IMPROVED OPERATIONAL SPACE CONTROL FRAMEWORK FOR COMPLIANT MOTION OF ROBOTIC MANIPULATORS

NGOC DUNG VUONG
B. Eng (Hons.), M. Eng, HCMC University of Technology, Vietnam

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2010

Acknowledgments

I would like to first express my gratitude to my supervisor, Prof. Marcelo H. Ang Jr, who added considerably to my graduate experience. His guidance, support and most importantly encouragements greatly influence my attitude on not only my research but also life. Also to my co-supervisor, Dr. Lim Ser Yong from the Singapore Institute of Manufacturing Technology (SIMTech), for all the interesting and long discussions. His hard but reasonable arguments are most appreciated because they did help me strengthen and widen my knowledge throughout my period of candidature.

I would like to thank Prof. Oussama Khatib from Stanford University, who laid the foundation of my research on my first year and continue to inspire my knowledge during his visiting in the later years. Thanks to Prof. Cezary Zielinski from Warsaw University of Technology for all his guidance on the MRROC++ framework during my attachment to his lab. Without his help, the real-time experimental results could never have been done this fast. Also thanks to Prof. Frank Lewis from University of Texas Arlington for all the discussions on the stability analysis of the dual-loop control structure.

The support from the Collaborative Research Project (CRP) between National University of Singapore and SIMTech is gratefully acknowledged. The attachment in SIMTech throughout my period of candidature brought me a lot of hands-on experience. Thanks to all my fellow members of the project: Mr. Lim Tao Ming, the programmer-guru of the Lab, for all enjoyable discussions, and Dr. Lim Chee Wang, the CRP's leader, for all the support during my attachment, Mr. Li Yuan Ping for helping me got started when I first joined the project, and Dr. Tao Pey Yuen for all the help on Latex.

Last but not least, I would like to thank my parents for everything that I have, and my

wife, Hang, who always supports me in everything since I decided to go for my PhD.

Contents

Summary

This thesis studies effects of model uncertainties on the force-based operational space control formulation. Although this control framework works perfectly in simulation, its performance is significantly degraded when faced with model uncertainties, as will be shown experimentally in this thesis.

Since the model plays an important role in the control framework, we first proposed a systematic procedure for identifying the robot dynamic model. To cater to the effects of the nonlinear joint friction, we suggested a simple and yet effective scheme to obtain a more accurate dynamic model. Experimental results on an actual industrial robot demonstrate the efficacy of our proposed procedure.

Using the identified dynamic model, it is shown that model uncertainties can produce different effects depending on the control space. The analytical results also suggest that the control space need to be chosen carefully in order to minimise the effects of model uncertainties on control performance. This is also one of the main reasons for the poor performance of the force-based operational space control.

The analyses raise a need of seeking for an alternative formulation to minimise the effects of model uncertainties while maintaining all the advantages of the force-based operational space control formulation. This is the main motivation for our proposed dual-loop operational space control structure. To justify the usefulness of the proposed control structure, intensive work on this control framework including stability analysis and real-time implementation on a real industrial robot have been carried out. Real-time experimental results have shown a significant improvement in comparison to the conventional approach.

Since compliant motion control capability is one of the key features of enlarging the applications of robots in real life, the proposed dual-loop control structure has been studied in a real application, the grinding application in the last chapter. Experimental results in this chapter revealed some potential issues that need to be addressed in future research.

Keywords: Compliant Motion, Robotic Manipulator, Model Identification, Operational Space Control, Singular Perturbation, Dual-loop Control Structure.

Nomenclature

$\bar{J}$	Dynamically consistency generalised inverse of J
$\boldsymbol{\mu}$	Task space Coriolis and Centrifugal vector
$\boldsymbol{\rho}$	Task space gravity vector
$\boldsymbol{\varphi}$	Task space disturbance vector
Λ	Task space inertia matrix
Λ_n^*	Inverse of null space inertia matrix
$\mathbf{\Gamma}$	Joint torque vector
$\mathbf{\Gamma}_{fric}$	Joint friction vector
$\mathbf{\Gamma}_{null}$	Join torque vector from desired null space tasks
$\mathbf{\Gamma}_{task}$	Join torque vector from desired task space tasks
$\mathbf{C}$	Joint space Coriolis and Centrifugal vector
$\mathbf{D}$	Joint space disturbance vector
$\mathbf{F}$	Control force vector at the operational point
$\mathbf{F}_{contact}$	Contact force vector at the operational point
$\mathbf{G}$	Joint space gravity vector
$\mathbf{h}$	Vector of the inertia parameters of the robot

$\mathbf{h}_b$	Vector of the base parameters of the robot
$\mathbf{q}$	Vector of joint space variables
$\mathbf{q}_n$	Vector of null space variables
$\mathbf{x}$	Vector of task space variable
J	Jacobian matrix of the operational frame expressed in base frame
J_n	Null space Jacobian
M	Joint space inertia matrix
S_n	Null space selection matrix

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Compliant Motion Tasks

Industrial robots have been used in various industries for nearly 50 years since General Motor introduced the first industrial robot, the Unimate, in 1961. Since then, there has been a steady increase in the use of robots in manufacturing [2]. One typical application of robots in the industrial environment is the pick-and-place task, i.e. robots are commanded to pick up an object from one location and place it to another along a predefined trajectory (Figure 1.1). Although this type of task is still commonly being used nowadays, there is an increasing interest on developing and applying the compliant motion control capability for industrial robots [3]. This can be observed from the fact that several big robot-manufacturing companies such as ABB and KUKA have been incorporating the force control capability into their new product lines in the last few years. With this additional capability, robots will be able to handle more complicated tasks such as screwing, deburring, grinding and so on. Figure 1.2 shows a typical example of these tasks, the grinding task that has been used as a case study in the last chapter of this work. In general, when tasks require the robot to interact with the environment, compliant motion control is a must-have capability. It is noted that the ability of sensing and controlling contact forces not only enables the robot to handle more tasks but also enables the robots to work in human environments where safety and cooperative ability are typically the two most important criteria.

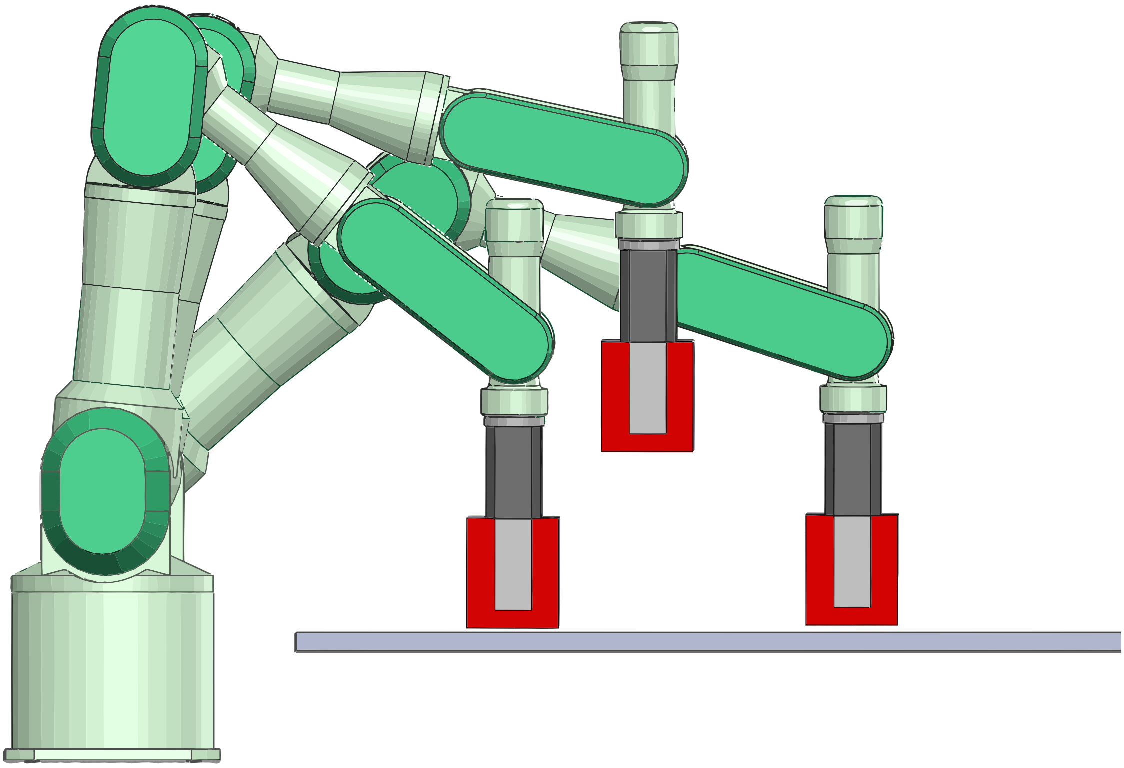

Figure 1.1: Pick-and-place task.

Figure 1.2: Grinding using the Mitsubishi PA10 manipulator.

1.2 State of the Art

There are two main approaches for handling interaction tasks ([4], [5]):

- **Indirect Force Control Approach**: in this approach, the desired motion and force are achieved by adjusting or controlling the mechanical impedance of the robot. The well-known stiffness/admittance control [6] and impedance control [7] are the typical examples of this approach. In these control schemes, the desired force in the compliant directions is realised by regulating the control parameters (e.g., the larger desired contact force, the higher robot stiffness is required). If the geometry of the working environment is perfectly known, high stiffness will be targeted for the free-motion directions to improve the motion tracking performance. However, since perfect knowledge is usually not possible, poor motion tracking accuracy is expected in practice [8]. If the dynamic model of the robot is available, it can be used to decouple the control system so that the robot impedance can be independently assigned. Moreover, if the stiffness of the environment is also known, it is possible to accurately resolve the required impedance for a desired contact force in the face of the absence of force/torque sensor [5].

- **Direct Force Control Approach**: this approach differs from the above indirect force control in the sense that the control loop is closed on the force errors rather than inferring the force errors from position/velocity errors. One typical example of this approach is the so-called hybrid motion/force control structure, which was first proposed by Craig [9]. This force control scheme is based on the observation that we can always decompose compliant tasks into task constraints [10]. Another notable control structure, which also belongs to the direct force control approach, is the inner-outer structure [11]. In this control framework, an external force feedback loop has been used to generate the position commands for the inner position controller. However, these above approaches fail to address the importance of the robot dynamics, which turns out to be critical to dynamically decouple the position and force in the operational space [12]. Since the control framework proposed

> by Craig does not take into account the dynamics of the end-effector, Khatib introduced the concept of task space dynamics as well as a control framework [13], the operational space control formulation, of the hybrid motion/force control for both non-redundant and redundant robots. The original operational space control framework does not consider the task prioritisation, which usually occurs when multi-tasks are needed to be achieved, as an important criterion. However, a recent work from the same group [12, 14] has extended the conventional operational space formulation to handle arbitrary prioritised task points. This new control framework can be regarded to be one of the most complete treatments for motion/force control of both non-redundant and redundant robots.

From the above discussion, it is clear that if motion/force tracking control performance is an important measured criterion, direct force control is preferable. Since most tasks using industrial robots require controlling the robots to follow a precise motion/force trajectory, this dissertation will mainly focus on the second compliant motion control approach, the direct force control approach.

1.3 Research Objectives

The robot dynamic model plays an important role in robot control. Especially, in the operational space control formulation, the dynamic model is used not only to linearise the nonlinear robot system but also to dynamically decouple the task space and the null space of redundant robots [15]. In this work, the term ”robot dynamic model” is considered to be comprised of two parts, the robot inertia and the nonlinear disturbances such as joint friction, motor dynamics and joint flexibility. Theoretically, the first part of the dynamic model, the robot inertia, can be computed from the robot CAD data. However, due to imperfections in the manufacturing and assembling process, the robot inertia is usually obtained through an identification process. The second part of the robot model, the nonlinear disturbances, on the other hand, is usually obtained using an empirical approach. It is worth pointing out that although parameter identification techniques for robot manipulators has received much attention in the robotics research community (for example [16–25] are a shortlist of researches that focused on identifying the robot dynamic model), the correctness of these identified results is hard to be justified because the robot inertia and the unknown disturbances are always coupled together. For instance, the work in [26] revealed that the identified dynamic model of the well-known PUMA 560 robot can vary significantly for different research groups although all these models were *claimed* to produce good results through experiments. This observation leads to a question of how a *reasonably good* model of any robot can be obtained for advanced control purpose. This is of critical importance since much research on advanced control for robotic manipulators is performed on the basis that the robot model is available to some degree of accuracy. Thus, a systematic procedure for obtaining a *usable* robot dynamic model is a must-have capability. This is also the first aim of this research.

Since a *perfect* dynamic model is inaccessible in practice, a proper controller should be designed to compensate for the unmodelled dynamics. Although much research has been done on designing such controllers for robotic manipulators, it should be noted that most of these studies only analyse the stability of the closed-loop system in the continuous domain [13, 14, 27–30]. In other words, the digitisation effects of the digital controllers are usually ignored in these studies. However, due to the fact that most robot controllers are digitally implemented, examining these digitised effects on the control performance has practical significance. Thus, the second aim of this work is to analyse the control performance of the operational space controller under the presence of model uncertainties and digitisation effects.

Although there are generally three types of operational space controllers (Chapter 2), this study mainly focuses on force-based operational space control because this control model can be considered as the most advanced control framework for both non-redundant and redundant robots. In order to maintain the advantages of the force-based operational space control, while still minimising the impacts of model uncertainties and digitised effects on the control performance, a new control structure, the dual-loop control structure, is proposed[1]. Note that the dual-loop control structure has been mentioned in some previous work such as [31–33]. However, in those studies, the robot dynamics is usually ignored at the outer-loop level i.e. the conversion from task space commands to joint space commands is usually done kinematically. On the other hand, our dual-loop operational space control (OSC) framework makes use of the robot dynamics at both levels. In other words, instead of using the robot model to linearise the system and to obtain the dynamic response in task space at the same time as in the conventional force-based OSC, the dual-loop OSC brings the inverse dynamics concept from task space to joint space (inner-loop), while the dynamic response is still obtained from task space (outer-loop). The reason for shifting the inverse dynamics concept from the operational space into joint space is to minimise the effects (if possible) of model uncertainties, as will be explained in detail in Chapter 4. To justify the usefulness of the proposed control structure, extensive experiments have been performed on the PA10 industrial manipulator. From the experimental results, it is shown that the dual-loop control structure with an inner inverse-dynamics loop can provide a considerably better control performance in comparison to the conventional force-based operational space control. Thus, the third aim of this work is to provide a detailed analysis of the proposed dual-loop control framework from both the theoretical and the empirical point of view.

The contributions of this PhD work are:

- A systematic procedure for identifying the robot dynamic model subjected to advanced model-based control. A simple and effective scheme was proposed to improve the quality of the identified dynamic model. The proposed identification method has been carried out on the PA10 industrial manipulator. Experimental results have shown significant performance improvement in comparison to conventional identification methods.

- Although the dynamic model can be used to linearise the nonlinear system of the robot in both joint space and task space, it is shown in this thesis that it is better

---

[1] We referred to the ”dual-loop operational space control” as the ”multi-rate operational space control” in our prior publications but ”dual-loop” is a more accurate term.

to use the imperfect dynamic model in joint space rather than in task space if a discrete control law is implemented. The validity of this observation has been shown in both simulation and experiment. This result is of crucial importance since it gives an explicit explanation of why the control performance of the force-based operational space control is significantly degraded in the presence of model uncertainties.

- Since model uncertainties always exist in practice, a new dual-loop operational space control structure has been proposed to better handle model uncertainties in comparison to the conventional operational space control framework. The proposed controller has been extensively studied based on both analytical and empirical points of view. Stability analysis is presented. Experimental results using this new controller scheme on an actual industrial robot, the PA10 manipulator, showed a great improvement in both motion and force control.

1.4 Thesis Outline

We first give a brief on the operational space control framework in Chapter 2. The following chapter, Chapter 3, is devoted to a detailed procedure to identify the robot dynamic model. Due to the importance of the robot model, a simple and yet effective scheme to obtain a more accurate dynamic model is proposed in this chapter. In Chapter 4, the effects of uncertainties on the operational space control are studied in detail. It will be shown that the model uncertainties can create different effects depending on the control space. Following this analysis, a dual-loop control structure is proposed in Chapter 5 to minimise the effects of model uncertainties while maintaining the advantages of the force-based operational space control framework. Stability analysis of the proposed controller as well as experimental results on an industrial manipulator, the Mitsubishi PA10 manipulator, is also presented in this chapter. Since the previous chapters mainly focus on motion control, Chapter 6 will be devoted to investigate the performance of the proposed controller in compliant motion tasks, the grinding task. Experimental results showed good performance can be achieved by adjusting the control gains. Potential problems are also highlighted in this chapter. Chapter 7 concludes this thesis by summarising the work done and contributions. Suggestions for future work are also presented.

Chapter 2

Compliant Motion Control Using Operational Space Control Framework

The main purpose of this chapter is to provide the necessary background theory for the readers who are not familiar with the operational space formulation, i.e. the force-based operational space control, which was first introduced by Khatib from Stanford University [34]. This chapter will first give a brief on the history of the operational space controllers. The force-based operational space control will then be explained in detail. A brief discussion on the source of the poor performance as well as some existing solutions for improving the control performance is also provided.

2.1 The Operational Space Controllers

One main motivation for creating robots is to help people perform some tasks. Intuitively, these tasks are specified in task space/operational space (as a sequence of the end-effector position and orientation) rather than in joint space. However, since most industrial robots are equipped with motion controllers at each joint, they lack the ability to directly resolve the task space commands. As a result, proper kinematic transformations need to be performed to translate tasks in the operational space into corresponding joint space instructions. Literally, there are two main approaches to resolve operational space tasks into joint space commands: the inverse-kinematics approach and the operational space control approach.

In the first approach, commands in the Cartesian coordinate are transformed into the corresponding joint space commands by performing the inverse kinematic transformation using either an analytical or a numerical approach. The joint commands are then executed in *joint space* by any joint space controller such as the well-known independent joint control or computed-torque control schemes in joint space where the robot dynamic model is used to linearise the system. Note that this approach fails to address other important aspects of the robot dynamics, which turns out to be critical in terms of defining the natural response of the robot [35]. For example, an important aspect that inverse dynamics control in joint space fails to address is the use of the inertia matrix to dynamically decouple end-effector dynamics (task space) from its internal motion dynamics (null space) for redundant manipulators (i.e., the joint space degree-of-freedom (DOF) is higher than the required task space DOF) [13]. As a result of the redundancy, there are infinite solutions for the inverse kinematic problem, thus, optimisation criterion should be imposed to get the optimum solution [36, 37]. A direct consequence of this add-on optimisation is that the computation becomes more intensive making it not suitable for real-time implementation.

On the other hand, the second approach involves *directly closing the control loop on the task space variables* i.e. pose and velocity of the operational point. Since it is hard to perform an explicit inverse kinematic transformation at the joint-position level, operational control approaches normally involve finding an equivalent joint space commands through the *inversion of the robot Jacobian*. Roughly speaking, operational space control approaches can be divided into three groups based on the way they handle the kinematic inversion. The first group resolves the kinematic inversion at the velocity level [38], while the second [36, 39] and the third [13] groups are based on the acceleration. The difference between the second and the third groups emerge when the robot is redundant. For a redundant robot, the task space and the null space are ”kinematically” decoupled in the second approach while they are dynamically decoupled in the third group. In other words, the third approach dynamically decouples the workspace into task space and null space by making use of the robot inertia matrix to weigh the pseudo-inverse solution. These two spaces are then separately controlled by model-based controllers. The third approach sometimes has been referred to as force-based operational space control [40] because of its dynamically-decoupling property.

Theoretically, force-based operational space control can be regarded as the most advanced control framework for redundant robots. One main reason is because it uses the robot inertia matrix to weigh the pseudo-inverse solution, thus, providing an optimal solution in the sense that the kinetic energy is minimised along the path [35]. In addition, this control framework can be used as a general framework for controlling redundant robots with many interesting features such as [14]:

- Motion and force can be simultaneously controlled through the hybrid control framework. By introducing the general selection matrix [13], tasks involving motion and force (or compliant motion tasks) can easily be achieved. Moreover, since this control framework is a force-based controller, i.e. it controls the so-called operational force at the operational point to achieve the goal, it is natural to extend this framework to handle tasks that require the robot to interact with the environment or with other robots (cooperative tasks) [34].
- Tasks can be prioritised in such a way that the higher priority tasks are always achieved first, while the lower priority tasks are executed separately without affecting the main tasks. In other words, the main tasks and sub-tasks can be totally decoupled. This capability is crucially important because it allows the users to incorporate some important constraints (as highest priority tasks) into the control framework. The remaining control degree of freedom can be used to perform other tasks without affecting the above important constraints. As a result, the whole robot body can be fully utilised in a dynamic manner (i.e., they are dynamically decoupled) [14].

In order to address the nonlinear effects due to link inertia, gravity and joint friction, Khatib [13] introduced the concept of task space dynamics where the joint space dynamics has been transformed into task space at the operational point. He also suggested a model-based PD controller to achieve exponential stability. It is worth noting that this exponential stability can only be assured when the robot model is *accurately known* and the controller is *continuously implemented*. However, since both assumptions are always violated in practice, the control performance can be significantly degraded due to the inaccuracy of the robot dynamic model.

Besides the modelling uncertainties, all operational space control approaches suffer from the singularity problem, which can be roughly described as losing the ability to move in certain directions. A direct consequence of this singularity issue is that the task space model of the robot becomes indefinite at singular configurations. Controlling the robot near/at a singularity is treated as a separate problem and will not be covered in this work. Interested readers can refer to [41] for a comprehensive review as well as details on robust singularity handling algorithms.

2.2 Force-based Operational Space Control

2.2.1 Background Theory

We consider that the robot manipulator is not operating at singular configurations. The joint space dynamic model of an n-DOF robot in contact with the environment (Figure 2.1) can be represented as:

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{G}(\mathbf{q}) + \mathbf{D}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{\Gamma}_{fric}(\dot{\mathbf{q}}, \mathbf{q}) + J^T\mathbf{F}_{contact} = \mathbf{\Gamma} \tag{2.1}$$

The equivalent task-space dynamics at any operational point can then be specified as follows:

$$\Lambda(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}(\dot{\mathbf{x}}, \mathbf{x}) + \boldsymbol{\rho}(\mathbf{x}) + \boldsymbol{\varphi}(\dot{\mathbf{x}}, \mathbf{x}) + \mathbf{F}_{fric}(\dot{\mathbf{x}}, \mathbf{x}) + \mathbf{F}_{contact} = \mathbf{F} \tag{2.2}$$

where $\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}$ are the generalised joint acceleration, joint velocity and joint position, $\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}$ are the generalised task space acceleration, task space velocity and task space position of an arbitrary operational point and $M, \Lambda, \mathbf{C}, \boldsymbol{\mu}, \mathbf{G}, \boldsymbol{\rho}, \mathbf{D}$ and $\boldsymbol{\varphi}$ are the inertia

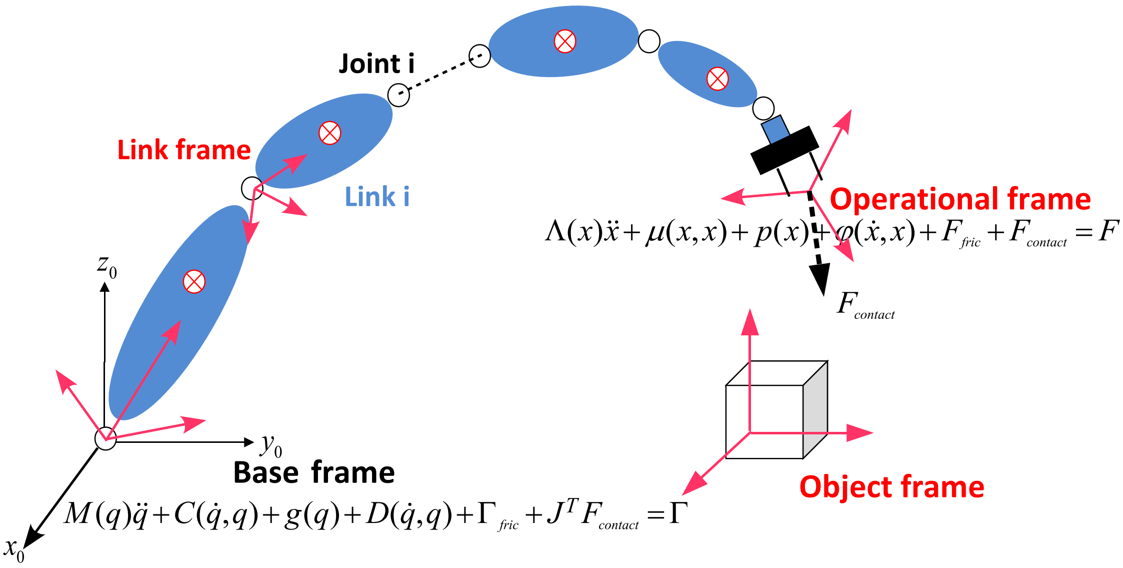

Figure 2.1: n-DOF robot.

matrix, Coriolis-Centrifugal, gravity and unknown disturbance in joint and task space, respectively. $\mathbf{F}_{contact}$ is the contact force acting at the operational point. $\mathbf{\Gamma}_{fric}$ and $\mathbf{F}_{fric}$ represents the friction torque at each joint and the equivalent resistance force caused by the joint friction in task space respectively. Note that the friction torque Γ_{fric-i} (the i component of $\mathbf{\Gamma}_{fric}$) is a local effect since it is assumed to depend only on the joint states (i.e. q_i and its derivative). However, the resistant force F_{fric} for any direction is a combination of the friction effects from all joints.

The relationship between the joint space and task space dynamics can be stated as follows:

$$\Lambda = (JM^{-1}J^T)^{-1} \tag{2.3}$$

$$\bar{J} = M^{-1}J^T\Lambda \tag{2.4}$$

$$\boldsymbol{\mu} = \bar{J}^T\mathbf{C} - \Lambda\dot{J}\dot{\mathbf{q}} \tag{2.5}$$

$$\boldsymbol{\rho} = \bar{J}^T\mathbf{G} \tag{2.6}$$

$$\boldsymbol{\varphi} = \bar{J}^T\mathbf{D} \tag{2.7}$$

$$\mathbf{F}_{fric} = \bar{J}^T\mathbf{\Gamma}_{fric} \tag{2.8}$$

$$\mathbf{\Gamma} = J^T\mathbf{F} \tag{2.9}$$

where J is the Jacobian associated with the desired operational point. The unified approach for motion and force control is then formulated as follows [13] (note that the superscript $\hat{(\_)}$ indicates that the associated symbol is an estimate of the symbol $(\_)$):

$$\mathbf{F}_{task} = \mathbf{F}_{motion} + \mathbf{F}_{force} \tag{2.10}$$

where:

$$\mathbf{F}_{motion} = \hat{\Lambda}\Omega\mathbf{u}_{motion} + \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\rho}} + \hat{\mathbf{F}}_{fric} \tag{2.11}$$

$$\mathbf{F}_{force} = \hat{\Lambda}\bar{\Omega}\mathbf{u}_{force} + \mathbf{F}_{sensor} \tag{2.12}$$

$$\mathbf{u}_{motion} = \ddot{\mathbf{x}}_d + K_{MD}(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + K_{MP}(\mathbf{x}_d - \mathbf{x}) \tag{2.13}$$

$$\mathbf{u}_{force} = K_{FP}(\mathbf{F}_d - \mathbf{F}_{sensor}) + K_{FI}\int(\mathbf{F}_d - \mathbf{F}_{sensor}) \tag{2.14}$$

The generalised task specification matrices Ω and $\bar{\Omega}$ are pre-defined depending on the tasks. $K_{MD}, K_{MP}, K_{FP}, K_{FI}$ and K_{FV} are the gains for motion and force control respectively. Note that Equations 2.10, 2.11, 2.12 linearise the task-space dynamics of the robot 2.2; thus, if $\hat{\Lambda} = \Lambda, \hat{\boldsymbol{\mu}} = \boldsymbol{\mu}, \hat{\boldsymbol{\rho}} = \boldsymbol{\rho}, \hat{\mathbf{F}}_{fric} = \mathbf{F}_{fric}, \mathbf{D} = \boldsymbol{\varphi} = \mathbf{0}$ and $\mathbf{F}_{sensor} = \mathbf{F}_{contact}$ (perfect model and sensing estimation) then the closed-loop system can be shown to be equivalent to n-double integrator:

$$\ddot{\mathbf{x}} = \mathbf{u} = \begin{bmatrix} \mathbf{u}_{motion} \\ \mathbf{u}_{force} \end{bmatrix} \tag{2.15}$$

and good control can be achieved by a proper choice of control gains [13].

For redundant manipulators, the dynamically consistent generalised inverse of the Jacobian matrix (Eq. 2.4) can be used to dynamically decouple the operational space to its null space as discussed in [13, 35, 42]. Note that tasks in the operational space are all transformed into the control force by Eq. 2.10, thus this control framework is sometimes referred to as the force-based operational space controller in the literature. The above control force is then transformed into joint space by:

$$\mathbf{\Gamma}_{task} = J^T \mathbf{F}_{task} \tag{2.16}$$

Since the task space controller need not make use of all DOF of redundant robots, the remaining DOF of the robot should be properly controlled by a null space controller. This null space controller can be formulated either in task space or joint space [15]. Since the work in this research mainly focuses on industrial robots which are usually constructed from joints in series (not tree structure), a joint space controller is preferable. The control of null space can be briefly described as follows [14].

Let us denote the set of joints that are controlled in the null space as:

$$\mathbf{q}_n = S_n \mathbf{q} \tag{2.17}$$

where S_n acts as the selection matrix. For example, to select joint 1 and 2 for the null space controller, S_n is:

$$S_n = \begin{bmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \end{bmatrix} \tag{2.18}$$

Define the null space Jacobian of the tasks as:

$$J_n = S_n(I - \bar{J}J) \tag{2.19}$$

The null space controller is then stated as:

$$\mathbf{F}_{null} = (\Lambda_n^*)^{\#}\mathbf{u}_n + (\Lambda_n^*)^{\#} S_n M^{-1}(\hat{\mathbf{C}} + \hat{\mathbf{G}} + \hat{\mathbf{D}} + \hat{\mathbf{F}}_{fric} - \mathbf{\Gamma}_{task}) \tag{2.20}$$

where the inverse of the null space inertia matrix Λ_n^* is [14]:

$$\Lambda_n^* = J_n M^{-1} J_n^T \tag{2.21}$$

The equivalent joint torque of the null space controller is:

$$\mathbf{\Gamma}_{null} = J_n^T \mathbf{F}_{null} \tag{2.22}$$

Thus, the total control joint torque becomes:

$$\mathbf{\Gamma} = \mathbf{\Gamma}_{task} + \mathbf{\Gamma}_{null} \tag{2.23}$$

By applying this control torque to the joint space dynamics (Eq. 2.1) under the assumption of perfect model estimation, the null space closed-loop equation is reduced to [14]:

$$\mathbf{q}_n = \mathbf{u}_n \tag{2.24}$$

which can also easily be stabilised by any conventional PD controller. It is worth pointing out that this null space controller reveals an important property of the force-based operational space in that it can *simultaneously achieve exponential stability* in *both task space and null space* under the assumption that *the robot model is available*.

The following simulation illustrates the above control framework. Consider a 3-DOF planar arm with three revolute joints (RRR) as shown in Figure 2.2. The main task here is to move the end-effector of the robot from $(x_i, y_i) = (1\text{m}, 1.5\text{m})$ by 0.5m in the x direction, i.e., $(x_d, y_d) = (1.5\text{m}, 1.5\text{m})$ in $t_f = 0.5$s. Quintic polynomial [43] is used for the trajectory planning. For instance, the desired trajectory $\ddot{\theta}_{di}, \dot{\theta}_{di}, \theta_{di}$ can be stated as:

$$\theta_{di}(t) = a_{0i} + a_{1i}t + a_{2i}t^2 + a_{3i}t^3 + a_{4i}t^4 + a_{5i}t^5 \tag{2.25}$$

$$\dot{\theta}_{di}(t) = a_{1i} + 2a_{2i}t + 3a_{3i}t^2 + 4a_{4i}t^3 + 5a_{5i}t^4 \tag{2.26}$$

$$\ddot{\theta}_{di}(t) = 2a_{2i} + 6a_{3i}t + 12a_{4i}t^2 + 20a_{5i}t^3 \tag{2.27}$$

where the coefficients $a_{0i}, a_{1i}, a_{2i}, a_{3i}, a_{4i}$ and a_{5i} can be computed from the initial con-

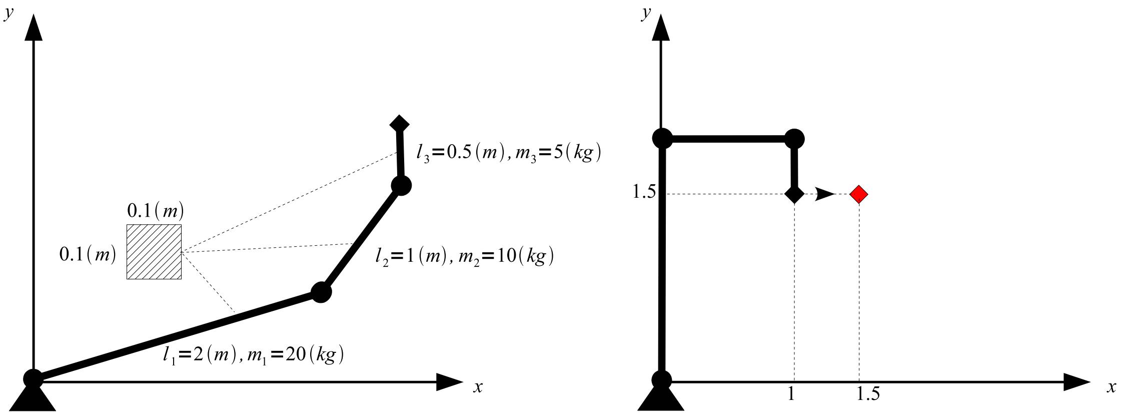

Figure 2.2: 3-DOF planar robot (three revolute joints). Each link is assumed to have the mass and shape as depict on the left. The figure on the right hand side shows the initial configuration of the robot $(q_1 = \frac{\pi}{2}, q_2 = -\frac{\pi}{2}, q_3 = -\frac{\pi}{2},)$ and the desired position of the end-effector: $(x_d, y_d) = (1.5m, 1.5m)$ (the red dot). This 3-DOF robot has been simulated using the SimMechanics Toolbox under Simulink environment. The integrator has been configured as *ode45*.

figuration θ_i, final configuration θ_f and the duration t_f as follows:

$$a_{0i} = \theta_i \tag{2.28}$$

$$a_{1i} = \dot{\theta}_i \tag{2.29}$$

$$a_{2i} = \frac{\ddot{\theta}_i}{2} \tag{2.30}$$

$$a_{3i} = \frac{20\theta_f - 20\theta_i - (8\dot{\theta}_f + 12\dot{\theta}_i)t_f - (3\ddot{\theta}_i - \ddot{\theta}_f)t_f^2}{2t_f^3} \tag{2.31}$$

$$a_{4i} = \frac{30\theta_i - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_i)t_f + (3\ddot{\theta}_i - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \tag{2.32}$$

$$a_{5i} = \frac{12\theta_f - 12\theta_i - (6\dot{\theta}_f + 6\dot{\theta}_i)t_f - (\ddot{\theta}_i - \ddot{\theta}_f)t_f^2}{2t_f^5} \tag{2.33}$$

Since this is a 2-DOF task, the robot is redundant in this case. The sub-task here is to control joint 3 from $-\pi/2$ to $-\pi/4$ ($q_{3d} = -\pi/4$) (also using quintic polynomial).

Assume that the robot dynamic model is available, by applying the above force-based operational space control framework with the task space control gains (Eq. 2.13): $K_p = 10^2 I_2, K_v = 20I_2$ and null space control gains (Eq. 2.20) $k_{pn} = 10^2, k_{vn} = 20$, where I_2 is the identity matrix of dimension 2×2. The task space and null space tracking error(s) (quintic polynomial for the trajectory planning) are shown in Figure 2.3. It is

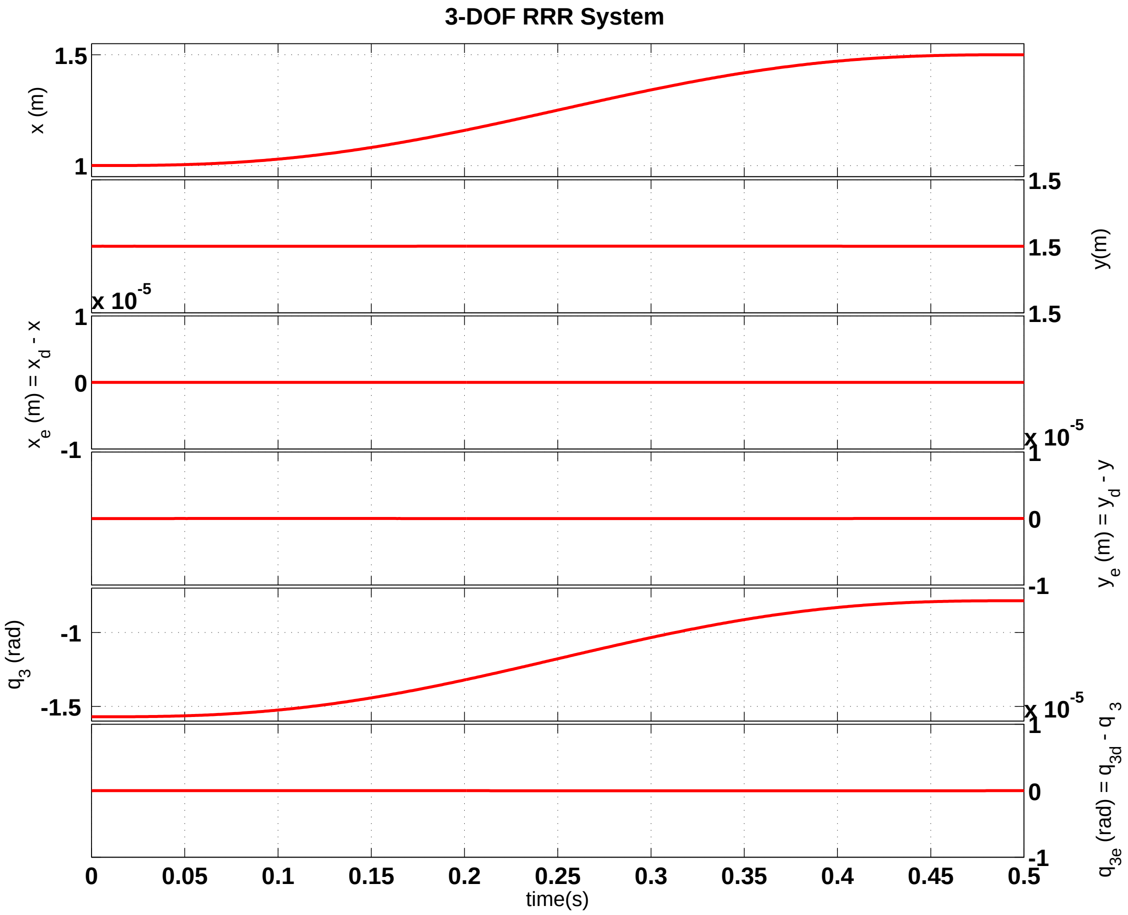

Figure 2.3: Task space and null space tracking errors of the 3-DOF planar robot for the given task. Note that quintic polynomial has been used for trajectory planning in this simulation. The top two graphs show the responses of the robot in (x, y) direction. The next two graphs show the tracking errors in task space. The performance of the null space control is shown in the last two graphs: the upper graph is the response of q_3, the bottom one is the tracking error of the null space controller.

clear from the simulation that the main task and the sub-task can be simultaneously achieved.

2.2.2 Model Uncertainties

It is worth stressing that the control performance in the above simulation can only be achieved under the assumption of perfect model estimation. In practice, complete knowledge about the robot model is hard to achieve. Further investigation on the robot dynamic model indicates that joint friction in the low-speed region is extremely complicated [44–47].

Although friction modelling is well documented in the literature, a perfect friction model, which can capture all the non-linear effects, especially friction in the low-velocity region, is still an open problem. In this region, friction is known to be highly nonlinear, with the hysteresis effect, Stribeck effect, position and/or time dependency and other nonlinearities [46]. From the literature, it seems that the LuGre friction model [44] is one of the most popular models that can be used to capture most of the above effects except the position/time dependence effects. However, the implementation of these friction compensator schemes often requires *reference joint velocities* and/or *good joint velocity estimation*. From the practical point of view, these requirements are not easily fulfilled because there are *no explicit reference joint velocities* for the case where the motion and force *are controlled in task space*. Moreover, since most industrial robots do not have joint velocity sensors, the feedback joint velocities are normally obtained through *the numerical differentiation of joint position measurements*. As a result, errors in the velocity estimation in low velocity region will *be amplified* by the friction compensators due to velocity dependence of the Stribeck and Coulomb friction models [48]. In other words, the poor performance of the friction compensator not only comes from the inaccuracy of the friction model but also comes from the noisy estimated velocity. Also note that the remaining unmodelled dynamics is transformed into task space by Eq. 2.7, which involves the estimated inertia matrix and the Jacobian. Thus, the same amount of uncertainty in joint space may reproduce much different effects in the task space dynamics, depending on the robot configuration. Since the force-based operational space control (OSC) closes the control loop in task space through the kinematic chains as shown in Eq. 2.11, the accuracy of the task space inertia matrix plays an important role on the control performance. In fact, if the task space inertia matrix is not accurate, the performance of the force-based OSC can be worse than other simpler task space controllers as pointed out in [40]. For example, Figures 4.6-4.9 (Chapter 4) shows the task space tracking errors when faced with model uncertainties for various control gains. A detailed discussion on how model uncertainties affect the control performance will be given in Chapter 4.

2.2.3 Solutions for Model Uncertainties

In view of the above discussion, it is clear that model uncertainties always exist in practice. Hence, a proper study needs to be performed to tackle this model uncertainty issue. In order to overcome the problem, a number of approaches, such as robust control and adaptive control approaches have been proposed in the literature. The main difference between the robust control approaches and adaptive control approaches is the way they process the uncertainties. The robust control approaches use the feedback signal to reject the uncertainties by *using a fixed control structure* [28,49–52], while the adaptive control approaches use the feedback signal to improve the quality of the control model [53–56] by *online adjusting control parameters*.

As mentioned above, robust control approaches do not change the control structure, thus they can only give an ”acceptable” performance. However, their advantage is that they do not require much tuning efforts as compared to the adaptive control approaches [49]. To have a fixed structure despite the model uncertainties, high control gains are normally used in robust control approaches. As a result, chattering can occur due to the signal noise and the discretising effects. For instance, the work in [57] showed that instability can occur when the control gains are above some threshold values. Although most robust controllers do not require perfect knowledge of the robot model, they often make use of the bounds of system parameters to define the control gains. Note that most robust control approaches are shown to be stable only in the continuous domain. As a result, it is still not clear whether the digital implementation of these control laws will have any adverse effect on the control performance.

Unlike robust control, adaptive control approaches attempt to ”learn” the uncertainties in order to improve the control performance [53]. Theoretically, adaptive control is supposed to be the best controller for any system in question due to its learning ability. However, because of the learning law, the complexity of these control laws is much more in comparison to the robust control cases. As a result, adaptive control may become impractical when the robot degree of freedom is high. Another disadvantage of the adaptive control approaches is that, to guarantee the convergence of the control law, the linear model assumption is often used in the stability analysis and control design. In practice, the linear model assumption may not be valid because of all the nonlinear effects discussed before; thus, the parameters of the adaptive control law can converge to a non-physical value [58].

Note that research in dynamic control of robotic manipulators has usually focused on the joint space control formulation since the joint space dynamics has many interesting properties such as the positive definiteness of the joint space inertia matrix or the linear-in-parameters of the joint space dynamic model [59,60]. Research in task space dynamic control, on the other hand, has mainly focused on the robust control approach [61]. One possible reason is that the linear-in-parameter property of the task space dynamics cannot be extended in a straightforward manner, especially for redundant robots. In other words, for redundant robots, two separate controllers (task space and null space controller) are needed to be considered for analysing the convergence of the estimated parameters. Moreover, since measurement noise and high frequency unmodelled dynamics always exists in practice, the robustness of the convergence of adaptive controllers is still an open problem [61]. On the other hand, robust control approach in task space has received more attention [12, 62–64] although the majority of work is still developed in joint space [61]. Note that while the same robust control concept can be applied in both joint space and task space for non-redundant robots, this observation is no longer true for the redundant case. The main reason is that the task space and null space dynamics of redundant robots are heavily coupled through the dynamic model [13, 40, 65]. As a result, the dynamics in one space (null space for example) can significantly affect the control performance of other space (task space for example).

In this work, instead of seeking for a new robust/adaptive controller to cope with model uncertainties, we will look at the uncertainty issue from a different perspective, that is, *what is the best way to use* the identified dynamic model in order to *minimise* the effects of model uncertainties on the control systems and *maximise* its usefulness in the operational space formulation [13,14]. Note that although only the standard PD controller is used in most of the experiments in this research, there is no limit to applying all known joint space controller schemes (both adaptive and robust controllers) to the proposed control structure in Chapter 5. In fact, since the inverse dynamics has been shifted from task space to joint space, our control framework can *benefit much* from the vast amount of work on the joint space control formulation. However, considering those control schemes here is out of the scope of this research and will be addressed in future work.

Chapter 3

Identification of Rigid Body Dynamics of an Industrial Robot

As shown in Chapter 2, the robot dynamic model is required in the implementation of the force-based operational space control scheme. The dynamic model is crucial because it is used to linearise the nonlinear robotics system in both joint space [66] and task space [13]. Even in some non-model-based control schemes, a rough estimation of the robot model is also important because it can simplify the gain-selection process [57]. Since the robot dynamic model is normally not available, proper procedures need to be carried out to identify these parameters. If the robot can be considered as a combination of multi-rigid bodies, an obvious way to identify its inertial parameters is to dismantle the robot and measure link-by-link [19]. Another approach is to make use of the CAD model (if available) to compute inertial parameters of the robot. However, it is obvious that these approaches are not always feasible in practice. In addition, the above two approaches cannot account for the effects of joint friction and other nonlinear dynamics.

In order to account for joint friction, several methods have been proposed. These methods can be roughly divided into two groups: to identify joint friction and rigid body dynamics separately [67,68] or to identify joint friction and rigid body dynamics simultaneously [18,20,23]. The former first identifies the friction parameters for each joint and then continues to identify the robot inertia by making use of the identified friction parameters. Since friction parameters are identified joint by joint, nonlinear dynamic friction models such as Stribeck and/or hysteresis effects can be easily incorporated. The main drawback of this method comes from the observation that friction can be time-dependent [23]. Thus, the error in friction compensation can affect the quality of the identified robot inertia. Moreover, friction forces/torques are always coupled to the inertial forces/torques. Thus, one cannot be accurately identified without the other. It is also argued that it is more tedious to identify friction parameters and rigid body dynamic parameters separately.

From the literature, more researchers adopt the latter method, i.e., to identify joint frictions and the rigid body dynamics at the same time. Note that the robot inertia can be linearised w.r.t its parameters. Thus, many proposed identification methods have been based on the assumption that joint frictions can also be modelled in a linear-in-parameter form so that the robot dynamic model can be fully linearised. However, this linearity is not valid for all velocities. In the low velocity region, the friction parameters exhibit some dynamics, and we refer to this region as the ”dynamic” region of friction. When velocities exceed a threshold velocity, the friction parameters become ”static” and the friction is now linear in the parameter form. We therefore refer to this region as the ”static or linear” region. The use of the linear friction model outside this linear region can lead to significant errors in the identified parameters as the assumption of a linear dynamic model is no longer valid. In this chapter, a simple and effective scheme will be introduced to identify the threshold/boundary velocity that separates the joint friction into dynamic (and nonlinear) and static (and linear in parameter) regions. Based on the identified threshold velocity, the robot dynamic model is then identified only in the linear region, thus more accurate dynamic parameters are obtained [16].

As the above discussion, the robot dynamic model is linear w.r.t its parameters, these parameters can be identified using the well-known least-squares estimator. Note that not all ten inertial parameters of each robot link can be identified due to the relative configuration of the links of the robot (they need not all be identified for control purpose). It is therefore necessary to reduce/simplify the robot model to ensure that the observation matrix of the least-squares estimator has full rank [69]. This problem can be solved either symbolically [69] or numerically [70].

Theoretically, the robot dynamic model can be accurately resolved by the least-squares estimator once enough motion data (i.e. $\{\ddot{\mathrm{q}}, \dot{\mathrm{q}}, \mathrm{q}\}$) and joint torque are acquired. In practice, since the measured torques is normally noisier than the measured position, a proper experiment should be designed to ensure the robustness of the identified results [71]. To guarantee the robustness of the estimation process, several criterions have been proposed in the literature such as maximising the determinant or minimising the condition number of the observation matrix [18], maximum the likelihood [72] and so on. Note that experimental design under the above criteria results in solving a nonlinear constrained optimisation problem. The results from this optimisation problem are the so-called exciting/optimal trajectory that can guarantee the excitation and robustness of all the parameters to be identified. Because of the complexity of the dynamic model, a good guess for the initial condition is hard to achieve. Thus, we have proposed a genetic algorithm (GA) to find the above optimal trajectory [17].

Note that the above exciting trajectory can only account for the uncertainties of the measured torque if least-squares estimator is used [21, 61]. In practice, uncertainties can also occur in the motion data (i.e., joint position, velocity and acceleration). Due to the fact that most industrial manipulators do not come with velocity and acceleration sensor, this information are normally obtained through numerical differentiation of the joint position measurements. As a result, the accuracy of the observation matrix of the least-squares estimator is compromised. A direct consequence of this observation is that the results from the least-squares estimator can be much deviated from their true value [73]. Since no constraints are imposed on the least-squares technique, it is possible for the least-squares estimator to produce results which are physically impossible [74, 75]. Although there are other methods to cope with uncertainties on the observation matrix such as the maximum likelihood method [72], most of them do not consider the physical feasibility of the identified parameters as an important criterion. Note that a physically non-feasible dynamic model cannot be used in most model-based control schemes because this model can result in a non-positive definite inertia matrix, thus, destabilise the closed loop control system. One promising solution for this problem is to use constrained optimisation tools to adjust the least-squares result [76]. However, this method requires the initial guess of the virtual parameters which are not always available in practice.

Although there is a vast amount of results on the model identification topics of robotic manipulators in the literature, a systematic procedure, which includes all the above considerations, is still missing. Thus, we develop a systematic procedure for identifying the robot dynamic model that can then be used in advanced model-based controllers [77].

3.1 Modelling

3.1.1 Base Parameters

It is well known that the dynamic model of an n-degree-of-freedom (n-DOF) serial manipulator can be expressed in the following analytical form [9]:

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{G}(\mathbf{q}) + \mathbf{\Gamma}_{fric} = \mathbf{\Gamma} \tag{3.1}$$

where:
- $\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}^1$ are $n \times 1$ vectors of joint acceleration, velocity and position, respectively.
- $M(\mathbf{q})^2, \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}), \mathbf{G}(\mathbf{q})$ are the inertia matrix, Coriolis-Centrifugal and gravity vector in joint space.
- $\mathbf{\Gamma}_{fric}$ is $n \times 1$ vector of joint friction and $\mathbf{\Gamma}$ is $n \times 1$ vector of force/torque at each joint.
For identification purpose, the above equation is re-written in the linear form [59]:

$$W(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, DH)\mathbf{h} + \mathbf{\Gamma}_{fric} = \mathbf{\Gamma} \tag{3.2}$$

---

[1] Bold-face block capital letter represents a vector quantity

[2] Block capital letter represents a matrix quantity

Here, DH is the kinematic parameters from the Denavit-Hartenberg parameters and $\mathbf{h}$ is a $10n \times 1$ vector of the inertial parameters:

$$\mathbf{h}_i = [xx_i, yy_i, xz_i, yy_i, yz_i, zz_i, mx_i, my_i, mz_i, m_i]^T \tag{3.3}$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 & ... & \mathbf{h}_n \end{bmatrix}^T \tag{3.4}$$

where $(xx_i, xy_i, xz_i, yy_i, yz_i, zz_i)$ are the inertial tensor of link i, (mx_i, my_i, mz_i) are the first moments and m_i is the link mass. Note that, here, we only focus on the inertial parameters of the links. The rotor inertia of motors are assumed to be known because these values are normally available from the motor specs.

Note that not all the inertial parameters contribute to the dynamic behaviour of the robot [69]; thus, a set of identifiable parameters (the so-called base parameters) should be deduced from Eq. 3.4. For instance, the original dynamic parameters of the 7-DOF Mitsubishi PA10 manipulator $\mathbf{h}$ has 70 parameters but the final identifiable dynamic parameters (excluded joint friction parameters) of the manipulator is reduced into 18 lumped-parameters [17]. By taking into account the base parameters, Eq. 3.2 becomes:

$$W_b(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, DH)\mathbf{h}_b + \mathbf{\Gamma}_{fric} = \mathbf{\Gamma} \tag{3.5}$$

where $\mathbf{h}_b$ is the base parameters. From Eq. 3.5, it is clear that if the joint friction model is linear with respect to its parameters, the problem of identifying the dynamic model is a linear problem. In the next section, the condition for which the linear-in-parameter friction model is valid will be derived.

3.1.2 Boundary Velocity and Linear Friction Model

Although joint friction is complex in reality, a simple model, which is the combination of viscous and Coulomb friction, is normally used to describe the friction phenomenon for all joint(s):

$$\tau_{fric\_i} = f_{ci} sign(\dot{q}_i) + f_{vi}\dot{q}_i \tag{3.6}$$

where f_{ci} and f_{vi} are Coulomb and viscous friction coefficients of joint i respectively. However, this assumption can be invalid in the low velocity region as shown in many works [44, 78, 79]. By analysing the experimental data and the dynamic friction effects described by the LuGre model [46], it is found that there is a threshold/boundary velocity that separates the dynamic friction into nonlinear and linear regions. In the nonlinear region, the friction effects mainly consist of stiction, Stribeck and hysteresis, while in the linear region, the dynamic friction converges to only Coulomb and viscous frictions (these two friction terms are known to be linear w.r.t. their parameters). Thus, by analysing the velocity-torque map, one should be able to identify the boundary velocity for each joint.

In order to get the insight of this approach, let us first study the LuGre friction which can be considered to be one of the most popular friction models that can be found in the literature. From the literature, The LuGre model is known to be able to capture most of the above effects except the position/time dependence effect. The mathematical model of the LuGre model can be described as follows:

$$\tau_{LuGre} = \sigma_1 \dot{z} + \sigma_0 z + f_v \dot{q} \tag{3.7}$$

where z describes the average deflection of bristles (please refer to [46] for detailed discussions):

$$\dot{z} = \dot{q} - \frac{\sigma_0 |\dot{q}|}{v(\dot{q})} z \tag{3.8}$$

The function $v(\dot{q})$ characterises the Coulomb and Stribeck effect and is commonly chosen as:

$$v(\dot{q}) = f_c + (f_s - f_c) e^{-(\frac{\dot{q}}{v_s})^2} \tag{3.9}$$

where f_c, f_s and v_s are the Coulomb, Stribeck and Stribeck velocity accordingly. One observation that can be made from Eq. 3.9 is that if $|\dot{q}| > \frac{3}{\sqrt{2}} v_s$, $e^{-(\frac{\dot{q}}{v_s})^2}$ goes to 0 approximately, Eq. 3.9 becomes: $v(\dot{q}) = f_c$ and Eq. 3.8 reduces to:

$$\frac{f_c}{\sigma_0 \dot{q}} \dot{z} \pm z = \frac{f_c}{\sigma_0} \tag{3.10}$$

Also note that Eq. 3.10 is actually a first-order system w.r.t the non-measurable state z. The time constant of this system is reduced when $\dot{q}$ is increased. This observation implies that $\dot{z} \rightarrow 0$ for any non-zero initial state. In other words, when $|\dot{q}| > \frac{3}{\sqrt{2}}v_s$, the z state will quickly converge to its static state (SS):

$$\dot{z}_{SS} \simeq 0 \Rightarrow z_{SS} \simeq \frac{sgn(\dot{q})v(\dot{q})}{\sigma_0} \tag{3.11}$$

and the LuGre friction model (Eq. 3.7) becomes:

$$\begin{cases} |\dot{q}| > \frac{3}{\sqrt{2}}v_s \\ \dot{z}_{SS} \simeq 0 \\ z_{SS} \simeq \frac{sgn(\dot{q})v(\dot{q})}{\sigma_0} \end{cases} \Rightarrow \tau_{LuGre} \simeq sgn(\dot{q})f_c + f_v\dot{q} \tag{3.12}$$

This observation leads to a conclusion that the nonlinear dynamic friction reduces to the linear form (Coulomb and viscous friction model) when joint velocity $|\dot{q}| > \frac{3}{\sqrt{2}}v_s$. The above observation can also be verified through the following simulation.
Consider a one-link system ($i_{zz} = 1$kgm^2/rad) system under the LuGre friction effect ($\sigma_0 = 10000, \sigma_1 = 316, f_v = 0.4, f_c = 1, f_s = 1.5, v_s = 0.1$). By increasing the applied torque to the system $\tau = 0.3t + 2sin(3t)$, the system response is presented in Figure 3.1. It is clear from Figure 3.1 that, when joint velocity $|\dot{q}| > \frac{3}{\sqrt{2}}v_s = 0.212$ (since the applied torque is increasing over time), $\dot{z} \rightarrow 0$ and $z \rightarrow 0.0001 = 10^{-4}$.

It is important to stress out that the existence of the above boundary velocity need to be verified on the real manipulator since much complicated joint friction is found in the actual manipulator. The following section describes the step-by-step procedure for finding out the boundary velocity (if any) for each joint.

- **Step 1**: Mount the manipulator in such a way that the gravity has no effect on the torque of the joint of interest. For example, consider the PA10 manipulator in Figure 3.2, by mounting the manipulator at two different configurations: vertical and horizontal configuration, gravity effect on the joint torque can be easily eliminated.

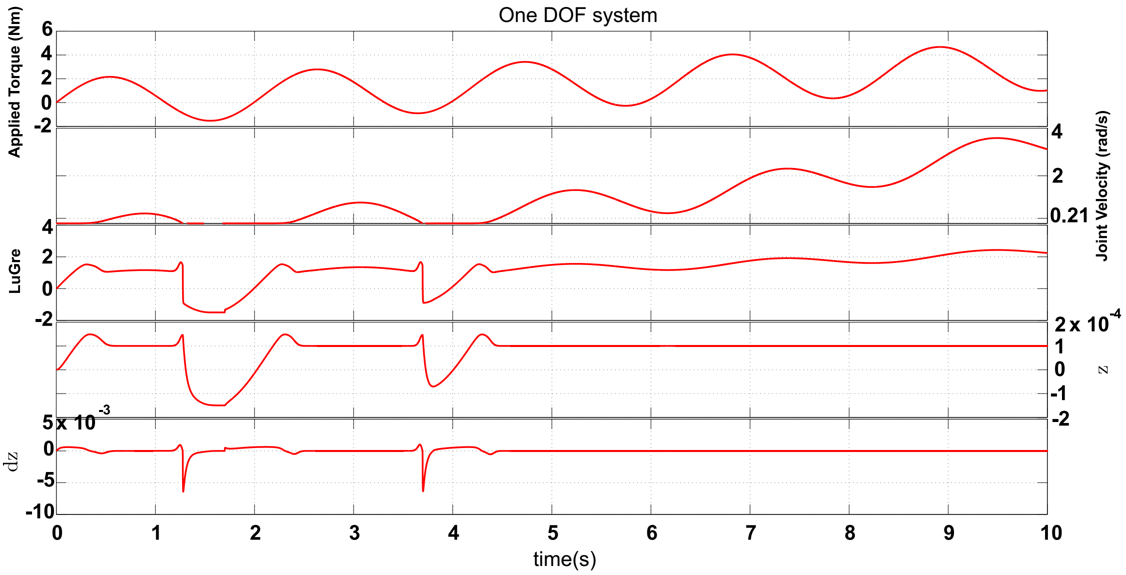

Figure 3.1: One-link system under LuGre friction (gravity free). From top to bottom: applied torque (Nm), joint velocity (rad/s), friction torque from the LuGre model (Nm), the internal state z and $\dot{z}$. This simulation has been done under the 20sim environment (*www.20sim.com*).

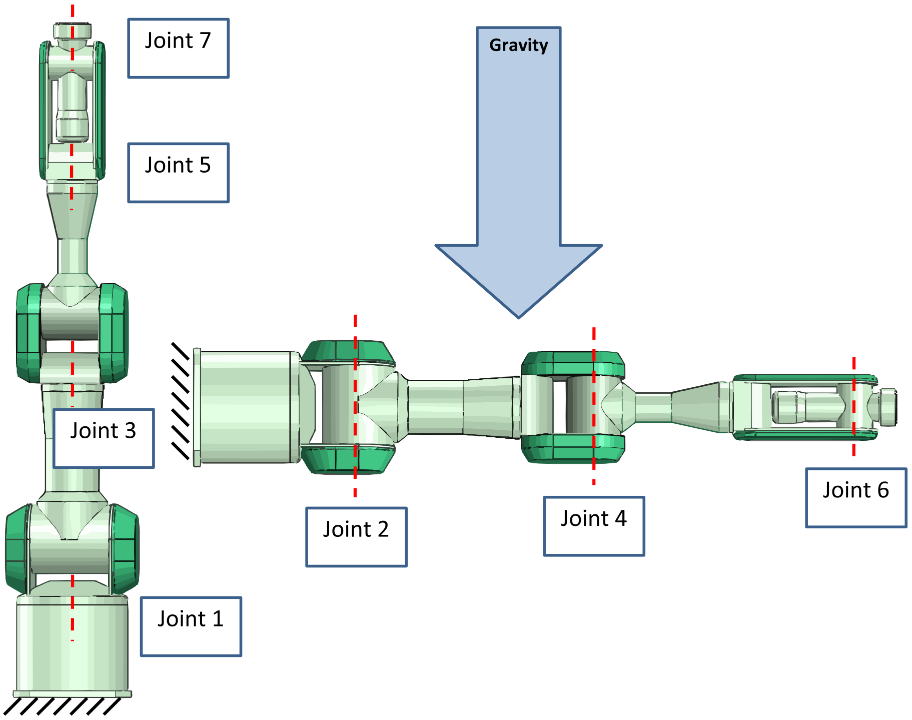

Figure 3.2: Eliminate the gravity effect by mounting the manipulator at different configurations.

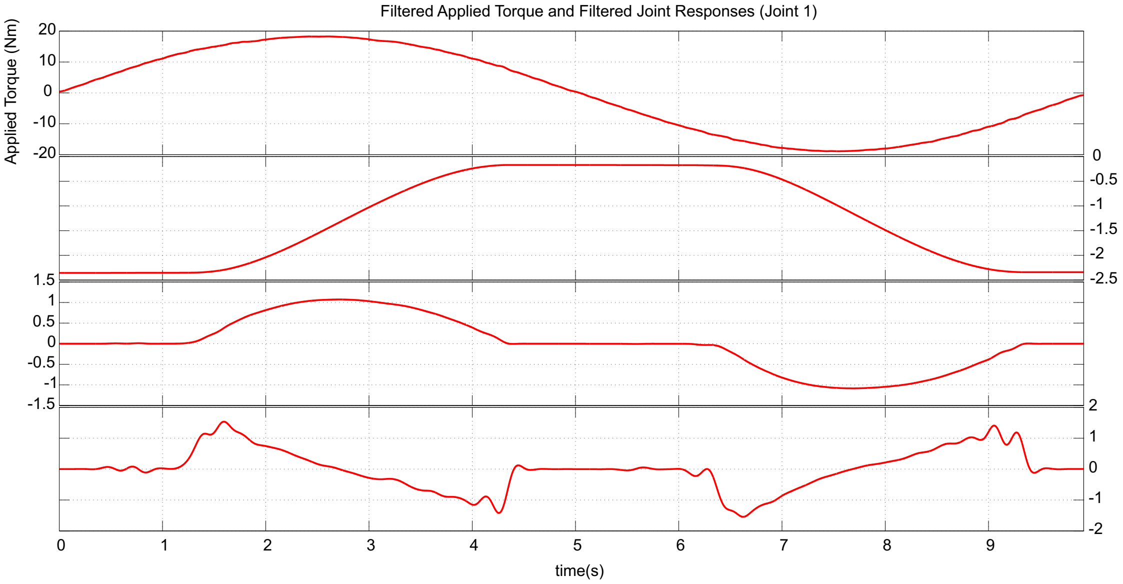

Figure 3.3: Response of the first joint of the Mitsubishi PA-10 for a sinusoidal torque. From top to bottom: applied torque (Nm), the responses of joint 1 $\{q_1, \dot{q}_1, \ddot{q}_1\}$. Note that the applied torque is clean because this torque has been feed-forward to the joint amplifier. Since only joint position (q_1) is available for measurement, joint velocity and acceleration have been obtained off-line using the central difference with zero-phase shift filter.

To excite joint friction, a sinusoidal torque can be applied to each joint. Notice that the frequency and magnitude of this signal have to be chosen in such a way that the resulting joint motion is within the joint limit and the motion also excites the dynamic friction. During this step, $(q_i, \dot{q}_i, \ddot{q}_i, \tau_i)_{1..n}$ are recorded (n is the number of recorded points). For instance, the responses of joint 1 of the PA10 for an open-loop sinusoidal torque are depicted in Figure 3.3.

Since only one joint is excited at a time, if joint friction can be modelled as Coulomb and viscous model, the equation of motion of the system becomes:

$$i_{zzi}\ddot{q}_i + f_{ci}sign(\dot{q}_i) + f_{vi}\dot{q}_i = \tau_i \tag{3.13}$$

where i_{zzi} is the lumped inertia at the current configuration, f_{ci} and f_{vi} are Coulomb and viscous friction coefficients of the joint of interest. Clearly, if Eq. 3.13 can describe the dynamic behaviours at joints, $(i_{zzi}, f_{ci}, f_{vi})$ can be resolved from the following linear system:

$$\begin{bmatrix} \ddot{q}_{i1} & sign(\dot{q}_{i1}) & \dot{q}_{i1} \\ \dots & \dots & \dots \\ \ddot{q}_{ir} & sign(\dot{q}_{ir}) & \dot{q}_{ir} \end{bmatrix} \begin{bmatrix} i_{zzi} \\ f_{ci} \\ f_{vi} \end{bmatrix} = \begin{bmatrix} \tau_{i1} \\ \dots \\ \tau_{ir} \end{bmatrix} \tag{3.14}$$

where $r \leq n$ is the number of data points which are used for the identification.

- **Step 2**: slowly increase $\dot{q}_{iThres}$ from 0 to $max|\dot{q}_i|$, solve the linear system of equation 3.14 for the parameters $(\hat{i}_{zzi}, \hat{f}_{ci}, \hat{f}_{vi})$ using only the data points for which $|\dot{q}_i| \geq \dot{q}_{iThres}$. Since the inertia i_{zzi} is constant at the current configuration, if Eq. 3.13 can properly describe the dynamic behaviours of joint i, the estimation of the inertia, $\hat{i}_{zzi}$, should converge to a fixed value regardless the amount of used data points. By analysing the convergence of the inertial parameter $\hat{i}_{zzi}$, one can experimentally find out the region in which the linear friction model is held. Based on this result, we can actually reconstruct the joint velocity vs friction torque graph (or friction map) for each joint. For instance, Figure 3.4 shows the convergence of $(\hat{i}_{zzi}, \hat{f}_{ci}, \hat{f}_{vi})$ of the first joint of the PA10 manipulator. As is seen, when $\dot{q}_{Thres} > 0.3(rad/s)$, $\hat{i}_{zzi}$ converges to around 1.67. This observation implies that when $|\dot{q}_1| > 0.3(rad/s)$, Eq. 3.13 can be used to represent the dynamic behaviour of the system. In other words, the linear friction model is only valid when joint velocity is outside the range $[-0.3, 0.3](rad/s)$. Based on the estimated inertia $\hat{i}_{zzi}$, the friction map of joint 1 of the PA10 manipulator is shown in Figure 3.5. From the friction map, it is clear that joint friction in the low-speed region is complicated and cannot be solely modelled by Eq. 3.13.

- **Step 3**: The experiment is then repeated for the rest of the joints. The resulting $\dot{q}_{iThres}$ for the first four joint of the PA10 are shown in Table 3.1. These values will be used as constraints in designing the exciting trajectory as presented in the next section.

In summary, if joint velocity is outside the range $(-\dot{q}_{Thres}, \dot{q}_{Thres})$, joint friction can

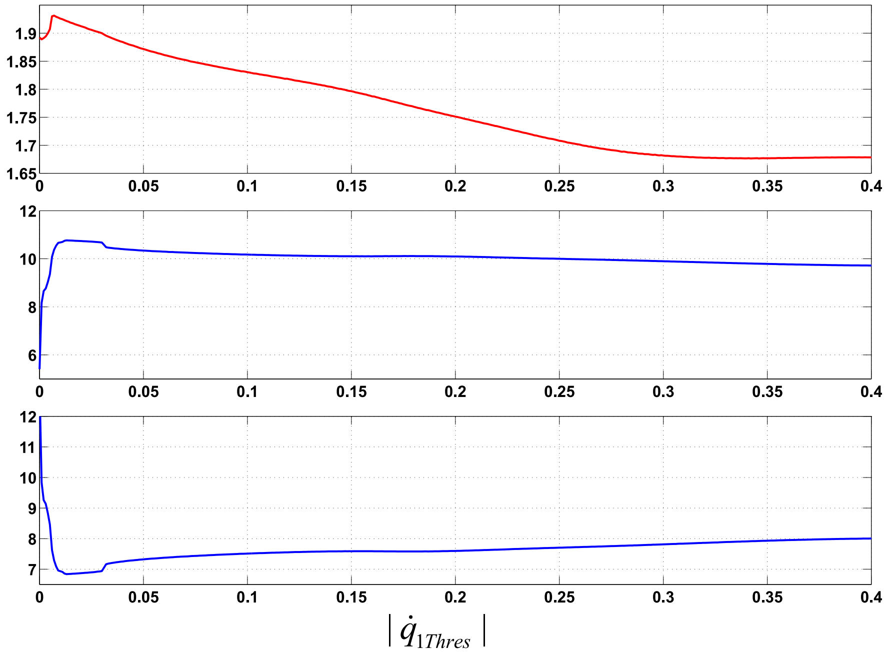

Figure 3.4: Parameter convergence of joint 1 of the PA10 (x_{axis} is $|\dot{q}_{1Thres}|$, y_{axis} is the estimated parameter). From top to bottom: estimated inertia $(\hat{i}_{zz1})$, estimated Coulomb friction coefficient $(\hat{f}_{c1})$ and estimated viscous friction coefficient $(\hat{f}_{v1})$.

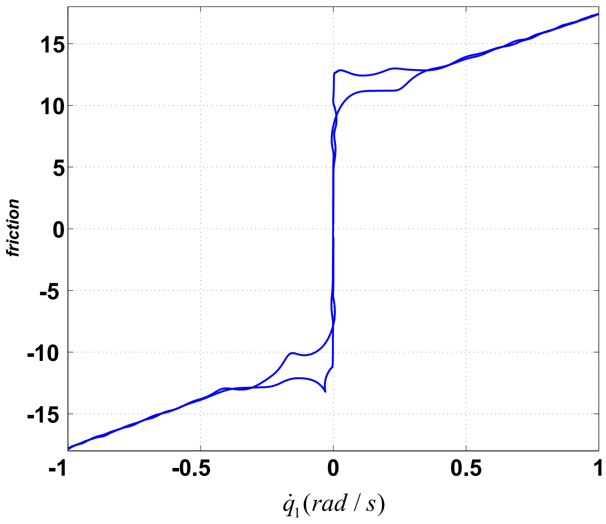

Figure 3.5: Velocity-Friction map of joint 1 of the PA10. This friction map is obtained by making use the estimated inertia $\hat{i}_{zz1}$ from the above analysis i.e. $\tau_{friction} = \tau - \hat{i}_{zz1}\ddot{q}_1$.

be modelled as a combination of Coulomb and viscous friction (Eq. 3.6). By incorporating Eq. 3.2 and 3.6, the robot dynamic model can be rewritten as:

$$W_c(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, DH)\mathbf{h}_c = \mathbf{\Gamma} \tag{3.15}$$

where W_c and $\mathbf{h}_c$ are:

$$W_c = \begin{bmatrix} W_{1,:} & sign(\dot{q}_1) & \dot{q}_1 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ W_{n,:} & 0 & 0 & ... & sign(\dot{q}_n) & \dot{q}_n \end{bmatrix}, \mathbf{h}_c = \begin{bmatrix} \mathbf{h}_b \\ f_{c1} \\ f_{v1} \\ ... \end{bmatrix} \tag{3.16}$$

Theoretically, by resolving Eq. 3.15, one can accurately estimate the inertial parameters $\mathbf{h}_c$ provided that the observation matrix W_c and the joint torque $\mathbf{\Gamma}$ can be accurately obtained. In practice, these assumptions are always violated since the measured torque and motion data are usually corrupted by noise. As a result, the identification experiment should be designed in such a way that the results from the least-squares estimator are robust w.r.t the signal noise. This observation leads to the discussion in the next section: the design of the exciting trajectory.

3.2 Experimental Design

3.2.1 Optimum Trajectory

In order to estimate $\mathbf{h}_c$ from Eq. 3.15, $(W_c, \mathbf{\Gamma})$ need to be acquired through the identification experiments. By stacking the matrix together, the observation matrix can be formed as follows:

$$W_o = \begin{bmatrix} W_{c1} \\ ... \\ W_{ck} \end{bmatrix}, \mathbf{\Gamma}_o = \begin{bmatrix} \mathbf{\Gamma}_{c1} \\ ... \\ \mathbf{\Gamma}_{ck} \end{bmatrix} \tag{3.17}$$

where k is the number of data points. Theoretically, as long as the determinant of $W_o^T W_o$ (which depends on the trajectory that has been used in the identification experiment) is non-zero (i.e. W_o is full rank), the unknown parameters $\mathbf{h}_c$ can be accurately estimated by the well-known least-squares estimator:

$$\hat{\mathbf{h}}_c = \left(W_o^T W_o\right)^{-1} {W_o}^T \mathbf{\Gamma}_o \tag{3.18}$$

Moreover, if the information on the signal noise is available, weighted least-squares can be used:

$$\hat{\mathbf{h}}_c = \left(W_o^T R W_o\right)^{-1} W_o^T R \mathbf{\Gamma}_o \tag{3.19}$$

where R is the inverse of the diagonal covariance matrix [21] that contains the standard deviation of the noise on the measured torque.

If only the measured torque is corrupted by noise, constraints can be imposed on the experiment trajectory to ensure the robustness of the identified results. Physically, finding a trajectory that satisfies these constraints is equivalent to finding an optimal trajectory that can excite the identified parameters the most. Several criteria have been proposed in literature [18] such as:

- A-optimality: minimises the trace of $\left(W_o^T W_o\right)^{-1}$.
- D-optimality: maximises the determinant of $\left(W_o^T W_o\right)^{-1}$.
- E-optimality: maximises the minimum singular value of $\left(W_o^T W_o\right)^{-1}$.
- Minimise the condition number of the observation matrix.
- Maximise the likelihood: as discussed in [72].

It is worth pointing out that contaminated data come from not only the measured torque but also the motion data i.e. $(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q})$. The reason is because most industrial robots only come with position sensors at joints, thus, velocity and acceleration are usually derived from numerical differentiation methods. Since numerical differentiation amplifies signal noise [72], preprocessing the data such as filtering technique is commonly used. Note that an improper use of these techniques usually lead to distorted signals, thus, needs to be avoided.

One way to resolve this issue is to limit/control the frequency components inside the trajectory that is used in the experiment [24]. By doing this, the validity of the used data processing techniques can be easily justified.

3.2.2 Trajectory Parameterisation

The exciting/optimal trajectory that is used for the identification experiment should be designed to benefit the data processing techniques later on. To this end, it is natural to limit the frequency components inside the exciting trajectory as discussed in [72]. This approach results in the so-called periodic exciting trajectory which can be parameterised as a sum of a finite Fourier series. The periodic optimal trajectory can be described as follows:

$$q_i(t) = q_{i0} + \sum_{l=1}^{n} a_{il}\sin(w_f lt) - b_{il}\cos(w_f lt) \tag{3.20}$$

$$\dot{q}_i(t) = \sum_{l=1}^{n} a_{il}w_f l\cos(w_f lt) + b_{il}w_f l\sin(w_f lt) \tag{3.21}$$

$$\ddot{q}_i(t) = \sum_{l=1}^{n} -a_{il}(w_f l)^2\sin(w_f lt) + b_{il}(w_f l)^2\cos(w_f lt) \tag{3.22}$$

where w_f and n is the fundamental frequency and the number of frequency components of the excitation trajectories. These parameters should be carefully chosen so as not to excite the un-modelled dynamics of the manipulator.

From the discussion of finding the optimum trajectory, the problem of designing the exciting trajectory becomes one that determines the trajectories that satisfy one of the optimal criteria listed in the previous section. In our case, the problem can be stated as finding the coefficients (q_{i0}, a_{ik}, b_{ik}) such that:

$$CF(q_{i0}, a_{ik}, b_{ik}) = \lambda_1 cond(W_o) + \lambda_2 \frac{1}{\sigma_0(W_o)} \tag{3.23}$$

is minimised, where the scalar λ_1 and λ_2 represent the relative weights between the condition number of the observation matrix: $cond(W_o)$ and its minimum singular value: $\sigma_0(W_o)$. Notice that, because we want to minimise the effect of the non-linear friction on the identified result(s), only the data points which have velocities above a threshold/boundary value (from the previous section) are considered. This differs from other researchers which normally take into account all data points along the optimum trajectory. By taking into account all the above considerations, the problem of finding the optimal trajectory can be stated as a constrained optimisation problem. Note that physical limits of joint position, velocity and acceleration are also needed to be included in this step also as constraints. As is seen from Eqs. 3.16 and 3.23, the cost function is nonlinear and discontinuous (e.g. the $sign$ function in 3.16). Thus, this can make the optimisation process become significantly difficult. In practice, one can avoid the discontinuity by replacing the $sign(\dot{q}_i)$ function in Eq. 3.16 with an approximated continuous function such as $atan(c\dot{q}_i)$. The extra coefficient c is used to adjust the steepness of the slope when $\dot{\mathbf{q}}$ approaches zero. Due to the complexity of the problem, a good initial guess for this optimisation is hard to achieve. We therefore use genetic algorithm (GA) to solve the above optimisation problem [17].

Once the optimisation has been solved, the optimum trajectories for all joints are obtained. The manipulator will be commanded to follow this optimal trajectory by any available controller. For instance, the independent joint control scheme which includes high-gain PID controllers at each joint can be used since we currently do not have the robot dynamic model. The responses of the robot along the trajectories will be recorded. In order to improve the data quality before using them to estimate the dynamic parameters, the data are fitted to the trajectories in Eqs. 3.20, 3.21, and 3.22 [72]. A brief description is as follows:

- Firstly, the joint position data can be filtered by a low-pass filter with an appropriate cut-off frequency which depends on the choice of the fundamental frequency (w_f, n) in Eqs. 3.20-3.22. This is reasonable because the frequency components in the optimal trajectory are predefined to be in the range.

- Secondly, since joint velocity and acceleration are not available due to the lack of joint sensors, this information can be obtained through a numerical differentiation. However, since the exciting trajectory are designed in the form of Eq. 3.20-3.22, a linear least-squares fit can be performed to estimate the coefficients (q_{i0}, a_{ik}, b_{ik}) of the actual optimal trajectory (i.e. the actual motion of the robot) as suggested in [72]:

  $$q_i(t) = \begin{bmatrix} q_i(t=0) \\ q_i(t=t_1) \\ ... \\ q_i(t=T_f) \end{bmatrix} = \begin{bmatrix} 1 & \sin(w_f 0) & -\cos(w_f 0) & ... & ... \\ 1 & \sin(w_f t_1) & -\cos(w_f t_1) & ... & ... \\ ... & ... & ... & ... & ... \\ 1 & \sin(w_f T_f) & -\cos(w_f T_f) & ... & ... \end{bmatrix} \begin{bmatrix} q_{i0} \\ a_{i1} \\ b_{i1} \\ ... \end{bmatrix} \tag{3.24}$$

  As a result, joint velocity and acceleration can be obtained by substituting these coefficients into Equations 3.21 and 3.22.

  Note that the above method should be used only if the control scheme is able to control the manipulator to closely follow the optimal trajectory. The reason is because the above approach only considers frequency components that are in the form of Eqs. 3.20-3.22 of the observation matrix (the left-hand side of Eq. 3.15). However, the measured torques (the right-hand side of Eq. 3.15) are independently filtered, thus, it is possible that the information on two-side of Eq. 3.15 are not consistent. For instance, if the high-gain controller (that is used to control the robot to follow the optimal trajectory) introduces high frequency components that are not in the predefined range (i.e. $\neq kw_f, k = 1:n$) due to the use of high-gain, those frequency components can appear in both side of Eq. 3.15. However, if $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ are fitted into Eq. 3.24, only frequency components that are in the predefined range remain. In other words, the fitted process has made the information on both side of Eq. 3.15 be inconsistent.

3.3 Parameter Estimation

Although the unknown inertial parameters can be estimated by a least-squares technique as shown in Eqs. 3.18 -3.19, there will be a potential problem on the identified results, the so-called physical feasibility of the results [74]. One promising solution for this problem is to use constrained optimisation tools to adjust the least-squares result [76]. By doing this, the physical meaning of the identified parameters can be guaranteed by imposing appropriate constraints on the estimator. The physically feasible characteristic is especially useful for advanced control schemes because it implies that the mass matrix $M(\mathbf{q})$ in Eq. 3.1 is always positive definite. In summary, the following two steps can be used to estimate the robot model from experimental data that can attain (not assure) the physical feasibility of the identified model:

- **Step 1**: the least-squares estimator: in this step, a least-squares based estimator (LS, weighted LS) is first performed to obtain the robot parameters. The validity of these parameters will be checked as described in Section 3.4.2. If the physical feasibility of the result(s) cannot be verified, the second step can be used to modify these result(s) to obtain the final answer.
- **Step 2**: the optimisation-based adjustment: motivated by the idea of virtual parameters [74], a constrained optimisation is used in order to find the unknown inertial parameters instead of the ordinary least-squares. The input vector $\mathbf{X}$ to the optimisation problem is:

$$\mathbf{X} = \begin{bmatrix} \mathbf{h}_{70\times 1} & f_{c1} & f_{v1} & \dots & \dots \end{bmatrix}^T \tag{3.25}$$

where $\mathbf{h}$ is the standard dynamic parameters of links as shown in Eq. 3.4. Constraints on the parameters $\mathbf{h}$ will be imposed in order to make sure that the result will always be physically feasible. The base parameters (vector $\mathbf{X}_c$), can be computed from Eq. 3.25 as:

$$\mathbf{X}_c = \begin{bmatrix} \mathbf{h}_b & f_{c1} & f_{v1} & \dots & \dots \end{bmatrix}^T \tag{3.26}$$

Let us consider a constraint optimisation problem as finding $\mathbf{X}$ such that:

$$CF(\mathbf{X}) = \alpha_1 \left\| W_o \mathbf{X}_c - \mathbf{\Gamma}_o \right\| + \alpha_2 \left\| \mathbf{X}_c - \hat{\mathbf{h}}_c \right\| \tag{3.27}$$

is minimised subjected to constraints on $\mathbf{X}$ to ensure the positive definite property of the inertia matrix $M(\mathbf{q})$ (Eq. 3.1):

$$m_i > 0 \tag{3.28}$$

$$I_i > 0^1 \tag{3.29}$$

where $\hat{\mathbf{h}}_c$ is the least-squares solution from Eq. 3.18 or 3.19, the two scalars α_1, α_2 define how believable the least-squares solution is and (m_i, I_i) are the mass and the 3×3 inertial tensor matrix of link i (Eq. 3.3). Note that the result of the above non-linear constrained optimisation problem will give us a set of physically feasible parameters $\hat{\mathbf{h}}_b$ (because of the constraints on $\mathbf{X}$) which also minimises the error between the measured and predicted torque (because of Eq. 3.27). Note that, theoretically, by only minimising the first term of the cost function (i.e. $\alpha_1 \left\| W_o \mathbf{X}_c - \mathbf{\Gamma}_o \right\|$), we should be able to obtain the physically feasible robot inertia. However, this method may not provide a proper solution in practice because the optimisation problem can fail (the solution does not converge). Thus, we introduce the second term of the cost function (i.e., $\alpha_2 \left\| \mathbf{X}_c - \hat{\mathbf{h}}_c \right\|$ to improve the convergence of the algorithm.

It is worth pointing out that the above optimisation problem can be used not only to find the robot parameters $\mathbf{h_b}$ but also to verify the physical feasibility of $\mathbf{h}_c$ (by assigning $\alpha_1 = 0$ and $\alpha_2 = 1$). Further discussion on the model verification will be given in the next section.

---

[1] Remark that $I > 0$ means that the matrix I is positive definite.

3.4 Model Verification

From the above section, the robot parameters can be obtained from the experimental data. However, due to the signal noise, inadequate sensing capability (for example, joint torque sensor, velocity and acceleration sensors), there is no guarantee for the convergence of the estimated parameters to the actual physical parameters of the robot. In other words, the constrained optimisation in the previous section can only guarantee that the estimated model will converge to a physical model (can be any robot model) that can produce about the same control torque for only the optimal trajectory which is used to excite the robot during the experiment. And for other trajectories, the identified physical model may not produce the correct torques. Thus, it is necessary to independently verify the identified results to check if the identified parameters match the real/actual robot parameters.

An obvious verification method is to compare the reconstructed the torque using the estimated model to the measured joint torques for an arbitrary trajectory. It is worth pointing out that if only the least-squares (LS) technique is used, it is possible for the LS estimator to produce a set of estimated parameters that are physically impossible [17, 74, 76] (for example, the inertia matrix M in Eq. 3.1 is not positive definite for all configuration). Thus, it is also necessary to verify the positive definiteness property of the estimated parameters before using it in any advanced model-based control.

3.4.1 Reconstructed Torque

Reconstructed torque is one the most common method for verifying the quality of the identified parameters. As noted in Section 3.2, the results of the above identification process are the parameters which include the base parameters and joint friction coefficients. Since the base parameters are lumped from the link inertial parameters, it is impossible to directly check the correctness of the identified parameters. Instead, the identified model is verified by comparing the reconstructed torques, which are generated from the identified model, and the measured torques, which are the actual joint torques that are used to control the manipulator. Since the major difference between the approach in this work and others is the use of the boundary velocity, it is necessary to check whether the identified parameters using the boundary velocity have any advantage (please refer back to section 3.1.2 for description of the ”boundary velocity”). Thus, to check the correctness of the estimated parameters, the manipulator is commanded to move along *an arbitrary trajectory*. Motion data such as position, velocity, acceleration $(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q})$ and controlled joint torque $(\mathbf{\Gamma}_c)$ are recorded along the trajectory. By making use of Eq. 3.2, the reconstructed torque can be obtained as follows:

$$\mathbf{\Gamma}_{reconstructed} = W(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q})\hat{\mathbf{h}}_c \tag{3.30}$$

This reconstructed torque is then compared to the measured joint torque, $\mathbf{\Gamma}_c$, that was used to achieve the testing trajectory.

3.4.2 Positive Definiteness of the Mass Matrix

As pointed out in the previous section, the estimated parameters cannot be verified directly because they are lumped parameters. This makes the problem of checking the physical feasibility of the identified parameters significantly harder since the physical meaning of each parameters can only be shown by individually analysing the link parameters i.e. $\mathbf{h}$. Motivated by the virtual parameter concept in [74], a ”brute-force” method is suggested to automate the checking procedure. Let us define the virtual parameters as shown in Eq. 3.25, a constrained optimisation problem can be stated as finding $\mathbf{X}$ such as:

$$CF(\mathbf{X}) = \left\| \mathbf{X}_c - \hat{\mathbf{h}}_c \right\| \tag{3.31}$$

is minimised under the constraints that $\mathbf{X}$ is physically feasible (Eqs. 3.28-3.29). The outcome of this optimisation problem is a set of virtual parameter $\mathbf{X}$ that can produce the same $\hat{\mathbf{h}}_c$ and yet physical feasibility.

In order to see the effectiveness of the above procedure, let us use the Mitsubishi PA10 manipulator as a case-study.

3.5 Case-study: The PA-10 Manipulator

Since the purpose of this chapter is to demonstrate a step-by-step procedure for identifying the dynamic model of industrial robots, the above procedure is applied to the first four links of the PA10 manipulator. The reason that we only identified the dynamic model of the first four links is because the inertia of the last three links of the PA10 is insignificant in comparison to the first four links. The results from the identification process have been verified by comparing the reconstructed torques and the measured torques for an arbitrary joint space trajectory. In addition, the identified model has been tested in a conventional computed-torque controller, the inverse dynamics controller in joint space. A significant improvement in terms of tracking errors was obtained, which also shows the usefulness of the identified model.

3.5.1 Experimental testbed

In order to achieve a critical real-time performance as well as a torque measurement capability, a custom controller has been used instead of the original controller. Seven custom amplifiers are installed together with an 8-axis data acquisition card. A PC which is running the QNX 6.3 operating system is used to control the manipulator as depicted in Figure 3.6.

3.5.2 Model Identification

The following steps were carried out in order to identify the dynamic model:

1. Derive the rigid dynamic model of the robot as shown in Eqs. 3.1 and 3.2. Note that the Coriolis-Centrifugal and gravity term are included in this model.

2. Identify the boundary velocity for the first four joints in which the dynamic friction model becomes linear for each joint (please refer to section 3.1.2 for detailed procedure). Table 3.1 shows the experimental results.

3. Obtain the optimum exciting trajectory as described in Section 3.2. By making

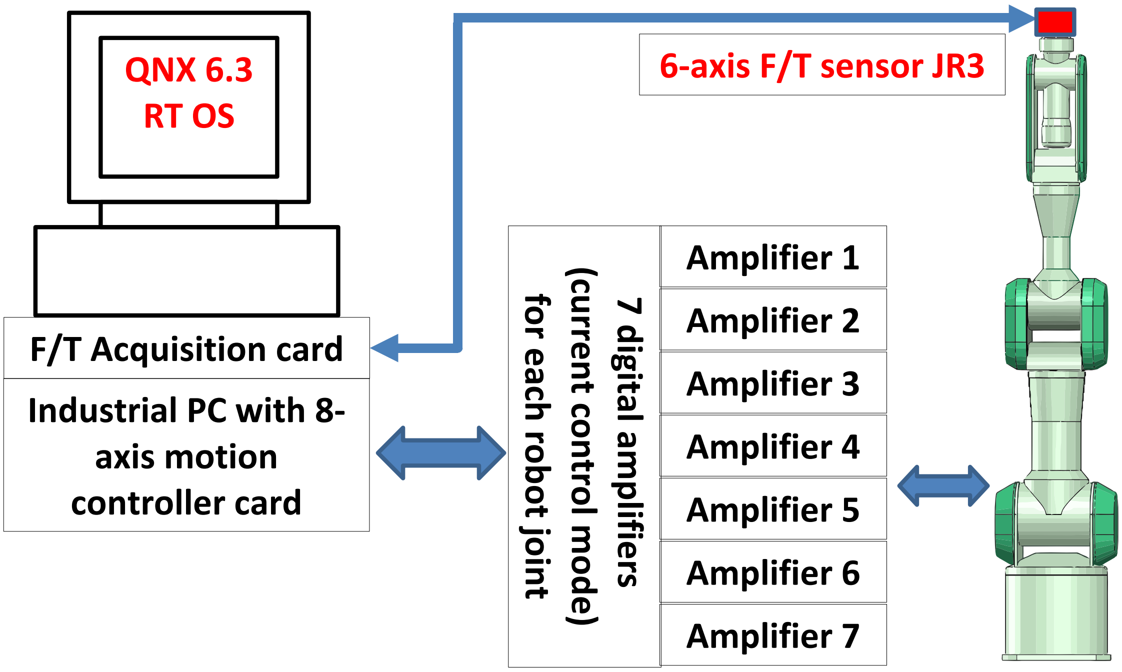

Figure 3.6: PA10 Customised Controller. Firstly, seven amplifiers are directly attached to the robot joints. These amplifiers are configured to operate in current control mode. Since each joint position sensor of the PA10 is a resolver, a custom circuit has been built to generate the reference signal to all the joints. To get joint position information, the response from the resolver at each joint is fed into the encoder emulator of the connected amplifier. Outputs from the encoder emulators are then captured by a Servo-to-go 8-axis ISA servo I/O data acquisition card (*www.servotogo.com/*), which is installed inside an industrial computer (CPU: 3GHz single-core, RAM: 256MB, HDD: 80GB). Finally, the control algorithms are implemented on the industrial PC which uses QNX Neutrino Realtime Operating System 6.3. To standardise all further developments, the implementation of the algorithms in this work adopted the **MRROC++** framework *(www.ia.pw.edu.pl/ zielinsk/)*. Please refer to **Appendix A** for further description.

Table 3.1: Boundary velocities of the first four joint of the Mitsubishi PA10 manipulator.

$joint$	$\dot{q}_{iThres}(rad/s)$
1	0.25
2	0.27
3	0.3
4	0.6

use of the Matlab Genetic Algorithm Toolbox, the optimum trajectory was found with the minimum condition number around 65.

4. Execute the optimum trajectory on the PA10; obtain the joint motion and joint torque data. Note that because the PA10 manipulator does not have joint torque sensor, the joint torques are obtained by measuring the motor currents as discussed in [80]. An independent joint control scheme is used at each joint to make the joints follow the reference/optimal trajectory.

5. The inertial parameters are estimated using the method described in Section 3.4. The equivalent virtual parameters (Eq. 3.25) are shown in Table 3.3. Note that these parameters were obtained with the constraint to ensure the physical feasibility of the identified results (i.e., positive definiteness of the inertia matrix $M(\mathbf{q})$ in Eq. 3.1).

3.5.3 Model Verification

As noted in Section 3.4.1, the major difference between the proposed approach and others is the use of the boundary velocity. It is necessary to check whether the identified parameters using the boundary velocity have any advantage. To this end, two sets of data have been used to identify $\hat{\mathbf{h}}_c$. The first set (set A) includes all the experimental points while the second set (set B) only includes the data points which have $|\dot{q}| > \dot{q}_{threshold}$. In the case of the PA10 manipulator, the number of data points in set B is about 30% of the number of data points in set A. Denote the identified model using set A as $\hat{\mathbf{h}}_{cA}$ and the identified model using set B as $\hat{\mathbf{h}}_{cB}$. Figures 3.7-3.10 shows the measured torques vs the re-constructed torques of joints 1-4 for an arbitrarily chosen different trajectory in joint space. In these Figures, the ”re-constructed torque” is computed as Eq. 3.30 using $\hat{\mathbf{h}}_{cA}$ and $\hat{\mathbf{h}}_{cB}$. Note that red represents the measured torque, blue represents the re-constructed torque using $\hat{\mathbf{h}}_{cA}$ (all velocities) and green represents the re-constructed torque using $\hat{\mathbf{h}}_{cB}$ (boundary velocities). The root-mean-square (RMS) errors between the measured torque and re-constructed torque are shown in Table 3.2.

Table 3.2: RMS errors between the measured torque and re-constructed torque.

$joint$	$\hat{\mathbf{h}}_{cA}$	$\hat{\mathbf{h}}_{cB}$
1	**3.9**	4.1
2	7.1	**5.9**
3	2.1	**1.5**
4	5.8	**3.7**

Table 3.3: Virtual parameters that can reproduce $\hat{\mathbf{h}}_{cB}(virtual)$ with $||\hat{\mathbf{h}}_{cB}(virtual) - \hat{\mathbf{h}}_{cB}|| \approx 10^{-5}$.

$link$	1	2	3	4
xx_i	0.3811	0.0347	0.0236	0.0758
yy_i	0.6884	1.8651	0.0119	0.0073
zz_i	0.0356	3.006	0.1705	0.588
xy_i	0.9833	0.7599	0.5881	0.8631
xz_i	1.1198	0.8079	0.5855	0.1214
yz_i	0.317	0.0001	1.5338	0.019
mx_i	0.8353	2.3296	1.3338	0.8852
my_i	0.8831	0.0827	0.4072	0.1501
mz_i	3.1872	0.4506	0.2006	1.6751
m	0.4696	0.2088	2.5769	3.7324

Theoretically, one should expect the quality of the identified parameter using set B to be worse than the one using set A, because there are more data in set A. However, as can be seen in Table 3.2, an almost opposite result was obtained. The RMS errors in set A are bigger than set B for most of the joints. This observation implies that the extra data points in set A contribute *negatively* to the accuracy of the identified result $\hat{\mathbf{h}}_{cA}$, especially in low velocity regions. Moreover, the physical feasibility of the identified parameters has also been verified using the virtual parameter method as discussed in the previous section. The equivalent virtual parameters of the identified model $\hat{\mathbf{h}}_{cB}$ is presented in Table 3.3. It is worth stressing that the virtual parameters from the optimisation process do not need to be the actual parameters of the manipulator (for example: $z_{c1} = 3.1872$ cannot be the parameter of the PA10). However, to assure the physical feasibility of the identified result, the only necessary condition is that there exists a physical model (the virtual parameters) that can reproduce the same $\hat{\mathbf{h}}_{cB}$.

Since the purpose of the identification process in this chapter is to obtain a model that can be used in advanced model-based control schemes, the identified model has

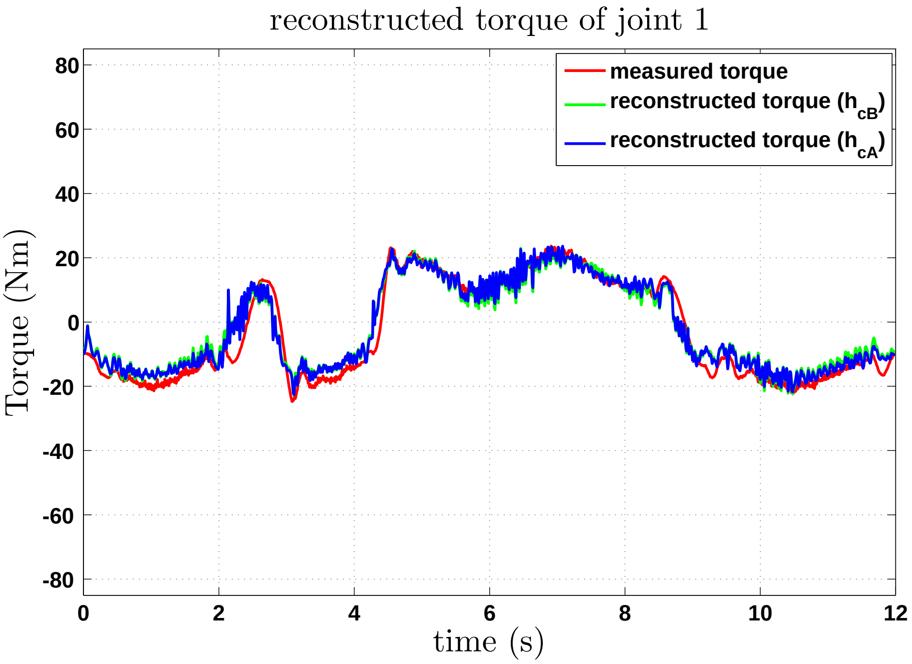

Figure 3.7: Measured torque vs reconstructed torque for joint 1.

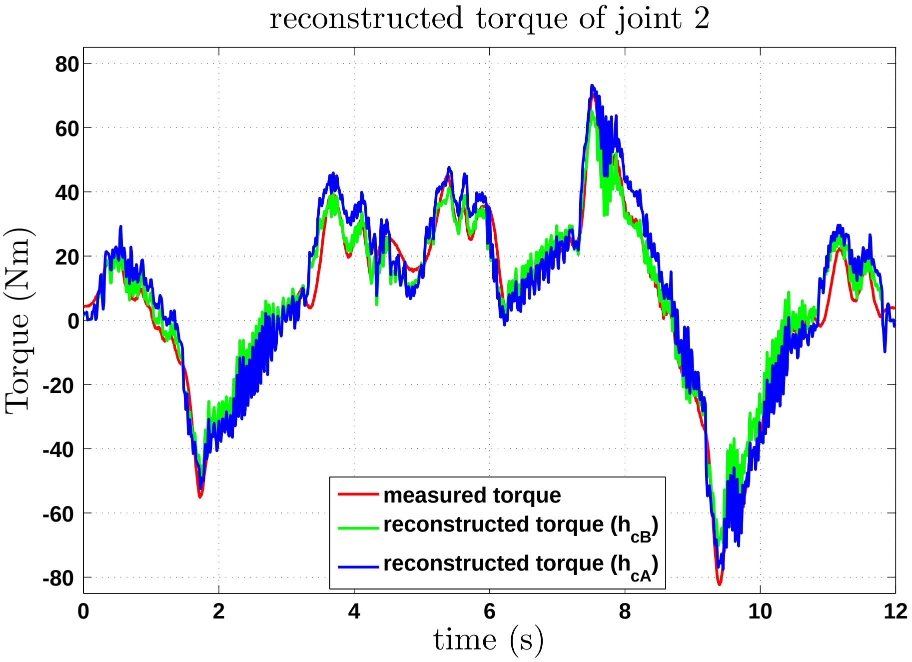

Figure 3.8: Measured torque vs reconstructed torque for joint 2.

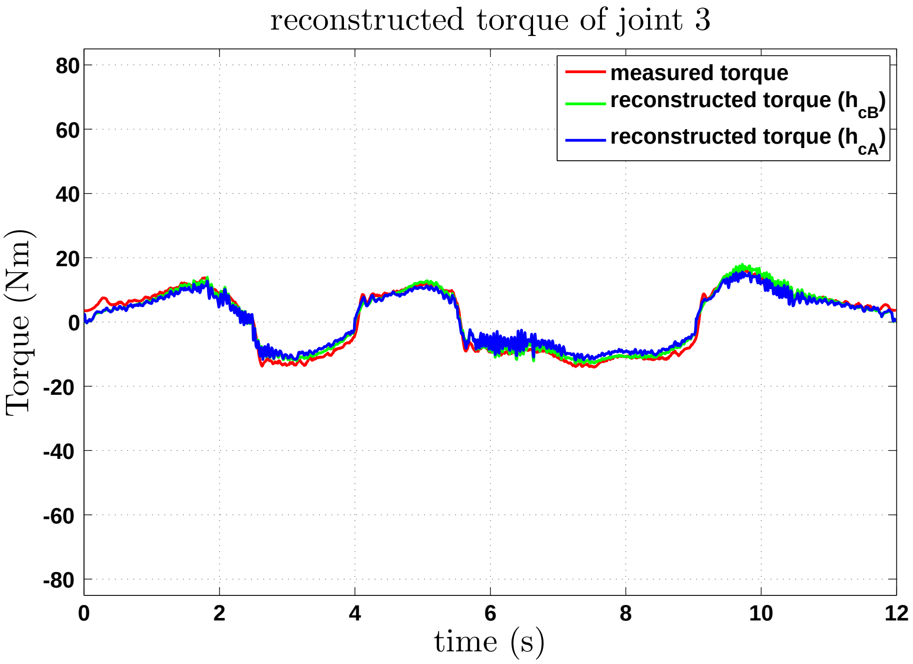

Figure 3.9: Measured torque vs reconstructed torque for joint 3.

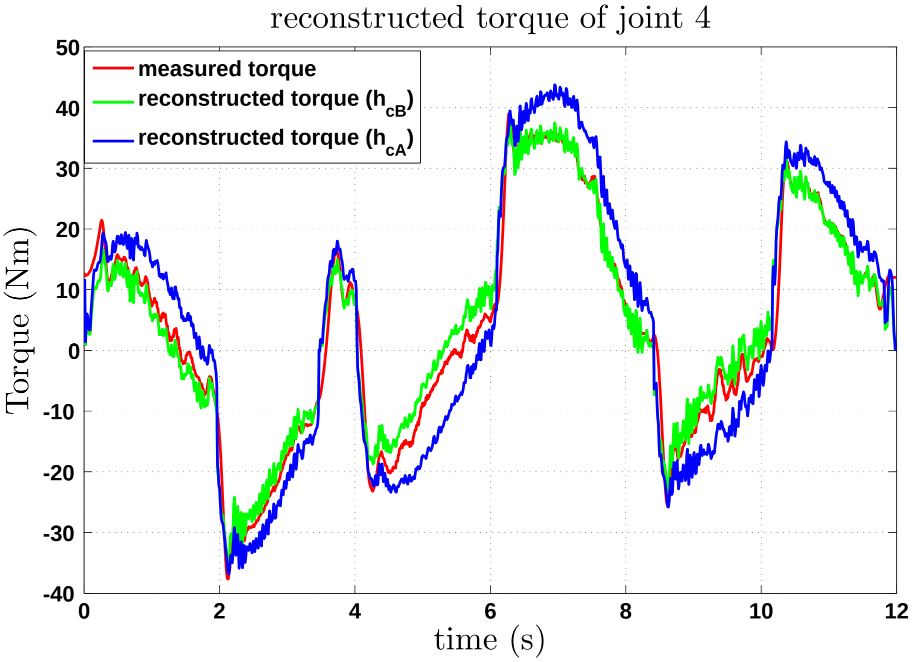

Figure 3.10: Measured torque vs reconstructed torque for joint 4.

been further tested in another experiment as described below:

1. The first four joints of the manipulator is commanded to follow a sinusoidal trajectory (amplitude: 30 degree, period = 4s).
2. Two controller schemes were implemented:
   - **Independent Joint Control**: no dynamic information was used. This control scheme is chosen because of its widely use in most industrial manipulators (simple to implement, cost-effective and modularisation).
   - **Inverse Dynamic Control**: the identified dynamic parameters were used to to decouple the dynamic behaviour among the axes. A standard joint space computed control was implemented.

     $$\mathbf{u} = \ddot{\mathbf{q}}_d + K_V(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + K_P(\mathbf{q}_d - \mathbf{q}) \tag{3.32}$$

     $$\mathbf{\Gamma} = \hat{M}\mathbf{u} + \hat{\mathbf{C}} + \hat{\mathbf{G}} + \hat{\mathbf{\Gamma}}_{fric} \tag{3.33}$$

     Notice that the feed-forward frictions i.e. the compensated frictions are computed based on the desired joint velocities.

The tracking errors for the first four joints are shown in Figure 3.11-3.14: blue represents the joint tracking errors using the independent joint control scheme; red represents the joint tracking errors using the dynamic control.

It is clear that there is a significant improvement in terms of the tracking error for joint 1, 2 and 4. The tracking error for joint 3 is not as much different as others. One explanation that is because of the structure of the PA10 that makes the inertial effects at joint 1, 2, 4 much easier to be excited than the rest of the joints. As a result, the quality of the identified parameters which contribute to the joint torque of joint 3 are poorer. This observation implies that further constraints need to be imposed on the optimum trajectories in order to excite the dynamic effects from different joints evenly.

It is worth pointing out that the above identified dynamic model was obtained in the high speed region. Consequently, it is necessary to see how good the identified model in

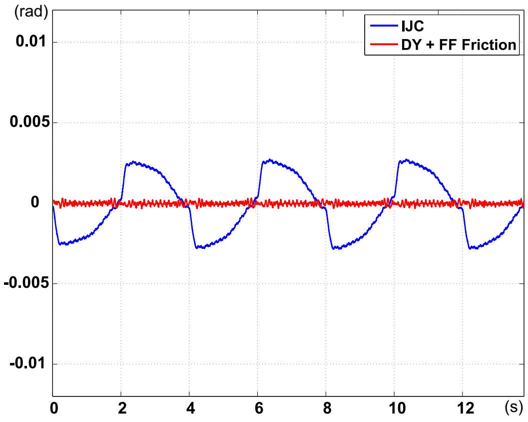

Figure 3.11: Tracking error of joint 1 (4s).

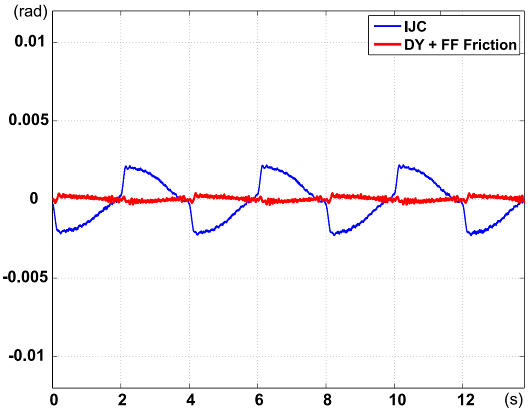

Figure 3.12: Tracking error of joint 2 (4s).

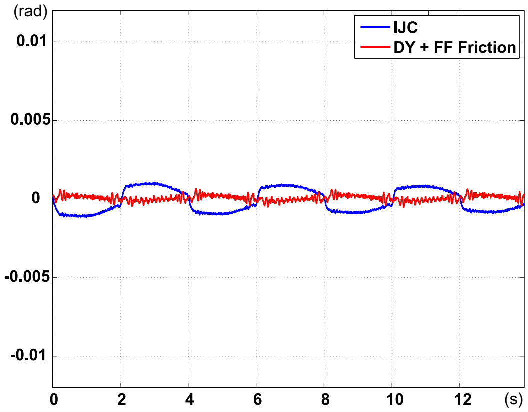

Figure 3.13: Tracking error of joint 3 (4s).

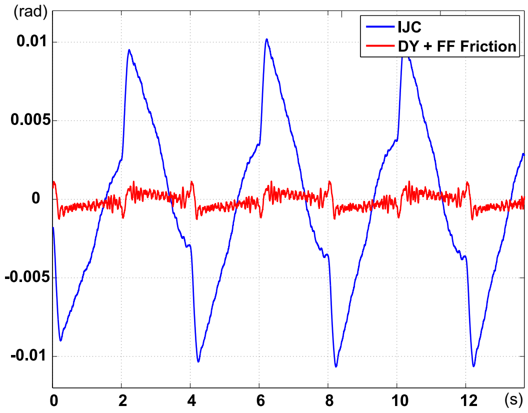

Figure 3.14: Tracking error of joint 4 (4s).

the low speed region is. In order to check the performance of the identified parameters in the low speed region, the above experiment has been redone with the period of the desired trajectory increased from $4s$ to $40s$. Tracking errors are shown in Figure 3.15-3.18 where blue represents independent joint control scheme; red represent dynamic control.

As is seen, the differences between the model-based control and non-model based control are no longer significant as shown in the previous case. One possible explanation is that the inertial effects of the dynamic model have been dominated by joint frictions at low speed region. As a result, the control performance will mainly depend on how joint frictions are compensated in this region. Since only a simple friction model is used, where Coulomb and viscous friction are considered, poor performance should be expected.

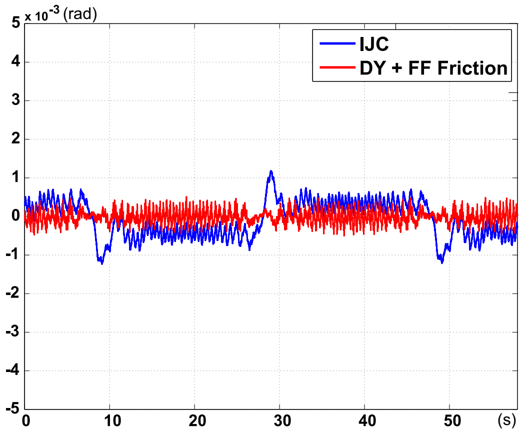

Figure 3.15: Tracking error of joint 1 (40s).

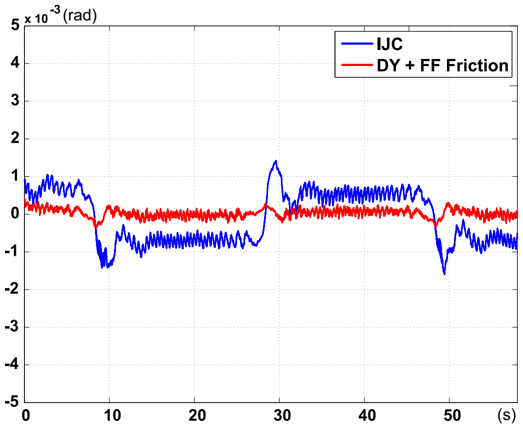

Figure 3.16: Tracking error of joint 2 (40s).

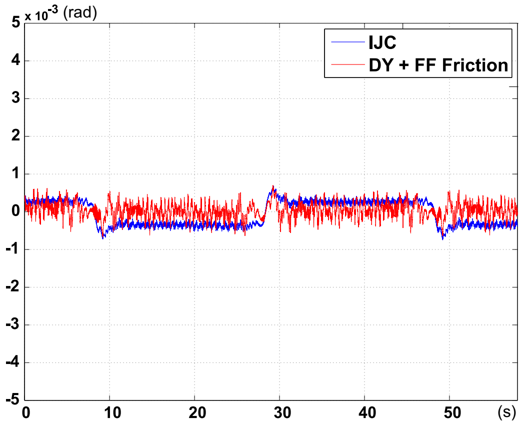

Figure 3.17: Tracking error of joint 3 (40s).

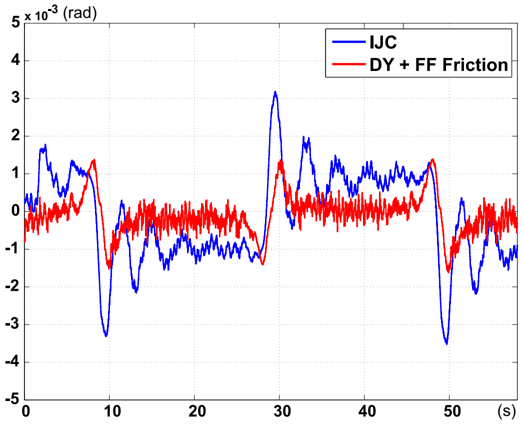

Figure 3.18: Tracking error of joint 4 (40s).

3.5.4 Summary

We presented an improved procedure for identifying the dynamic model of robot manipulators by taking into account the effects of dynamic friction. As the linearity property of the robot dynamic model is valid only in the static region of the dynamic friction, a scheme was proposed to identify boundary velocity that separates the dynamic friction into nonlinear and linear regions. The above boundary velocities have been used as a constraint in both experimental design and parameter estimation. In addition, the physical feasibility of the identified parameters was also considered as an important criterion. A constrained optimisation problem has been introduced to adjust the identified result(s). Experiments are completed for the first four joints of the Mitsubishi PA10 manipulator to validate the effectiveness of the proposed identification procedures.

Chapter 4

Model Uncertainties and Their Effects on Discrete Controllers

As discussed in Chapter 1, compliant motion control capability is an essential requirement for interaction tasks. Although this capability has been realised by many researchers using laboratory robots, there are limited applications of these advanced control schemes on industrial manipulators. One possible explanation among others is that advanced control schemes require specialised knowledge about the robot such as the robot inertia and joint friction which are not always available in practice. In the previous chapter, we have discussed the problem of model identification for industrial robots subjected to advanced model-based control. From the discussion, it should be clear that a perfect model is impossible to achieve due to modelling error and the lack of sensing capability. Thus, it is necessary to investigate the effects of model uncertainties when the model is used in model-based control. Since this research focuses on the compliant motion capability of industrial robots using the force-based operational space control as discussed in Chapter 1, this chapter will be devoted to study the effects of model uncertainties on the control performance from both theoretical and practical points of view.

It is worth stressing that although the force-based operational space framework works perfectly in the simulation environment, it is known to be sensitive to the accuracy of the robot model as pointed out in [40]. This conclusion is further reinforced when we implemented the force-based OSC using the identified model as discussed in the previous chapter (this implementation is referred to as the conventional implementation of the force-based OSC). From the experiment, it seems that the task space control gains are not only hard to tune but also trajectory dependent. In fact, it is impossible to choose the same control gains in task space (Eq. 2.13-2.14) to have a reasonable performance because of the model uncertainties. For instance, consider the task as depicted in Figure 4.1 which is to move the PA10's robot end-effector $20cm$ along y_0 direction in $2s$. Quintic polynomials are used for trajectory planning as described in Chapter 2. Null space control objective is to maintain the position of the first joint i.e. $q_{1d} = q_{1i} = 0$. To realise this task, the conventional force-based OSC as presented in Chapter 2 has been implemented. The identified dynamic model was obtained as discussed in Chapter 3. The control gains are chosen to have critical damped behaviour in all task space directions as discussed in [13, 34]. These control gains were tuned to give the best possible performance as mentioned in [40]. The task space tracking errors in the (x, y, z) direction are plotted in Figures 4.2-4.4. From the figures, it is clear that the tracking performance for the three directions is large and inconsistent in terms of the magnitude (x_{axis} got the smallest tracking error and z_{axis} got the largest tracking error). Further investigation on the robot model indicates that joint friction in the low-speed region is complex as mentioned in the previous chapter. Unfortunately, most of the interaction tasks which are frequently used in industry such as grinding, deburring and polishing often results in slow motion at joints.

In order to see the effects of joint friction on the conventional OSC implementation, let us look at how torque control is implemented at each joint. For almost all industrial manipulators, the desired joint torque is achieved through controlling the motor current because of the lack of joint torque sensor. Figure 4.5 shows a typical torque control loop, $\tau_{desired}$ is the desired joint torque, $i_{desired}$ is the equivalent desired motor current, i is the motor current, τ_M is the actual torque provided by the motor, τ_L is the joint torque at the load side, n is the gear ratio, τ_{fric} is the friction torque and τ_{dist} is the unknown disturbance. Note that only motor current is controlled and it is assumed that torque

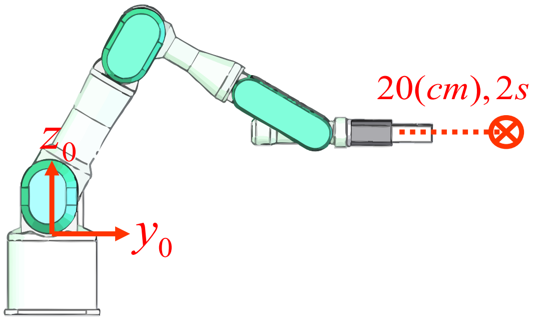

Figure 4.1: Free-motion task: the end-effector is commanded to move 20cm in y_0 direction in 2s.

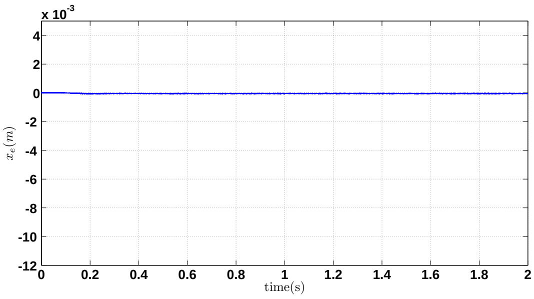

Figure 4.2: Task space tracking error in x direction.

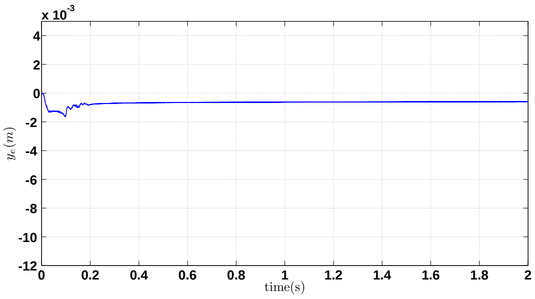

Figure 4.3: Task space tracking error in y direction.

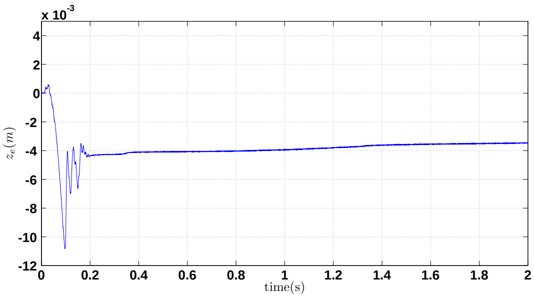

Figure 4.4: Task space tracking error in z direction.

control is achieved through a known motor torque constant. Frictions in the gear box and load side are not considered and need to be compensated by an outer loop. For the OSC framework, the outer feedback loop is closed via the task space variables (position and orientation) and contact force. In joint space control, the effects of the joint friction are collocated in the individual joint variables and thus easy to compensate by any joint space controller. However, this is no longer true in task space control because the OSC has to indirectly compensate these effects through the kinematic chain. In other words, the uncertainty dynamics at one joint will be compensated by adjusting *all task space control gains* since these model uncertainties at one joint are propagated into task space through the robot kinematics. As a result, improper friction compensation in joint space can degrade the task space tracking in motion control and cause the applied force to enter a limit cycle [48]. Experimental results on the PA10 manipulator have also confirmed this observation.

Although friction modelling and compensation are well documented in the literature,

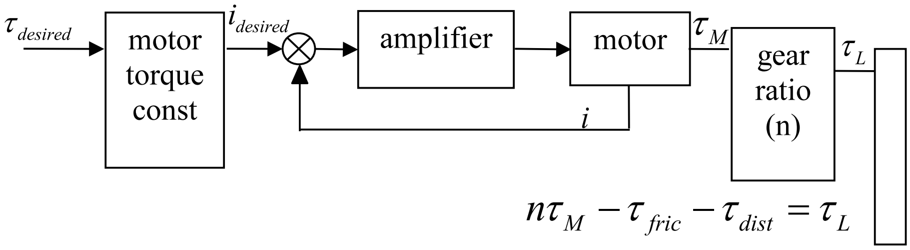

Figure 4.5: One link without torque sensor.

a perfect friction model which can capture all the non-linear friction effects, especially frictions at the low-velocity region is still an open problem. In this region, friction is known to be highly nonlinear, with the hysteresis effect, Stribeck effect, position and/or time dependency and other nonlinearities. Note that the implementation of these friction compensator schemes often requires reference joint velocities and/or good joint velocity estimation. From the practical point of view, these requirements are not easily fulfilled for the case where the motion and force are controlled in task space since there are *no explicit reference joint velocities* in this case. Moreover, the feedback joint velocities are normally obtained through numerical differentiation of the joint position measurements since most industrial robots do not have joint velocity sensors. As a result, errors in the velocity estimation in low velocity region will be magnified in the friction compensation due to velocity dependence of Stribeck and Coulomb friction models.

From the above discussion, model uncertainty is an inherent problem. Although there are many solutions have been proposed for this problem such as the robust and adaptive control schemes, they do not consider the space where the controller takes place i.e. joint space or task space. Thus, it is necessary to investigate the effects of the uncertainties on joint space and task space controller.

4.1 Effects of Model Uncertainties on JS and TS Control - Analytical Approach

For convenience, the dynamic model of a n-DOF rigid body in joint space (Eq. 2.1) and task space (2.2) in free space are re-written as follows:

$$M\ddot{\mathbf{q}} + \mathbf{N}(\dot{\mathbf{q}}, \mathbf{q}) = \mathbf{\Gamma} \tag{4.1}$$

$$\Lambda\ddot{\mathbf{x}} + \mathbf{H}(\dot{\mathbf{x}}, \mathbf{x}) = \mathbf{F} \tag{4.2}$$

where:

$$\mathbf{N}(\dot{\mathbf{q}}, \mathbf{q}) = \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{G}(\mathbf{q}) + \mathbf{D}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{\Gamma}_{fric} \tag{4.3}$$

and

$$\mathbf{H}(\dot{\mathbf{x}}, \mathbf{x}) = \boldsymbol{\mu}(\dot{\mathbf{x}}, \mathbf{x}) + \boldsymbol{\rho}(\mathbf{x}) + \boldsymbol{\varphi}(\dot{\mathbf{x}}, \mathbf{x}) + \mathbf{F}_{fric} \tag{4.4}$$

Let the identified or estimated dynamic model which is used in joint space control scheme as $(\hat{M}, \hat{\mathbf{N}})$. The equivalent task space dynamics at the end-effector are:

$$\hat{\Lambda} = (J\hat{M}^{-1}J^T)^{-1} \tag{4.5}$$

$$\hat{\mathbf{H}} = \hat{\boldsymbol{\mu}}(\dot{\mathbf{x}}, \mathbf{x}) + \hat{\boldsymbol{\rho}}(\mathbf{x}) + \hat{\mathbf{F}}_{fric} \tag{4.6}$$

where $J(\mathbf{q})$ is the Jacobian matrix of the end-effector expressed in the base frame. Note that we have assumed here that the robot kinematic model is precisely known. It is reasonable in practice because the kinematic model is usually provided by the robot manufacturer. Considering uncertainties in **both** kinematic and dynamic model is out of the scope of this work.

4.1.1 Effects of Model Uncertainties on JS and TS Control - Continuous Case

Let us apply the well-known computed torque control scheme the inverse dynamics control technique in joint space and task space to Eqs. 4.1 and 4.2:

- **Joint Space**: Assume that the robot is commanded to follow a desired trajectory in joint space $\{\ddot{\mathbf{q}}_d, \dot{\mathbf{q}}_d, \mathbf{q}_d\}$. The control torque can be computed as:

  $$\mathbf{\Gamma} = \hat{M}\mathbf{u}_{JS} + \hat{\mathbf{N}} \tag{4.7}$$

  where:

  $$\mathbf{u}_{JS} = \ddot{\mathbf{q}}_d + K_{Dq}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + K_{Pq}(\mathbf{q}_d - \mathbf{q}) \tag{4.8}$$

  By applying this control torque to Eq. 4.1, the closed-loop equation becomes:

  $$\ddot{\tilde{\mathbf{q}}} + M^{-1}\hat{M}K_{Dq}\dot{\tilde{\mathbf{q}}} + M^{-1}\hat{M}K_{Pq}\tilde{\mathbf{q}} = M^{-1}\tilde{M}\ddot{\mathbf{q}}_d + M^{-1}\tilde{\mathbf{N}} \tag{4.9}$$

  where:

  $$\ddot{\tilde{\mathbf{q}}} = \ddot{\mathbf{q}}_d - \ddot{\tilde{\mathbf{q}}} \tag{4.10}$$

  $$\dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}}_d - \dot{\tilde{\mathbf{q}}} \tag{4.11}$$

  $$\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q} \tag{4.12}$$

  $$\tilde{M} = M - \hat{M} \tag{4.13}$$

  $$\tilde{\mathbf{N}} = \mathbf{N} - \hat{\mathbf{N}} \tag{4.14}$$

- **Task Space**: Assume that the robot end-effector is commanded to follow a desired trajectory in task space: $\{\ddot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \mathbf{x}_d\}$. The control torque can be computed as:

  $$\mathbf{F} = \hat{\Lambda}\mathbf{u}_{TS} + \hat{\mathbf{H}} \tag{4.15}$$

  $$\mathbf{\Gamma} = J^T\mathbf{F} \tag{4.16}$$

where:

$$\mathbf{u}_{TS} = \ddot{\mathbf{x}}_d + K_{Dx}(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + K_{Px}(\mathbf{x}_d - \mathbf{x}) \tag{4.17}$$

By applying this control torque to Eq. 4.2, the closed-loop equation becomes:

$$\ddot{\tilde{\mathbf{x}}} + \Lambda^{-1}\hat{\Lambda}K_{Dx}\dot{\tilde{\mathbf{x}}} + \Lambda^{-1}\hat{\Lambda}K_{Px}\tilde{\mathbf{x}} = \Lambda^{-1}\tilde{\Lambda}\ddot{\mathbf{x}}_d + \Lambda^{-1}\tilde{\mathbf{H}} \tag{4.18}$$

where:

$$\ddot{\tilde{\mathbf{x}}} = \ddot{\mathbf{x}}_d - \ddot{\tilde{\mathbf{x}}} \tag{4.19}$$

$$\dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}}_d - \dot{\tilde{\mathbf{x}}} \tag{4.20}$$

$$\tilde{\mathbf{x}} = \mathbf{x}_d - \mathbf{x} \tag{4.21}$$

$$\tilde{\Lambda} = \Lambda - \hat{\Lambda} \tag{4.22}$$

$$\tilde{\mathbf{H}} = \mathbf{H} - \hat{\mathbf{H}} \tag{4.23}$$

If $\hat{M} = M, \hat{\mathbf{N}} = \mathbf{N}$, thus $\hat{\Lambda} = \Lambda, \hat{\mathbf{H}} = \mathbf{H}$ (perfect model estimation), the joint space and task space closed-loop equations (Eq. 4.9 and Eq. 4.18) reduce to n second-order system as follows:

$$\ddot{\tilde{\mathbf{q}}} + \ K_{Dq}\dot{\tilde{\mathbf{q}}} + K_{Pq}\tilde{\mathbf{q}} = \mathbf{0} \tag{4.24}$$

$$\ddot{\tilde{\mathbf{x}}} + \ K_{Dx}\dot{\tilde{\mathbf{x}}} + K_{Px}\tilde{\mathbf{x}} = \mathbf{0} \tag{4.25}$$

As a result, one can choose the control gains $\{K_{Dq}, K_{Pq}\}$ to achieve exponential stability.

However, if $\hat{M} \neq M, \hat{N} \neq N$, the equilibrium $\tilde{\mathbf{q}} = \mathbf{0}$ of Eq. 4.24 is no longer exponentially stable. Instead, a weaker stability, the uniformly boundedness stability, can be shown as in [57]. Note that the size of the bound depends on the system parameter and control gains. Moreover, this size can be made arbitrarily small by increasing the control gains to infinity. In fact, [57] showed that the high-gain controller can make the equilibrium of the above closed-loop system asymptotical stable for any non-zero initial value of $\tilde{\mathbf{q}}(0)$. A similar analysis can also be performed for the task space controller.

Since the control gains can be assigned arbitrarily high, the performance of the joint space and task space for any same/equivalent trajectory can always be equivalent by adjusting the control gains $\{K_{Dq}, K_{Pq}\}$ and $\{K_{Dx}, K_{Px}\}$.

The above observation implies that the uncertainties in the dynamic model can be easily overcome by playing with the control gains. In fact, if there is no upper limit on the control gains, both the joint space and task space inverse dynamics control schemes (Eq. 4.7-4.15) can give a similar performance for the same inaccurate dynamic model $(\hat{M}, \hat{\mathbf{N}})$. For instance, consider the 3-DOF (RRR) robot as presented in Chapter 2 (Figure 4.6). Let us simplify the analysis by making the length of the three links the same $(l_1 = l_2 = l_3 = 2m)$. The initial configuration is chosen as shown in Figure 4.7. Under the assumption that only the estimated mass of each link is inaccurate i.e. $(\hat{m}_1 = 10kg, \hat{m}_2 = 5kg, \hat{m}_3 = 2.5kg)$, the above two controllers Eq. 4.7 and Eq. 4.15 are implemented to control joint 1 from the initial configuration $(q_1 = \pi/3)$ to the desired configuration $(q_1 = \pi/2)$ while maintaining the end-effector position at $(0, 0)$. Quintic polynomials are used for trajectory planning as described in Chapter 2. Since the three link lengths are the same, the joint responses for the above task using the task space and joint space controllers will be the same. To see the effect of the control gains on the control performance, let us choose the control gains (as the discussion in [34,57]) i.e.,:

$$K_{P_{q/x}} = w^2 I \tag{4.26}$$

$$K_{D_{q/x}} = 2wI \tag{4.27}$$

Figure 4.8 and 4.10 show the task space tracking errors for some w using the joint space controller law (Eq. 4.7). Figure 4.9 and 4.11 show the task space tracking errors for some w using the operational space controller (Eq. 4.15). As is seen, the tracking errors are inversely proportional to w for both joint space and task space controller. As a result, the effect of model inaccuracy can be easily overcome by increasing the control gains (as long as the system stability is guaranteed). It is worth noting that the above controllers are implemented in the continuous domain.

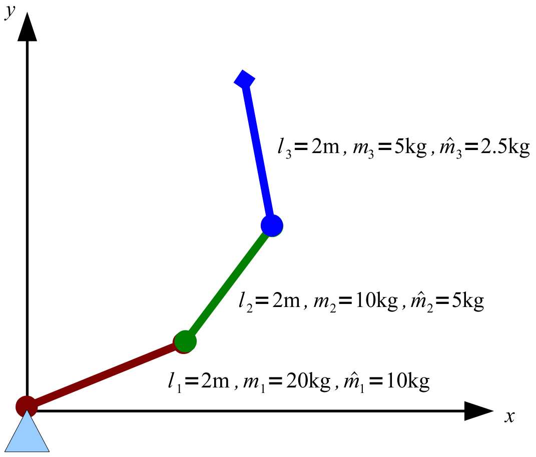

Figure 4.6: 3-DOF RRR robot.

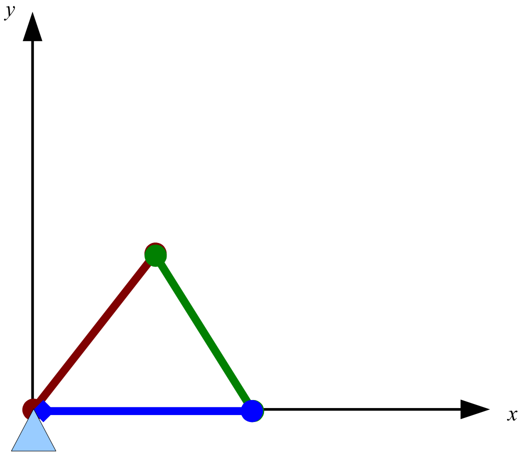

Figure 4.7: 3-DOF initial configuration.

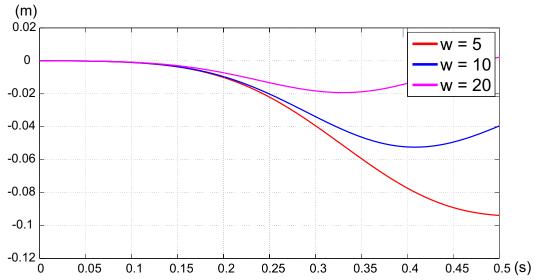

Figure 4.8: Tracking error in x direction using the **joint space** controller.

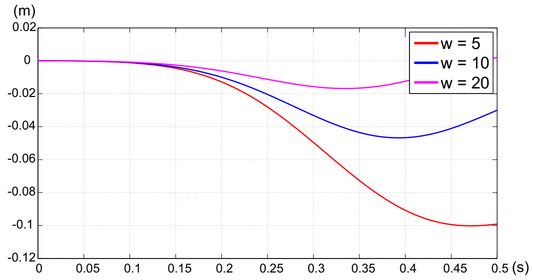

Figure 4.9: Tracking error in x direction using the **task space** controller.

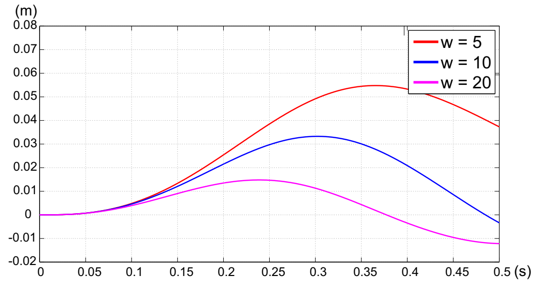

Figure 4.10: Tracking error in y direction using the **joint space** controller.

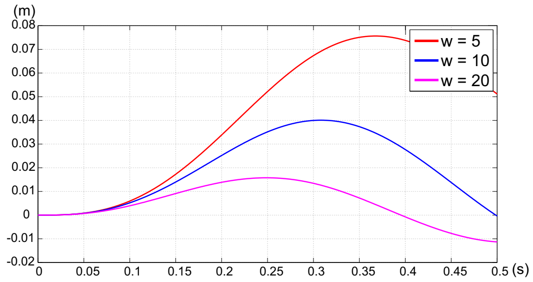

Figure 4.11: Tracking error in y direction using the **task space** controller.

In practice where the above control law are mostly digitally implemented, the control gains cannot be increased above certain threshold values. If the control gains are above these thresholds, the closed-system can inherit the so-called chattering effects [81]. In the next section, an upper limit of the control gains for digital controller will be derived. Also note that there are more factors that contribute to the chattering effect such as noise in the measured signals. However, in this section, only the effect of digital controller will be addressed.

4.1.2 Effects of Model Uncertainties on JS and TS Control - Discrete Case

To see the effects of the digital controller on the overall system performance, let us consider a set-point controller in joint space and task space as follows:

$$\mathbf{\Gamma} = \hat{M}\mathbf{u}_q + \hat{\mathbf{N}} = \hat{M}(-K_{Dq}\dot{\mathbf{q}} + K_{Pq}(\mathbf{q}_d - \mathbf{q})) + \hat{\mathbf{N}} \tag{4.28}$$

$$\mathbf{F} = \hat{\Lambda}\mathbf{u}_x + \hat{\mathbf{H}} = \hat{\Lambda}(-K_{Dx}\dot{\mathbf{x}} + K_{Px}(\mathbf{x}_d - \mathbf{x})) + \hat{\mathbf{H}} \tag{4.29}$$

where:

- $\mathbf{q}_d$ and $\mathbf{x}_d$ are the desired position in joint space and task space.

- K_{Pq}, K_{Dq}, K_{Px} and K_{Dx} are control gains:

$$K_{Dq} = k_{Dq}I_{n\times n}, K_{Pq} = k_{Pq}I_{n\times n} \tag{4.30}$$

$$K_{Dx} = k_{Dx}I_{6\times 6}, K_{Px} = k_{Px}I_{6\times 6} \tag{4.31}$$

Note that a smooth quintic-polynomial input trajectory as presented in the previous section has been replaced by a step input to simplify the analysis. In addition, comparing the transient responses from step input is more intuitive than comparing the tracking errors along an input trajectory. The closed-loop systems using the control Eq. 4.28 and Eq. 4.29 now becomes:

$$\ddot{\mathbf{q}} = M^{-1}\hat{M}(-K_{Dq}\dot{\mathbf{q}} + K_{Pq}(\mathbf{q}_d - \mathbf{q})) - M^{-1}\tilde{\mathbf{N}} \tag{4.32}$$

$$\ddot{\mathbf{x}} = \Lambda^{-1}\hat{\Lambda}(-K_{Dx}\dot{\mathbf{x}} + K_{Px}(\mathbf{x}_d - \mathbf{x})) - \Lambda^{-1}\tilde{\mathbf{H}} \tag{4.33}$$

Denote:

$$\varepsilon_q = \frac{1}{k_{Dq}}, \alpha_q = \frac{k_{Pq}}{k_{Dq}} \tag{4.34}$$

$$\varepsilon_x = \frac{1}{k_{Dx}}, \alpha_x = \frac{k_{Px}}{k_{Dx}} \tag{4.35}$$

$$\mathbf{s}_q = \dot{\mathbf{q}} - \alpha_q(\mathbf{q}_d - \mathbf{q}) \tag{4.36}$$

$$\mathbf{s}_x = \dot{\mathbf{x}} - \alpha_x(\mathbf{x}_d - \mathbf{x}) \tag{4.37}$$

The above closed-loop equations become:

$$\ddot{\mathbf{q}} = \frac{-1}{\varepsilon_q} M^{-1}\hat{M}\mathbf{s}_q - M^{-1}\tilde{\mathbf{N}} \tag{4.38}$$

$$\ddot{\mathbf{x}} = \frac{-1}{\varepsilon_x} \Lambda^{-1}\hat{\Lambda}\mathbf{s}_x - \Lambda^{-1}\tilde{\mathbf{H}} \tag{4.39}$$

By introducing the fast time scale:

$$\tau = \frac{t}{\varepsilon} \tag{4.40}$$

and note that:

$$\dot{\mathbf{s}}_q = \ddot{\mathbf{q}} + \alpha_q\dot{\mathbf{q}} \tag{4.41}$$

$$\dot{\mathbf{s}}_x = \ddot{\mathbf{x}} + \alpha_x\dot{\mathbf{x}} \tag{4.42}$$

the above closed-loop equations become:

$$\begin{cases} \frac{d\mathbf{q}}{dt} = \dot{\mathbf{q}} \\ \frac{d\mathbf{s}_q}{d\tau} = -M^{-1}\hat{M}\mathbf{s}_q + \varepsilon_q(\alpha_q\dot{\mathbf{q}} - M^{-1}\tilde{\mathbf{N}}) \end{cases} \tag{4.43}$$

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}} \\ \frac{d\mathbf{s}_x}{d\tau} = -\Lambda^{-1}\hat{\Lambda}\mathbf{s}_x + \varepsilon_x(\alpha_x\dot{\mathbf{x}} - \Lambda^{-1}\tilde{\mathbf{H}}) \end{cases} \tag{4.44}$$

From the above equations (4.43,4.44) , it is clear that the disturbance $\varepsilon_q(\alpha_q\dot{\mathbf{q}} - M^{-1}\tilde{\mathbf{N}})$ and $\varepsilon_x(\alpha_x\dot{\mathbf{x}} - \Lambda^{-1}\tilde{\mathbf{H}})$ will vanish at high-gains since $\varepsilon_{q,x} \to 0$. As a result, the closed-loop equations reduce to the fast reduced subsystem:

$$\frac{d\mathbf{s}_q}{d\tau} = -M^{-1}\hat{M}\mathbf{s}_q = -\Omega_q\mathbf{s}_q \tag{4.45}$$

$$\frac{d\mathbf{s}_x}{d\tau} = -\Lambda^{-1}\hat{\Lambda}\mathbf{s}_x = -\Omega_x\mathbf{s}_x \tag{4.46}$$

Note that the inertia matrices M and Λ are always positive definite, thus if the estimated/identified inertia matrices $(\hat{M}, \hat{\Lambda})$ are also positive definite as discussed in the previous chapter, the eigenvalues of Ω_q and Ω_x will be all positive [82] (note that this does not imply that (Ω_q, Ω_x) are positive definite). As a result, $\mathbf{s}_q$ and $\mathbf{s}_x$ tend to $\mathbf{0}$ exponentially as discussed in [81]. As is seen from above analysis, as long as the control gains can be increased, the effects of the model uncertainties:

$$\mathbf{D}_q = \varepsilon_q(\alpha_q\dot{\mathbf{q}} - M^{-1}\tilde{\mathbf{N}}) \tag{4.47}$$

$$\mathbf{D}_x = \varepsilon_x(\alpha_x\dot{\mathbf{x}} - \Lambda^{-1}\tilde{\mathbf{H}}) \tag{4.48}$$

on the closed-loop response can be made arbitrary small and the behaviour of the closed-loop systems (Eq. 4.38, 4.39) can be defined by adjusting α_q and α_x. In practice, because the control laws (Eq. 4.28, 4.29) are usually implemented using digital computers, thus, the control gains will have upper limits as discussed in [81] and section IV of [83]. This observation raises a question on how these gain limits restrict the response of the closed-loop systems (Eq. 4.38- 4.39) in practice.

Before discussing the effects of the discrete high-gain computed-torque control in joint space and task space, let us summarise the question in hand again using the following assumptions:

(i) Assume that we have an identified dynamic model of the robot in joint space $(\hat{M}, \hat{\mathbf{N}})$. The equivalent task space dynamics can be obtained using (Eqs. 4.5-4.6).

(ii) Let the task in joint space and task space be exactly the same i.e. $\mathbf{x}_d = P(\mathbf{q}_d)$, where $P(\mathbf{q})$ is the forward kinematics of the robot. In addition, let us assume that the kinematics model is accurately known.

(iii) Let the control laws (Eqs. 4.28-4.29) be digitally implemented with the same sampling period T, and assume that the control gains are chosen high enough so that the closed-loop systems can be approximated by (Eqs. 4.45-4.46).

The question we are interested in here is how the responses of the closed-loop systems (Eq. 4.43, 4.44) will be.

Let us first rewrite the above closed-loop systems as:

$$\frac{d\mathbf{s}_q}{d\tau} = -M^{-1}\hat{M}\mathbf{s}_q$$

$$\Leftrightarrow \frac{d\mathbf{s}_q}{dt} = M^{-1}\hat{M}\frac{-1}{\varepsilon_q}\mathbf{s}_q$$

$$\Leftrightarrow \ddot{\mathbf{q}} = -\alpha_q\dot{\mathbf{q}} + M^{-1}\hat{M}\mathbf{u}_q$$

$$\Leftrightarrow \frac{d}{dt}\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -\alpha_q\end{bmatrix}\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix} + \begin{bmatrix}0_{n\times n}\\ I_{n\times n}\end{bmatrix}M^{-1}\hat{M}\mathbf{u}_q$$

$$\Leftrightarrow \frac{d}{dt}\mathbf{Q} = A_q\mathbf{Q} + B_qM^{-1}\hat{M}\mathbf{u}_q \tag{4.49}$$

and similarly

$$\frac{d\mathbf{s}_x}{d\tau} = -\Lambda^{-1}\hat{\Lambda}\mathbf{s}_x$$

$$\Leftrightarrow \frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -\alpha_x\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\Lambda^{-1}\hat{\Lambda}\mathbf{u}_x$$

$$\Leftrightarrow \frac{d}{dt}\mathbf{X} = A_x\mathbf{X} + B_x\Lambda^{-1}\hat{\Lambda}\mathbf{u}_x \tag{4.50}$$

It is important to remark that, for the ease of representation, 0 and 1 in the equations should be read as the **zero** and **identity** matrix with **appropriate** dimension in this Chapter. Assume that $M^{-1}\hat{M}$ can be approximated by $\Omega_q[k]$ and $\Lambda^{-1}\hat{\Lambda}$ can be approximated by $\Omega_x[k]$ during the k sampling period (i.e. $kT \le t < (k+1)T$). The discrete forms of the above equations (zero-order hold), under the assumption that the computation time of the control law is negligible, are:

$$\mathbf{Q}[k+1] = e^{A_q T}\mathbf{Q}[k] + \left(\int_0^T e^{A_q \tau} d\tau\right) B_q M^{-1}\hat{M}\mathbf{u}_q[k] \tag{4.51}$$

$$\mathbf{X}[k+1] = e^{A_x T}\mathbf{X}[k] + \left(\int_0^T e^{A_x \tau} d\tau\right) B_x \Lambda^{-1}\hat{\Lambda}\mathbf{u}_x[k] \tag{4.52}$$

where:

$$e^{A_{q,x} T} = \begin{bmatrix} 1 & \frac{1-e^{-\alpha_{q,x}T}}{\alpha_{q,x}} \\ 0 & e^{-\alpha_{q,x}T} \end{bmatrix} \tag{4.53}$$

$$\left(\int_0^T e^{A_x \tau} d\tau\right) B_x = \begin{bmatrix} \frac{-1+e^{-\alpha_{q,x}T}+\alpha_{q,x}T}{\alpha_{q,x}^2} \\ \frac{1-e^{-\alpha_{q,x}T}}{\alpha_{q,x}} \end{bmatrix} \tag{4.54}$$

For simplicity, let us assume that $\mathbf{q}_d = \mathbf{x}_d = 0$. Since the control input $\mathbf{u}_{q,x}$ can be rewritten as:

$$\mathbf{u}_q = \begin{bmatrix} -\frac{\alpha_q}{\varepsilon_q} & \frac{1}{\varepsilon_q} \end{bmatrix} \mathbf{Q}[k] \tag{4.55}$$

$$\mathbf{u}_x = \begin{bmatrix} -\frac{\alpha_x}{\varepsilon_x} & \frac{1}{\varepsilon_x} \end{bmatrix} \mathbf{X}[k] \tag{4.56}$$

The discrete forms of the closed-loop systems become:

$$\begin{aligned} &\begin{bmatrix} \mathbf{q}[k+1] \\ \dot{\mathbf{q}}[k+1] \end{bmatrix} = \\ &\frac{1}{\varepsilon_q}\begin{bmatrix} 1+\frac{1}{\alpha_q}\left(1-e^{-\alpha_q T}-\alpha_q T\right)\Omega_q & \frac{1-e^{-\alpha_q T}}{\alpha_q}+\frac{1}{\alpha_q^2}\left(1-e^{-\alpha_q T}-\alpha_q T\right)\Omega_q \\ \left(e^{-\alpha_q T}-1\right)\Omega_q & e^{-\alpha_q T}+\frac{1}{\alpha_q}\left(e^{-\alpha_q T}-1\right)\Omega_q \end{bmatrix} \begin{bmatrix} \mathbf{q}[k] \\ \dot{\mathbf{q}}[k] \end{bmatrix} \\ &\Leftrightarrow \mathbf{Q}[k+1] = \Phi_q \mathbf{Q}[k] \end{aligned} \tag{4.57}$$

and

$$\begin{aligned}&\begin{bmatrix}\mathbf{x}[k+1]\\ \dot{\mathbf{x}}[k+1]\end{bmatrix}=\\ &\frac{1}{\varepsilon_x}\begin{bmatrix}1+\frac{1}{\alpha_x}\left(1-e^{-\alpha_x T}-\alpha_x T\right)\Omega_x & \frac{1-e^{-\alpha_x T}}{\alpha_x}+\frac{1}{\alpha_x^2}\left(1-e^{-\alpha_x T}-\alpha_x T\right)\Omega_x\\ \left(e^{-\alpha_x T}-1\right)\Omega_x & e^{-\alpha_x T}+\frac{1}{\alpha_x}\left(e^{-\alpha_x T}-1\right)\Omega_x\end{bmatrix}\begin{bmatrix}\mathbf{x}[k]\\ \dot{\mathbf{x}}[k]\end{bmatrix}\\ &\Leftrightarrow \mathbf{X}[k+1]=\Phi_x\mathbf{X}[k]\end{aligned} \tag{4.58}$$

Since the eigenvalues of $\Phi_{q,x}$ are $(e^{-T\alpha_{q,x}}, \frac{\varepsilon_{q,x}-T\lambda_{qi,xi}}{\varepsilon_{q,x}})$, where $\lambda_{qi,xi}$ is the eigenvalues of Ω_q and Ω_x, the stability of the above systems can only be guaranteed when then magnitude of the eigenvalues are within the unit circle [84]. As a result, the maximum value that the control gains can take will depend on the maximum of the eigenvalues of Φ_q and Φ_x.

To see the effect of the upper limit of the control gains on the joint space and task space control performance, let us first consider the following Lemma:

Lemma 4.1: Under the above assumptions $(i), (ii), (iii)$, the upper limits of the control gains of the joint space and task space controller (Eqs. 4.28-4.29) are the same.

Proof: Substitute Eq. 4.5 into $\Omega_{\mathbf{x}}$ (assume that there are no redundancy and singularity) leads to:

$$\Omega_x=\Lambda^{-1}\hat{\Lambda}=JM^{-1}\hat{M}J^{-1}=J\Omega_q J^{-1} \tag{4.59}$$

Note that:

$$\begin{bmatrix}J & 0\\ 0 & J\end{bmatrix}^{-1}=\begin{bmatrix}J^{-1} & 0\\ 0 & J^{-1}\end{bmatrix} \tag{4.60}$$

because

$$\begin{bmatrix}J & 0\\ 0 & J\end{bmatrix}\begin{bmatrix}J^{-1} & 0\\ 0 & J^{-1}\end{bmatrix}=\begin{bmatrix}J^{-1} & 0\\ 0 & J^{-1}\end{bmatrix}\begin{bmatrix}J & 0\\ 0 & J\end{bmatrix}=\begin{bmatrix}I & 0\\ 0 & I\end{bmatrix} \tag{4.61}$$

Thus, Φ_x can be rewritten as:

$$\begin{aligned}\Phi_x =& \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix} \\ &\begin{bmatrix} 1+\frac{1}{\alpha_x}\left(1-e^{-\alpha_x T}-\alpha_x T\right)\Omega_q & \frac{1-e^{-\alpha_x T}}{\alpha_x}+\frac{1}{\alpha_x^2}\left(1-e^{-\alpha_x T}-\alpha_x T\right)\Omega_q \\ \left(e^{-\alpha_x T}-1\right)\Omega_q & e^{-\alpha_x T}+\frac{1}{\alpha_x}\left(e^{-\alpha_x T}-1\right)\Omega_q \end{bmatrix} \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix}^{-1} \\ =& \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix} \bar{\Phi}_q \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix}^{-1}\end{aligned} \tag{4.62}$$

Note that by similar matrix property, Φ_x and $\bar{\Phi}_q$ have the same set of eigenvalues [85]. As the above discussion, Eq. 4.62 implies that the upper limits of the control gains for both systems are the same.

From *Lemma 4.1*, it is clear that the responses of the closed-loop systems Eqs. 4.43-4.44 will now depend on how significant the disturbances are (Eqs. 4.47-4.48). The reason is because $\varepsilon_{q,x}$ cannot be arbitrarily reduced to zero to eliminate the effects of the model uncertainties, as discussed in the continuous case. To see the effects of model uncertainties on the closed-loop systems, let us further expand the disturbance terms (Eq. 4.47-4.48):

$$\mathbf{D}_q = \alpha_q \dot{\mathbf{q}} - M^{-1}\tilde{\mathbf{C}} \tag{4.63}$$

$$\mathbf{D}_x = \alpha_x \dot{\mathbf{x}} - \Lambda^{-1}\tilde{\boldsymbol{\mu}} = J(\alpha_x \dot{\mathbf{q}} - M^{-1}\tilde{\mathbf{C}}) + \left(I - JM^{-1}\hat{M}J^{-1}\right)\dot{J}\dot{\mathbf{q}} \tag{4.64}$$

As is seen, the joint space closed-loop system Eq. 4.43 is disturbed by Eq. 4.63 and the responses can be transformed to the operational space using the kinematics relationship between the joint space and task space. However, if the control is done in task space, the closed-loop system Eq. 4.44 has to cope with the disturbance as shown in Eq. 4.64 which is the result of the joint space disturbance multiplied by the Jacobian. Moreover, the uncertainties of the inertia matrix also appear as an extra term in the disturbance equation. As a result, the discrepancy between the joint space and task space control performance can be explained as the amplification of errors due to the Jacobian coupled with the upper control gain limit (i.e. control gains cannot be further increased to

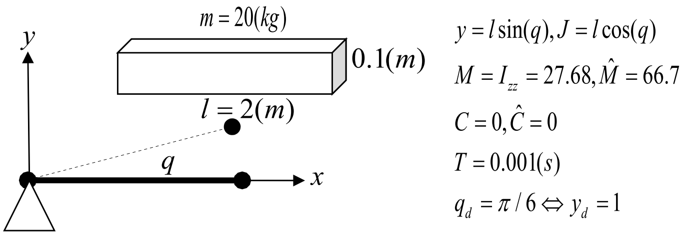

Figure 4.12: One link robot model.

compensate for the model uncertainties) as shown in the previous section.

To verify the above observation, let us consider the simulation of a 1-DOF robot as shown in Figure 4.12. This simulation was done using SimMechanics Toolbox under MatLAB/Simulink environment. For simplicity, let us choose the control gains k_D, k_P as (Hurwitz polynomial):

$$k_D = 2w = \frac{1}{\varepsilon} \tag{4.65}$$

$$k_P = w^2 = \alpha\frac{1}{\varepsilon} \tag{4.66}$$

The control laws for joint space and task space are Eqs. 4.28 and 4.29 accordingly. Based on the above discussion, the upper limit of the control gains is $w < 200$ for both joint space and task space. Two simulations have been carried out to show the effect of model uncertainties on the control performance. The first simulation has been used to illustrate the existence of the upper limit of the control gains. Figure 4.13 shows the response of the joint space controller (Eq. 4.28) for some w (the responses in task space are similar and thus not shown here). Clearly when w is near to the theoretical unstable value (i.e. 200), chattering occurred. In order to evaluate the control performances, the difference between the task space responses $y_e = y_x - y_q$ is plotted in Figure 4.14 for the second simulation. Here, $y_q = l\sin(q)$ is the response of the controller Eq. 4.28 and $y_x = l\sin(q)$ is the response of the controller Eq. 4.29. As is seen, the difference y_e tends to be negative which implies that y_q is toward y_d faster than y_x. In other words, under the same control gains (the maximum gains that the discrete high-gain system can take), the PD joint space controller provides a better task space response in comparison

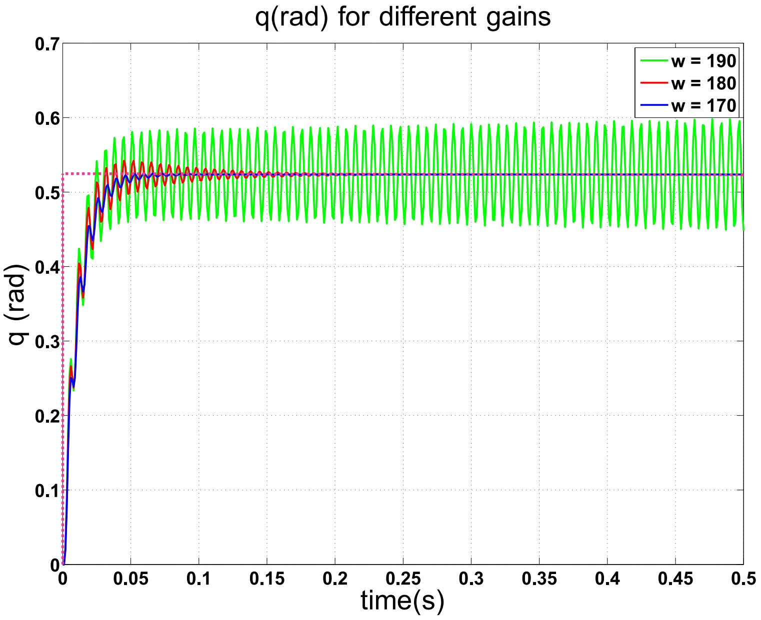

Figure 4.13: Joint space responses of the one-DOF system under various control gains.

to the conventional task space controller.

4.2 Effects of Model Uncertainties on JS and TS Control - Experiments

To reinforce the above observation from the simulation, let us implement the above simulation on the last link, link 7, of the Mitsubishi PA10 manipulator. For simplicity, let us consider the following approximation model of link 7:

$$i_{zz7}\ddot{q}_7 + \tau_{fric} = \tau \tag{4.67}$$

$$y = lsin(q_7) \tag{4.68}$$

It is noted that the gravity effect has been eliminated in the above equation by putting the robot in such a way that the gravity vector is parallel to the z axis of joint 7 (Chapter 3). Since i_{zz7} and τ_{fric} are practically unknown, the following values are used in the

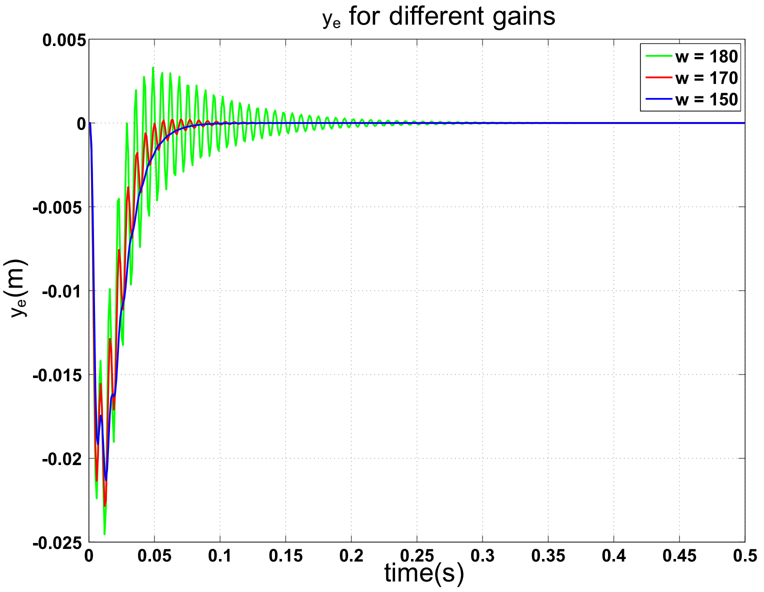

Figure 4.14: Task space response's difference between the joint space and task space set-point control for various control gains.

controller:

$$l = 1 \tag{4.69}$$

$$\hat{i}_{zz7} = 1 \tag{4.70}$$

$$\hat{\tau}_{fric} = 0 \tag{4.71}$$

The joint space and task space set-point controller are implemented as shown in Eq. 4.28 and Eq. 4.29 in the QNX real-time operating system at 1 kHz i.e. $T = 0.001s$. The control gains i.e. K_P, K_D have been chosen as mentioned in the simulation case. The initial configuration is $q_7 = 0$ or $y = lsin(q_7) = 0$ and the desired configuration is $q_d = \frac{\pi}{12} = 0.2618rad$ or $y_d = 0.2588m$. Figures 4.15-4.16 show the joint responses for some w using the joint space set point controller. As is seen, when $w > 8$, chattering did occur.

The responses from the task space set point controller are shown in Figure 4.17-4.18. It is noted that the chattering effect also occurs when $w > 8$ which is consistent to the above analysis ((Φ_q, Φ_x) have the same set of eigenvalues).

The difference between the responses of the joint space and task space controller y_e

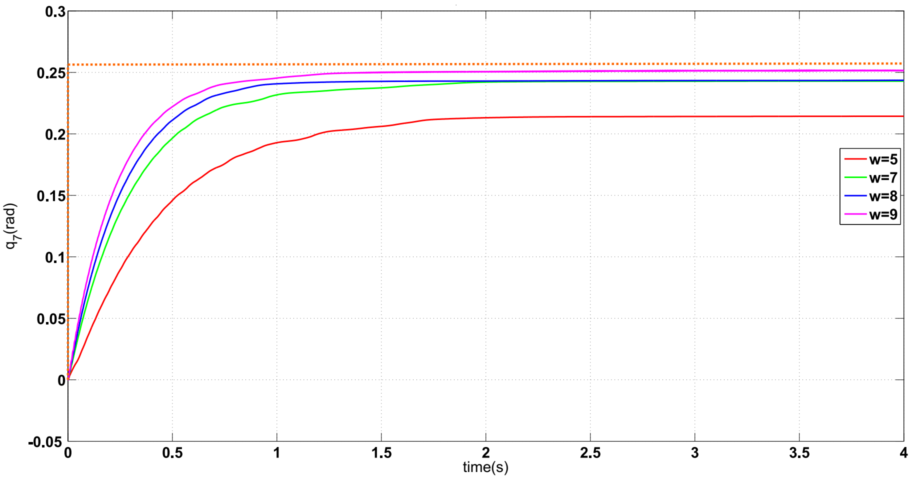

Figure 4.15: Responses from the joint space set-point controller.

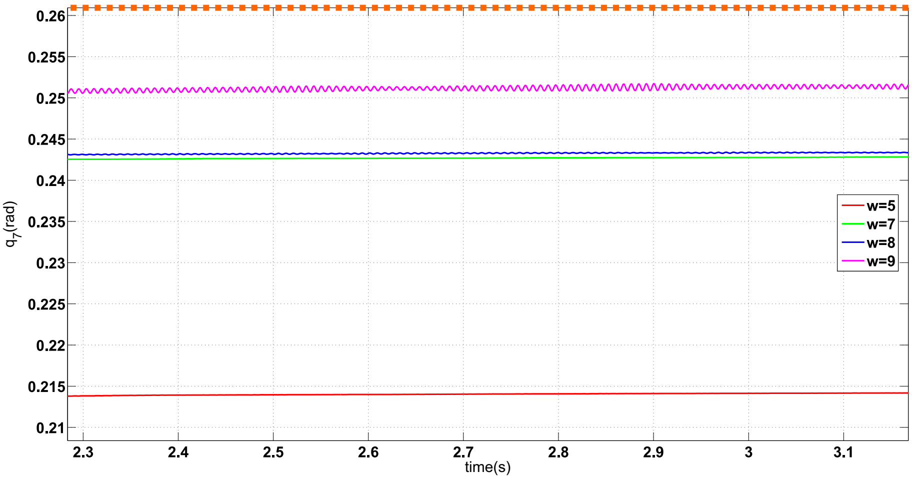

Figure 4.16: Responses (zoom-in) from the joint space set-point controller.

at $w = 9$ is plotted in Figures 4.19-4.20. Note that $y_e = y_x - y_q$ tends to be negative indicate that the response of y_q is faster. In other words, for the same control gain, y_q (response from Eq. 4.28) approaches $y_d = 0.2588m$ faster than y_x (response from Eq. 4.29). It is worth stressing that the difference between the responses in this experiment is expected to be small because only a simple one-DOF system is considered. In practice, this difference can be much more significant depending on how the kinematic and dynamic parameters of the robot are structured.

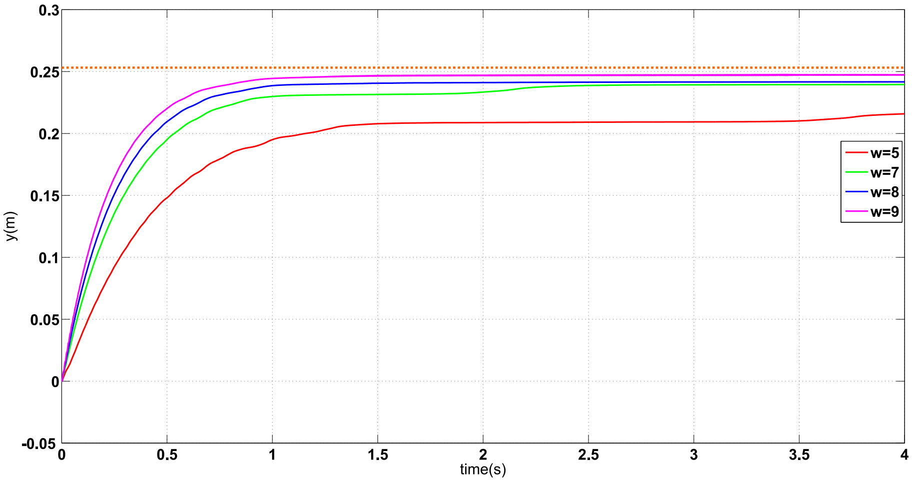

Figure 4.17: Responses from the task space set-point controller.

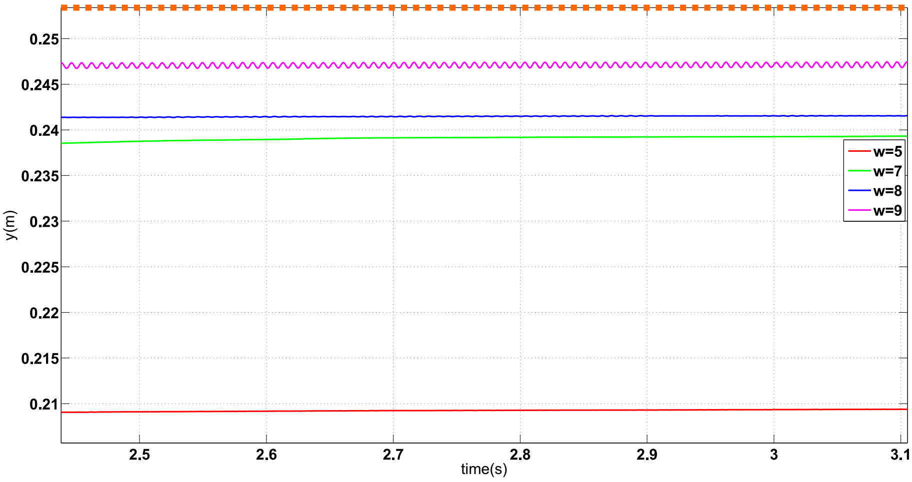

Figure 4.18: Responses (zoom-in) from the task space set-point controller.

4.3 Conclusion

In conclusion, the above analysis suggests that the inaccuracies of the dynamic model can produce different effects depending on the space that it is used. It is also shown in this Chapter that due to the discretisation effects, the control gains for the popular PD set-point controller in joint space and task space are the same. This observation suggests that the closed-loop control performance can be different for different control space. If the kinematic model of the robot happens to magnify the modelling errors, the task space control performance can be much degraded due to the upper limit of control gains. The work in this chapter not only provide an insight of the effect of model

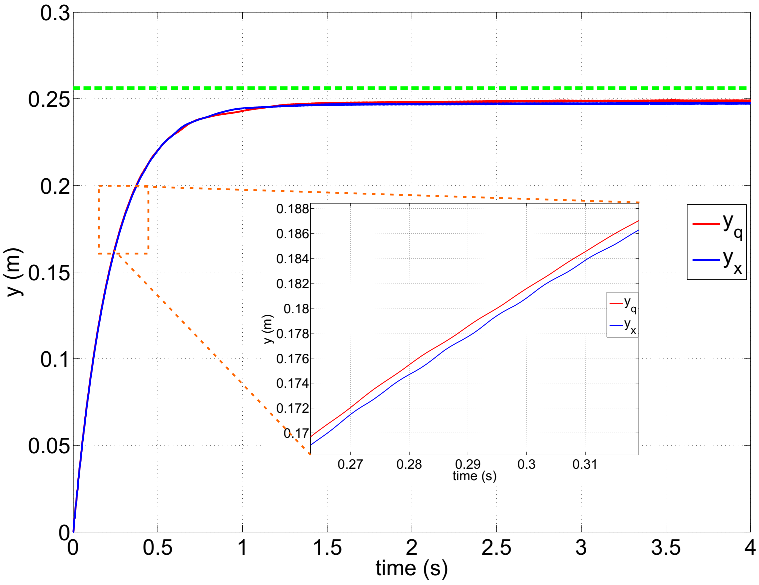

Figure 4.19: Task space response's of the joint space and task space set-point controller of the 1-DOF robot at $w = 9$.

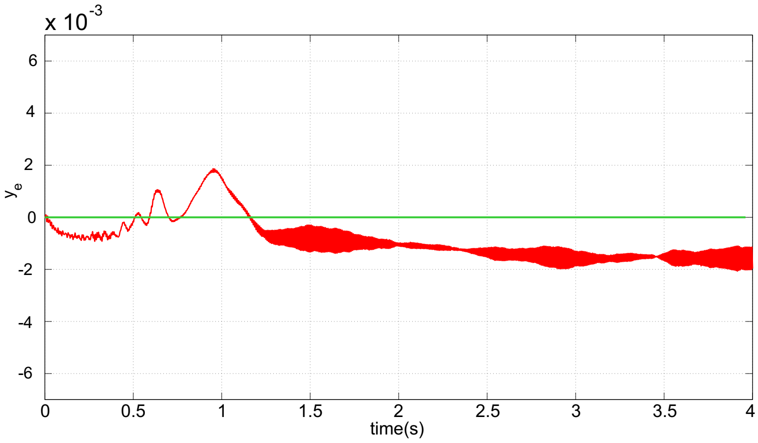

Figure 4.20: Task space response's difference between the joint space and task space set-point controller of the 1-DOF robot at $w = 9$.

uncertainties on control performance but also suggest that it is necessary to seek for an alternative control structure for the force-based operational space since the performance of this control framework heavily depends on the accuracy of the task space dynamic model.

Chapter 5

Dual-loop Control Structure for The Force-based Operational Space Control

From the discussion in the previous chapter, it was clear that the identified dynamics should be rather used in joint space than task space to minimise its effects on the control performance. However, since task space dynamics play an important role on compliant motion tasks [13, 14], it is necessary to seek for an alternative formulation to minimise the effects of the model uncertainties on task space control performance, while maintaining all the advantage of the above force-based operational space. One promising approach is to use a high frequency velocity feedback control loop at each joint. This approach will result in a so called dual-loop/inner-outer loop implementation [31, 32]. It is interesting to note that one of the motivations of this approach comes from the fact that industrial robots usually accompany a motion controller at each joint, thus, task space control capability can only be achieved by making an outside loop. In other words, the outer loop is closed in task space to generate the reference joint velocity using resolved motion rate control scheme. Subsequently, this reference joint velocity is passed to the inner loop which will compensate for the effects of joint friction and realise this reference velocity in a faster time scale compared with the outer loop. However, this control scheme does not take into account the system dynamics in both task space and null space. Therefore, the task (range) space and null space cannot be dynamically decoupled as in the force-based operational space control. In this chapter, we will explore an alternative solution, our dual-loop structure for enhancing the force-based operational space control framework. Note that this control structure is not new in the sense that it has been mentioned in some previous work such as [31–33]. The contribution here is that the inverse dynamics concept has been shifted from the operational space into joint space in order to minimise the effects (if possible) of the model uncertainties. It is also interesting to note that the experiment result(s) in the previous research was shown to give a good performance in task space control (for instance, Section 6 of [32]). Nevertheless, there is no explicit explanation of why the dual-loop structure is insensitive to model uncertainties. In our proposed control structure, since the motivation is to minimise the effects of model uncertainties, the analyses from Chapter 4 can be considered to be one of the major reasons of the significant improvement in task space control.

The rest of this chapter will be presented as follows: first, we propose the dual-loop control structure for the force-based operational control framework. Stability analysis of the proposed controller in the continuous domain is presented soon after. It is shown that the proposed control structure is uniformly ultimately bounded. It is important to stress that the stability analysis in this section only serves as a necessary condition for the usefulness of the proposed controller because we do not account for discretising effects, signal noise and so on. Thus, the last section of this chapter will present the intensive experimental result(s) of the proposed control structure on the Mitsubishi PA-10 manipulator to show its usefulness in practice.

5.1 Dual-loop Operational Space Control Structure

In order to see why this control structure was proposed, let us recall on how the force-based operational space control make use of the robot dynamic model to achieve its advantages. One of the key advantages of the OSC which was first proposed in [13] is to describe the robot dynamics (at the operational point) in the operational space. As a result, the control formulation has also been formulated using the so-called task space dynamics. Under the assumption of perfect dynamics model, it was shown that the task space dynamics can be used to:

- Linearise the nonlinear robot system: as a result, the closed-loop system reduces to n double integrator which can be easily stabilised by a standard PD controller [13, 14]. A direct consequence is that the task space control performance is supposed to be isotropic for any working condition (i.e. high-speed and low-speed tasks) and any direction.

- Unify the motion and force control within one control framework, the operational space framework: as a result, motion and force can be controlled independently. This is significant for applications that require a stringent control performance such as small tracking errors for both motion and force directions.

- Dynamically decouple the task (range) space and null space for redundant robots: as a result, the robot working space can be easily decomposed into spaces. These spaces can be prioritised and controlled by separated controllers as discussed in [14]. This capability is significant because it utilises the full DOF of the robot for any given task. Interested readers can refer to the reference [14] for a detailed description.

In practice, perfect dynamics model are normally not achievable. Thus, the control performance of the above force-based operational space control framework can be significantly degraded as discussed in [40] and Chapter 4 of this work. Also from the discussion in Chapter 4, the inaccurate dynamic model is better used in joint space than in task space to minimise its effects on the control performance. It is thus natural to move the feed-back linearisation concept from the task space to joint space. To this end, a dual-loop operational space control structure is proposed as follows [86][1]:

- Firstly, a lower-sampling-rate outer OSC loop is used to achieve motion and force tracking in task space and dynamically decouple the task and null space. To do this, the operational space command force is computed as shown in Eq. 2.10-2.14

---

[1] We referred to the ”dual-loop operational space control” as the ”multi-rate operational space control” in our prior publications but ”dual-loop” is a more accurate term.

using the identified dynamics model as a reference. This task space command is then applied to the joint space identified model to get the joint acceleration commands. For example, for free motion control task, $\ddot{\mathbf{q}}_d$ can be computed as follows.

$$\ddot{\mathbf{q}}_d = \hat{M}^{-1} J^T \hat{\Lambda}(\mathbf{u}_x - \dot{J}\dot{\mathbf{q}}) + (I - \hat{M}^{-1} J^T \hat{\Lambda} J)\hat{M}^{-1} S_n^T (\mathbf{F}_{null} - \hat{\mathbf{N}}) \tag{5.1}$$

Since

$$\hat{\bar{J}} = \hat{M}^{-1} J^T (J \hat{M}^{-1} J^T)^{-1} \tag{5.2}$$

Eq 5.1 can be rewritten as:

$$\ddot{\mathbf{q}}_d = \hat{\bar{J}}(\mathbf{u}_x - \dot{J}\dot{\mathbf{q}}) + (I - \hat{\bar{J}} J)\hat{M}^{-1} S_n^T \mathbf{F}_{null} \tag{5.3}$$

$$\mathbf{u}_x = \ddot{\mathbf{x}}_d + K_V(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + K_P(\mathbf{x}_d - \mathbf{x}) \tag{5.4}$$

It is worth noting that Eq. 5.2 is actually a inertia-weighted pseudo-inverse at the acceleration level [42]. Thus, Eq. 5.3 will give a joint space response with respect to a task space command $\{\ddot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \mathbf{x}_d\}$ through the identified dynamic model $\hat{M}$. In other words, the purpose of the outer loop is to transform the task space command to the joint space command using the identified dynamic model. By integrating this reference acceleration output, the reference joint velocities for the inner loop can be obtained. These reference joint velocities are then realised by the PI inner-loop (higher sampling rate) control as shown below.

- Secondly, a higher-sampling-rate inner joint velocity control loop which makes use of the identified dynamic model is used in *joint space* to reject the effects of local disturbances. Since the dynamic model enables the use of high-gain as is seen in Eq. 4.45-4.46, the identified dynamic model can be used here to enhance the performance as discussed in [32]. The input of this controller is the reference joint velocities and accelerations. The controller can be stated as:

$$\mathbf{\Gamma}_{JS} = \hat{A}\mathbf{u}_{JS} + \hat{\mathbf{C}} + \hat{\mathbf{G}} + \hat{\mathbf{\Gamma}}_{Fric} \tag{5.5}$$

$$\mathbf{u}_{JS} = \ddot{\mathbf{q}}_d + K(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + K_I \int (\dot{\mathbf{q}}_d - \dot{\mathbf{q}})dt \tag{5.6}$$

where K, K_I are control gains in joint space. The desired joint velocity $\dot{\mathbf{q}}_d$ can be obtained by integrating the desired joint acceleration Eq. 5.3:

$$\dot{\mathbf{q}}(t + \Delta t)_d \triangleq \int_{t}^{t+\Delta t} \ddot{\mathbf{q}}(t)_d dt \tag{5.7}$$

with the initial condition as the current $\{\dot{\mathbf{q}}, \mathbf{q}\}$. Clearly, if the inner velocity control loop is able to bring the manipulator from the current state $\{\dot{\mathbf{q}}(t), \mathbf{q}(t)\}$ to $\{\dot{\mathbf{q}}(t + \Delta t), \mathbf{q}(t + \Delta t)\}$ after Δt (sec), the behaviour of the robot will be exactly determined by the identified dynamic model. Because the assumption that the inner velocity control loop can change the system states in Δt (sec) is usually violated in practice, an outer loop, which is the force-based operational space control, is always necessary to ensure the task space control performance.

Figure 5.1 depicts the above dual-loop operational space control structure that has been used in the implementation in the experiment section. To further demonstrate the above control structure, consider the 3-DOF(RRR) robot with the same task as presented in Section 2.2.1 (Figure 5.1). Assume that the robot model is available, two control schemes, the original force-based OSC and the above dual-loop OSC, are used in the simulation. Figure 5.3-5.5 show the simulation responses of joint 1-3. Blue is the responses of the original force-based operational space control (OSC) and red is the responses of the proposed control structure, the modified OSC (mOSC). Clearly, if the robot model is accurately known, the responses of the original force-based OSC and the dual-loop OSC are **identical**.

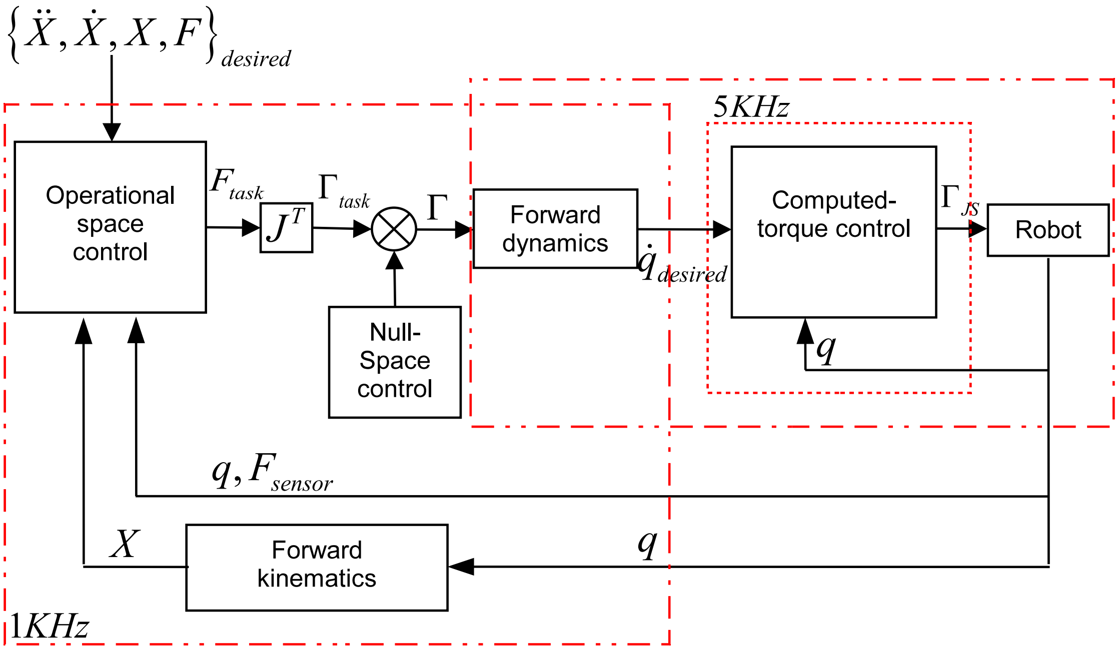

Figure 5.1: The dual-loop operational space control structure.

5.2 Stability Analysis

Since the above dual-loop OSC uses the robot model at two different levels (outer and inner loop), it is necessary to investigate the stability of the proposed controller, at least for the continuous case. For simplicity, let us only focus on the stability of the motion controller (trajectory tracking) since force control capability of the hybrid motion/force control is also achieved through position regulation [34]. Moreover, since the concentration of the proposed control structure is on the task space control performance, it is reasonable to solely focus on non-redundant case (i.e. no null space control). Note that the stability analysis in this section only serves as *a necessary condition* for the usefulness of the proposed controller. If the control law is digitally implemented, the performance of the closed-loop system will now depend on how high the control gains can take as discussed in Chapter 4.

To analyse the stability of the proposed controller, let us apply Eq 5.6 to the robot dynamics equation (Eq. 4.1), the closed-loop equation can be rewritten as (after some

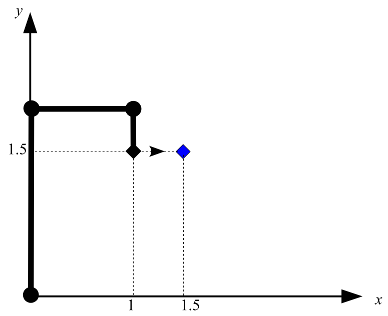

Figure 5.2: Initial configuration the 3-DOF(RRR) robot. The desired position has been marked as a blue square. Note that quintic polynomial has been used for trajectory planning.

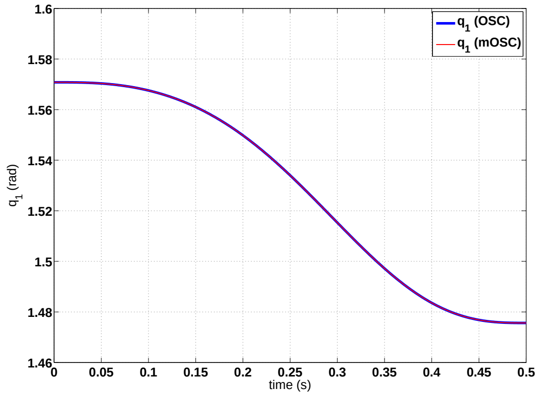

Figure 5.3: Response of q_1 for the conventional OSC (OSC) and the dual-loop OSC (mOSC).

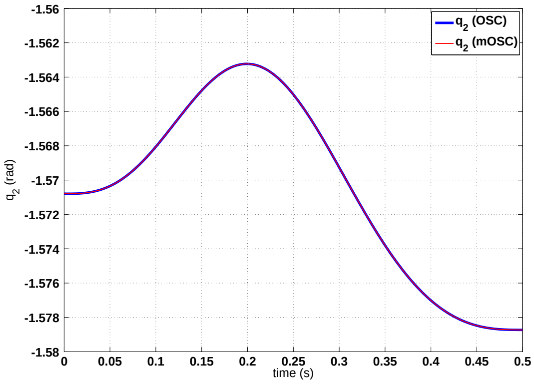

Figure 5.4: Response of q_2 for the conventional OSC (OSC) and the dual-loop OSC (mOSC).

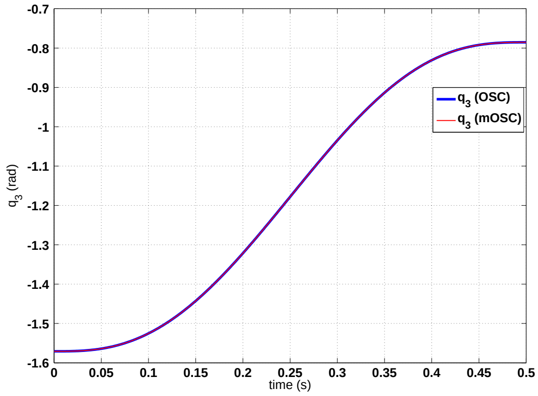

Figure 5.5: Response of q_3 for the conventional OSC (OSC) and the dual-loop OSC (mOSC).

manipulation):

$$\begin{aligned} \ddot{\tilde{\mathbf{x}}} + K_V\dot{\tilde{\mathbf{x}}} + K_P\tilde{\mathbf{x}} &= J\dot{\tilde{\mathbf{w}}} \\ \dot{\tilde{\mathbf{w}}} + M^{-1}\hat{M}K\tilde{\mathbf{w}} + M^{-1}\hat{M}K_I\mathbf{z} &= M^{-1}\tilde{M}\ddot{\mathbf{q}}_d + M^{-1}\tilde{\mathbf{H}} \\ \dot{\mathbf{z}} &= \tilde{\mathbf{w}} \end{aligned} \tag{5.8}$$

where:

$$
\begin{aligned}
\ddot{\tilde{\mathbf{x}}} &= \ddot{\mathbf{x}}_d - \ddot{\mathbf{x}} \\
\dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}}_d - \dot{\mathbf{x}} \\
\tilde{\mathbf{x}} &= \mathbf{x}_d - \mathbf{x} \\
\tilde{\mathbf{w}} = \dot{\tilde{\mathbf{q}}} &= \dot{\mathbf{q}}_d - \dot{\mathbf{q}} \\
\dot{\tilde{\mathbf{w}}} = \ddot{\tilde{\mathbf{q}}} &= \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} \\
\tilde{\mathbf{C}} = C_M\dot{\mathbf{q}} - \hat{C}_M\dot{\mathbf{q}} &= \mathbf{C} - \hat{\mathbf{C}} \\
\tilde{\mathbf{G}} &= \mathbf{G} - \hat{\mathbf{G}} \\
\tilde{\mathbf{H}} &= \tilde{\mathbf{C}} + \tilde{\mathbf{G}} + \tilde{\mathbf{\Gamma}}_{Fric}
\end{aligned}
\tag{5.9}
$$

Note that the first equation of Eqs. 5.8 implies that the task space control performance will be disturbed by the joint space dynamics $\dot{\tilde{\mathbf{w}}}$ multiplies the Jacobian.

Rewriting the equations in the matrix form results in:

$$
\frac{d}{dt}\begin{bmatrix} \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \\ \tilde{\mathbf{w}} \\ \mathbf{z} \end{bmatrix} =
$$

$$
\begin{bmatrix}
0 & I & 0 & 0 \\
-K_P & -K_V & -JM^{-1}\hat{M}K & -JM^{-1}\hat{M}K_I \\
0 & 0 & -M^{-1}\hat{M}K - M^{-1}C_M - \alpha M^{-1}M - M^{-1}K & -M^{-1}\hat{M}K_I - \alpha M^{-1}C_M \\
0 & 0 & I & 0
\end{bmatrix}
\begin{bmatrix} \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \\ \tilde{\mathbf{w}} \\ \mathbf{z} \end{bmatrix}
+
\begin{bmatrix}
0 \\
JM^{-1}\tilde{M}\ddot{\mathbf{q}}_d + JM^{-1}\tilde{\mathbf{H}} \\
\left(M^{-1}C_M + \alpha M^{-1}M + M^{-1}K\right)\tilde{\mathbf{w}} + \left(-\alpha M^{-1}C_M\right)\mathbf{z} + M^{-1}\tilde{M}\ddot{\mathbf{q}}_d + M^{-1}\tilde{\mathbf{H}} \\
0
\end{bmatrix}
\tag{5.10}
$$

$$
\Leftrightarrow \dot{\mathbf{X}} = \Omega\mathbf{X} + \mathbf{B} \tag{5.11}
$$

Here, we are interested in the stability property of the equilibrium:

$$\begin{bmatrix} \tilde{\mathbf{x}} & \dot{\tilde{\mathbf{x}}} & \tilde{\mathbf{w}} & \mathbf{z} \end{bmatrix}_e^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.12}$$

of the system Eq. 5.11 since it reflect the stability of the proposed controller. Also note that there are extra terms $-M^{-1}C_M - \alpha M^{-1}M - M^{-1}K$ in the Ω matrix, and they have been cancelled out later in vector $\mathbf{B}$. The purpose of these terms is to simplify the analysis for the nominal system as shown in the next section.

To analyse the stability of Eq. 5.11, we adopt the methodology proposed by Khalil [87], that is:

- Firstly, the asymptotic stability property of the nominal system $\dot{\mathbf{X}} = \Omega\mathbf{X}$ is studied.
- Next, the solution of the overall system Eq. 5.11 is shown to be uniformly ultimately bounded.

5.2.1 Stability of the Nominal System

Let us consider the following Lyapunov function candidate inspired by [50]:

$$V = \frac{1}{2}\mathbf{X}^T \begin{bmatrix} K_v & I & 0 & 0 \\ I & I & 0 & 0 \\ 0 & 0 & M & \alpha M \\ 0 & 0 & \alpha M & \alpha K + \alpha^2 M \end{bmatrix} \mathbf{X} = \mathbf{X}^T P \mathbf{X} \tag{5.13}$$

where:

$$K = kI \tag{5.14}$$

$$K_p = k_p I \tag{5.15}$$

$$K_v = k_v I \tag{5.16}$$

$$K_I = kiI \tag{5.17}$$

$$\alpha = \frac{ki}{k} > 0 \tag{5.18}$$

Note that P is positive definite when $k > 0$ and $k_v > 1$ by Lemma 5.1 (Appendix B). The derivative of V is (after making use of the skew-symmetric property of the inertia matrix $\mathbf{X}^T \left(\dot{M} - C_M - C_M^T \right) \mathbf{X} = 0$):

$$\begin{aligned} \dot{V} &= - \begin{bmatrix} \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \\ \tilde{\mathbf{w}} \\ \mathbf{z} \end{bmatrix}^T \begin{bmatrix} K_P & \mathbf{0} & JM^{-1}\hat{M}K & JM^{-1}\hat{M}K_I \\ K_P & K_V - I & JM^{-1}\hat{M}K & JM^{-1}\hat{M}K_I \\ \mathbf{0} & \mathbf{0} & \hat{M}K + K & \hat{M}K_I \\ \mathbf{0} & \mathbf{0} & \alpha\hat{M}K & \alpha\hat{M}K_I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \\ \tilde{\mathbf{w}} \\ \mathbf{z} \end{bmatrix} \\ &= -\mathbf{X}^T \begin{bmatrix} Q_1 & Q_2 \\ \mathbf{0} & Q_3 \end{bmatrix} \mathbf{X} = -\mathbf{X}^T Q \mathbf{X} \end{aligned} \tag{5.19}$$

After some manipulation, using Lemma 5.2 (Appendix B) it can be shown that $Q > 0$ when:

(a). $Q_3 > 0$ i.e.

$$S(Q_3) = Q_3 = \begin{bmatrix} \hat{M}K + K & \alpha\hat{M}K \\ \alpha\hat{M}K & \alpha^2\hat{M}K \end{bmatrix} > 0 \tag{5.20}$$

by Lemma 5.1 (Appendix B): $Q_3 > 0 \Leftrightarrow \alpha, k > 0$

(b). The second condition $4\lambda_{\min}\left(S\left(Q_1\right)\right) > \left\|Q_2\right\|^2 \left\|S\left(Q_3\right)^{-1}\right\| > 0$ can be seen by noting that:

(i) The eigenvalues of $S(Q_1)$:

$$S(Q_1) = \begin{bmatrix} K_P & \frac{1}{2}K_P \\ \frac{1}{2}K_P & K_V - I \end{bmatrix} \tag{5.21}$$

is:

$$\lambda_i = \frac{1}{2}\left(-1 + k_p + k_v \pm \sqrt{k_p^2 + (k_p + 1 - k_v)^2} \right) \tag{5.22}$$

(ii) The induced norm of matrix $||S(Q_3)^{-1}||$ (from here onward, $||A||$ will be referred as the induced norm of matrix A [85]) can be found as follows:

$$S(Q_3) = Q_3 = \begin{bmatrix} \hat{M}K + K & \alpha\hat{M}K \\ \alpha\hat{M}K & \alpha^2\hat{M}K \end{bmatrix} = k\begin{bmatrix} \hat{M} + I & \alpha\hat{M} \\ \alpha\hat{M} & \alpha^2\hat{M} \end{bmatrix} \tag{5.23}$$

Since $\hat{M}$ is symmetric and positive definite, let $\hat{M}$ be transformed by:

$$\hat{M} = T\Lambda_{\hat{M}}T^{-1} \tag{5.24}$$

where $\Lambda_{\hat{M}}$ is diagonal matrix. $S(Q_3)$ now becomes:

$$S(Q_3) = k\begin{bmatrix} T\Lambda_{\hat{M}}T^{-1} + TIT^{-1} & \alpha T\Lambda_{\hat{M}}T^{-1} \\ \alpha T\Lambda_{\hat{M}}T^{-1} & \alpha^2 T\Lambda_{\hat{M}}T^{-1} \end{bmatrix} \tag{5.25}$$

$$= k\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix}\begin{bmatrix} \Lambda_{\hat{M}} + I & \alpha\Lambda_{\hat{M}} \\ \alpha\Lambda_{\hat{M}} & \alpha^2\Lambda_{\hat{M}} \end{bmatrix}\begin{bmatrix} T^{-1} & 0 \\ 0 & T^{-1} \end{bmatrix} \tag{5.26}$$

Since $Q_3 = Q_3^T > 0$, the induced matrix norm of Q_3 is: $\|Q_3{}^{-1}\| = \frac{1}{\min|\lambda_i(Q_3)|} = \frac{1}{\min\lambda(Q_3)}$. Because $min_{\lambda(S(Q_3))} \leq \frac{k}{2}\left(1 + \underline{\lambda}_{\hat{M}} + \alpha^2\underline{\lambda}_{\hat{M}} - \sqrt{(1 + \underline{\lambda}_{\hat{M}} + \alpha^2\underline{\lambda}_{\hat{M}})^{\,2} - 4\alpha^2\underline{\lambda}_{\hat{M}}}\right)$, where $\underline{\lambda}_{\hat{M}}$ is the lower smallest eigenvalue of $\hat{M}$,

$$\left\|S\left(Q_3\right)^{-1}\right\| = \left\|Q_3{}^{-1}\right\| \leq \frac{2}{1 + \underline{\lambda}_{\hat{M}} + \alpha^2\underline{\lambda}_{\hat{M}} - \sqrt{(1 + \underline{\lambda}_{\hat{M}} + \alpha^2\underline{\lambda}_{\hat{M}})^{\,2} - 4\alpha^2\underline{\lambda}_{\hat{M}}}} \tag{5.27}$$

(iii) The upper bound of the induced norm of

$$Q_2 = \begin{pmatrix} JM^{-1}\hat{M}K & JM^{-1}\hat{M}K_I \\ JM^{-1}\hat{M}K & JM^{-1}\hat{M}K_I \end{pmatrix} = \begin{pmatrix} L & \alpha L \\ L & \alpha L \end{pmatrix} \tag{5.28}$$

can be found by noting that ([88], Chapter 2):

$$\|Q_2\| = \left\| \begin{pmatrix} L & \alpha L \\ L & \alpha L \end{pmatrix} \right\| \leq \sqrt{\max\{2\|L\|, 2\alpha\|M\|\}}\sqrt{\max\{\|L\| + \alpha\|L\|, \|L\| + \alpha\|L\|\}} \tag{5.29}$$

Since $L = JM^{-1}\hat{M}K = kJM^{-1}\hat{M}$, L is bounded by: $||L|| \leq k\delta_L$ (because $M, \hat{M}, J$ are all functions of $sin(q_i), cos(q_i)$), thus $\|Q_2\|^2 \leq 2(\alpha+1)k^2k_L^2$.

After some manipulation, the above conditions **(a), (b)** becomes:

$$\begin{cases} k_p > 0 \\ k_v > \frac{1}{4}k_p + 1 \\ 1 > \alpha > 0 \\ \frac{1}{2}(k_p + k_v - 1 - \sqrt{k_p^2 + (k_p - k_v + 1)^2}) > \frac{(\alpha+1)k^2k_H^2}{(1+(1+\alpha^2)-\sqrt{(1+(1+\alpha^2))^2-4\alpha^2})} \end{cases} \tag{5.30}$$

As is seen, if the control gain k is fixed (i.e. after the inner-loop control is tuned), Eq 5.30 can always be satisfied by increasing the task space gains (k_p, k_v). As a result, the nominal system $\dot{\mathbf{X}} = \Omega\mathbf{X}$ is exponentially stable because [87]:

$$\dot{V} = -\mathbf{X}^TQ\mathbf{X} < -\lambda_{\min}(Q)||\mathbf{X}||^2, \lambda_{\min}(Q) > 0 \tag{5.31}$$

5.2.2 Stability of the Overall System

Although the equilibrium point $\mathbf{X}_e = \mathbf{0}$ of the nominal system (which is the error system without the disturbance term $\mathbf{B}$) is exponentially stable with a proper choice of gains, the equilibrium of the actual error system Eq. 5.11 can only be shown to have bounded stability. One main reason is because the disturbance $\mathbf{B}$ is a function of joint position and velocity, which are also the states of the error system. Before showing the boundedness stability, let us first recall some common and useful properties of the dynamic model (the proofs for these properties can be found from [59, 82, 89]):

- The joint space inertia matrix is bounded:

$$0 < \|M\| = \max|\lambda_i(M)| = \max\left(\lambda_i(M)\right) \leq \delta_M \tag{5.32}$$

$$0 < \|M^{-1}\| = \frac{1}{\min|\lambda_i(M)|} \leq \delta_{M^{-1}} \tag{5.33}$$

$$0 < \left\|\hat{M}\right\| = \max\left|\lambda_i\left(\hat{M}(q(t))\right)\right| \leq \delta_{\hat{M}} \tag{5.34}$$

$$0 < \left\|\tilde{M}\right\| = \left\|M - \hat{M}\right\| \leq \|M\| + \left\|\hat{M}\right\| = \delta_{\tilde{M}} \tag{5.35}$$

- The joint space gravity is bounded:

$$0 \leq \|\mathbf{G}\| \leq \delta_G \tag{5.36}$$

$$0 \leq \left\|\tilde{\mathbf{G}}\right\| = \left\|\mathbf{G} - \hat{\mathbf{G}}\right\| \leq \|\mathbf{G}\| + \left\|-\hat{\mathbf{G}}\right\| = \|\mathbf{G}\| + \left\|\hat{\mathbf{G}}\right\| = \delta_{\tilde{G}} \tag{5.37}$$

- Assume that the induce norm of the Jacobian is bounded i.e. $\|J\| = \sqrt{\lambda_{\max}\left(J^T J\right)} \leq \delta_J$, under the assumption of singularity-free, it can be shown that [27, 50]:

$$0 \leq \|\dot{\mathbf{q}}\| \leq \delta_{pJ}\left(\delta_{\dot{x}_d} + \|\mathbf{X}\|\right) = \delta_{\dot{q}_1} + \delta_{\dot{q}_2}\|\mathbf{X}\| \tag{5.38}$$

- Using the similar approach as shown in [50], it can be shown that:

$$0 \leq \|C_M\left(\mathbf{q}, \dot{\mathbf{q}}\right)\| \leq \delta_C\|\dot{\mathbf{q}}\| \leq \delta_{C1} + \delta_{C2}\|\mathbf{X}\| \tag{5.39}$$

$$0 \leq \|C_M\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}}\| \leq \delta_C\|\dot{\mathbf{q}}\| \leq \delta_{Cq1} + \delta_{Cq2}\|\mathbf{X}\| + \delta_{Cq3}\|\mathbf{X}\|^2 \tag{5.40}$$

$$0 \leq \|\tilde{C}_M\dot{\mathbf{q}}\| \leq \delta_{\tilde{Cq}1} + \delta_{\tilde{Cq}2}\|\mathbf{X}\| + \delta_{\tilde{Cq}3}\|\mathbf{X}\|^2 \tag{5.41}$$

$$0 \leq \|\tilde{\mathbf{N}}\| \leq \delta_{\tilde{N}_1} + \delta_{\tilde{N}_2}\|\mathbf{X}\| + \delta_{\tilde{N}_3}\|\mathbf{X}\|^2 \tag{5.42}$$

$$0 \leq \|\ddot{\mathbf{q}}_d(t)\| \leq \delta_{\ddot{q}_{\mathrm{d}1}} + \delta_{\ddot{q}_{d2}}\|\mathbf{X}\| + \delta_{\ddot{q}_{d3}}\|\mathbf{X}\|^2 \tag{5.43}$$

Note that:

$$\left\| \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \\ \mathbf{y} \\ \mathbf{0} \end{pmatrix} \right\| = \left\| \begin{pmatrix} \|\mathbf{x}\| \\ \|\mathbf{y}\| \end{pmatrix} \right\| \tag{5.44}$$

for any vector $\mathbf{x}, \mathbf{y}$. Thus, after some manipulations, the disturbance $\mathbf{B}$ can be shown to be bounded as follows:

$$\|\mathbf{B}\| = \left\| \begin{pmatrix} \mathbf{0} \\ JM^{-1}\tilde{M}\ddot{\mathbf{q}}_d + JM^{-1}\tilde{\mathbf{N}} \\ \left(M^{-1}C_M + \alpha M^{-1}M + M^{-1}K\right)\tilde{\mathbf{w}} + \alpha M^{-1}C_M\mathbf{z} + M^{-1}\tilde{M}\ddot{\mathbf{q}}_d + M^{-1}\tilde{\mathbf{N}} \\ \mathbf{0} \end{pmatrix} \right\|$$

$$< \zeta_1 + \zeta_2||\mathbf{X}|| + \zeta_3||\mathbf{X}||^2 \tag{5.45}$$

where $\zeta_1, \zeta_2, \zeta_3 > 0$ are the system parameters. Using the same Lyapunov function (Eq. 5.13), $\dot{V}$ now becomes:

$$\dot{V} = -\mathbf{X}^T Q\mathbf{X} + \mathbf{X}^T B \leq ||\mathbf{X}||(\zeta_1 + (\zeta_2 - \lambda_{\min}(Q))||\mathbf{X}|| + \zeta_3||\mathbf{X}||^2) \tag{5.46}$$

Note that when:

$$\zeta_2 - \lambda_{\min}(Q) < 0 \tag{5.47}$$

$$(\zeta_2 - \lambda_{\min}(Q))^2 - 4\zeta_1\zeta_3 > 0 \tag{5.48}$$

by a proper choice of control gains, $\dot{V} < 0$ for $||\mathbf{X}|| \in (\gamma_1, \gamma_2)$, where $\gamma_1, \gamma_2) > 0$ are the roots of the polynomial $\zeta_1 + (\zeta_2 - \lambda_{\min}(Q))||\mathbf{X}|| + \zeta_3||\mathbf{X}||^2$ (Figure 5.6). Thus, by applying the same concept as in Lemma 3.5 in [57], the overall system can be shown to be uniformly ultimately bounded. Note that the purpose of this stability analysis is only to show that the proposed control law *can be stabilised* by a proper choice of control gains. However, in practice, the size of the uniform/uniform ultimate bound cannot be made arbitrarily small because of the upper limits on the control gains as discussed in the pre-

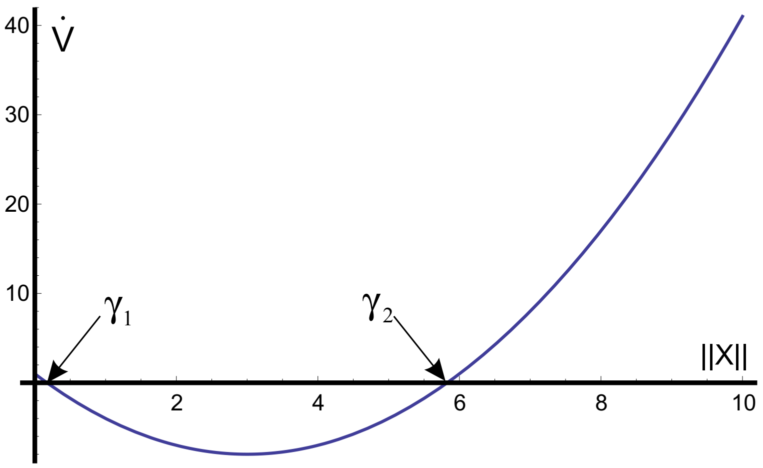

Figure 5.6: $||\dot{V}||$ vs $||\mathbf{X}||$.

vious Chapter. The next section will be devoted to show the experimental result(s) on a real robotic manipulator in order to validate the performance of the proposed controller.

5.3 Case-study: The PA10 Manipulator

Since the proposed controller has shifted the inverse dynamics concept from task space to joint space (the inner loop), the proposed controller is expected to outperform the conventional operational space control. To show the efficiency of the proposed scheme, the dual-loop controller is implemented on the Mitsubishi 7-DOF PA10 manipulator as described below.

5.3.1 Experiment testbed

In order to achieve real-time torque control capability, the original controller of the PA10 has been replaced by our custom controller which is depicted in Figure 3.6. The inner control loop has been implemented at 5 kHz, while the outer control loop is running at 1 kHz as depicted in Figure 5.1. The dynamic model of the manipulator was identified as shown in Chapter 3. Since the dynamic model has been used, the control gains in the below experiment are supposed to be isotropic and was chosen to have a critically damped behaviour. To show the usefulness of the proposed controller, two schemes are implemented:

- Conventional force-based OSC implementation (OSC): a straightforward implementation of the OSC as presented in Chapter 2. Note that due to the effects of model uncertainties, motion control gains of the OSC were tuned to reach the performance limits using a similar approach as described in section 6 of [40].

- The dual-loop OSC implementation or the modified OSC (mOSC): a two-layered hierarchical controller as illustrated in Figure 5.1 is used. Note that only a simple Euler method was used to integrate Eq. 5.7 because the integrator always used the current states $\mathbf{q}(t)$ as an initial condition, thus, the integration errors will not be accumulated.

It is noted that although there are other techniques to improve the control performance, only a simple model-based PID controller is used in this section. The reason is because the purpose of this section is *to evaluate the control performance* when the task space dynamics is shifted to joint space by the dual-loop control structure.

5.3.2 Task Space Free Motion Control

In this test, the end-effector of the manipulator was commanded to move 0.2 metre in the y-direction of the base frame in 2 seconds from the same initial configuration for both control schemes. Note that the end-effector pose has been obtained using the forward kinematic model (i.e. no external measurement) since only control errors are considered. Also, quintic polynomials are used for trajectory planning as described in Chapter 2. Null space control objective for both cases is to maintain the position of the first joint i.e. $q_{1d} = q_{1i} = 0$. The task space position tracking errors are shown in Figures 5.8-5.10. Here, blue represents the tracking error of the conventional implementation, red represents the tracking error of the proposed controller. As is seen, the response time for both control schemes are quite similar ($0 < t < 0.2s$) because the control gains for both schemes are at the same order of magnitude (Lemma 4.1). However, since the conventional OSC needs to cope with the joint space uncertainties through the robot kinematics, it performance is worse than the propose controller as discussed in Chapter 4. The poor performance of the conventional implementation can also be explained by analysing the friction effects at each joint. Experimental results (not shown here) indicated that joint frictions of the PA10 are significantly nonlinear due to the Stribeck effect, hysteresis effect and is found to be position and time dependent. Clearly, the simple friction model (viscous and Coulomb) is not able to fully describe these effects. These effects are magnified through the kinematic model into operational space; the performance is therefore degraded significantly for the conventional OSC case. On the other hand, the proposed dual-loop OSC has an inner-loop control to suppress these nonlinear effects efficiently at the joint level. Thus, disturbances from the inaccurate dynamics can be easily overcome by the joint space high-gain controllers. The result(s) from this experiment show how significant the joint space disturbances can degrade the task space control performance in practice. Also notice that the tracking error(s) in the y and z direction all have tendency to go up at the beginning ($t < 0.1s$). This phenomena is the result of the interaction between the integral action and joint stiction. As can also be seen in Figures 5.8-5.10, tracking errors in y and z direction are significantly larger than the tracking error in the x direction. One possible explanation is because of the desired task and the initial configuration of the robot (Figure 5.7). In fact, for the given task, joint 1, 3 and 5 (joints contribute motion in x direction) do not need to have any motion. As a result, the uncertainties at those joints were not excited. The same explanation also can be made for the difference between the tracking error in the y and z direction. In other words, the uncertainties at joint 6 create more effects on the z direction than those on the y direction because of the initial configuration.

5.3.3 Task Space Motion Control: Low-speed vs High-speed

The performance of the proposed control scheme is also evaluated for the high-speed and low-speed tasks. In this case, the end-effector is commanded to move in a star-like trajectory as illustrated in Figure 5.11 (this trajectory has been used to measure the performance of different operational space controllers in [40]).

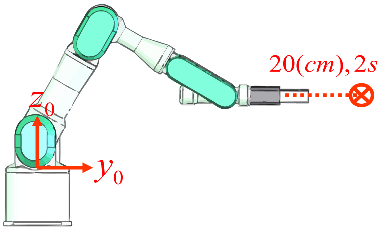

Figure 5.7: Free-motion task on the Misubishi PA10. The manipulator is initiated at the inverse configuration as the above figure.

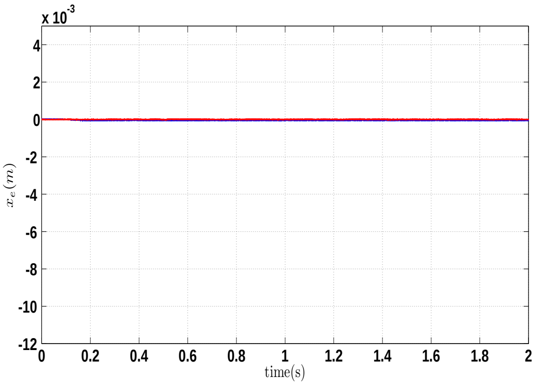

Figure 5.8: Task space tracking error in the x direction (blue: OSC, red: mOSC).

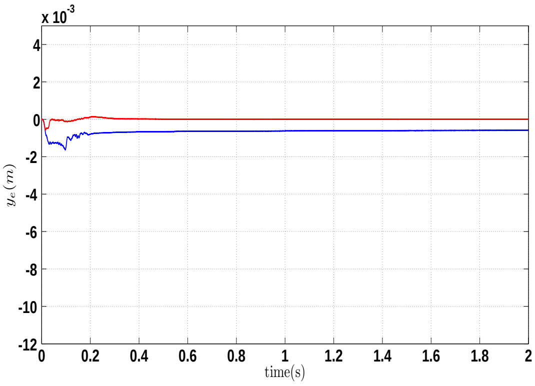

Figure 5.9: Task space tracking error in the y direction (blue: OSC, red: mOSC).

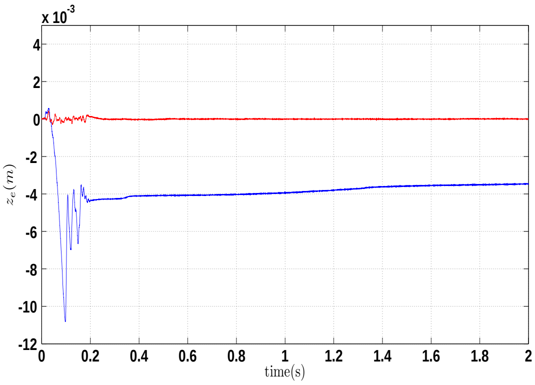

Figure 5.10: Task space tracking error in the z direction (blue: OSC, red: mOSC).

The length of each segment is 0.1m. Quintic polynomials are used for trajectory planning for each segment as described in Chapter 2. The end-effector of the robot is initiated at the centre of the star, the desired task is to move the end-effector from the initial configuration to one tip of the star, then move back to the initial configuration (both using quintic polynomial for trajectory planning). For the high-speed task, each segment is commanded to move in 1 second. For low-speed case, this value becomes 2 seconds. Position tracking errors are shown in Figures 5.12-5.13 (note that the control gains for both cases are the same). Since the dual-loop OSC makes use of the robot dynamic model at two separated level, outer-loop: the dynamics is used to generate the joint space command, inner-loop: the robot dynamics is used to compensate for the joint space nonlinear dynamics, the purpose of this experiment is to verify the effectiveness of the dynamic model in compensating for the nonlinear dynamics, especially the effects of link inertia. Theoretically, the control performance should be identical for any given

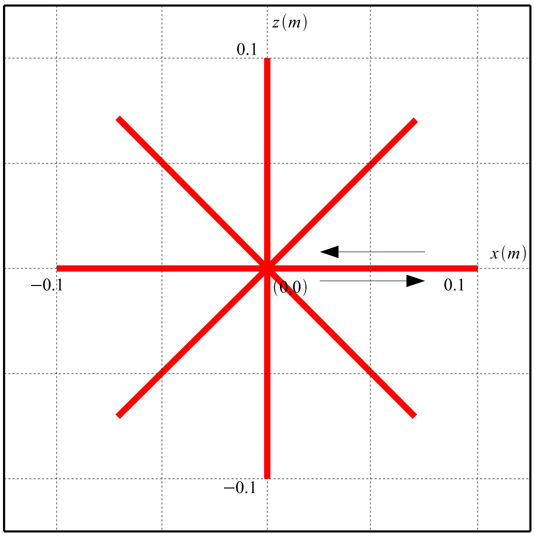

Figure 5.11: Star-shape trajectory.

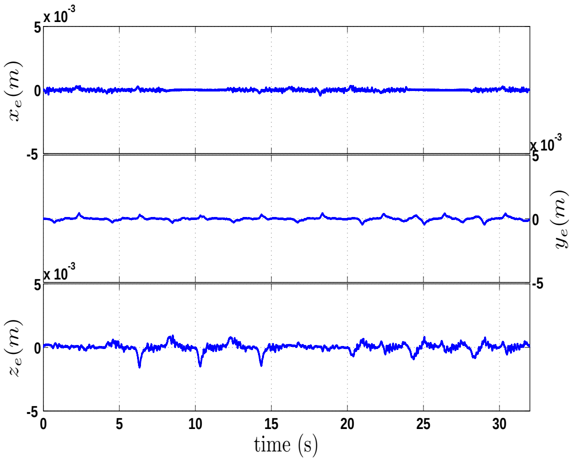

Figure 5.12: Tracking errors (low-speed). From top to bottom: tracking error in x, y and z direction.

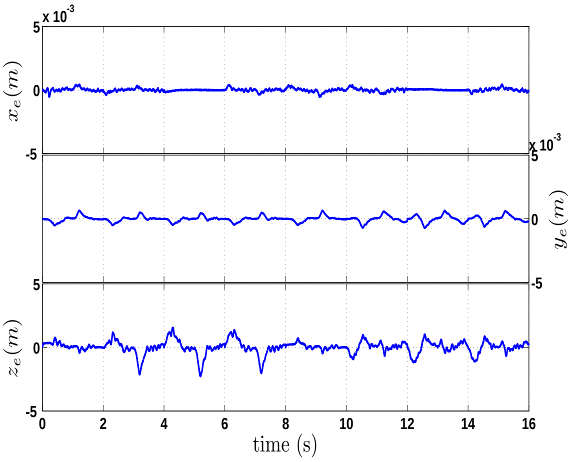

Figure 5.13: Tracking errors (high-speed). From top to bottom: tracking error in x, y and z direction.

task since the robot nonlinear dynamics has been cancelled out by the inverse dynamic control technique. However, in practice, differences in terms of the control performance for different task are expectable due to the model mismatch. As can be seen from Figures 5.12-5.13, although the tracking errors at low speed and at a sufficiently high-speed 0.1m/s are not exactly similar; the tracking errors in the high-speed case (Figure 5.13) are still around 0.1mm which can validate the effectiveness of the dynamic model compensation at the inner-loop.

5.3.4 Motion and Force Control

The motion and force control capability of the proposed control scheme was evaluated using two experiments as follows.

Force regulation

As mentioned before, regulating the contact force is a challenge for the conventional implementation of the OSC in the presence of model uncertainties. In fact, because of the unmodelled dynamics, it is impossible to maintain 1N contact force at the end-effector. However, as shown in Figure 5.14, the proposed controller is able to maintain a light (1N) as well as a slightly heavy (10N) contact force without any retuning of force gains. Note that in this experiment, only y-direction (base frame) is force controlled. Other axes (x, z and orientation) are motion controlled. Note that the poor control performance of the conventional OSC (Figure 5.16) can be explained as the interaction between the joint stiction and the integral of the force controller [48]. Unfortunately, the force controller usually needs the integral action to achieve zero steady state error because stiff environment limits the proportional gain (thus create large steady state error) in practice. On the hand, the dual-loop control structure already has the integral action at the joint level (the inner loop), thus, the outer loop, which is where the force controller takes place, can be considered to be free of all the above joint space disturbance. As a result, the dual-loop OSC structure can offer a better force control performance (Figures 5.15 and 5.17). Note that, in all the above experiments, task space damping (i.e. $-k\dot{\mathbf{x}}$) in the force control direction has been used to stabilise the system during the impact period [12].

Motion and force control

In order to have a better benchmark of the force control capability, the proposed control scheme has been compared to the commercial force control solution from PushCorp. Because PushCorp's compliance module only has the capability of regulating the force in one axis, the compliance module (AFD1100) was attached to the end-effector of the ABB robot (IRB4400) as depicted in Figure 5.20. The ABB robot is then commanded to

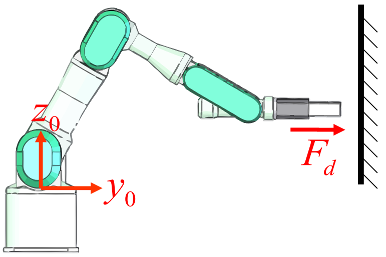

Figure 5.14: One axis force regulation. The robot has been initiated at the configuration as the above figure. The desired contact force is $1N$ in the y axis of the base frame. the workpiece is made from steel.

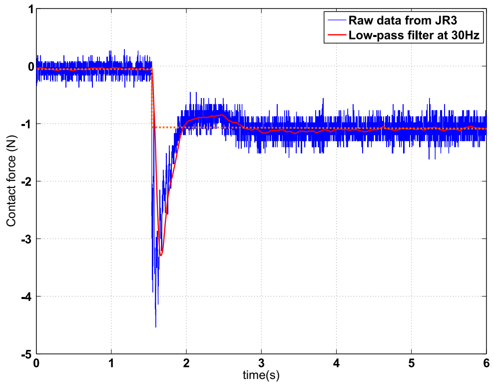

Figure 5.15: 1N contact force (mOSC).

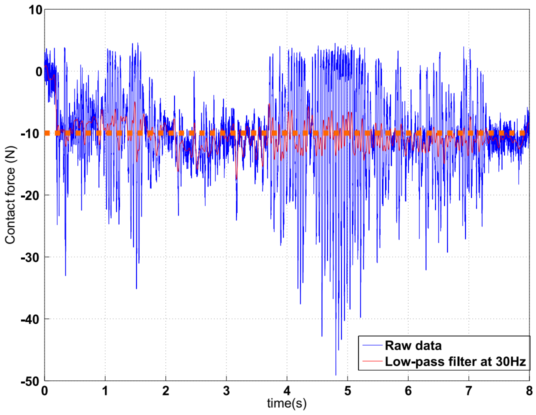

Figure 5.16: 10N contact force (OSC).

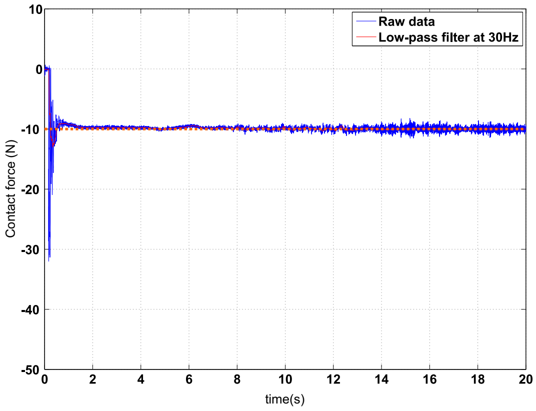

Figure 5.17: 10N contact force (mOSC).

move in x-direction while the PushCorp compliance module regulated the contact force in the y-direction as illustrated in Figure 5.18.

In this experiment, the task is to move the end-effector from the starting point to the end point while maintaining the contact force to be 20N. In order to have a fair comparison, an *independent JR3 sensor* has been installed at the bottom of the workpiece (Figure 5.19). The measured forces from sensor for the proposed force control using PA10 and the PushCorp's compliance module are presented in Figure 5.21. As the result shows, the force tracking error of the PushCorp's module and the PA10 force control are are similar in terms of the magnitude. However, the PA10 controller has to control a 7-DOF system while the PushCorp device is only concerned with the force control of a single axis. This observation implied that the proposed controller is able to perform as good as the 1-DOF system while maintaining the flexibility of a 7-DOF system.

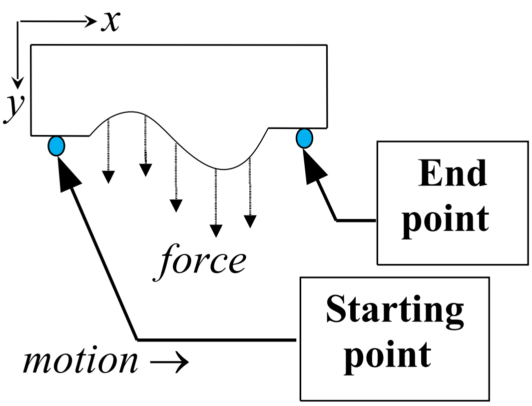

Figure 5.18: Hybrid Motion/Force Task.

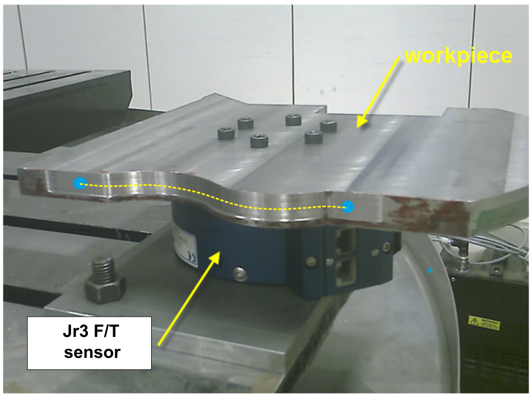

Figure 5.19: The workpiece.

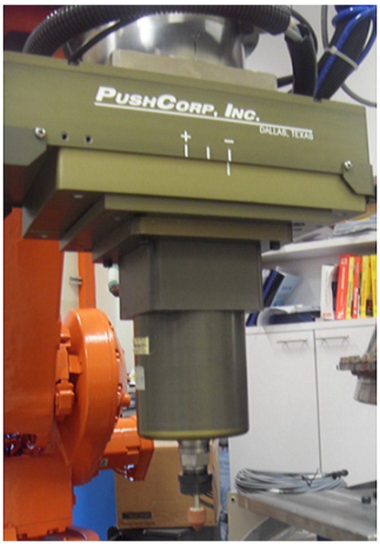

Figure 5.20: PushCorp on the ABB.

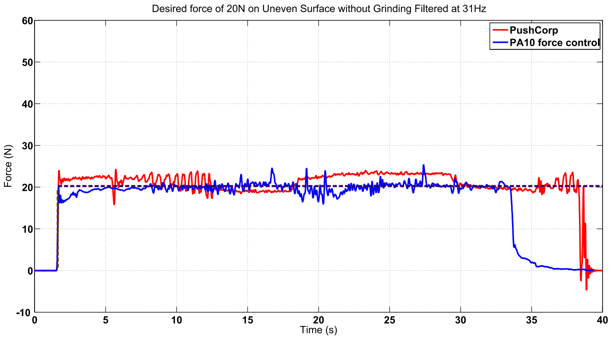

Figure 5.21: Force responses from the PushCorp + ABB and the dual-loop OSC. Note that robot end-effector has been initiated sufficiently near to the workpiece to reduce the impact force.

5.4 Conclusion

Based on the discussion in Chapter 4, a dual-loop operational space control structure has been introduced in this chapter. Since the use of the robot model has been shifted from task space to join space, the effects of model uncertainties on the control performance are expected to be minimised. To maintain the advantages of the force-based OSC, the identified robot model is used again at the outer loop to generate the joint space commands. The proposed control structure was shown to be uniformly ultimately bounded by Lyapunov's direct method in the continuous domain. The usefulness of the structure has also been verified on actual industrial robot, the Mitsubishi PA10 manipulator, using the QNX real-time operating system. Experimental results indicated a significant improvement in terms of the control tracking errors (both motion and force) in comparison to the conventional one.

Chapter 6

Industrial Application: Grinding Task

Although the performance of the proposed dual-loop operational space control (mOSC) structure in the previous chapter has been verified through intensive experiments, they are yet to be verified in a practice setting. Thus, it is necessary to analyse the usefulness of the mOSC from a practical stand point. In this chapter, a real industrial application is selected to evaluate the performance of the proposed control framework. We will investigate the performance of the mOSC through the grinding process which is one of the most common tasks in manufacturing industry. Before going into detailed description of the application, let us first justify the reason why we need compliant motion control capability for such grinding task.

6.1 Why Force Control for Grinding Task

Edge profiling is a common task in manufacturing industry since most machining processes produce unwanted features such as sharp edges or burrs. These undesirable features must be removed for part fitting and/or for safety reasons.

It is worth pointing out that although computer numerical controlled (CNC) (Figure 6.1) machines can be used for edge profiling tasks, these processes are still commonly carried out by human using handheld tools. The reason that CNC machines are not preferable is because they are costly and not suitable for large workpieces (such as aerospace/marine components). Although human equipped with suitable tools can be

Table 6.1: Comparison between CNC machines and robot systems.

CNC Machines	**Robotised Finishing Systems**
Higher rigidity	Lower rigidity
Higher accuracy $(< 10\mu m)$	Lower accuracy $(\sim 100\mu m)$
CAD/CAM offline programming	Intuitive teaching and programming
Position control	Position/Force/Adaptive process control
Limited workspace and low dexterity	Large workspace and high dexterity
High cost	Low cost
Suitable for high-precision $(< 10\mu m)$ machining of smaller parts with simple geometrical features	Suitable for precision surface finishing $(100 \sim 200\mu m)$ of larger parts with complicated geometrical features
Expensive $(> 600K)$	Affordable $(30K \sim 200K)$

more efficient than CNC machines for large parts, cost (higher skills, higher cost) and repeatability are some of the major disadvantages when high production rate is needed. On the other hand, since robotic systems (Figure 6.2) can provide a rather high workspace with great flexibility and repeatability at an affordable price, there is a recent trend of using robots to automate the above process. Table 6.1 summarises the advantages and disadvantages of CNC machines versus robotised systems for edge profiling tasks. As is seen from Table 6.1, although the accuracy provided by robotic systems is not as high as CNC machines ($< 10\mu m$ vs $100 \sim 200\mu m$), there are still substantial opportunities for using robotic systems to carry out cost-effective edge profiling process. This is due to the fact that, edge finishing for many large components does not require very high geometrical accuracy.

Figure 6.1: 5-axis CNC machine (*www.makino.de*).

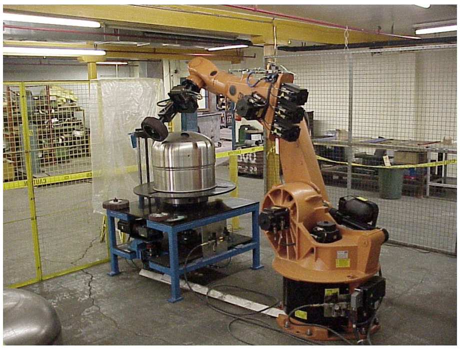

Figure 6.2: Robotised finishing system.

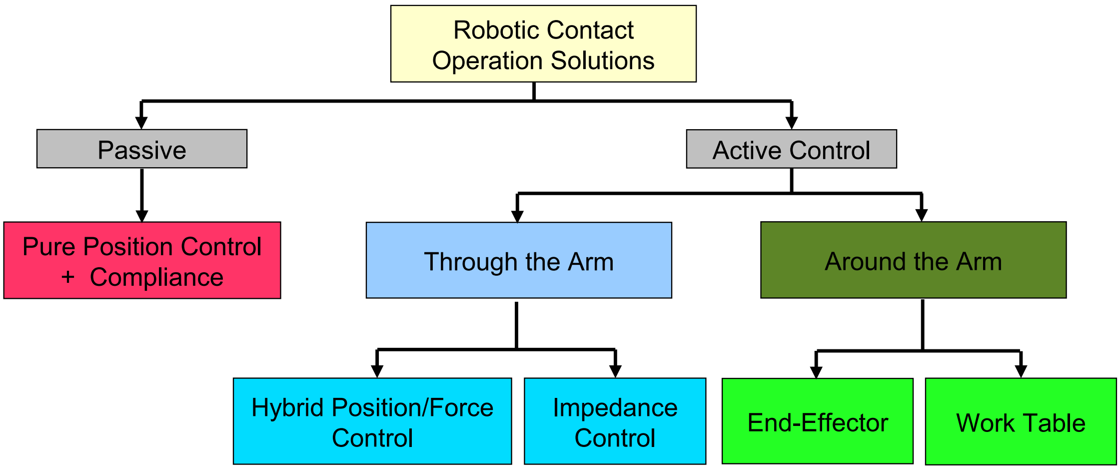

Figure 6.3: Contact force control approaches. Note that the terminology and literature in this chapter may not be consistent to the one from Chapter 1 since it has been done from the industry point of view [1].

Ideally, if the relationship between a robot and a workpiece is accurately known, one can use the robot as a positioning system with tools at the end-effector to machine the desired surfaces as in the case where CNC machines are used. However, in a typical robotic workcell, the task space accuracy of the robot and workpiece can be fairly poor. This will result in poor finishing quality, if the material removal approach is merely based on positional and geometrical accuracy of the workcell. In addition, since the robotic system serves as an independent positioning system, the dynamics of the machining process (which is the results from the interaction between the robot and workpiece) is totally ignored. Thus, the cutting tool can be rapidly worn and eventually damages the workpiece. One possible solution for this problem is to provide the compliant motion control capability for the robot as discussed in Chapter 1. By doing this, the dynamics of the machining process can now be adjusted through controlling the compliant behaviour of the robot.

It is worth pointing out that there are at least two approaches to provide compliance at the end-effector of industrial robots [90]: the passive compliant approach and the active compliant approach as depicted in Figure 6.3. In the passive compliant approach, the applied force is passively achieved through passive compliant tools (Figure 6.4-6.5) that are attached to the robot end-effector, while the robot operates in positioning mode.

Since passive compliant tools cannot directly control the applied force, the applications of these tools are limited to those require loose tolerance material removal ($> 500\mu m$). On the other hand, in the active compliant approach, the applied force is actively controlled either through the robot (by controlling the torque at each robot joint) or ”around the robot” [90] (by attaching another module at the robot end-effector). Note that when the applied force is achieved around the arm, the robot is still operated as a positioning system. In this case the compliant behaviour can be produced by the add-on module either at the robot end-effector or at the workpiece. Figure 6.6 shows a commercially available add-on module, the compliant module AFD1100 from PushCorp. This compliant tool can achieve active force control by adopting the ”around-the-arm” approach. Note that most available add-on compliant modules only provide force control for one axis. In addition, in the ”around-the-arm” approach, the compliant module operates independently to the robotic system, thus, motion planning for the robot end-effector needs to take into consideration the reaction force between the compliant tool and the workpiece in order to have a good force control performance (since the reaction force is treated as an unknown disturbance to the robot motion controller).

On the other hand, although force control using the ”through the arm” approach is quite established in the research community using laboratory robots, the application of these force control schemes on industrial robots is still limited [91]. So far, only ABB Corp (*www.abb.com*) published more convincing results [1,92] although their force control approach requires significant effort of programming (i.e teaching for the entire edge profile is required for good force control performance). Figures 6.8-6.9 show the force response of the IRC5 ABB force control for the same experiment as described in the previous chapter in comparison to the responses of the PushCorp compliant module (AFD1100) and the proposed mOSC using the Mitsubishi PA10 manipulator. As can be seen, without a intensive teaching effort, the response from the ABB’s force control performs the worst. Note that in this example, only starting-point and end-point are visible for all the three force control systems. If sufficient teaching points (points along the surface) are engaged, ABB’s force control performance can be improved to a satisfac-

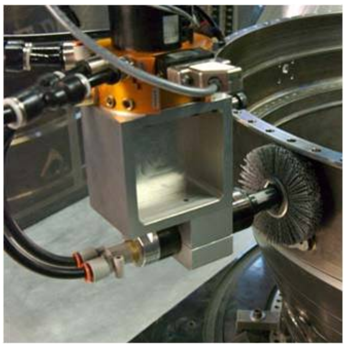

Figure 6.4: Passive compliant tool.

Figure 6.5: ATI's deburring tools (*www.ati-ia.com*).

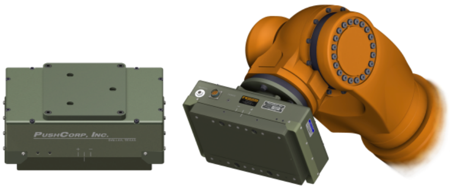

Figure 6.6: Around the arm approach: Pushcorp's active compliant tool (AFD1100) (*www .pushcorp.com*).

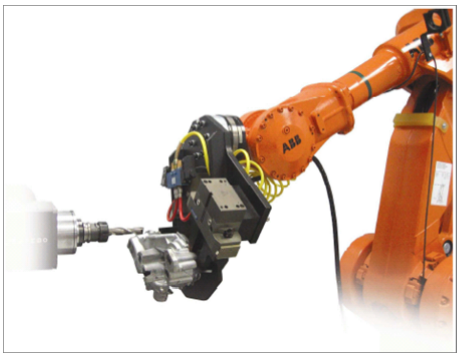

Figure 6.7: Through the arm approach: ABB's force control (*www.abb.com*).

tory level for the aforementioned grinding task. Thus, it is interesting to find out what is the performance of the dual-loop operational space control in practice for the same application (i.e. grinding task using conical tool) when only starting-point and end-point are known.

6.2 Grinding Application

From the above discussions, it is clear that the mOSC belongs to the active force control approach. Since this approach makes use of the hybrid position/force control framework, the desired compliant task can be achieved by decomposing it to motion and force control tasks. For instance, let us consider a task of using a grinding wheel to create a chamfer on a sharp edge as depicted in Figures 6.10. Obviously, this task can be decomposed into:

- Controlling the robot to move along the desired cutting edge and

- Controling the applied force in the direction that is orthogonal to the motion control directions.

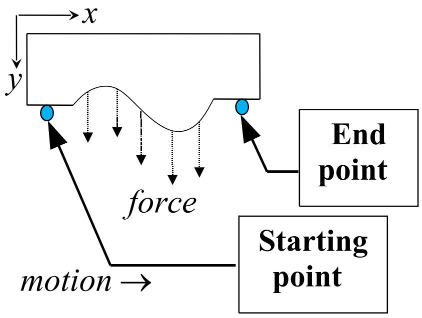

Figure 6.8: Hybrid Motion/Force Task.

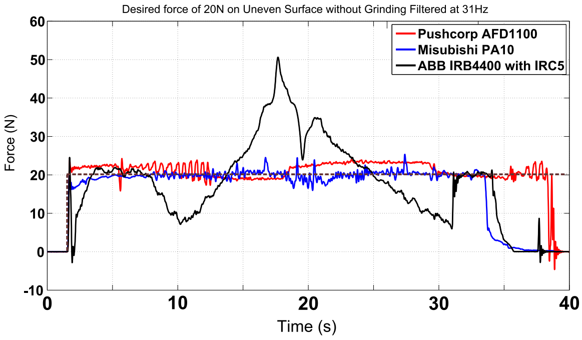

Figure 6.9: Force responses from the PushCorp+ABB, the mOSC and ABB's force control.

A visualisation of the above description is shown in Figure 6.11. As can be seen from Figure 6.11, the quality of the chamfered surface will mainly depend on the normal contact force during the cutting process. Two possible robot-workpiece setups to realise this grinding task are shown in Figure 6.12:

- The robot carries the workpiece, while the grinding spindle is stationary. This setup is only applicable for a workpiece that is small and light enough for the robot to carry. Since the workpiece is mounted after the force-sensor, inertial force caused by the workpiece inertia will be coupled with the contact force during the sensing process. As a result of this coupling, separating the contact force from the inertial force is one of the major issues of this approach. In addition to the aforementioned issue, the tool centre point (where the workpiece and tool contact) can also change during the operation for this setup, thus further complicating the force control scheme.

- The robot carries the grinding tool and the workpiece is stationary. Since the grinding tool has its own dynamics causing by the spinning wheel (even no contact and motion are involved), the force reading from the force/torque sensor can be contaminated. Also note that the previous mentioned coupling problem between the inertial force and contact force still exist in this setup if the tool inertia is large and therefore cannot be ignored.

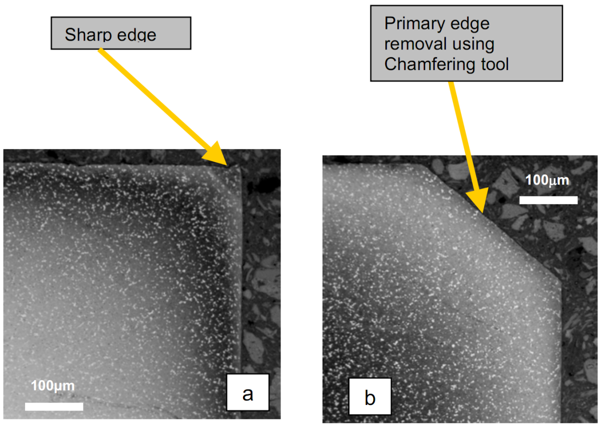

Figure 6.10: Sharp edge chamfering.

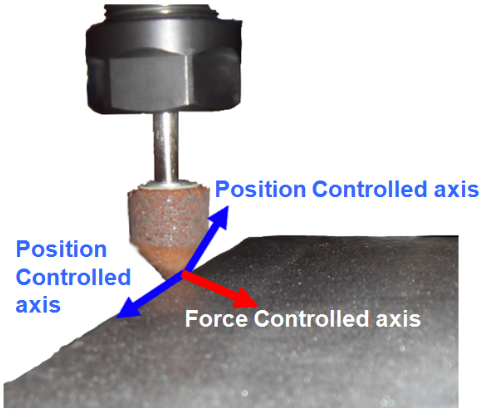

Figure 6.11: Grinding process.

For experiments in this chapter, the second setup i.e. the workpiece is mounted on the ground, while the pneumatic grinding spindle is attached to the PA10 manipulator is adopted. Since the purpose of this experiment is to evaluate the force control performance of the mOSC on edge profiling tasks with targeted to large aerospace/marine components, process study has been conducted on actual components to find out the suitable process parameters. From the process study, the required contact force can be as small as 2N [93]. As mentioned in Chapter 5, maintaining this light contact force is really a challenge for the conventional operational space formulation under the presence of model uncertainties. However, as also shown in Chapter 5, the proposed dual-loop operational space structure is able to maintain the contact force not only as low as 1N but also as 10N without re-tuning the force control gains, thus, this control framework will be adopted in this experiment. The experiment setup for the grinding task is described as follows.

6.2.1 Experiment Setup

- **Hardware setup**: the dimensions and model number of the grinding wheel are shown in Figure 6.13. The workpiece is made from steel with the dimension of $0.15\text{m} \times 0.15\text{m} \times 0.01\text{m}$. The initial position of the grinding wheel has been

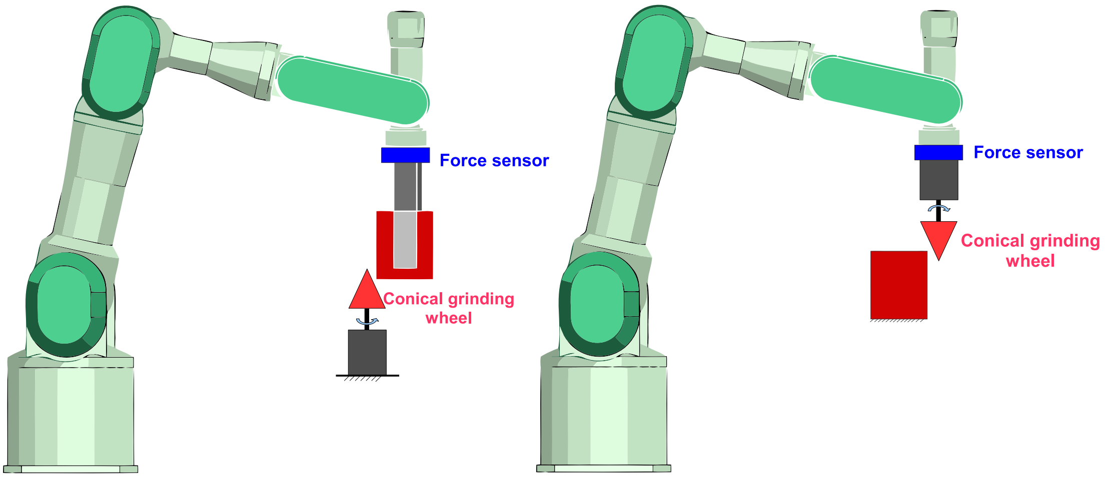

Figure 6.12: Experiment setups: robot carries the workpiece (left) and robot carries the grinding tool (right).

configured to be 0.5cm away from the workpiece. The pneumatic grinding spindle has been operated at a fix air pressure, 4bar, in this experiment.

In this experiment, the desired contact force for the grinding process is set to 3N. The same impact control strategy (i.e. adding damping in the force control direction) as described Chapter 5 is used to stabilise the system during the transition from free motion to constrained motion. To further reduce the effect of impact force on the surface to be ground, step-response (1N) at the beginning is used. The desired contact force is then gradually increased to 3N in 2s as depicted in Figure 6.14.

- **Software setup**:

  Control gains: the control gains of the force controller were first tuned without the dynamic effects of the grinding wheel (i.e. the wheel is turned off). This set of control gains is then used in the grinding experiment without any modification.

6.2.2 Practical Issues

Force sensing: since the dual-loop operational space control requires direct force sensing to close the force control loop, it is necessary to describe how force information is obtained.

Note that the problem of observing the contact force in the presence of inertial force and other dynamics is usually referred to as the sensor fusion problem [94–96]. The reason of why fusing data are needed comes from the fact that acceleration information, which can typically be obtained from the accelerometer, is required in estimating the contact force. In this experiment, since the inertia of the tool is negligible, effect of the tool inertia on the force measurement is ignored (i.e. only static compensation was incorporated, inertial effect on the force reading is treated as an unknown disturbance).

To the best of our knowledge, most research on the problem of estimating contact force for contact operations only considers the inertial effect of the heavy tool which is mounted after the force sensor. The dynamics of the tool when it is operating is usually ignored due to its complexity [96]. In this experiment, to minimise the effects of the dynamics of the grinding tool on the control performance, a low-pass filter has been used to limit the high frequency components (caused by the dynamics of the turning wheel) to enter the closed-loop and excite other unmodelled dynamics. The cut-off frequency of the low-pass filter has been chosen to be 5Hz by trial and error. Also note that the use of low-pass filter in force control has also been proven to enlarge the stable region of the entire system [97, 98]. Although the use of the low-pass filter on the force reading has significantly simplified the tuning effort of the force control gains, one should be aware that a very low cut-off frequency introduces a significant lag into the system. It is worth pointing out that to avoid instability when contacting with high stiffness environment, small proportional gain is expected in force control. Since small proportional gain usually leads to high steady-state error, integral control is commonly added to eliminate this large steady-state error. However, this add-in integral action can have adverse effects on the stability of the system, especially for system with high lag as the above discussion.

6.2.3 Experimental Results

The experimental results of the above grinding task for different initial conditions are shown in Figure 6.15 - 6.16.

From the Figures, it seems that the force control performance is very sensitive to the process dynamics. For example, by shifting the contact point between the conical

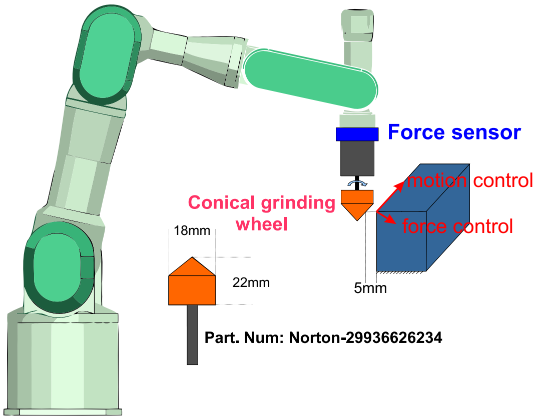

Figure 6.13: Sharp edge deburring task.

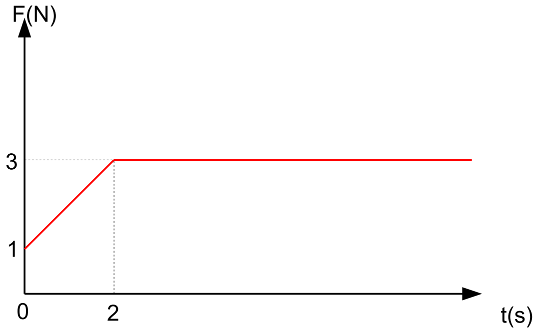

Figure 6.14: Desired contact force profile.

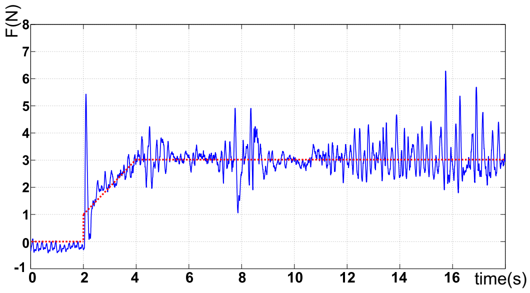

Figure 6.15: Force response for the case the contact point is about 10mm from the tip.

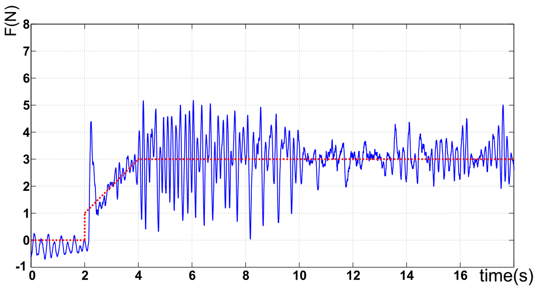

Figure 6.16: Force response for the case the contact point is about 12mm from the tip.

wheel and the workpiece 2mm upward, the force responses can be much different (for the same set of control gains). Note that it is possible to re-tune the force control gains for case where the contact point is 12mm away from the tip to have a reasonably good performance, the purpose of Figures 6.15 - 6.16 is to show how sensitive the force control performance to the position of the contact point is. Although low-pass filter has been used to limit the effect of high frequency disturbance, it is clear that this commonly used approach is not efficient in practice where precise geometrical constraints are hard to guarantee. Also note that lower the cut-off frequency of the low-pass filter should not always be possible in practice since the delay producing by a very low cut-off frequency can cause the force response to enter a limit cycle. To this end, it is necessary to develop a robust algorithm to separate the fast contact force response from other dynamics. It is worth stressing that most of the work on the problem of separating the contact force from other dynamics [94, 95] usually made use of the ”slowly time-varying environmental force” assumption (Section II of [94] for instance) to simplify the gain selection process. To some extent, this assumption is equivalent to have low-pass filter on the force reading as in our case, thus, the above approaches may not be suitable for the grinding application (since slowly time-varying assumption usually leads to the same disadvantage as discussed in the low-pass filter approach). The design of such algorithms should be addressed as an important topic in future research.

Chapter 7

Conclusions

The overall objective of this thesis was to analyse the theoretically more advanced force-based operational space control framework from the practical point of view. In view of this, a detailed analysis on the above control framework in the presence of model uncertainties and digitisation effects was conducted. Based on the analysis, an improved version of the above control framework was proposed. Comprehensive studies from both theoretical and empirical point of view were carried out to evaluate the performance of the proposed controller. In addition, an improved identification process, which was used to obtain the dynamic model of robots, was also suggested.

Firstly, the control performance of the force-based operational space controller under the presence of model uncertainties and digitisation effects was analysed. The focus of this work was only on the force-based operational space control because this control framework can be considered as the more advanced control framework for redundant robots. Experimental results in Chapter 4 indicated that the accuracy of the robot dynamic model plays an important role on the operational space control performance. Since one does not have access to the exact dynamic model, the mismatch between the estimated and real model can significantly degrade the operational space control performance. In seeking for an explicit explanation of why the force based OSC cannot perform well when faced with modelling errors, [99] showed that the upper limit of control gains for both joint-space inverse dynamics control and task-space inverse dynamics control are the same under the discretisation (sampling) effects of digital controllers.

This finding is of crucial importance because it has provided a conclusive evidence for why the theoretically more advanced force-based operational space controllers exhibit significant dependence on the accuracy of the dynamic model. By making use of the singular perturbation theory, conditions of when model uncertainties affect the force based operational space controller the most were also presented.

Based on the above analysis, the computed torque techniques need to be accomplished in joint space to avoid magnifying the modelling errors through the robot kinematics. As a result, the second aim of this study was to seek for a controller that maintains the advantages of the force-based operational space control, while still minimising the effects of model uncertainties and digitisation effects on the control performance. The analysis in Chapter 4 suggested one possible solution for the above problem, the dual-loop operational space control. The major contribution of this dual-loop control structure is that the inverse dynamic concept has been shifted from the operational space into joint space in order to minimise the effects of model uncertainties. Comprehensive experimental studies were conducted to evaluate the effectiveness of the proposed control as presented in Chapter 5. The experimental results showed that the dual-loop control structure with an inner inverse-dynamics loop provides a considerably better control performance in comparison to the conventional force-based operational space control. As also mentioned in Chapter 5, regulating a light contact force is really a challenge for the conventional operational space control in the presence of model uncertainties. In fact, due to the un-modelled dynamics, it is impossible to maintain 1N contact force at the end-effector. However, the proposed controller is able to maintain a light (1N) as well as a slightly heavier (10N) contact force without any re-tuning of force gains. The reason for this significant improvement is that the inner joint-space controller is less sensitive to the model uncertainties than the one used in task-space. A similar result for the hybrid position/force control was also obtained through experiments. In the experiment, the proposed controller has been benchmarked with the commercial one-axis compliance force from PushCorps. The experimental result showed that the force tracking error of the PushCorp module and the proposed force control are similar. However, the proposed controller has to control a 7-DOF system, while the PushCorp device is only concerned with controlling the force of a single axis. This observation implied that the proposed controller is able to perform as good as the 1-DOF system while maintaining the flexibility of a 7-DOF system.

Although the proposed controller has been shown to outperform the conventional force-based operational space control, it is still necessary to investigate the stability of the proposed controller. Based on the Lyapunov second method, the work in Chapter 5 indicated that the closed-loop equation of the overall system is uniformly ultimately bounded. To show the usefulness of the dual-loop operational space control in practice, the proposed controller has also been evaluated through a practical grinding application, which is a common task in manufacturing industry. Experimental results in Chapter 6 have revealed that observing the fast contact force in the presence of inertial force and other high frequency dynamics is of crucial importance for improving the force control performance. Thus, this problem should be properly addressed in future research.

In this study, it is assumed that the estimated robot model can provide a reasonably good control performance. This assumption can be achieved by a proper identification process as the work presented in Chapter 3. In this Chapter, an improved procedure for identifying the dynamic model of robot manipulators, which also takes into account the effects of dynamic friction, has been proposed. Since the linear property of the robot dynamic model is valid only in the linear region of dynamic friction, a scheme was proposed to identify the boundary velocity that separates the dynamic friction into nonlinear and linear (steady state) regions. By making use of the above boundary velocities, explicit condition of when the robot model can be represented in the linear form has been derived. Realistically, uncertainties exist in both measured motion data (i.e., joint position, velocity and acceleration) and joint torques, estimated parameters using the conventional least squares estimator will deviated from the actual value because no constraints are imposed on the least-squares technique. As a result, it is possible for the least-squares estimator to produce results which are physically impossible. To account for this issue, a constrained optimisation for obtaining the estimated parameters was proposed. Important constraints such as the steady state of dynamic friction and positive definitiveness of the inertia matrix were also conveniently incorporated into this optimisation, thus, the physical feasibility of the identified parameters can be guaranteed. Experimental results on the first four joints of the Mitsubishi PA10 manipulator were provided in order to validate the effectiveness of the proposed identification procedures. Although the proposed identification method has achieved better results, there is still no guarantee for the convergence of the identified parameters as discussed in Chapter 3. This observation implies that further research is needed to improve the quality of the estimated robot model, in particular model identification for advanced control purpose.

Bibliography

[1] C. W. Lim and P. Y. Tao, "Enhancing robotic applications in the industry through force control," Singapore Institute of Manufacturing Technology, Tech. Rep., 2009.

[2] R. Bischoff, "Closer academia industry collaboration," EURON Annual Meeting, Leuven, Tech. Rep., 2009.

[3] EUROP, "Robotic vision to 2020 and beyond," *The Strategic Research Agenda for Robotics in Europe*, 2009.

[4] B. Siciliano and L. Villani, *Robot Force Control*, K. A. Publishers, Ed. The Kluwer International Series In Engineering and Computer Science, 1999.

[5] S. Chiaverini, B. Siciliano, and L. Villani, "Survey of robot interaction control schemes with experimental comparison," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, pp. 273–285, 1999.

[6] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *Proceedings of the 19th IEEE Conference on Decision and Control Including the Symposium on Adaptive Processes*, 1981.

[7] N. Hogan, "Impedance control: An approach to manipulation, part 2 - implementation," *Journal of Dynamic systems, Measurements, and Control*, vol. 107, 1985.

[8] T. Yoshikawa, "Force control of robot manipulators," in *Proceedings 2000 IEEE International Conference on Robotics and Automation*, 2000, pp. 220–6.

[9] M. Raibert and J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 102–127, 1981.

[10] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-11, pp. 418–32, 1981.

[11] J. De Schutter and H. van Brussel, "Compliant robot motion. ii. a control approach based on external control loops," *International Journal of Robotics Research*, vol. 7, no. 4, pp. 18–33, 1988.

[12] J. Park, "Control strategies for robots in contact," Ph.D. dissertation, Stanford University Stanford, CA, US, 2006.

[13] O. Khatib, "A unified approach for motion and force control of robot manipulators: the operational space formulation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[14] L. Sentis, "Synthesis and control of whole-body behaviors in humanoid systems," Ph.D. dissertation, Stanford University Stanford, CA, US, 2007.

[15] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 4, pp. 505–518, 2005.

[16] N. D. Vuong and M. H. J. Ang, "Improved dynamic identification of robotic manipulators in the linear region of dynamic friction," in *The 9th International IFAC Symposium on Robot Control.*, 2009, pp. 270–276.

[17] ——, "Dynamic model identification for industrial manipulator subject to advanced model based control," in *4th HNICEM. Manila, Philippines.*, 2008.

[18] G. Antonelli, F. Caccavale, and P. Chiacchio, "A systematic procedure for the identification of dynamic parameters of robot manipulators," *Robotica*, vol. 17, pp. 427–35, 1999.

[19] B. Armstrong, O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the puma 560 arm," in *Proceedings 1986 IEEE International Conference on Robotics and Automation*, 1986, pp. 510–18.

[20] B. Armstrong, “On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics,” *International Journal of Robotics Research*, vol. 8, no. 6, pp. 28–48, 1989.

[21] M. Gautier, “Dynamic identification of robots with power model,” in *Proceedings 1997 IEEE International Conference on Robotics and Automation*, vol. 3, 1997.

[22] M. Gautier, A. Janot, and P. O. Vandanjon, “Didim: A new method for the dynamic identification of robots from only torque data,” ser. Proceedings - IEEE International Conference on Robotics and Automation, 2008, pp. 2122–2127.

[23] M. Grotjahn, M. Daemi, and B. Heimann, “Friction and rigid body identification of robot dynamics,” *International Journal of Solids and Structures 38*, vol. 38, pp. 1889–902, 2001.

[24] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter, and H. Van Brussel, “Optimal robot excitation and identification,” *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 5, pp. 730–740, 1997.

[25] J. Swevers, W. Verdonck, and J. De Schutter, “Dynamic model identification for industrial robots,” *Control Systems Magazine, IEEE*, vol. 27, no. 5, pp. 58–71, 2007.

[26] P. I. Corke and B. Armstrong-Helouvry, “A search for consensus among model parameters reported for the puma 560 robot,” in *Proceedings 1994 IEEE International Conference on Robotics and Automation*, vol. Vol. 2, 1994, pp. 1608–13.

[27] R. Kelly, V. Santibanez, and A. Loria, *Control of Robot Manipulators in Joint Space*, ser. Advanced textbooks in control and signal processing. Springer, 2005.

[28] Z. Qu, J. F. Dorsey, Z. Xinfan, and D. M. Dawson, “Robust control of robots by the computed torque law,” *Systems Control Letters*, vol. 16, no. 1, pp. 25–32, 1991.

[29] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, “A unifying framework for robot control with redundant dofs,” *Autonomous Robots*, vol. 24, no. 1, pp. 1–12, 2008.

[30] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, “Resolved-acceleration control of robot manipulators: A critical review with experiments,” *Robotica*, vol. 16(05), pp. 565–573, 1998.

[31] E. Freund and J. Pesara, “High-bandwidth force and impedance control for industrial robots,” *Robotica*, vol. 16, pp. 75–87, 1998.

[32] R. Kelly and J. Moreno, “Manipulator motion control in operational space using joint velocity inner loops,” *Automatica*, vol. 41, no. 8, pp. 1423–32, 2005.

[33] R. Garrido, E. Canul, and A. Soria, “Task space robot control using an inner pd loop,” in *Proceedings 2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1268–72.

[34] O. Khatib, “Inertial properties in robotic manipulation: An object-level framework,” *International Journal of Robotics Research*, vol. 14, pp. 19–36, 1995.

[35] H. Bruyninckx and O. Khatib, “Gauss’ principle and the dynamics of redundant and constrained manipulators,” in *Proceedings 2000 IEEE International Conference on Robotics and Automation*, vol. 3, 2000, pp. 2563–2568 vol.3.

[36] J. Hollerbach and S. Ki, “Redundancy resolution of manipulators through torque optimization,” *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 308–316, 1987.

[37] Y. Nakamura and H. Hanafusa, “Optimal redundancy control of robot manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.

[38] Y. Nakamura, *Advanced robotics: redundancy and optimization*, ser. Addison-Wesley series in electrical and computer engineering. Control engineering. Addison-Wesley Pub. Co., 1991.

[39] P. Hsu, J. Hauser, and S. Sastry, “Dynamic control of redundant manipulators,” *Journal of Robotic Systems*, vol. 62, 1989.

[40] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, “Operational space control: A theoretical and empirical comparison,” *International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[41] D. N. Oetomo, “Mobile manipulation: Singularity analysis and handling,” Ph.D. dissertation, National University of Singapore, 2002.

[42] B. Featherstone and O. Khatib, “Load independence of the dynamically consistent inverse of the jacobian matrix,” *International Journal of Robotics Research*, vol. 16, no. 2, pp. 168–70, 1997.

[43] J. Craig, *Introduction to robotics: mechanics and control*, 3rd, Ed. Prentice Hall, 2004.

[44] C. C. de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, “New model for control of systems with friction,” *IEEE Transactions on Automatic control*, vol. 40, no. 3, pp. 419–425, 1995.

[45] C. C. de Wit, “Control of systems with dynamic friction,” in *CCA’99 Workshop*, 1999.

[46] H. Olsson, K. J. Astrom, C. C. de Wit, M. Gafvert, and P. Lischinsky, “Friction models and friction compensation,” *European Journal of Control*, vol. 4, no. 3, pp. 176–95, 1998.

[47] C. C. de Wit, “Experimental results on adaptive friction compensation in robot manipulators: low velocities,” in *Experimental Robotics I*, 1990, pp. 196–214.

[48] W. T. Townsend and J. Salisbury, J. K., “The effect of coulomb friction and stiction on force control,” in *Proceedings 1987 IEEE International Conference on Robotics and Automation*, 1987, pp. 883–9.

[49] C. Abdallah, D. M. Dawson, P. Dorato, and M. Jamshidi, “Survey of robust control for rigid robots,” *IEEE Control Systems Magazine*, vol. 11, pp. 24–30, 1991.

[50] Z. Qu and J. Dorsey, “Robust tracking control of robots by a linear feedback law,” *IEEE Transactions on Automatic control*, vol. 36, no. 9, pp. 1081–1084, 1991.

[51] X. Wang and L. K. Chen, “Proving the uniform boundedness of some commonly used control schemes for robots,” in *IEEE International Conference on Robotics and Automation, 1989.*, ser. Proceedings. 1989 IEEE International Conference on Robotics and Automation (Cat. No.89CH2750-8), 1989, pp. 1491–6.

[52] V. D. Yurkevich, *Design of Nonlinear Control Systems with The Highest Derivative in Feedback*. World Scientific, 2004.

[53] J. J. E. Slotine and L. Weiping, “On the adaptive control of robot manipulators,” *International Journal of Robotics Research*, vol. 6, pp. 49–59, 1987.

[54] J. J. E. Slotine and W. Li, “Adaptive manipulator control: A case study,” *IEEE Transactions on Automatic control*, vol. 33, pp. 995–1003, 1988.

[55] P. Tomei, “Adaptive pd controller for robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 565–570, 1991.

[56] R. Ortega and M. W. Spong, “Adaptive motion control of rigid robots: a tutorial,” *Automatica*, vol. 25, pp. 877–88, 1989.

[57] Z. Qu and D. M. Dawson, *Robust Tracking Control of Robot Manipulators*. IEEE Press, 1996.

[58] C. C. Cheah, C. Liu, and J. J. E. Slotine, “Adaptive tracking control for robots with unknown kinematic and dynamic properties,” *International Journal of Robotics Research*, vol. 25, pp. 283–96, 2006.

[59] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2006.

[60] B. Siciliano and L. Sciavicco, *Robotics: Modelling, Planning and Control - 3rd*. Springer Verlag, 2008.

[61] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer-Verlag New York, Inc., 2007.

[62] J. Slotine, O. Khatib, and D. Ruth, “Robust control in operational space for goal-positioned manipulator tasks,” *International Journal of Robotics & Automation*, vol. 31, 1988.

[63] C.-Y. Kuo and S.-P. Wang, “Robust position control of robotic manipulator in cartesian coordinates,” *IEEE Transactions on Robotics and Automation*, vol. 75, 1991.

[64] K. Kaneko, K. Komoriya, K. Ohnishi, and K. Tanie, “Manipulator control based on a disturbance observer in the operational space,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994.

[65] B. Nemec and L. Zlajpah, “Null space velocity control with dynamically consistent pseudo-inverse,” *Robotica*, vol. 18, no. 05, pp. 513–518, 2000.

[66] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-based control of a robot manipulator*, ser. The MIT Press series in artificial intelligence. MIT Press, 1988.

[67] M. Daemi and B. Heimann, “Separation of friction and rigid body identification for an industrial robot,” *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES*, pp. 35–42, 1998.

[68] F. Benimeli, V. Mata, and F. Valero, “A comparison between direct and indirect dynamic parameter identification methods in industrial robots,” *Robotica*, vol. 24, pp. 579–90, 2006.

[69] M. Gautier and W. Khalil, “Direct calculation of minimum set of inertial parameters of serial robots,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 368–373, 1990.

[70] M. Gautier, “Numerical calculation of the base inertial parameters of robots,” *Journal of Robotic Systems*, vol. 8, no. 4, pp. 485–506, 1991.

[71] C. Presse and M. Gautier, “New criteria of exciting trajectories for robot identification,” in *Proceedings 1993 IEEE International Conference on Robotics and Automation*, 1993, pp. 907–12.

[72] M. M. Olsen, J. Swevers, and W. Verdonck, “Maximum likelihood identification of a dynamic robot model: implementation issues,” *International Journal of Robotics Research*, vol. 21, no. 2, pp. 89–96, 2002.

[73] A. Janot, P. Vandanjon, and M. Gautier, “Using robust regressions and residual analysis to verify the reliability of ls estimation: Application in robotics,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[74] K. Yoshida and W. Khalil, “Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters,” *International Journal of Robotics Research*, vol. 19, no. 5, pp. 498–510, 2000.

[75] K. Yoshida, K. Osuka, H. Mayeda, and T. Ono, “When is the set of base parameter values physically impossible?” in *Proc. of the 1994 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, 1994, pp. 335–342.

[76] V. Mata, F. Benimeli, N. Farhat, and A. Valera, “Dynamic parameter identification in industrial robots considering physical feasibility,” *Advanced Robotics*, vol. 19, no. 1, pp. 101–19, 2005.

[77] N. D. Vuong and M. H. J. Ang, “Dynamic model identification for industrial robots,” *Acta Polytechnica Hungarica*, vol. 6(5), pp. 51–68, 2009.

[78] L. Marton and B. Lantos, “Modeling, identification, and compensation of stick-slip friction,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 511–521, Feb. 2007.

[79] R. H. A. Hensen, “Controlled mechanical systems with friction,” Ph.D. dissertation, Technische Universiteit Eindhoven, 2002.

[80] P. I. Corke, “In situ measurement of robot motor electrical constants,” *Robotica*, vol. 14(4), pp. 433–436, 1996.

[81] S.-T. Wu, “Digital high-gain pd control of robot manipulators,” *Journal of Robotic Systems*, vol. 14, no. 5, pp. 375–87, 1997.

[82] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[83] T. Yabuta, “Nonlinear basic stability concept of the hybrid position/forcecontrol scheme for robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 663–670, 1992.

[84] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Addison-Wesley, 1998.

[85] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[86] N. D. Vuong, M. H. J. Ang, T. M. Lim, and S. Y. Lim, "Multi-rate operational space control of compliant motion in robotic manipulators," in *IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 3175–3180.

[87] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[88] M. Bebendorf, *Hierarchical matrices: A means to efficiently solve elliptic boundary value problems*. Springer Verlag, 2008.

[89] M. Vidyasagar, *Nonlinear systems analysis*, 2nd ed., ser. Classics in applied mathematics 42. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

[90] L. Godwin and I. PushCorp, "Programming with force control," RIA Grinding, Deburring and Finishing Workshop, St. Paul, Minnesota, Tech. Rep., 1996.

[91] H. Zhang, Z. Gan, T. Brogard, J. Wang, and M. Isaksson, "Learning skills - robotics technology in automotive powertrain assembly," ABB Review (www.abb.com), Tech. Rep., 2004.

[92] J. Wang, H. Zhang, and T. A. Fuhlbrigge, "Force control technologies for new robotic applications," in *IEEE International Conference on Technologies for Practical Robot Applications*, 2008.

[93] G. Yang, C. W. Lim, and P. L. Liu, "Robotised precision finishing," Singapore Institute of Manufacturing Technology, Tech. Rep., 2009.

[94] J. Garcia, A. Robertsson, J. Ortega, and R. Johansson, "Sensor fusion of force and acceleration for robot force control," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.*, 2004.

[95] ——, "Automatic calibration procedure for a robotic manipulator force observer," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005.

[96] ——, "Sensor fusion for compliant robot motion control," *IEEE Transactions on Robotics*, vol. 24(2), 2008.

[97] H. P. Qian and J. De Schutter, “The role of damping and low pass filtering in the stability of discrete time implemented robot force control,” in *Proceedings. 1992 IEEE International Conference on Robotics And Automation (Cat. No.92CH3140-1), 12-14 May 1992*, ser. Proceedings. 1992 IEEE International Conference on Robotics And Automation (Cat. No.92CH3140-1), 1992, pp. 1368–73.

[98] ——, “Stabilizing robot force control through low pass filtering with low cut off frequency,” in *Proceedings of 1992 31st IEEE Conference on Decision and Control, 16-18 Dec. 1992*, ser. Proceedings of the 31st IEEE Conference on Decision and Control (Cat. No.92CH3229-2), 1992, pp. 1893–6.

[99] N. D. Vuong, M. H. J. Ang, T. M. Lim, and S. Y. Lim, “An analysis of the operational space control of robots,” in *IEEE International Conference on Robotics and Automation, 2010.*, 2010, pp. 4163 –4168.

[100] C. Zielinski, “The mrroc++ system,” in *Workshop on Robot Motion and Control*, 1999.

[101] F. Zhang, *Matrix theory : Basic Results and Techniques*. Springer, 1999.

Appendix A

Real-time Control Framework

In this Section, we briefly introduce the software framework that has been used to implement all the work in this thesis. Note that the control framework is mainly based on the MRROC++ framework, which originates from Warsaw University of Technology. A quick fact about the MRROC++ framework is as follows [100]:

- History:
  - RORC: 80s
  - MRROC: 90s
  - MRROC++ for QNX 4.2: later 90s
  - MRROC++ for QNX 6.3.1: present
- Language:
  - Object oriented C++
  - Real-time performance: QNX
  - Communication among nodes in the network: QNET (real-time), 500 Hz
- Advantages:
  - High to low level task specification
  - Hierarchical structure
  - Error handling
  - Decentralised computation support: from ECP level onward
  - Multi-robot: support task coordinator
  - Stable

It is worth pointing out that the MRROC++ is a control structure rather than a library (i.e. most of the source of the MRROC++ can only be used as a reference because it is hardware dependence). The significant of the framework is that it divides the whole control framework into modules.

Basically, the control framework is combined from the following processes (Figure A.1):

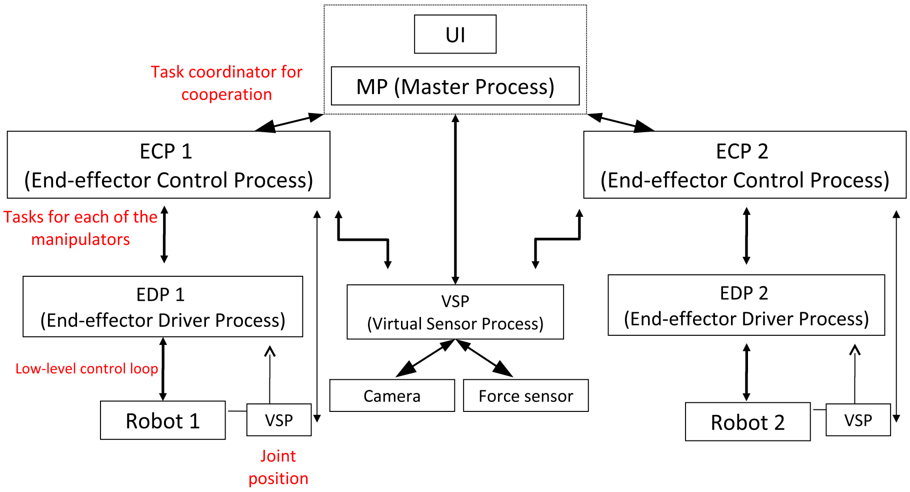

Figure A.1: MRROC++ Framework.

- End-effector Driver Process (EDP): this process responds for controlling the motor at each robot joint. This process is typically running at the node that has a direct connect to the robot. In our case, the EDP serves as the inner-loop level i.e. the velocity controller. This EDP is running at 5 kHz.

- End-effector Control Process (ECP): this process responds for controlling the elementary tasks. In our case, ECP is the outer-loop (1 kHz), which is the operational space controller.

- Master Process (MP): is used to coordinate the tasks if more than one robot is involved to complete the assigned task. In our case, MP is just a dummy process that calls the ECP and EDP when it is first initiated.

- Virtual Sensor Process (VSP): this process is mainly used for acquiring information from sensors. Different sensors will have different VSP. However, all VSPs use the same protocol to communicate to the ECP and EDP.

- User Interface Process (UI): this module is only used to display the process information such as the motion data (position, velocity and acceleration). UI is also contained a simple input command mechanism for user. The command syntax is divided in to three part: *command - 1st params - 2nd params - 3rd params*. For example, *moveto q 5 30* will command joint 5 to move to the position that have $q_5 = 30^0$. Please refer to the source code for all the available commands.

Initially, user should call the MP from the UI. Initialisation sequence has been incorporated into the MP and will be automatically run once MP is called. Typically, all the EDP, ECP and MP is running on the same node. However, this behaviour can be change by modifying the $.ini$ in the $/bin$ folder. Note that if processes are spcified to run at different node, the maximum communication rate among the processes is $500Hz$. Thus, it is recommended to run all the above process (MP, ECP, EDP) on the same node if enough computation power is available. Also note that the original MRROC++ framwork (downloadable from (*www.ia.pw.edu.pl/ zielinsk/*)) supports the virtual mode i.e. the physical hardware (robots, sensors) can be replaced by virtual ones (such as one in Player-Stage (*playerstage.sourceforge.net*)), however, this feature is not yet available in our framework.

Appendix B

Useful Lemmas

Lemma 5.1: Consider the block matrix [101]:

$$A = \begin{bmatrix} A_1 & B \\ B^T & A_2 \end{bmatrix} \tag{B.1}$$

If either of the following conditions is satisfied:

1. $A_1 = A_1^T > 0$ and $A_2 = A_2^T > B^T A_1^{-1} B > 0$
2. $A_2 = A_2^T > 0$ and $A_1 = A_1^T > B A_2^{-1} B^T > 0$

then $A > 0$.

Proof: First notice that A_1 and $A_2 - B^T A_1^{-1} B$ are both symmetric, thus,:

$$\Psi = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 - B^T A_1^{-1} B \end{bmatrix} \tag{B.2}$$

is symmetric. Moreover, because $A_1 > 0 \Leftrightarrow Eig[A_1] > 0$ and $A_2 - B^T A_1^{-1} B > 0 \Leftrightarrow Eig[A_2 - B^T A_1^{-1} B] > 0$ ($Eig[X]$ is the eigenvalue of X). As a result:

$$Eig \begin{bmatrix} A_1 & 0 \\ 0 & A_2 - B^T A_1^{-1} B \end{bmatrix} = \{Eig[A_1] \cup Eig[A_2 - B^T A_1^{-1} B]\} > 0 \tag{B.3}$$

Combine the two observations i.e. $\{\Psi = \Psi^T, Eig[\Psi] > 0\}$, the block matrix $\Psi > 0$:

$$\Psi = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 - B^T A_1^{-1} B \end{bmatrix} > 0$$

$$\Leftrightarrow \begin{bmatrix} I & 0 \\ B^T A_1^{-1} & I \end{bmatrix} \begin{bmatrix} A_1 & 0 \\ 0 & A_2 - B^T A_1^{-1} B \end{bmatrix} \begin{bmatrix} I & A_1^{-1} B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A_1 & B \\ B^T & A_2 \end{bmatrix} > 0 \qquad \text{(B.4)}$$

The second condition (2) can be proven in a similar manner.□

<u>Lemma 5.2:</u> Consider the block matrix:

$$P = \begin{bmatrix} A & B \\ 0 & D \end{bmatrix} \qquad \text{(B.5)}$$

If the following conditions are satisfied:

$$\begin{cases} D > 0 \\ 4\lambda_{min}(S(A)) > \|B\|^2 \|S(D)^{-1}\| > 0 \end{cases} \qquad \text{(B.6)}$$

where $S(D) = \frac{1}{2}(D + D^T)$ is the symmetric part of matrix D and $\lambda_{min}(S(A))$ is the minimum eigenvalue of matrix $S(A)$, then $P > 0$.

Proof: note that:

$$S(P) = \begin{bmatrix} S(A) & \frac{1}{2}B \\ \frac{1}{2}B^T & S(D) \end{bmatrix} \qquad \text{(B.7)}$$

Apply Lemma 5.1:

$$S(P) > 0 \Leftrightarrow \begin{cases} D > 0 \\ \mathbf{x}^T S(A)\mathbf{x} - \frac{1}{4}\mathbf{x}^T(BS(D)^{-1}B^T)\mathbf{x} > 0, \forall \mathbf{x} \in R^m \end{cases} \qquad \text{(B.8)}$$

<u>Remark 1</u>: If $A > 0 \Leftrightarrow S(A) > 0 \Leftrightarrow \mathbf{x}^T S(A)\mathbf{x} > \lambda_{min}(S(A)) \|\mathbf{x}\|^2 > 0, \forall \mathbf{x} \in R^m$.

<u>Remark 2</u>: From the spectral norm properties, $\frac{1}{4}\mathbf{x}^T \left(BS(D)^{-1}B^T\right)\mathbf{x} \le \left|\frac{1}{4}\mathbf{x}^T \left(BS(D)^{-1}B^T\right)\mathbf{x}\right| \le \frac{1}{4}\left\|BS(D)^{-1}B^T\right\| \|\mathbf{x}\|^2 \le \frac{1}{4}\|B\|^2 \|S(D)^{-1}\| \|\mathbf{x}\|^2$.

If the following condition is satisfied:

$$\lambda_{\min}(S(A))\|\mathbf{x}\|^2 > \frac{1}{4}\|B\|^2 \left\|S(D)^{-1}\right\| \|\mathbf{x}\|^2$$

$$\Leftrightarrow 4\lambda_{\min}(S(A)) > \|B\|^2 \left\|S(D)^{-1}\right\| \qquad \text{(B.9)}$$

then:

$$\begin{aligned}
\mathbf{x}^T(S(A))\mathbf{x} \geq \lambda_{\min}(S(A))\|\mathbf{x}\|^2 > \frac{1}{4}\|B\|^2 \left\|S(D)^{-1}\right\| \|\mathbf{x}\|^2 \geq \\
\left|\frac{1}{4}\mathbf{x}^T \left(BS(D)^{-1}B^T\right) \mathbf{x}\right| \geq \mathbf{x}^T \left(\frac{1}{4}BS(D)^{-1}B^T\right) \mathbf{x}, \forall \mathbf{x} \in R^m \\
\Leftrightarrow \mathbf{x}^T(S(A))\mathbf{x} > \mathbf{x}^T \left(\frac{1}{4}BS(D)^{-1}B^T\right) \mathbf{x}, \forall \mathbf{x} \in R^m \\
\Leftrightarrow \mathbf{x}^T(S(A))\mathbf{x} - \mathbf{x}^T \left(\frac{1}{4}BS(D)^{-1}B^T\right) \mathbf{x} > 0, \forall \mathbf{x} \in R^m \\
\Leftrightarrow \mathbf{x}^T \left(S(A) - \frac{1}{4}BS(D)^{-1}B^T\right) \mathbf{x} > 0, \forall \mathbf{x} \in R^m \\
\Leftrightarrow S(A) - \frac{1}{4}BS(D)^{-1}B^T > 0
\end{aligned}$$

As a result, $P > 0. \square$