Model-Based Compensation for Serial Manipulators through Semi-Parametric Gaussian Process Regression

Sathish Krishnamoorthi

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Model-Based Compensation for Serial Manipulators through Semi-Parametric Gaussian Process Regression

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Sathish Krishnamoorthi

August 13, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Model-Based Compensation for Serial Manipulators through Semi-Parametric Gaussian Process Regression

by

Sathish Krishnamoorthi

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: August 13, 2018

Supervisor(s):

dr.-Ing. Jens Kober

Reader(s):

Prof.dr.ir. Martijn Wisse

dr. Manon Kok

Abstract

Industrial robots can be found in automotive, food, chemical, and electronics industries. These robots are often caged and are secluded from human beings. A recent trend in a subclass of industrial robots named collaborative robots allows the humans to interact with the robots safely. The word "safety" mentioned above is of supreme importance. The safety is achieved in these robots by their lightweight and sleek design. Often, robots are operated under low stiffness conditions to achieve less impact force during an unavoidable collision. A severe damage to the environment may occur if the robot becomes unstable under any conditions. It is of paramount importance for the controller present in the robot to stabilize the system under all conditions. One such controller is the joint impedance controller, which helps the robot to interact with an unknown environment by causing no harm to humans.

The thesis marks its importance, as it is closely related to ensuring safety in collaborative robots and is mainly focused on tackling the situations whenever the controller fails. The controller in these manipulators has an Inverse Dynamics Model (IDM) and a Proportional Derivative (PD) controller. Under low stiffness and damping conditions, the PD gains are low and the manipulator is entirely compensated by the inverse dynamics model. This inverse dynamics model can become problematic in the presence of the un-modeled dynamics like flexibility, friction, dynamics of hydraulic tubes, actuators and cable drives or if the IDM model is inherently inaccurate. Consequently, the in-built joint impedance and position controller will fail to work under low stiffness and damping conditions, in-turn making the robot unstable. If this robot was to be used on an industrial platform and the problem is unresolved, it might cause some danger to the humans working closely and also damage the environment and itself.

Since the robot is entirely compensated by the IDM under low stiffness and damping conditions, the thesis tries to acquire the accurate IDM of the robot for control purposes. To do so, two cases were modeled in this thesis, one with the internal IDM with correct base parameters and another one with the incorrect internal IDM by adding offset in the base parameters. But in both cases, the internal IDM model failed to compensate for the un-modeled dynamics occurring in the manipulator.

The thesis incorporates a semi-parametric Gaussian process regression to tackle the two cases. A semi-parametric model consists of a parametric term and a non-parametric term. First, the parametric term is identified using the least squares approach. Later, the parametric term is used as mean to capture the non-parametric term using the Gaussian Process Regression.

The proposed methods were tested on the PUMA 560 robot and the two-link manipulator in MATLAB. From the simulation results, the semi-parametric model was able to provide accurate feed-forward control torques to compensate for the model inaccuracies and the unmodeled dynamics at low stiffness and damping conditions. Additionally, implementing these proposed methods on a real robot will be a future scope of improvement on this topic.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to thank my master thesis advisor Dr.-Ing. Jens Kober for trusting in me and giving me a golden opportunity to work in Robotics. Even though in his busy schedule, he never hesitated in giving appointments for meeting every week. I got introduced to the world of machine learning only through this project given by him. He always pointed me in the right direction whenever I got stuck. Only with his guidance and right feedback, I was able to complete my master thesis.

Secondly, a special thanks to my friend Nivas Kumar in helping me in proof-reading my report. He always enjoyed listening to my work and was never reluctant in helping me out. Sometimes, he used to help me with general MATLAB commands. In addition, I wanted to extend my thanks to Naveen Rajasekaraen and Navadeep. Often, we discuss the intersections of our thesis. With those fruitful conversations, I was able to understand my thesis topic in-depth and from a broader perspective.

Third, I want to thank my best friend Salai Varun. More than a friend, he is a brother to me. Whenever I was depressed, he was always there to motivate and encourage me.

Last but foremost, I would extend my gratitude to my mother Usha and father Krishnamoorthi. Even with million miles apart, their love and care for me never ends. Only because of them, I am whom I am today. Their constant prayers and moral support always pushed me further, whenever I got bogged down.

Delft, University of Technology — Sathish Krishnamoorthi
August 13, 2018

"If you fail, never give up because F.A.I.L. means First Attempt In Learning. End is not the end, in fact E.N.D. means Effort Never Dies. If you get No as an answer, remember N.O. means Next Opportunity."

"If you want to shine like sun, first burn like a sun"

— *Dr. A.P.J. Abdul Kalam*

Chapter 1

Introduction

Robotic manipulators can be found in many industries namely, automotive, electronics, chemical, and food industries. The demand for robotic manipulators is predicted to rise in the upcoming years [3]. An important growing subclass that needs more attention and development is collaborative robotics [1]. So, the project closely focuses on the research and development of collaborative robotics. In Section 1-1, a general information on collaborative robotics is given. In Section 1-2, the problem statement for the thesis is formulated. Subsequently in Section 1-3, approaches to solve the problem stated in Section 1-2 is discussed. Section 1-4 discusses in detail, the reasons for the problem stated in Section 1-2 and the importance of carrying out this project. Section 1-5 talks about the new contributions offered by this thesis towards solving the problem statement. Lastly, Section 1-6 ends the chapter with an outline of the entire thesis.

1-1 Collaborative Robotics

In earlier days, industrial robots were caged and cut off from human beings for safety conditions. Whereas in the past decade, there is a growing interest in bringing humans and robots closer together in the manufacturing working environment [2]. The subclass of industrial robots that solves the above purpose is a COBOT (COllaborative industrial roBOT). So, when the robots share their work-space with humans, safety is of paramount importance. The safety in these robots is usually ensured by low weight robot design and good compliance behavior [5]. Some robots even have sensitive detection of contacts and collisions to allow for a safer interaction with humans. On the one hand, the compliance in these robots is achieved by introducing springs and other flexible mechanisms in their joints. This kind of robot with mechanical springs at all joints is called passive compliance. On the other hand, some manipulators give the end-users to set the desired stiffness and damping to achieve the desired compliance control [7]. This class of compliance falls under the category of active compliance. In general, compliance helps in decoupling the inertia of the impacting link with the rest of the robot. Therefore, only the impacting link is felt during the collision. Virtually all major manufacturers produce collaborative robots, including Fanuc, ABB, KUKA,

Rethink Robotics, Universal Robot, and Bosch. The robots produced by these companies may have a different structure, size, and capabilities but they all possess the aforementioned characteristics to ensure safety.

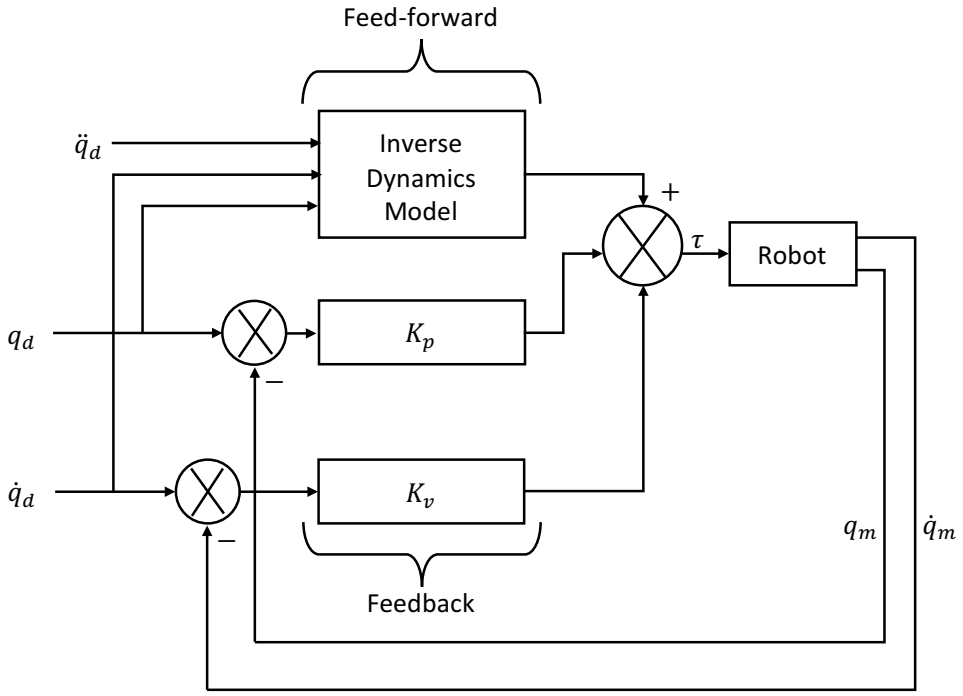

Figure 1-1: Inverse Dynamics + PD controller, where K_p, K_v is the proportional and derivative gain, τ is the joint torque, $q_d, \dot{q}_d, \ddot{q}_d$ represents the desired joint positions, velocities and accelerations, $q_m, \dot{q}_m$ denotes the measured joint positions and velocities.

1-2 Problem Formulation

In General, the serial manipulators have two controllers namely, the feed-forward and the feedback controller as depicted in Figure 1-1 [11]. The feed-forward controller uses the inverse dynamics model of the robot. Whereas, the feedback controller is basically a Proportional-Derivative (PD) controller. The robot is entirely compensated by the inverse dynamics model of the robot under low stiffness and damping conditions. Often, the inverse dynamics model is inherently inaccurate or it fails to capture the un-modeled dynamics like flexibility, dynamics of hydraulic tubes, actuator, and cable drives. Thereby, making the system unstable at low PD gains. Therefore, the remainder of the report focuses on

"***Acquiring the accurate inverse dynamics model of the manipulator by capturing the model inaccuracies, un-modeled dynamics and other perturbations for model-based compensation***"

1-3 Approach

In this thesis, the problem statement is formulated into two cases. The brief description of the two cases and the approach incorporated to tackle the cases is depicted below.

- **Case 1** - For the first case, the robot depicted in Figure 1-1 is compensated by an internal Inverse Dynamics Model (IDM) with correct base parameters. However, the internal IDM fails to compensate for the un-modeled dynamics occurring in the manipulator. It is assumed in this case, that it is possible to replace or turn OFF the internal IDM.

  The un-modeled dynamics that cannot be explained by the internal IDM, is captured in two steps to tackle the first case. First, the least squares approach is used to estimate the base parameters and the linear coefficients of viscous and Coulomb friction of the robot. With these estimated parameters, an Rigid Body Dynamics (RBD) model can be built, which gives a unique relationship between the joint torques with the joint position, velocity, and acceleration. Second, the Gaussian Process Regression (GPR) is used to capture the non-linearities that cannot be explained by the estimated RBD model. After GPR, the internal IDM is replaced by the estimated RBD model. Then, the feed-forward torques from the Gaussian process model are given to the robot to compensate for the un-modeled dynamics that cannot be compensated by the estimated RBD model

- **Case 2** - Unlike case 1, the robot depicted in Figure 1-1 is compensated by an incorrect internal IDM for the second case. The internal IDM is made incorrect by introducing offset in the base parameters. Like case 1, the internal IDM also fails to compensate for the un-modeled dynamics occurring in the manipulator. But in this case, it is assumed that the internal IDM cannot be replaced or set to zero.

  To solve this issue, first, the RBD error model is identified using the least squares approach. This model compensates for the model inaccuracies occurring in the internal IDM. Subsequently, the RBD error model is used as mean to capture the un-modeled dynamics occurring in the manipulator using the GPR. After GPR, the feed-forward signals are given to the robot in the presence of incorrect internal IDM. These feed-forward signals are combinations of torque signals from the RBD error and Gaussian process models for a given joint position, velocity, and acceleration.

The second case is more interesting than the former as many robots do not provide the end-users the possibility to set the internal IDM to zero. Instead, some robot gives the users the opportunity to overlay torque signals with the torque signals coming from the low-level controller. For example, the KUKA LWR 7 R800 allows the user to overlay torque signals using a platform named Fast Research interface [78].

1-4 Motivation

From Section 1-2, it can be inferred that the project demands an accurate model to do model-based control. It is required to compensate for the inaccurate inverse dynamics model used

in the joint position and impedance controller under low stiffness and damping conditions. Hence, the necessities of model-based control, joint impedance and position control, low stiffness and its relevance to day-to-day applications are listed below.

Need for joint impedance and position control: The importance of the joint position and the impedance controller can be traced to many applications like welding, cutting of metals, deburring, polishing, pick and place of heavy objects, assembling, etc.. The joint position controller helps in tracking the desired trajectory asymptotically. For example, cutting of sheet metals in automotive industries in the desired manner. Meanwhile, the joint impedance controller ensures that the end-effector follow the desired trajectory in a stable way in the presence of external forces and torques [8]. So, it is widely used in areas where the contact forces play a major role, for e.g., deburring, polishing, assembling, etc.. So, the active working of these controllers at all situations is of prime importance.

Although the proposed methods in this thesis, can be extended to Cartesian space control with an additional forward kinematics transformation, the thesis concentrates more on the joint space control over the Cartesian space control. In Cartesian space control, the task description is represented in a sequence of end-effector coordinates in the Cartesian space. Later, inverse kinematics transformation is used to convert the information of end-effector coordinates in the Cartesian spaces to a series of angular positions in the joint space. Then, the end-effector is controlled indirectly by controlling the joint angles which correspond to the end-effector coordinates through forward kinematics. This kind of indirect approach to end-effector control is both computationally inefficient and becomes complicated as different combinations of joint angles might exist for a given end-effector condition. Furthermore, small errors in the joint angles may result in accumulation of large errors in the end-effector coordinates, depending on the manipulator geometry. So in this thesis, the joint space control is preferred over the Cartesian space control.

Need for low stiffness: When it comes to industrial robots like welding robots, a stiff actuator improves the precision, stability, and bandwidth of position control. But, high stiffness may not satisfy all applications, for e.g. application of robot doing pick and place operation, packaging, material handling or assembling. It requires the robot to work under variable stiffness condition. Therefore, irrespective of any stiffness value, it is of paramount importance for the robot to be stable at all conditions. In addition, the impact of force during the collision of the robot under low stiffness condition is low [5]. So, cobot working in low stiffness condition improves safety conditions as well.

Need for Model Based Control: Acquiring an accurate inverse dynamics model is important as the dependency on feedback term will reduce. Generally, the feedback term compensates for the inaccuracies in the inverse dynamics model and non-linearities acting on the robot. Being said, the joint impedance controller feedback term becomes almost zero at low stiffness and damping conditions. If the inverse dynamics model is imperfect, the demand arises to do a model based control to acquire a perfect inverse dynamics model to make the system stable [48], [65], [76]. Meanwhile, increasing the feedback gain to compensate for the offset will increase the stiffness of the robot. Higher stiffness leads to decreased compliance, saturation of actuators and may cause severe damage in case of collision [5].

1-5 Research Contribution

The thesis combines the works of linear identification techniques addressed in [49], [32], [35], [33], [34], [37], [41], [36] and semi-parametric Gaussian Process Regression from [65] to do model-based control of serial manipulators. The previous research works [49], [32], [35], [33], [34], [37], [41], [36] does not take into account the dynamics of electrical cables, actuator dynamics and flexibility. Meanwhile, [65] do address the aforementioned un-modelled dynamics but has the following shortcoming.

- GPR is trained with an excitation trajectory which is a superposition of two sine waves. The amplitude and frequency values of the sine signal were chosen randomly. Due to this, the optimally of the trajectory was not considered. The trajectory lacks its richness in exciting all the base parameters and un-modeled dynamics.

- D.N. Tuong and J. Peters addressed case 1 (refer Section 1-3) for the 7 DOF real Barrett WAM and 7 DOF SARCOS robot arm. However, the semi-parametric Gaussian process regression was not incorporated to capture the inaccuracies or offsets occurring in the base parameters of the internal inverse dynamics model of the robot.

The project handles the first shortcoming by exciting all the base parameters and the un-modeled dynamics. The excitation trajectory is generated by minimizing a criterion to acquire the optimal amplitude values for the trajectory. Additional signals were used to sweep the entire closed-loop bandwidth frequency of the manipulator to capture the un-modeled dynamics. The second shortcoming is modeled by deliberately including offset in the internal rigid body dynamics of the robot. Then, the issue is solved by acquiring the inaccuracies in IDM through least squares approach and further capturing the nonlinearities through Gaussian process regression.

On a lighter note, the thesis also differs from [65] from the kind of prior knowledge taken for training the Gaussian process regression. The RBD model of 7 DOF Barrett WAM robot is obtained from CAD data, which is the approximate values of all standard inertial parameters. It is utilized as mean for GPR to capture the non-linearities in [65]. In this project, the RBD model is obtained using the least squares technique and the model is a reduced model set of the manipulator with base parameters.

Additionally, the thesis uses the setting used for case 1 for different evaluation studies. In this thesis, three different evaluation studies were done using the first case and they are given as follows:

- Study of over-fitting in the model selection during Gaussian Process Regression.

- Studying the importance of richness in the excitation during the least squares identification and GPR.

- Studying the performance of the semi-parametric models trained with different trajectories, which are generated using different optimization criteria.

1-6 Thesis Outline

The rest of the report is structured as follows,

- Chapter 2 discusses modeling of serial manipulators in general. The model is divided into a parametric and a non-parametric equivalent in this chapter, where it meticulously ponders on these terms. In addition, the chapter confers about a method for model reduction in the Rigid Body Dynamics (RBD) to obtain a minimum set of base parameters for identification.
- Chapter 3 gives insights on the previous research works related to the RBD identification with optimal excitation of the manipulators. Later, the chapter focuses on a particular class of trajectory named Fourier series. In addition, the chapter introduces a novel excitation signal for the RBD identification and training the Gaussian Process Regression.
- Chapter 4 talks about previous research work on the identification of the inverse dynamics model of the manipulator. Then, it lays out the reasons for adopting a certain technique for identification and discusses them in detail. Later, the chapter introduces a new approach to capture the offsets occurring in the internal rigid body dynamics model to the solve the problem statement.
- Chapter 5 tests the proposed methods discussed in Chapters 2, 3, and 4 on the 6-DOF PUMA-560 manipulator and a two-link manipulator. It models and addresses the two cases depicted in Section 1-3. Furthermore, the problem of the over-fitting in model selection during GPR is discussed. The chapter also gives a comparison study on the tracking performance of the semi-parametric models trained with different trajectories, generated with different optimization criteria.
- Chapter 6 ends the report with a conclusion. In addition, recommendations were given in this chapter for future research work on this topic.

Chapter 2

Modelling

A semi-parametric model includes a parametric and a non-parametric term. Here, the parametric term is the Rigid Body Dynamics (RBD) of the manipulator. It gives a unique and global relationship between the joint actuator torques with joint positions, velocities and accelerations [65]. Whereas, the non-parametric term includes the un-modeled dynamics. Therefore, this chapter discusses the two aforementioned terms in detail. The Euler-Lagrange method is used in Section 2-1 to formulate the parametric term. In Section 2-2, the parametric term is discussed meticulously. In addition, it also discusses model reduction to obtain the minimum set of base parameters. Section 2-3 discusses the non-parametric term. Lastly, the chapter ends with a conclusion.

2-1 Euler-Lagrange Formulation

The inverse dynamics model of a n DOF robot can be obtained by two standard approaches. The first method is the Newton-Euler's formulation, where it yields the model in a recursive form for each link separately. The second method is based on the Euler-Lagrange formulation, where it delivers equation for the entire robot body. Unlike Newton-Euler, this approach is simple and systematic as the closed-form symbolic equations can be directly obtained [11]. The closed-form expressions will be proved to be useful in the parameter estimation and trajectory optimization, which will be discussed in further chapters. So, the energy-based Euler-Lagrange formulation is incorporated and it is given by

$$\tau = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}}\right]^T - \left[\frac{\partial L}{\partial q}\right]^T + \tau_f, \tag{2-1}$$

where τ_f denotes the torque due to friction, q, $\dot{q}$ are the joint position and velocity respectively and L is the Lagrangian of the robot, which is taken to be difference between the kinetic energy E and the potential energy U. After developing the Equation (2-1), we obtain,

$$\tau(q, \dot{q}, \ddot{q}) = \underbrace{M(q,\nu)\ddot{q} + C(q,\dot{q},\nu)q + G(q,\nu)}_{\tau_i} + \underbrace{F_v\dot{q} + F_c\mathrm{Sgn}(\dot{q})}_{\tau_f}. \tag{2-2}$$

Table 2-1: The definition of parameters defined in Equation 2-2

Notations	
q, $\dot{q}$, $\ddot{q}$	Joint position, velocity and acceleration
τ, τ_i, τ_f	Joints torques, joints inertial torques, joints friction torques
$M(q)$, $C(q,\dot{q})$, $g(q)$	Inertia matrix, Coriolis matrix and gravitational term
F_v, F_c	Viscous and Coulomb friction

The Equation 2-2 considers the widely accepted linear friction model with viscous and Coulomb friction [45]. The variables used in Equation (2-2) are defined in Table 2-1. In Equation (2-2), the inertial matrix $M(q,\nu)$, Coriolis matrix $C(q,\dot{q},\nu)$ and $G(q,\nu)$ are the non-linear functions of model parameters ν, which includes the Center of Mass (CoM), inertia and mass of each link. By barycentric parmaterization, the Equation (2-2) can be written in a linear fashion which will be discussed in the upcoming section.

2-2 Parametric Term

Using the barycentric parameters (χ) and the Denavit Hartenberg (DH) convention, Equation (2-2) can be re-grouped in a linear fashion [11]. It is given as,

$$\tau(q,\dot{q},\ddot{q}) = \underbrace{W(q,\dot{q},\ddot{q})\chi}_{\text{Parametric}}, \tag{2-3}$$

where $\chi \in \mathbb{R}^{m\times 1}$ is the set of dynamic model parameters and $W \in \mathbb{R}^{n\times m}$ is the observation matrix. The observation matrix W in Equation (2-3) can be obtained by using the Lagrange-Euler method recursively. The matrix W is a upper triangular matrix as a function of joint positions, velocities and accelerations and it is given by,

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \vdots\\ \tau_n\end{bmatrix} = \begin{bmatrix}W_{11} & W_{12} & \cdots & W_{1n}\\ 0 & W_{22} & \cdots & W_{2n}\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & W_{nm}\end{bmatrix}\begin{bmatrix}\chi_1\\ \chi_2\\ \vdots\\ \chi_m\end{bmatrix}. \tag{2-4}$$

where m is the number of base parameters, n is the DOF of the robot. The dynamic parameter χ in Equation (2-3) consist of the following terms to be estimated

$$\chi = \begin{bmatrix}XX_i & XY_i & XZ_i & YY_i & YZ_i & ZZ_i & MX_i & MY_i & MZ_i & M_i & f_{vi} & f_{ci}\end{bmatrix}, \tag{2-5}$$

where M_i is the link mass, $\begin{bmatrix}MX_i & MY_i & MZ_i\end{bmatrix}$ are the three components of the first moment of inertia of link i, $\begin{bmatrix}XX_i & XY_i & XZ_i & YY_i & YZ_i & ZZ_i\end{bmatrix}$ are the six components of the inertial tensor of link i and f_{vi}, f_{ci} are the coefficients of viscous and Coulomb friction respectively [11]. The relationship between the inertial tensor with respect to the DH coordinate frame and the inertial tensor with respect to the coordinate frame which is fixed at the center of mass can be obtained by applying either the parallel axis theorem or the Steiner theorem. Some

manufacturers share the approximate values of the inertia tensor, masses and the center of masses of each link of the manipulator. But, this information is provided with the respect to the center of mass as the frame of each link and cannot be used for model-based control directly. So, the Steiner theorem becomes handy in defining the inverse dynamics model to do model-based control. In addition, it will be useful for cross-validation of the obtained base parameters through linear identification, if the approximate initial data is available.

It is worth noting that not all dynamic parameters χ contribute to the robot dynamics. Indeed, not all the parameters are observable through measuring the joints positions, velocities, accelerations and torques. As a result, the matrix W will not be full row rank as some parameters will be in linear combination with other parameters. Hence, to reduce the computational effort and to acquire a robust dynamic model, it is paramount to derive a minimum set of inertial parameters (Base parameters) to be estimated [15]. To do so, first the frames are assigned for each joints and Denavit Hartenberg (DH) table is filled out. Later, the dynamic parameters are regrouped with the help of theorem depicted in Section A-1-1. This sequential procedure can be applied to all manipulators. One example is illustrated in Appendix A, where the model reduction is done for the 6 DOF PUMA 560 robot.

After regrouping of the dynamic parameters, a reduced parameter set called the base parameters (χ_b) can be obtained. Subsequently, the matrix ϕ_b can be built with the columns corresponding to the base parameters. So, the Equation 2-3 is transformed into,

$$\tau(q, \dot{q}, \ddot{q}) = \phi_b(q, \dot{q}, \ddot{q})\chi_b. \tag{2-6}$$

Since the joint position, velocity, and acceleration are functions of time, the matrix is stacked row-wise for each time instant. So, the observation matrix constructed for p measurements is given as follows,

$$W_b = \begin{bmatrix} \phi_b(q(t_1), \dot{q}(t_1), \ddot{q}(t_1))_{n\times m} \\ \phi_b(q(t_2), \dot{q}(t_2), \ddot{q}(t_2))_{n\times m} \\ \vdots \\ \phi_b(q(t_p), \dot{q}(t_p), \ddot{q}(t_p))_{n\times m} \end{bmatrix}. \tag{2-7}$$

where W_b is the observation matrix built with the columns of the base parameters (χ_b) for p measurements. The matrix (W_b) will be crucial in generating the excitation trajectory for the identification of the robot model, and its importance can be witnessed in the upcoming chapters.

Deriving the columns of observation matrix W is quite complex for a robot with more than 2 DOF [14] [12]. Open source softwares like Sympybotics[1], SYMORO (SYmbolic MOdeling of RObots) [13] or Damarob [2] can be used to compute the columns of observation matrix W. In this thesis, Sympybotics is used and the software takes DH table as input and returns the columns of the matrix W as output. More information on acquiring the matrix W by using Sympybotics can be referred to Appendix A.

[1] https://github.com/cdsousa/SymPyBotics
[2] http://www.damarob.altervista.org/download.html

2-3 Non-Parametric Term

The nonlinear friction τ_f can be modelled in many ways. One standard approach is to represent the friction τ_f as given in the Equation (2-2). This kind of linear friction modelling is used in [32], [35], [45], [41]. But, the modelled friction fails to capture Stribeck, stiction at low velocity region. In addition, modelling errors also occurs from un-modeled dynamics like flexibility, dynamics of hydraulic tubes, actuator and cable drives etc which will lead to poor model based control. Hence, a RBD model with linear friction will not be suffice. To resolve this, the Equation (2-6) is modified into

$$\tau(q, \dot{q}, \ddot{q}) = W_b(q, \dot{q}, \ddot{q})\chi_b + \underbrace{\epsilon(q, \dot{q}, \ddot{q})}_{\text{Non-Parametric}} , \tag{2-8}$$

where ϵ represents the non-linear terms due to un-modeled dynamics [65].

2-4 Discussions

First, the chapter discussed about the modelling of serial robots in general. Subsequently, a solution to obtain the minimum base parameter set and the columns of observation matrix W was reported. Lastly, the robot was modelled with an additional term (ϵ) to captures all the non-linearities and the un-modeled dynamics. The observation matrix W_b discussed in this chapter will be proved to be crucial in the upcoming chapters, especially in Chapter 4 where it would be vital for trajectory optimization and least squares estimation. Furthermore, the report is structured in a such way that RBD model (parametric term) is obtained through linear identification technique. Later, it is used as mean for Gaussian Process Regression to the capture the non-parametric term (See Chapter 4).

Chapter 3

Excitation Trajectory

A key to proper system identification lies in its collected input-output data. Only certain combinations of joint positions, velocities, and accelerations give the needed information to identify the dynamic model properly. The trajectory which gives this information is called an excitation trajectory. To generate such an excitation trajectory for a manipulator, the trajectory has to be parametrized and the parameters have to be estimated using an optimization problem obeying the physical constraints of the robot. Section 3-1 lists the previous research works on optimal robot excitation and in addition, it points out the pros and cons of different excitation trajectories. Section 3-2 discusses an excitation trajectory named Fourier series (FS) in detail. In Section 3-3, an optimization problem is designed to estimate the parameters of the trajectory discussed in Section 3-2. Section 3-4 proposes a new excitation trajectory and a modified trajectory optimization criterion to acquire a better semi-parametric model. Lastly, the chapter ends with a conclusion.

3-1 Previous Works

The most standard trajectory used for robot excitation is the Fourier Series (FS). It is used for identification of the RBD model of the 6 DOF Staubli TX-90 robot in [41], the 6 DOF KUKA IR:250 industrial robot in [36], the 6 DOF KUKA KR150 in [46] and the 7 DOF in KUKA lightweight robot IV in [33]. Sine series as an excitation trajectory was adopted in [50] for identification of the IDM of a 2-DOF SCARA robot. Subsequently, from the class of periodic trajectories, cosine series with fifth order polynomial was used for identification of inverse dynamics model of the 6 DOF CRS A465 industrial robot in [51] and the 7 DOF KUKA LBR iiwa 14 r800 in [49]. Aforementioned trajectories enjoy the following advantages.

- They have improved signal to noise ratio as it allows for data averaging in the time domain which in turn helps in calculation of joint torque and position covariances. On the other hand, data averaging also reduces the number of data measurements.
- The joint velocities and accelerations can be calculated analytically. Hence, numerical differentiation method can be avoided, which is an error-prone procedure.

- With these trajectories, frequency domain differentiation can be done to obtain joint velocities and accelerations from joint positions. This approach of differentiation of joint positions is simple, efficient and accurate. Due to the periodicity of the signal, there is no leakage error for the transformation of the signal from time domain to frequency domain [46], [51].

- The filtering of the joint positions of the Fourier Series can be easily done in the frequency domain by using the property of windowing the spectrum. This technique will reduce the noise at the desired spectrum without introducing any phase distortions.

In the class of non-periodic trajectories, Jubien et al. used Point-To-Point (PTP) excitation trajectories to identify the model of the 7 DOF KUKA LWR 4 robot in [34] and [32]. PTP consists of typical trapezoidal acceleration shapes where the desired end positions of each joint were given as input to the robot. Accordingly, the robot calculates which joint needs the longest time to move, by considering the distance, velocity and acceleration limits. The disadvantage of this method is quite obvious as it does not gives the end user to shape the trajectory according to their needs. Following, B-splines as excitation trajectory was tested on the 7 DOF KUKA LWR 4 robot for the identification of inertial base parameters in [52]. B-splines are smooth curves and are formed by compounding multiple Bezier curves. This kind of trajectory was used for the identification of inertia parameters alone and does not have enough excitation for the identification of full base parameters [52] [49]. Overall, the non-periodic trajectories do not possess the advantages exhibited by the periodic trajectories and have minimal excitation as compared to periodic trajectories [49].

The main problem concerning the excitation trajectory is that it should be able to excite all the base parameters obeying all the robot constraints. The idea started with moving one or two joints at a time and identifying the base parameters corresponding to that joint by the Least Squares (LS) method. This approach of segmented identification was used by authors like H. Mayeda, P. Khosla, C.G. Atkeson, H.B. Olsen and T. Beckey. But, neither this method yielded a global LS solution nor the optimality of trajectories were considered. B. Armstrong found a solution to the aforementioned problem by minimizing the condition number of an observation matrix (W_b), which is estimated from the joint positions, velocities, and accelerations (See Chapter 2). Thereby, laying the foundation for trajectory optimization. Minimizing the condition number of observation matrix W_b is a standard approach and it is used in [56], [26], [46], [35], [57], [36], [30], and [47] for the optimal robot excitation. Later on, different objective functions were constructed for trajectory optimization. A slight modification was done to the minimization of condition number by including the torque co-variance matrix into account for the computation of optimal parameters of the periodic wave. This approach was successfully implemented in [33], [44], [50], [51], [45]. In [41], a d-optimality criterion was proposed which is basically the minimization of $-\log(\det(W_b^T \Sigma^{-1} W_b))$. Later, a slight variation of the d-optimality criterion was proposed in [41] to reduce the computational cost. The basic idea is to replace the determinant expression in d-optimality criterion with Hadamard's inequality [38]. Due to this, the complexity of computing the determinant of $W_b^T W_b$ for a $m \times n$ matrix dropped from $O(mn^2 + n^3)$ to $O(n)$.

3-2 Trajectory Parametrization

From Section 3-1, it is clear that the class of periodic signals are better compared to non-periodic signals. The advantage of noise free differentiation exhibited by periodic signals can be utilized for constructing the observation matrix W_b, which is the function of the desired joint positions, velocities, and accelerations. Therefore, the Fourier series is incorporated in this thesis. The Fourier series is a finite sum of sine and cosine functions of each joint i as represented in Equation (3-1). As mentioned in Section 3-1, the analytical differentiation of joint positions can be obtained using the following equation

$$q_i(t) = \sum_{l=1}^{L} \frac{a_{i,l}}{\omega_f l} \sin(\omega_f l t) - \frac{b_{i,l}}{\omega_f l} \cos(\omega_f l t), \tag{3-1}$$

$$\dot{q}_i(t) = \sum_{l=1}^{L} a_{i,l} \cos(\omega_f l t) + b_{i,l} \sin(\omega_f l t), \tag{3-2}$$

$$\ddot{q}_i(t) = \omega_f l \sum_{l=1}^{L} -a_{i,l} \sin(\omega_f l t) + b_{i,l} \cos(\omega_f l t), \tag{3-3}$$

where i runs from $1, 2, .., n$ with n being the degree of freedom of the robot, q_i, $\dot{q}_i$, $\ddot{q}_i$ are the desired joint positions, velocities and accelerations respectively, ω_f is the fundamental frequency, $a_{i,l}$ and $b_{i,l}$ are the amplitudes of the cosine and sine function, and L denotes the number of sine and cosine terms.

It is to be noted from Equations (3-1), (3-2), (3-3) that fundamental frequency ω_f is common for all joints. This makes the excitation trajectory periodic. The Fourier series represented in Equation (3-1) covers a frequency ranges between $[\omega_f, N\omega_f]$. It is advisable to choose a low fundamental frequency ω_f, as it would lead to longer excitation period. The longer measurement time will cover a larger part of the robot workspace for a given maximum joint velocity. In addition, it will improve the information content of the measurements and the accuracy of the parameter estimates [45]. Each Fourier series contains $2 \times L + 1$ parameters that are needed to generate the reference trajectory. The parameters $a_{i,l}$ and $b_{i,l}$ can be determined through trajectory optimization which will be discussed in the upcoming section.

3-3 Trajectory Optimization

The parameters of the trajectory can be identified without solving any optimization problem by trial and error method. But, this way of random selection of amplitude values will result in poor excitation of base parameters. In addition, the observation matrix W_b which is constructed from the desired joint position, velocity and acceleration can become rank deficient and it might pose some problems when carrying out least squares estimation as the inverse will not exist [47]. Therefore, it is essential to construct a proper objective and constraints function for trajectory generation before carrying out the identification experiment. Subsection 3-3-1 discusses different constraints that are needed to be obeyed for the excitation trajectory. Subsection 3-3-2 presents a proper objective function that is needed to be minimized.

3-3-1 Trajectory Constraints

FS has a drawback due to sudden change in velocity and acceleration at the start and end of motion, i.e., $\dot{q}_i(0) \neq 0$, $\dot{q}_i(t_f) \neq 0$, $\ddot{q}_i(0) \neq 0$, $\ddot{q}_i(t_f) \neq 0$ for $i = 1, 2, ..., n$, which may cause the robot to vibrate. Due to this, it is difficult for robots to track FS signal closely. Thereby, deteriorating the identification accuracy [44]. Therefore, to avoid the large control action for tracking non zero initial conditions, the constraint Equations (3-4),(3-5), (3-6) were constructed. It ensures that the trajectory has zero initial and final joint positions, velocities and accelerations.

$$q_i(t_f) = q_i(0) = \sum_{l=1}^{L} \frac{b_{i,l}}{\omega_f l} + q_{i,0} = 0 \tag{3-4}$$

$$\dot{q}_i(t_f) = \dot{q}_i(0) = \sum_{l=1}^{L} a_{i,l} = 0 \tag{3-5}$$

$$\ddot{q}_i(t_f) = \ddot{q}_i(0) = \sum_{l=1}^{L} \omega_f l b_{i,l} = 0 \tag{3-6}$$

In Equations (3-4), (3-5), (3-6), $q_i(0)$, $\dot{q}_i(0)$, $\ddot{q}_i(0)$ are the initial joint positions, velocities and accelerations respectively and $q_i(t_f)$, $\dot{q}_i(t_f)$, $\ddot{q}_i(t_f)$ are the final joint positions, velocities and accelerations respectively. Likewise, a second group of constraints can be constructed based on robot's physical limits. In order to avoid collision of the robot with the environment and also with itself, the joint positions, velocities and accelerations of the desired trajectories should be limited by the following, constraints

$$|q_i(t)| \leq \sum_{l=1}^{L} \frac{1}{l\omega_f} \sqrt{a_{i,l}^2 + b_{i,l}^2} + |q_{i,0}| \leq q_{i,max}, \tag{3-7}$$

$$|\dot{q}_i(t)| \leq \sum_{l=1}^{L} \sqrt{a_{i,l}^2 + b_{i,l}^2} + |q_{i,0}| \leq \dot{q}_{i,max}, \tag{3-8}$$

$$|\ddot{q}_i(t)| \leq \omega_f \sum_{l=1}^{L} l\sqrt{a_{i,l}^2 + b_{i,l}^2} \leq \ddot{q}_{i,max}, \tag{3-9}$$

where $q_{i,max}$, $\dot{q}_{i,max}$, $\ddot{q}_{i,max}$ are the maximum joint position, velocity and acceleration respectively and $|q_i(t)|$, $|\dot{q}_i(t)|$, $|\ddot{q}_i(t)|$ denotes the minimum or lower bound on joint position, velocity and acceleration respectively. The aforementioned constraints will be incorporated in a non-linear constrained optimization problem with an objective function J, to find the amplitudes $a_{i,l}$ and $b_{i,l}$ for rich excitation which will be discussed in the upcoming subsection.

3-3-2 Trajectory Objective Function

Different objective functions will be discussed and compared in this project. The complexity is not an issue as the trajectory optimization is done offline. Therefore, the variant of d-optimality with Hadamard's inequality is avoided. In this section, the most standard objective function cond(W_b) will be discussed. The condition number of the matrix W_b is the measure of the sensitivity of the least squares solutions to the perturbations on the observation matrix

W_b and the joint torque τ (Refer Equation (4-3)) [39]. There are three different measures to compute the condition number of a matrix and they are given as follows

- 2-norm condition number using SVD decomposition,
- 2-norm condition number using QR decomposition,
- condition number with Frobenius norm - $\text{cond}_F(W_b) = \|A\|_F \|A^{-1}\|_F$.

These measures were compared and tested in [47]. According to [47], the last measure gave a faster convergence rate as compared to the other two, but all the three gave similar results. So, the 2-norm condition number using SVD decomposition will be adopted in this project as the computation effort is not a big priority. The 2-norm condition number is given as the ratio of the maximum singular value ζ_{max} and the minimum singular value ζ_{min} of the matrix W_b as depicted in Equation (3-10) [49].

$$J = \min_{a_{i,l}, b_{i,l}} \text{cond}(W_b) = \min_{a_{i,l}, b_{i,l}} \frac{\zeta_{max}}{\zeta_{min}} \tag{3-10}$$

The smaller condition number leads to the well-equilibrated observation matrix and higher excitation of the base parameters. In the meanwhile, the impact of perturbations on the least squares solution is reduced [36] [47].

3-4 New Approach

The main goal of the thesis is to identify the accurate inverse dynamics model of the manipulator. A brief description on how to acquire it is discussed in Chapter 2. It is difficult and quite complex to model the dynamics of hydraulic tubes, actuator and cable drives and other non-linearities. So, GPR is incorporated to tackle the issue. However, for the GPR to capture these un-modeled dynamics, it is imperative to excite all these non-linearities. In [54], general combinations of sinusoidal trajectories were used as the reference for every joint with relatively high accelerations so that the dynamics effects were observable. This kind of excitation trajectory was used in [54] to train the neural network for inverse dynamics learning. Likewise in [65], [74], excitation trajectory with two sinusoids having different frequencies and amplitudes were chosen for training the GPR. The aforementioned used random trajectory parameters and it is highly questionable if these trajectories excited all the non-linearities.

In [41], [36], [46], [33], [52], [49], [53], [45], [44], [43] and [40] a linear viscous and Coulomb friction model was embedded in the observation matrix and the trajectory was optimized. Often, the information of the non-linearities and what kind of friction acting on the robot is not predictable. In this thesis, a different approach is taken where the two excitation trajectories were generated for the identification experiment. First, a Fourier series is optimized and generated in such a way that it excites all the base parameters of the manipulator alone. The generated trajectory has the frequency ranging from ω_n to $N\omega_n$, where N denotes the number of Fourier terms, ω_n is the fundamental frequency. Secondly, an another Fourier Series is generated that covers a frequency range from $[N\omega_n, \omega_b]$, where is the ω_b is the closed-loop bandwidth of the system. This trajectory is a high frequency and a high acceleration trajectory

where the amplitudes are chosen in a such way, it obeys all the constraints mentioned in Equation (3-4),(3-5), (3-6), (3-7), (3-8) and (3-9). By incorporating this kind of trajectory for identification, the disturbances due to non-linearities can be purposefully captured in Gaussian process regression.

The sensitivity of the least squares approach for the RBD identification depends on the condition number of the observation matrix (W_b). Therefore, attaining the lowest condition number for the matrix W_b will lead to better excitation and better least squares estimates. Therefore, the trajectory optimization criterion mentioned in Subsection 3-3-2 can be used for the identification only if the observation matrix is well equilibrated. In some cases, smaller link parameters are difficult to identify as they have less influence on robot dynamics and can eventually lead to an ill-conditioned W_b matrix. This problem is addressed in [17] by incorporating a new criterion for trajectory optimization. G. Venture used two optimization criterion to generate the excitation trajectory with low condition number. The corresponding objective functions are depicted in Equations (3-11), (3-12), where the first criterion excites the regressor matrix built with static parameters like the masses and center of mass of each link (W_{bi}^S) separately and the second criterion excites the regressor matrix built with the dynamic parameters like the inertia of each link (W_{bi}^D) separately. The dynamic postures were exclusively obtained by optimal excitation between the static postures. Subsequently, the obtained joint configuration for static postures q_S and dynamic postures q_D was interpolated by B-splines techniques.

$$q_S = \min_{q_S} \sum_{i=1}^{N} M_i \text{cond}(\bar{W}_{bi}^S) \tag{3-11}$$

$$q_D = \min_{q_D} \sum_{i=1}^{N} M_i \text{cond}(\bar{W}_{bi}^D) \tag{3-12}$$

In this thesis, the idea of dividing the total observation matrix (W_b) into sub-regressors containing the information of individual links was incorporated. Unlike [17], the sub-regressors are not further split into separate regressor containing static and dynamic parameter information. In [17], the condition number is multiplied with the link mass (M_i) to give more importance to the larger links. Whereas in this thesis, the weights are taken to be unity to give equal importance to all the links and to attain better excitation for all links. By doing this, a lower condition number can be obtained as compared to the criterion specified in Subsection 3-3-2. The proposed minimization criteria is given as follows,

$$J = \min_{a_{i,l}, b_{i,l}} \sum_{i=1}^{N} \text{cond}(W_{bi}), \tag{3-13}$$

where the $a_{i,l}, b_{i,l}$ denotes the amplitude values of the Fourier series, W_{bi} denotes the sub-regressor for each link i, J is the objective function and N represent the number of links.

3-5 Discussions

The chapter started with the discussion of previous research works related to the trajectory parametrization and optimization. It laid out the reasons for incorporating Fourier series as

excitation trajectory. In addition, it proposed a new trajectory to capture the un-modeled dynamics by introducing frequency dependent signals. Later, the chapter introduced a modified objective function for trajectory optimization to treat the aspect of numerical issues arising during the least squares identification. Following this chapter, the generated trajectory will be used for the least squares identification and training the GPR. The importance of low condition number in excitation trajectories highlighted in this chapter can be witnessed in the upcoming chapters.

Chapter 4

Identification

Chapter 2 modeled the serial manipulator into a parametric and a non-parametric equivalent. Now, this chapter ponders on the methods to identify these two terms with the excitation trajectory that is generated from Chapter 3. The non-parametric Gaussian process regression learns the entire Inverse Dynamics Model (IDM) from scratch [65] [71]. It does not use of any prior knowledge of the robot. This kind of regression techniques has several drawbacks. First, it requires a huge amount of data to learn the accurate IDM of the robot. Second, the non-parametric GPR model will fail to generalize for any unknown data, if the excitation trajectory is not sufficiently exciting or if the data set is poor with less information [74] [65]. In this chapter, the strengths of non-parametric and parametric model learning were combined to obtain a semi-parametric regression framework. First, the parametric term is identified by the least squares technique. Second, the non-parametric term is identified using the Gaussian Process Regression (GPR) with the obtained parametric term as mean. Unlike the non-parametric GPR, learning the entire IDM of the robot is avoided and the learning is done to captures only the un-modeled dynamics and other non-linearities. Therefore, the Semi-parametric GPR (SGPR) has better generalization as compared to the non-parametric GPR as SGPR uses the RBD model of the robot for any unknown data.

The remainder of this chapter will be organized as follows, Section 4-1 presents the previous research works on the identification of the inverse dynamics model. It points out the advantages and the disadvantages of each method and paves the way for other sections to reason the methods incorporated in this project. Section 4-2 ponders on the Weighted Least Squares and the Least squares technique for identifying the rigid body dynamics of the robot. In addition, it points out the different class of perturbations occurring in the manipulator in detail. Section 4-3 incorporates the GPR to the capture the un-modeled dynamics occurring in the manipulator. Section 4-4 proposes a new approach based on the method discussed in Section 4-3 to address the problem of inaccuracies occurring in the internal inverse dynamics model. Lastly, a discussion is given about the proposed methods in Section 4-5.

4-1 Previous Works

In the domain of linear identification of the robot's RBD model, the most popular technique is the Weighted Least Squares (WLS)/Least Squares (LS) approach. It is an offline non-iterative method, which was successfully tested on a 7 DOF KUKA LWR 4 robot in [32], [34], 7 DOF KUKA LBR iiwa R820 in [49], 7-DOF Mitsubishi PA10 in [35], [33], 6-DOF PUMA 560 in [37], 6-DOF Stäubli TX-90 in [41] and a 3-DOF KUKA IR 361 in [36]. Unlike WLS/LS, W. Wu incorporated a statistical approach named Maximum Likelihood Estimation (MLE) to estimate the base parameters. It was proved to work successfully on the first three axes of QIANJIANG-I 6-DOF robot [24]. Irrespective of the measurement noise, this method yields unbiased estimates with minimum uncertainty. Meanwhile, it is a non-linear optimization approach and requires the initial guess of parameters to converge to a minimum. In addition, the convergence of global optimum cannot be guaranteed. The Extended Kalman Filter is an another statistical framework approach. It is an online recursive estimation technique, which was successfully adopted for the 2-DOF SCARA robot in [30] and [31] and the 5 DOF KUKA youBot in simulation in [48]. Since the system to be identified is non-linear, this approach proves to be useful as it linearizes the non-linear model using the Taylor expansion at every instant. The main disadvantage of this method is that it is sensitive with respect to initial values. So, a prior knowledge is required to obtain a better estimation of the base parameters. Also, it is unlikely for the zero initial values to yield good results as in some cases the CAD data is unavailable. In addition, the calculation of the Jacobian matrix (used in Taylor expansion) for the extended state is computationally heavy and consumes a lot of time [31].

The aforementioned works use only the Inverse Dynamic Model (IDM) for the base parameter estimation. M. Gautier and A. Janot in 2008 proposed a strategy that employs both the direct dynamic model and the inverse dynamic model to avoid the numerical computation of $\ddot{q}$, $\dot{q}$ by utilizing only torque data for dynamic model identification [27], [26]. The main disadvantage of this method is that it requires the control law used by the robot controller for the direct dynamic model, which is often a black-box.

A. Janot and P.O. Vandanjon dealt with noisy observation matrix by using the instrument variable approach to estimate the base parameters of the 6-DOF Stäubli TX-40 robot, which is a combination of the direct and the inverse dynamic model of the robot. The main advantage of this approach is that it yields an unbiased estimate by dealing with noisy observation matrix W and has faster convergence as compared to MLE. Janot compared the obtained estimates from the IV approach with the results of a classical LS approach. According to their results, the IV approach does not lead to better estimations than the Least Squares formulation. In addition, the direct dynamic model is built from the approximation of the noise-free model of the physical process to be identified, which is often unknown.

In 2013, M. Gautier introduced a new method named power model which expresses the base parameters in terms of total energy and the power of the robot system. This method was adopted to avoid the tedious symbolic computation of the observation matrix used in the IDM model [29]. Gautier also compared the estimation results from the power model approach with the IDM - LS approach. It was inferred that the estimated values were closer to IDM-LS estimates but did not improve IDM -LS.

Some robots give the opportunity for the end-users to calculate the link inertial matrix and

gravity vector at the current robot configuration through its Fast Research Interface (FRI). C. Gaz exploited this advantage and introduced a reverse engineering approach to estimate the inverse dynamics model of the robot. Reverse engineering approach was successfully applied on a 7-DOF KUKA LWR IV robot for all joints in [28]. The main disadvantage of this method is that it is not applicable for the robots where the inertial and gravity matrix at every robot configuration is not measurable.

In the domain of machine learning, learning of IDM can be broadly classified into offline and online techniques. The most popular offline estimation techniques are Support Vector Regression (SVR) and Gaussian Process Regression. In [71], the support vector regression was successfully implemented for learning the entire IDM of the 7 DOF SARCOS robot. On the contrary, D.N. Tuong uses the GPR to capture only the un-modeled dynamics occurring in the 7 DOF Barrett WAM robot [65]. In addition, D.N. Tuong introduces different methods to use the prior information, i.e., CAD data of the robot. The prior knowledge was used as mean and kernel for capturing the un-modeled dynamics. Both GPR and SVR are computationally heavy and a slow process. This kind of learning can be adopted when the computational effort is not a big priority.

Meanwhile, online techniques help in capturing the time-varying un-modeled dynamics. Thereby, assisting the robot controller to adapt itself for time-varying non-linearities. The popular methods are Local Gaussian Process Regression and Locally Weighted Projection Regression (LWPR). These methods were tested successfully on the 7 DOF SARCOS master arm and 7 DOF Barrett WAM robot in [68]. The major disadvantages faced by these methods are given as follows,

- Achieving inverse dynamics learning and torque prediction at a rate greater than 1000 Hz ($<$ 1ms) will be difficult.
- Even if the above said is achieved. There will be more data incoming, for e.g., for the sampling rate of 1 ms, 600000 data points will be collected for 10 minutes. Without proper handling, the online technique will be difficult.
- The model has to be adapted continuously for new data points.
- LWPR and LGP are not better than offline Gaussian Process Regression (GPR) and Support Vector Regression (SVR) in terms of accuracy [68].

4-2 Inverse Dynamic Model - Least Squares / Weighted Least Squares (IDM LS/WLS)

From Section 4-1, it can be inferred that every methods has its own advantages and disadvantages. Unlike other methods,the Least Squares approach (LS) is less complex as it estimates the parameters in a single-step. It is better compared to other methods mentioned in Section 4-1 and in addition, it does not require any prior information about the estimates beforehand. So, LS method is preferred for the identification of Rigid Body Dynamics (RBD) of the manipulator in this thesis. Generally, the IDM of a manipulator is given as

$$\tau_{\text{IDM}} = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q), \tag{4-1}$$

where τ_{IDM} is the joint torque of the IDM model. The actual torque τ varies from τ_{IDM} by an error ρ as depicted in Equation (4-3). The error ρ witnessed in robotic manipulators can be classified as follows,

- Measurement noise ($\tilde{\phi}_b$) - It occurs due to the sensor noise and numerical noise arising from the numerical differentiation of the joint positions to obtain joint velocities and accelerations.
- Motion noise ($\tilde{\tau}$) - These are the uncertainties acting on the joint torques. It occurs due to complex friction and random noise acting on the robot.
- Systematic error (ν) accounts for the un-modeled dynamics of the robot like flexibilities, dynamics of hydraulic tubes, actuators, and cable drives etc.

The aforementioned noises are represented in the following equations

$$\tau = (\phi_b + \tilde{\phi}_b)\chi_b + \tilde{\tau} + \nu, \tag{4-2}$$

$$\tau = \tau_{\text{IDM}} + \rho = \phi_b(q, \dot{q}, \ddot{q})\chi_b + \rho, \tag{4-3}$$

where ϕ_b is the observation matrix, χ_b is the minimum set of inertial parameters or the base parameters of the robot and ρ is the total perturbation on the joint torques. The observation matrix (ϕ_b) is an upper triangular matrix constructed with the columns of the base parameters [11]. For p measurements, the observation matrix and the torque vector is constructed as follows

$$W_b = \begin{bmatrix} \phi_b(q(t_1), \dot{q}(t_1), \ddot{q}(t_1))_{n\times m} \\ \phi_b(q(t_2), \dot{q}(t_2), \ddot{q}(t_2))_{n\times m} \\ \vdots \\ \phi_b(q(t_p), \dot{q}(t_p), \ddot{q}(t_p))_{n\times m} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \tau(t_1)^T \\ \tau(t_2)^T \\ \vdots \\ \tau(t_p)^T \end{bmatrix}, \tag{4-4}$$

where m is the number of base parameters, n is the DOF of the robot, Γ is the vector of torque signals for p measurements and W_b is the observation matrix corresponding to the base parameters χ_b constructed for p measurements. By using Equation (4-4), the Weighted Least Squares (WLS) solution to Equation (4-3) can be given as

$$\hat{\chi}_b = \min_{\chi_b} \|\rho\|_2^2 = (W_b^T \Sigma^{-1} W_b)^{-1} W_b^T \Sigma^{-1} \Gamma, \tag{4-5}$$

$$| \ \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2), \tag{4-6}$$

where Σ is the diagonal co-variance and σ_i^2 is the torque co-variance of the joint i. Since the chosen excitation trajectory for identification is periodic (See Chapter 3), the variances of the measurement can be calculated by recording multiple periods ($k = 1, 2, \ldots, K$) and averaging them over K periods. The mean and co-variance of the measured data can be given by the following equations

$$\bar{x}_i(m) = \frac{1}{K} \sum_{k=1}^{K} x_{k,i}(m), \tag{4-7}$$

$$\sigma_i^2 = \frac{1}{K} \sum_{k=1}^{K} (x_{k,i}(m) - \bar{x}_i(m))(x_{k,i}(m) - \bar{x}_i(m))^T, \tag{4-8}$$

where x can be joint positions or joint torques, $\bar{x}_i(m)$ is the mean, σ_i^2 is the co-variance, m is the number of observation points (samples) and i runs from 1 to n for n links [49]. The WLS can be converted to LS problem by dropping the scaling with inverse of torque co-variance by considering constant variance in the torque error and the equation pertinent to it is given by

$$\Sigma = I, \tag{4-9}$$

$$\hat{\chi}_b = \min_{\chi_b} \|\rho\|_2^2 = (W_b^T W_b)^{-1} W_b^T \Gamma. \tag{4-10}$$

Like other methods, Least Squares has its own weakness, it yields a biased estimate, if W_b is correlated with ρ. One way to tackle the issue is by using a proper filter before computing the LS solution. Since, the method cannot captures complex friction and unknown dynamics. The report addresses the issue by using the Gaussian Process Regression with the obtained LS estimate as mean which will be further discussed in the upcoming sections.

4-3 Semi-Parametric Gaussian Process Regression (GPR)

The offline machine learning technique is adopted in this thesis as it is more suitable for model-based compensation due to the disadvantages listed for online learning techniques in Section 4-1. But, the GPR has an upper hand as the hyperparameters of the kernel function can be obtained easily by maximizing the log-likelihood. Furthermore, GPR gives the property of combining the automatic feature selection with learning using the ARD kernel. Unlike SVR, GPR can handle uncertainties in the unknown function f by averaging, not minimization. Due to these reasons, GPR is chosen over SVR. In this project, the GPR is not used to learn the entire IDM of the robot. Instead, the GPR is used to capture the un-modeled dynamics and other non-linearities. Therefore, this section ponders deeply on how to incorporate the prior information obtained from least squares as mean to do semi-parametric Gaussian process regression.

The GPR framework discussed in Appendix B can be easily extended for inverse dynamics learning of manipulators. From Chapter 2, the inverse dynamics model can be given as follows

$$\tau = W_b \chi_b + \epsilon(q, \dot{q}, \ddot{q}), \tag{4-11}$$

where $\epsilon(q, \dot{q}, \ddot{q})$ captures the non-linearities occurring in the robot. Using the GPR technique, the robot dynamics from Equation (4-11) can be modelled as

$$\tau \sim \mathcal{GP}(m(x), k(x, x^T)), \tag{4-12}$$

where $x = [q, \dot{q}, \ddot{q}]^T$ is the function of joint position, velocity and acceleration, $m(x)$ is the mean function and $k(x, x^T)$ is the covariance or kernel function. The regression becomes non-parametric if no prior knowledge is used or the mean is assumed to be zero. This can be visualized in Equation (4-13).

$$\tau \sim \mathcal{GP}(0, k(x, x^T)). \tag{4-13}$$

The main concern with this approach is that the parameters are learned from scratch without any use of prior information of the robot. In [65], a comparison was made between non-parametric GPR and GPR with prior knowledge for a 7 DOF real Barrett WAM robot. It was found that the non-parametric GPR had poor tracking performance as compared to the

GPR with RBD as mean. So, the RBD of the manipulator obtained from the least squares technique in Section 4-2 is used as mean for the GPR to make the non-parametric regression to a semi-parametric regression. Subsequently, the Equation (4-13) is converted to

$$\tau_n \sim \mathcal{GP}(W_b\chi_b, k(x, x^T)), \tag{4-14}$$
$$\tau_n \sim W_b\chi_b + \mathcal{GP}(0, k(x, x^T)). \tag{4-15}$$

By incorporating the fixed mean as represented in Equation (4-15), the Gaussian process model is biased towards this information. So, whenever the GPR encounters an unseen data, it uses the RBD model of the manipulator. The kernel function k in Equation (4-15) gives the similarity measure between two points X and x_*. Exponential, squared exponential, Matern 3/2, Matern 5/2, rational quadratic etc. are some common Kernel functions used for GPR. One example of a kernel function is given below

$$k_*(X, x_*) = \sigma_f^2 \exp\Big(-\frac{1}{2}\frac{(X - x_*)^T(X - x_*)}{\sigma_l^2}\Big), \tag{4-16}$$

where $k_*(X, x_*)$ is the squared exponential kernel, σ_l is the width of the kernel and σ_f is the input signal standard deviation. Similarly, the torque prediction obtained by using RBD as mean for n DOF can be given as follows

$$\bar{\tau}_n = w(x_*^T)\chi_b + k_*^T(K + \sigma_n^2 I)^{-1} \underbrace{(\tau_n - W_b(X)\chi_b)}_{\text{error}}, \tag{4-17}$$

where $\bar{\tau}_n$ is the torque prediction obtained for input query point x_*, w is the observation matrix estimated for the query point x_*, W_b is the observation matrix obtained using training data X, k_* is the covariance vector estimated on training data set X and the query point x_*, K is the covariance matrix obtained from training input data X, and τ_n is the filtered torque signal for the joint n. The error term vanishes from Equation (4-17), if RBD correctly portrays the robotic manipulator. In addition, the error term vanishes even if X is much greater than x_* (if x_* is far away from X).

The major limitation faced by GPR is that it takes a lot of computation time during the training and prediction step. During training, evaluating the inversion of the kernel matrix $(K(X, X))$ of size $n \times n$ and computing $\log(P(\tau|X))$ is $O(n^3)$ respectively. Therefore, the complexity becomes $O(kn^3)$ for a k number of function evaluations for maximization and n number of observations. Since the training of Gaussian process regression is done offline, the computational complexity is of minor significance. On the other hand, the prediction is done online with the trained GPR. So, the complexity is an important factor to be considered. The prediction step depicted in Equation (4-17) has computational complexity of $O(n^3)$ for the evaluation of the term $(K + \sigma_n^2 I)^{-1}$. But, this term can be stored and it is not required to be computed every time. The memory requirement of the aforementioned term scales to only $O(n^2)$. The issue can also be addressed by not predicting torque signals at every time instant but with certain time intervals.

4-4 New approach - Learning the Model Inaccuracies through Semi-parametric GPR

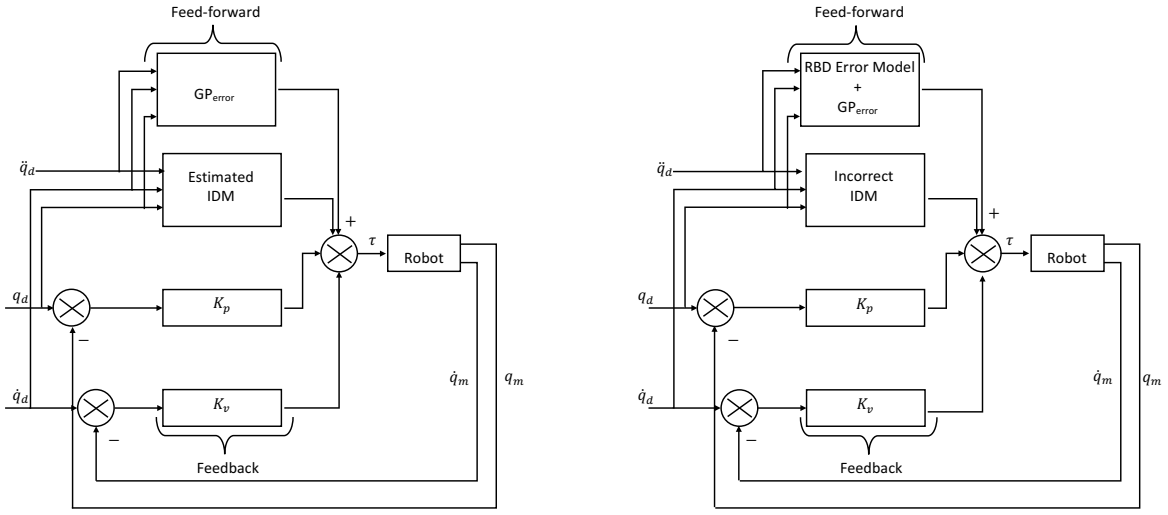

Figure 4-1: Left - Compensation of un-modeled dynamics using the Semi-parametric GPR (S-GPR) for Case 1, Right - Compensation of model inaccuracies and un-modeled dynamics using the Semi-parametric GPR (S-GPR) for Case 2.

The difference between the research carried out in [65] and in this thesis can be visualized in Figure 4-1. The first case depicted in Section 1-3 can be tackled using the approach suggested in Section 4-3 (similar to the work done in [65]). In the previous section, first, the internal IDM which failed to compensate for the un-modeled dynamics is replaced by the RBD model estimated from the least squares. Subsequently, feed-forward signals from Gaussian process model are given to the robot to compensate for the non-linearities that cannot be explained by the estimated RBD model. Whereas in this section, the same technique is used to capture the model inaccuracies and un-modeled dynamics occurring in the internal inverse dynamics model (refer Figure 4-1). Further, it is assumed in this section, that it is not possible to replace or turn OFF the internal IDM controller of the robot. So, it can be inferred from Figure 4-1, that the feed-forward torque signals from the semi-parametric model, are given in the presence of the inaccurate IDM of the robot. This section will now ponder deeply on the method to acquire the semi-parametric model for this special case.

At low stiffness and damping conditions, the robot becomes completely unstable as the internal inverse dynamics is inherently unstable. This demands for proper feed-forward signals to compensate for the inaccuracies in IDM and the un-modeled dynamics. Generally, in these case, the high PD gains are set to make the robot rigid and stable. Subsequently, the identification experiment is carried out. The high feedback gain will compensate for the inaccuracies in the internal IDM and un-modeled dynamics. Hence, the total torque given to the robot to track the desired trajectory is given as

$$\tau = \tau_{\text{PD}} + \tau_{\text{I-IDM}}, \tag{4-18}$$

where τ is the total joint torque given to the robot, τ_{PD} is the feedback toque and $\tau_{\text{I-IDM}}$ is the torque due to the Inaccurate internal Inverse Dynamics Model (I-IDM). The feedback torque can be measured using the relation

$$\tau_{\text{PD}} = K_p(q_d - q_m) + K_d(\dot{q}_d - \dot{q}_m), \tag{4-19}$$

where K_p and K_d are the proportional and derivative gains respectively, q_d and $\dot{q}_d$ are the desired joint position and velocity respectively. In Equation (5-7), the parameters K_p, K_d, q_m, $\dot{q}_m$, q_d, $\dot{q}_d$ are all measurable, thereby making the τ_{PD} measurable. With the measured feedback torque as input and measured joint position, velocity and acceleration as output, the least square is incorporated to estimate the error in the base parameters (χ_{error}). Subsequently with the obtained error model as mean, the Gaussian process regression is used to capture the remaining un-modeled dynamics and other complex perturbations. The semi-parametric Gaussian process regression framework for learning the inaccuracies and other perturbations can be given as,

$$\tau_{\text{PD}} \sim W_b(q_d, \dot{q}_d, \ddot{q}_d)\chi_{\text{error}} + \mathcal{GP}(0, k(x, x^T)), \tag{4-20}$$

where $x = [q, \dot{q}, \ddot{q}]^T$ is the function of joint position, velocity and acceleration, $W_b(q_d, \dot{q}_d, \ddot{q}_d)\chi_{\text{error}}$ is the mean function and $k(x, x^T)$ is the covariance or kernel function. The corresponding torque prediction for new query point x_* is given by the following relation

$$\bar{\tau}_n = w(x_*^T)(\chi_{\text{inc}} + \chi_{\text{error}}) + k_*^T(K + \sigma_n^2 I)^{-1} \underbrace{(\tau_{\text{PD}} - W_b(X)\chi_{\text{error}})}_{\text{error}}, \tag{4-21}$$

where χ_{inc} is the incorrect base parameters, $\bar{\tau}_n$ is the torque prediction obtained for input query point x_*, w is the observation matrix estimated for the query point x_*, W_b is the observation matrix obtained using training data X, k_* is the covariance vector estimated on training data set X and the query point x_*, K is the covariance matrix obtained from training input data X, and τ_{PD} is the measured feedback torque signal. The error term vanishes from Equation (4-21) when no complex friction and other non-linearities act on the robot and if the RBD error model is able to compensate for the inaccuracies in the internal IDM. In addition, the error term also vanishes even if the X is much greater than x_* (if x_* is far away from X). In this case, the semi-parametric model will compensate for the inaccuracies in the internal RBD model but will fail to capture the un-modeled dynamics and other perturbations due to an unseen data.

The feed-forward signals are given to the robot in the presence of incorrect internal IDM after obtaining the semi-parametric model. These feed-forward signals are the combinations of torque signals from the RBD error and Gaussian process models for a given joint position, velocity, and acceleration as shown in Figure 4-1. This makes the robot stable at low compliant conditions and to become less dependent on feedback.

4-5 Discussions

In this chapter, the previous works related to inverse dynamics learning were discussed. It laid out the advantages and the disadvantages of each method. In addition, it pointed out the need to adopt the least squares and GPR towards IDM learning. In short, the rigid body dynamics was estimated using the least squares method and it was used as mean to

capture the perturbations through GPR. By doing this, the chapter highlighted the fact that SGPR will never fail to generalize an unknown data as the used RBD mean gives the unique relationship between joint torques and joint position, velocity and acceleration. Later, the chapter proposed a new technique to address the inaccuracies occurring in the internal IDM by incorporating the semi-parametric approach. Following this chapter, the identification methods will be tested on the PUMA-560 and a twolink manipulator.

Chapter 5

Evaluations

In this thesis, the robot is compensated by a PD + inverse dynamics model as depicted in Figure 1-1. The feedback term becomes less at low stiffness and damping conditions. As a result, the manipulator becomes entirely dependent on the internal inverse dynamics model. Therefore to reduce the dependency on feedback and to achieve good performance at low compliant conditions, learning accurate inverse dynamics model becomes crucial. Hence, two different cases were considered in this chapter. For the first case, the rigid body dynamics model with the correct base parameters is used for controlling the robot. Whereas in the second case, the rigid body dynamics model was deliberately made inaccurate by including an offset in the base parameters. However, the rigid body dynamics failed to capture the un-modeled dynamics and the complex friction in both the cases. The modeling of the robots was done using Peter Corke's - Robotic Toolbox. The GPR and trajectory optimization was done using the machine learning and optimization toolbox respectively provided by MATLAB. Furthermore, the columns of the observation matrix (W) were generated using a python based open software named Sympybotics. The procedure to acquire the matrix using the software is illustrated in the Appendix A.

The remainder of this chapter is structured as follows, Section 5-1 discusses modeling of the PUMA 560 and the twolink manipulator in MATLAB. The chosen values for the inertia, mass, Center of Mass (CoM), and friction for the manipulators were reported. Furthermore, it talks about the considered noise model in the PUMA 560 and the twolink manipulator. Later in this section, the control scheme of the robot is discussed and it also points out the reasons to use the particular scheme. In Section 5-2, a reference excitation trajectory is generated with the help of the constraint designed for the PUMA 560 and the twolink manipulator. Section 5-3 talks about the identification of the rigid body dynamics of the twolink and the PUMA 560 manipulator with the least squares approach. Further, it concentrates on capturing the un-modeled dynamics and other perturbation using Gaussian process regression. Lastly, in Section 5-4, the second case is solved by first learning the offset in the internal RBD model through the LS approach. Subsequently, with the learned error model as mean, GPR is used to capture the un-modeled dynamics and other perturbations.

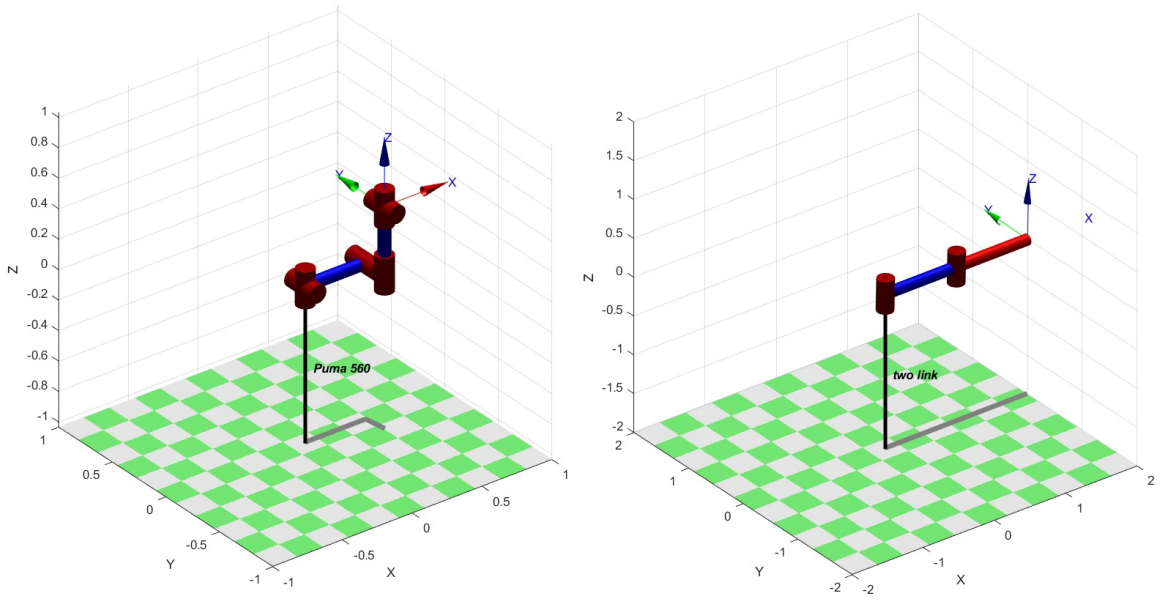

Figure 5-1: Left - PUMA 560 Manipulator, Right - Twolink Manipulator

Table 5-1: Dynamic and electrical parameters of the PUMA 560 robot

DOF	Mass $[kg]$	COM $[m]$	I $[kgm^2]$	J_m	B_m	T_c	G
1	0	[0,0,0]	diag([0,0.35,0])	0.0002	0.00148	[0.395 -0.435]	-62.61
2	17.4	[-0.3638,0.006,0.2275]	diag([0.13,0.524,0.539])	0.0002	0.000817	[0.126 -0.071]	107.8
3	4.8	[-0.0203,-0.0141,0.07]	diag([0.066,0.086,0.0125])	0.0002	0.00138	[0.132 -0.105]	-53.79
4	0.82	[0,0.0019,0]	diag([0.0018,0.0013,0.0018])	3.3×10^{-5}	7.12×10^{-5}	[0.0112 -0.0169]	76.04
5	0.34	[0,0,0]	diag([0.0003,0.0004,0.0003])	3.3×10^{-5}	8.26×10^{-5}	[0.00926 -0.0145]	71.92
6	0.09	[0,0,0.032],	diag([0.00015,0.00015,4×10^{-5})	3.3×10^{-5}	3.67×10^{-5}	[0.00396 -0.0105]	76.69

5-1 Simulation Setup

In this thesis, all the proposed methods were tested on the PUMA 560 and the twolink manipulator. The visual representation of these manipulators is depicted in Figure 5-1. The Unimate Puma 560 is a 6 DOF robot. The kinematic, dynamic and electrical parameters for the PUMA 560 robot have been compared and reported in the literature [79]. The thesis incorporates these parameter values for the PUMA 560 robot. The twolink manipulator has two links with equal length. The kinematic, dynamic and electrical parameters for the twolink manipulator were chosen from [50], where the experiment was carried out on a two-link IMI manipulator. The dynamic and electrical parameters for the PUMA 560 and the twolink manipulator can be found in the Tables 5-1, 5-2 respectively. Meanwhile, the kinematic model for the PUMA 560 and the twolink manipulator can be found in Tables A-1, A-2, where the model is represented using the Denavit Hartenberg convention. Furthermore, the methods to obtain the reduced model set for identification can be found in Appendix A. The remainder of this section is structured as follows. In subsection 5-1-1, the different perturbations on the PUMA 560 and the twolink manipulator are discussed. Subsequently, Subsection 5-1-2 discusses the control structure of the robot that is incorporated in this thesis. In addition, it ponders on the reasons to incorporate the particular control structure.

Table 5-2: Dynamic and electrical parameters of the Twolink Manipulator

DOF	Mass $[kg]$	COM $[m]$	I $[kgm^2]$	J_m	B_m	T_c	G
1	1	[-0.5,0,0]	diag([0,0,0])	0.0002	0.0005	[0.1 -0.1]	-20.61
2	1	[-0.5,0,0]	diag([0,0,0])	0.0002	0.00025	[0.05 -0.03]	20.82

Table 5-3: The definition of parameters used in Table 5-1 and 5-2

COM $[m]$	Center Of Mass link with respect to link coordinate frame.
I $[kgm^2]$	Moment of Inertia with respect to link COM
J_m	Motor Inertia
B_m	Link Viscous Friction
T_c	Link Coulomb Friction
G	Gear Ratios

5-1-1 Noise Model

As discussed in Chapter 4, there are three major perturbations occurring in the manipulators namely, measurement noise, motion noise, and systematic noise. Even-though the Fourier series gives the property of analytical and frequency domain differentiation of the joint positions. The joint velocities and accelerations are obtained by numerical differentiation of joint positions. By doing this, the measurement noise can be deliberately included into the model. The joint position, velocity, and acceleration will be crucial in constructing the matrix W_b, which is further utilized in trajectory optimization and least squares identification. The error due to numerical differentiation of the joint position to obtain joint acceleration can be visualized in Figure 5-2.

Peter Corke's toolbox provides the end-users the possibility to set the frictional parameters of the robots. Therefore, the motion noise can be introduced in these robots by including the Coulomb and viscous friction. The friction parameters for the twolink and the PUMA560 manipulator are depicted in Tables 5-2 and 5-1 respectively. The frictional forces vary with respect to the direction of motion. The - or + shows the negative and positive direction of rotation of the manipulator. Apart from Coulomb and viscous friction, the aforementioned manipulators exhibit stiction and Stribeck at low-velocity regions [80].

Peter Corke's toolbox provides a linear inverse dynamics model for the two-link and the PUMA 560 robot for control purposes. Furthermore, it gives the end-users two options for choosing the IDM of the robot. One with compensation for viscous and Coulomb friction (modeled as given in Equations (2-3), (2-5)) and other without compensation for viscous and Coulomb friction (modeled as given in Equations (5-3), (5-4)) [80]. When the IDM with viscous and Coulomb compensation, was used for controlling the PUMA 560 robot, the measured joint position was unable to track the desired joint positions at zero PD gains as depicted in Figure 5-3. This signifies that the parametric term modeled with Equations (2-3), (2-5) have un-modeled dynamics. One of the un-modeled dynamics includes motor inertia of the PUMA 560 and the two-link manipulator. In this way, the systematic error can be implemented, for the upcoming experiments.

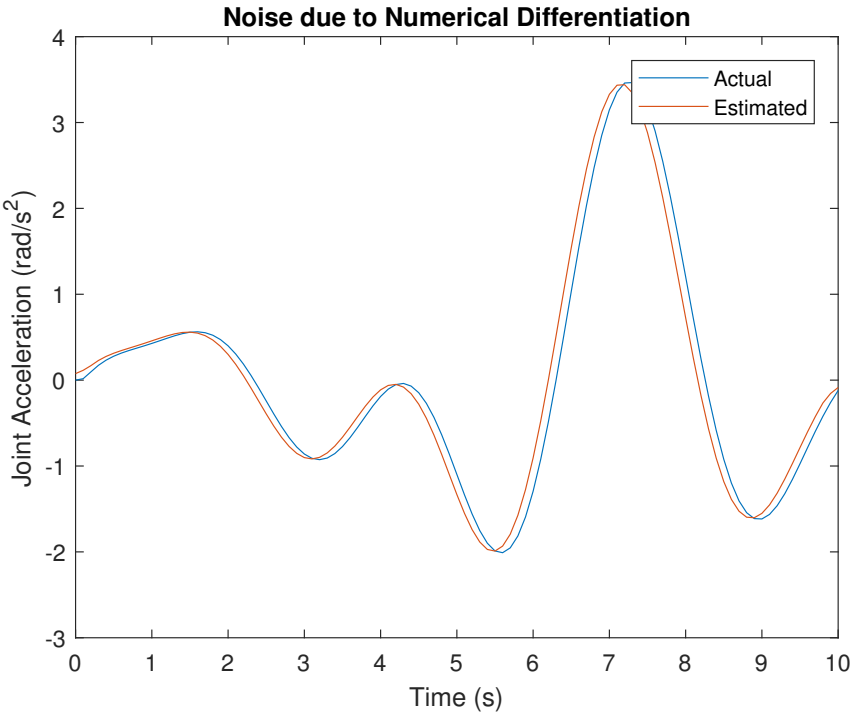

Figure 5-2: Comparison of actual joint acceleration and joint acceleration obtained through numerical differentiation of joint position for the 1st DOF PUMA 560 robot

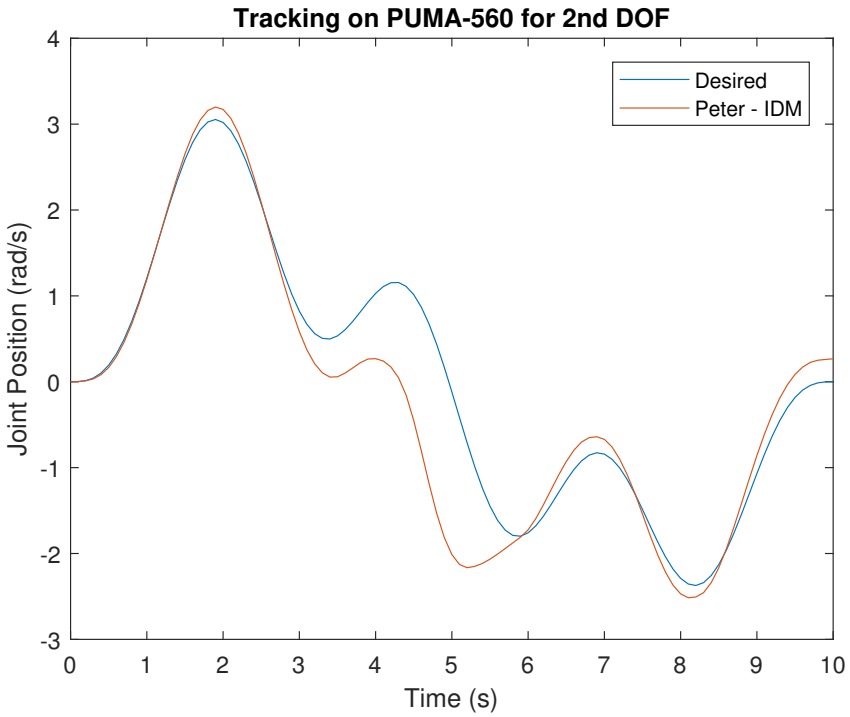

Figure 5-3: Tracking Performance of the PUMA 560 robot for 2nd DOF with the control scheme depicted in Figure 1-1, where the PD gains are set to zero

Table 5-4: Physical limits of the twolink manipulator ($q_{\text{i,max}}$ $\dot{q}_{\text{i,max}}$ $\ddot{q}_{\text{i,max}}$ are the maximum joint position, velocity, and acceleration)

Joint	$q_{\text{i,max}}$	$\dot{q}_{\text{i,max}}$	$\ddot{q}_{\text{i,max}}$
1	1.5708	2	2
2	1.5708	2	2

5-1-2 Nonlinear Feed-Forward Controller

The nonlinear feed-forward controller has the control scheme similar to the one depicted in Figure 1-1. It is found to be efficient in tracking high velocity trajectories with high accuracy as compared to PID and PD + gravity controller. The controller does this by cancelling the non-linearities in the robot dynamics [11]. On the other hand, the popular passivity based controller cannot be implemented as the motor torques and motor positions are not measurable in Peter Corke's robotic toolbox. Therefore, non-linear feed-forward controller is incorporated in this project. The control diagram is presented in Figure 1-1 and the control equation pertaining to it, can be described as follows

$$u = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d) + G(q_d) + K_p(q_d - q_m) + K_d(\dot{q}_d - \dot{q}_m), \tag{5-1}$$

where q_d, $\dot{q}_d$, $\ddot{q}_d$ are the desired joint position, velocity and acceleration, q_m, $\dot{q}_m$ are the measured joint position and velocity. Combining the rigid body dynamics from Chapter 4 (Equation (4-1)) with Equation (5-1), the closed error dynamics of the non-linear feedback controller is given as

$$\ddot{e} + M^{-1}K_v\dot{e} + M^{-1}K_p e = 0, \tag{5-2}$$

where $e = q_d - q$, $\dot{e} = \dot{q}_d - \dot{q}$, $\ddot{e} = \ddot{q}_d - \ddot{q}$, K_p is the proportional gain and K_v is the derivative gain. The feedback gains are chosen in such a way that the error Equation (5-2) is stable [65].

Like other controller, the mentioned control scheme has its own disadvantages. The control structure is highly dependent on the inverse dynamics model of the manipulator. So, the robot becomes unstable if the inertia matrix $M(q)$, centrifugal matrix $C(q, \dot{q})$ is imperfect. Even though the uncertainties in the inverse dynamics model can be addressed by setting high gains. The situation is not always applicable as some robots work under low compliant conditions as pointed out in the Chapter 1. Therefore, the controller will be improved by adding an extra feed-forward term to have less dependence on feed-back term and to compensate for the imperfectness in the inverse dynamics model. The extra feed-forward term will be estimated using GPR, which will be discussed in the upcoming sections.

5-2 Excitation Trajectory

As discussed in Chapter 3, the Fourier Series is used as the excitation trajectory for the identification of rigid body dynamics and training the Gaussian process regression. The number of Fourier terms N is taken to be 4 and 3 for the PUMA 560 and the twolink manipulator respectively. The fundamental frequency for the Fourier series is set to 0.1 Hz to get a period of excitation for 10s. With these settings, the excitation trajectory covers a frequency range of [0.1 Hz - 0.4 Hz] for the PUMA 560 robot and [0.1 Hz - 0.3 Hz] for

Table 5-5: Physical limits of the PUMA-560 robot ($q_{\text{i,max}}$ $\dot{q}_{\text{i,max}}$ $\ddot{q}_{\text{i,max}}$ are the maximum joint position, velocity, and acceleration)

Joint	$q_{\text{i,max}}$	$\dot{q}_{\text{i,max}}$	$\ddot{q}_{\text{i,max}}$
1	2.79	5	6
2	3.92	5	6
3	0.78	5	6
4	2.96	5	6
5	1.74	5	6
6	4.64	5	6

the twolink manipulator. The maximum joint position, velocity and acceleration limits of all the joints of the PUMA-560 and the twolink manipulator are depicted in the Tables 5-5, 5-4. These tables will help in building up the constraints mentioned in Equations (3-7), (3-8) and (3-9). The trajectory optimization is a nonlinear and a non-convex constrained optimization problem, which can be solved by fmincon solver in MATLAB. Interior-point, trust-reflective region, Sequential Quadratic Programming (SQP), SQP-legacy, and active set are the methods provided by the fmincon solver to the end-users. For PUMA-560, there are 18 linear equality and nonlinear equality constraints and has 48 trajectory parameters to be solved. Meanwhile, the twolink has 6 equality and nonlinear equality constraints and has 12 trajectory parameters. Therefore, the algorithm has to selected according to their ability to solve the complex large-scale optimization problem.

SQP, SQP-legacy and active set algorithms are avoided as it would require longer execution time and significant amount of memory to store the matrices. On the other hand, trust-reflective region can be used for large-scale optimization problem but it requires the information of gradient. Unfortunately, the gradient information of the objective function is not available. Therefore, the parameters $a_{i,l}$ and $b_{i,l}$ are obtained by solving the optimization problem with the interior-point algorithm.

The parameters of the Fourier series obtained after optimizing the condition number of the observation matrix (W_b) is depicted in the Tables C-1 and C-2. These estimated values are used to generate the Fourier series. The generated trajectories for the PUMA 560 robot and the two-link manipulator are depicted in the Figures 5-4 and 5-5 respectively. From Figures 5-4 and 5-5, it can be observed that trajectories obey the robot physical constraints specified in Tables 5-5 and 5-4. Furthermore, the generated trajectories also obey the second set of constraints (refer Equations (3-4), (3-5) and (3-6)) as the initial and final joint positions were found to be zero.

5-3 Case 1 : Semi-parametric Gaussian Process Regression

To solve the problem statement depicted in Section 1-2, two cases were considered and modeled in MATLAB. The first case is addressed in this section, where the two-link and the PUMA 560 manipulator is compensated by an internal inverse dynamics model. But, the internal IDM model failed to capture the noises depicted in Subsection 5-1-1. It is assumed in this Section that the internal IDM of the robot represented in the Figure 1-1 is replaceable. So with this assumption, the remainder of this section will be focusing on addressing

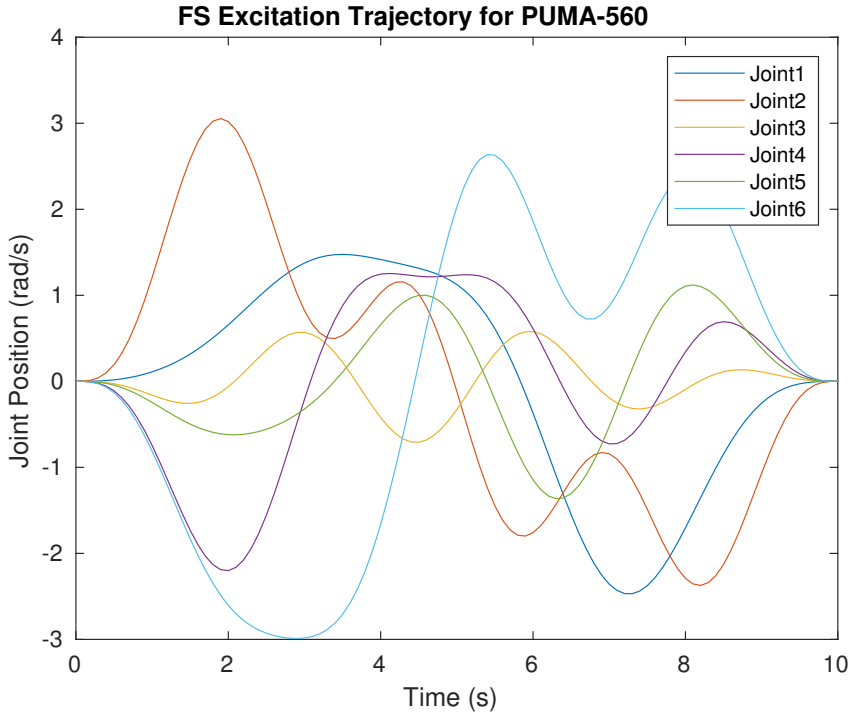

Figure 5-4: Trajectory 1 - Desired excitation trajectory obtained for the PUMA-560 robot

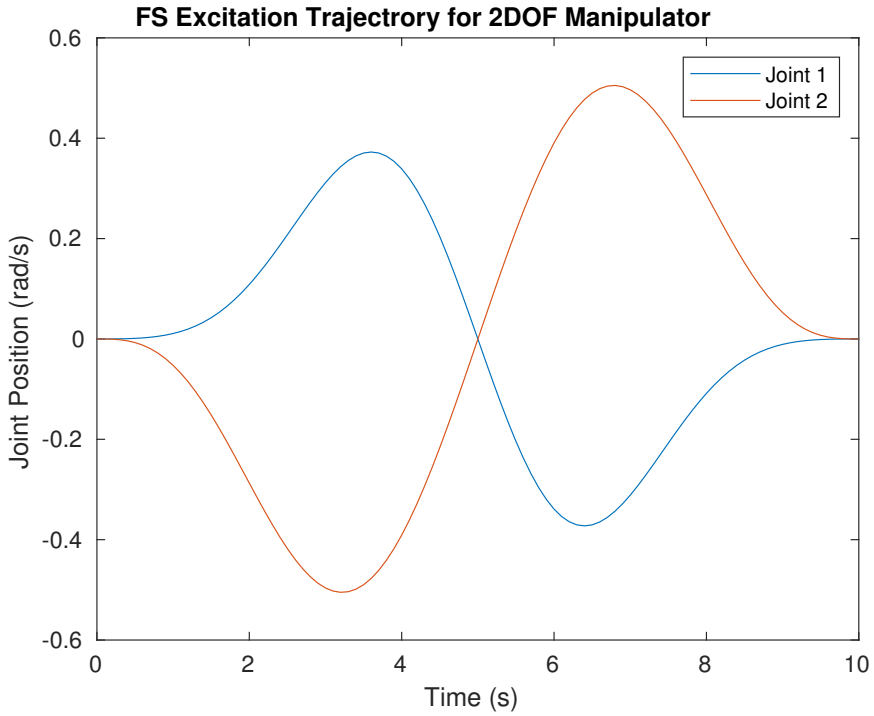

Figure 5-5: Desired excitation trajectory obtained for the Twolink Manipulator

the first case. In addition, it compares and discusses different evaluation studies under this setting. In Subsection 5-3-1, the first case is solved by learning the un-modeled dynamics occurring in the PUMA 560 and the twolink manipulator through semi-parametric Gaussian Process Rgression. In Subsection 5-3-2, the importance of exciting the friction parameters is highlighted. Lastly, the Subsection 5-3-3 compares the semi-parametric models trained with different excitation trajectory optimized with different optimization criterions for the PUMA 560 manipulator.

Table 5-6: Proportional and Derivative Gains for the PUMA-560 robot

	Proportional Gain	Derivative Gain
Identification	diag(75,75,75,75,75,75)	diag(0.32,0.32,0.32,0.32,0.32,0.32)
Testing	diag(10,10,10,10,10,10)	diag(0.2,0.2,0.2,0.2,0.2,0.2)

Table 5-7: Proportional and Derivative Gains for the twolink manipulator

	Proportional Gain	Derivative Gain
Identification	diag(75,75)	diag(0.32,0.32)
Testing	diag(10,10)	diag(0.2,0.2)

5-3-1 Identification

For identification, the IDM depicted in the Figure 1-1 is obtained from Peter Corke's library. The IDM provided by the Peter Corke's is modeled as

$$\tau(q, \dot{q}, \ddot{q}) = W(q, \dot{q}, \ddot{q})\chi, \tag{5-3}$$

$$\text{Where, } \chi = \begin{bmatrix} M_i & MX_i & MY_i & MZ_i & XX_i & XY_i & XZ_i & YY_i & YZ_i & ZZ_i \end{bmatrix}^T. \tag{5-4}$$

The definitions of above parameters can be referred back to chapter 2 [80]. It can be inferred from the Equations (5-3) and (5-4) that IDM provided by the Peter Corke's fails to capture friction and other noises presented in the Section 5-1-1. So, the PUMA 560 robot will fail to track the desired trajectory under zero PD gains. Therefore, high proportional gains were chosen to establish stable error dynamics. The obtained gain values are depicted in Tables 5-6 and 5-7. The adopted proportional and derivative gains were taken to be diagonal to avoid the coupling effect from the feedback term. Therefore, the feedback for each DOF of the robot is strictly independent of other joints. This prevents inaccuracies in one joint to actively disturb the control action on other joints.

The identification experiment is carried out with the excitation trajectories depicted in Figures 5-4 and 5-5 for the corresponding manipulators. The trajectory was simulated for 100s generating a data set with 6006 samples for the PUMA 560 robot and 2002 samples for the twolink manipulator. After the identification experiment, the observation matrix is constructed with the measured joint position, velocity, and acceleration samples to get the least squares estimates. The condition number of the observation matrix built with the measured joint position, velocity and acceleration were found to be 52.47 and 8.78 for the PUMA 560 and the twolink manipulator respectively, which indicates the fact that the system is well excited. After estimating the base parameters, the model is tested at very low gains to check

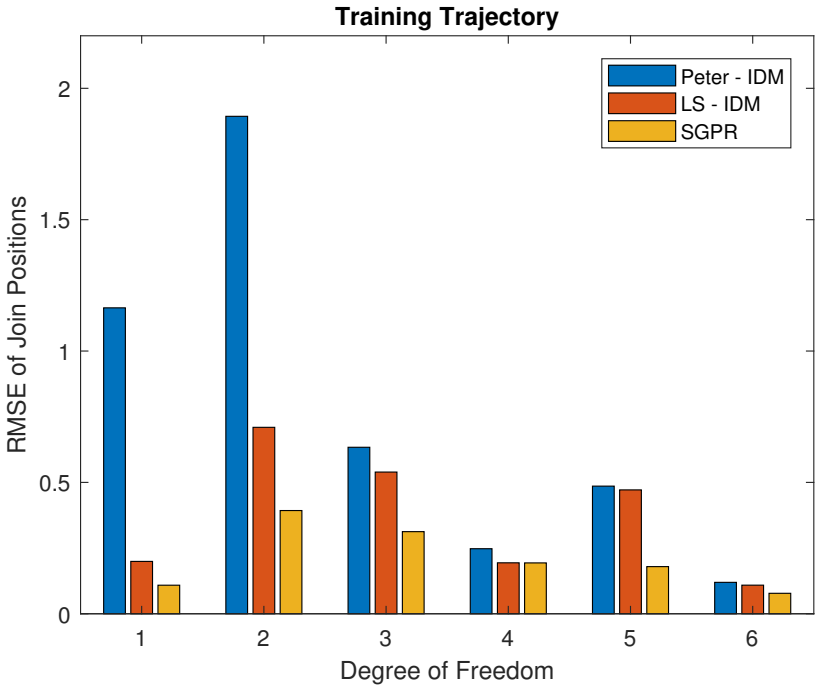

Figure 5-6: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (LS) (LS - IDM) and the Peter Corke's IDM under the training set for the PUMA 560 robot

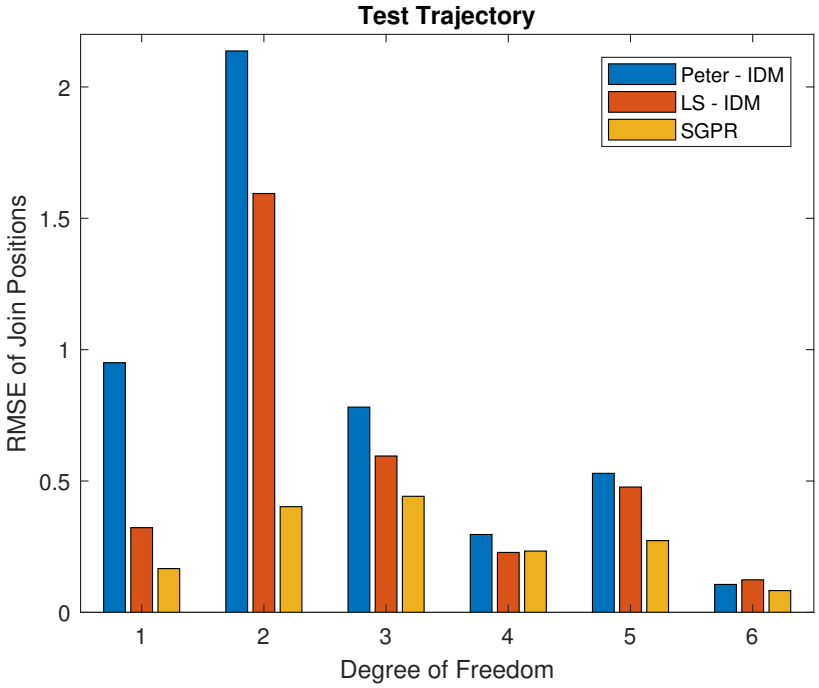

Figure 5-7: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (LS) (LS - IDM) and the Peter Corke's IDM under the test set for the PUMA 560 robot

the efficiency of the estimated RBD model. The estimated RBD model was successful in capturing the inverse dynamics of the robot, viscous and the Coulomb friction. But, it fails to capture the un-modeled dynamics like the motor inertia of each joint, Stribeck, stiction and numerical noises arising due to differentiation of joint positions. Due to this, the perfect cancellation of the robot model did not happen at low PD gains. This can be visualized in Figures 5-6 and 5-8, where the robot has the problem in tracking the desired trajectory only with the RBD model. Subsequently, the Gaussian Process Regression is incorporated to capture the remaining noises that is not explained by the estimated RBD model.

First, a kernel function has to be chosen to do GPR. Since there is no specific thumb rule to choose a kernel function, the squared exponential kernel is chosen. Subsequently, the hyper-parameters of the squared exponential kernel is estimated by optimizing the marginal log-likelihood. As a next step, the obtained RBD model is used as mean to do a semi-parametric GPR. The GPR is trained with the error term $(\tau - W_b(q_d, \dot{q}_d, \ddot{q}_d)\chi)$ and the desired joint position, velocity and, acceleration. Since, it is assumed that it is possible to replace the internal IDM of the robot, the internal IDM from the Peter Corke's library is replaced with the estimated RBD through least squares. Then, additional feed-forward signals from Gaussian process model is given to the robot to compensate for the un-modeled dynamics that cannot be explained by the estimated RBD model.

To validate the results, a different Fourier series is generated and tested for 10s. The data samples were taken sufficiently different from the training data set to highlight the generalization ability of the GPR. After testing, the following could be inferred from the Figures 5-6, 5-7, 5-8, and 5-9.

- The estimated IDM is better than the Peter Corke's IDM. It is due to the ability of the estimated IDM to capture the viscous and Coulomb friction. The relative difference between the estimated IDM and the Peter Corke's IDM is high for the first three joints, as the Coulomb and viscous friction are high for the first three joints as compared to the rest (refer Table 5-1). This effect can be visualized in the Figures 5-6 and 5-7.

- The semi-parametric model tries to learn the error made by the estimated RBD model. Since the perturbations on the fourth joint of the PUMA 560 manipulator is less, the RMSE error of the estimated RBD and the semi-parametric model is equal. In this case, the semi-parametric model relies more on the parametric term, .i.e., estimated RBD model (refer Figures 5-6 and 5-7).

- The semi-parametric model outperforms the estimated RBD model and Peter Corke's IDM. This is due to the ability of the semi-parametric model to address all the noises occurring in the PUMA 560 robot with its non-parametric term, i.e., Gaussian process model. This effect can be visualized in Figures 5-6 and 5-7 for joints 1, 2, 3, 5 and 6 of the PUMA 560 manipulator.

- The same aforementioned reasoning applies to the twolink manipulator as well. From Figures 5-8, 5-9, it can be seen that the semi-parametric GPR has better tracking performance than the estimated RBD model. This signifies the fact that the semi-parametric model has efficiently captured the unknown non-linearities which cannot be explained by the estimated RBD model.

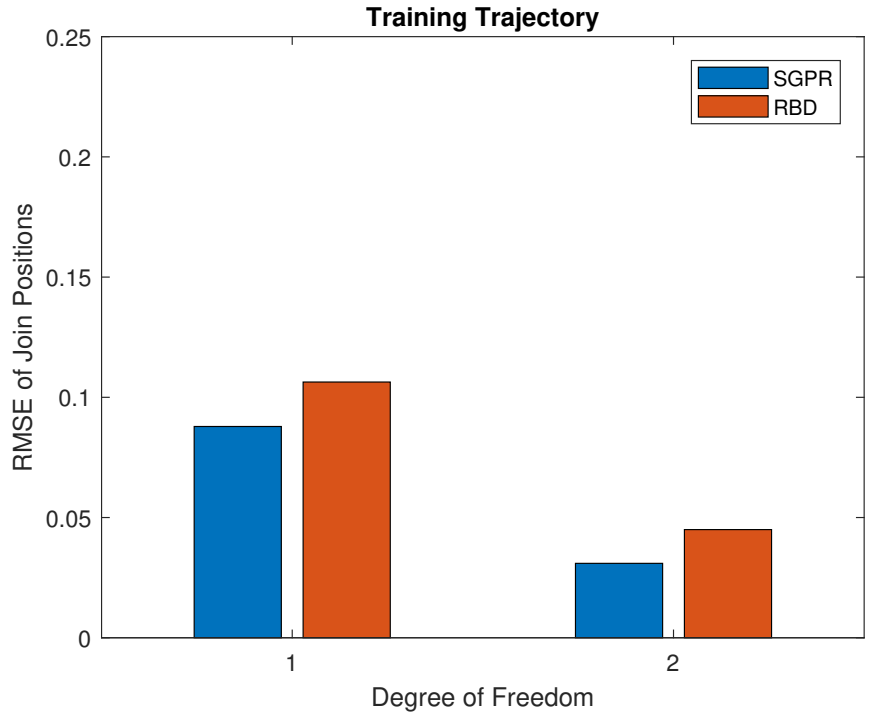

Figure 5-8: Tracking performance of the semi-parametric model and the estimated RBD model under the training set for the twolink manipulator

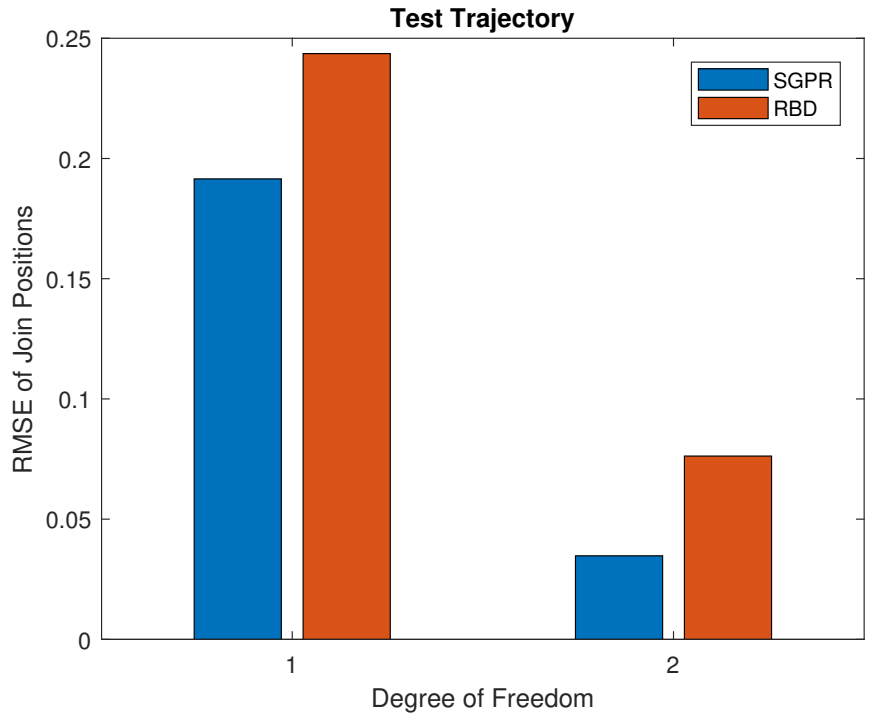

Figure 5-9: Tracking performance of the semi-parametric model and the estimated RBD model under the test set for the twolink manipulator

- Generally, the semi-parametric model under the test set should work equally or worse as compared to the semi-parametric model under the training set. From Figure 5-9 it can be seen that the relative difference between the estimated RBD model and the semi-parametric model is more for the test set as compared to the relative difference the estimated RBD model and the semi-parametric model under the training set. This is because the given test trajectory was simple (refer Figures C-15 and C-16) and the GPR model was able to generalize better. Unlike complex test trajectories given to the PUMA 560 robot, it was done deliberately to test the performance of the semi-parametric model under simple test trajectories for the twolink manipulator.

The similarity function used for both twolink and the PUMA 560 robot is a non-ARD Squared exponential (K_{SEK}) function and it is given as follows

$$K_{\text{SEK}}(x_i, x_j|\theta) = \sigma_f^2 \text{exp}\Big(-\frac{1}{2}\frac{(x_i - x_j)^T(x_i - x_j)}{\sigma_l^2}\Big), \tag{5-5}$$

where σ_f is the signal standard deviation, σ_l is the width of the kernel, x_i and x_j are the input points, θ is the hyper-parameters that is needed to be estimated. The hyper-parameters of the squared exponential kernel ((K_{SEK}) are given as $\theta_1 = \log(\sigma_l)$, $\theta_2 = \log(\sigma_f)$. The main disadvantage of using this kernel type is that it assigns same weights for the length scales (σ_l) for different features, i.e, joint positions, velocities and accelerations of different joints. To overcome this, an Auto Relevance Determination Kernel is used. It is the special variant of non-ARD kernel where the hyper-parameters are not considered same. The co-variance function of this type is given below,

$$K_{\text{ARD}}(x_i, x_j|\theta) = \sigma_f^2 \text{exp}\Big(-\frac{1}{2}\sum_{m=1}^{d}\frac{(x_{im} - x_{jm})^2}{\sigma_m^2}\Big), \tag{5-6}$$

where σ_f is the signal standard deviation, σ_m are the width of the kernel, K_{ARD} represent the ARD - squared exponential kernel with the hyper-parameters $\theta_1 = \log(\sigma_m)$ with $m = 1, 2, 3, \ldots, d$, $\theta_2 = \log(\sigma_f)$. Unlike in Equation (5-5), the K_{ARD} considers different length scales (σ_m) for the joint positions, velocities and accelerations for different joints.

The difference between the performance of the aforementioned kernels will be tested for the twolink manipulator. The kernels differs by the number of hyper-parameters, which is 2 and 7 for the squared exponential and the ARD squared exponential kernel function respectively. As discussed before, the hyper-parameters are obtained by maximizing the marginal log-likelihood technique. Subsequently, the GPR is trained with the excitation trajectory different from Figure 5-5 for the twolink manipulator with the two aforementioned kernels. After training the GPR with two different kernels, the tracking performance was analyzed and the RMSE of the semi-parametric model under training and validation data set were recorded in Figures 5-10 and 5-11.

Generally, the over-fitting problem can occur, when training the GPR and while selecting the hyper parameters. The later problem, i.e, over-fitting in model selection occurs when GPR trains to memorize the training data set but fails to generalize the observed trend in the training set. Due to this model memorization, the model fails to discover a general predictive behaviour by consuming the noise in the data. In addition, over-fitting in model selection is likely to be most severe when the sample of data is small and the number of hyper-parameters

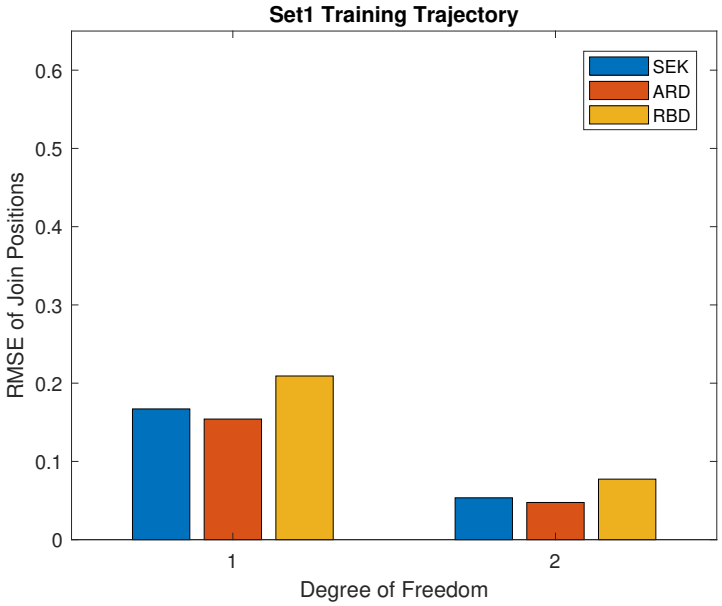

Figure 5-10: Comparison of tracking performance of the estimated RBD model and the semi-parametric model trained with Squared Exponential Kernel (SEK) and ARD - SEK for the twolink manipulator for the training trajectory, where the juxtaposition is given as Root Mean Square Error of joint position for each DOF.

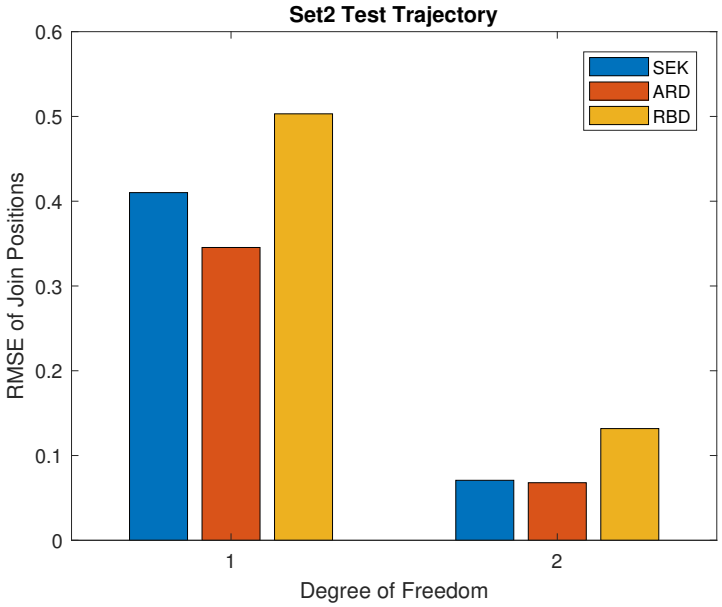

Figure 5-11: Comparison of tracking performance of the estimated RBD and the semi-parametric model trained with Squared Exponential Kernel (SEK) and ARD - SEK for the twolink manipulator for the test trajectory, where the juxtaposition is given as Root Mean Square Error of joint position for each DOF.

to be tuned is relatively large [82]. Cawley suggest a method to detect over-fitting in model selection. In that proposed method, the GPR is fitted with an Auto Relevance Determination (ARD) kernal function and a non-ARD kernel function. It was found that ARD kernel fails to perform better as compared to the non-ARD kernel for the validation set due to the over-fitting in tuning the hyper-parameters.

From the Figures 5-10 and 5-11, it can be inferred that the performance of the GPR trained with ARD squared exponential kernel has better performance as compared to the GPR trained with squared exponential kernel. Therefore, according to Cawley there are no over-fitting in model selection for the ARD squared exponential kernel for the two-link manipulator.

5-3-2 Importance of Excitation Trajectory

Accurate identification of a model requires the input to be persistently exciting such that it excites all the parameters to be identified [9].

This section focuses to highlight the aforementioned fact. To do so, three different excitation trajectories were generated.

- Trajectory 1 - It excites all the base parameters along with viscous and Coulomb friction parameters of all the joints of PUMA-560 robot. The trajectory is generated as discussed in Section 5-2 and it can be visualized in Figure 5-4.
- Trajectory 2 - It excites all the base parameters except viscous and Coulomb friction. This is established by excluding the columns of viscous and Coulomb friction coefficients in the observation matrix (W_b). The trajectory generated after optimizing the condition number of matrix W_b, excluding the columns of viscous and Coulomb friction is depicted in Figure 5-12.
- Trajectory 3 - It is a combination of trajectory 2 and the trajectory depicted in Figure 5-14. Both trajectories does not excite viscous and Coulomb friction.

The trajectories discussed above is used to obtain three different semi-parametric models. The procedure incorporated in this section to obtain the semi-parametric model is similar to the one depicted in Subsection 5-3-1. The first two semi-parametric models are obtained by using the Trajectory 1 and 2 with 6001 data samples of joint positions, velocities and, accelerations for each trajectory. Whereas, the third semi-parametric model is obtained by using the Trajectory 3 with 12002 data samples of joint positions, velocities and, accelerations. Subsequently, a test trajectory is generated to test the tracking performance of the obtained semi-parametric models. The test trajectory is chosen to excite the viscous and Coulomb friction of all the joints. The comparison between the three different semi-parametric models trained with three different trajectories under the test set can be visualized in the Figure 5-13 and the following inferences can be made,

- The semi-parametric model trained with a trajectory that has friction excitation has better tracking performance as compared to the semi-parametric model trained with a trajectory that has no friction excitation. This is because the semi-parametric model uses the RBD model whenever it encounters an unseen data. But, the RBD model

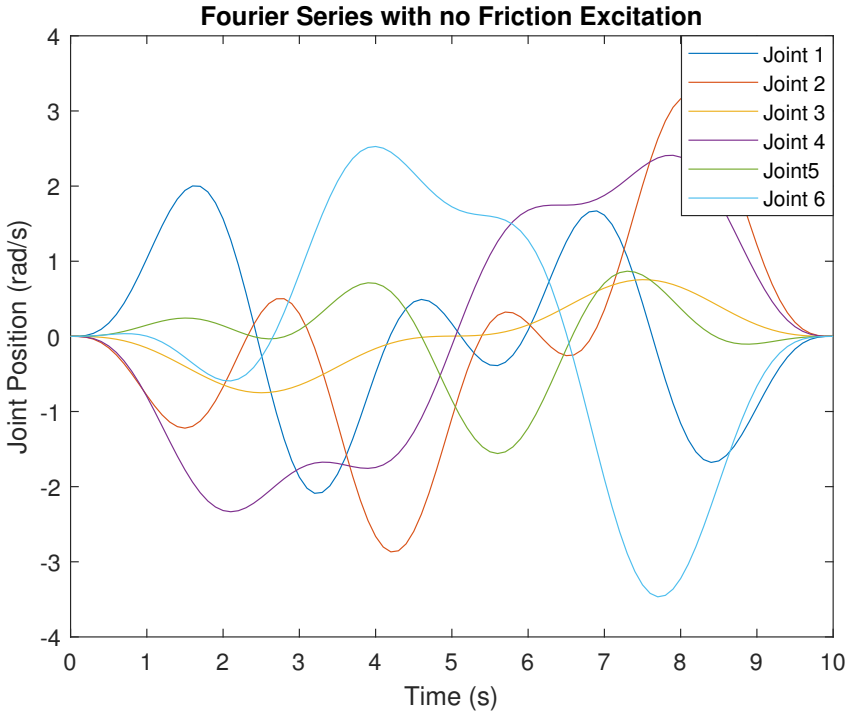

Figure 5-12: Trajectory 2 - Fourier series trajectory with no friction excitation generated for first 10s for the PUMA-560 robot

estimated from the LS approach failed to capture the correct viscous and Coulomb friction coefficients as the trajectories 2 and 3 fails to excite those coefficients.

- Even though any un-modeled dynamics could be captured by the GPR. It is important that the input/output data used for training the GPR should have sufficient information. Since the Trajectory 2 and 3 failed to excite the Coulomb and viscous friction, their corresponding semi-parametric model fails to predict the correct torque when it encounters an unseen data. This due to the fact that the unseen data is the test trajectory that excited the viscous and Coulomb friction coefficients. Therefore, it can be concluded that the semi-parametric model will fail to capture the non-linearities if they are not excited.

- Even though, the trajectory 3 has more data points. It can be seen from Figure 5-13 that, it is not better as compared to the semi-parametric model trained with Trajectory 1. Therefore, the betterment of the semi-parametric model does not lie in the number of samples taken unless all the parameters are excited. The GPR generalizes better only if the information is rich rather if it sees the same data again.

- The relative difference between the semi-parametric model trained with different trajectories is high for the first three joints as compared to joint 4, 5 and, 6. This is because the viscous and Coulomb friction is set high for the first three joints as compared to the other joints (refer Table 5-1).

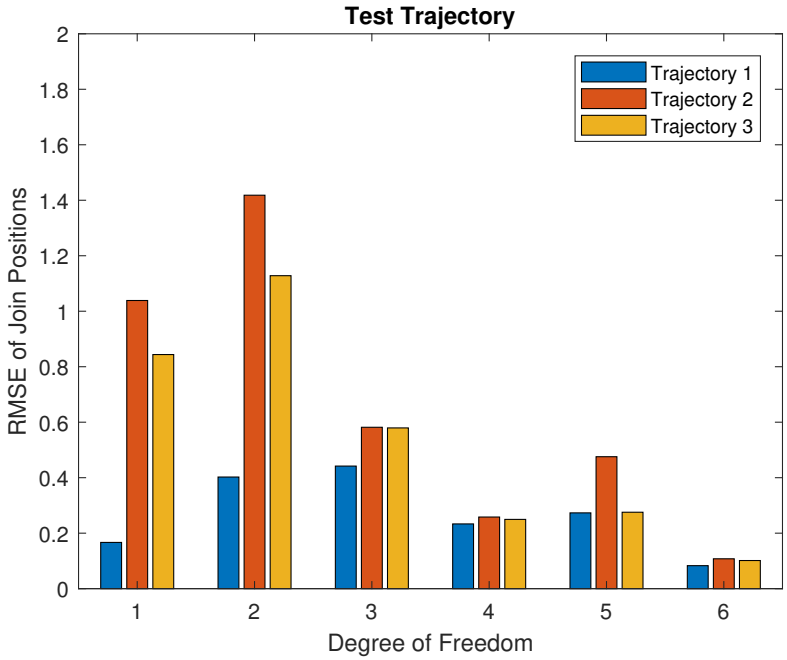

Figure 5-13: Comparison of tracking performance of three different GPR trained with three different training set forthe PUMA 560, where the juxtaposition is given as Root Mean Square Error of joint position for each DOF.

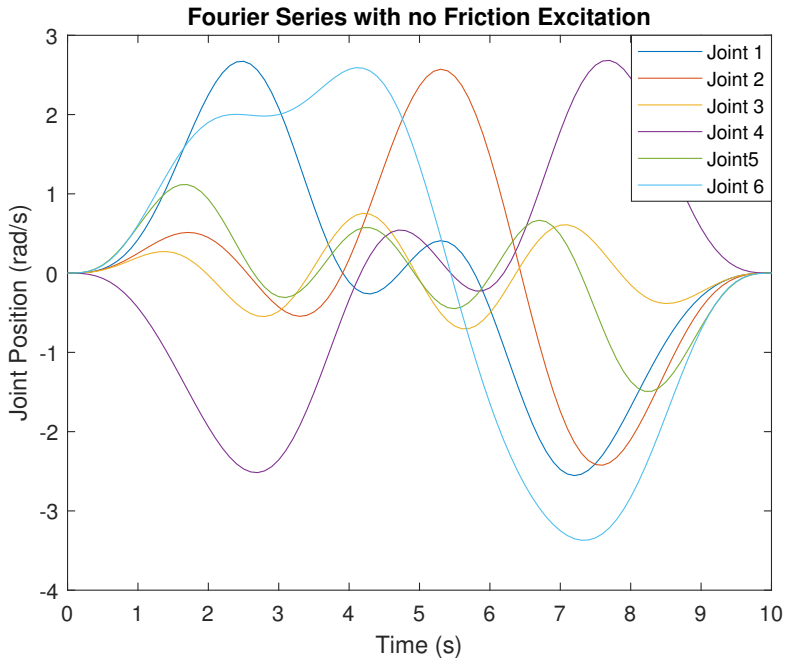

Figure 5-14: Trajectory with no friction excitation generated for the PUMA560 robot for 10s

5-3-3 Comparison of Different Optimization Criterion

In this subsection, the performance of the semi-parametric model trained with different trajectories is compared and examined. The procedure incorporated in this section to obtain the semi-parametric model is similar to the one depicted in Subsection 5-3-1. However, it is important to understand how these different trajectories were generated before analyzing the comparison. The trajectories used for training the semi-parametric model were generated by incorporating different optimization criteria. These criteria are briefly discussed below.

- Cond(W_{fe}) - The trajectory is optimized by minimizing the condition number of the observation matrix W_b. The matrix W_b is built with the columns corresponding to the base parameters and the columns of Coulomb and viscous friction of each link.
- Cond(W_{nfe}) - The trajectory is optimized by minimizing the condition number of the observation matrix W_b by excluding the columns of Coulomb and viscous friction.
- $\sum_{i=1}^{6}$ Cond(W_{fe}) - The observation matrix W_b is built with the columns of base parameters, viscous, and Coulomb friction corresponding to each link separately. Subsequently, the condition number of observation matrix for each link is summed up and optimized. The separation of base parameters and friction parameters of each link for PUMA 560 robot is given in Section A-1-1
- $\sum_{i=1}^{6}$ Cond(W_{nfe}) - The observation matrix W_b is built similarly as done for $\sum_{i=1}^{6}$ Cond(W_{fe}). But here, it does not take the columns of Coulomb and viscous friction of each link into account.

The combination of trajectories generated using the aforementioned optimization criteria with a High-frequency Fourier Series (HFS) signal, is used for training the semi-parametric model. The HFS signal is a high acceleration Fourier wave generated for the PUMA 560 robot. The maximum closed-loop bandwidth of the PUMA 560 robot with the control scheme depicted in Figure 1-1 was found to be 5 Hz. Hence, the HFS signal was generated to cover the entire bandwidth frequency. With these signals, eight different combinations were used, for training the semi-parametric model as represented in Table 5-8.

Three different test sets are used to check the generalization ability of the obtained semi-parametric models. The chosen test sets are completely different from each other and the training set. The differences between the semi-parametric model trained with different trajectories for the PUMA 560 robot, can be visualized in Figure 5-15. Figure 5-15 represents the tracking performance of the PUMA 560 for joint 2, validated with one of the three test trajectories. Furthermore, the mean of Root Mean Square Error (RMSE) for each joint for three different test sets is calculated and reported in Table 5-8.

The following inference could be made from the Table 5-8:

- The semi-parametric model obtained after training with the Trajectory 5 has better tracking performance as compared to the Trajectory 1. It is because the division of the total observation matrix (W_b) into sub-regressors has led to better excitation and a lower condition number. Consequently, a better RBD and a better Gaussian process model are obtained.

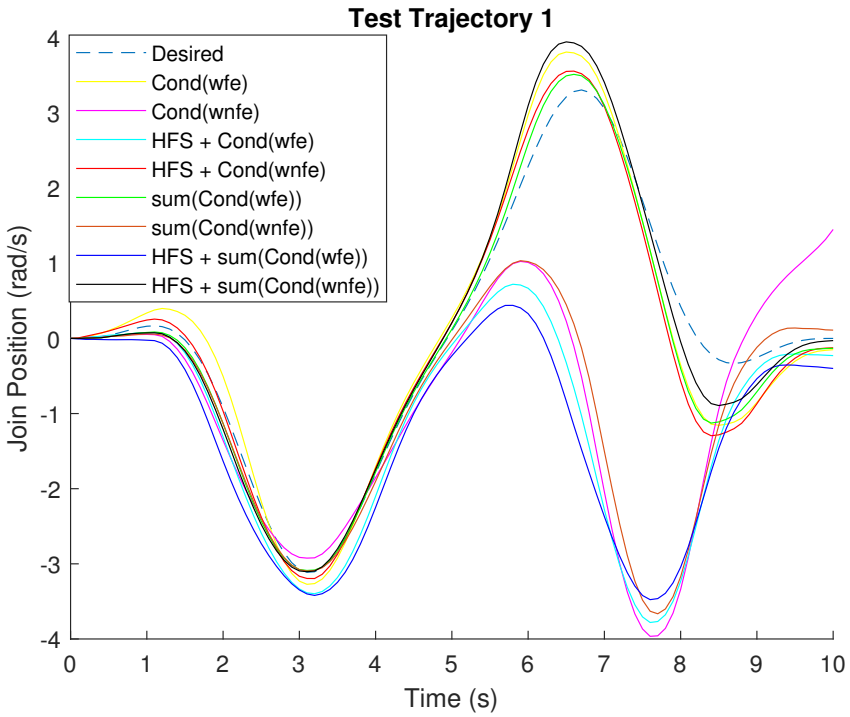

Figure 5-15: Tracking Performance of the semi-parametric model trained with different trajectories for 2 DOF of the PUMA 560 robot.

Table 5-8: μ_i - Mean of RMSE of the semi-parametric model trained with 8 different trajectories for the PUMA 560 robot, where i represents the joints of the robot.

S.no	Trajectory	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	$\sum \mu_i$
1	$\mathrm{Cond}(W_{fe})$	0.16	0.46	0.35	0.203	0.236	0.082	1.491
2	$\mathrm{Cond}(W_{nfe})$	1.203	2.09	0.606	0.373	0.53	0.71	5.512
3	$\mathrm{HFS} + \mathrm{Cond}(W_{fe})$	0.21	2.12	0.44	0.203	0.23	0.113	3.316
4	$\mathrm{HFS} + \mathrm{Cond}(W_{nfe})$	0.183	0.386	0.353	0.22	0.243	0.099	1.4840
5	$\sum_{i=1}^{6} \mathbf{Cond}(W_{fe})$	**0.166**	**0.386**	**0.233**	**0.223**	**0.31**	**0.088**	**1.416**
6	$\sum_{i=1}^{6} \mathrm{Cond}(W_{nfe})$	1.053	1.94	0.51	0.29	0.43	0.503	4.726
7	$\mathrm{HFS} + \sum_{i=1}^{6} \mathrm{Cond}(W_{fe})$	0.243	1.98	0.406	0.22	0.32	0.1066	3.275
8	$\mathrm{HFS} + \sum_{i=1}^{6} \mathrm{Cond}(W_{nfe})$	0.2	0.33	0.35	0.22	0.323	0.17	1.593

- The semi-parametric model trained with the Trajectory 2 and 6 fails miserably as the most dominant viscous and Coulomb friction parameters were not excited. As depicted in Subsection 5-3-2, the semi-parametric model will fail to capture the friction unless the parameters are excited.
- Trajectory 4 and 8 performs better than Trajectory 2 and 6, even though the friction parameters were not excited. This is due to the addition of extra HFS signal for the identification of the semi-parametric model. The generated HFS signal has perfectly excited all the noises present in the frequency range.
- Irrespective of the trajectories used, the mean of RMSE for joint 4, 5, 6 is approximately equal for all the semi-parametric models. This is because the viscous and Coulomb friction is sufficiently set high for the first three DOF (refer Table 5-1). The semi-parametric model trained with trajectories 1, 4, 5 and 8 were successful in capturing the Coulomb and viscous friction for the first three DOF as compared to the rest. Hence, the relative difference could be seen for the first three joints.
- When the number of data points increases beyond a certain point, the GPR tends to over-fit the data points and fails to observe the general trend. So, by including the HFS signal along with the trajectory that has already excited the viscous and Coulomb friction. The performance of the semi-parametric model trained with Trajectory 3 and 7 performs worse as compared to the semi-parametric model trained with Trajectory 1, 4, 5 and 8 due to over-fitting of data points.

5-4 Case 2 : Semi-parametric Gaussian Process Regression

Unlike the case 1 depicted in Section 5-3, the internal inverse dynamics model represented in Figure 1-1 is inherently inaccurate for the second case. The internal IDM model also fails to compensate for all the noises represented in Subsection 5-1-1 like the first case. Furthermore, it is assumed in this section, that the internal IDM cannot be replaced or set to zero.

In this section, the case 2 is modeled, with the help of the twolink manipulator. To start with, the inverse dynamics model of the twolink manipulator has four base parameters. With the help of the Table 5-2 and the Steiner's theorem, the values of these base parameters (χ_b) was estimated to be $[1.25, -0.5, 0, 0.5]^T$. As discussed before, the internal IDM is made inaccurate by adding offset to the base parameters. Hence, the internal IDM used for identification and control is given as follows,

$$\tau_{\text{I-IDM}} = W_b(q_d, \dot{q}_d, \ddot{q}_d) \begin{bmatrix} 2.25 \\ -1.5 \\ 1 \\ 1.5 \end{bmatrix} \tag{5-7}$$

where $\tau_{\text{I-IDM}}$ is the torque due to Inaccurate-Inverse Dynamics Model (I-IDM). The internal IDM represented in Equation 5-7, is not modeled with the linear coefficients of viscous and Coulomb friction. So, it fails to capture the friction and other un-modeled dynamics occurring in the two-link manipulator. Due to which, the two-link manipulator is unable to track the desired trajectory properly under the low stiffness and damping conditions. This effect can be

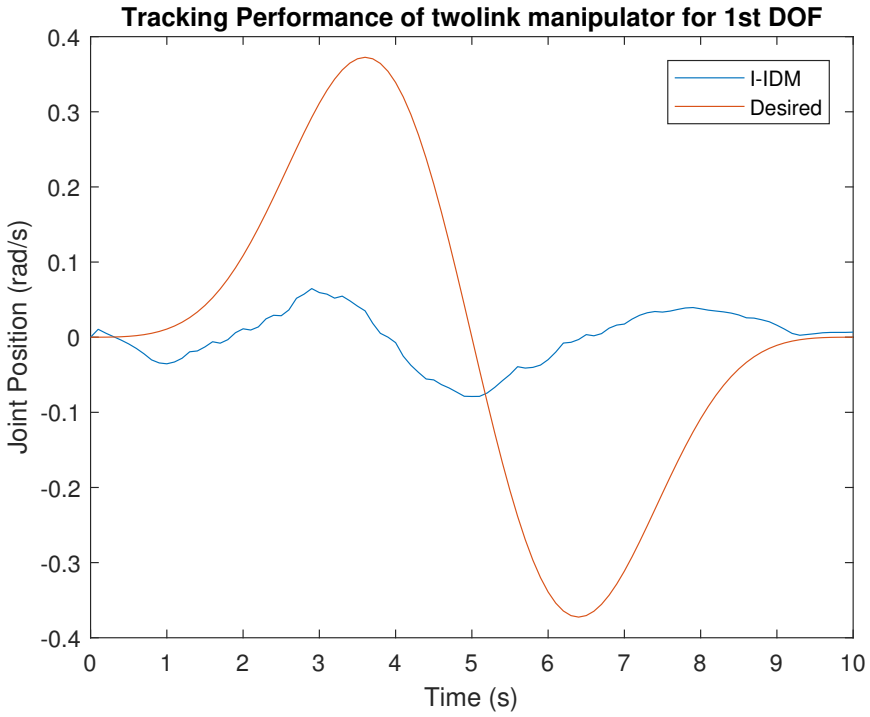

Figure 5-16: Tracking performance of the twolink manipulator for its 1st DOF compensated by Inaccurate-Inverse Dynamics Model (I-IDM) under zero stiffness and damping conditions.

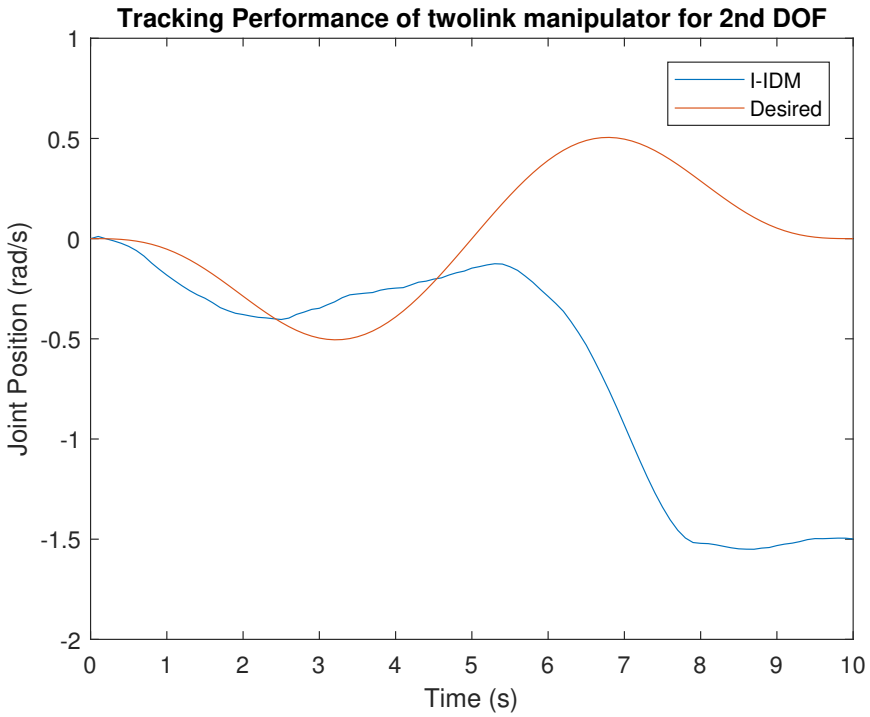

Figure 5-17: Tracking performance of the twolink manipulator for its 2nd DOF compensated by Inaccurate-Inverse Dynamics Model (I-IDM) under zero stiffness and damping conditions.

visualized in the Figures 5-16 and 5-17. Since the twolink manipulator is not stable at low feedback gains, the closed loop is made stable by increasing the proportional and derivative gain, before carrying out the semi-parametric GPR. The obtained gain values are mentioned in the Table 5-7. Like the previous section, the same excitation trajectory (Figure 5-5) was incorporated, for the least squares identification and the GPR.

After recording the feedback torques for 100s, the RBD error model ($W_b\chi_{\text{error}}$) is identified using the least squares approach. Then, it is used as mean to capture all the noises occurring in the manipulator using the GPR. The GPR takes the desired joint position, velocity and acceleration as input and the feedback error term ($\tau_{\text{PD}} - W_b(q_d, \dot{q}_d, \ddot{q}_d)\chi_{\text{error}}$) as output. Like before, the hyperparameters of the squared exponential kernel function are estimated using the maximization of marginal log-likelihood. Subsequently, after training the GPR, the feed-forward signals are given to the robot in the presence of incorrect internal IDM. These feed-forward signals are combinations of torque signals from the RBD error model and the Gaussian process model for a given joint position, velocity, and acceleration (refer Figure 4-1).

A trajectory different from the training set is chosen to test the generalization ability of the obtained semi-parametric model. The performance of the two-link manipulator, compensated by the internal IDM with the semi-parametric model and without the semi-parametric model, is depicted in the Figures 5-18 and 5-19. Subsequently, the following inferences can be from these figures:

- The Root Mean Square Error (RMSE) of the joint position for the two-link manipulator compensated by the semi-parametric model is less as compared to the manipulator compensated by the inaccurate internal inverse dynamics model. This is due to the ability of the semi-parametric model to capture the model inaccuracies and un-modeled dynamics occurring in the twolink manipulator.
- The internal IDM was inherently inaccurate and was not modeled to capture the viscous and Coulomb friction. Since the test trajectory was chosen to excite the viscous and Coulomb friction of the two-link manipulator, the relative differences between the internal IDM and the semi-parametric model could be seen (refer Figure 5-19).

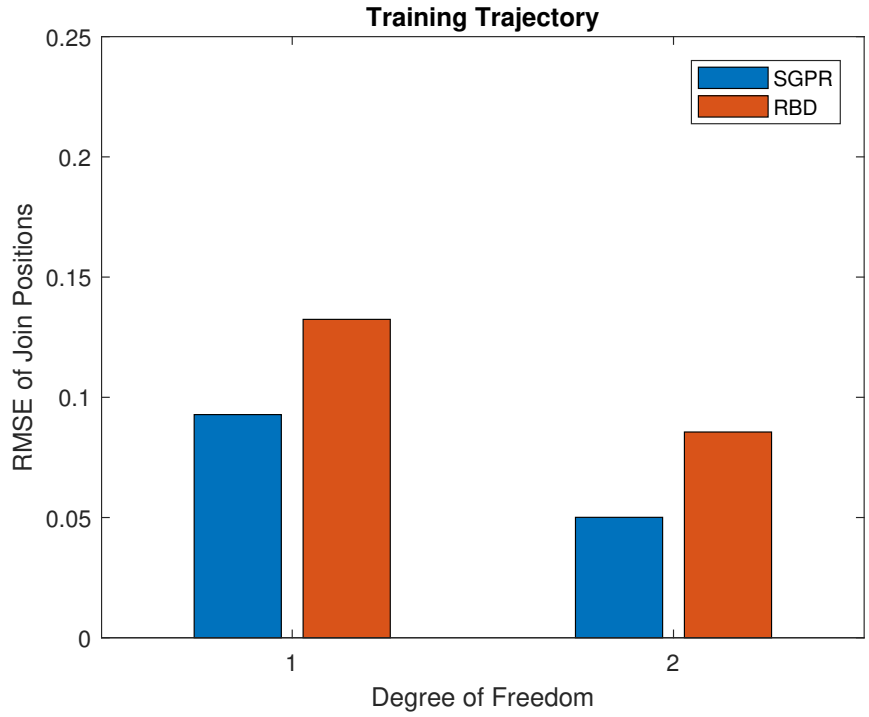

Figure 5-18: Tracking performance of the semi-parametric model and the inaccurate IDM under the training data set for the twolink manipulator.

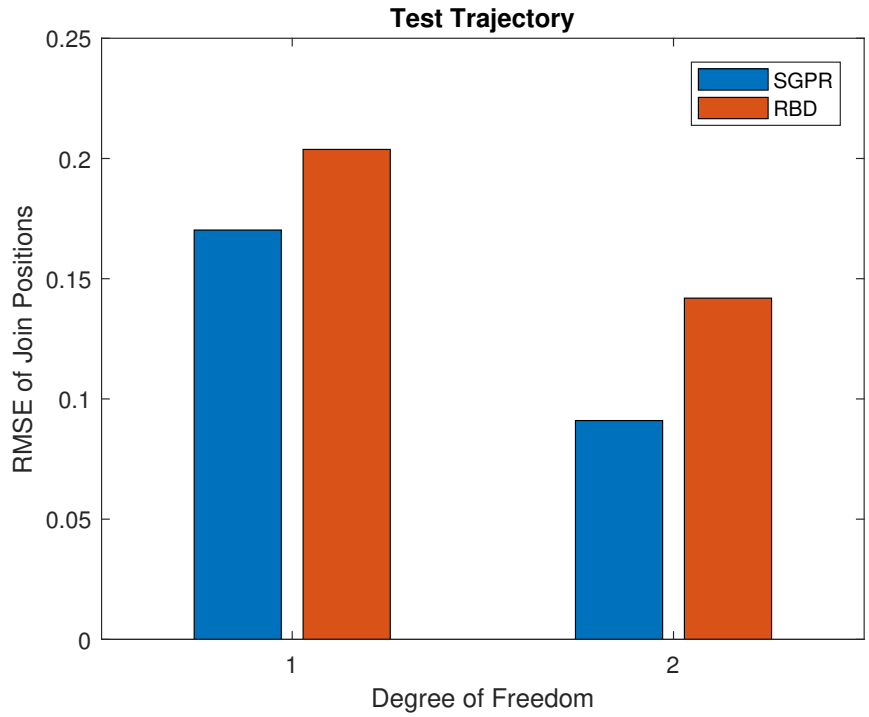

Figure 5-19: Tracking performance of the semi-parametric model and the the inaccurate IDM under the test data set for the twolink manipulator.

Chapter 6

Conclusion and Recommendations

The problem stated in Section 1-2 is tackled by considering two cases in this thesis. The internal inverse dynamics model having correct base parameters was used for controlling the robot for the first case. Whereas, the internal inverse dynamics model was made inherently inaccurate by adding offset in the base parameters in the second case. But in both cases, the internal IDM did not compensate for un-modeled dynamics occurring in the manipulator. So, it demanded proper feed-forward torque signals to compensate for the noises occurring in the manipulator. The thesis incorporated a semi-parametric Gaussian process regression to tackle the two cases. First, the parametric term was identified using the least squares approach. The parametric term was considered to be RBD model and RBD error model for the first case and second case respectively. Later, the parametric term was used as mean to capture the non-parametric term using the Gaussian Process Regression. The methods discussed in this thesis towards solving the problem statement was implemented on the PUMA 560 and the two-link manipulator in MATLAB. Furthermore, the thesis proposed a new excitation trajectory and an optimization criterion and showed its efficiency in capturing the noises occurring in the manipulator in Chapter 5. So, from the experiments carried out on the PUMA 560 and the two link manipulator, the following conclusions were given in Section 6-1. Subsequently, in Section 6-2, a few recommendations were given for future research work on this topic.

6-1 Conclusion

Industry 4.0 is a setup where the automation and computers come together in a completely new way. The robots used in these industries are connected remotely to the computer systems equipped with machine learning algorithms. These algorithms can learn and control the robots with little input from human operators. Many manufacturing industries are trying to transform themselves into industry 4.0. Since the robots in these industries work closely with the human beings, the amount of safety and the robot's efficiency to work under all the conditions are of utmost importance. The thesis contributes to the research and the development of collaborative robots using the machine learning algorithms. The main aim

of the thesis is to improve the performance of the serial manipulator under low compliant conditions. Under low stiffness and damping conditions, the robot is entirely compensated by the inverse dynamics model. If the model is imperfect, the robot becomes unstable and fails to perform better at low compliant conditions. To tackle this issue, an offline method named semi-parametric Gaussian process regression was incorporated. The experiments were carried out on the PUMA 560 manipulator and the two-link manipulator. With the obtained simulation results, the following conclusions can be made,

Lowest Condition number leads to better excitation
Attaining lowest condition number for the matrix W_b during trajectory optimization will lead to better excitation and better least squares estimates. The sensitivity of the least squares approach for the RBD identification depends on the condition number of the observation matrix (W_b). With this idea, the observation matrix was divided into sub-regressor for each joint separately. By doing this, the obtained condition number was less as compared to minimizing the condition number of the whole W_b matrix. Subsequently, the tracking performance of the semi-parametric model was better. Subsection 5-3-3 clearly highlighted the above fact.

Using High Frequency Fourier series has lead to excitation of all the noises
Unknown dynamics cannot be modeled or added in the W_b matrix for trajectory optimization. So, sweeping the signal with sufficient amplitude over the entire frequency range will excite all the noises. Hence, a high acceleration and high-frequency Fourier series were generated to sweep the entire frequency range of the robot. The proposed high-frequency Fourier series proved to excite all the noise occurring in PUMA 560 robot and its efficiency could be visualized in Figure 5-15 (refer Subsection 5-3-3).

In terms of modeling accuracy, the semi-parametric models outperforms RBD model
The Semi-parametric model performs equally or better than the RBD model. It performs equally with the RBD model when it encounters an unknown data. During this time, it uses the information of the RBD model which was incorporated as mean. Meanwhile, the semi-parametric model performs better than the RBD, when the GPR is able to recognize a new query point. Since the semi-parametric model was able to capture the un-modeled dynamics that cannot be explained by the RBD model, it outperforms the RBD model. This can be visualized from the Figures 5-6, 5-7, 5-8, 5-9, 5-18 and 5-19 presented in the Sections 5-3 and 5-4.

In terms of learning speed, generalization capabilities and modeling accuracy, the semi-parametric models outperforms standard GPR
Though, standard GPR was not tested and compared with the semi-parametric model in this thesis. Intuitively, it can be reasoned that the semi-parametric model will outperform the standard GPR. This is due to the fact the standard GPR requires a huge amount of data to get trained accurately. The standard GPR fails when it sees a data from the unexplored space. Whereas, the semi-parametric model incorporates the RBD model when it encounters an unknown data. Since, the RBD model gives a unique relationship between the joint torques and joint position, velocity and acceleration. The semi-parametric model will outperform the standard GPR.

Estimating different weighting for the hyper-parameters through ARD kernel outperforms non-ARD kernel
Unlike the non-ARD kernel, the hyper-parameters of the ARD kernel have different scaling

for different inputs namely, the joint position, velocity, and acceleration. Due to this, the tracking performance of the semi-parametric model trained with an ARD kernel was found to be better than the semi-parametric model trained with a non-ARD kernel. If the above fact is violated, then there is over-fitting in selecting the hyper parameters for the ARD kernel.

The semi-parametric model will not capture friction unless the excitation trajectory excites them
The semi-parametric model trained with a trajectory that has friction excitation has better tracking performance as compared to the semi-parametric model trained with a trajectory that has no friction excitation. This is because the Gaussian Process model failed to capture the non-linearities as they were not excited. Furthermore, the betterment of the GPR does not lie in the number of samples taken unless all the parameters are excited, i.e., the input should be rich enough.

6-2 Recommendations

The recommendations for future work on this topic is given as follows,

To Test the suggested methods on a real manipulator
The dynamics of hydraulic tubes, actuator, and cable drives are not modeled in the serial manipulators incorporated in this thesis. These errors occur prominently in a real manipulator. It would be difficult and challenging to model all the aforementioned dynamics in simulation as these modeling errors may vary from the errors witnessed on a real manipulator. But, the proposed methods and concepts will not change when applied on a real manipulator.

Applicability of proposed methods for other control structures
The control structure assumed in this thesis is a PD + Inverse Dynamics Model controller for the manipulator (Refer Figure 1-1). Some robot controllers do not use the inverse dynamics model for control For e.g., a passivity-based controller. In this case, if the robot allows the end-users to set the internal controller to zero, then applying the semi-parametric GPR similar to the one applied in Section 5-3 will solve the issue. On the contrary, if the internal controller cannot be set to zero, then improving the solution given in this thesis to meet all control structures will be a future scope for improvement.

Extending the proposed method for time varying dynamics
The entire thesis focused on learning the robot dynamics under time-invariant conditions. In some cases, the manipulator is used to pick and place of unknown objects, which may cause instantaneous system changes. This kind of work has been addressed in [48] but the optimally of the excitation trajectory was not considered during the identification process. In addition, the idea of feedback learning with inaccurate inverse dynamics model using semi-parametric approach was not incorporated. So, extending this thesis to time-variant conditions and addressing the issue with the online semi-parametric GPR will be a future scope for improvement.

Appendix A

Dynamics Model Reduction

The dynamic model parameters can be reduced to base parameters by using analytical approach or by numerical approach as given in [11]. Numerical method for estimation of base inertial parameters is based on QR decomposition. This approach is tedious and error prone. Hence, it would be wise to take an analytical approach for dynamic model reduction, as it is simple and straightforward. This section discusses about analytical approach in detail and its steps to arrive at minimum base parameter set. In Section A-1, the rules for assigning the frames for the robot structure is discussed. Then, Denavit Hartenberg (DH) table is filled using the assigned frames for the PUMA 560 and twolink manipulator. Subsequently in Subsection A-1-1, the minimum inertial parameters/base parameters are derived using the DH table from the Section A-1. Section A-2 talks about the software named sympybotics, which is used to obtain the columns of the observation matrix W.

A-1 Denavit Hartenberg

The robot kinematics and differential kinematics are required to calculate the dynamic model of the robot manipulator. The robot kinematics can be assigned with the help of Denavit-Hartenberg convention. To use this convention, the manipulator has to be assigned frames for each joints. A serial manipulator has multiple joints where the links are assumed to be rigid. The joints can be either prismatic or revolute. In this thesis, the joints of the PUMA 560 and the twolink manipulator were assumed to be rigid and revolute. For assigning the frames for each joints, the following rules has to be obeyed as given in [11],

- z axis should be in the direction of joint axis
- x axis is perpendicular to both z_i and z_{i-1}. If no unique perpendicular could be found, the x axis goes from z_{i-1} to z_n direction
- y axis should follow Right hand rule
- x_i axis must intersect the axis z_{i-1},

where $[x_i \ \ y_i \ \ z_i]$ represent co-ordinates of each frame with i running from 1 to n. Subsequently, the DH table can be filled using the co-ordinate frames from the Figure A-1. The DH parameters for the PUMA 560 and the twolink manipulator are represented in Table A-1, A-2 respectively. The definition of the parameters used in DH table can be referred to Table A-3.

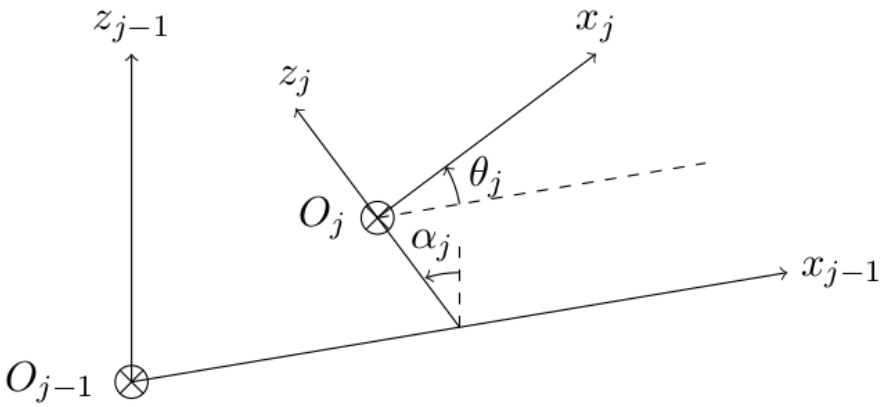

Figure A-1: Co-ordinate frame for a serial link structure [49]

Table A-1: Denavit Hartenberg Tabulation for the PUMA 560 robot

Link	σ_i	α_i	d_i	θ_i	r_i
1	0	$\pi/2$	0	q_1	0
2	0	0	0	q_2	0.4318
3	0	$-\pi/2$	-0.15005	q_3	0.0203
4	0	$\pi/2$	0.4318	q_4	0
5	0	$-\pi/2$	0	q_5	0
6	0	0	0	q_6	0

Table A-2: Denavit Hartenberg Tabulation for the twolink manipulator

Link	σ_i	α_i	d_i	θ_i	r_i
1	0	0	0	q_1	1
2	0	0	0	q_2	1

A-1-1 Regrouping of Dynamic Parameters

With the help of DH table from Subsection A-1, the dynamic model for the serial manipulator can be reduced. The theorem from [11] is reported below, which can be used for obtaining the base parameters.

Table A-3: The definition of parameters used in Table A-1 and A-2

Notations	
σ_i	0 if joint j is revolute and 1 if the joint is prismatic
α_i	Angle between Z_{j-1} and Z_j about X_{j-1}
d_i	Distance between Z_{j-1} and Z_j about X_{j-1}
θ_i	Angle between X_{j-1} and X_j about Z_j
r_i	Distance between X_{j-1} and X_j about Z_j

Table A-4: The definition of terms used in Section A-1-1

Notations	
$\begin{bmatrix} MXR_i & MYR_i & MZR_i \end{bmatrix}$	Three components of regrouped first moment of inertia with respect to frame i
$\begin{bmatrix} XXR_i & XYR_i & XZR_i & YYR_i & YZR_i & ZZR_i \end{bmatrix}$	Six components of the regrouped inertial tensor with respect to frame i

If joint j is revolute, the parameters YY_j, MZ_j ***and*** M_j ***can be grouped with the parameters of link j and link j-1. the resulting grouped parameters are:***

$$
\begin{aligned}
XXR_j &= XX_j - YY_j \\
XXR_{j-1} &= XX_{j-1} + YY_j + 2r_jMZ_j + r_j^2M_j \\
XYR_{j-1} &= XY_{j-1} + d_jS\alpha_jMZ_j + d_jr_jS\alpha_jM_j \\
XZR_{j-1} &= XZ_{j-1} - d_jC\alpha_jMZ_j - d_jr_jC\alpha_jM_j \\
YYR_{j-1} &= YY_{j-1} + CC\alpha_jYY_j + 2r_jCC\alpha_jMZ_j + (d_j^2 + r_j^2CC\alpha_j)M_j \\
YZR_{j-1} &= YZ_{j-1} + CS\alpha_jYY_j + 2r_jSS\alpha_jMZ_j + (d_j^2 + r_j^2SS\alpha_j)M_j \\
MXR_{j-1} &= MX_{j-1} + d_jM_j \\
MYR_{j-1} &= MY_{j-1} - S\alpha_jMZ_j - r_jS\alpha_jM_j \\
MZR_{j-1} &= MZ_{j-1} + C\alpha_jMZ_j + r_jC\alpha_jM_j \\
MR_{j-1} &= M_{j-1} + M_j
\end{aligned}
$$

Using the above theorem, the dynamic model of the manipulator can be reduced. After regrouping with the help of aforementioned equations, the number of parameters to be estimated dropped from 60 to 48 for PUMA 560 robot and from 20 to 8 for twolink manipulator (including the coefficients of viscous and Coulomb friction of each joints). The minimum parameters set (χ_b) for the PUMA 60 robot is given as follows,

Joint 1:
$YY_1 + YY_2 + ZZ_3 - 0.3MY_3 - 0.18M_2 - 0.16M_3 - 0.16M_4 - 0.16M_5 - 0.16M_6$
fv_1
fc_1

Joint 2:
$XX_2 - YY_2 + 0.18M_2 + 0.18M_3 + 0.18M_4 + 0.18M_5 + 0.18M_6$
XY_2
$XZ_2 - 0.43MZ_2 + 0.43MY_3 - 0.06M_3 - 0.06M_4 - 0.06M_5 - 0.06M_6$
YZ_2
$ZZ_2 - 0.18M_2 - 0.18M_3 - 0.18M_4 - 0.18M_5 - 0.18M_6$
$MX_2 + 0.43M_2 + 0.43M_3 + 0.43M_4 + 0.43M_5 + 0.43M_6$
MY_2
fv_2
fc_2

Joint 3:
$XY_3 - 0.02MY_3$
XZ_3
$YY_3 + ZZ_4 + 0.86MY_4 - 0.0004M_3 + 0.18M_4 + 0.18M_5 + 0.18M_6$
YZ_3
$MX_3 + 0.02M_3 + 0.02M_4 + 0.02M_5 + 0.2M_6$
$MZ_3 + MY_4 + 0.43M_4 + 0.43M_5 + 0.43M_6$
fv_3

Joint 4:
$XX_4 - ZZ_4 + ZZ_5$
XY_4
XZ_4
$YY_4 + ZZ_5$
YZ_4
MX_4
$MZ_4 - MY_5$
fv_4
fc_4

Joint 5:
$XX_5 - ZZ_5 + YY_6$
XY_5
XZ_5
$YY_5 + YY_6$
YZ_5
MX_5
$MZ_5 + MZ_6$
fv_5
fc_5

Joint 6:
$XX_6 - YY_6$
XY_6

XZ_6
YZ_6
ZZ_6
MX_6
MY_6
fv_6
fc_6

As stated in Section 2-1, Lagrange-Euler and Newton Euler are the standard approaches for obtaining the matrix W which is a function of joint position, velocity and acceleration. Moreover, the algorithm for obtaining the matrix W is quite complex and not practical to derive for a 6 DOF robot by hand as stated in [14] [12]. To overcome this, an open source software named Sympybotics is used to compute the observation matrix W. Later, the observation matrix W is transformed to ϕ_b by considering only the columns corresponding to the reduced parameter set (χ_b).

A-2 Sympybotics

Sympybotics is used for symbolic framework for modeling and identification of robot dynamics. The software runs on a linux platform and requires Sympy and Numpy libraries before installation. The ways to install could be found in the link described in the footnote [1]. After installing the software, open the terminal in linux and start the python installed in the computer. The software takes the Denavit Hartenberg convention of the manipulator as input. The code used for the generating the regressor matrix for PUMA 560 robot is given below.

```
import sympy
import numpy
import sympybotics
pi = sympy.pi
q = sympybotics.robotdef.q
puma560_def = sympybotics.RobotDef('Puma 560 Robot',
                [('pi/2',0,0,'q'),
                 (0,0.4318,0,'q'),
                 ('-pi/2','0.0203', '0.15005', 'q'),
                 ('pi/2',0,0.4318,'q'),
                 ('-pi/2',0,0,'q'),
                 (0,0,0,'q')],
                 dh_convention='standard')
puma560_def.gravityacc = sympy.Matrix([0.0, 0.0, -9.81])
puma560 = sympybotics.RobotDynCode(puma560_def)
W_Regressor = sympybotics.robotcodegen.robot_code_to_func(
        'C', puma560.H_code, 'H', 'H_puma560', puma560_def)
print(W_Regressor)
```

After running the above code, the regressor matrix W will be generated in C format.

[1] https://github.com/cdsousa/SymPyBotics

Appendix B

Gaussian Process Regression

Gaussian process regression is basically a non-parametric kernel-based method [69]. For a given training set with unknown distributions $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$, the GPR model predicts a response y_* for a new input vector x_*. Generally, the Gaussian process does these predictions by introducing latent variables, $f(x_i), i = 1, 2, ..., n$, and explicit basis functions h. The covariance function of the latent variables called the kernel function helps in capturing the smoothness of the response and basis functions help in projecting the inputs x into a p-dimensional feature space.

In general, Gaussian process is a collection of random variables with finite number of them having a joint Gaussian distribution. "*If $f(x), x \in \mathbb{R}^d$ is a GP, then given n observations $x_1, x_2, ..., x_n$, the joint distribution of the random variables $f(x_1), f(x_2), ..., f(x_n)$ is a Gaussian*" [69]. A Gaussian process can be represented as $f(x) \sim \mathcal{GP}(m(x), k(x_p, x_q))$ where the covariance function $k(x_p, x_q) = E[(f(x_p) - m(x_p))(f(x_q) - m(x_q))]$ and its mean function $m(x) = E[f(x)]$. With this, a Gaussian process model can be given as

$$y = h(x)^T \beta + f(x), \tag{B-1}$$

where $f(x)$ is a Gaussian process with zero mean and kernel function $k(x, x')$, $h(x)$ is set of basis functions that transform the original feature vector x in $\mathbb{R}^d$ into a new feature vector $h(x)$ in $\mathbb{R}^p$. β is a p-by-1 vector of basis function coefficients. Since GPR is a non-parametric probabilistic model, the posterior distribution of latent function $f(x)$ for a given set of outputs and inputs ($p(f(X)|y)$ is obtained by using the Baye's rule and it is given by the following equations

$$\text{Posterior} = p(f(X)|y) = \frac{p(y|f(X))p(f(X))}{p(y)}, \tag{B-2}$$

$$p(y|f, X) \sim \mathcal{N}(y|H\beta + f, \sigma^2 I), \tag{B-3}$$

$$p(f(X)) = \mathcal{N}(0, K(X, X)), \tag{B-4}$$

$$p(y) = \int p(f(X))p(y|f(X))dX. \tag{B-5}$$

It can be inferred from Equation (B-2) that the posterior distribution is proportional to the product of the prior ($p(f(X))$) and the likelihood ($p(y|f(X))$. Also, it is to be noted from Equations (B-3) and (B-4) that prior and likelihood are Gaussian distributed, which makes the posterior to be Gaussian distributed as well. The joint probability of the training data set and the query point is defined to get the predictions $\bar{f}(x_*)$ for the new query point x_* and it is represented as follows

$$\begin{bmatrix} y \\ \bar{f}(x_*) \end{bmatrix} \sim \mathcal{GP}\Big(\begin{bmatrix} m(X) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & k(X,x_*) \\ K(x_*,X) & k(x_*,x_*) \end{bmatrix} \Big), \tag{B-6}$$

where $K(X,X)$ is the covariance matrix built with the training data set as given in Equation (B-9) and $K(X,x_*)$ is the covariance calculated on the training set X and the new query point x_*. Likewise, the mean predicted value $\bar{f}$ and the variance V for every query point can be predicted by conditioning the joint probability depicted in Equation (B-6). Consequently, the obtained mean and variance is represented in Equation (B-7), (B-8).

$$\bar{f}(x_*) = k(x_*^T, X)(K(X,X) + \sigma_n^2 I)^{-1}(y - H(\beta)) \tag{B-7}$$

$$V(x_*) = k(x_*, x_*) - k(x_*^T, X)(K(X,X) + \sigma_n^I)^{-1} k(X, x_*). \tag{B-8}$$

$$K(X,X) = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & . & . & . & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & . & . & . & k(x_2,x_n) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ k(x_n,x_1) & k(x_n,x_2) & . & . & . & k(x_n,x_n) \end{bmatrix} \tag{B-9}$$

Appendix C

Observations

C-1 Amplitude Values of the Fourier Series

The amplitude values $a_{i,l}, b_{i,l}$ of the Fourier series obtained after optimizing the criterion the cond(W_b) for the PUMA 560 and the twolink manipulator is given in Table C-1, C-2 respectively.

Table C-1: Amplitude Values of Fourier Series obtained after optimization for the PUMA-560 robot

Variable	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
a_1	0.88	1.20	0.0008	-0.28	-0.09	-1.44
a_2	0.26	-0.13	0.0006	0.41	-0.026	0.19
a_3	-0.54	0.41	0.066	-0.88	-0.86	0.26
a_4	-0.76	0.21	0.11	-0.99	-0.22	-1
b_1	-0.57	0.27	-0.63	0.24	0.75	0.091
b_2	0.26	0.28	-0.30	-0.067	0.74	1.33
b_3	0.23	-1.89	0.56	0.92	0.20	1.08
b_4	0.11	-0.29	0.17	0.44	-0.44	-0.54

C-2 Case 1 : Correct IDM Model

The PUMA 560 and the twolink manipulator are compensated by a correct inverse dynamics model. But, they fail to capture the noises occurring in the robot. A semi-parametric Gaussian process regression was incorporated to capture the un-modeled dynamics and other perturbations with obtained the RBD model as mean. Subsequently, the tracking performance of the obtained semi-parametric model for the PUMA 560 and the twolink manipulator under training and test set is plotted, which can be visualized from the Figures C-1, C-2, C-3, C-4, C-5, C-6, C-7, C-8, C-9, C-10, C-11, C-12, C-13, C-14, C-15, and C-16.

Table C-2: Amplitude Values of Fourier Series obtained after optimization for the two-link manipulator

Variable	Joint 1	Joint 2
a_1	0.15	-0.25
a_2	-2.80×10^{-5}	-6.59×10^{-5}
a_3	-0.21	0.22
b_1	8.96×10^{-5}	0.00021
b_2	0.0648	0.029
b_3	-5.04×10^{-5}	-0.00011

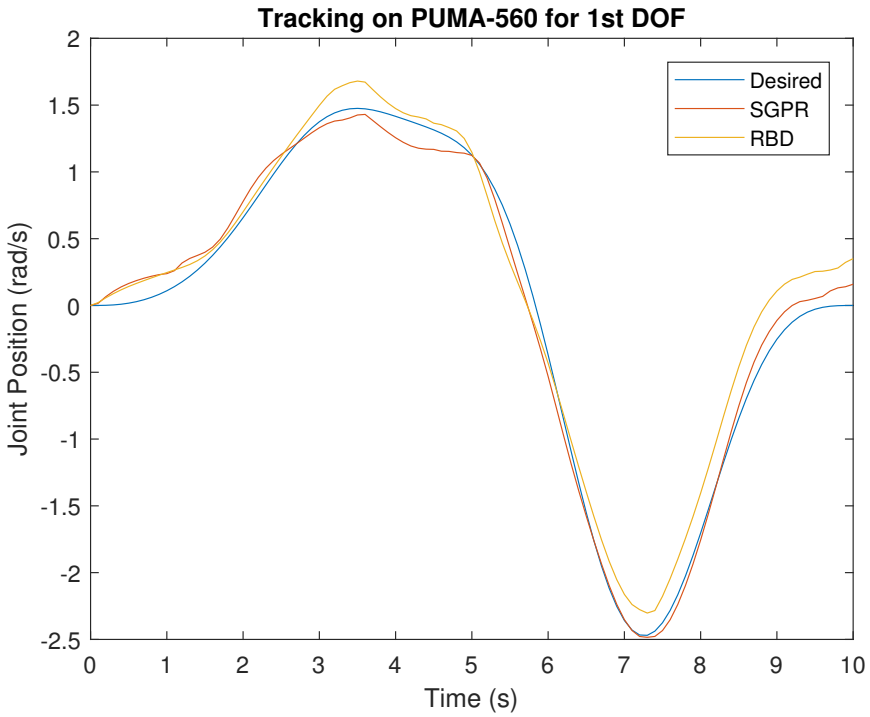

Figure C-1: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 1st DOF PUMA 560 robot

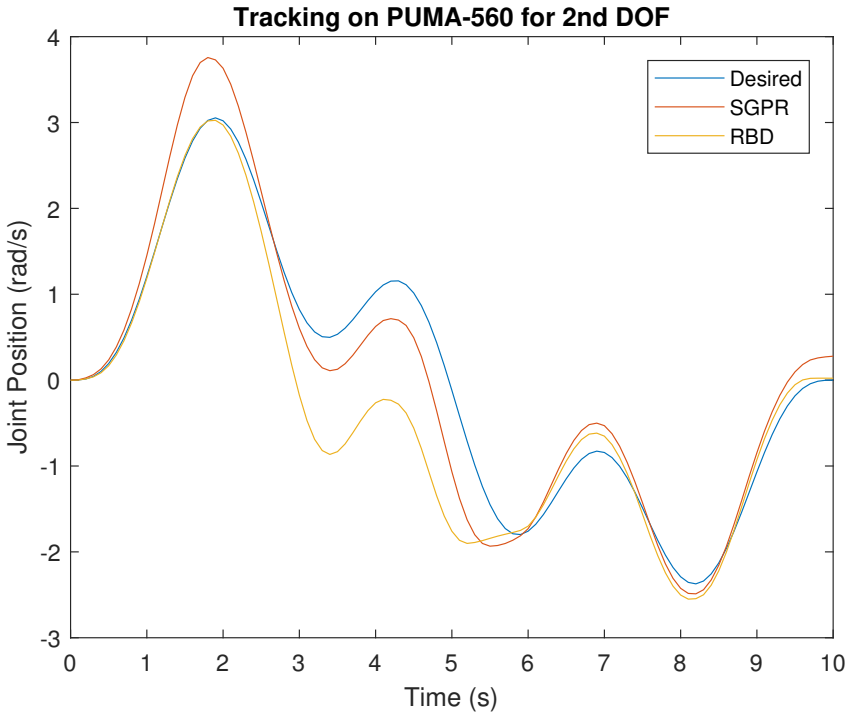

Figure C-2: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 2nd DOF PUMA 560 robot

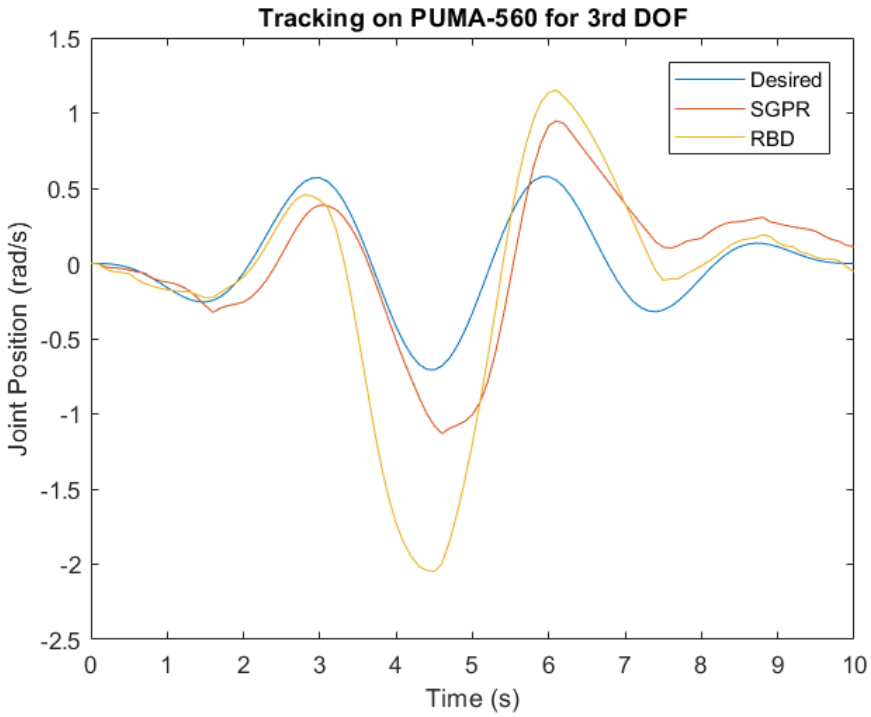

Figure C-3: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 3rd DOF PUMA 560 robot

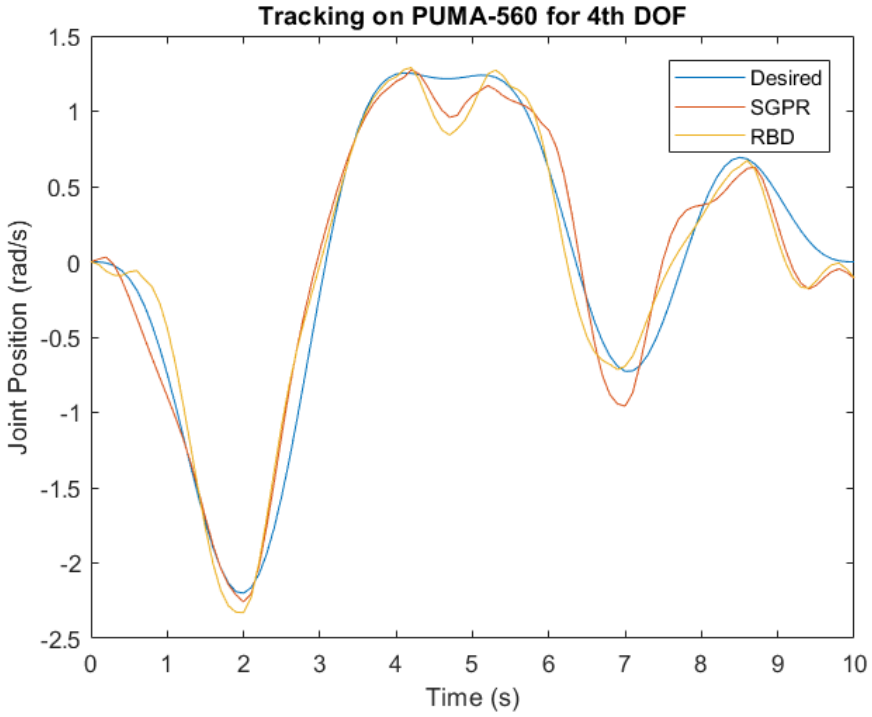

Figure C-4: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 4th DOF PUMA 560 robot

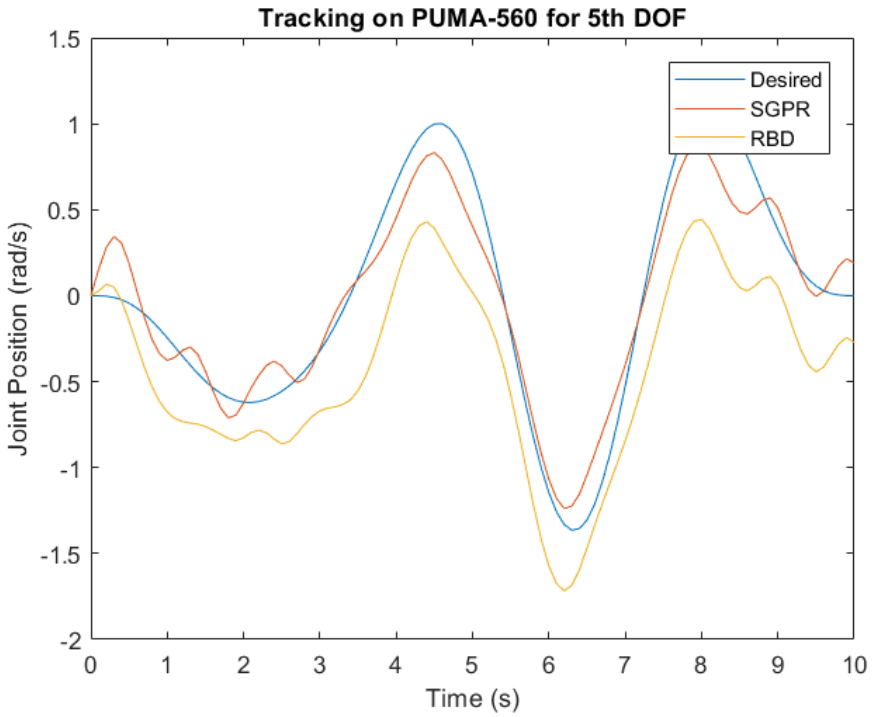

Figure C-5: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 5th DOF PUMA 560 robot

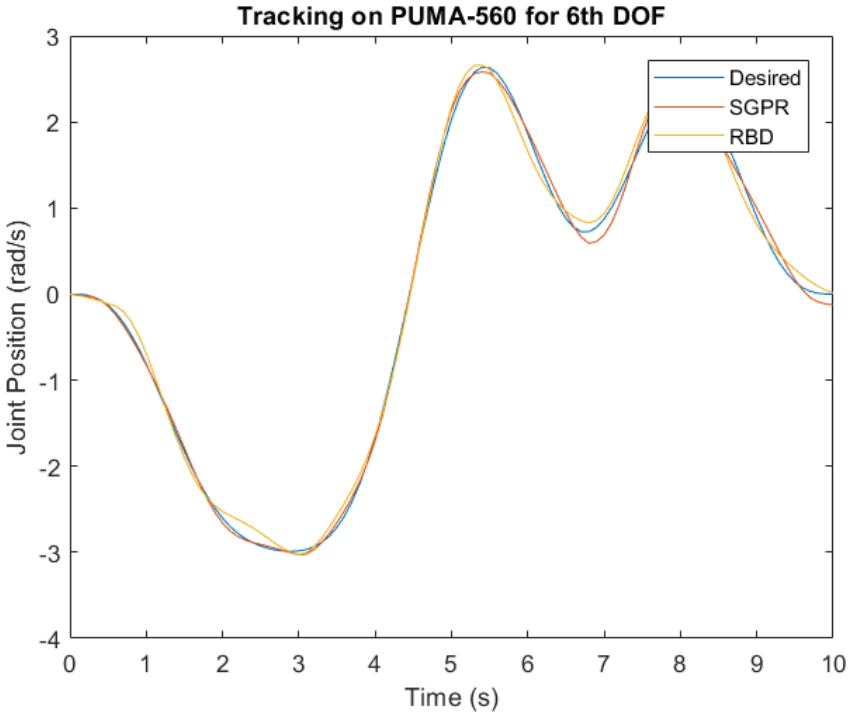

Figure C-6: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 6th DOF PUMA 560 robot

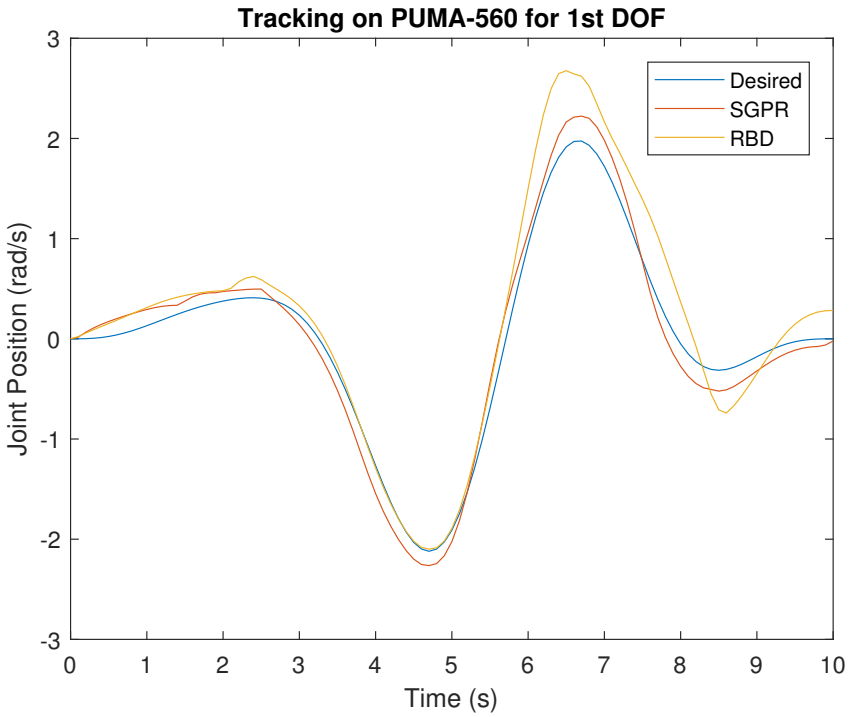

Figure C-7: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 1st DOF PUMA 560 robot

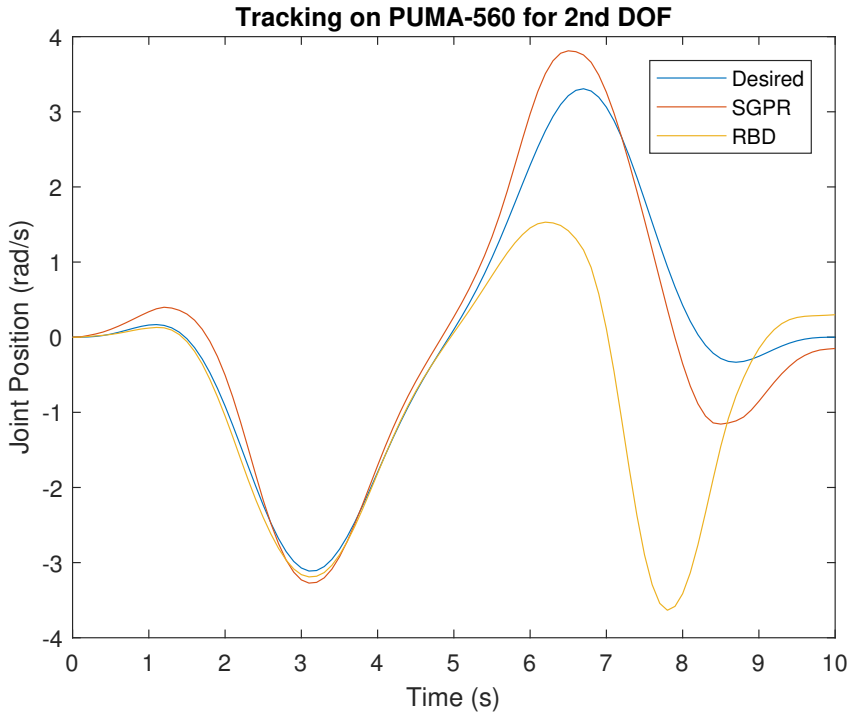

Figure C-8: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 2nd DOF PUMA 560 robot

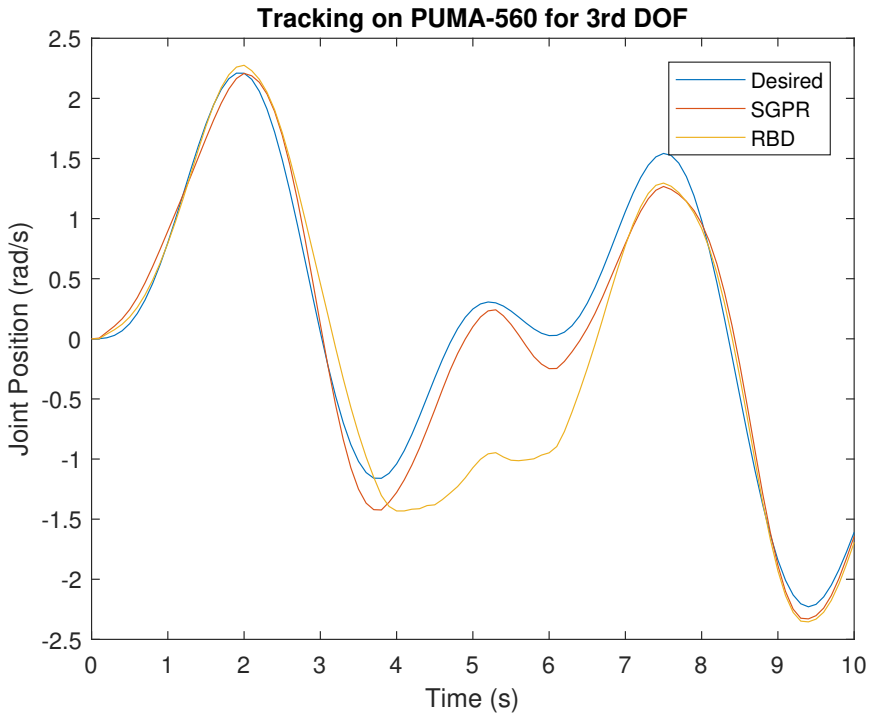

Figure C-9: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 3rd DOF PUMA 560 robot

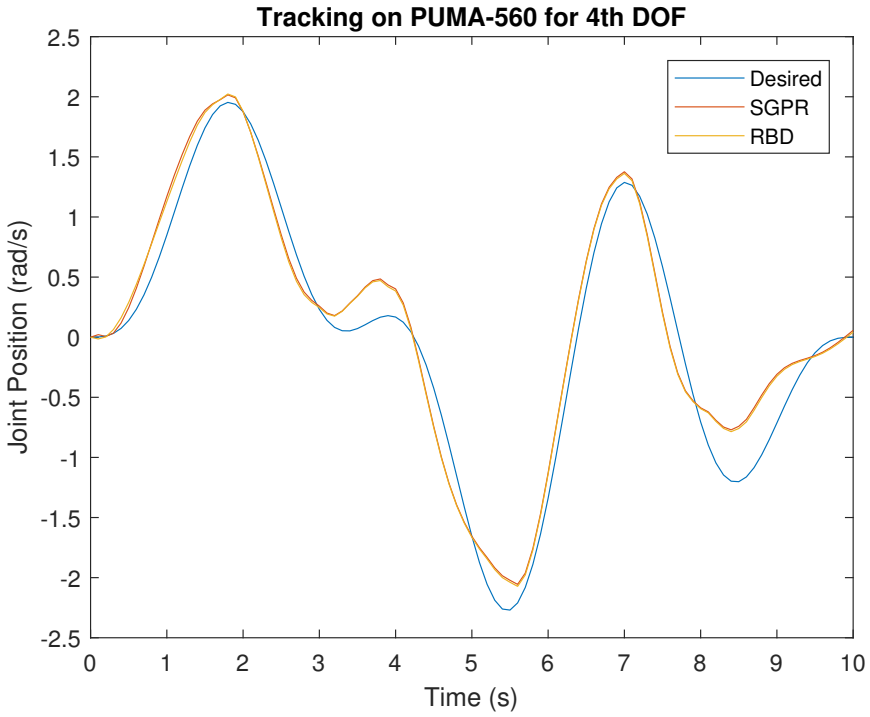

Figure C-10: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 4th DOF PUMA 560 robot

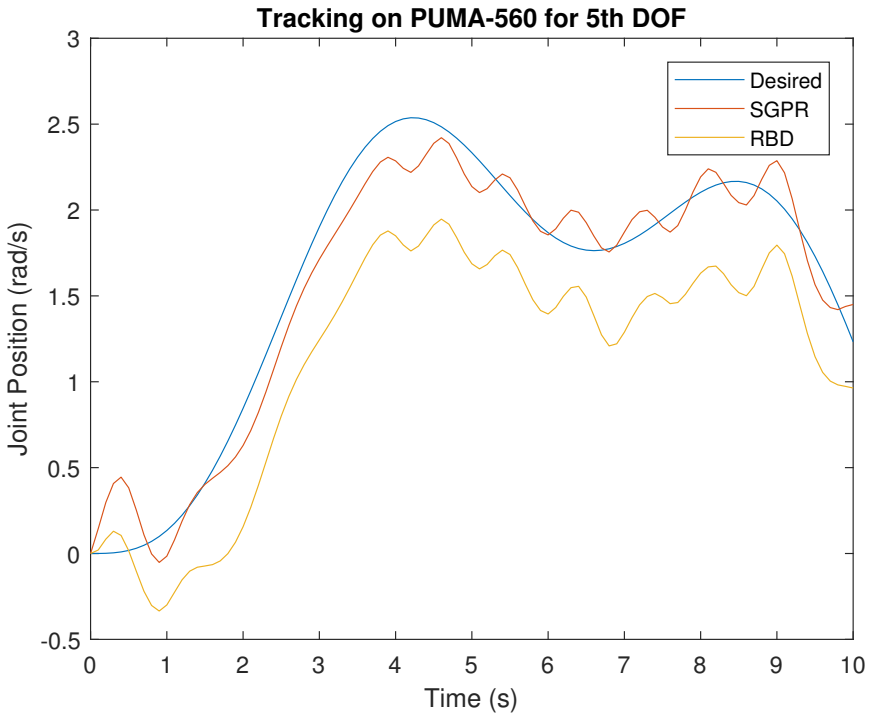

Figure C-11: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 5th DOF PUMA 560 robot

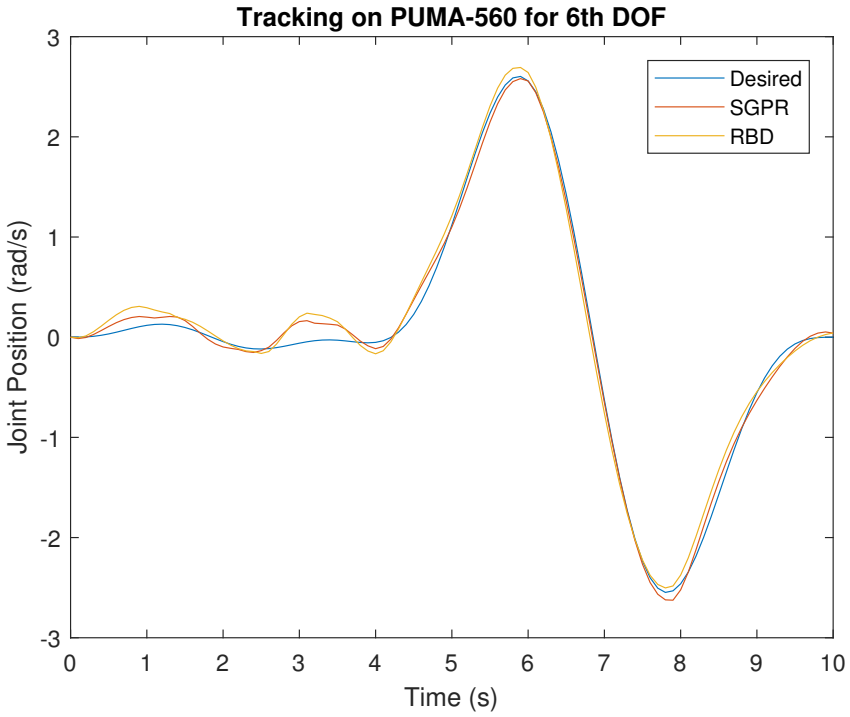

Figure C-12: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 6th DOF PUMA 560 robot

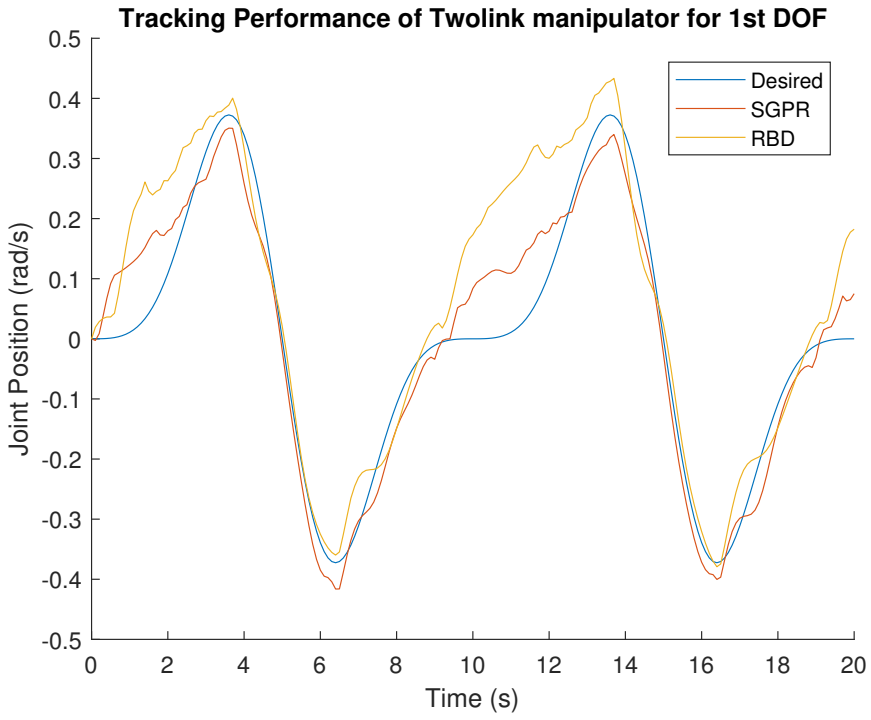

Figure C-13: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 1st DOF twolink robot

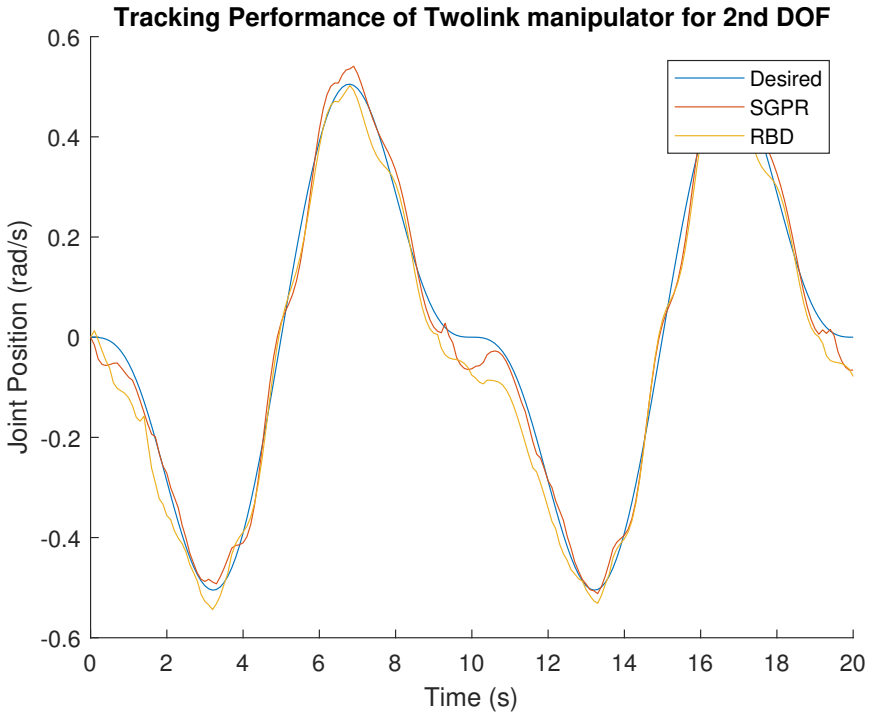

Figure C-14: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the training set for the 2nd DOF twolink robot

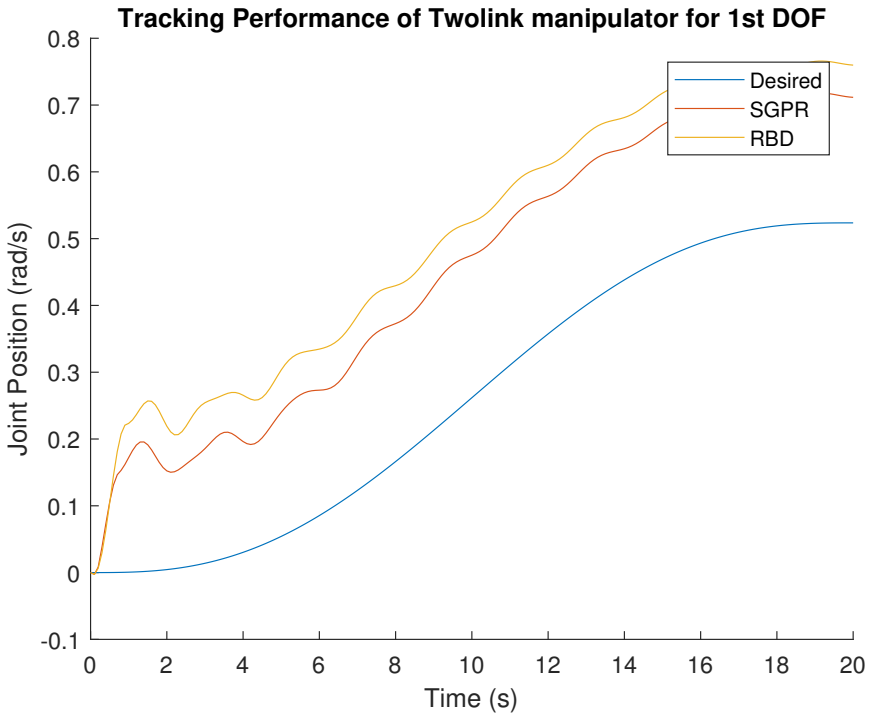

Figure C-15: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 1st DOF twolink robot

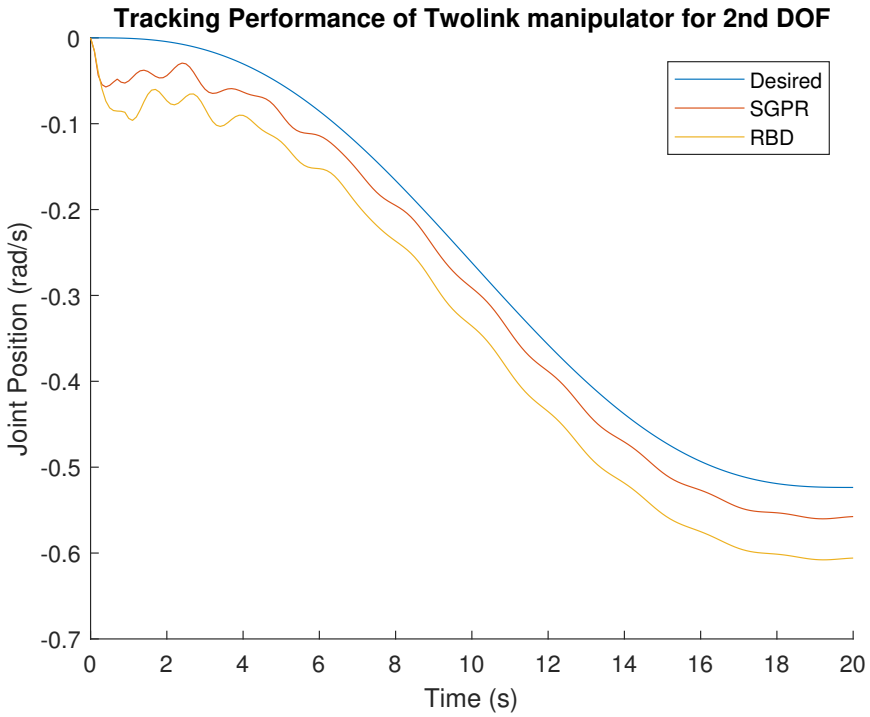

Figure C-16: Tracking performance of the semi-parametric model (SGPR), estimated RBD model from Least Squares (RBD) under the test set for the 2nd DOF twolink robot

C-3 Case 2 : Incorrect RBD Model

The twolink manipulator is compensated by an incorrect inverse dynamics model. In addition, it failed to capture the noises occurring in the robot. To tackle the issue, first an error model was identified using Least Squares approach to capture the model inaccuracies. Later, the error model was used as mean to capture the remaining un-modeled dynamics and other perturbations using GPR. Subsequently, the tracking performance of the obtained semi-parametric model under training and test set is plotted, which can be visualized from the Figures C-17, C-18, C-19 and C-20.

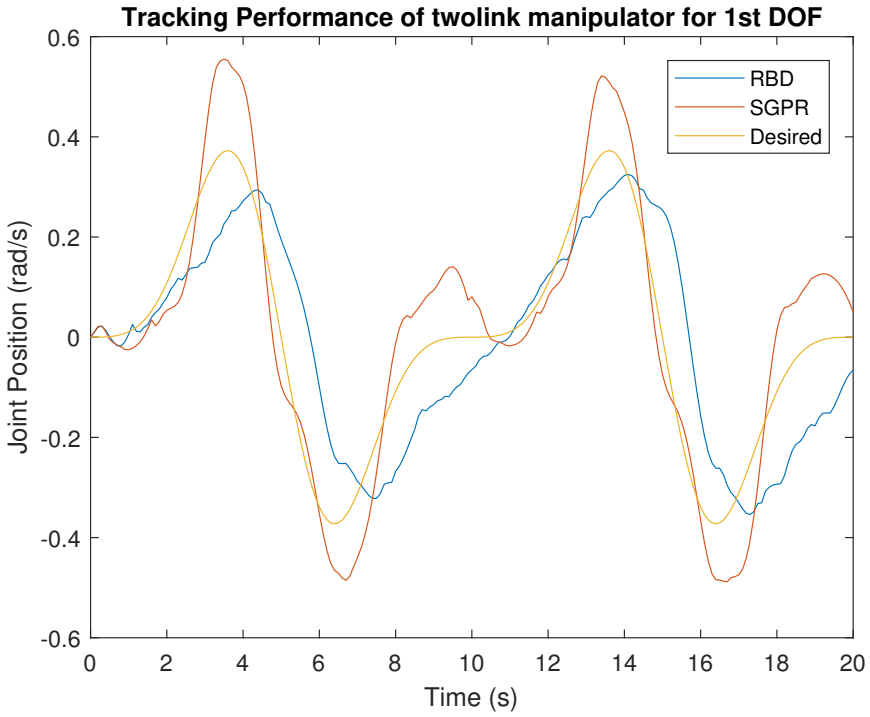

Figure C-17: Tracking performance of the semi-parametric model (SGPR), inaccurate internal IDM (RBD) under the training set for the 1st DOF twolink robot

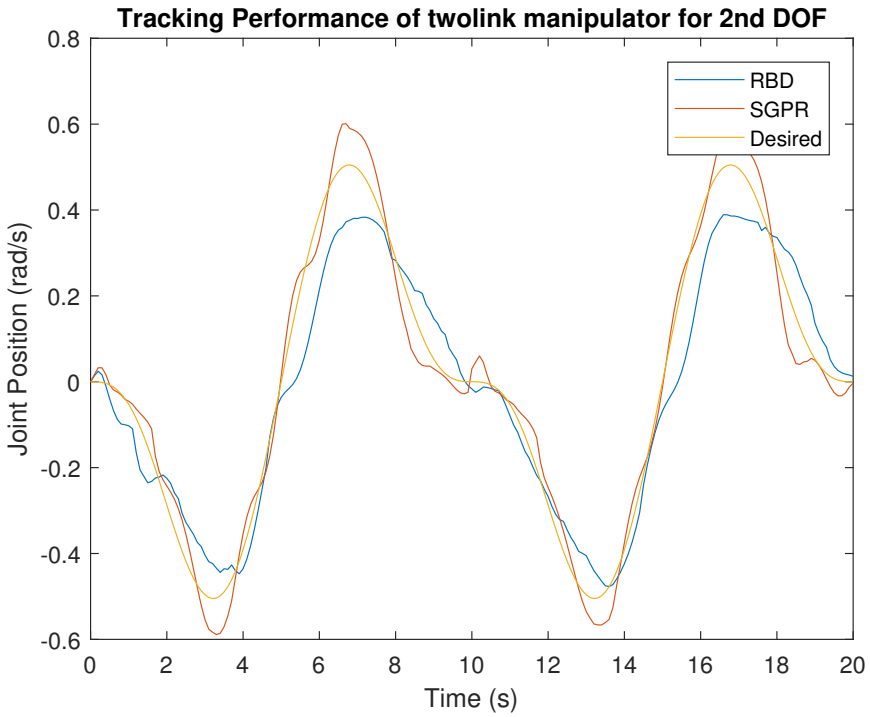

Figure C-18: Tracking performance of the semi-parametric model (SGPR), inaccurate internal IDM (RBD) under the training set for the 2nd DOF twolink robot

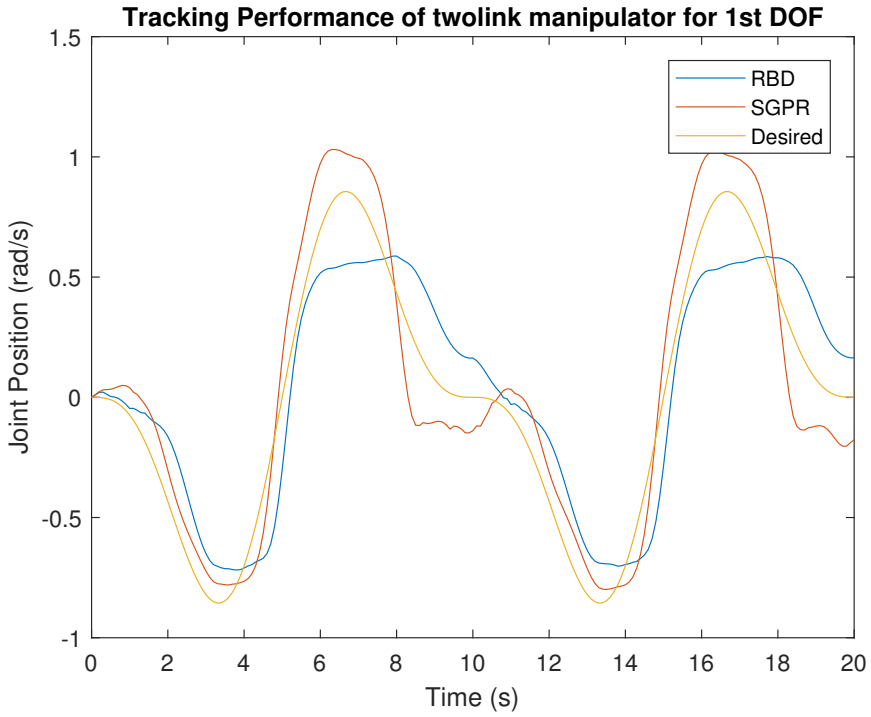

Figure C-19: Tracking performance of the semi-parametric model (SGPR), inaccurate internal IDM (RBD) under the test set for the 1st DOF twolink robot

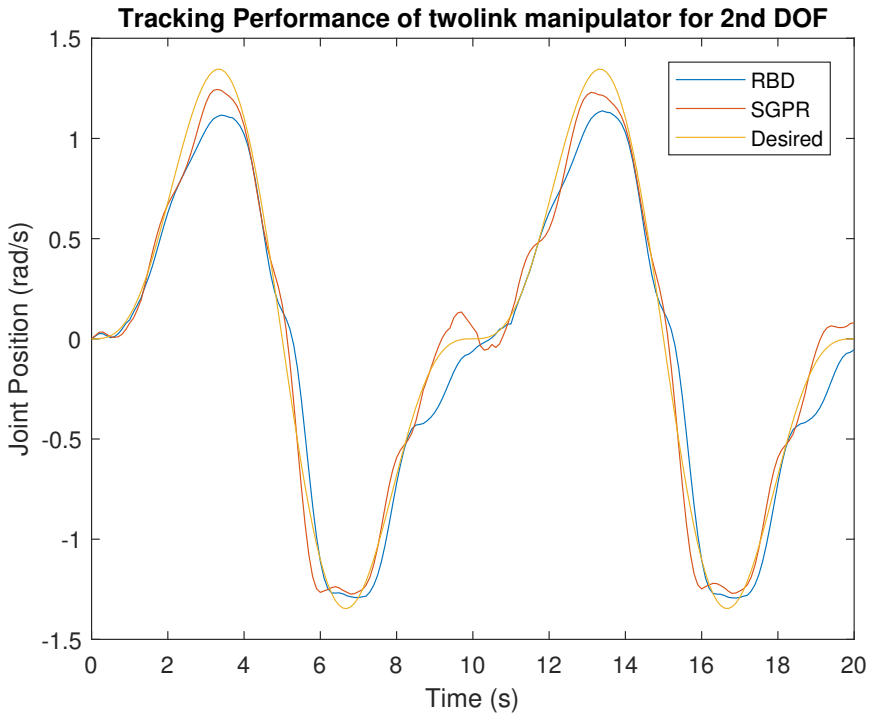

Figure C-20: Tracking performance of the semi-parametric model (SGPR), inaccurate internal IDM (RBD) under the test set for the 2nd DOF twolink robot

Bibliography

[1] T.M. Anandan, "The Business of Automation, Betting on Robots." http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/The-Business-of-Automation-Betting-on-Robots/content_id/6076, 2016.

[2] C. Heyer, "Human-Robot Interaction and Future Industrial Robotics Applications", in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Taiwan, 2010.

[3] International Federation of Robotics, "Executive Summary World Robotics 2016 Industrial Robots", http://www.ifr.org/industrial-robots/statistics/, 2016.

[4] M. Bélanger-Barrette, "Collaborative Robot Ebook: Sixth Edition." http://blog.robotiq.com/collaborative-robot-ebook, 2015.

[5] M. Damme, P. Beyl, B. Vanderborght, R. Ham, I. Vanderniepen, A. Matthys, P. Cherelle, D. Lefeber, "The Role of Compliance in Robot Safety"

[6] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, G. Hirzinger, "The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing" in IEEE, Berlin, 2010

[7] "http://www.eucognition.org/eucog-wiki/Compliant_robots"

[8] C. Wit, B. Siciliano and G. Bastin, "Theory of Robot Control", in Springer, London, 1996.

[9] M. Verhargen and V. Verdult. "Filtering and System Identification - A Least Squares App", Delft, Netherlands 2007.

[10] H. Hoifodt, "Dynamic Modeling and Simulation of Robot Manipulators - The Newton-Euler Formulation", Master thesis in Norwegian University of Science and Technology, Norway, 2011.

[11] W. Khalil and E. Dombre, "Modeling,Identification and Control of Robots", London, 2002.

[12] E. Villagrossi, "Robot Dynamics Modelling and Control for Machining Applications", P.H.D thesis at Universitá degli Studi di Brescia, Italy, 2016.

[13] W. Khalil, A. Vijayalingam, B. Khomutenko, I. Mukhanov, P. Lemoine, G. Ecorchard, "OpenSYMORO: An open-source software package for Symbolic Modelling of Robots" in IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM), France, 2014.

[14] W. Khalil, "Dynamic Modelling of Robots Using Recursive Newton-Euler Techniques", in "ICINCO2010, Portugal, 2010.

[15] P. Khosla, T. Kanade, "Parameter identification of robot dynamics" in 24^{th} Conf. on Decision and Control, USA, 1985.

[16] B. Armstrong, "On Finding Exciting Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics", in the Int. Journal of Robotics Research, USA, 1989.

[17] V. Bonnet, P. Fraisse, A. Crosnier, M. Gautier, A. Gonzalez, and G. Venture, "Optimal Exciting Dance for Identifying Inertial Parameters of an Anthropomorphic Structure", IEEE Transactions on Robotics, August 2016.

[18] A. Hayat, V. Abhishek, S. Saha, "Dynamic Identification of Manipulator: Comparison between CAD and Actual Parameters" in 2^{nd} int. and 17^{th} National Conf. on Machines and Mechanisms, Kanpur, 2015.

[19] U. Forssell and L. Ljung, "Closed loop Identification revisited", in Automatica, Sweden, 1999.

[20] http://www.dcsc.tudelft.nl/ discsysid/DISC2012-6-4up.pdf

[21] M. Ostring, S. Gunnarsson and M. Norrlof, "Closed-loop identification of an industrial robot containing flexibilities" from Elsevier Science Ltd, Sweden, 2002.

[22] B. Bona and A. Curatella, "Identification of Industrial Robot Parameters for Advanced Model-Based Controllers Design", in IEEE int. Conf. on Robotics and Automation, Spain, 2005.

[23] L. Sun, H. Ohmori and A. Sano, "Direct Closed-Loop Identification Approach to Unstable Plant" in IEEE Conference on Decision and Control, Sydney, Australia, 2000, sec. 7, pp. 1684-1685.

[24] W. Wu, S. Zhu, X. Wang and H. Liu, "Closed-Loop Dynamic Parameter Identification of Robot Manipulators Using Modified Fourier Series" in Int. Journal of Advanced Robotic Systems, China, 2012.

[25] A. Janot, P.O. Vandanjon and M. Gautier, "Identification of 6 DOF Rigid Industrial Robots with the Instrumental Variable Method" in 16th IFAC Symp. on System Identification, The Int. Federation of Automatic Control Brussels, Belgium, 2012.

[26] M. Gautier, A.Janot and P.O.Vandanjon, "A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics" in IEEE Trans. on Control Systems Technology, 2013.

[27] M. Gautier, A. Janot, and P.O. Vandanjon, "DIDIM: A New Method for the Dynamic Identification of Robots from only Torque Data" in 2008 IEEE Int. Conf. on Robotics and Automation, USA, 2008.

[28] C. Gaz F. Flacco A.D. Luca, "Identifying the Dynamic Model Used by the KUKA LWR: A Reverse Engineering Approach", in IEEE Int. Conference on Robotics and Automation, China, 2014.

[29] M. Gautier and S. Briot, "Dynamic Parameter Identification of a 6 DOF Industrial Robot using Power Model" in 2013 IEEE Int. Conf. on Robotics and Automation (ICRA), Germany, 2013.

[30] M. Gautier and P. Poignet, "Extended Kalman filtering and Weighted Least Squares Dynamic Identification of Robot" in Control Engineering Practice 9, France, 2001.

[31] P. Poignet and M. Gautier, "Comparison of Weighted Least Squares and Extended Kalman Filtering Methods for Dynamic Identification of Robots" in IEEE Int. Conf. on Robotics and Automation, USA, 2000.

[32] A. Jubien, M. Gautier, and A. Janot, "Dynamic Identification of the KUKA Light Weight Robot: Comparison between actual and confidential KUKA's parameters" in IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM), France, 2014.

[33] V. Bargsten, P. Zometa and R. Findeisen, "Modeling, Parameter Identification and Model-Based Control of a Lightweight Robotic Manipulator", in IEEE Int. Conf. on Control Applications (CCA), India, 2013.

[34] A. Jubien, M. Gautier and A. Janot, "Dynamic Identification of the KUKA LWR Robot using Motor Torques and Joint Torque Sensor Data", in Preprints of the 19^{th} World Congress the International Federation of Automatic Control, South Africa, 2014.

[35] N.D. Vuong and M.H. Ang, "Dynamic Model Identification for Industrial Robots", Acta Polytechnica Hungarica, 2009.

[36] J. Swevers, C. Ganseman and J.de. Schutter, and H.V. Brussel, "Experimental Robot Identification using optimized periodic Trajectories", Mechanical Systems and Signal processing, 1996.

[37] V. Mata, F. Benimeli, N. Farhat, and A. Valera, "Dynamic Parameter Identification in Industrial Robots considering physical Feasibility", Advanced Robotics, 2005.

[38] Garling DJH, "Inequalities:a journey into linear analysis" Cambridge University Press, UK, 2007

[39] G. H. Golub and C. F. Van Loan, "Matrix Computations" The Johns Hopkins Univ. Press, Baltimore, 1989.

[40] V. Bargsten, P. Zometa, and R. Findeisen, "Modeling, Parameter Identification and Model-Based Control of a Lightweight Robotic Manipulator", in IEEE Int. Conf. on Control Applications (CCA), Hyderabad, India, 2013.

[41] J. Jin and N. Gans, "Parameter Identification for Industrial Robots with fast and robust Trajectory Design Approach", Robotics and Computer-Integrated Manufacturing, 2015.

[42] C.D. Sousa and R. Cortesão, "Physical Feasibility of Robot Base Inertial Parameter Identification: A linear Matrix Inequality Approach", The International Journal of Robotics Research, 2014.

[43] J. Swevers, C. Ganseman, J. Schutter, and H.V. Brussel, "Experimental Robot Identification using optimized periodic Trajectories", Mechanical Systems and Signal processing, 1996.

[44] J. Swevers, C. Ganseman, D.B.Tükel, J. Schutter and H.V. Brussel, "Optimal Robot Excitation and Identification" in IEEE Transactions on Robotics and Automation, 1997.

[45] J. Swevers, W. Verdonck and J. Schutter, "Dynamic Model Identification for Industrial Robots" in IEEE Control Systems Magazine, 2007.

[46] J. Swevers, W. Verdonck, B. Naumer, S. Pieters, and E. Biber, "An Experimental Robot Load Identification Method for Industrial Application", The Int. Journal of Robotics Research, 2002.

[47] M. Gautier, W. Khalil, "Exciting Trajectories for the Identification of Base Inertial Parameters of Robots" in Int. Journal of Robotics Research, USA, 1992.

[48] N.L.D. Marck, "Semi-parametric identification of manipulator dynamics in a time varying environment", Master thesis at TU Delft, Netherlands, 2017

[49] L.M. Affolter, "Parameter Identification of a 7-DOF Robot Manipulator Considering Physical Feasibility", Master thesis at ETH Zurich, Switzerland, 2016.

[50] G. Calafiore, M. Indri, and B. Bona, "Robot Dynamic Calibration: Optimal Excitation Trajectories and Experimental Parameter Estimation", Journal of Robotic Systems, 2001.

[51] K. Park, "Fourier-based optimal Excitation Trajectories for the Dynamic Identification of Robots", in Robotica, 2006.

[52] W. Rackl, R. Lampariello, and G. Hirzinger, "Robot Excitation Trajectories for Dynamic Parameter Estimation using optimized B-Splines", in IEEE Int. Conf. on Robotics and Automation, USA, 2012.

[53] L. Biagiotti and C. Melchiorri, "Trajectory Planning for Automatic Machines and Robots", Springer-Verlag Berlin Heidelberg, 2008.

[54] F.D. Ledezma and S. Haddadin, "First-Order-Principles-Based Constructive Network Topologies: An Application to Robot Inverse Dynamics" in IEEE-RAS 17th Int. Conf. on Humanoid Robotics, UK, 2017.

[55] F. Saupe and A. Knoblach, "Design of Excitation Signals for the Closed Loop Identification of Industrial Robots", in IEEE INt. Conf. on Control Applications (CCA), Croatia, 2012.

[56] A. Janot, P. Vandanjon and M. Gautier, "A Generic Instrumental Variable Approach for Industrial Robot Identification" in IEEE Transactions on Control Systems Technology, 2014.

[57] N. Farhat, M. Rodriguez and V. Mata, "Dynamic Parameter Identification of Parallel Robots Considering Physical Feasibility and Nonlinear Friction Models" in 12^{th} IFToMM World Congress, France, 2007.

[58] B. Sciliano, L. Sciavicco, L. Villani and G. Oriolo, "Modelling, Planning and Control", in Springer, Glasgow, 2008.

[59] A.A. Schäffer, C. Ott and G. Hirzinger, "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots" in the Int. Journal of Robotics Research, Germany, 2007.

[60] C. Ott, "Cartesian Impedance Control of Redundant and Flexible-Joint Robots", in springer tracts in advanced robotics, vol. 49, Japan : Springer, 2008, chap - 2,5,6,7

[61] M. Östring, "Identification, Diagnosis, and Control of a Flexible Robot Arm", Thesis No. 948, Linköping Studies in Science and Technology, Sweden, 2002.

[62] G. Testa, "Experimental stiffness identification in the joints of a lightweight robot", master thesis at Universitat Politécnica de Catalunya (UPC), Spain, 2017.

[63] D.N. Tuong, J. Peters and M. Seeger, "Local Gaussian Process Regression for Real Time Online Model Learning and Control", Max Planck Institute for Biological Cybernetics, Germany.

[64] J. Peters, K. Mülling, J. Kober, D.N. Tuong and O. Krömer, "Robot Skill Learning",2012.

[65] D.N. Tuong and J. Peters, "Using Model Knowledge for Learning Inverse Dynamics",2012.

[66] D.N. Tuong and J. Peters, "Incremental Online Sparsification for Model Learning in Real-time Robot Control",Max Planck Institute for Biological Cybernetics, TÃijbingen.

[67] D.N. Tuong and J. Peters, "Model learning for robot control: a survey", in Springer-Verlag, 2010.

[68] D.N. Tuong, M. Seeger, and J. Peters, "Model Learning with Local Gaussian Process Regression", Advanced Robotics, vol. 23, no. 15, pp. 2015âĂŞ2034, 2009.

[69] C.E. Rasmussen, C.K.I. Williams, "Gaussian Processes for Machine Learning", the MIT Press, 2006.

[70] B. Schölkopf and A. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond" in MIT-Press, Cambridge, 2002.

[71] D.N. Tuong, J. Peters, M. Seeger, B. Schölkopf,"Learning Inverse Dynamics: a Comparison", in ESANN, Germany, 2008.

[72] F. Abdessemed, "SVM-Based Control System for a Robot Manipulator", in INTECH, Algeria, 2012.

[73] S. Vijayakumar and S. Schaal, "Locally Weighted Projection Regression : An O(n) Algorithm for Incremental Real Time Learning in High Dimensional Space", in ICML, USA, 2000.

[74] D.N. Tuong, M. Seeger, J. Peters, "Computed torque control with nonparametric regression models", in IEEE American Control Conference, USA, 2008.

[75] D. Kappler, F. Meier, N. Ratliff and S. Schaal, "A New Data Source for Inverse Dynamics Learning", Cornell Unveristy Library (arvix.org), USA, 2017.

[76] N. Ratliff, F. Meier, D. Kappler and S. Schaal, "Direct Online Optimization of Modeling Errors in Dynamics", Cornell Unveristy Library (arvix.org), USA, 2016.

[77] D. Kappler, F. Meier, N. Ratliff and S. Schaal, "Towards Robust Online Inverse Dynamics Learning" in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, USA, 2016.

[78] KUKA Robot Group. KUKA Sunrise.Connectivity FRI 1.7. Zugspitzstrasse 140, Augsburg, 1st edition,2015.

[79] P. Corke and B. Armstrong, "A Search for Consensus Among Model Parameters Reported for the PUMA 560 Robot" in IEEE, USA, 1994.

[80] P.I. Corke, "Robotics, Vision and Control", Springer 2017.

[81] R.O. Mohammed and G.C. Cawley, "Over-Fitting in Model Selection with Gaussian Process Regression", in Springer Int. Publishing AG, UK, 2017.

[82] G.C. Cawley and N.L.C. Talbot, "On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation", in Journal of Machine Learning Research, UK, 2010.

Glossary

List of Acronyms

COBOT	COllaborative industrial roBOT
RBD	Rigid Body Dynamics
DH	Denavit Hartenberg
SYMORO	SYmbolic MOdeling of RObots
FS	Fourier series
WLS	Weighted Least Squares
EKF	Extended Kalman Filter
MLE	Maximum Likelihood Estimation
EKF	Extended Kalman Filter
GPR	Gaussian Process Regression
IDM-LS	Inverse Dynamic Model - Least Squares
LWPR	Locally Weighted Projection Regression
LGP	Local Gaussian Process

List of Symbols

α_i	Angle between Z_{j-1} and Z_j about X_{j-1}
χ	Dynamic parameters
χ_b	Reduced base parameter set
$\ddot{q}_{i,max}$	Maximum joint acceleration

$\ddot{q}$	Joint acceleration
$\dot{q}$	Joint velocity
$\dot{q}_{i,max}$	Maximum joint velocity
ϵ	Unmodelled dynamics
ρ	Error vector
σ_i	Revolute or Prismatic joint
τ	Joint torque
τ_f	Joint friction torque
τ_i	Joint inertial torque
θ_i	Angle between X_{j-1} and X_j about Z_j
ζ_{max}	Maximum singular value
ζ_{min}	Minimum singular value
$a_{i,l}$	Amplitude of cosine function
$b_{i,l}$	Amplitude of sine function
$C(q, \dot{q})$	Coriolis/Centrifugal matrix
d_i	Distance between Z_{j-1} and Z_j about X_{j-1}
E	Kinetic energy
F_c	Coulomb friction
F_v	Viscous friction
$g(q)$	Gravity vector
L	Number of sine and cosine terms
M_i	Link mass
n	Number of degree of freedom
q	Joint position
$q_{i,max}$	Maximum joint position
r_i	Distance between X_{j-1} and X_j about Z_j
U	Potential energy
W	Observation matrix

Index