shown. The ZV takes the average of the two points at every sampling time, namely, one for the current value, $\mathbf{q}_r^d(t)$ and one for the previous one, $\mathbf{q}_r^d(t-T_z)$. Therefore, unless the planned path is a straight-line in joint space, the planned path is distorted. And such distortion becomes more serious as the filter time is longer and the nonlinearity of the planned path is significant. The same argument also applies to the SMA or any other filters that utilizes the previous values because these filters take the average values including past trajectories.

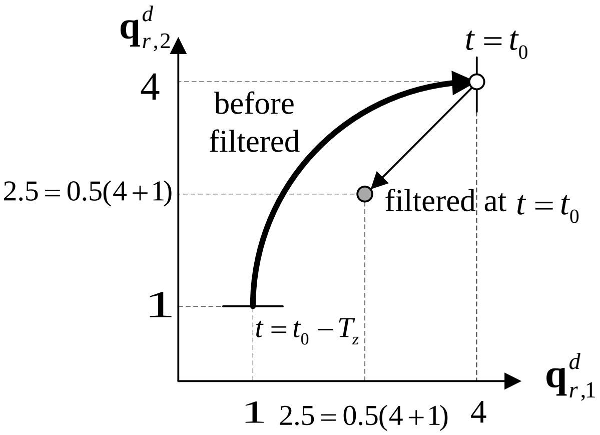


Figure 4.4. The distortion of the geometric shape by the ZV. The filtered point (gray circle) at t_0 is not on the planned path (thick line).

4.2 Control Approach

So far, the planning approach has been discussed, which aims for high motion performance by improving the reference trajectory assuming a simple PD control. In this section, control methods that have been proposed for industrial robots to improve tracking of the trajectory are addressed. In particular, two control methods that are most widely used in industry will be considered, namely nonlinear feedforward control (De Luca, 2000) and the decentralized full-state feedback control (Miyazaki *et al.*, 2004). Both controllers can be derived based on the flexible joint dynamic model (3.1). The feedforward controller requires only motor feedback and thus can be easily combined with the traditional PD controller on the motor side. The full-state controller only requires feedback up to velocity. Accordingly, it does not need to estimate high order states such as acceleration and jerk on the link side. However, the full-state control (Miyazaki *et al.*, 2004) is a decentralized scheme. Therefore, it cannot represent a true performance of a full-state controller that takes account dynamic coupling in the feedback system. Nonetheless, it is assumed that the selected full-state controller is a good choice at least for the comparative study in the thesis.

4.2.1 Feedforward Control

The key idea of the feedforward control is to “input” the required torque for the robot represented as (3.1) to follow the reference trajectory. The feedforward torque can be calculated as

$$\mathbf{u}_{ff} = \mathbf{M}_m \ddot{\mathbf{q}}_m^d + \mathbf{r}^{-1}\mathbf{K}\left(\mathbf{r}^{-1}\mathbf{q}_m^d - \mathbf{q}_r^d\right) + \mathbf{f}_m^d \,. \tag{4.27}$$

Then, the error dynamics is given by

$$\mathbf{r}\mathbf{M}_m \ddot{\mathbf{e}}_m + \mathbf{K}\left(\mathbf{r}^{-1}\mathbf{e}_m - \mathbf{e}_r\right) + \mathbf{u}_{fb,m} = \mathbf{0} \,, \tag{4.28}$$

where $\mathbf{u}_{fb,m}$ is a feedback control on the motor side. The desired motor reference is obtained by utilizing (3.1a),

$$\mathbf{q}_m^d = \mathbf{r}\mathbf{K}^{-1}\left(\mathbf{M}_r^d \ddot{\mathbf{q}}_r^d + \mathbf{C}_r^d + \mathbf{G}^d\right) + \mathbf{r}\mathbf{q}_r^d \;. \tag{4.29}$$

Utilizing the same assumption used in (3.7), we have

$$\mathbf{e}_m = \mathbf{r}\mathbf{e}_r \,. \tag{4.30}$$

Inserting (4.30) to (4.28), we obtain

$$\mathbf{r}\mathbf{M}_m \ddot{\mathbf{e}}_m + \mathbf{u}_{fb,m} = \mathbf{0} \,. \tag{4.31}$$

Therefore, it is clear that a PD control utilizing only motor measurement is sufficient to make the error dynamics in (4.31) stable. One significant benefit of the above feedforward control is that it can be combined with the industrial PD control discussed in Chapter 3. However, the feedforward torque (4.27) requires the second derivative of the motor reference (4.29). Therefore, the reference of the link side up to fourth derivative is necessary, namely,

$$\ddot{\mathbf{q}}_m^d = \frac{d^2}{dt^2}\left[\mathbf{r}\mathbf{K}^{-1}\left(\mathbf{M}_r^d\ddot{\mathbf{q}}_r^d + \mathbf{C}_r^d + \mathbf{G}^d\right) + \mathbf{r}\mathbf{q}_r^d\right]. \tag{4.32}$$

In general, the variation of the reference trajectory is much faster than the dynamic terms. In this case, one can utilize a simplified version

$$\ddot{\mathbf{q}}_m^d = \mathbf{r}\mathbf{K}^{-1}\mathbf{M}_r^d\overset{(4)}{\mathbf{q}_r^d} + \mathbf{r}\ddot{\mathbf{q}}_r^d . \tag{4.33}$$

The conversion from the reference, $\mathbf{q}_r^d$, to the motor reference, $\mathbf{q}_m^d$ in (4.29) can be regarded as a dynamic input shaper for a high gain cascaded PD controller. To see this, let us apply (4.1) to (3.1a) to obtain:

$$\mathbf{r}\mathbf{K}^{-1}\left(\mathbf{M}_r\ddot{\mathbf{q}}_r + \mathbf{C}_r + \mathbf{G}\right) + \mathbf{r}\mathbf{q}_r = \mathbf{q}_m^d . \tag{4.34}$$

Utilizing the assumption (4.3), Eq. (4.34) can be expressed in the frequency domain at steady state

$$\mathbf{q}_r(j\omega) = \left(\mathbf{I} - \omega^2\mathbf{K}^{-1}\mathbf{M}_r\right)^{-1}\mathbf{r}^{-1}\mathbf{q}_m^d(j\omega), \tag{4.35}$$

where $\mathbf{I} \in \Re^{n\times n}$ is the identity matrix.

Similarly, the mapping (4.29) can also be expressed in the frequency domain at steady state as

$$\mathbf{q}_m^d(j\omega) = \mathbf{r}\left(\mathbf{I} - \omega^2\mathbf{K}^{-1}\mathbf{M}_r\right)\mathbf{q}_r^d(j\omega). \tag{4.36}$$

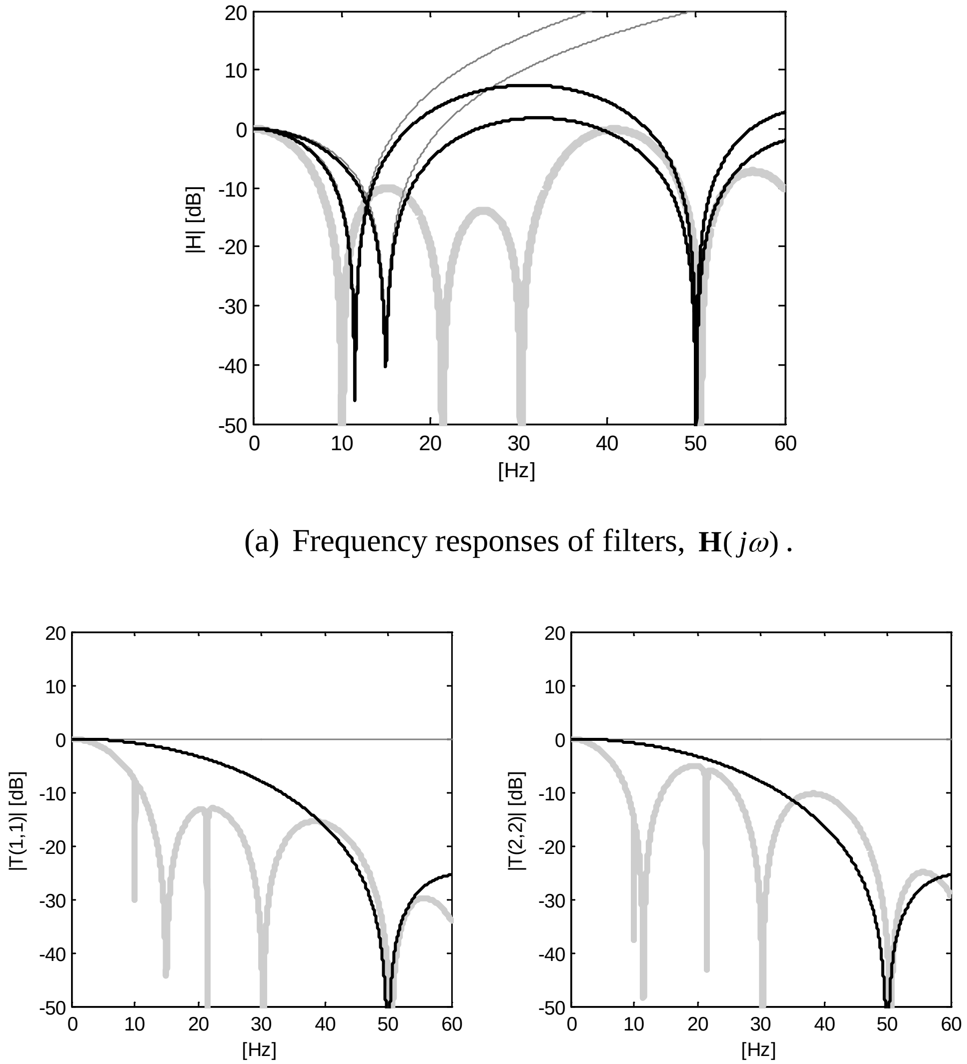
By inserting (4.36) into (4.35), the relationship between the reference and the output on the link side becomes

$$\mathbf{q}_r(j\omega) = \mathbf{q}_r^d(j\omega). \tag{4.37}$$

Therefore, the conversion (4.36) can be thought of as a multivariable dynamic input shaper such that all flexible modes of the link dynamics are cancelled simultaneously. In comparison, the traditional IST in (4.21) cancels all the poles of the link dynamics by applying a series of impulse responses to the reference trajectory and the total delay to achieve this goal is

$$T_{z,total} = \sum_{i=1}^{n} T_{z,i} \ . \tag{4.38}$$

Notably, the dynamic input shaper (4.36) achieves exactly the same goal with zero phase lag (no path-distortion) by performing the exact inverse of the transfer function of the link dynamics. In addition, the realization of the IST (4.21) requires an additional computation to solve the eigenvalue problem (4.8) which is not necessary for (4.36). Both methods assume the exact knowledge of the link inertia and joint stiffness. In addition, the dynamic input shaper is implemented in the time domain (4.29), which is as simple as the IST. Therefore, it can be concluded that the dynamic filter used in the feedforward control is more advantageous than the IST in terms of cycle time and path accuracy. The only concern regarding the dynamic input shaper is that the transfer function (4.36) is not proper. Therefore, unwanted frequency content in the reference trajectory may excite unmodeled high frequency modes of the system. It is well known that such problem can be resolved if fast poles are added so that the transfer function (4.36) becomes proper. Furthermore, the feedforward control requires trajectories whose fourth derivatives are available. Assuming that the trajectory is simple TVP (2nd order) then we need to choose an additional filter that not only makes the dynamic input shaper proper but also increases the trajectory profile to fourth order. The simplest choice for this role is the double SMA filter. In Figure 4.5, the traditional ZV and the dynamic input shaper are compared. In the figure, a double SMA filter (bandwidth 50 Hz) is used to make the dynamic input shaper proper. One can note a much smaller phase lag for the dynamic input shaper in comparison with the ZV.



(a) Frequency responses of filters, $\mathbf{H}(j\omega)$.

(b) $\mathbf{T}(j\omega) = \mathbf{H}(j\omega)\mathbf{G}(j\omega)$.

Figure 4.5. Comparison between the multi-mode ZV and the dynamic input shaper (4.36) for the model used in Figure 4.2 (thick gray: multi-mode ZV, thin gray: ideal dynamic input shaper, black: proper dynamic input shaper). Note that the elements of dynamic filter, $\mathbf{H}_{1,1}(j\omega)$ and $\mathbf{H}_{2,2}(j\omega)$ are shown together in Figure. 4.5(a).

However, the inherent limitation of the feedforward controller and the planning approach is that the feedback control is performed on the motor side, i.e., only a partial state feedback. It has been verified that the full-state control can significantly improve the tracking performance on the link side compared with the motor feedback (De Luca *et al*., 2007).

4.2.2 Full-State Feedback Control

The limitation of the feedforward controller discussed previously can be overcome by utilizing the full-state feedback. Assuming the link states are available for feedback, consider the following control law

$$\mathbf{u} = \mathbf{u}_{ff} + \mathbf{u}_{fb,m} + \mathbf{u}_{fb,r} \in \Re^n, \tag{4.39}$$

where $\mathbf{u}_{ff}$ is (4.27) and $\mathbf{u}_{fb,m}$ is the PD control on the motor side with integral control in the inner loop (see Fig. 4.6). The control law (4.39) is the motor controller combined with another feedback control based on the link states

$$\mathbf{u}_{fb,r} = \mathbf{K}_s\mathbf{e}_s + \mathbf{K}_r\dot{\mathbf{e}}_r, \tag{4.40}$$

where $\mathbf{e}_s \triangleq \mathbf{r}^{-1}\mathbf{e}_m - \mathbf{e}_r$. Then, we obtain the following error dynamics for the inner-loop (Figure 4.6) expressed in terms of the link states

$$\mathbf{J}\mathbf{M}_r\mathbf{e}_r^{(4)} + \hat{\mathbf{K}}_v\mathbf{M}_r\mathbf{e}_r^{(3)} + \left[\left(\hat{\mathbf{K}}_i + \hat{\mathbf{K}}_s\right)\mathbf{M}_r + \mathbf{K}\left(\mathbf{J} + \mathbf{M}_r\right)\right]\ddot{\mathbf{e}}_r + \mathbf{K}\left(\hat{\mathbf{K}}_v + \hat{\mathbf{K}}_r\right)\dot{\mathbf{e}}_r + \hat{\mathbf{K}}_i\mathbf{e}_r = \mathbf{0} \in \Re^n, \tag{4.41}$$

where $\mathbf{J} \triangleq \mathbf{r}^2\mathbf{M}_m$, $\hat{\mathbf{K}}_v \triangleq \mathbf{r}^2\mathbf{K}_v$, $\hat{\mathbf{K}}_i \triangleq \mathbf{r}^2\mathbf{K}_i$, $\hat{\mathbf{K}}_s \triangleq \mathbf{r}\mathbf{K}_s$, and $\hat{\mathbf{K}}_r \triangleq \mathbf{r}\mathbf{K}_r$.

Due to its coupled nature it is not obvious how the gain scheduling must be performed. If we assume that the feedforward control effectively cancels the coupling effect, then one may consider a decentralized scheme for feedback control, namely,

$$JJ_r s^4 + \hat{K}_v J_r s^3 + \left[J_r \left(\hat{K}_i + \hat{K}_s \right) + k \left(J + J_r \right) \right] s^2 + k \left(\hat{K}_v + \hat{K}_r \right) s + \hat{K}_i = 0 . \tag{4.42}$$

The transfer function of the link side is obtained as

$$\frac{q_r(s)}{q_r^d(s)} = \frac{\hat{K}_v J_r s^3 + J_r \left(\hat{K}_v + \hat{K}_s \right) s^2 + k \left(\hat{K}_v + \hat{K}_r \right) s + \hat{K}_i k}{JJ_r s^4 + \hat{K}_v J_r s^3 + \left[J_r \left(\hat{K}_i + \hat{K}_s \right) + k \left(J + J_r \right) \right] s^2 + k \left(\hat{K}_v + \hat{K}_r \right) s + \hat{K}_i} . \tag{4.43}$$

The denominator indicates that it is possible to place the poles at desired locations with a unique set of gain. However, the zeros vary depending on the gain (the numerator). In general, as the poles are further placed to the left on the real-axis, the peak magnitude of the sensitivity function (Skogestad and Postlethwaite, 1996) is increased. This implies that it is not possible to make the bandwidth of the inner loop very high. Therefore, one can expect that the above full-state controller will have poorer performance than the cascaded PD controller in terms of disturbance rejection and tracking on the motor side. Nonetheless, the main benefit of the full-state controller is that it is possible to stabilize the error dynamics on the link side.

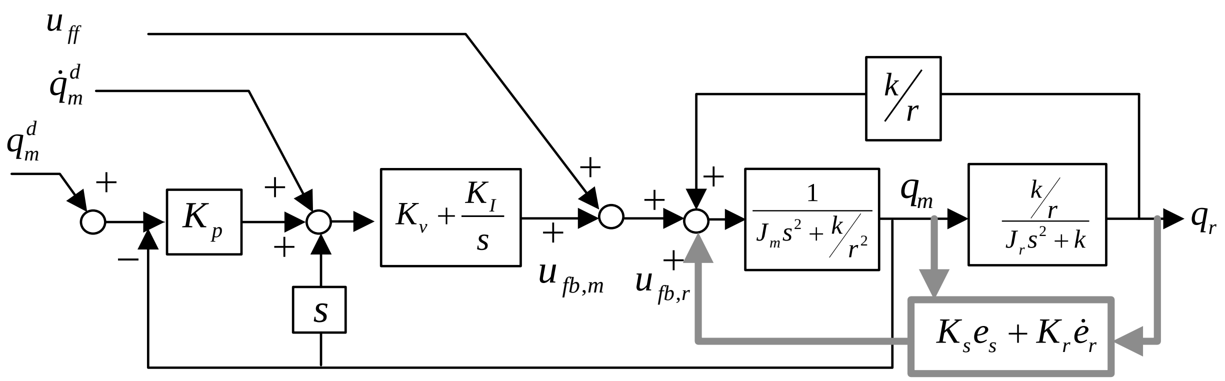


Figure 4.6. The full-state controller (Miyazaki *et al.*, 2003). The additional part for $u_{fb,r}$ is shown in thick gray lines.

4.3 Experiment

In order to verify the above discussions, the four methods have been implemented on the Hi5a controller and tested on a Hyundai robot, HA006B (Figure 4.7). A relevant simulation result can be found in Kim and Croft (2015). The planned trajectory generated by the Hi5a controller is time-optimal in the sense that the speed and acceleration are generated based on the robot dynamics (Kim *et al*., 2010). In the experiment, the robot's TCP was moved along a half circle (radius: 50 mm) in the vertical plane from its home position. A 3D laser tracker, Tracker3^{TM} is used to measure the TCP. The specification of the measurement system is available at http://apitechnical.com/Downloads/2012/T3-Product-Specifications.pdf.

In Table 4.1, the conditions for the four tested cases are summarized. The first two cases are for testing the planning approach while the other two are for the control approach. Therefore, for Case I and II, the cascaded PD controller was assumed, which is the default controller of HA006B. The bandwidth of the inner-loop and the outer-loop is approximately 100 Hz and 10 Hz, respectively. In Case I, the optimal S-curve trajectory was used, where the jerk is tuned at the first mode, which is near 10 Hz. In Case II, III, and IV, the time-optimal TVP was used. Therefore, the maximum values of speed and acceleration for the tested trajectories are the same for all four cases. In Case II, the filter proposed in Section 4.1.2 was applied to the TVP trajectory. The filter was tuned such that it suppresses modes between 10 Hz and 20 Hz (Figure 4.3). In Case III and IV, the time-optimal trajectory was filtered by a series of two small SMA filters in order to generate reference trajectory up to the fourth derivative for feedforward control and to make the dynamic input shaper to be proper. In Case IV, the nonlinear observer (Jankovic, 1995; Khalil, 2014) was utilized for the second and third joint because these two joints were dominant amongst the six joints for the tested motion. The observer was implemented outside the control loop, which runs at 500 Hz. It is because the control platform only allows a decentralized scheme. The control loop runs at 2 kHz. The feedback gain for Case IV was tuned such that bandwidth of the inner loop and the observer are about 10 Hz.

The parameters for the dynamic model were obtained from the robot's CAD data and the joint stiffness was found by the multivariable FRF (Frequency Response Functions) (Öhr *et al.*, 2006; Wernholt and Gunnarsson, 2008). In Figure 4.8, the result for the second and third joints is shown. The model was well matched with the real system up to 20 Hz which is well above the bandwidth of the outer-loop, 10 Hz.

In Figure 4.9, the planned trajectories and the reference trajectories are shown. Notice that, the duration of the reference trajectory is the longest for Case I due to the jerk period. Also notice that the time for the Case II's reference is much shorter than that of Case I even though they are both tuned at the first mode.

In Figure 4.10(a), the tracking errors on the motor side are shown. Case III achieved the best performance while Case IV generated the worst performance. This is because the bandwidth of the inner-loop for Case IV is much smaller (10 Hz) compared with other cases (100 Hz).

In Figure 4.10(b, c), the TCP measurements are shown. It is assumed that these results can represent the performance on the link side. Case I and Case IV generated smallest residual vibrations. While such reduced vibration for Case I comes from the smooth trajectory, it is due to the active damping on the link side for Case IV. The path error for Case II was much larger than other cases. This is due to the distortion of the planned trajectory by the filter. The filter delay for Case II is much larger than the others and the planned path is not a straight-line, thus generating a large distortion of the planned path, as predicted in Section 4.1.2.

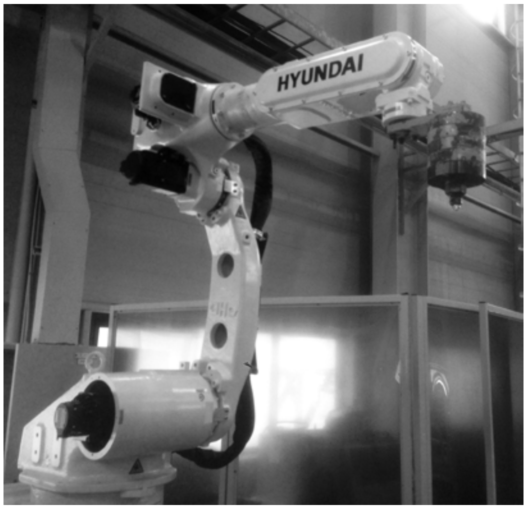


Figure 4.7. HA006B (DOF: 6, maximum payload: 6 kg, http://www.hyundai-robotics.com/robot/robot01.asp).

Case	Trajectory	Filter	Control
I	Time-optimal S-curve	SMA 50 Hz	PD control
II	Time-optimal TVP	ZV + SMA 4Hz	PD control
III	Time-optimal TVP	SMA 10 Hz	Feedforward motor control
IV	Time-optimal TVP	SMA 10 Hz	Full-state control

Table 4.1. Four cases used in experiment.

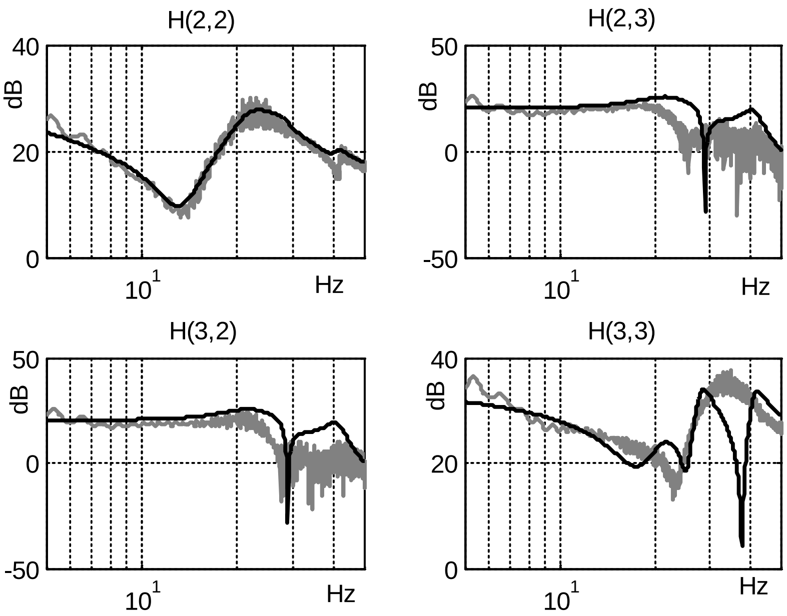


Figure 4.8. FRF of the linearized flexible dynamic model, $\mathbf{H}(i, j)$, where i and j is the joint index for the motor velocity (output) and the motor torque (input), respectively (gray: measurement, black: model).

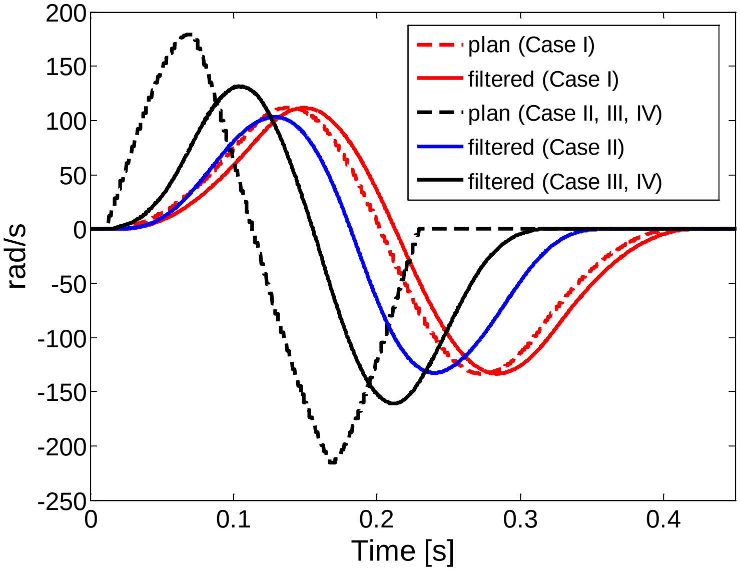
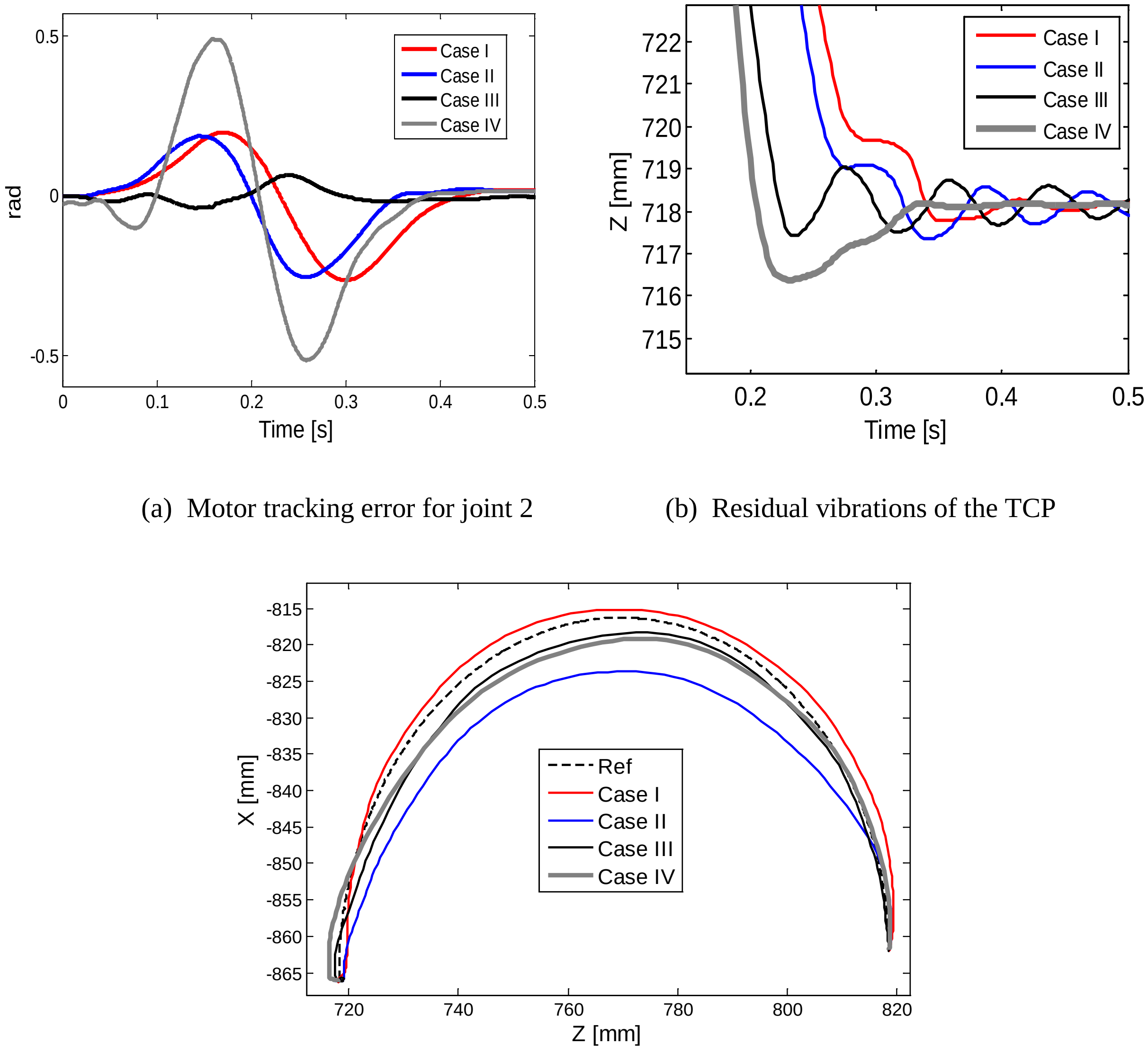


Figure 4.9. Motion trajectories for joint 2.



(a) Motor tracking error for joint 2

(b) Residual vibrations of the TCP

(c) TCP paths

Figure 4.10. Experimental results.

4.4 Summary

A comparative study on motion control methods for industrial robots has been performed. The compared methods include both planning approach and control approach. First, the planning approach was improved such that the S-curve profiling and the IST such that the jerk of the S-curve trajectory is optimal and the IST can cope with the complex dynamics of industrial robots.

The planning approach was then compared with two well-known nonlinear controllers. The theoretical analysis revealed that the planning approach has some fundamental limitations on the achievable performance due to the slow motion time (S-curve profiling) and poor path accuracy (the IST). More specifically, it was shown that the cycle-time for a point-to-point motion achieved by the S-curve profiling cannot be smaller than three times of the period of the first mode if the jerk is chosen such that no residual vibrations occur. This means that the S-curve profiling will generate slow motions in robot applications where a short-pitch motion is dominant such as spot welding. The proposed hybrid filter (IST plus SMA) can handle a time-varying multiple modes of industrial robots unlike the IST. The benefit of the proposed filter is that it provides a very simple way to design the overall motion control system because the planned trajectory can be simple as well as the control loop. One major limitation of the filtering technique, whether it is the proposed filter or some other filters, is the distortion of the planned path, that can result in a large path error especially when the robot's natural frequency is low (large filter delay) and the robot's speed is high on a curved path (a high curvature within the filter window). Therefore, the filtering technique will exhibit a poor performance in robot applications where high path accuracy is required such as machining. The filtering technique may be a valid approach only if the natural frequency of the system is very high, e.g., for CNC machines with natural frequencies much greater than industrial robots (Khoshdarregi *et al.*, 2014).

In general, the control approach has obtained better performance than the planning approach in terms of cycle-time and path accuracy (the slowest motion: S-curve, the largest path error: input shaping). The feedforward motor controller is as simple as the planning approach because the method can be implemented with a PD controller. It has been shown that a delay-free dynamic input shaper is a part of the feedforward control. Therefore, if the dynamic parameters, the link inertia and joint stiffness, are known, it is recommended that the feedforward control be used instead of the IST. The full-state controller makes the overall control system much more complex than the other methods due the additional estimation on the link states. However, the best performance was obtained by the full-state controller. In summary, the feedforward controller is a preferred choice over the planning approach if the control loop should be designed with PD control. A full-state controller must be pursued if the robot is used in applications where a high demand for speed and path accuracy must be met.

The results obtained in this chapter show that the motion control strategy proposed in Chapter 3, i.e., *the low-order trajectory with control based on elastic model*, performs better than *the high-order trajectory with PD control*. Therefore, in the next two chapters, methods for time-optimal trajectory planning and nonlinear control based on the flexible joint dynamic model are proposed.

Chapter 5: Online Path-Invariant Minimum-Time Trajectory Planning[2]

In Chapter 3, a time-optimal TVP along a specified path was suggested as it can be implemented in a low-cost controller and satisfy the PITOM. Chapter 4 demonstrated that the time-optimal TVP combined with model-based control can obtain better performance than a smooth trajectory with PD control. Motivated by this previous work, a new online trajectory planning strategy is proposed in this chapter. For online applications, the trajectory is planned along an arbitrary path with arbitrary boundary conditions. The maximum number of system-dynamics computations is limited to four per path segment, enabling generation of a time-optimal trajectory in a small number of control cycles depending upon the computational capacity of the trajectory generator. A method that can transit over multiple segments is also presented. The resulting motion is near time-optimal with respect to the system dynamics, and segments followed by the robot are not distorted by the speed at which the robot moves along the path, thus fully satisfying the PITOM, requirements proposed in Chapter 3.

A limitation of the proposed method is that some assumptions are made regarding the dynamic model and paths. Specifically, each term of the dynamic model expressed in a path parameter is assumed to be piecewise linear (between waypoints). This assumption is necessary to permit solving dynamics at the limited number of points. When the assumption is not satisfied, the proposed method may generate a trajectory that violates the dynamic constraint. However, as shown in simulation, the violation is quite small, i.e., only ±3% of maximum torque violation among 44 tested motions. The error is even less significant when considering that the dynamic

[2] A version of this chapter has been accepted for publication in *IEEE Transactions on Robotics* (Kim and Croft. Co-designed time-optimal trajectory planning and feedforward control for industrial robots).

model used for trajectory planning is only an approximated one, assumed to be perfectly rigid with no parameter uncertainty. Therefore, the “sub-optimality” of the trajectory may become less important in reality since the modeling error could be more significant. The beneficial trade-off of the proposed “sub-optimal” trajectory is the light computational burden while achieving a similar performance to existing time-optimal trajectories.

5.1 Dynamic Model and Constraints

In this section, the dynamic model used for trajectory planning is described. Since the proposed trajectory is second order, the flexible joint dynamic model (3.1) is approximated as a second order rigid model by neglecting the joint elasticity i.e., $\mathbf{K} \to \infty$. This results in the well-known rigid-body dynamic model written compactly as:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{r}\mathbf{f}_m(\mathbf{r}\dot{\mathbf{q}}) \in \Re^n , \tag{5.1}$$

where $\boldsymbol{\tau} \triangleq \mathbf{r}\,\boldsymbol{\tau}_m$, $\mathbf{q} \triangleq \mathbf{q}_r$, and $\mathbf{M} \triangleq \mathbf{M}_r + \mathbf{r}^2\mathbf{M}_m$.

Following the work (Shin and McKay, 1985; Bobrow *et al*., 1985), consider a case where the robot needs to follow a smooth path, $\mathbf{f}(s)$, where $\mathbf{f}(s)$ is a smooth path with continuity C^2 and $s = \{s \in \Re \mid 0 \le s \le 1\}$. If the path is given in joint space, the following relations hold

$$\begin{cases} \mathbf{q} = \mathbf{f}(s) \\ \dot{\mathbf{q}} = \mathbf{f}'\dot{s} \\ \ddot{\mathbf{q}} = \mathbf{f}'\ddot{s} + \mathbf{f}''\dot{s}^2 \end{cases} , \tag{5.2}$$

where $\mathbf{f}' \triangleq \dfrac{d}{ds}\mathbf{f}(s)$ and $\mathbf{f}'' \triangleq \dfrac{d^2}{ds^2}\mathbf{f}(s)$.

As indicated in (5.2), it is assumed that the second derivative of the path is available. Thus, the path must be at least C^2 – continuous. For Cartesian space paths, Eq. (5.2) holds by defining

$$\begin{cases} \mathbf{f}(s) \triangleq INV(\mathbf{x}(s)) \\ \mathbf{f}'(s) \triangleq \mathbf{J}^{-1}(s)\,\mathbf{x}'(s) \\ \mathbf{f}''(s) \triangleq \mathbf{J}^{-1'}(s)\mathbf{x}'(s) + \mathbf{J}^{-1}(s)\mathbf{x}''(s) \end{cases}, \tag{5.3}$$

where INV : inverse kinematics, $\mathbf{x}(s) \in SE(3)$, and $\mathbf{J}(s) \in \Re^{n\times n}$: manipulator Jacobian. By inserting (5.2) into (5.1) and assuming a static friction model, $\mathbf{f}_m = \mathbf{f}_v \dot{\mathbf{q}}_m + \mathbf{f}_c \operatorname{sign}(\dot{\mathbf{q}}_m)$, we obtain a dynamic model that is represented in terms of only path parameter, s , i.e.,

$$\boldsymbol{\tau} = \begin{cases} \mathbf{m}(s)\,\ddot{s} + \mathbf{b}(s)\dot{s}^2 + \mathbf{d}(s)\dot{s} + \mathbf{e}(s,\dot{s}) \in \Re^n,\ \dot{s} \geq 0 \\ \mathbf{m}(s)\,\ddot{s} + \mathbf{b}(s)\dot{s}^2 + \mathbf{d}(s)\dot{s} + \mathbf{e}(s) \in \Re^n,\ \dot{s} > 0 \end{cases}, \tag{5.4}$$

where $\mathbf{m}(s) \triangleq \mathbf{M}\mathbf{f}'$, $\mathbf{b}(s) \triangleq \mathbf{M}\mathbf{f}'' + \mathbf{C}\mathbf{f}'$, $\mathbf{d}(s) \triangleq \mathbf{r}^{\,2}\mathbf{f}_v\mathbf{f}'$, $\mathbf{e}(s,\dot{s}) \triangleq \mathbf{G} + \mathbf{r}\mathbf{f}_c \operatorname{sign}(\mathbf{f}'\dot{s})$, and $\mathbf{e}(s) \triangleq \mathbf{G} + \mathbf{r}\mathbf{f}_c \operatorname{sign}(\mathbf{f}')$. Note that all the dynamic terms in (5.4) are a function of s only for $\dot{s} > 0$, i.e., robot's position but not velocity or acceleration. For trajectory planning, it is assumed that the dynamic model (5.4) is subject to the following constraints.

$$-\left|\dot{\mathbf{q}}_{\max,i}\right| \leq \dot{\mathbf{q}}_i \leq \left|\dot{\mathbf{q}}_{\max,i}\right|,\ i = 1,\ldots,n \tag{5.5}$$

$$-\left|\boldsymbol{\tau}_{\max,i}\right| \leq \boldsymbol{\tau}_i \leq \left|\boldsymbol{\tau}_{\max,i}\right|,\ i = 1,\ldots,n \tag{5.6}$$

The inequalities, (5.5) and (5.6), are referred to as *the kinematic constraint* and *the dynamic constraint*, respectively. Given the kinematic constraint, the maximum path velocity is calculated as

$$\dot{s}_{\max}(s) = \min_{i=1,\ldots,n} \dot{\mathbf{s}}_{\max,i}(s)\,, \tag{5.7}$$

where $\dot{\mathbf{s}}_{\max,i}(s) = \left|\frac{\dot{\mathbf{q}}_{\max,i}}{\mathbf{f}_i'(s)}\right|$. The maximum acceleration satisfying (5.6) is obtained as

$$\ddot{s}_{\max}(s,\ \dot{s}) = \min_{i=1,\dots,n} \ddot{\mathbf{s}}_{\max,i}(s,\ \dot{s}) \tag{5.8a}$$

$$\ddot{s}_{\min}(s,\ \dot{s}) = \max_{i=1,\dots,n} \ddot{\mathbf{s}}_{\min,i}(s,\ \dot{s}) \tag{5.8b}$$

where

$$\ddot{\mathbf{s}}_{\max,i}(s,\dot{s}) = \begin{cases} \dfrac{\left|\boldsymbol{\tau}_{\max,i}\right| - \mathbf{c}_i(s,\dot{s})}{\mathbf{m}_i(s)}, \ \mathbf{m}_i(s) > 0 \\ \dfrac{-\left|\boldsymbol{\tau}_{\max,i}\right| - \mathbf{c}_i(s,\dot{s})}{\mathbf{m}_i(s)}, \ \mathbf{m}_i(s) < 0 \end{cases}, \ \ddot{\mathbf{s}}_{\min,i}(s,\dot{s}) = \begin{cases} \dfrac{-\left|\boldsymbol{\tau}_{\max,i}\right| - \mathbf{c}_i(s,\dot{s})}{\mathbf{m}_i(s)}, \ \mathbf{m}_i(s) > 0 \\ \dfrac{\left|\boldsymbol{\tau}_{\max,i}\right| - \mathbf{c}_i(s,\dot{s})}{\mathbf{m}_i(s)}, \ \mathbf{m}_i(s) < 0 \end{cases},$$

and, $\mathbf{c}(s,\dot{s}) \triangleq \mathbf{b}(s)\dot{s}^2 + \mathbf{d}(s)\dot{s} + \mathbf{e}(s,\dot{s})$.

5.2 Minimum-Time Trajectory Planning along Parameterized Paths

This section first considers a point-to-point case, where the robot must travel from the initial state, $\begin{bmatrix} s_0 & \dot{s}_0 \end{bmatrix}$, to the final state, $\begin{bmatrix} s_f & \dot{s}_f \end{bmatrix}$, along a specified path, $\mathbf{f}(s)$, $0 \le s_0 \le s \le s_f \le 1$, in minimum-time subject to (5-5) and (5-6), where $\dot{s}(t)$ is a TVP, i.e., a second order trajectory of C^1– continuity. This result is then extended to a case for multiple path segments.

5.2.1 Initial Trajectory

A TVP trajectory consists of three phases: i.e., a ramp, a cruise and a ramp (Figure 5.1). For a ramp phase, a constant acceleration is used. Therefore, in order to obtain the minimum-time TVP, we need to obtain three values, the initial maximum acceleration/deceleration (positive/negative acceleration), the cruise velocity, and the final maximum acceleration. Notice that the maximum velocity and acceleration are state-dependent as shown in (5.7) and (5.8). Constraints (5.5) and (5.6) are coupled since the dynamics is also a function of velocity. One way to simplify this problem is to treat (5.5) and (5.6) separately. That is, one can first consider the *Kinematic Limit Curve* (*KLC*) in the phase-plane defined by (5.7). Next, consider the *Dynamic Limit Curve* (*DLC*), the curves in the same plane that travels from the initial state to the final state by bang-bang, in which the initial and final accelerations are obtained by solving (5.8) at the boundaries.

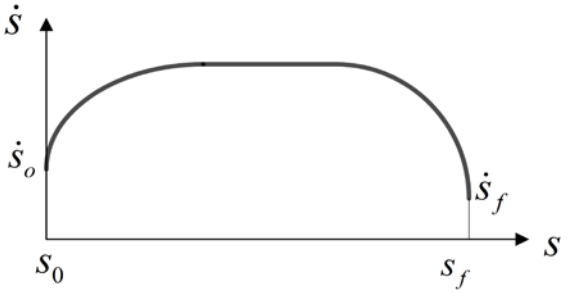


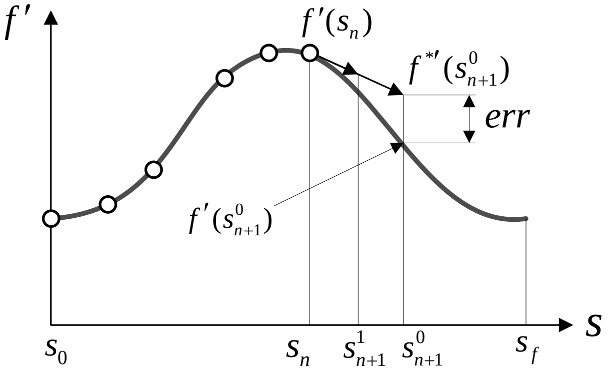
Figure 5.1. TVP shown in the phase plane for a ramp-up, a cruise, and a ramp-down.

5.2.1.1 The Kinematic Limit Curve

The *KLC* is a highly nonlinear function since it is coupled amongst joints and position-dependent. Therefore, it is difficult to represent the *KLC* as a simple mathematical function. A commonly used method to resolve this problem is to solve (5.7) prior to runtime using a small step size of s for $s_0 \le s \le s_f$ and to keep the values in a look-up table. This method is unsuitable for online planning, where a trajectory must be planned immediately whenever a new path is demanded and the computational burden must be minimized on an industrial controller. In particular, if the path is given in Cartesian space, the computational burden will increase because Eq. (5.7) includes the Jacobian matrix. A more efficient approach is to interpolate (5.7) by a spline curve with selected knot points. Representing (5.7) as a spline is beneficial in term of computational cost since a spline is a scalar function that can be easily calculated. However, finding the optimal knots is challenging for arbitrary paths due to the highly nonlinear nature of (5.7). This problem is addressed by selecting the knots based on the second derivatives of the path, $\mathbf{f}''$, i.e., the path curvature. There exist many ways to construct the algorithm using this concept. One example is shown below and in Figure 5.2.

```
function $\Delta s$ = OptimalKnot($s_n$, $s_0$, tol)
  err = BigNumber;
  k = 0;
  while err > tol
      $\Delta s = \frac{\Delta s_0}{k + 1}$;
      $s^k_{n+1} = s_n + \Delta s$;
      $\mathbf{f}^{*\prime}(s^k_{n+1}) = \mathbf{f}'(s_n) + \Delta s\, \mathbf{f}''(s_n)$;
      $err = \left\| \mathbf{f}'(s^k_{n+1}) - \mathbf{f}^{*\prime}(s^k_{n+1}) \right\|$;
      k = k + 1;
  end
end
```

(a) Pseudo code for knot point selection (tol: tolerance)



(b) Representative plot of knot finding method.

Figure 5.2. Optimal selections of the knots for spline interpolations.

At the selected knots, Eq. (5.7) is solved to construct a discrete *KLC*. The discrete *KLC* is then interpolated by a cubic spline function to obtain a continuous *KLC*. Note that the optimal knots depend upon the initial step size, Δs_0. In general, the smaller value, the more accurate results can be obtained, but at the expense of an increased computational burden. Therefore, the initial step size and the tolerance must be chosen considering the computation capacity of the planning system. Alternately, one can obtain the *KLC* by solving (5.7) with a fixed step size in order to limit the maximum computational burden for the *KLC*.

5.2.1.2 The Dynamic Limit Curve

The *DLC* is obtained by maximizing acceleration at the boundaries. Since each boundary has two extreme values by (5.8), there are a total of four maximum accelerations. Therefore, four possible types of bang-bang trajectories exist that can travel from the initial state to the final state, denoted as

$$DLC_{ij} \triangleq \left[DLC_o(s)_i \quad DLC_f(s)_j \right], \tag{5.9}$$

where

$$DLC_0(s)_i \triangleq \sqrt{\dot{s}_0^2 + 2\,\ddot{s}_{0,i}^*(s - s_0)}\ ,\ i = 1,\ 2, \tag{5.10a}$$

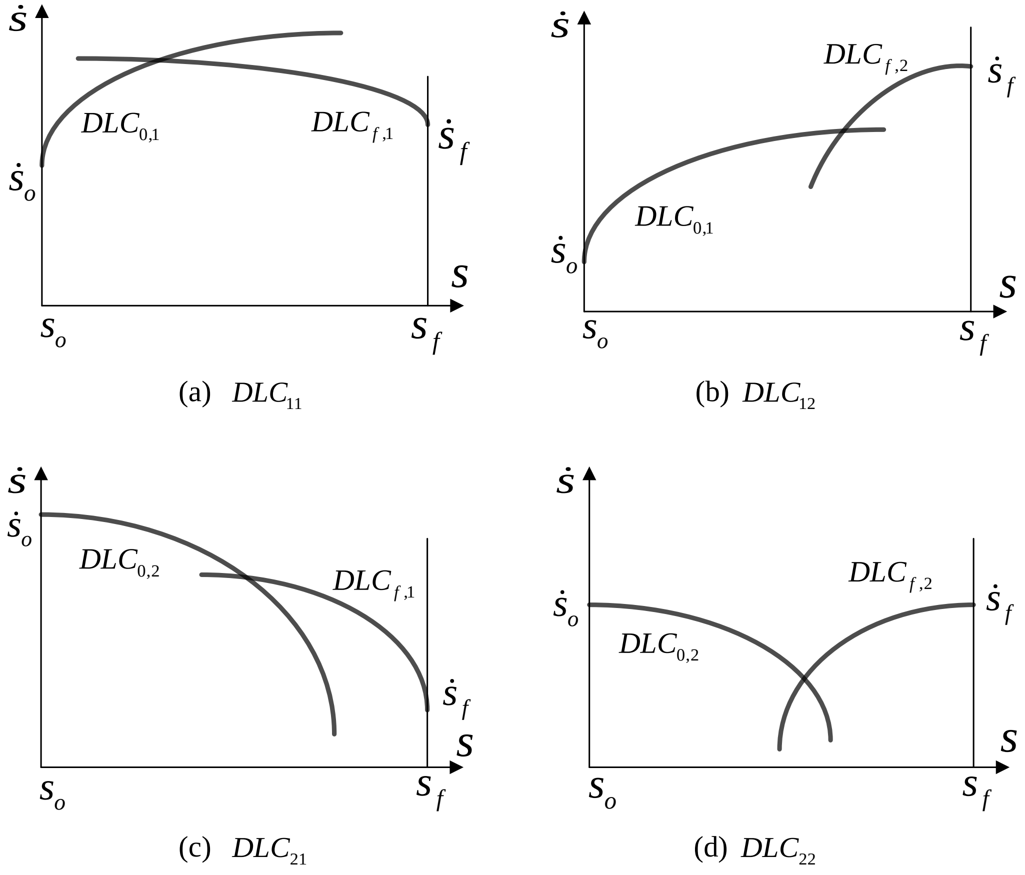
$$DLC_f(s)_j \triangleq \sqrt{\dot{s}_f^2 + 2\ddot{s}_{f,j}^*(s - s_f)}\,,\ j = 1,\ 2, \tag{5.10b}$$

$\ddot{\mathbf{s}}_0^* \triangleq \left[\ddot{s}_{\max,0} \quad \ddot{s}_{\min,0} \right]$, $\ddot{\mathbf{s}}_f^* \triangleq \left[\ddot{s}_{\min,f} \quad \ddot{s}_{\max,f} \right]$, $\ddot{s}_{0,i}^* \in \ddot{\mathbf{s}}_0^*$, and $\ddot{s}_{f,j}^* \in \ddot{\mathbf{s}}_f^*$.

Herein, we assume that the *DLC* satisfies the following condition.

$$\exists s \in [s_0,\ s_f]:\ DLC_0(s) = DLC_f(s) \tag{5.11}$$

The condition (5.8) is required to guarantee that there exists a feasible TVP trajectory generated at the boundaries such that the initial state can reach the final-state by a single switching. In case when the condition (5.8) is not met, for example, when the final velocity is too high compared with the initial velocity or *vice versa*, the higher boundary velocity needs to be lowered. In Figure 5.3, the *DLC* satisfying (5.11) are shown.



(a) DLC_{11} (b) DLC_{12}

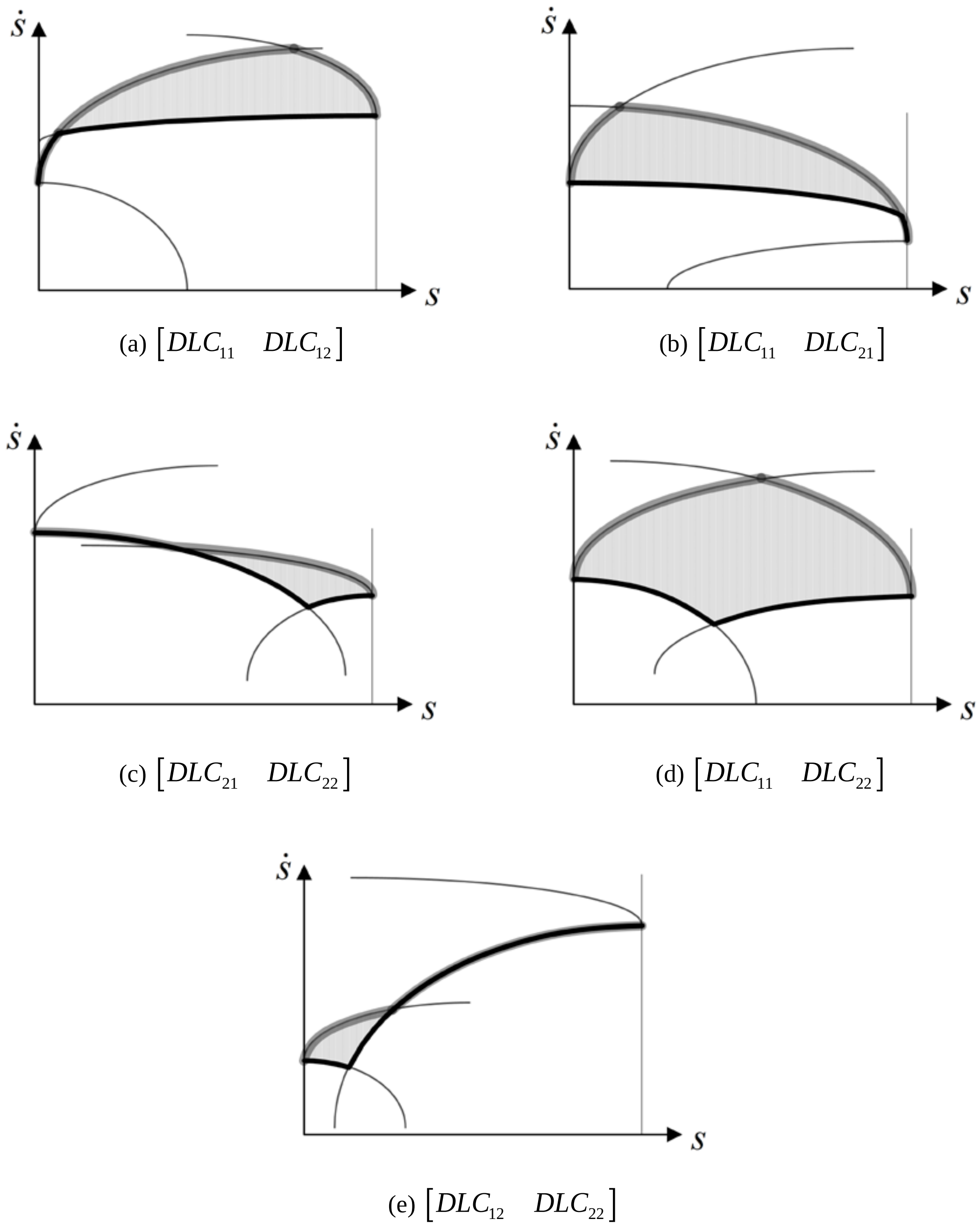
(c) DLC_{21} (d) DLC_{22}

Figure 5.3. The elements of the *DLC* generated at the boundary under (5.8).

A combination of two elements of the *DLC* defines the upper and lower bound of the bang-bang trajectory, denoted as the μDLC and λDLC, respectively. The region between μDLC and λDLC is referred to as *the admissible region,* Υ,

$$\lambda DLC \leq \Upsilon \leq \mu DLC . \tag{5.12}$$

Note that the admissible region (5.12) may violate the dynamic constraints since the *DLC* is calculated only at the boundaries. The validity of the admissible region will be addressed in Section 5.2.1.3. For the moment, let us assume that the admissible region (5.12) is valid for the entire path segment. There are five possible types of the admissible regions (Figure 5.4). These regions can be found efficiently based on the following observation. If DLC_{11} is the upper-bound, then one of the other elements is the lower-bound. And, if DLC_{11} is not the upper-bound, then either DLC_{12} or DLC_{21} is the upper-bound and DLC_{22} is the lower-bound. The admissible region collapses to one of the *DLC* bounds as shown in all figures except Figure 5.4(d).



(a) $\left[DLC_{11} \quad DLC_{12}\right]$

(b) $\left[DLC_{11} \quad DLC_{21}\right]$

(c) $\left[DLC_{21} \quad DLC_{22}\right]$

(d) $\left[DLC_{11} \quad DLC_{22}\right]$

(e) $\left[DLC_{12} \quad DLC_{22}\right]$

Figure 5.4. The admissible region is the area that can be traversed from the initial state to the final state (thick gray lines: μDLC , thick black lines: λDLC).

5.2.1.3 Finding Switching Points

The trajectory switching points[3] are found by utilizing the *KLC* and the admissible region. For a feasible trajectory to be found, the *KLC* must be above λDLC for the entire path segment, namely,

$$KLC(s) \geq \lambda DLC(s)\text{, for } s_0 \leq s \leq s_f \tag{5.13}$$

In Figure 5.5, an example of where the KLC falls below the admissible region is shown. In this case, the only way to generate a feasible trajectory is to lower the velocity at the boundaries. For the case shown in Figure 5.5, the final velocity must be lowered.

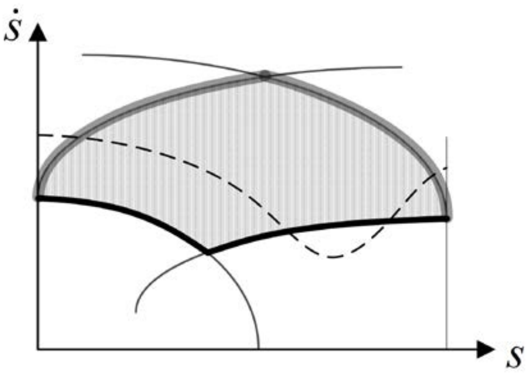
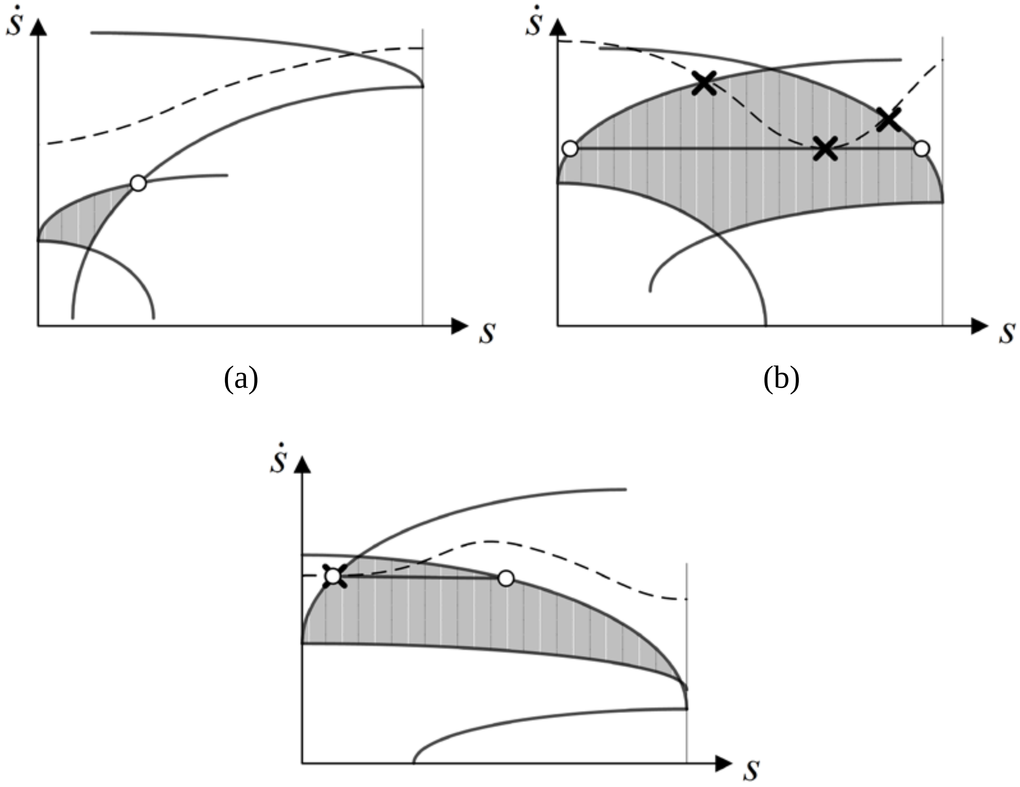


Figure 5.5. A case that the *KLC* (dashed) is below the λDLC.

If the *KLC* is above the λDLC, the switching points can be found. To do so, it is required to find the closest points to each boundary where the μDLC intersects the *KLC*. If there are no points of intersection, then the μDLC becomes the trajectory (see Figure 5.6(a)). If there exists a point of intersection, then it is necessary to find the smallest local minimum between the two intersection

[3] In general, the switching points refer to points where the directions of acceleration change. Therefore, the boundary points are also switching points. However, in the subsequent parts in the chapter, they are referred to as the ones between the boundary points.

points (see Figure 5.6(b,c)). It should be emphasized that the computational burden required to find the switching points is very small since the *KLC* and the μDLC are simple scalar functions. A numerical method, e.g., Newton-Raphson's method, can provide a fast but reasonably accurate solution.



(a)

(b)

(c)

Figure 5.6. Cases that the *KLC* (dashed) is above the λDLC (circles: switching points, x : intersection points and local minimum).

The trajectory proposed in this section is a time-optimal TVP trajectory that satisfies the kinematic limit (5.5) for the entire path segment and the dynamic limit (5.6) at the boundaries, i.e., *the initial trajectory*.

5.2.2 Trajectory Generation and Tuning

In this section, a method is proposed to tune the initial trajectory. In tuning the trajectory, it is desirable to avoid expensive dynamics computation to maintain real-time operability. To achieve this, the methods proposed in Antonelli *et al*., 2007 and Kim *et al*., 2010, are considered, where the trajectory is tuned at the switching points. However, the methods (Antonelli *et al*., 2007; Kim *et al*., 2010) do not guarantee an exact solution by solving dynamics only once at a selected point since the robot's position where the switching occurs varies as a result of the trajectory tuning. The proposed method guarantees an exact solution by a single computation of dynamics.

Before proceeding to the method, it is necessary to investigate the validity of the proposed approach. Tuning the trajectory only at switching points can be justified by analyzing Eq. (5.4). To simplify the analysis, let us first consider the case that the coefficients in Eq. (5.4) are constant.

$$\boldsymbol{\tau} = \mathbf{m}\ddot{s} + \mathbf{b}\dot{s}^2 + \mathbf{d}\dot{s} + \mathbf{G} + \mathbf{r}\mathbf{f}_c\,\mathrm{sign}(\mathbf{f}'\dot{s})\,. \tag{5.14}$$

Writing (5.14) more explicitly in the time domain, we obtain

$$\boldsymbol{\tau}(t) = \mathbf{b}\dot{s}(t)^2 + \mathbf{d}\dot{s}(t) + \mathbf{a}\,. \tag{5.15}$$

where $\mathbf{a} \triangleq \mathbf{m}\ddot{s} + \mathbf{G} + \mathbf{r}\mathbf{f}_c\,\mathrm{sign}(\mathbf{f}'\dot{s})$. Noting that $\dot{s}(t)$ is piecewise linear in the time domain (Figure 5.7), the torque in (5.15) consists of a piecewise linear function, $\mathbf{d}\dot{s}(t)$, a piecewise constant, $\mathbf{a}$, and a piecewise nonlinear function, $\mathbf{b}\dot{s}(t)^2$ on the interval

$$t \in (0,\, t_a)\,,\ (t_a,\, t_c)\,, \text{ and } (t_c,\, t_f)\,, \tag{5.16}$$

where t_a , t_c , and t_f are the time in a TVP trajectory, defined in Figure 5.7.

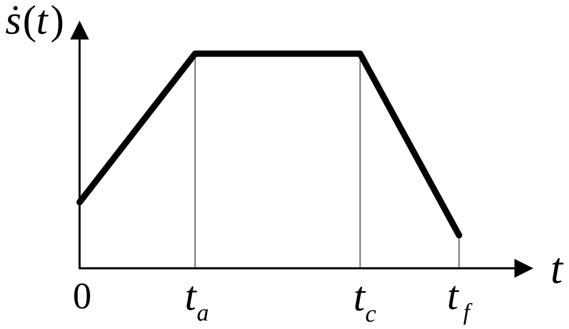


Figure 5.7. A TVP trajectory shown in the time-domain.

Note that the torque (5.15) is discontinuous at the boundaries and switching points. The term, $\mathbf{b}\,\dot{s}(t)^2$, is strictly increasing or decreasing in both the ramp phases, $t \in (0,\, t_a)$ and $(t_c,\, t_f)$. Then, the maximum torque of (5.15) must occur either at the boundaries or at switching points because the torque is a sum of piecewise linear functions and a strictly monotonic function on the interval (5.16). Now let us eliminate the assumption of the constant coefficients in (5.14). The coefficients are function of s only. That is, these coefficients vary depending upon the robot configuration only. Therefore, the above statement should be true if the variation of dynamics with respect to robot's configuration is small on each interval. Therefore, the proposed method will not work well for long path segments or near singularity configuration where the robot dynamics varies significantly.

Assuming with small variations of the dynamic terms (5.14), the method of tuning the initial trajectory is now proposed. Let us denote the torque calculated at a switching point as

$$\boldsymbol{\tau}^* \triangleq \boldsymbol{\tau}\left(s^*,\, \dot{s}^*,\, \ddot{s}^*\right) \ . \tag{5.17}$$

In case $\left|\boldsymbol{\tau}_i^*\right| > \left|\boldsymbol{\tau}_{\max,i}\right|$, $i = 1, \ldots, n$, the torque must be lowered such that the new torque, $\underline{\boldsymbol{\tau}}$, satisfies

$$-\left|\boldsymbol{\tau}_{\max,i}\right| \leq \underline{\boldsymbol{\tau}}_i \leq \left|\boldsymbol{\tau}_{\max,i}\right| , \ i = 1, \ldots, n \tag{5.18}$$

One way to achieve this is to lower either the maximum velocity or the maximum acceleration, or both. In general, this will result in a new set of switching points at different configurations and thus require new computed dynamic terms. This process continues until the torque is no longer saturated at the switching points. Alternatively, the dynamic model expressed in (5.4) permits rescaling of the torque without changing the robot configuration. Since $\dot{s} > 0$ at switching points, Eq. (5.17) can be expressed as

$$\boldsymbol{\tau}^* = \mathbf{m}(s^*)\ddot{s}^* + \mathbf{b}(s^*)\dot{s}^{*2} + \mathbf{d}(s^*)\dot{s}^* + \mathbf{e}(s^*) . \tag{5.19}$$

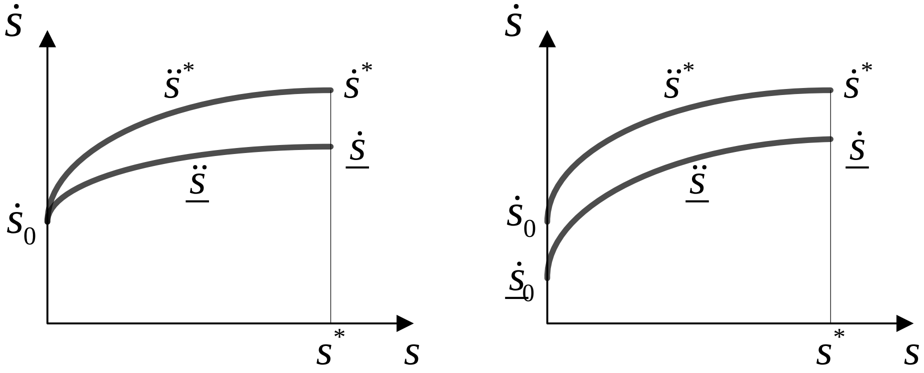
Rewriting (5-19) more simply, it obtains

$$\boldsymbol{\tau}^* = \boldsymbol{\tau}\left(s^*, \dot{s}^*, \ddot{s}^*\right) = \mathbf{m}^* \ddot{s}^* + \mathbf{b}^* \dot{s}^{*2} + \mathbf{d}^* \dot{s}^* + \mathbf{e}^* . \tag{5.20}$$

As noted in Section 5.2.1, the dynamic terms do not depend on the path velocity and acceleration. This means that it is possible to scale the torque by changing the path velocity and acceleration alone without changing the robot configuration. The new torque $\underline{\boldsymbol{\tau}}$ at the same robot configuration as (5.20) becomes

$$\underline{\boldsymbol{\tau}} = \boldsymbol{\tau}\left(s^*, \underline{\dot{s}}, \underline{\ddot{s}}\right) = \mathbf{m}^* \underline{\ddot{s}} + \mathbf{b}^* \underline{\dot{s}}^{\,2} + \mathbf{d}^* \underline{\dot{s}} + \mathbf{e}^* . \tag{5.21}$$

Eq. (5.21) simplifies trajectory tuning since it is only necessary to find $\underline{\dot{s}}$ and $\underline{\ddot{s}}$ that satisfy (5.18) with known dynamic terms. Therefore, the problem for trajectory tuning reduces to solving quadratic inequalities with constant coefficients. When the velocity and acceleration are scaled, two possible cases exist, depending on the boundary velocity. One is the case where it is possible to lower $\dot{s}^*$ and $\ddot{s}^*$ without changing the boundary velocity (Figure 5.8(a)), referred to as *the fixed-boundary velocity*. The other case is that the boundary velocity must also be lowered together with $\dot{s}^*$ and $\ddot{s}^*$ (Figure 5.8(b)), referred to as *the free-boundary velocity*.



(a) Fixed-boundary velocity (b) Free-boundary velocity

Figure 5.8. Two possible scenarios for trajectory tuning at a switching point.

A switching point for a constant velocity segment has two adjoining accelerations, i.e., zero and non-zero. Therefore, Eq. (5.21) can be written separately as

$$\underline{\boldsymbol{\tau}}_{(1)} \triangleq \boldsymbol{\tau}\left(s^*,\ \underline{\dot{s}},\ \underline{\ddot{s}}\right) = \mathbf{m}^*\,\underline{\ddot{s}} + \mathbf{b}^*\,\underline{\dot{s}}^{\,2} + \mathbf{d}^*\underline{\dot{s}} + \mathbf{e}^*\,, \tag{5.22a}$$

$$\underline{\boldsymbol{\tau}}_{(2)} \triangleq \boldsymbol{\tau}\left(s^*,\ \underline{\dot{s}},\ 0\right) = \mathbf{b}^*\,\underline{\dot{s}}^{\,2} + \mathbf{d}^*\underline{\dot{s}} + \mathbf{e}^*\,. \tag{5.22b}$$

For the fixed-boundary velocity, the new velocity and acceleration are constrained by

$$\underline{\dot{s}}^{\,2} = \dot{s}_0^2 + 2\,\underline{\ddot{s}}\,(s^* - s_0) \tag{5.23}$$

Here, only the first switching point is considered for simplicity. The same method can be applied for the second switching point. By inserting (5.23) into (5.22), it obtains:

$$\underline{\boldsymbol{\tau}}_{(\mathrm{k})} = \boldsymbol{\alpha}_{(k)}\,\underline{\dot{s}}^{\,2} + \boldsymbol{\beta}\,\underline{\dot{s}} + \boldsymbol{\gamma}\ ,\ k = 1,\, 2, \tag{5.24}$$

where $\boldsymbol{\alpha}_{(1)} \triangleq \dfrac{\mathbf{m}^*}{2(s^* - s_0)} + \mathbf{b}^*$, $\boldsymbol{\alpha}_{(2)} \triangleq \mathbf{b}^*$, $\boldsymbol{\beta} \triangleq \mathbf{d}^*$, and $\boldsymbol{\gamma} \triangleq \mathbf{e}^*$. By solving $-\left|\boldsymbol{\tau}_{\max,i}\right| \le \underline{\boldsymbol{\tau}}_{(k),i} \le \left|\boldsymbol{\tau}_{\max,i}\right|$, one can find $\underline{\dot{s}}$. Then, the acceleration can be obtained by solving (5.23).

For the free-boundary velocity, a different strategy is needed because Eq. (5.23) is no longer valid and there are three unknown variables, i.e., $\underline{\dot{s}}$, $\underline{\ddot{s}}$, and $\underline{\dot{s}}_0$. It turns out that it is possible to ensure that the switching point has the same configuration after rescaling the trajectory for the free-boundary velocity if it is chosen as:

$$\begin{cases} \underline{\dot{s}} = \lambda \dot{s}^* \\ \underline{\dot{s}}_0 = \lambda \dot{s}_0^{\ *} \\ \underline{\ddot{s}} = \lambda^2 \ddot{s}^* \end{cases}, \tag{5.25}$$

since

$$s^* = \frac{\dot{s}^{*2} - \dot{s}_0^{*2}}{2\ddot{s}^*} + s_0 = \frac{\left(\dot{s}^{*2} - \dot{s}_0^{*2}\right)\lambda^2}{2\ddot{s}^*\lambda^2} + s_0 = \frac{\underline{\dot{s}}^{\,2} - \underline{\dot{s}}_0^{\,2}}{2\,\underline{\ddot{s}}} + s_0\,. \tag{5.26}$$

By inserting (5.25) into (5.21), we obtain:

$$\underline{\boldsymbol{\tau}} = \boldsymbol{\lambda}^2\left(\mathbf{m}^*\ddot{s}^* + \mathbf{b}^*\dot{s}^{*\,2}\right) + \boldsymbol{\lambda}\,\mathbf{d}^*\dot{s}^* + \mathbf{e}^*\ \ , \tag{5.27}$$

where $\boldsymbol{\lambda} \triangleq diag\left[\lambda_1 \quad \cdots \quad \lambda_n\right]$. By solving $-\left|\boldsymbol{\tau}_{\max,i}\right| \le \underline{\boldsymbol{\tau}} \le \left|\boldsymbol{\tau}_{\max,i}\right|$ and choosing the smallest λ, the three unknowns can be obtained by (5.25).

5.3 Continuous Motions for Multiple Path Segments

So far, trajectory generation for a single path segment has been considered. In this section, the method presented previously is extended to continuous motions. In Figure 5.9 and Figure 5.10, the problem for trajectory planning for multiple segments is shown. Assume that the robot is initially stopped at $\mathbf{p}_{k-1}$. Suppose that the robot is now requested to move to $\mathbf{p}_k$ along a specified path segment, $\mathbf{f}_k$ (Figure 5.9(a)). The path parameter, s_k, satisfies $\mathbf{f}_k(0) = \mathbf{p}_{k-1}$ and $\mathbf{f}_k(1) = \mathbf{p}_k$.

It first plans a trajectory $s_k(t)$ utilizing the proposed method in previous sections. Next, suppose that the trajectory $s_k(t)$ is executed and the robot is moving along $\mathbf{f}_k$ accordingly when a possible collision is detected by a higher-level command (Figure 5.9(b)). The higher-level command assigns a new target, $\mathbf{p}_{k+1}$, a new path segment, $\mathbf{f}_{k+1}$ and the boundary positions, $s_{k,i}$ and $s_{k+1,j}$ required to avoid the collision (Figure 5.10). Then, it is possible to find a smooth curve, $\mathbf{f}_c$ between $\mathbf{f}_k$ and $\mathbf{f}_{k+1}$ such that the robot can transit without colliding with the object.

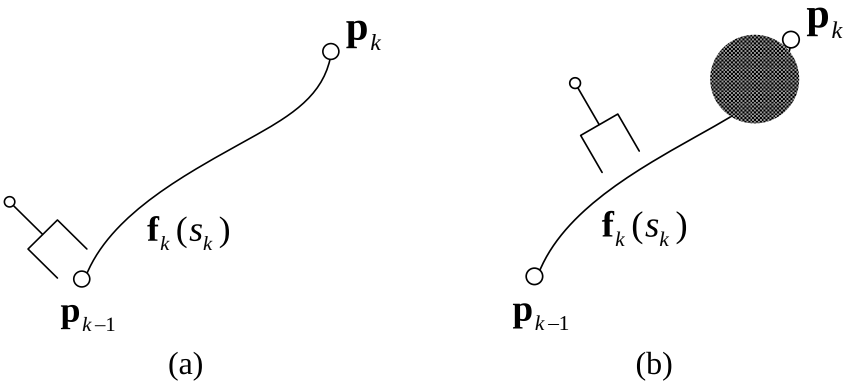


Figure 5.9. A robot moving along a single path segment with an imminent collision.

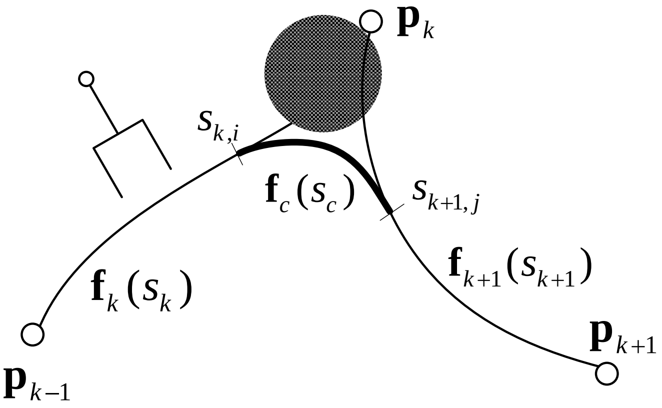


Figure 5.10. A robot moving along multiple segments.

Next, the minimum-time trajectory, $s_{k+1}(t)$ is planned for $\mathbf{f}_{k+1}$ assuming zero boundary conditions at the endpoints, $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$. Since the proposed trajectory is only C^1 – continuous, the connecting segment, $\mathbf{f}_c$, must only satisfy the following boundary conditions

$$\begin{cases} \dot{\mathbf{f}}_c(s_{c,0}) = \dot{\mathbf{f}}_k(s_{k,i}) \\ \dot{\mathbf{f}}_c(s_{c,f}) = \dot{\mathbf{f}}_{k+1}(s_{k+1,j}) \end{cases}. \tag{5.28}$$

Rewriting (5.28), we obtain

$$\begin{cases} \dot{s}_{c,0}\left\|\mathbf{f}'_c(s_{c,0})\right\|\mathbf{u}_{c,0} = \dot{s}_{k,i}\left\|\mathbf{f}'_k(s_{k,i})\right\|\mathbf{u}_{k,i} \\ \dot{s}_{c,f}\left\|\mathbf{f}'_c(s_{c,f})\right\|\mathbf{u}_{c,f} = \dot{s}_{k+1,j}\left\|\mathbf{f}'_{k+1}(s_{k+1,j})\right\|\mathbf{u}_{k+1,j} \end{cases}, \tag{5.29}$$

where $\mathbf{u}$ is the unit direction vector. In order to satisfy these equalities for arbitrary boundary velocities, $\dot{s}_{k,i}$ and $\dot{s}_{k+1,j}$, the direction vectors at both sides must vanish, i.e.,

$$\mathbf{u}_{c,0} = \mathbf{u}_{k,i} \text{ and } \mathbf{u}_{c,f} = \mathbf{u}_{k+1,j}\,. \tag{5.30}$$

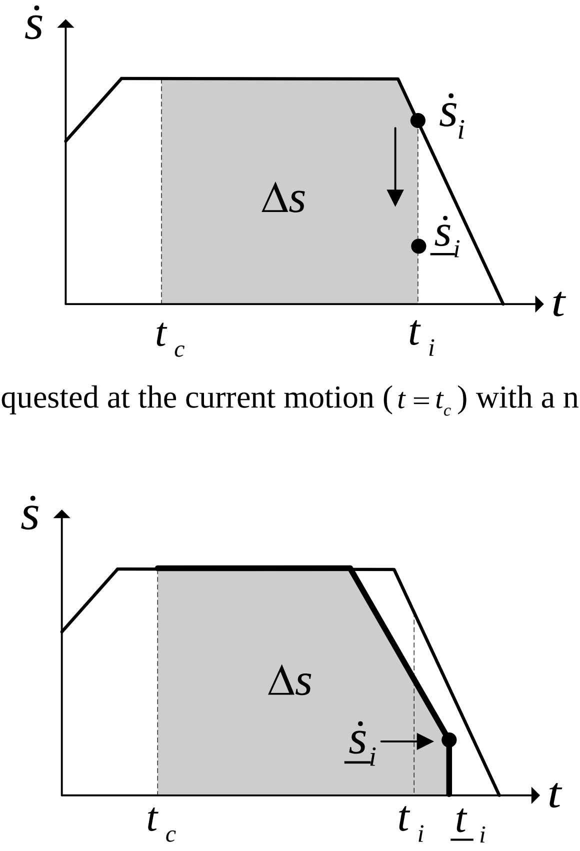
Assuming that the condition (5.30) is satisfied, we obtain

$$\begin{aligned} \dot{s}_{c,0}\left\|\mathbf{f}'_c(s_{c,0})\right\| &= \dot{s}_{k,i}\left\|\mathbf{f}'_k(s_{k,i})\right\| \\ \dot{s}_{c,f}\left\|\mathbf{f}'_c(s_{c,f})\right\| &= \dot{s}_{k+1,j}\left\|\mathbf{f}'_{k+1}(s_{k+1,j})\right\| \end{aligned}\,. \tag{5.31}$$

As seen from (5.31), the connecting segment, the only requirement for $\mathbf{f}_c$ is to satisfy (5.30). Thus, various curves can be used. Bézier curves are a good candidate for this role since the first derivatives of the path, $\mathbf{f}'_c(s_{c,0})$ and $\mathbf{f}'_c(s_{c,f})$, can be easily controlled by setting control points and it has the convex-hull property; the curve exists inside the control polygon (Shoemake, 1985; Park and Ravani, 1995). As a result, it is possible to plan $\mathbf{f}_c$ such that it exists between $\mathbf{f}_k$ and $\mathbf{f}_{k+1}$. Assuming that $\mathbf{f}_c$ has been planned, then the boundary velocities for $\mathbf{f}_c$ can be obtained as

$$\dot{s}_{c,0} = \frac{\left\|\mathbf{f}'_k(s_{k,i})\right\|}{\left\|\mathbf{f}'_c(s_{c,0})\right\|}\dot{s}_{k,i} \text{ and } \dot{s}_{c,f} = \frac{\left\|\mathbf{f}'_{k+1}(s_{k+1,j})\right\|}{\left\|\mathbf{f}'_c(s_{c,f})\right\|}\dot{s}_{k+1,j}\,. \tag{5.32}$$

Note that all the terms in the right-hand side in (5.32) are known because the trajectories for $\mathbf{f}_k$ and $\mathbf{f}_{k+1}$ have been already planned. Now it is possible to plan the trajectory, $s_c(t)$, with the boundary conditions obtained by (5.32). First, plan the trajectory according to the method described in the previous sections. In case when the boundary velocities are lowered (*free-boundary velocity*), the trajectories for $s_k(t)$ and $s_{k+1}(t)$ are replanned with new boundary conditions given by (5.32). For replanning $s_k(t)$ and $s_{k+1}(t)$, it must solve for the *fixed-boundary velocity*. However, during replanning $s_k(t)$, it is possible that no feasible trajectory for $s_k(t)$ with fixed-boundary velocities can be found. This case can occur when the transition from $\mathbf{f}_k$ to $\mathbf{f}_c$ must be executed immediately at a high speed while the boundary velocity, $\dot{s}_{k,i}$ is very small. In Figure 5.11, an example of replanning $s_k(t)$ is described. Let us assume that the new path segment $\mathbf{f}_{k+1}$ is entered at $t = t_c$ and assume that the boundary condition (5.32) has been obtained. Then, it may be required that the original velocity at $s_{k,i} = s_k(t_i)$ be lowered from $\dot{s}_{k,i}$ to $\underline{\dot{s}}_{k,i}$. In this case, $s_k(t)$ is replanned from the current state, $\left[s_k(t_c) \quad \dot{s}_k(t_c)\right]$, i.e., the new initial state, to the new final state, $\left[s_{k,i} \quad \underline{\dot{s}}_{k,i}\right]$. However, successful replanning of $s_k(t)$ may not be possible if the time between t_c and t_i is too short because the maximum deceleration required to arrive at the final state can violate the dynamic constraint. In this case, the transition path, $\mathbf{f}_c$, must be modified to find a feasible trajectory. Unlike the case for $s_k(t)$, the problem for replanning $s_{k+1}(t)$ is much simpler due to the fact that the final velocity of $s_{k+1}(t)$ is fixed at zero. Note that the final velocity of $s_{k+1}(t)$ must be zero because a further path segment after $\mathbf{f}_{k+1}$ may not be requested by the high-level motion generator.



(a) A replanning is requested at the current motion ($t = t_c$) with a new target state $\left[s(t_i) \quad \underline{\dot{s}}_i \right]$.

(b) The time to arrive at s_i is increased after replanning.

Figure 5.11. An example of replanning $s_k(t)$. Note that the distance to be planned (shaded area) is the same before and after replanning.

5.4 Simulation

The method proposed in Sections 5.2 and 5.3 has been simulated in Matlab$^{\text{TM}}$ for a 2DOF robot assuming a rigid-body model. The parameters used in simulation can be found in Table 5.1. The paths are given in joint space. For the single-segment case, a total of 44 path segments were tested (24 straight line segments and 20 second order Bézier curve, see Figure 5.12). The joint coordinates shown in Figure 5.12 follow the standard convention for a planar 2DOF robot, e.g., Hollerbach (1984). To evaluate the performance of the proposed trajectory, the following performance measures are defined.

$$R_{v,k} \triangleq \max_{0 \le t \le T_k} R_v(t) \tag{5.33}$$

$$R_{\tau,k} \triangleq \max_{0 \le t \le T_k} R_\tau(t) \tag{5.34}$$

$$R_{p,k} \triangleq \frac{1}{T_k} \int_{t=0}^{t=T_k} \max\left[R_v(t) \quad R_\tau(t) \right] dt \tag{5.35}$$

where $R_v(t) \triangleq \max_{i=1,\ldots,n} \left| \frac{\dot{q}_i(t)}{\dot{q}_{\max,i}} \right|$, $R_\tau(t) \triangleq \max_{i=1,\ldots,n} \left| \frac{\tau_i(t)}{\tau_{\max,i}} \right|$, k : test index, and T_k : motion time.

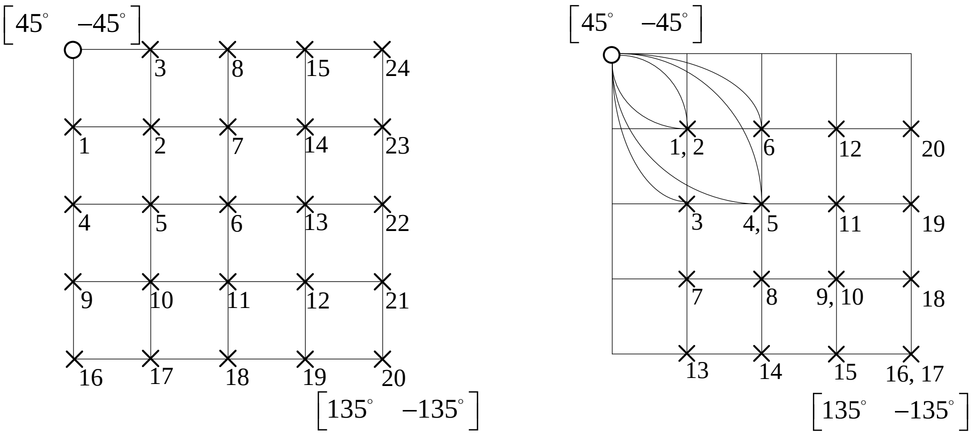
The first two measures represent the peak ratio of the velocity and torque, respectively, achieved along the trajectory amongst all joints. The last one is the average ratio of the actuator utilization. Thus, the measures (5.33) and (5.34) indicate how well the proposed method satisfies the constraints (5.5) and (5.6) whereas the measure (5.35) can be used to compare with a true time-optimal trajectory. The measure, $R_{p,k}$, is one only if the joint velocity and torque are saturated at all times. Therefore, the proposed method will generate a $R_{p,k}$ less than one because the trajectory is designed such that the joint velocity and torque are saturated only once along a path segment.

The simulation results are plotted in Figure 5.13. For the straight line case, the resulting trajectories are very close to time-optimality ($R_{v,k} = 97 \pm 4.8\%$, $R_{\tau,k} = 101 \pm 1.2\%$, and $R_{p,k} = 98 \pm 1.3\%$, expressed in a mean value with standard deviation). For curves, the trajectories are less efficient than the straight line case ($R_{v,k} = 97 \pm 7.2\%$, $R_{\tau,k} = 100 \pm 0.3\%$, $R_{p,k} = 75 \pm 7.7\%$). Since $R_{\tau,k} > 100\%$, the torque constraints were violated. However, the violation was small. The maximum torque violation was about 3 %. The trajectories along the straight lines are closer to the true time-optimal trajectories compared with the ones obtained along the curved paths. This result is consistent with the work in Sahar and Hollerbach (1986), where it is shown that an unconstrained time-optimal path is close to a straight line in joint space.

For the multiple-segment case, it is assumed that the new path segments are commanded at $0s$, $0.1s$, and $0.42s$. The segments are comprised of straight lines and second order Bézier curves. An exemplar multi-segment trajectory is shown in Figure 5.14.

Parameter	Link 1	Link 2
$m\,[kg]$	10	21
$J_r[kg \cdot m^2]$	1	1.5
$J_m\ [10^3 kg \cdot m^2]$	0.909	0.169
$f_v\,[Nm \cdot s / rad]$	0.0017	0.0004
$f_c\,[Nm]$	0.7819	0.4709
r	145	137

Table 5.1. Model parameters for simulation. The distance of the center of mass from each joint is 0.36 m and 0.6 m.

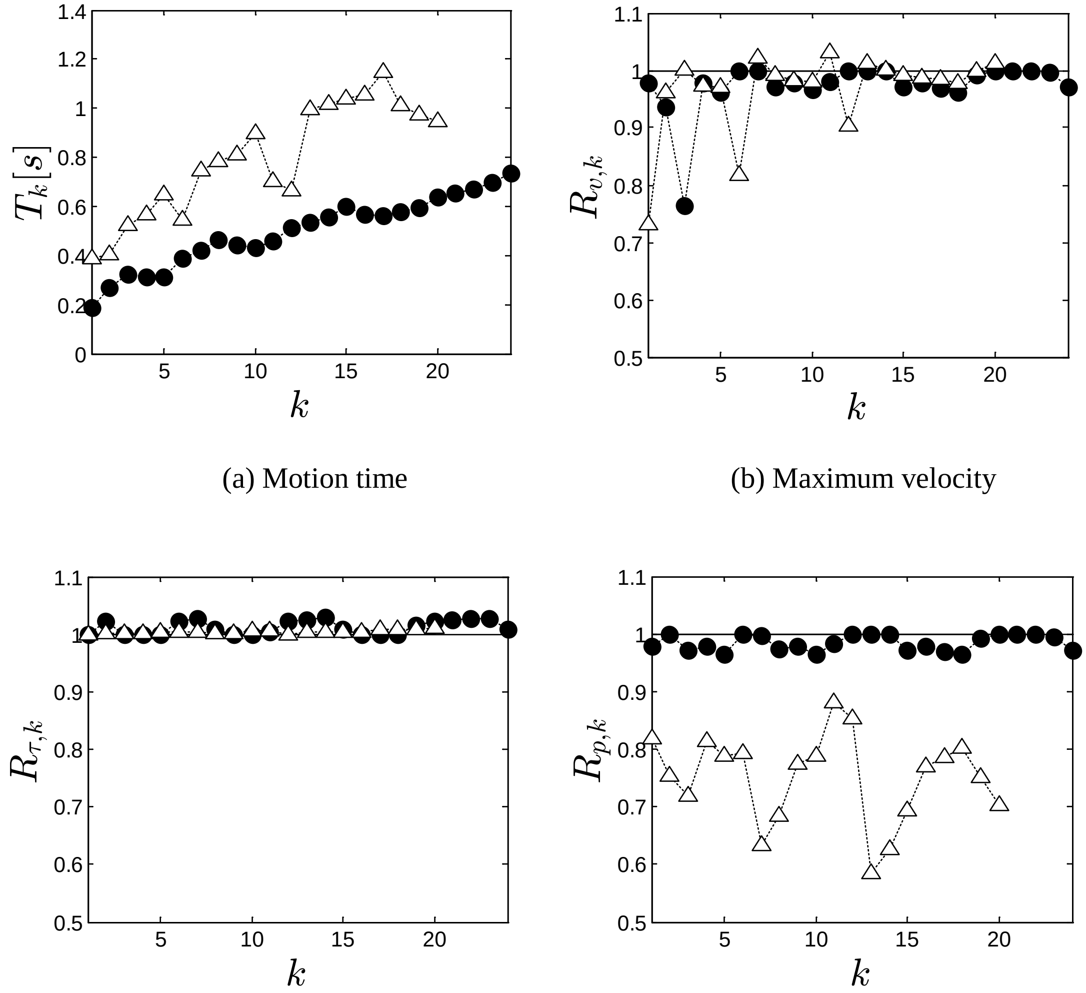


$\begin{bmatrix}45^\circ & -45^\circ\end{bmatrix}$
3
8
15
24
1
2
7
14
23
4
5
6
13
22
9
10
11
12
21
16
17
18
19
20
$\begin{bmatrix}135^\circ & -135^\circ\end{bmatrix}$
$\begin{bmatrix}45^\circ & -45^\circ\end{bmatrix}$
1, 2
6
12
20
3
4, 5
11
19
7
8
9, 10
18
13
14
15
16, 17
$\begin{bmatrix}135^\circ & -135^\circ\end{bmatrix}$

(a) Straight line

(b) second order Bézier

Figure 5.12. Target positions for simulation (circle: initial position, numbers: simulation indices).



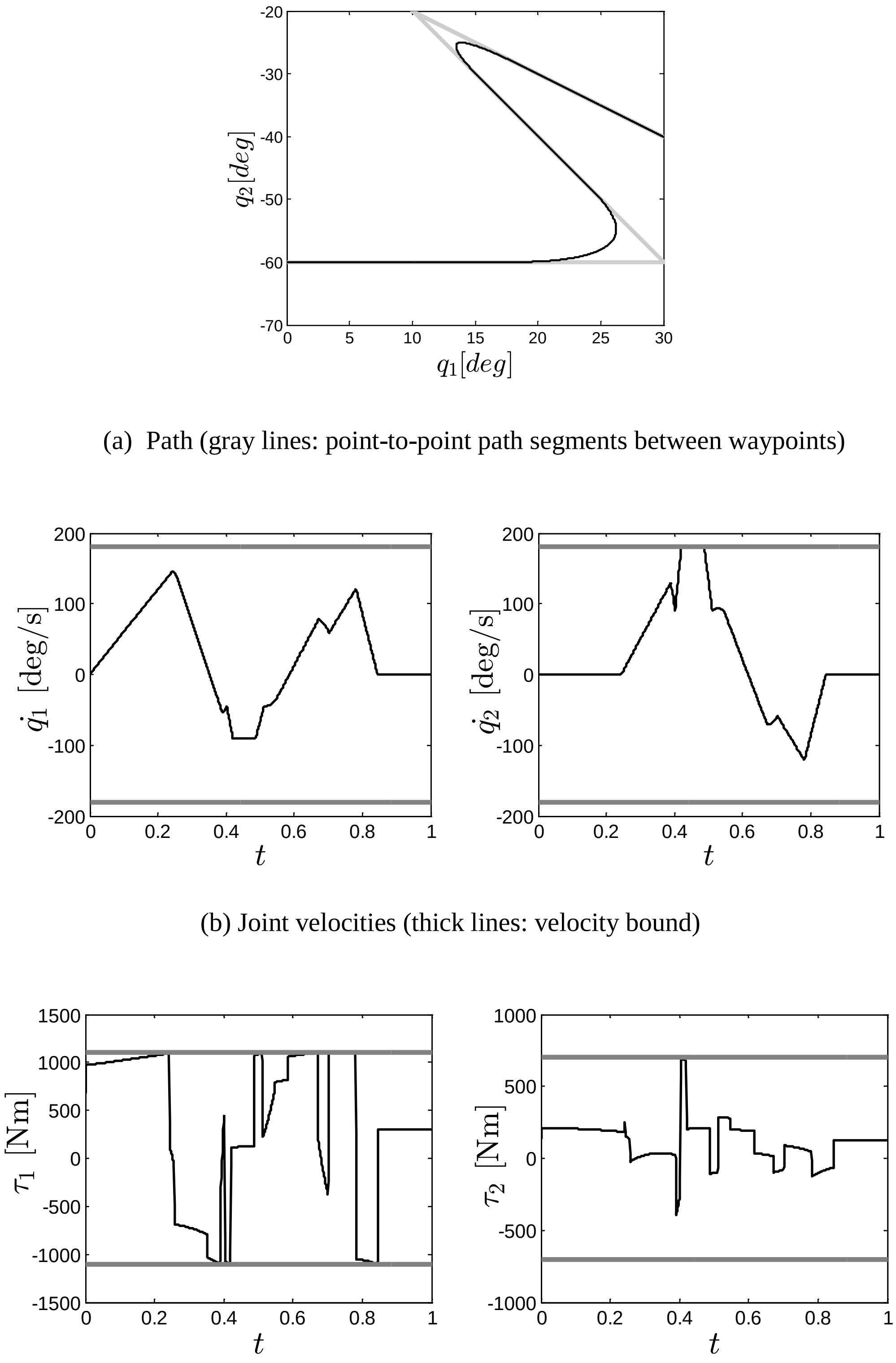
(a) Motion time

(b) Maximum velocity

(c) Maximum torque

(d) Maximum actuator utilization

Figure 5.13. Simulation result for single-segment case (●: straight line, Δ :2nd order Bézier, k : simulation index).



(a) Path (gray lines: point-to-point path segments between waypoints)

(b) Joint velocities (thick lines: velocity bound)

(c) Joint torques (thick lines: torque bound)

Figure 5.14. Simulation results for an exemplar multiple-segment case.

5.5 Experimental Results

The motion control method presented herein has been tested experimentally with HA006B (Figure 4.7). For the purpose of the experiment, the proposed time-optimal TVP trajectory and the feedforward controller were implemented in the Hi5a controller. The robot program used in the experiment consists of four points including the home position. As shown in Figure 5.15, three path segments are used; a linear path in joint space, denoted as *J*, and two linear paths in Cartesian space, *L*. Three different conditions for the path speed are used in the experiment. That is, the robot is asked to move the path with 100 mm/s, 500 mm/s, and 1000 mm/s for the same path. The TCP distance between each way-point is 100 mm. It is assumed that the transitions occur at 20 mm circle from each way-point. The transition path between $J - L$ is planned by a Bézier curve whereas a circular path is used for $L - L$.

The motion control of the Hi5a controller has two parts, trajectory generation and motor control, which runs at 500 Hz and 2.5 kHz, respectively, in separate hardware systems, where a shared memory is used for communication between the two systems. In Figure 5.16, the block diagram of the implemented method is described. The trajectory planner and the feedforward control were implemented in the trajectory generation module.

In order to meet the constraint on the computational burden of the controller, the trajectory planner was implemented in the following way. First of all, the *KLC* is estimated using a fixed step size, i.e., at five points that are evenly spaced along the given path. In addition, in one sampling period, only a single segment is planned. In Table 5.2, the computational load of the proposed method is shown assuming that the path is given in Cartesian space. The load is shown in terms of the number of times that the controller must compute the dynamic terms (DT), and the Jacobian matrix (JT), simplifying comparison to other methods without having to delve into the complexity of each controller's computational architecture.

At the instant when the new path segment is requested (Figure 5.9(b)), the trajectory for the next path segment, $s_{k+1}(t)$, is calculated during the current sampling period. Then, at the next period, the trajectory for the transition path, $s_c(t)$, is generated. Finally, the trajectories, $s_k(t)$ and $s_{k+1}(t)$, are replanned in the third period. Therefore, the complete trajectory for desired trajectory is planned 'on the fly' in three sampling periods, i.e., 6 ms. The maximum torque limit was set to 80 % of the maximum joint torque to cope with parameter uncertainty of the dynamic model.

In Table 5.3, the methods used in the experiment are shown. The proposed scheme is compared with the conventional scheme. Both schemes assume the proposed TVP trajectory with PD control. The only difference is the feedforward action. The proposed scheme utilizes the method discussed in Chapter 4 whereas the conventional scheme uses the feedforward control based on a rigid-body model, which can be regarded as a feedforward version of computed torque control. The feedback controller of the Hi5a controller is a cascaded PD control. In the experiment, the default gains were used.

The proposed second order trajectory is filtered by double SMA filter. For the proposed scheme, the purpose of the filter is to increase the order of the trajectory for feedforward control. For the conventional scheme, however, it can also smooth the trajectory such that the high frequency content that excites the flexible modes is eliminated. One can note that the filter size can be arbitrarily small for the proposed scheme if the tested robot is perfectly represented as an ideal flexible joint dynamic model (3.1). However, the filter size must be carefully selected for the proposed scheme in reality due to model uncertainty. Too small filter size generates a large overshoot and significant vibrations even with feedforward action based on a relatively accurate flexible model. As shown in Figure 5.17, the flexible model is sufficiently accurate because the measurement and the model are well matched up to 20 Hz, twice the bandwidth of the control system, 10 Hz. The filters were tuned such that there were no residual vibrations by visual inspection. The conventional scheme required much larger filter size, 160 ms (100 ms, 60 ms), whereas for the proposed scheme the filter size is only 80 ms (50 ms, 30 ms). Without the filter, the both schemes exhibited large vibrations and the 3D laser tracker, Tracker3^{TM} failed to measure the TCP positions. The main negative effect of the filter, other than introducing constant delay, is that the geometric shape planned by the TVP trajectory is distorted as mentioned in Chapter 4. The path-distortion grows as the filter size or robot speed increases. As shown in Figure 5.18, the conventional scheme generates a large deviation of the path (near waypoints), although the tracking errors on the motor side are relatively small due to the high gain PD control. The path deviation is much smaller in the proposed scheme due to the smaller filter size and better tracking performance (Figure 5.19).

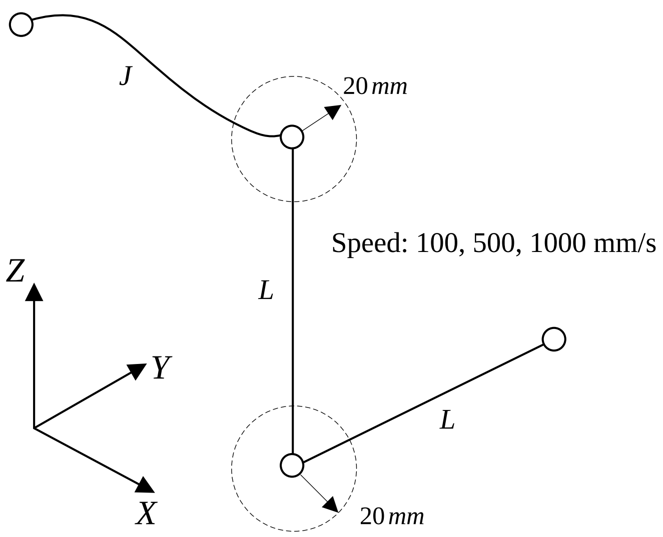


Figure 5.15. The robot program used in the experiment.

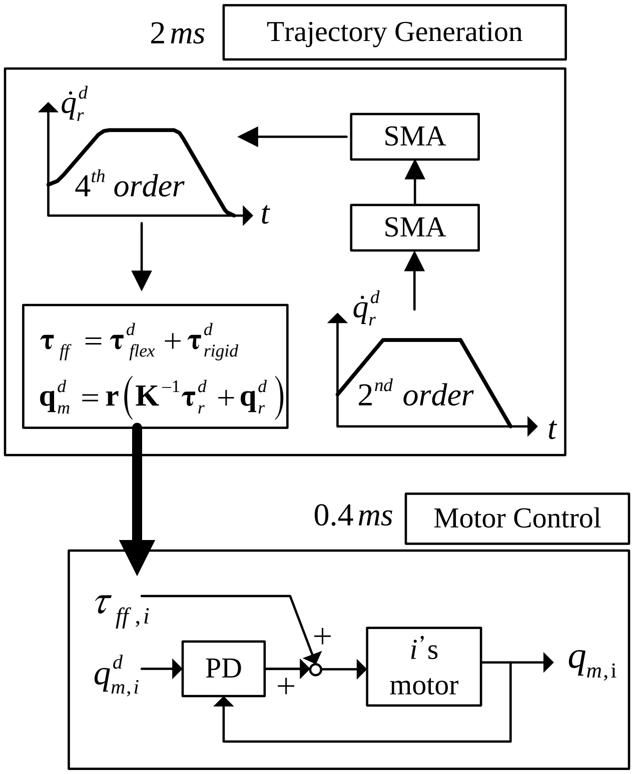

2 ms
Trajectory Generation
$\dot{q}_r^d$
4^{th} order
t
SMA
SMA
$\dot{q}_r^d$
2^{nd} order
t
$\boldsymbol{\tau}_{ff} = \boldsymbol{\tau}_{flex}^d + \boldsymbol{\tau}_{rigid}^d$
$\mathbf{q}_m^d = \mathbf{r}\left(\mathbf{K}^{-1}\boldsymbol{\tau}_r^d + \mathbf{q}_r^d\right)$
0.4 ms
Motor Control
$\tau_{ff,i}$
$q_{m,i}^d$
PD
+
+
i's motor
$q_{m,\mathrm{i}}$

Figure 5.16. The block diagram of the control architecture implemented in the Hi5a controller.

Module	Period of event	Load (time constraint)
Trajectory planner	random (≥ 6 ms)	$s_{k+1}(t)$: 4 DT + 5 JT (2 ms) $s_c(t)$: 4 DT + 5 JT (2 ms) $s_k(t)$: 4 DT (2 ms)
Trajectory generation/Feedforward control	2 ms	1 DT (2ms)

Table 5.2. Computational load of the proposed trajectory planner, where 1 DT $\triangleq$ the time for the controller to calculate the dynamic terms once, and 1 JT $\triangleq$ the time for the controller to calculate the Jacobian matrix once.

Conditions	Conventional scheme	Proposed scheme
Motion planner	Time-optimal TVP	Time-optimal TVP
Trajectory pre-filter	4 Hz BW	8 Hz BW
Feedforward control	Rigid-body dynamic model (3.6)	Flexible joint dynamic model (4.27)
Feedback control	Cascaded PD control (Figure 3.5)	

Table 5.3. Experiment conditions.

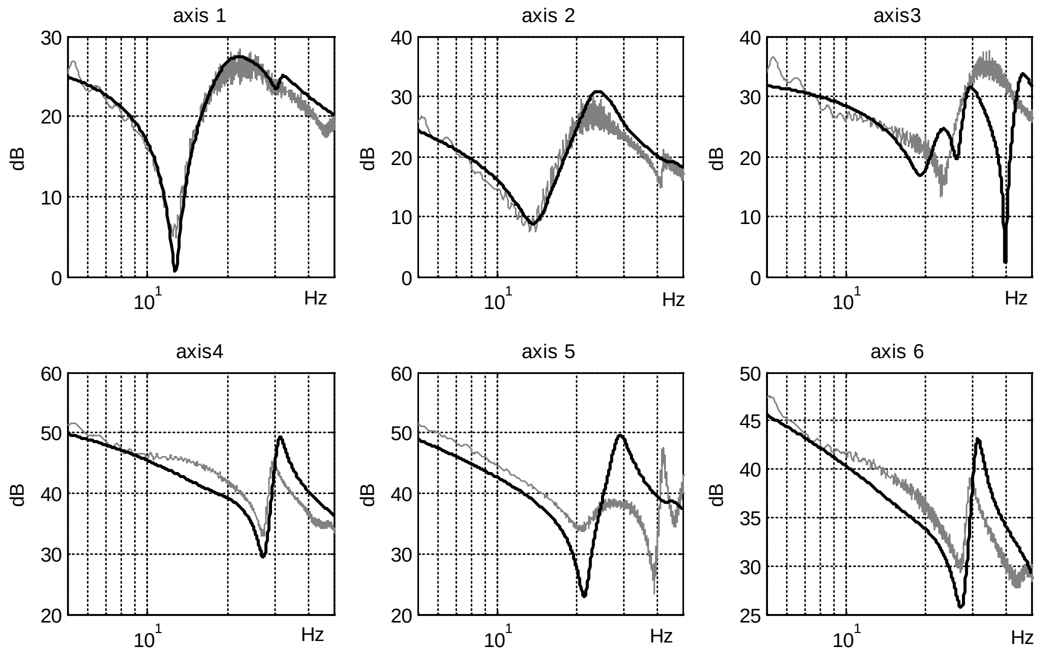
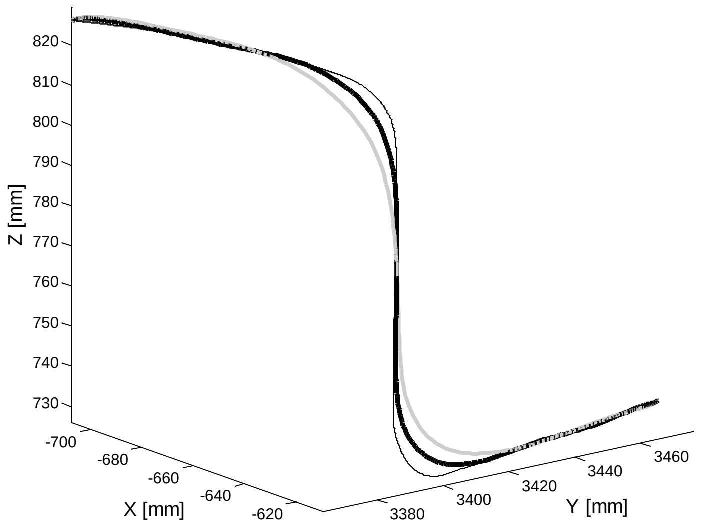
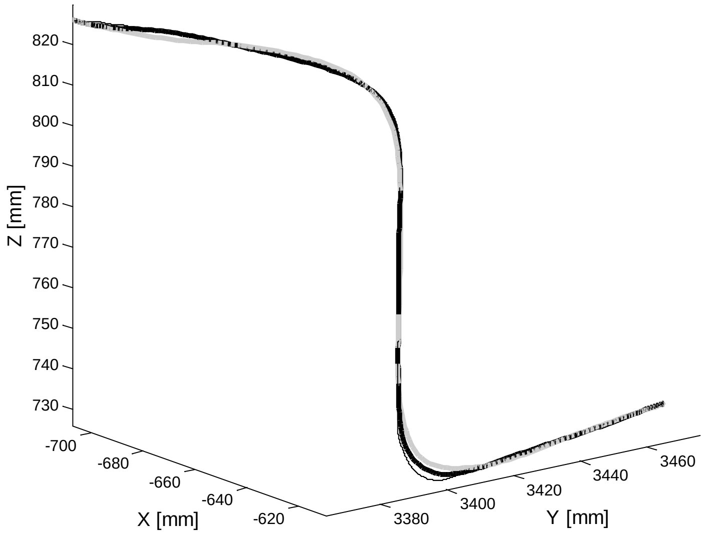


Figure 5.17. The flexible joint dynamic model was verified at the home position by the multivariable FRF for the motor torque as the input and the motor velocity as the output. FRF of motor torque and motor velocity (gray lines: measurement, black lines: model).



(a) Conventional scheme



(a) Proposed scheme

Figure 5.18. TCP paths for different speed (thin lines: speed 100 mm/s, thick black lines: speed 500 mm/s, gray lines: speed 1000 mm/s). The reference path is almost identical with the path generated at 100 mm/s.

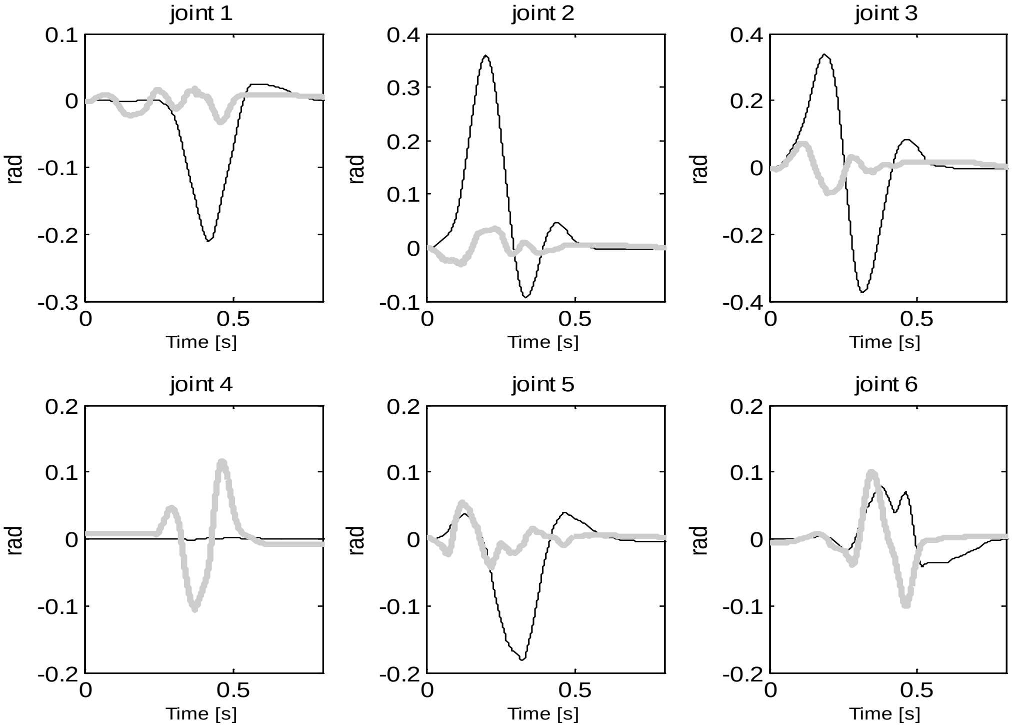


Figure 5.19. Tracking errors on the motor side for speed 1000 mm/s (black lines: conventional, gray lines: proposed).

5.6 Summary

Most existing online trajectory generators aim for good tracking performance by limiting the bandwidth of the reference inputs. Such strategy is not always acceptable in industry because in some robot applications, e.g., spot welding, handling, the cycle time is critical. A pragmatic approach for trajectory generation and control proposed in this chapter is to plan using a low order high-bandwidth trajectory and to increase the control bandwidth to compensate.

In this chapter, a method for online trajectory planning that is computationally efficient, time-optimal, and maintains path-following integrity is presented. The fundamental idea behind our approach is to combine the well-known offline trajectory planning along a specified path and the TVP trajectory that is widely used in industry. Utilizing the TVP trajectory allows one to simplify the problem of dealing with non-zero boundary conditions that inevitably appear when solving the problem of replanning on the fly. The resulting trajectory is near-time optimal in the dynamical sense but the number of calculations for dynamics is dramatically reduced compared with the offline method. Furthermore, the proposed method allows a smooth transition between any two parameterized paths with arbitrary boundary conditions.

In combination with this trajectory planning method, it was demonstrated that a nonlinear feedforward control based on the flexible joint dynamic model is a good candidate to increase the tracking performance if the joint elasticity limits the control bandwidth. The feedforward control can be easily combined with the PD controller widely utilized in industry. The experimental result confirms the validity of the scheme proposed in Chapter 3, i.e., the time-optimal TVP plus nonlinear control based on elastic model. However, as shown in Chapter 3, a full-state controller can achieve better performance than feedforward control for time-optimal trajectories due to active damping on the link side of the former. In the next chapter, a novel full-state controller is proposed to further improve the tracking performance for the time-optimal trajectory.

Chapter 6: A Singular Perturbation Method with Feedforward Control[4]

In Chapter 5, a time-optimal TVP along specified path is presented. The experimental result shows that the proposed time-optimal trajectory can fully satisfy the specification of the PITOM. In this chapter, a full-state controller is proposed to further improve the tracking performance of the proposed trajectory. Due to the time-optimal design, the proposed trajectory is very demanding. In the previous chapter, it is shown that the feedforward control based on elastic model combined with the traditional PD control is effective to track the proposed trajectory. However, as shown in Chapter 4, such control scheme can be further improved by employing full-state feedback, especially, the tracking on the link side. The full-state controller introduced in Chapter 4 is a decentralized scheme. Therefore, the performance is less effective if the dynamic coupling is significant such as the dynamic model (3.1). In this chapter, a novel full-state control method is proposed. The method is obtained by combing the feedforward control (De Luca, 2000) and singular perturbation control (SPC). It is assumed that the feedforward control is dominant during fast transients (coming from the time-optimal trajectory) and the SPC can quickly damp out the residual vibrations at steady state. Unlike the full-state controller discussed in Chapter 4, the proposed SPC can fully address the dynamic coupling as it is a multivariable full-state controller. Furthermore, the proposed SPC can place the closed-loop poles arbitrarily which lacks in the existing SPC.

[4] A version of this chapter has been submitted for publication in *IEEE Transactions on Control Systems Technology* (Kim and Croft. A practical approach for feedback linearization control for industrial robots with elastic joints).

This chapter is organized as follows. First, a linearized singular perturbation model is obtained by applying the feedforward control. The linear model is then utilized to investigate some of the fundamental properties of the conventional SPC. Next, two SPC methods are proposed. The first SPC is derived by explicit computation of the invariant manifold of the feedback system. The conditions under which this control guarantees stability are also discussed. The second SPC method is obtained by utilizing the full model. Such an approach results in a very simple way to calculate the feedback gain such that it can place the closed-loop poles at arbitrary locations. The experiment with a 6DOF commercial industrial robot (Figure 6.1) is carried out to validate this approach.

In Appendix A.1, brief introduction to SPC is provided. A more detailed discussion on the SPC can be found in Kokotovic *et al*. (1999).

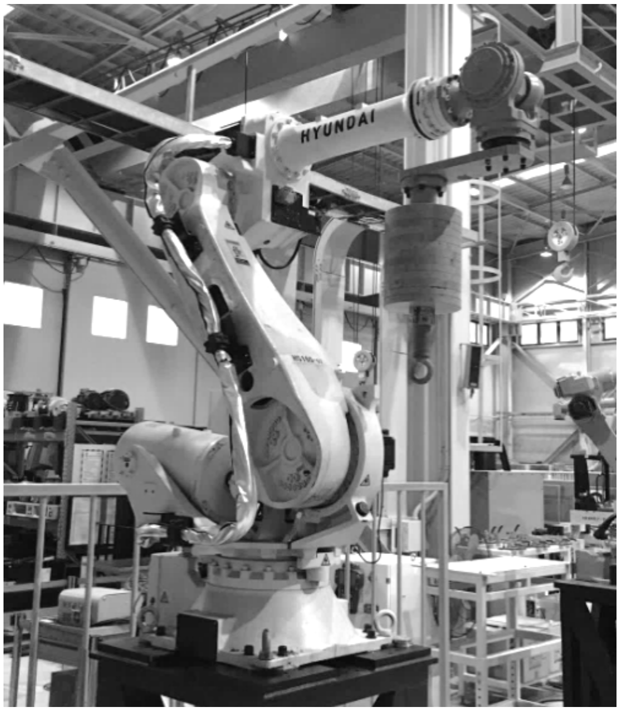


Figure 6.1. HS165 (Hyundai 6DOF industrial robot, maximum payload: 165 kg, main application: spot welding, http://www.hyundai-robotics.com/robot/robot01.asp).

6.1 Dynamic Model and Feedforward Control

By applying the feedforward control (4.27) to (3.1), the following error dynamics is obtained as:

$$\mathbf{M}_r\ddot{\mathbf{e}}_r + \mathbf{K}\left(\mathbf{e}_r - \mathbf{r}^{-1}\mathbf{e}_m\right) = \mathbf{0} \in \Re^n, \tag{6.1a}$$

$$\mathbf{M}_m\ddot{\mathbf{e}}_m + \mathbf{r}^{-1}\mathbf{K}\left(\mathbf{r}^{-1}\mathbf{e}_m - \mathbf{e}_r\right) + \mathbf{u}_{fb} = \mathbf{0} \in \Re^n. \tag{6.1b}$$

Let us assume that the feedforward control is dominant during transients. This assumption will be verified later by experiment. At the end of motion or steadystate, the system (6.1) can completely determine the system. At steady state, it is further assumed that the oscillation is small and the variation of the inertia can be ignored. If such assumption holds, then the system (6.1) is linear with constant coefficients. By defining $\boldsymbol{\mu} \triangleq \mathbf{K}\left(\mathbf{r}^{-1}\mathbf{e}_m - \mathbf{e}_r\right)$, the system (6.1) can be written as a singularly perturbed form

$$\ddot{\mathbf{e}}_r = \mathbf{M}_r^{-1}\boldsymbol{\mu} \tag{6.2a}$$

$$\varepsilon^2\ddot{\boldsymbol{\mu}} = -\bar{\mathbf{K}}\left(\mathbf{J}^{-1} + \mathbf{M}_r^{-1}\right)\boldsymbol{\mu} - \bar{\mathbf{K}}\mathbf{J}^{-1}\mathbf{v}\ , \tag{6.2b}$$

where $\mathbf{v} \triangleq \mathbf{r}\mathbf{u}_{fb}$, $\bar{\mathbf{K}} = \varepsilon^2\mathbf{K} = O(1)$, and $0 < \varepsilon \ll 1$: the perturbation parameter.

The system (6.2) can be converted into a standard state-space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \tag{6.3a}$$

$$\varepsilon\dot{\mathbf{z}} = \mathbf{C}\mathbf{z} + \mathbf{D}\mathbf{v}, \tag{6.3b}$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{e}_r & \dot{\mathbf{e}}_r \end{bmatrix}^T$, $\mathbf{z} = \begin{bmatrix} \boldsymbol{\mu} & \varepsilon\dot{\boldsymbol{\mu}} \end{bmatrix}^T$, $\mathbf{A} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\mathbf{B} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{M}_r^{-1} & \mathbf{0} \end{bmatrix}$, $\mathbf{C} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\bar{\mathbf{K}}\left(\mathbf{J}^{-1} + \mathbf{M}_r^{-1}\right) & \mathbf{0} \end{bmatrix}$, and $\mathbf{D} \triangleq \begin{bmatrix} \mathbf{0} \\ -\bar{\mathbf{K}}\mathbf{J}^{-1} \end{bmatrix}$.

Consider the boundary layer system

$$\varepsilon\dot{\boldsymbol{\eta}} = \varepsilon\dot{\mathbf{z}} - \varepsilon\dot{\boldsymbol{\varphi}}(\mathbf{x},\ \mathbf{v}_s(\mathbf{x})), \tag{6.4}$$

where M_ε : $\mathbf{z} = \boldsymbol{\varphi}(\mathbf{x}, \varepsilon)$ is the invariant manifold. $\boldsymbol{\eta} \triangleq \mathbf{z} - \boldsymbol{\varphi}(\mathbf{x}, \varepsilon)$ is the deviation of the fast variable, $\mathbf{z}$, from the invariant manifold. The term $\mathbf{v}_s(\mathbf{x})$ is the slow control in the composite feedback control, $\mathbf{v} = \mathbf{v}_f(\boldsymbol{\eta}) + \mathbf{v}_s(\mathbf{x})$.

In the fast time-scale, $\boldsymbol{\eta}' \triangleq \frac{d}{d\tau}\boldsymbol{\eta}$, where $\frac{1}{\varepsilon} \triangleq \frac{d\tau}{dt}$, Eq. (6.4) becomes

$$\boldsymbol{\eta}' = \mathbf{C}\boldsymbol{\eta} + \mathbf{D}\mathbf{v}_f, \tag{6.5}$$

By utilizing (6.3b), the equilibrium manifold, $M_{\varepsilon=0}$, is obtained as

$$\boldsymbol{\varphi}_0 = -\mathbf{C}^{-1}\mathbf{D}\mathbf{v}_s = \begin{bmatrix} -\mathbf{M}_r\left(\mathbf{J} + \mathbf{M}_r\right)^{-1}\mathbf{v}_s \\ \mathbf{0} \end{bmatrix}. \tag{6.6}$$

Therefore, the slow model defined in M_0 or $M_{\varepsilon=0}$ becomes

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\varphi}_0 = \begin{bmatrix} \dot{\mathbf{e}}_r \\ -\left(\mathbf{J} + \mathbf{M}_r\right)^{-1}\mathbf{v}_s \end{bmatrix}. \tag{6.7}$$

It can be observed that the slow model (6.7) is the rigid error dynamics on the link side since it is equivalent to

$$\left(\mathbf{J}+\mathbf{M}_r\right)\ddot{\mathbf{e}}_r+\mathbf{v}_s=\mathbf{0}. \tag{6.8}$$

6.2 Conventional Composite Control

6.2.1 Fast Control

Notice that the poles of the boundary layer (6.5) are located along the imaginary axis. In order to make the boundary layer system asymptotically stable, the fast input can be chosen as

$$\mathbf{v}_f=\mathbf{L}\boldsymbol{\eta}, \tag{6.9}$$

where $\mathbf{L}=\begin{bmatrix}\mathbf{L}_p & \mathbf{L}_d\end{bmatrix}$. $\mathbf{L}_p$ and $\mathbf{L}_d$ are diagonal and positive definite matrices, i.e., PD control.

However, implementation of the fast control (6.9) is challenging since $\boldsymbol{\eta}$ is not measurable directly. Alternately, it is chosen as

$$\mathbf{v}_f=\mathbf{L}\,\mathbf{z}\ . \tag{6.10}$$

One can verify that for the fast time scale, it holds $\boldsymbol{\eta}=\mathbf{z}$ for $\varepsilon=0$. Inserting (6.9) into (6.5), we obtain the closed-loop boundary layer system as

$$\boldsymbol{\eta}'=\overline{\mathbf{C}}\boldsymbol{\eta}, \tag{6.11}$$

where $\overline{\mathbf{C}}\triangleq\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\overline{\mathbf{K}}\left[\mathbf{J}^{-1}\left(\mathbf{I}+\mathbf{L}_p\right)+\mathbf{M}_r^{-1}\right] & -\overline{\mathbf{K}}\mathbf{J}^{-1}\mathbf{L}_d\end{bmatrix}$.

However, it is complex to choose the gain $\mathbf{L}$ such that the poles of (6.11) are located at some desired locations unless the inertial coupling is ignored.

6.2.2 Slow Control

For a non-zero $\mathbf{L}_p$ in the fast input, the original equilibrium manifold M_0 is modified into

$$\left[\mathbf{J}+\left(\mathbf{I}+\mathbf{L}_p\right)\mathbf{M}_r\right]\ddot{\mathbf{e}}_r+\mathbf{v}_s=\mathbf{0} \tag{6.12}$$

Then, the slow input is designed for (6.12) typically as a PD control with the gain

$$\mathbf{v}_s=\bar{\mathbf{K}}_s\mathbf{x} \tag{6.13}$$

where $\bar{\mathbf{K}}_s=\left[\mathbf{J}+\left(\mathbf{I}+\mathbf{L}_p\right)\mathbf{M}_r\right]\mathbf{K}_s$, $\mathbf{K}_s=\left[\mathbf{K}_p \quad \mathbf{K}_d\right]$, and $\mathbf{K}_p$ and $\mathbf{K}_d$ are constant diagonal matrices and positive definite. With (6.13), the closed-loop slow system becomes

$$\ddot{\mathbf{e}}_r+\mathbf{K}_d\dot{\mathbf{e}}_r+\mathbf{K}_p\mathbf{e}_r=\mathbf{0}\,. \tag{6.14}$$

6.2.3 Corrective Control

Since the slow input (6.13) is designed on M_0, the model (6.7), or $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\boldsymbol{\varphi}_0)$, has an error of $O(\varepsilon)$. The corrective control compensates for this error by adding some corrective terms to the slow input. In most cases, the corrective control assumes that the original M_0 (6.6) is not affected by the fast control. Therefore, only the case $\mathbf{L}_p=\mathbf{0}$ is considered. Now let us pursue the corrective control for $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\,\boldsymbol{\varphi}_0+\varepsilon\boldsymbol{\varphi}_1)$. To do so, choose $\boldsymbol{\varphi}=\boldsymbol{\varphi}_0+\varepsilon\boldsymbol{\varphi}_1$ and $\mathbf{v}_s=\mathbf{v}_0+\varepsilon\mathbf{v}_1$, then inserting them into the manifold condition (6.3b), we have

$$\varepsilon\left(\dot{\boldsymbol{\varphi}}_0+\varepsilon\dot{\boldsymbol{\varphi}}_1\right)=\mathbf{C}\left(\boldsymbol{\varphi}_0+\varepsilon\boldsymbol{\varphi}_1\right)+\mathbf{D}\left(\mathbf{v}_0+\varepsilon\mathbf{v}_1\right). \tag{6.15}$$

Since ε is non-zero, Eq. (6.15) must satisfy

$$\mathbf{C}\boldsymbol{\varphi}_0 + \mathbf{D}\mathbf{v}_0 = \mathbf{0} \tag{6.16}$$

$$\mathbf{D}\mathbf{v}_1 + \mathbf{C}\boldsymbol{\varphi}_1 - \dot{\boldsymbol{\varphi}}_0 = \mathbf{0}\,. \tag{6.17}$$

The condition (6.16) suggests that $\mathbf{v}_0$ be designed on M_0, whereas Eq. (6.17) is expanded by the power series,

$$\mathbf{D}\mathbf{v}_1 = \frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) - \mathbf{C}\left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}}\right)^{-1}_{z=\varphi_0} \frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0)\,. \tag{6.18}$$

Thus, $\mathbf{D}\mathbf{v}_1 = \mathbf{0}$ because $\frac{\partial \mathbf{g}}{\partial \mathbf{z}} = \mathbf{C}$ and it is non-singular. The result (6.18) implies

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon \boldsymbol{\varphi}_1)\,, \tag{6.19}$$

because

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon \boldsymbol{\varphi}_1) \\ &= \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\varphi}_0 + \varepsilon \mathbf{B}\mathbf{C}^{-1} \frac{\partial \boldsymbol{\varphi}_0}{\partial x}\left(\mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\varphi}_0\right)\,. \\ &= \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\varphi}_0 \ = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) \ (\because \mathbf{B}\mathbf{C}^{-2}\mathbf{D} = \mathbf{0}) \end{aligned} \tag{6.20}$$

It can be easily verified that the following is satisfied

$$C(\boldsymbol{\varphi}_0) \perp C(\boldsymbol{\varphi}_1)\,, \tag{6.21}$$

where $C(\bullet)$ is the column space of a matrix $\bullet$.

Therefore, the corrective control designed on $\boldsymbol{\varphi}_1$ cannot reach the equilibrium manifold, $\boldsymbol{\varphi}_0$, because the vector spaces spanned by each manifold are orthogonal to each other. Instead, the corrective control for the modeling error of $O(\varepsilon^2)$ is pursued, or

$$\mathbf{v}_s = \mathbf{v}_0 + \varepsilon^2 \mathbf{v}_2 . \tag{6.22}$$

Again, applying it to the manifold condition,

$$\varepsilon\left(\dot{\boldsymbol{\varphi}}_0 + \varepsilon\dot{\boldsymbol{\varphi}}_1\right) = \mathbf{C}\left(\boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1\right) + \mathbf{D}\left(\mathbf{v}_0 + \varepsilon^2\mathbf{v}_2\right) \tag{6.23}$$

The term with ε^2 satisfies

$$\begin{aligned} \mathbf{D}\mathbf{v}_2 &= \mathbf{C}^{-1}\ddot{\boldsymbol{\varphi}}_0 \\ &= \mathbf{C}^{-1}\frac{d}{dt}\left(\frac{\partial\boldsymbol{\varphi}_0}{\partial\mathbf{x}}\left(\mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\varphi}_0\right)\right) \\ &= \mathbf{C}^{-1}\frac{\partial\boldsymbol{\varphi}_0}{\partial\mathbf{x}}\left(\mathbf{A}\mathbf{x} + \mathbf{B}\frac{\partial\boldsymbol{\varphi}_0}{\partial\mathbf{x}}\right)\left(\mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\varphi}_0\right) \end{aligned} \quad . \tag{6.24}$$

Utilizing $\boldsymbol{\varphi}_0 = -\mathbf{C}^{-1}\mathbf{D}\mathbf{v}_0$ and $\mathbf{v}_0 = \bar{\mathbf{K}}_s\mathbf{x}$ yields

$$\frac{\partial\boldsymbol{\varphi}_0}{\partial\mathbf{x}} = -\mathbf{C}^{-1}\mathbf{D}\bar{\mathbf{K}}_s . \tag{6.25}$$

Eq. (6.24) is written as

$$\mathbf{D}\mathbf{v}_2 = -\mathbf{C}^{-2}\mathbf{D}\bar{\mathbf{K}}_s\left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{D}\bar{\mathbf{K}}_s\right)^2\mathbf{x} . \tag{6.26}$$

Therefore,

$$\mathbf{v}_2 = \mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{M}_r\left[\left((\mathbf{K}_d{}^2 - \mathbf{K}_p)\mathbf{K}_d - \mathbf{K}_d\mathbf{K}_p\right)\dot{\mathbf{e}}_r + \left((\mathbf{K}_d{}^2 - \mathbf{K}_p)\mathbf{K}_p\right)\mathbf{e}_r\right] \tag{6.27}$$

So far, we have calculated the corrective control (6.27) by solving the manifold condition with the standard form. However, it is questionable how much better the corrective control (6.27) compared with the $\mathbf{v}_s = \mathbf{v}_0$ because (6.27) can only compensate the modeling error of $O(\varepsilon^2)$ but not $O(\varepsilon)$. Therefore, it is necessary to perform an analysis on the validity of the corrective control (6.27). To do so, first the reduced model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon^2 \boldsymbol{\varphi}_2)$ is obtained and this model is compared with $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0)$ .Then, the two models are compared with the full-model.

In order to obtain the model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon^2 \boldsymbol{\varphi}_2)$, it is required to obtain $\boldsymbol{\varphi}_2$. After some lengthy manipulation, one can obtain

$$\boldsymbol{\varphi}_2 = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}}\right)^{-1}_{\mathbf{z}=\boldsymbol{\varphi}_0} \left(\frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} \boldsymbol{\varphi}_1 + \frac{\partial \boldsymbol{\varphi}_1}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) - \left.\frac{\partial^2 \mathbf{g}}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} {\boldsymbol{\varphi}_1}^2 \right) . \tag{6.28}$$

See Appendix A.2 for the detailed derivation for (6.28). Utilizing the following results

$$\begin{aligned} \boldsymbol{\varphi}_1 &= \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}}\right)^{-1}_{\mathbf{z}=\boldsymbol{\varphi}_0} \frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) \\ &= -\mathbf{C}^{-2}\mathbf{D}\bar{\mathbf{K}}_s \left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{D}\bar{\mathbf{K}}_s\right)\mathbf{x} \end{aligned} , \tag{6.29}$$

$$\frac{\partial \boldsymbol{\varphi}_1}{\partial \mathbf{x}} = -\mathbf{C}^{-2}\mathbf{D}\bar{\mathbf{K}}_s \left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{D}\bar{\mathbf{K}}_s\right), \tag{6.30}$$

we obtain

$$\boldsymbol{\varphi}_2 = \left[\mathbf{C}^{-2}\mathbf{D}\bar{\mathbf{K}}_s\mathbf{B}\mathbf{C}^{-2}\mathbf{D}\bar{\mathbf{K}}_s - \mathbf{C}^{-3}\mathbf{D}\bar{\mathbf{K}}_s\left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{D}\bar{\mathbf{K}}_s\right)\right]\left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{D}\bar{\mathbf{K}}_s\right)\mathbf{x} . \tag{6.31}$$

It can be verified that $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_2$ share the same subspace,

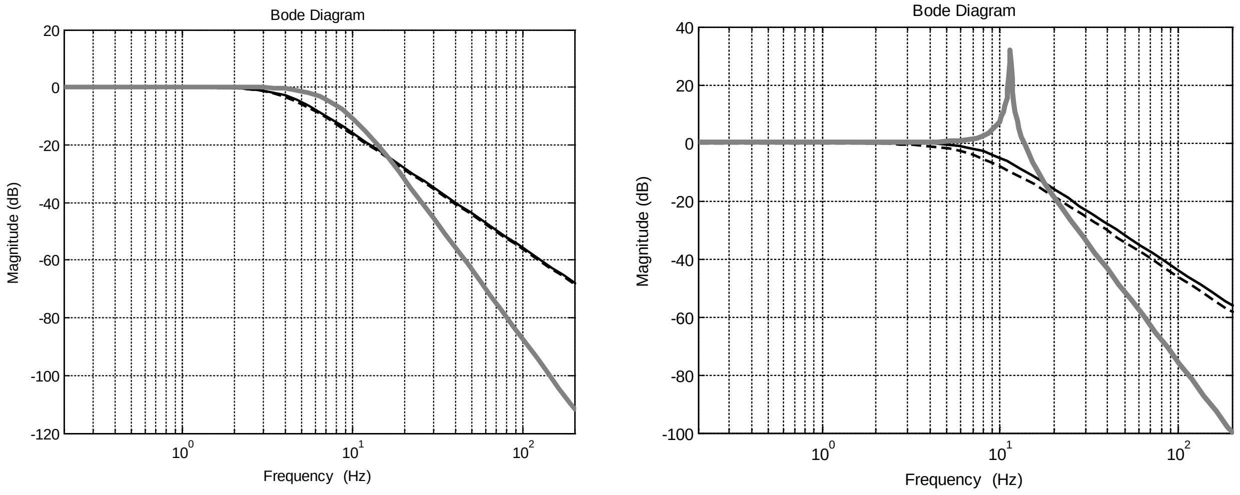
$$C(\boldsymbol{\varphi}_0) = C(\boldsymbol{\varphi}_2)\,. \tag{6.32}$$

As an example, let us consider a single-DOF case, where the model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0 + \varepsilon^2 \boldsymbol{\varphi}_2)$ is expressed as

$$\ddot{e}_r + \left[1 + \varepsilon^2 \frac{JJ_r}{J+J_r}\left(2k_p - k_d{}^2\right)\right] k_d \dot{e}_r + \left[1 + \varepsilon^2 \frac{JJ_r}{J+J_r}\left(k_p - k_d{}^2\right)\right] k_p e_r = 0\,, \tag{6.33}$$

where $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 & 0 \\ -\dfrac{1}{J_r} & 0 \end{bmatrix}$ $\mathbf{C} = \begin{bmatrix} 0 & 1 \\ -\dfrac{1}{J_r} - \dfrac{1}{J} & 0 \end{bmatrix}$ $\mathbf{D} = \begin{bmatrix} 0 \\ -\dfrac{1}{J} \end{bmatrix}$, $\mathbf{K}_s = \begin{bmatrix} k_p & k_d \end{bmatrix}$.

In Figure 6.2, the three models $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0)$, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0 + \varepsilon^2 \boldsymbol{\varphi}_2)$, and the full model are compared in the frequency domain with the model parameter obtained from the first joint of the industrial robot shown in Figure 6.1. The reduced models are close to the full model only at low frequency and the modeling errors increases dramatically as the frequency increases. This explains why the conventional corrective control cannot cope with a time-optimal trajectory. The time-optimal trajectory contains high frequency content, which increases the modeling error of the reduced models and potentially generates unstable motion. Second, the frequency responses of the first two models are nearly the same with only small deviation at high frequency. One can check the locations of the poles are almost identical between the two models. Also notice that even if the gain is chosen such that the reduced model is stable, the full model could exhibit a resonance phenomenon (Fig. 6.2(b)). Therefore, it can be concluded that the corrective control (6.27) has almost no effect because the two models, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0)$ and $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0 + \varepsilon^2 \boldsymbol{\varphi}_2)$ are nearly the same. In addition, neither of the reduced models can be used for time-optimal trajectories due to the significant modeling error at high frequency.



(a) $\xi = 0.7$, $\omega_s = 4\,Hz$ (b) $\xi = 0.7$, $\omega_s = 8\,Hz$

Figure 6.2. Comparison of the reduced models ($\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0)$: black solid lines, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \boldsymbol{\varphi}_0 + \varepsilon^2 \boldsymbol{\varphi}_2)$: dotted lines, full model: gray lines). It is assumed that $l_d = 2\zeta J \omega_f$, where $\omega_f = \sqrt{\dfrac{1}{J} + \dfrac{1}{J_r}}$, and $l_p = 0$. The figures are plotted for the characteristic equation of the slow system, $s^2 + 2\zeta\omega_s s + \omega_s{}^2 = 0$ with the model parameters, $J = 407\, kg\, m^2$, $k = 3 \times 10^6 \; {}^{Nm}\!/\!_{rad}$.

6.3 Proposed Composite Control

The significant modeling error of the conventional composite control is the property (6.21). From (6.20), it is sufficient to satisfy to following condition in order for the first order correction $\varepsilon \boldsymbol{\varphi}_1$ to be non-zero. That is,

$$\mathbf{B}\mathbf{C}^{-1}\mathbf{D} \neq \mathbf{0} \;\rightarrow\; \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) \neq \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon \boldsymbol{\varphi}_1)\,. \tag{6.34}$$

6.3.1 Decoupling Control for the Fast Subsystem

The simplest choice to satisfy $\mathbf{B}\mathbf{C}^{-1}\mathbf{D} \neq \mathbf{0}$ is that the matrix $\mathbf{C}$ is modified by the fast control. However, the PD control (6.10) results in a coupled closed-loop system, complicating the gain calculation. Therefore, the proposed method applies decoupling control to the boundary layer. To do so, we rewrite the boundary layer (6.5) as

$$\mathbf{v}_f = -\mathbf{J}\bar{\mathbf{K}}^{-1}\boldsymbol{\mu}'' - \mathbf{J}\left(\mathbf{J}^{-1} + \mathbf{M}_r^{-1}\right)\boldsymbol{\mu}. \tag{6.35}$$

Consider the following control law

$$\mathbf{v}_f = -\mathbf{J}\bar{\mathbf{K}}^{-1}\boldsymbol{\nu} - \mathbf{J}\left(\mathbf{J}^{-1} + \mathbf{M}_r^{-1}\right)\boldsymbol{\mu}, \tag{6.36}$$

where $\boldsymbol{\nu}$ is a new control yet to be determined. Comparing (6.35) with (6.36) implies that $\boldsymbol{\nu} = \boldsymbol{\mu}''$. Choosing

$$\boldsymbol{\nu} = -\mathbf{L}_d\boldsymbol{\mu}' - \mathbf{L}_p\boldsymbol{\mu}, \tag{6.37}$$

the closed-loop boundary layer system is obtained as

$$\boldsymbol{\mu}'' + \mathbf{L}_d\boldsymbol{\mu}' + \mathbf{L}_p\boldsymbol{\mu} = \mathbf{0}, \tag{6.38}$$

which is a decoupled system. An explicit form of the fast input, $\mathbf{v}_f$, is expressed as

$$\mathbf{v}_f = \mathbf{L}_f\,\mathbf{z}, \tag{6.39}$$

where $\mathbf{L}_f = \left[\mathbf{J}\left(\bar{\mathbf{K}}^{-1}\mathbf{L}_p - \mathbf{J}^{-1} - \mathbf{M}_r^{-1}\right) \quad \mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{L}_d\right]$. By inserting (6.39) into (6.3b), we obtain

$$\varepsilon\dot{\mathbf{z}} = \bar{\mathbf{C}}\,\mathbf{z} + \mathbf{D}\mathbf{v}_s, \tag{6.40}$$

where $\bar{\mathbf{C}} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{L}_p & -\mathbf{L}_d \end{bmatrix}$.

The following terms are necessary in the derivation of the reduced model.

$$\bar{\mathbf{C}}^{-1} = \begin{bmatrix} -\mathbf{L}_p\mathbf{L}_d & -\mathbf{L}_p \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{6.41}$$

$$\bar{\mathbf{C}}^{-2} = \mathbf{L}_p \begin{bmatrix} \mathbf{L}_d\mathbf{L}_p\mathbf{L}_d - \mathbf{I} & \mathbf{L}_d\mathbf{L}_p \\ -\mathbf{L}_d & -\mathbf{I} \end{bmatrix} \tag{6.42}$$

$$\mathbf{B}\bar{\mathbf{C}}^{-1}\mathbf{D} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_r^{-1}\mathbf{L}_p^{-1}\bar{\mathbf{K}}\,\mathbf{J}^{-1} \end{bmatrix} \neq \mathbf{0} \tag{6.43}$$

With the fast control (6.40), we have another singularly perturbed system,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \tag{6.44a}$$

$$\varepsilon\dot{\mathbf{z}} = \bar{\mathbf{C}}\,\mathbf{z} + \mathbf{D}\mathbf{v}_s\,, \tag{6.44b}$$

with the equilibrium manifold, defined by

$$\boldsymbol{\varphi}_0 = -\bar{\mathbf{C}}^{-1}\mathbf{D}\,\mathbf{v}_s \tag{6.45}$$

and the reduced model on M_0 ,

$$\ddot{\mathbf{e}}_r + \mathbf{M}_r^{-1}\mathbf{L}_p^{-1}\bar{\mathbf{K}}\mathbf{J}^{-1}\mathbf{v}_s = \mathbf{0}\,. \tag{6.46}$$

The model (6.46) suggests that the slow input be chosen as

$$\mathbf{v}_s = \bar{\mathbf{K}}_s\mathbf{x}\,, \tag{6.47}$$

where $\bar{\mathbf{K}}_s \triangleq \mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{L}_p\mathbf{M}_r\mathbf{K}_s$, so that the closed-loop slow system becomes (6.14). By the result in (6.43), it is guaranteed that there exists a reduced model with error of only $O(\varepsilon^2)$, or $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1)$. Applying (6.47) to (6.45), we have

$$\boldsymbol{\varphi}_0 = \begin{bmatrix} -\mathbf{M}_r\mathbf{K}_p & -\mathbf{M}_r\mathbf{K}_d \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{x}. \tag{6.48}$$

As in (6.29), one obtains

$$\boldsymbol{\varphi}_1 = -\bar{\mathbf{C}}^{-2}\mathbf{D}\bar{\mathbf{K}}_s\left(\mathbf{A} - \mathbf{B}\bar{\mathbf{C}}^{-1}\mathbf{D}\bar{\mathbf{K}}_s\right)\mathbf{x} \tag{6.49}$$

Utilizing (6.41), (6.42), and (6.43),

$$\bar{\mathbf{C}}^{-2}\mathbf{D}\bar{\mathbf{K}}_s = \begin{bmatrix} -\mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\mathbf{K}_p & -\mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\mathbf{K}_d \\ \mathbf{M}_r\mathbf{K}_p & \mathbf{M}_r\mathbf{K}_d \end{bmatrix} \tag{6.50}$$

$$\mathbf{B}\bar{\mathbf{C}}^{-1}\mathbf{D}\bar{\mathbf{K}}_s = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{K}_p & \mathbf{K}_d \end{bmatrix} \tag{6.51}$$

Therefore,

$$\boldsymbol{\varphi}_1 = \begin{bmatrix} -\mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\mathbf{K}_d\mathbf{K}_p & \mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\mathbf{K}_p - \mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\mathbf{K}_d^2 \\ \mathbf{M}_r\mathbf{K}_d\mathbf{K}_p & -\mathbf{M}_r\mathbf{K}_p + \mathbf{M}_r\mathbf{K}_d^2 \end{bmatrix}\mathbf{x} \tag{6.52}$$

The first order approximation of the invariant manifold, M_e, can be assumed in the form

$$\boldsymbol{\varphi} = \boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1 = \begin{bmatrix} \boldsymbol{\gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}_{21} & \boldsymbol{\gamma}_{22} \end{bmatrix}\mathbf{x}. \tag{6.53}$$

Therefore, the following reduced model can be expressed as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1) \\ &= \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{M}_r^{-1}\boldsymbol{\gamma}_{11}\mathbf{x}_1 + \mathbf{M}_r^{-1}\boldsymbol{\gamma}_{12}\mathbf{x}_2 \end{bmatrix}\end{aligned} \tag{6.54}$$

More explicitly,

$$\ddot{\mathbf{e}}_r + \left[\mathbf{K}_d + \varepsilon\mathbf{M}_r^{-1}\mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\left(\mathbf{K}_d^2 - \mathbf{K}_p\right)\right]\dot{\mathbf{e}}_r + \left[\mathbf{K}_p + \varepsilon\mathbf{M}_r^{-1}\mathbf{L}_p^{-1}\mathbf{L}_d\mathbf{M}_r\mathbf{K}_d\mathbf{K}_p\right]\mathbf{e}_r = \mathbf{0} \in \Re^n \tag{6.55}$$

It should be noted that the error of the reduced model (6.55) is $O(\varepsilon^2)$ without a corrective control. The input of the model is assumed as a PD control (6.47), which requires only position and velocity feedback on the link side. Thus, it is not necessary to take derivatives on the input as it is usually done in the conventional SPC.

6.3.2 Stability of the Composite Control Law

In this section, the condition that the system (6.44), modified singularly perturbed system by the decoupling fast control, is stable is discussed. With the control laws (6.39) and (6.47), the closed-loop system is represented in separate time-scales.

$$\ddot{\mathbf{e}}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{K}_p\mathbf{e}_r = \mathbf{0} \tag{6.56a}$$

$$\boldsymbol{\mu}'' + \mathbf{L}_d\boldsymbol{\mu}' + \mathbf{L}_p\boldsymbol{\mu} = \mathbf{0} \tag{6.56b}$$

It is assumed that the system (6.56) can be expressed in the following form.

$$\ddot{\mathbf{e}}_r + 2\zeta\omega_s\dot{\mathbf{e}}_r + \omega_s{}^2\mathbf{e}_r = \mathbf{0} \tag{6.57a}$$

$$\boldsymbol{\mu}'' + 2\zeta\omega_f \boldsymbol{\mu}' + {\omega_f}^2 \boldsymbol{\mu} = \mathbf{0} \tag{6.57b}$$

where ω_s is the natural frequency of the slow system in the slow time scale, t, and ω_f is the natural frequency of the fast system in the fast time scale, τ, and $0 < \xi \le 1$ is damping ratio. Eq. (6.57b) can be expressed in the slow time scale, t as $\ddot{\boldsymbol{\mu}} + 2\zeta\left(a\omega_s\right)\dot{\boldsymbol{\mu}} + \left(a\omega_s\right)^2 \boldsymbol{\mu} = \mathbf{0}$, where $a > 0$ represents how much the fast system is faster than the slow system.

Now let us consider the error dynamics of the full model (6.2). Writing (6.2) in terms of the link error, we have

$$\varepsilon^2 \mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{M}_r \overset{(4)}{\mathbf{e}_r} + \left(\mathbf{J} + \mathbf{M}_r\right)\ddot{\mathbf{e}}_r + \mathbf{v} = \mathbf{0}. \tag{6.58}$$

The composite control law is given by

$$\mathbf{v} = \mathbf{v}_s + \mathbf{v}_f = \mathbf{L}_f \mathbf{z} + \bar{\mathbf{K}}_s \mathbf{x}. \tag{6.59}$$

Utilizing the fact $\boldsymbol{\mu} = \mathbf{M}_r \ddot{\mathbf{e}}_r$ and $\dot{\boldsymbol{\mu}} = \mathbf{M}_r \dddot{\mathbf{e}}_r$, the input can also be written in terms of the link error,

$$\mathbf{v} = \varepsilon \mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{L}_d \mathbf{M}_r \overset{(3)}{\mathbf{e}_r} + \left(\mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{L}_p \mathbf{M}_r - \mathbf{J} - \mathbf{M}_r\right)\ddot{\mathbf{e}}_r + \mathbf{J}\bar{\mathbf{K}}^{-1}\mathbf{L}_p \mathbf{M}_r \left(\mathbf{K}_d \dot{\mathbf{e}}_r + \mathbf{K}_p \mathbf{e}_r\right). \tag{6.60}$$

By inserting (6.60) into (6.58), the error dynamics of the full model becomes

$$\varepsilon^2 \mathbf{M}_r \overset{(4)}{\mathbf{e}_r} + \varepsilon \mathbf{L}_d \mathbf{M}_r \overset{(3)}{\mathbf{e}_r} + \mathbf{L}_p \mathbf{M}_r \left(\ddot{\mathbf{e}}_r + \mathbf{K}_d \dot{\mathbf{e}}_r + \mathbf{K}_p \mathbf{e}_r\right) = \mathbf{0} \tag{6.61}$$

Let us choose the gain as

$$\mathbf{L}_d = l_d \mathbf{I}, \quad \mathbf{L}_p = l_p \mathbf{I}, \tag{6.62a}$$

$$\mathbf{K}_d = k_d\mathbf{I}\,,\ \mathbf{K}_p = k_p\mathbf{I}\ , \tag{6.62b}$$

Then, the full model (6.61) becomes decoupled and can be written in the Laplace domain as

$$s^4 + \varepsilon^{-1} l_d s^3 + \varepsilon^{-2} l_p s^2 + \varepsilon^{-2} l_p k_d s + \varepsilon^{-2} l_p k_p = 0\,. \tag{6.63}$$

To satisfy (6.57), choose, $l_p = \varepsilon^2 a^2 \omega_s{}^2$, $l_d = \varepsilon 2\zeta a\,\omega_s$, $k_p = \omega_s{}^2$, and $k_d = 2\zeta\omega_s$.

The Routh-Hurwitz table for (6.63) is shown below.

$$\begin{array}{llll}
s^4 & 1 & \omega_s{}^2 & a^2\omega_s{}^4 \\
s^3 & 2\zeta a\,\omega_s & 2\zeta a^2\omega_s{}^3 & \\
s^2 & a(a-1)\omega_s{}^2 & a^2\omega_s{}^4 & \\
s^1 & \dfrac{2\zeta a^2\omega_s{}^3(a-2)}{a-1} & & \\
s^0 & a^2\omega_s{}^4 & &
\end{array}$$

Therefore, the system (6.63) is stable only if

$$a > 2\,. \tag{6.64}$$

Note that the system (6.57) can be regarded as a generalized closed-loop system of a singularly perturbed system. Therefore, the same conclusion can be obtained as long as the closed-loop system is assumed to be in the form (6.57). The constant, a, represents how much the boundary layer should be faster than the quasi-steady state. The condition (6.64) states that *the boundary layer must be at least twice as fast as the quasi-steady state for stability*. Note that the stability condition (6.64) is true regardless of the perturbation parameter, ε (at least at steady state) as the closed-loop system (6.63) has become independent of ε by the proposed gain selection.

6.3.3 Gain Selection

In this section, simple yet practical methods to compute the gain (6.62) are presented. The first method is based on the reduced model (6.55) with stability condition (6.64) whereas the pole-placement technique is applied to the second method utilizing the full model.

6.3.3.1 Reduced Model-Based Scheme

By applying (6.62) to (6.38) and (6.55), the two subsystems can be written as

$$\ddot{\mathbf{e}}_r + \left[k_d + \frac{2\zeta}{a\omega_n}\left(k_d^2 - k_p\right) \right]\dot{\mathbf{e}}_r + \left[k_p + \frac{2\zeta}{a\omega_n} k_d k_p \right]\mathbf{e}_r = \mathbf{0} \tag{6.65a}$$

$$\boldsymbol{\mu}'' + l_d \boldsymbol{\mu}' + l_p \boldsymbol{\mu} = \mathbf{0}\,. \tag{6.65b}$$

The goal of tuning the gain is to make the system (6.65) to be equal to (6.57). Assuming that a is chosen according to (6.64), the gain for the slow input, k_p and k_d, is calculated as

$$k_p = \frac{a\,\omega_s^{\;3}}{a\,\omega_s + 2\zeta k_d} \tag{6.66a}$$

$$k_d = \frac{\beta^{1/3}}{6} - \frac{a\,\omega_s}{3\zeta} + \frac{a\,\omega_s^{\;2}\left(12\zeta^2 + a\right)}{6\beta^{1/3}\zeta^2}\ , \tag{6.66b}$$

where $\beta \triangleq 6\sqrt{3\alpha} + \frac{a\,\omega_s^{\;3}}{\zeta^3}\left[a^2 + 18\left(a+3\right)\zeta^2\right] > 0$, and

$$\alpha \triangleq \frac{a^2\omega_s{}^6}{\xi^4}\left[\left(1-\xi^2\right)a^2+2\xi^2\left(9-8\xi^2\right)a+27\xi^2\right]>0\,.$$

The gain (6.66) makes the closed-loop poles of the reduced model, $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\boldsymbol{\varphi}_0+\varepsilon\boldsymbol{\varphi}_1)$, in M_ε, equal to (6.57a). However, it is interesting to investigate whether the system, $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\boldsymbol{\varphi}_0)$ in M_0 with the gain (6.66) will also be stable. To do so, it suffices to show that the gain is strictly positive, i.e., $k_p>0$ and $k_d>0$. First, let us check for k_d. It holds

$$k_d>0 \;\Leftrightarrow\; 6\beta^{1/3}\zeta^2k_d>0\,. \tag{6.67}$$

Multiply $6\beta^{1/3}\zeta^2$ to (6.66b) to obtain

$$\begin{aligned}6\beta^{1/3}\zeta^2k_d &= \zeta^2\beta^{2/3}-2\zeta a\,\omega_s\beta^{1/3}+a\,\omega_s{}^2\left(12\zeta^2+a\right)\\ &= \zeta\beta^{1/3}\left(\zeta\beta^{1/3}-2a\,\omega_s\right)+a^2\omega_s{}^2+12\zeta^2a\,\omega_s{}^2\end{aligned}\,. \tag{6.68}$$

It is trivial to show that

$$\beta^{1/3}>\frac{a\,\omega_s}{\zeta}\,. \tag{6.69}$$

Applying (6.69) to (6.68),

$$\begin{aligned}&6\beta^{1/3}\zeta^2k_d > a\,\omega_s\left(a\,\omega_s-2a\,\omega_s\right)+a^2\omega_s{}^2+12\zeta^2a\,\omega_s{}^2\\ &\to 6\beta^{1/3}\zeta^2k_d > 12\zeta^2a\,\omega_s{}^2\\ &\to 6\beta^{1/3}\zeta^2k_d > 0\\ &\to k_d > 0\end{aligned} \tag{6.70}$$

By (6.66a), it holds $k_d>0 \to k_p>0$.

In Figure 6.3, the reduced model (6.55) is compared with the full model with the same model parameters in Figure 6.2. Notice that the error of the reduced model is reduced compared with the previous case. The unstable behavior of the full model in the previous case is not observed in Figure 6.3.

6.3.3.2 Full Model-Based Scheme

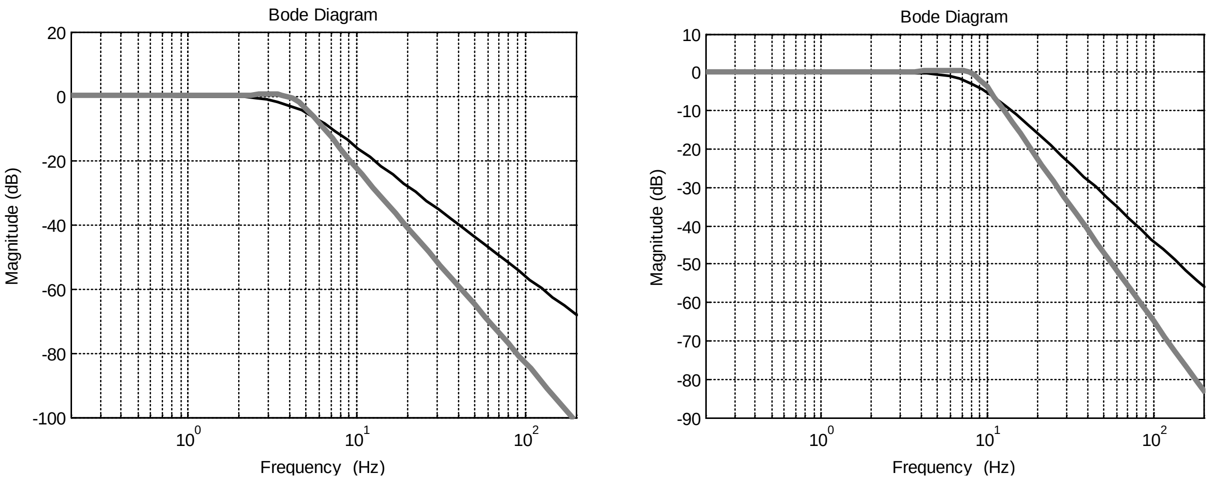
Eq. (6.63) is the characteristic equation of the full model if the composite control law (6.59) is applied as long as the gain satisfies (6.62). Suppose that the desired closed-loop performance of (6.63) is described by

$$s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0 = 0, \tag{6.71}$$

where $c_i > 0$. Then, we obtain the unique set of gain

$$l_d = \varepsilon c_3,\ l_p = \varepsilon^2 c_2,\ k_d = \frac{\varepsilon^2 c_1}{l_p} = \frac{c_1}{c_2},\ k_p = \frac{\varepsilon^2 c_0}{l_p} = \frac{c_0}{c_2}. \tag{6.72}$$

Notice that all the gain is constant. Furthermore, they are all strictly positive, implying that the gain is also valid in the sense of separate time-scales (6.56). Even though the result obtained in this section is very simple, in fact it is a rather significant. That is, it has been shown that the closed-loop poles of the full model can be placed arbitrarily with only position and velocity feedback. It is emphasized again that the existing control methods using the full model typically require feedback up to jerk. Unless the inertial coupling is neglected, it is not trivial to obtain a control law that can achieve an arbitrary performance with the full model (Miyazaki *et al*., 2003). Such attempts will typically result in a coupled term with the motor inertia, $\mathbf{J}$, on the link acceleration. The key part of eliminating $\mathbf{J}$ is the fast input, or decoupling control applied to the fast subsystem, which enables us to obtain the closed-loop system (6.61) without $\mathbf{J}$.



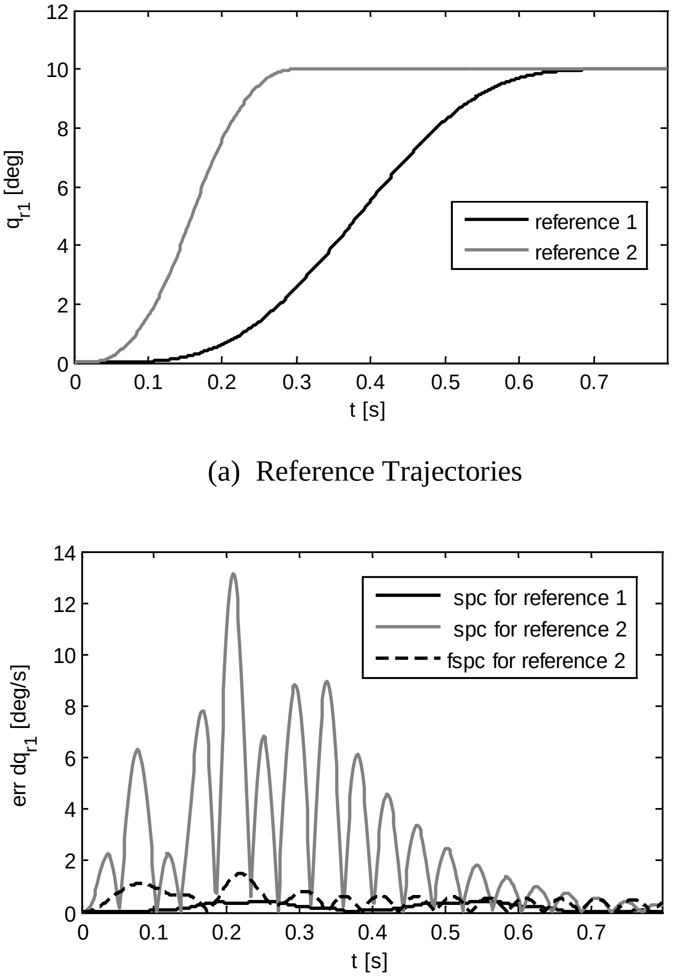
(a) $\xi = 0.7$, $\omega_s = 4Hz$

(b) $\xi = 0.7$, $\omega_s = 8Hz$

Figure 6.3. Comparison of the reduced model (black lines: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1)$, gray lines : the full model).

6.4 Simulation

In Figure 6.4, the simulation result is shown. The conventional SPC (Spong, 1987) is assumed with/without feedforward control. The same model parameter is assumed as in Figure 6.2 and Figure 6.3. For both controllers, the gain is chosen according to (6.64). Two reference trajectories are used. One is a slowly varying trajectory (reference 1) and the other is a fast varying trajectory similar to the time-optimal trajectory used in industry (reference 2). The reference 1 is obtained by adding a large low-pass filter to reference 2. To avoid a perfect feedforward control, a parameter uncertainty of 10 % on the inertia and joint stiffness is assumed. As shown in the figure, the conventional method generates a large tracking error for a time-optimal trajectory. The proposed control generates much less tracking error during transient. But the residual vibrations at steady state are similar in both controllers for the fast varying trajectory.

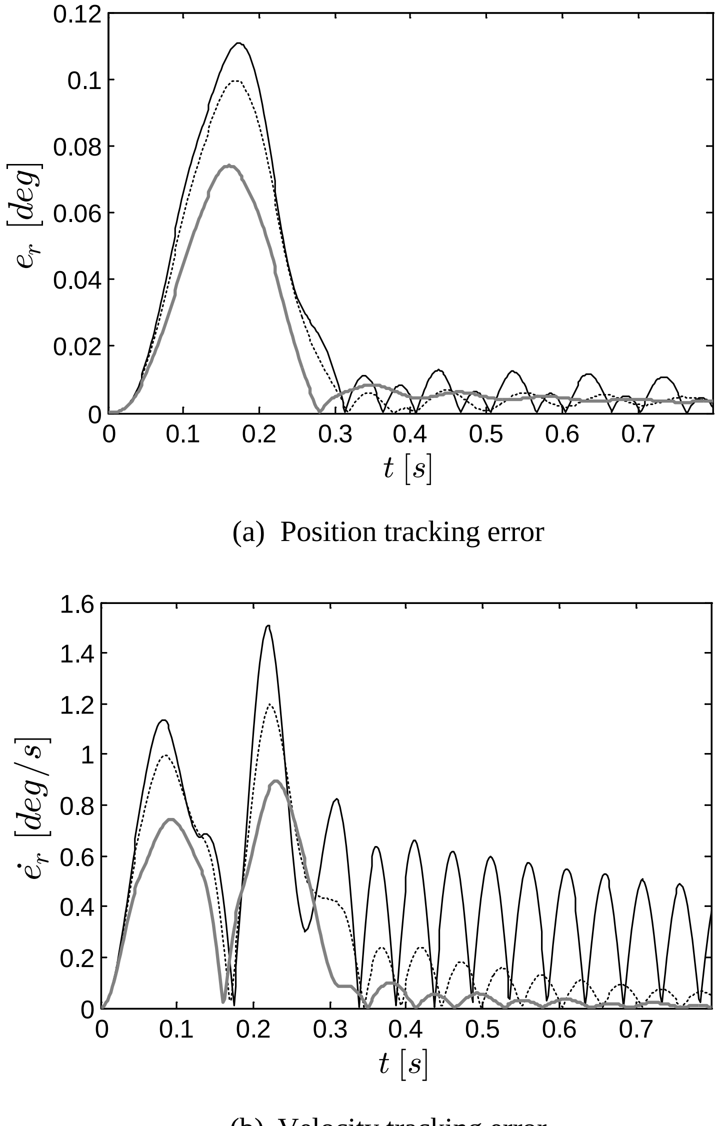


(a) Reference Trajectories

(b) Tracking error at the link side

Figure 6.4. The effect of feedforward control with the SPC ($\zeta = 0.7$, $\omega_n = 8\,Hz$, $a = 2.5$).

Next, the proposed control method described in Section 6.2, referred to as *the feedforward-SPC (f-SPC)* is evaluated for the reference 2. Three cases are tested for different methods of calculating the gain. As shown in Figure 6.5, the largest error occurs for the case when the gain is calculated based on the reduced model, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0)$. A much better performance is obtained when the gain is tuned based on the model, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1)$. The best performance is achieved when the gain is tuned utilizing the full model.



(a) Position tracking error

(b) Velocity tracking error

Figure 6.5. Comparison of tracking errors depending upon gains, where black solid line: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0)$, dotted-line : $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon\boldsymbol{\varphi}_1)$, and gray line: the full-model ($\zeta = 0.7$, $\omega_n = 8Hz$, $a = 2.5$).

6.5 Experiment

The experiment was performed on a large industrial robot (Figure 6.1) as the perturbation parameter could be large at a configuration with high inertia. The proposed control method was implemented in the Hi5a controller (Figure 1.1, right). Because the robot has only measurement on the motor side, a nonlinear observer (Jankovic, 1995; Khalil 2014) was implemented in the controller. At this time, the observer was applied to the first three joints. For comparison, a total of five different control methods were implemented (Table 6.1). Controller 1 is the conventional *f-SPC* as described in the previous section. Controller 2 and 3 are the proposed controllers discussed in Section 6.2. Controller 4 is the default control method of the Hi5a controller, a cascaded PD controller with the position and velocity feedback on the motor side with feedforward control based on rigid model (similar to the computed torque controller). Controller 5 is that same as Controller 4 except that the feedforward input was designed based on the flexible joint model.

The control methods were tested in the Hardware-in-the-loop (HIL) simulation before running the experiment. The controller was connected to a real-time PC where the flexible joint dynamic model was simulated by the 4th-order Runge-Kutta method at 2.5 kHz. The simulation was successful for all the control methods except for Controller 1. Controller 1 became unstable immediately after the control action was initiated. Even in case when the gain is very small (< 1 Hz), the system has become unstable.

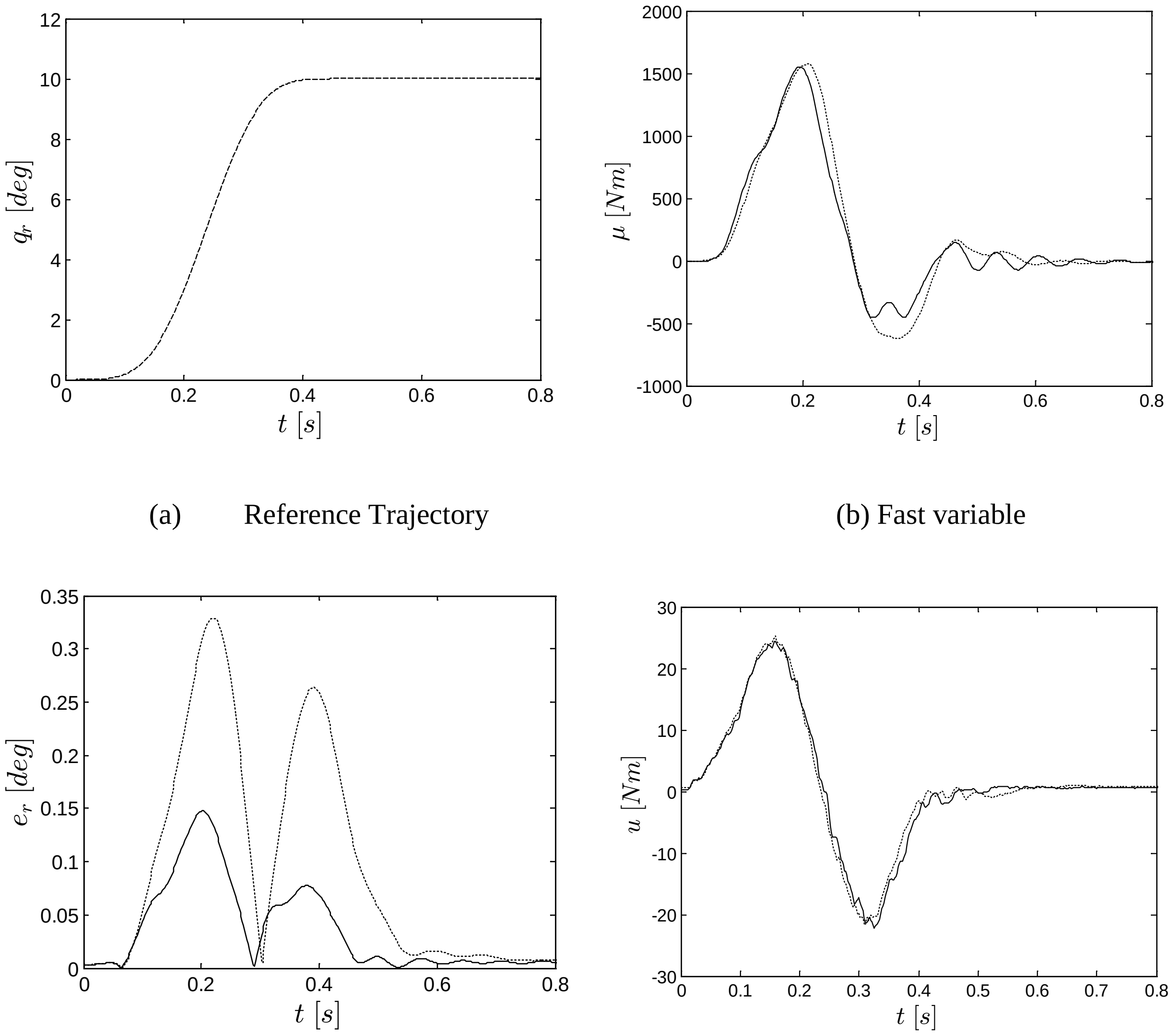
After verifying all control methods in the hardware in the HIL simulation, a real-world experiment was carried out. Controller 1 was first tested for the first joint to see if the same problem found in the simulation would occur. As in the simulation, the robot has become quickly unstable as soon as the motor was engaged. It was impossible to find a condition, i.e., gain and robot configurations, such that Controller 1 did not exhibit instability. Therefore, Controller 1 was excluded in the experiment for safety and to prevent from damage on the system. Controller 2 was also tested for the first joint. Instability was also observed as Controller 1. However, Controller 2 was able to stabilize the robot at a configuration with a low inertia (Figure 6.1). However, Controller 2 became unstable when it was applied to the second joint. Controller 3, 4, and 5 did not exhibit instability at all. The reason why Controller 1 and 2 exhibit instability could be explained based on the gain. Note that the only difference amongst Controller 1, 2, and 3 is the gain. It was found that the gain, $\mathbf{K}_d$ and $\mathbf{K}_p$ became the largest for Controller 1 and smallest for Controller 3 even though the gain is calculated for the same bandwidth. Such high gain could generate input with high frequency noise, which could effectively excite the unmodeled dynamics at high frequency. As shown in Figure 6.2 and Figure 6.3, the gain tuned by Controller 1 and 2 may misplace the poles due to the modeling error especially, the high frequency. And the error increases as the system becomes less stiff. This explains why Controller 2 only works at the configuration with low inertia.

In Figure 6.6, the experimental result for Controller 2 for configuration shown in Figure 6.1 is shown compared with Controller 3. In the test, the robot was requested to move the first joint by 10 degrees. The reference trajectory generated by the Hi5a controller is the time-optimal TVP with a small low-pass filter (Figure 6.6a) proposed in Chapter 5. Since Controller 3 did not exhibit instability at all, it was tested at a configuration with high inertia (Figure 6.7) at which the robot would be normally tested for motion performance. The robot moved along a short straight line, 100 mm along X+, Y+, and Z- simultaneously, with maximum speed. The motion was selected as it is similar to motions used in spot welding applications. The dynamic model was verified by utilizing the multivariable FRF (Figure 6.8). To measure the TCP, Tracker3$^{\text{TM}}$ is used.

In Figure 6.9, 6.10 and 6.11, the experimental results for Controller 3, 4, and 5 are compared. The tracking performance and the torque of Controller 3 and 5 were very similar during transient. Therefore, the assumption that the feedforward control is dominant during transient was validated. The rigid controller, Controller 4, generated a large tracking error with significant residual vibrations. The residual vibrations are dramatically reduced in Controller 3 compared with other two controllers. The input of Controller 3 is as smooth as other two controllers. In general, Controller 3 obtained by-far the best performance without chatter.

Controller index	Control method	Gain
1	*f-SPC*	$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0)$
2		$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0 + \varepsilon \boldsymbol{\varphi}_1)$
3		Full model
4	PD + feedforward with rigid model	Default
5	PD + feedforward with flexible model	Default

Table 6.1. Control methods used in experiment.



(a) Reference Trajectory

(b) Fast variable

(c) Tracking error on the link side

(d) control input

Figure 6.6. Experimental results for Controller 2 and 3 at a low inertia (sold lines: Controller 2, dotted lines: Controller 3, $\zeta = 0.7$, $\omega_n = 7\,Hz$, $a = 2.5$). The tracking error for Controller 3 is larger than that of Controller 2 because the gains, $\mathbf{K}_p$ and $\mathbf{K}_d$, are larger in Controller 2.

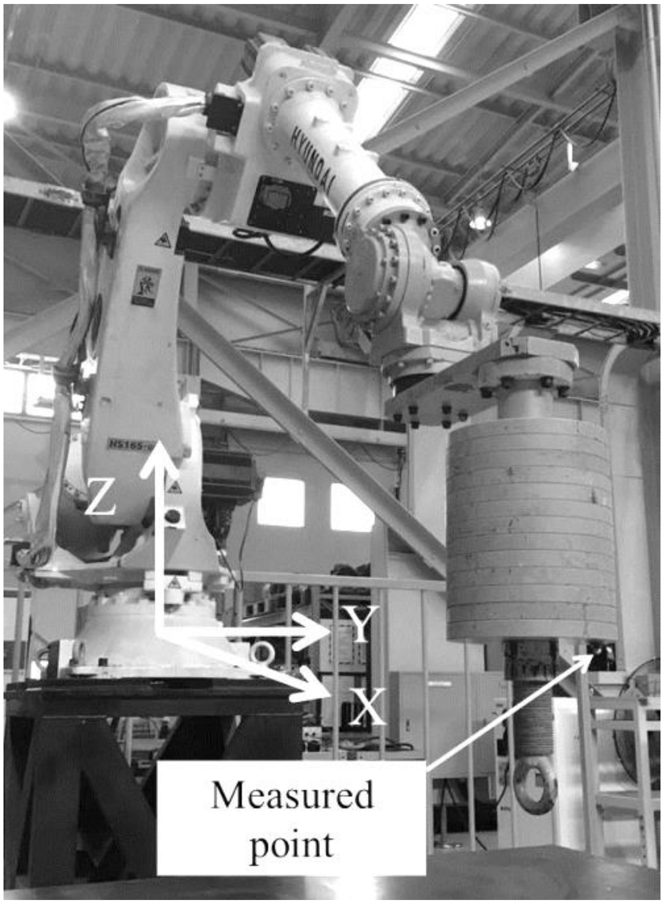
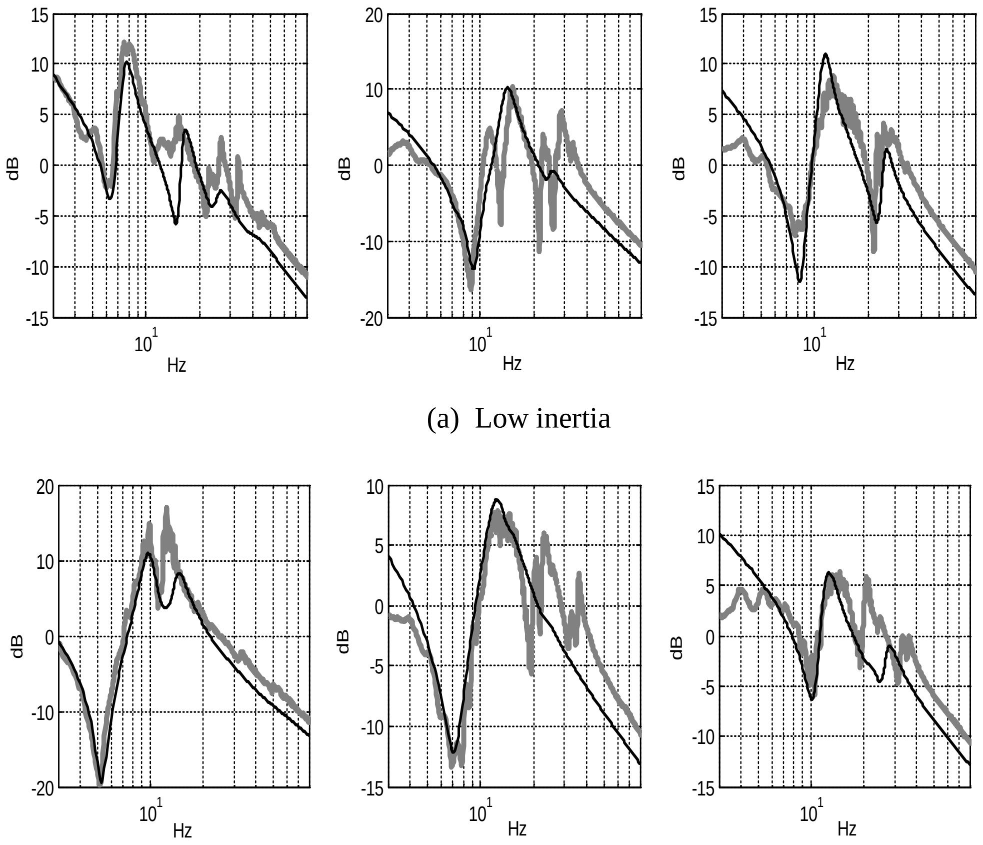


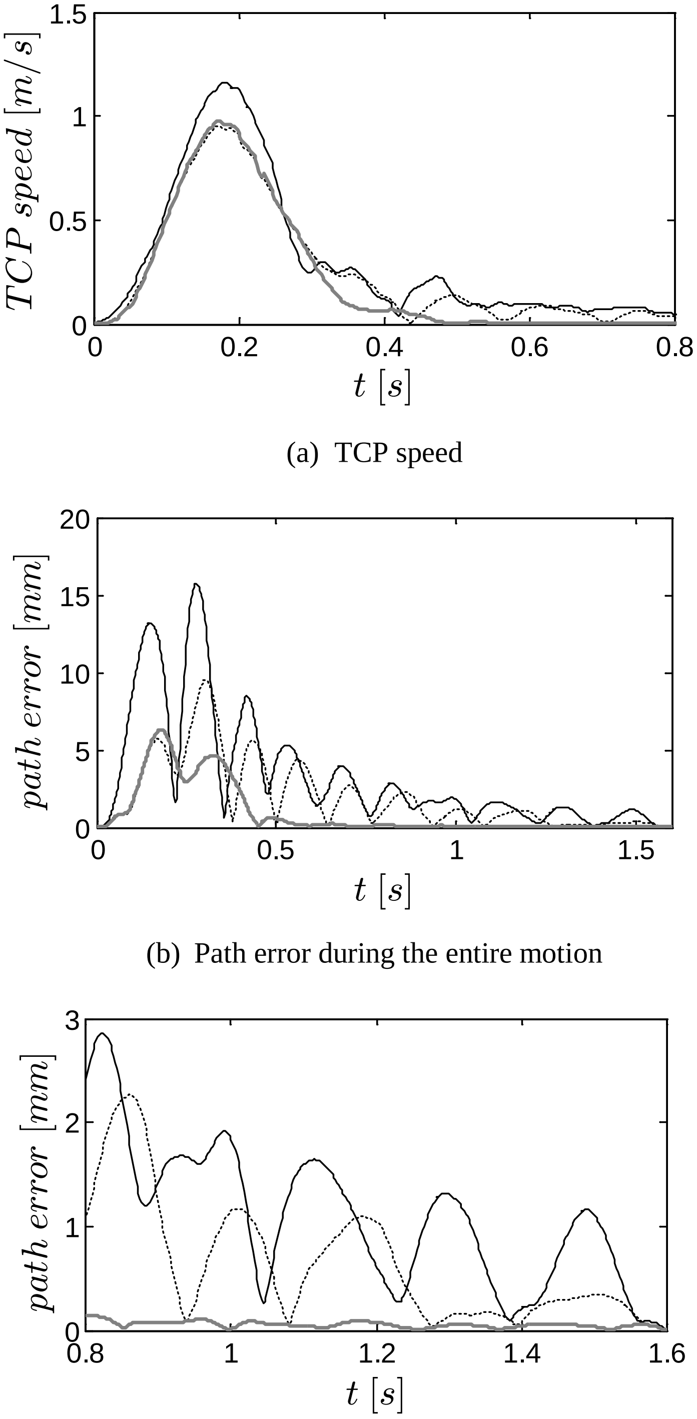
Figure 6.7. The configuration of HS165 at a high inertia.



(a) Low inertia

(b) High inertia

Figure 6.8. The FRF of the three main axes (shown in order, joint 1, joint 2, and joint 3 from left, The lines with smaller noise represents the model).



(a) TCP speed

(b) Path error during the entire motion

(c) Path error at steady state

Figure 6.9. Experimental results for Controller 3, 4, and 5 at a high inertia (gray lines: Controller 3, black solid lines: Controller 4, dotted lines: Controller 5).

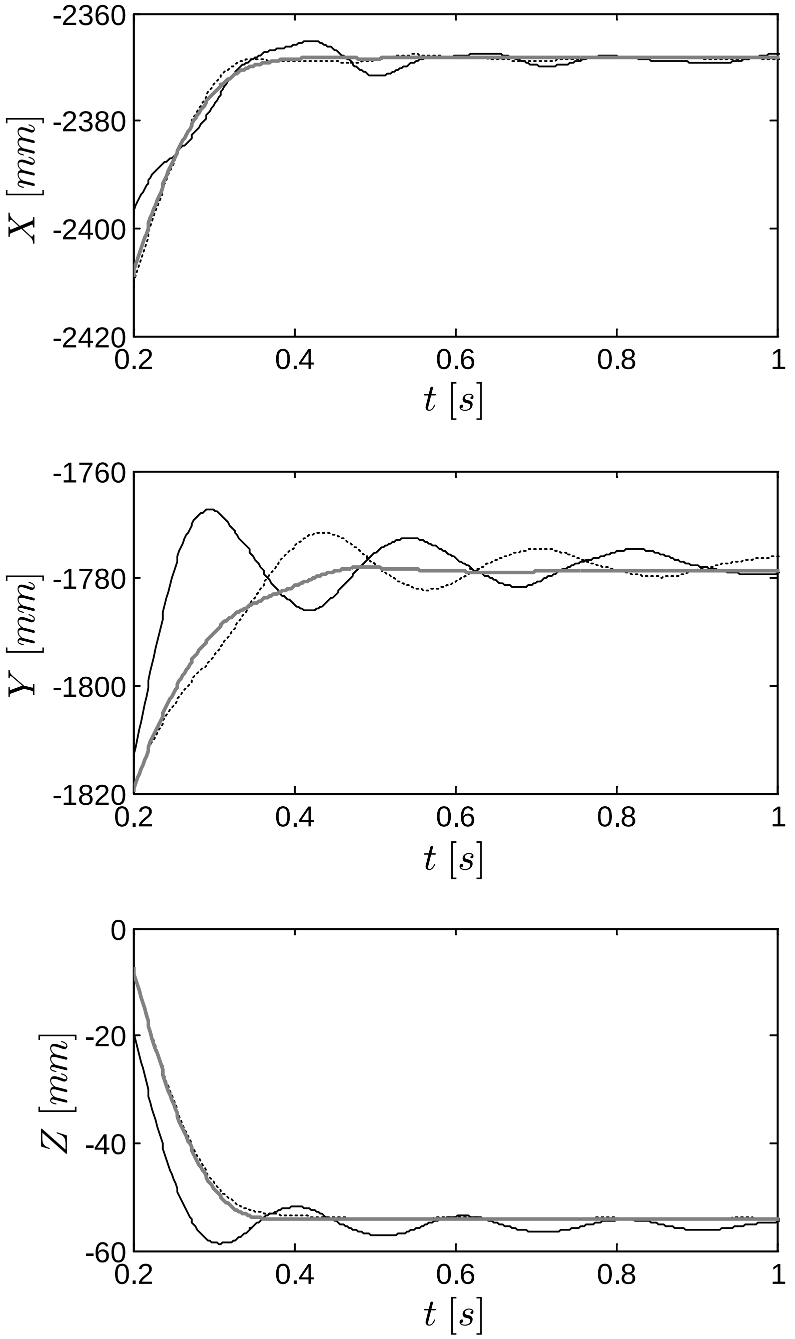


Figure 6.10. Experimental results for Controller 3, 4, and 5 at a high inertia: residual vibrations of TCP (gray lines: Controller 3, black solid lines: Controller 4, dotted lines: Controller 5).

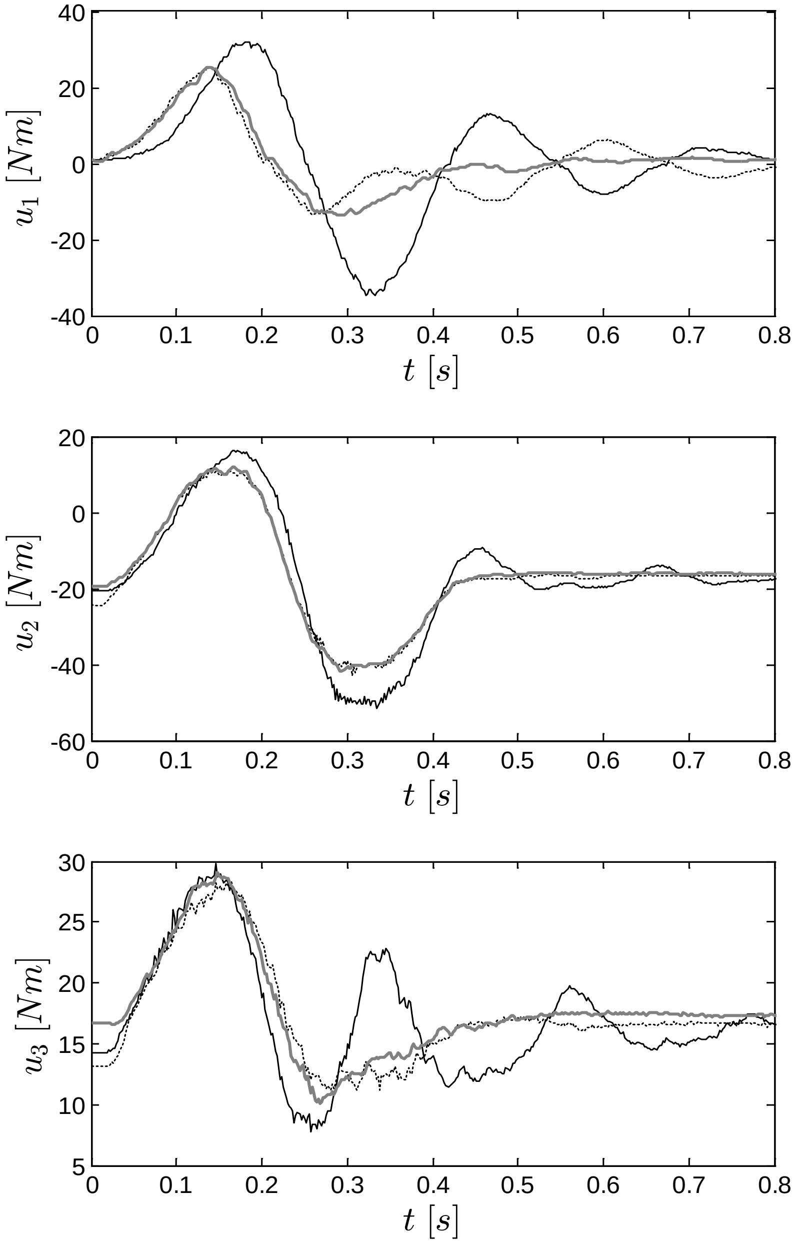


Figure 6.11. Experimental results for Controller 3, 4, and 5 at a high inertia: control input of the first three joints (gray lines: Controller 3, black solid lines: Controller 4, dotted lines: Controller 5).

6.6 Summary

The existing control strategies for flexible joint robots take one of two approaches. One is to utilize the full dynamic model from derivation of control laws. This approach generally leads to a controller that requires feedback up to jerk on the link side, which makes it difficult to implement due to possible chatter. The other approach uses the singular perturbation technique, which allows one to design a controller with feedback up to velocity. In spite of its theoretical benefit, a practical use of the SPC has been limited due to its inability to follow a fast varying trajectory. Two remedies have been proposed that can improve the conventional SPC. The feedforward control based on the flexible model is effective to cancel the nonlinearity during fast transients. With feedforward control, it was possible to linearize the feedback at steady state, obtaining a linearized singular perturbation model for the feedback system. For the feedback system, an invariant manifold without corrective control was found with a condition such that the closed-loop poles of the full model are located strictly on the left half plane. It was also shown that the decoupling control for the fast subsystem with PD control on the slow subsystem generates a decoupled closed-loop system which allows one to place the closed-loop poles arbitrarily with only position and velocity feedback. A real-world experiment with a 6DOF industrial robot demonstrated that the composite control based on the invariant manifold may not be sufficient as it exhibited instability unless the inertia is low. The composite control with gain tuned by the full model was successfully tested for the proposed time-optimal TVP trajectory proposed in Chapter 5, at an outstretched robot configuration where the perturbation parameter could be large.

Chapter 7: Conclusion

This thesis has investigated practical methods to obtain high motion performance of an industrial robot, especially one with a low-cost generic controller. The literature review suggested that time-optimal motion is difficult to implement with a high order trajectory due to the computational requirements. Therefore, a low order trajectory with model-based control was proposed as the motion control strategy. The validity of the strategy was verified by a comparative study. Then, two methods were proposed for planning and control. For planning, an online time-optimal TVP that can be planned along multiple path segments was presented. For control, a nonlinear multivariable control that requires only position and velocity feedback was proposed. In Section 7.1, the main contribution made in the thesis is summarized. The future work is discussed in Section 7.2.

7.1 Summary of Contributions

- **A comparative study on motion control methods**

  A comparative study on motion control methods for industrial robots has been presented. Two distinctive approaches, planning and control, which had been studied independently, were considered in a common framework with respect to the dynamic model and experimental platform. To improve the planning approach, a practical method to find the optimal jerk for smooth trajectory planning was proposed. Some limitations of the filtering technique were also discussed. The comparative study revealed that applying the planning approach to a robotic application where high motion performance is demanded is fundamentally limited either by slow motion or poor path accuracy even using an optimal design. It was shown that the key component that makes the control approach more attractive is the delay-free dynamic input shaper which is embedded in the feedforward control. The theoretical analysis was verified by a real-world experiment with a commercial industrial robot.

- **Online minimum-time trajectory planning**

  A novel trajectory planning method has been presented towards online industrial robot applications. The proposed method generates trajectories that are computationally efficient, dynamically near time-optimal, and maintain path-following integrity in high-frequency planning-and-control cycles. The method is based on the well-known minimum-time trajectory planning along specified paths. It was shown that this trajectory can be quickly approximated with a TVP, resulting in near time-optimal trajectories along specified paths requiring only four system-dynamics computations per path segment. For continuous motions, a method to safely transit between adjacent optimal path segments within geometric bounds was also presented. It was also shown that how the proposed second order, high frequency trajectory can be successfully used as the reference input for trajectory tracking by improving the control loop with a nonlinear feedforward control based on an elastic model. A real-world experiment with a 6DOF industrial robot validated the proposed approach.

- **Nonlinear control for the flexible joint dynamic model**

  A multivariable nonlinear controller that requires only position and velocity feedback has been proposed for industrial robots with elastic joints. The control method is a combination of a nonlinear feedforward control and the singular perturbation method for the linearized feedback system. Decoupling control was applied to the fast subsystem and then two control methods for the slow subsystems were proposed. First, the approximated reduced model that did not require a corrective control was derived by an explicit computation of the invariant manifold. The stability analysis revealed that the system was stable as long as the fast subsystem was at least twice as fast as the quasi-steady state. The second method was based on the full dynamic model to find the feedback gain. It was shown that the second method resulted in a multivariable full-state feedback controller that could obtain an arbitrary closed-loop performance with only position and velocity feedback. A real-world experiment with a 6DOF commercial industrial robot was carried out to verify these results.

7.2 Future Work

The following topics can be investigated further to improve the motion performance of industrial robots.

- **Estimation of Joint Stiffness**

  In order for the proposed scheme to work successfully, obtaining an accurate dynamic model (3.1) is indispensable. In general, it is relatively easy to obtain an accurate model for the rigid part. However, estimating the joint stiffness is quite challenging. According to a recent study in Moberg (2010), the flexible joint dynamic model (3.1) cannot globally represent the elastic behavior of industrial robots. It was suggested to use *the extended flexible joint model* to represent the dynamic model more accurately. However, it is not clear how the extended flexible joint model can be used for control because the added passive joints make the model extremely complicated for control design in terms of controllability and observability. Further study is required to find a suitable strategy to estimate the joint stiffness using a more simplified model for practical control design.

- **Friction and nonlinear gearbox**

  In this thesis, the main study is focused on time-optimal motions, namely, fast transients followed by residual vibrations. The flexible joint dynamic model (3.1) can represent the dominant behaviors during time-optimal motions such as inertial coupling, the Coriolis-centrifugal effect, and mechanical oscillations due to joint stiffness. However, the flexible joint dynamic model may not be good enough during slow motions. At low speed, the friction becomes dominant. Obtaining a good friction model for a typical industrial robot is extremely difficult due to its dependence on temperature and robot configuration (Section 3.2.3 in Moberg, 2010). Another important phenomenon during slow motions is the angular transmission error generated in the gearbox (Yoshioka *et al*., 2014), as well as the hysteresis, nonlinear stiffness, and backlash. In order to achieve high motion performance regardless of speed, obtaining good models for friction and the gearbox is required.

- **Nonlinear state-observer**

  A typical industrial robot has measurement only on the motor side. However, an accurate estimation on the link side is essential for fast damping of the residual vibrations on the link side. Even though the proposed control method in this thesis has alleviated the problem of estimating the link states by reducing the order from four (up to jerk) to two (up to velocity), a good estimation of the link states is still an open problem. However, it seems that the problem of estimating the link states based on the motor measurement has not received much attention from the robotics community. Notably, the work in Jankovic

(1995) is the only observer based on the flexible joint dynamic model that can estimate the link position and velocity from the motor measurements. Most of the related works assume that the link position is available for measurement (Nocosia *et al*., 1988; Nicosia and Tornambè, 1989; Nicosia and Tomei, 1990; Tomei, 1990). The observer proposed in Jankovic (1995) is a nonlinear reduced observer. The major challenge of this observer is that the robustness is guaranteed by high gain and accurate estimation of the joint stiffness. Therefore, it is difficult to select the observer gain in case of model uncertainty and limited sampling rate. Further studies are needed for state observers in order to estimate the link states more robustly for practical implementation.

Bibliography

Albu-Schäffer, A. and G. Hirzinger (2001). A globally state feedback controller for flexible joint robots. *Advanced Robotics*, Vol.15, pp. 799–814.

Albu-Schäffer, A., C. Ott, and G. Hirzinger (2007). A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The International Journal of Robotics Research*, 26(1), pp. 23–39.

Altintas, Y. and K. Erkorkmaz (2003). Feedrate optimization for spline interpolation in high speed machine tools. *CIRP Annals – Manufacturing Technology*, Vol. 52, pp. 297–302.

Altintas, Y. and M. R. Khoshdarregi (2012). Contour error control of CNC machine tools with vibration avoidance. *CIRP Annals – Manufacturing Technology*, Vol. 61, pp. 335–338.

An, C. H., C. G. Atkeson, and J. M. Hollerbach (1998). *Model-Based Control of a Robot Manipulator*. The MIT Press.

Antonelli, G., S. Chiaverini, G. Gerio, M. Palladino, and G. Renga (2007). SmartMove4: an industrial implementation of trajectory planning for robots. *Industrial Robot: An International Journal*, 34(3), pp. 217–224.

Antonelli, G., C. Curatella, and A. Marino (2009). Constrained motion planning for open-chain industrial robots. *Robotica*, Vol. 29, pp. 403–420.

Asada, H. and T. Kanade (1983). Control of a direct-drive arm. *Journal of Dynamics Systems, Measurement and Control*, 105(3), pp. 136–142.

Biagiotti, L. and C. Melchiorri (2008). *Trajectory planning for automatic machines and robots*, Springer.

Bobrow, J. E., S. Dubowsky, and J. S. Gibson (1985). Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research*, 4(3), pp. 3–17.

Björkman, M., T. Brogårdh, and S. Hanssen (2008). A new concept for motion control of industrial robots. *Proc. IFAC World Congress*, pp. 15714–15715, Seoul, Korea.

Brogårdh, T. (2009). Robot control overview: an industrial perspective. *Modeling, Identification and Control*, 30(3), pp. 167–180.

Brogliato, B., R. Ortega, and R. Lozano (1995). Global Tracking Controllers for Flexible-joint Manipulators: a Comparative Study. *Automatica*, 31(7), pp. 941–956.

Brogliato, B., D. Rey, A. Pastore, and J. Barnier (1998). Experimental comparison of nonlinear controllers for flexible joint manipulators. *The International Journal of Robotics Research*, 17(3), pp. 260–281.

Broquère, X., D. Sidobre, and I. Herrer-Aguilar (2008). Soft motion trajectory planner for service manipulator robot. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2808–2813, Nice, France.

Chang, P. and H. Park (2005). Time-varying input shaping technique applied to vibration reduction of an industrial robot. *Control Engineering Practice*, Vol. 13, pp. 121-130.

Constantinescu, D. and E. A. Croft (2000). Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of Robotic Systems*, 17(5), pp. 233–249.

Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control, Third Edition,* Pearson Prentice Hall.

Dahl, O. (1992). Path constrained robot control. Ph.D. thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Dahl, O. (1994). Path-constrained robot control with limited torques – experimental evaluation. *IEEE Transactions on Robotics and Automation*, 10(5), pp. 658–669.

Dahl, O. and L. Nielsen (1990). Torque-limited path following by on-line trajectory time scaling. *IEEE Transactions on Robotics and Automation*, 6(5), pp. 554–561.

de Queiroz, M. S., S. Donepudi, T. Burg, and D. M. Dawson (1998). Model-based control of rigid-link flexible-joint robots: an experimental evaluation. *Robotica*, 16(1), pp. 11–21.

De Luca, A. (2000). Feedforward/feedback laws for the control of flexible robots. *Proc. IEEE international Conference on Robotics and Automation*, pp. 233–240, San Francisco, USA.

De Luca, A. and R. Farina (2002). Dynamic scaling of trajectories for robots with elastic joints. *Proc. International Conference on Robotics and Automation,* pp. 2436–2442, Washington DC, USA.

De Luca, A. and P. Lucibello (1998). A general algorithm for dynamic feedback linearization of robots with elastic joints. *Proc. IEEE International Conference on Robotics and Automation,* pp. 504–510, Leuven, Belgium.

De Luca, A., D. Schroder, and M. Thümmel (2007). An Acceleration-based State Observer for Robot Manipulators with Elastic Joints. *Proc. IEEE International Conference on Control, Automation and Systems*, pp. 10–14, Roma, Italy.

De Luca, A., B. Siciliano, and L. Zollo (2005). PD control with on-line gravity compensation for robots with elastic joints: theory and experiments. *Automatica*, 41(10), pp. 1809–1819.

Dong, J., P. Ferreria, and J. Stori (2007). Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture*, 47(12-13), pp. 1941–1955.

Ezair, B., T. Tassa, and Z. Shiller (2014). Planning high order trajectories with general initial and final conditions and asymmetric bounds. *The International Journal of Robotics Research*, 33(6), pp. 898–916.

Gandhi, P. S. and F. Ghorbel (2005). High-speed precision tracking with harmonic drive systems using integral manifold control design. *International Journal of Control*, 78(2), pp. 112–121.

Ghorbel, F. and M. W. Spong (2000). Integral manifolds of singularly perturbed systems with application to rigid-link flexible-joint multibody systems. *International Journal of Non-Linear Mechanics*, *35*(1), pp. 133–155.

Good, M., L. Sweet, and K. Strobel (1985). Dynamic models for control system design of integrated robot and drive systems. *Journal of Dynamic Systems, Measurement, and Control*, 107(1), pp. 53–59.

Guarino Lo Bianco, C. and O. Gerelli (2011). Online trajectory scaling for manipulators subject to high-order kinematic and dynamic constraints. *IEEE Transactions on Robotics*, 27(6), pp. 1144–1152, 2011.

Haschke, R., E. Weitnauer, and H. Ritter (2008). On-line planning of time-optimal, jerk-limited trajectories. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3248–3253, Nice, France.

Hillsley, K. and S. Yurkovich (1993). Vibration control of a two-link flexible robot arm. *Dynamics and Control*, 3(3), pp. 261–280.

Hollerbach, J. M. (1984). Dynamic scaling of manipulator trajectories. *Journal of Dynamics Systems, Measurement and Control*, Vol. 106, pp. 102–106.

Hung, J. Y. (1991). Control of industrial robots that have transmission elasticity. *IEEE Transactions on Industrial Electronics*, *38*(6), pp. 421–427.

Jankovic, M. (1995). Observer based control for elastic joint robots. *IEEE Transactions on Robotics and Automation*, 11(4), pp. 618-623.

Katzschmann, R., T. Kröger, T. Asfour, and O. Khatib (2013). Towards online trajectory generation considering robot dynamics and torque limits. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5644–5651,Tokyo, Japan.

Khalil, H. K. (2014). High-gain observers in nonlinear feedback control. *International Journal of Robust and Nonlinear Control*, 24(6), pp. 993–1015.

Kelly, R., R. Ortiga, A. Ailon, and A. Loria (1994). Global regulation of flexible joint robots using approximate differentiation. *IEEE Transactions on Automatic Control*, 39(6), pp. 1222–1224.

Kelly, R. and R. Salgado (1994). PD control with computed feedforward of robot manipulators: a design procedure. *IEEE Transactions on Robotics and Automation*, 10(4), pp. 566–571.

Khorasani, K. and P. Kokotovic (1986). A corrective feedback design for nonlinear systems with fast actuators," *IEEE Transactions on Automatic Control*. 31(1), 67-69.

Khorrami, F., S. Jain, and A. Tzes (1995). Experimental results on adaptive nonlinear control and input preshaping for multi-link flexible manipulators. *Automatica*, 31(1), pp.83–97.

Khoshdarregi, M. R., S. Tapped, and Y. Altintas (2014). Integrated five-axis trajectory shaping and contour error compensation for high-speed CNC machine tools. *IEEE/ASME Transactions on Mechatronics*, 19(6), pp. 1859–1871.

Kokotovic, P., H. K. Khalil, and J. O'Reilly (1999). *Singular Perturbation Methods in Control*, SIAM.

Kieffer, J., A. J. Cahill, and M. R. James (1997). Robust and accurate time-optimal path-tracking control for robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(6), pp. 880–890.

Kim, J. and E. A. Croft (2015). A benchmark study on the planning and control for industrial robots with elastic joints. *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1378–1383, Busan, Korea.

Kim, J. and E. A. Croft (accepted). Co-designed trajectory planning and feedforward control for industrial robots. *IEEE Transactions on Robotics*.

Kim, J. and E. A. Croft (submitted in February 2017). A practical approach to feedback linearization for industrial robots with elastic joints. *IEEE Transactions on Control Systems Technology*.

Kim, J. and E. A. Croft (submitted in October 2016). A comparative study on motion control methods for industrial robots with elastic joints. *IEEE/ASME Transactions on Mechatronics*.

Kim, J., S-R. Kim, S-J, Kim, and D-H Kim (2010). A practical approach for minimum-time trajectory planning for industrial robots. *Industrial Robot: An International Journal*, 37(1), pp. 51–61.

Kröger, T. and F. M. Wahl (2010). Online trajectory generation: basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 26(1), pp. 94–111.

Lambrechts, P., M. Boerlage and M. Steinbuch (2005). Trajectory planning and feedforward design for electromechanical motion systems. *Control Engineering Practice*, Vol. 13, pp. 145–157.

Le, M. D., Z. M. Gong, and W. Lin (2009). Motion Profile Design to Reduce Residual Vibration of High-Speed Positioning Stages. *IEEE/ASME Transactions on Mechatronics*, 14(2), pp. 264–269.

Leahy, M. B. and G. N. Saridis (1987). Compensation of unmodeled PUMA manipulator dynamics. *Proc. IEEE International Conference on Robotics and Automation*, pp. 151–156, Raleigh, USA.

Lessard, L., P. Bigras, Z. Liu, and B. Hazel (2012). Characterization, modeling and vibration control of a flexible joint for a robotic system. *Journal of Vibration and Control*, 20(6), pp. 943–960.

Lloyd, J. and V. Hayward (1993). Trajectory generation for sensor-driven and time-varying tasks. *The International Journal of Robotic Research*, 12(4), pp. 380–393.

Lozano, R. and B. Brogliato (1992). Adaptive control of robot manipulators with flexible joints," *IEEE Transactions on Automatic Control*, 37(2), pp.174–181.

Luh, J. Y. S., M. W. Walker, and R. P. C. Paul (1980). On-line computation scheme for mechanical manipulators. *Journal of Dynamics Systems, Measurement and Control*, 102(2), pp. 69-76.

Macfarlane, S. and E. A. Croft (2003). Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Transactions on Robotics and Automation*, 19(1), pp. 42–52.

Meckl, P. H. and M. C. Woods (1998). Optimized S-curve motion profiles for minimum residual vibration. *Proc. The American Control Conference*, pp. 2627–2631, Philadelphia, USA.

Miyazaki, T., S. Ohtaki, S. Tungpataratanawong, and K Ohishi (2004). High speed motion control method of industrial robot based on dynamic torque compensation and two-degrees-of-freedom control system. *IEEJ Transactions on Industry Applications*, 123(5), pp. 525–532 (In Japanese).

Moberg, S. (2010). Modeling and control of flexible manipulators. Ph.D. thesis, Department of Electrical Engineering, Linköping University.

Moberg, S. and S. Hanssen (2008). On feedback linearization for robust tracking control of flexible joint robots. *Proc. IFAC World Congress,* Seoul, Korea.

Moberg, S., E. Wernholt, S. Hanssen, and T. Brogårdh (2014). Modeling and parameter estimation of robot manipulators using extended flexible joint models. *Journal of Dynamics Systems, Measurement and Control*, 136(3), p.031005.

Nicosia, S. and P. Tomei (1992). A method to design adaptive controllers for flexible joint robots. *Proc. IEEE International Conference on Robotics & Automation*, pp.701–706, Nice, France.

Nicosia, S., P. Tomei, and A. Tornambè (1988), A nonlinear observer for elastic robots. *IEEE Journal of Robotics and Automation*, 4(1), pp. 45–52.

Nicosia, S. and A. Tornambè (1989). High-gain observers in the state and parameter estimation of robots having elastic joints. *Systems and Control Letters*, Vol. 13, pp. 331–337.

Nicosia, S. and P. Tomei (1990). A method for the state estimation of elastic joint robots by global position measurements. *International Journal of Adaptive Control and Signal Processing*, 4(6), pp. 475–486.

Nicosia, S. and P. Tomei (1998). On the feedback linearization of robots with elastic joints. *Proc. 27nd Conference on Decision and Control*, pp. 180–185, Austin, USA.

Nguyen, K., T. Ng, and I. Chen (2008). On algorithms for planning S-curve motion profiles. *International Journal of Advanced Robotic Systems*, 5(1), pp. 99–106.

Öhr, J., S. Moberg, E. Wernholt, S. Hanssen, J. Pettersson, and S. Sander-Tavallaey (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. *Proc. ISMA 2006 International Conference on Noise and Vibration Engineering, Leuven, Belgium,* pp. 3305–3314.

Ott, C., A. Albu-Schäffer, A. Kugi, and G. Hirzinger (2008). On the passivity-based impedence control of flexible joint robots. *IEEE Transactions on Robotics*, 24(2), pp. 416–429.

Park, F. C., J. E. Bobrow, and S. R. Ploen (1995). A Lie group formulation of robot dynamics. *The International Journal of Robotics Research*, 14(6), pp. 609–618.

Park, F. C. and B. Ravani (1995). Bézier curves on Riemannian manifolds and Lie groups with kinematics applications. *ASME Journal of Mechanical Design*, Vol. 117, pp. 36–40.

Park, J., P. Chang, H. Park, and E. Lee (2006). Design of learning input shaping technique for residual vibration suppression in an industrial robot. *IEEE/ASME Transactions on Mechatronics*, Vol. 11, No.1, pp. 55–65.

Parker, G. G., K. Groom, J. E. Hurtado, J. Feddema, R. D. Robinett, and F. Leban (1999). Experimental verification of a command shaping boom crane control system. *Proc. American Control Conference*, pp. 86–90, San Diego, USA.

Pfeiffer, F. and R. Johanni (1987). A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, 3(2), pp. 115-123.

Pham, Q.-C. (2014). A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics*, 30(6), pp. 1533–1540.

Readman, M. C. and P. R. Belanger (1992). Stabilization of the Fast Modes of a Flexible-Joint Robot. *The International Journal of Robotics Research*, 11(2), pp. 123–134.

Reynoso-Mora, P., W. Chen, and M. Tomizuka (2013). On the time-optimal trajectory planning and control of robotic manipulators along predefined paths. *Proc. American Control Conference*, pp. 371–377, Washington DC, USA.

Sahar, G. and J. M. Hollerbach (1986). Planning of minimum-time trajectories for robot arms. *The International Journal of Robotics Research*, Vol. 5, pp. 90–100.

Salmasi, H., R. Fotouhi, and P. N. Nikiforuk (2010). A manoeuvre control strategy for flexible-joint manipulators with joint dry friction. *Robotica*, 28(4), pp. 621–635.

Santibañez, V. and R Kelly (2001). PD control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, 19(1), pp. 11–19.

Seibert, P. and R. Suarez (1990). Global stabilization of cascade system. *Systems & Control Letter*, Vol. 14, pp. 347–352.

Shiller, Z. and H-H. Lu (1992). Computation of path constrained time optimal motions with dynamic singularities. *Journal of Dynamic Systems, Measurement, and Control*, 114(1), pp. 34–40.

Shin, K. G. and N. D. McKay (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, AC-30, pp. 531–541.

Shoemake, K. (1985). Animating rotation with quaternion curves. *Proc. The Annual Conference on Computer Graphics and Interactive Techniques*, 19(3), pp. 245–254.

Slotine, J-J. and W. Li (1988). Adaptive manipulator control: a case study. *IEEE Transactions on Automatic Control*, 33(11), pp. 995–1003.

Slotine, J-J. and W. Li (1991). *Applied Nonlinear Control*. Prentice-Hall.

Siciliano, B. and W. J. Book (1988). A singular perturbation approach to control of lightweight flexible manipulators. *The International Journal of Robotics Research*, *7*(4), pp. 79–90.

Singer, N. C. and W. P. Seering (1990). Preshaping command inputs to reduce system vibration. *Journal of Dynamic Systems, Measurement, and Control*, 112(1), pp. 76–82.

Singhose, W. (2009). Command shaping for flexible systems: A review of the first 50 years. *International Journal of Precision Engineering and Manufacturing*, 10(4), pp. 153–168.

Singhose, W. (2010). Command generation for flexible systems by input shaping and command smoothing. *Journal of Guidance, Control, and Dynamics*, 33(6), pp. 1697–1707.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control: Analysis and Design*, John Wiley & Sons, Chichester.

Spong, M. W. (1987). Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 109(4), pp. 310–318.

Spong, M. W., K. Khorasani, and P. V. Kokotovic (1985). A slow manifold approach to feedback control of nonlinear flexible systems. *Proc. American Control Conference*, pp. 1386–1391.

Spong, M. W., K. Khorasani, and P. V. Kokotovic (1987). An integral manifold approach to the feedback control of flexible joint robots. *IEEE Journal on Robotics and Automation*, 3(4), pp. 291–300.

Sweet, L. and M. Good (1984). Re-definition of the robot motion control problem: effects of plant dynamics, drive system constraints, and user requirements. *Proc. Decision and Control*, pp. 724–732, Las Vagas, USA.

Swevers, J., D. Torfs, M. Adams, J. De Schutter, and H. Van Brussel (1991). Comparison of control algorithms for flexible joint robots implemented on a KUKA IR 161/60 industrial robot. *Proc. International Conference on Advanced Robotics*, pp. 120–125, Pisa, Italy.

Taghirad, H. D., and M. A. Khosravi (2012). Stability analysis and robust composite controller synthesis for flexible joint robots. *Advanced Robotics*, 20(2), pp. 181–211.

Thümmel, M., M. Otter, and J. Bals (2001). Control of robots with elastic joints based on automatic generation of inverse dynamics models. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 925-930, Hawaii, USA.

Tomei, P. (1990). An observer for flexible joint robots. *IEEE Transactions on Automatic Control*, 35(6), pp. 739–743.

Tomei, P. (1991). A simple PD controller for robots with elastic joints. *IEEE Transactions on Automatic Control*, 36(10), pp. 1208–1213.

Tsay, D. and C. F. Lin (2005). Asymmetrical Inputs for Minimizing Residual Response. *Proc. IEEE International Conference on Mechatronics*, pp. 235–240.Taipei, Taiwan.

Tungpataratanawong, S., K. Ohishi, and T. Miyazaki (2005). Robust motion control of industrial robot based on robot parameter identification and feedforward control considering resonant frequency. *IEEJ Transactions on Industry Applications*, 125(6), pp. 586–574.

Vakil, M., R. Fotouhi, and P. N. Nikiforuk (2011). End-effector trajectory tracking of a class of flexible link manipulators. *Journal of Vibration and Control*, 17(1), pp. 55–68.

Verscheure, D., B. Demeulenaere, J. Swevers, J. Schutter, and M. Diehl (2009), Time-optimal path tracking for robots: a convex optimization approach. *IEEE Transactions on Automatic Control*, 54(10), pp. 2318–2327.

Wernholt, E. and S. Gunnarsson (2008). Estimation of nonlinear effects in frequency domain identification of industrial Robots. *IEEE Transactions on Instrumentation and Measurement*, 57(4), pp. 856–863.

Wigstrom, O., B. Lennartson, A. Vergnano, and C. Breitholtz (2013). High-level scheduling of energy optimal trajectories. *IEEE Transactions on Automation Science and Engineering*, 10(1), pp. 57–64.

Wilson, G. A. and G. W. Irwin (1993). Tracking control of manipulators with elastic joints. *Proc. IEEE Conference on Control Applications*, pp. 165–170.

Yoshioka, T., Y. Hirano, K. Ohishi, T. Miyazaki, and Y. Yokokura (2014). Vibration suppressing control method of angular transmission error of cycloid gear for industrial robots. *Proc. The International Power Electronics Conference*, pp. 1956–1961.

Yuan, J. and Y. Stepanenko (1993). Composite adaptive control of flexible joint robots. *Automatica*, 29(3), pp. 609–619.

Appendices

Appendix A

A.1 Singular Perturbation Theory

The standard form of a singularly perturbed system can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{z}) \in \Re^n \tag{A.1a}$$

$$\varepsilon \dot{\mathbf{z}} = \mathbf{g}(\mathbf{x}, \mathbf{z}, \varepsilon) \in \Re^m , \tag{A.1b}$$

where $0 < \varepsilon \ll 1$: the singular perturbation parameter, $\mathbf{x}$: slow variable , and $\mathbf{z}$: fast variable. Suppose an n–dimensional smooth manifold, M_ε, residing in the $n+m$– dimensional state space

$$M_\varepsilon : \mathbf{z} = \boldsymbol{\varphi}(\mathbf{x},\ \varepsilon) \in \Re^m . \tag{A.2}$$

M_ε is referred to as *the invariant manifold* if the solution of (A.1) satisfies

$$\mathbf{z}(t^*) = \boldsymbol{\varphi}(\mathbf{x}(t^*),\ \varepsilon) \ \rightarrow z(t) = \boldsymbol{\varphi}(\mathbf{x}(t),\ \varepsilon) \ \text{ for } \ \forall t \geq t^* . \tag{A.3}$$

When the condition (A.3) is not satisfied, or solutions of (A.1) are not on M_ε, there is a deviation between the fast variable and M_ε,

$$\boldsymbol{\eta} = \mathbf{z} - \boldsymbol{\varphi}(\mathbf{x}, \varepsilon) \in \Re^m . \tag{A.4}$$

Inserting (A.4) into (A.1), we obtain two subsystems in separate time-scales,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\ \boldsymbol{\varphi}(\mathbf{x},\varepsilon) + \boldsymbol{\eta}) \tag{A.5a}$$

$$\boldsymbol{\eta}' = \mathbf{g}(\mathbf{x},\boldsymbol{\varphi}(\mathbf{x},\varepsilon) + \boldsymbol{\eta}) - \frac{\partial\boldsymbol{\varphi}}{\partial\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\varphi}(\mathbf{x},\varepsilon) + \boldsymbol{\eta})\,, \tag{A.5b}$$

where $\frac{1}{\varepsilon} \triangleq \frac{d\tau}{dt}$ and $\boldsymbol{\eta}' \triangleq \frac{d}{d\tau}\boldsymbol{\eta}$. On M_ε, it holds

$$\boldsymbol{\eta} = \mathbf{0},\ \dot{\boldsymbol{\eta}} = \mathbf{0}\ . \tag{A.6}$$

By inserting (A.6) into (A.5), we obtain *the exact slow model*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\ \boldsymbol{\varphi}(\mathbf{x},\varepsilon))\,, \tag{A.7}$$

and *the manifold condition*

$$\varepsilon\frac{\partial\boldsymbol{\varphi}}{\partial\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\varphi}(\mathbf{x},\varepsilon)) = \mathbf{g}(\mathbf{x},\boldsymbol{\varphi}(\mathbf{x},\varepsilon))\,. \tag{A.8}$$

At $\varepsilon = 0$, the system (A.5) is decoupled in separate time-scales. Namely, the slow system or *the quasi-steady state*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\ \boldsymbol{\varphi}(\mathbf{x},0))\,, \tag{A.9}$$

and the fast system or *the boundarylayer*

$$\boldsymbol{\eta}' = \mathbf{g}(\mathbf{x},\boldsymbol{\varphi}(\mathbf{x},0) + \boldsymbol{\eta})\,. \tag{A.10}$$

Assuming that the boundary layer is stable about the equilibrium point, $\boldsymbol{\eta} = \mathbf{0}$, $\dot{\boldsymbol{\eta}} = \mathbf{0}$, then (A.10) becomes

$$\mathbf{0} = \mathbf{g}(\mathbf{x},\boldsymbol{\varphi}(\mathbf{x},0))\,. \tag{A.11}$$

In the fast time scale, (A.7) can be written as

$$\mathbf{x}' = \varepsilon\,\mathbf{f}\,(\mathbf{x}, \boldsymbol{\varphi}(\mathbf{x}, \varepsilon)) . \tag{A.12}$$

Clearly, (A.12) is stationary at $\varepsilon = 0$. Therefore, the two systems (A.9) and (A.10) are all stationary in the fast time-scale at $\varepsilon = 0$ and the invariant manifold, M_ε, approaches to *the equilibrium manifold*, M_0. According to Tikhonov's theorem, M_ε exists if (A.11) has distinctive real roots and the boundary layer is strictly stable.

A typical control strategy based on the singular perturbation method pursues a composite control law for the quasi-steady state and the boundary layer. That is, the control law is designed independently for each subsystem as

$$\dot{\mathbf{x}} = \mathbf{f}\,(\mathbf{x},\ \boldsymbol{\varphi}_0) \tag{A.13a}$$

$$\boldsymbol{\eta}' = \mathbf{g}(\mathbf{x}, \boldsymbol{\varphi}_0 + \boldsymbol{\eta}) . \tag{A.13b}$$

If the boundary layer is stable, i.e., $\boldsymbol{\eta} = \mathbf{0}$ and $\dot{\boldsymbol{\eta}} = \mathbf{0}$, then the error of the model (A.13a) is $O(\varepsilon)$. If ε is not negligible, one can pursue a more accurate slow model with error of $O(\varepsilon^2)$. To do so, assume

$$\boldsymbol{\varphi} = \boldsymbol{\varphi}_0 + \varepsilon\,\boldsymbol{\varphi}_1 . \tag{A.14}$$

In general, the term $\boldsymbol{\varphi}_0$ is obtained by solving for (A.1b) at $\varepsilon = 0$ as

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \boldsymbol{\varphi}_0, 0) \ . \tag{A.15}$$

The term $\boldsymbol{\varphi}_1$ can be obtained by a power series expansion on the manifold condition about $\varepsilon = 0$, resulting in

$$\boldsymbol{\varphi}_1 = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right)^{-1}_{\mathbf{z}=\boldsymbol{\varphi}_0} \frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) . \tag{A.16}$$

The detailed calculation of (A.16) can be found in Appendix A.2. Therefore, *the approximate slow model* with the error $O(\varepsilon^2)$ is obtained as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\ \boldsymbol{\varphi}_0 + \varepsilon \boldsymbol{\varphi}_1) . \tag{A.17}$$

A.2 Approximated Invariant Manifold by the Power Series Expansion

A power series expansion is applied to the manifold condition

$$\varepsilon \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x},\ \boldsymbol{\varphi}) = \mathbf{g}(\mathbf{x},\ \boldsymbol{\varphi}) . \tag{A.18}$$

Let

$$\boldsymbol{\varphi} = \boldsymbol{\varphi}_0 + \varepsilon \boldsymbol{\varphi}_1 + \varepsilon^2 \boldsymbol{\varphi}_2 . \tag{A.19}$$

The Taylor expansion about $\varepsilon = 0$ results in

$$\begin{aligned} &\varepsilon \left(\frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} + \varepsilon \frac{\partial \boldsymbol{\varphi}_1}{\partial \mathbf{x}} + \varepsilon^2 \frac{\partial \boldsymbol{\varphi}_2}{\partial \mathbf{x}} \right) \\ &\times \left(\mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \right|_{\mathbf{z}=\boldsymbol{\varphi}_0} (\varepsilon \boldsymbol{\varphi}_1 + \varepsilon^2 \boldsymbol{\varphi}_2) + \left. \frac{\partial^2 \mathbf{f}}{\partial \mathbf{z}} \right|_{\mathbf{z}=\boldsymbol{\varphi}_0} (\varepsilon \boldsymbol{\varphi}_1 + \varepsilon^2 \boldsymbol{\varphi}_2)^2 \right) . \\ &= \mathbf{g}(\mathbf{x}, \boldsymbol{\varphi}_0) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right|_{\mathbf{z}=\boldsymbol{\varphi}_0} (\varepsilon \boldsymbol{\varphi}_1 + \varepsilon^2 \boldsymbol{\varphi}_2) + \left. \frac{\partial^2 \mathbf{g}}{\partial \mathbf{z}^2} \right|_{\mathbf{z}=\boldsymbol{\varphi}_0} (\varepsilon \boldsymbol{\varphi}_1 + \varepsilon^2 \boldsymbol{\varphi}_2)^2 \end{aligned} \tag{A.20}$$

Utilizing $\mathbf{g}(\mathbf{x}, \boldsymbol{\varphi}_0) = \mathbf{0}$ and ignoring the terms higher than ε^2 , we obtain

$$
\begin{aligned}
&\varepsilon \frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) + \varepsilon^2 \left(\frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} \boldsymbol{\varphi}_1 + \frac{\partial \boldsymbol{\varphi}_1}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) \right) \\
&= \varepsilon \left.\frac{\partial \mathbf{g}}{\partial \mathbf{z}}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} \boldsymbol{\varphi}_1 + \varepsilon^2 \left(\left.\frac{\partial \mathbf{g}}{\partial \mathbf{z}}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} \boldsymbol{\varphi}_2 + \left.\frac{\partial^2 \mathbf{g}}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} {\boldsymbol{\varphi}_1}^2 \right)
\end{aligned} \quad . \tag{A.21}
$$

Equating the terms for ε and ε^2, the following results are obtained.

$$
\boldsymbol{\varphi}_1 = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right)^{-1}_{\mathbf{z}=\boldsymbol{\varphi}_0} \frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) \tag{A.22}
$$

$$
\boldsymbol{\varphi}_2 = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right)^{-1}_{\mathbf{z}=\boldsymbol{\varphi}_0} \left(\frac{\partial \boldsymbol{\varphi}_0}{\partial \mathbf{x}} \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} \boldsymbol{\varphi}_1 + \frac{\partial \boldsymbol{\varphi}_1}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\varphi}_0) - \left.\frac{\partial^2 \mathbf{g}}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\boldsymbol{\varphi}_0} {\boldsymbol{\varphi}_1}^2 \right) \tag{A.23}
$$