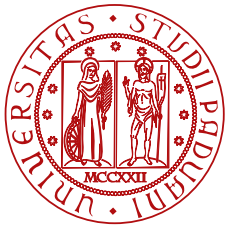

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN
Control Systems Engineering

"Robust Controller Design for Trajectory Tracking of Robot Manipulators in the Presence of Uncertain Dynamical Models "

Relatore: Prof. Carli Ruggero

Laureando: Hassan Barjoud

ANNO ACCADEMICO 2022-2023
Data di laurea 23/10/2023

To my Family and Friends

Contents

Abstract

The purpose of this activity is to design and simulate a robust controller for a robot manipulator tracking a trajectory in case of uncertainties in the dynamical model.

In this study the main goal is to introduce a way to overcome the usual assumptions made in the design of the robustness which depends on a previous knowledge of some parameters.

Matlab has been adopted as a computing environment for the development of the design. The analysis of the obtained results has been carried out through Simulink toolbox.

After the derivation of the model describing the dynamics of the robot under analysis, a simple feedback control system has been implemented to control the position of the manipulator. To enhance the robustness of the system stability, the previous structure has been subsequently improved by introducing a term that represents the robust contribution that counteracts the indeterminacy in computing the nonlinear terms that depend on the manipulator state.

To overcome the assumptions made in the design of robustness term that need previous knowledge of some parameters, a technique which depends on the evaluation of the Lyapunov function is used to determine this term.

Chapter 1

Introduction

The relationship between robotics and control theory has a long and storied history, spanning over half a century. Control theory has played a crucial role in solving fundamental challenges in robotics, while problems encountered in robotics have driven the development of new control theories. In recent years, robotics has experienced rapid progress, and the outlook for the future remains promising.

In the early days, the machine tool industry dominated robotics, leading to the design of stiff mechanisms with independent single-input/single-output (SISO) linear control for each joint. Simple tasks such as material transfer and spot welding were accomplished through point-to-point control, while more complex tasks like arc welding and spray painting utilized continuous-path tracking. However, sensing of the external environment was limited.

As the demand for more advanced tasks, such as assembly, emerged, regulating contact forces and moments became necessary. Achieving higher speed operation and payload-to-weight ratios required a deeper understanding of the complex nonlinear dynamics of robots, which, in turn, drove the development of new theoretical control approaches such as nonlinear, robust, and adaptive control.

The first industrial robot in the United States was the Unimate, installed in a General Motors plant in 1961 for moving die castings and welding parts

on auto bodies. Production began in 1966, with other companies in Japan and Europe also entering the market in the 1970s. Early robotics primarily focused on manipulator arms and basic factory automation tasks, like handling materials, welding, and painting.[33]

Progress in robot control faced challenges such as high computation costs, limited sensors, and a lack of understanding of robot dynamics. These barriers led to research efforts aimed at increasing fundamental knowledge of dynamics, architecture, and system-level design. However, early control schemes based on approximate linear models and the separation of mechanical and control system design had limitations.

The advancement of robot control was also influenced by Moore's Law, as increasing computation speed and decreasing costs enabled the implementation of advanced sensor-based control systems.

Pioneering research in robotics delved into the inventive application of well-established control techniques and the generation of novel concepts, some of which left a lasting impact on control research as a whole. Notably, early investigations into computed torque and inverse dynamics control in [25]. It's worth highlighting the historical context by noting that, up until the mid-1980s, research papers on robot control consistently featured assessments of the computational demands associated with their implementations. In the mid-1980s, robot manipulators became a standard control application, and were recognized and exploited in research, The inverse dynamics control motivated the differential geometric method of feedback linearization that has been used in a lot of practical problems within and outside of robotics.[14]

The usage of inverse dynamics control laws imposes an accurate knowledge of the parameters of the system dynamic model, and the equations of motion are computed in real time. These conditions cannot be insured in practice because the model is usually known with the presence of uncertainty due to imperfect knowledge of manipulator mechanical parameters, and existence of unmodelled dynamics.

Over the past few decades, substantial research has focused on enhancing or creating controllers for systems with uncertainties. Key elements of

established nonlinear robust control approaches include adaptive control, sliding mode control (SMC), a blend of adaptive and sliding mode control, passivity-based control, and robust control based on Lyapunov stability.[50] In conventional nonlinear adaptive controllers, the controller aims to acquire knowledge about uncertain parameters associated with specific structured dynamics. This allows for precise control and compensation for structured uncertainties as well as bounded disturbances. However, adaptive control methods are limited in their ability to parameterize known functional relationships when the constants within those relationships are unknown. Consequently, these limitations can impact the performance of existing nonlinear adaptive controllers in scenarios where the dynamic model is poorly understood or when rapid real-time control is essential .[20, 2, 6, 38, 37, 35, 3]

On the other hand, robust variable structure controllers utilizing sliding mode control (SMC) are naturally appealing due to their capacity to handle uncertainties, exhibit excellent transient performance (i.e., minimal tracking error), and provide swift responses. However, the discontinuous nature of the control law in sliding mode control (SMC) can lead to a phenomenon known as "chattering," which has the potential to trigger unmodeled high-frequency dynamics. Attempts to mitigate this chattering through techniques like the boundary-layer method entail a trade-off between performance and chattering reduction. Additionally, achieving robustness and convergence in SMC requires a priori knowledge of the upper bounds of the perturbation vector, which represents the unknowns in the system. Designing controllers based on worst-case scenarios may result in overly conservative approaches . [38, 3, 42, 49, 39]

Efforts have been made to address the limitations of SMC, including chattering and the need for prior knowledge of perturbation bounds. Perturbation estimation was investigated for a specific class of nonlinear uncertain systems in[38, 8, 19, 30], but these challenges remained largely unresolved.

Robust adaptive controllers, such as combined adaptive sliding mode controllers, have been explored as a means to overcome the drawbacks of

both adaptive control and SMC. This approach employs adaptive control to estimate unknown system parameters and SMC to handle unmodeled dynamics and external disturbances. However, these combined adaptive controllers require a linearly parameterized model of the system and prior knowledge of uncertainty bounds. Moreover, dealing with a large number of parameters and adaptation gains corresponding to each unknown parameter introduces complexity. Issues related to adaptation time and computational burden can pose stability and robustness challenges, particularly in scenarios where rapid real-time control is necessary .[37, 46, 47],

Addressing the design of robust adaptive controllers without knowledge of uncertainty bounds was tackled in [12] and related references, along with [4], which necessitated a specific linearly parameterized model of the system for controller construction.

Passive controllers leverage the passive nature of controlled dynamical systems, which generally cannot provide more energy than is input into them. Implementing such controllers is relatively straightforward, but they do not readily yield quantifiable performance measures [1]. In contrast, robust Lyapunov stability-based approaches offer insights into the asymptotic behavior of solutions without the need to solve differential equations. However, akin to sliding mode control, these approaches yield discontinuous control laws that are susceptible to chattering .[38, 23]. The

authors of [36] propose a robust controller to address the uncertainties discussed earlier. This controller offers a promising solution but hinges on three critical assumptions. While two of these assumptions are often met in practice, the third one necessitates prior knowledge of the range of uncertainties. The primary objective of this thesis is to present an innovative approach to fulfill this assumption without the requirement for explicit knowledge of these parameters.

The core investigation and experimentation in this thesis primarily focus on a relatively simple 2R planar manipulator. Through extensive analysis and testing on this platform. Subsequently, the research extends its

scope to a more intricate and dynamic structure, namely the widely employed SCARA robot. This expansion allows for a broader assessment of the developed control strategies and their adaptability across different robotic systems.

By pursuing these research objectives, the thesis aims to contribute to the advancement of robotics control and address practical challenges related to handling uncertainties in real-world applications. The transition from a simpler manipulator to a more complex one like the SCARA robot underscores the robustness and generalizability of the proposed approach, making it a valuable contribution to the field of robotics and control theory.

This document is organized as follows:

-Chapter 2: Provides a concise introduction to the process of deriving the dynamic model using the Lagrange formulation. It also includes an illustrative example of this derivation applied to the 2R robot, which is the robot under examination.

-Chapter 3: We delve into the concept of Inverse Dynamics Control and present the design details of our robust controller. Additionally, we provide an overview of Sliding Mode Control (SMC), which is the category our controller falls under. In this chapter, we also introduce the PD+ non linear controller as an alternative approach to address uncertainties in the system dynamics. Furthermore, we conduct a comparative analysis of simulation results for both controllers.

-Chapter 4: Expands our study by applying our Robust Controller to a more complex robot structure, specifically, the well-known SCARA robot. We meticulously analyze the simulation results in this context.

-Chapter 5: We reach the culmination of this thesis, offering essential insights. We conclude our extensive exploration of Robust control for robotic manipulators, emphasizing both the established and novel methodologies we've applied

Chapter 2

Dynamical Model

Robot Dynamics is the subject that studies the motions of robots taking into account the forces and torques that cause them.[22] Deriving the dynamical model of a robotic manipulator or any mechanical system is of crucial importance for various reasons:

Control and Trajectory Planning:The dynamical model provides the mathematical description of how the system's motion is affected by the forces and torques applied to it. This information is essential for designing control algorithms to achieve desired trajectories and achieve accurate and stable motion control. Simulation and Analysis:The dynamical model allows to simulate the behavior of the mechanical system under different conditions. This helps in understanding the system's performance, identifying critical points, predicting behavior, and evaluating its stability.[36] Design and Optimization: Knowledge of the dynamical model is vital for designing mechanical systems with optimized performance. It enables to select appropriate actuator specifications, joint designs, link lengths, and other parameters to meet specific performance criteria. Safety and Robustness:Understanding the system's dynamics is crucial for ensuring the safety of both the manipulator and its surroundings. It helps in identifying potential risks and designing safety measures. Moreover, a robust dynamical model helps in designing control strategies that can handle uncertainties and disturbances during operation. The Dynami-

cal equations associated also referred to as the equations of motion are a set of second-order differential equations. When we talk about dynamics there are main topics that are forward dynamics and inverse dynamics,the forward problem is to determine the robot's acceleration given the state and the joint torques and forces,while the inverse problem is to find the forces and torques corresponding to the state of the robot and the desired acceleration.

There are two main methods that are used for deriving the equations of motion one of them is based on the *Lagrange formulatio*n derived from the kinetic and potential energy of the robot, and the second one is based on the *NewtonEuler formulation* that is a direct application of Newton's and Euler's dynamic equations for a rigid body. The Lagrangian formalism is elegant and efficient for robots with simple structures, such as those with three or fewer degrees of freedom. However, as the number of degrees of freedom increases, the calculations can become cumbersome. For more complex robots with open chains, the Newton-Euler formulation offers efficient recursive algorithms for both inverse and forward dynamics. These algorithms can be assembled into closed-form analytic expressions, providing a practical and efficient approach for handling the dynamics of such robots[22][13].

In this chapter the *Lagrange formulation* method is briefly explained for the derivation of the equations of motion of a manipulator in the joint space, which is conceptually simple and systematic. We will use the equations of motion in chapter 3 and chapter 4, for deeper explanation refer to [36].

2.1 Lagrange Formulation

The dynamic model of a manipulator describes how the joint actuator torques are related to the motion of the structure. The use of *Lagrange formulation* gives the advantage of deriving the equations of motion in a systematic way independently of the reference coordinate frame. In this method The derivation of the dynamic model of the manipulator begins with the determination of the mechanical system's kinetic energy and

potential energy[36]. When the *generalized coordinates* that are the set of variables $q_i, i = 1, \ldots, n$, are chosen which effectively describe the positions of an n-DOF manipulator, the *Lagrangian* of the mechanical system is a function of the generalized coordinates:

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{2.1}$$

where $\mathcal{T}$ denotes the total *Kinetic energy*,and $\mathcal{U}$ denotes the total *potential energy* of the system. The *Lagrange equations* are

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \quad \text{for } i = 1, \ldots, n \tag{2.2}$$

where ξ_i is the *generalized force* associated with the generalized coordinate q_i , we can rewrite (2.2) in a compact form :

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial q}\right)^T = \xi \tag{2.3}$$

In a manipulator with an open kinematic chain, the vector of joint variables q gathers the generalized coordinates. The contributions to the generalized forces are determined by the nonconservative forces, where the joint actuator torques, joint friction torques, and the joint torques resulting from end-effector forces applied during contact with the environment. The equations in (2.2) establish the relationships between the generalized forces applied to the manipulator and the joint positions, velocities, and accelerations.

2.1.1 Kinetic Energy Computation

Considering a manipulator that has *n-rigid links* the sum of the contributions relative to the motion of each link and the contributions relative to the motion of each joint actuator.

$$\mathcal{T} = \sum_{i=1}^{n} (\mathcal{T}_{\ell_i} + \mathcal{T}_{m_i}) \tag{2.4}$$

where $\mathcal{T}_{\ell_i}$, $\mathcal{T}_{m_i}$ respectively are the kinetic energy of Link i and the kinetic energy of the motor that actuate Joint i. The contribution of the kinetic energy of link i is given by

$$\mathcal{T}_{\ell_i} = \frac{1}{2} \int_{V_{\ell_i}} (\dot{p}_i^*)^T \dot{p}_i^* \rho dV \tag{2.5}$$

where $\dot{p}_i^*$ is the linear velocity vector and ρ is the density of the elementary particle of volume dV ; V_{ℓ_i} is the volume of link i. Considering the position vector p_i^* of the elementary particle, and the position vector Pc_i of the link center of mass, both of them with respect to the base frame :

$$r_i = [r_{ix} \quad r_{iy} \quad r_{iz}]^T = p_i^* - p_{\ell_i} \tag{2.6}$$

where

$$p_{\ell_i} = \frac{1}{m_{l_i}} \int_{V_{l_i}} p_i^* \rho \, dV \tag{2.7}$$

where m_{l_i} is the link mass. Hence the point velocity of the link can be expressed as

$$\dot{p}_i^* = \dot{p}_{\ell_i} + \omega_i \times r_i = \dot{p}_{\ell_i} + S(\omega_i) r_i \tag{2.8}$$

where $\dot{p}_{l_i}$ and ω_i respectively are the center of mass linear velocity and the link's angular velocity see figure(2.1).

By substituting the velocity expression (2.8) into(2.5) , we can observe that the kinetic energy of each link is composed of the following contributions: **Translational** The contribution is :

$$\frac{1}{2} \int_{V_{\ell_i}} \dot{p}_{\ell_i}^T \dot{p}_{\ell_i} \rho \, dV = \frac{1}{2} m_{\ell_i} \dot{p}_{\ell_i}^T \dot{p}_{\ell_i} \tag{2.9}$$

Mutual

The contribution is:

$$2 \left(\frac{1}{2} \int_{V_{\ell_i}} \dot{p}_{\ell_i}^T S(\omega_i) r_i \rho \, dV \right) = 2 \left(\frac{1}{2} \dot{p}_{\ell_i}^T S(\omega_i) \int_{V_{\ell_i}} (p_i^* - p_{\ell_i}) \rho \, dV \right) = 0 \tag{2.10}$$

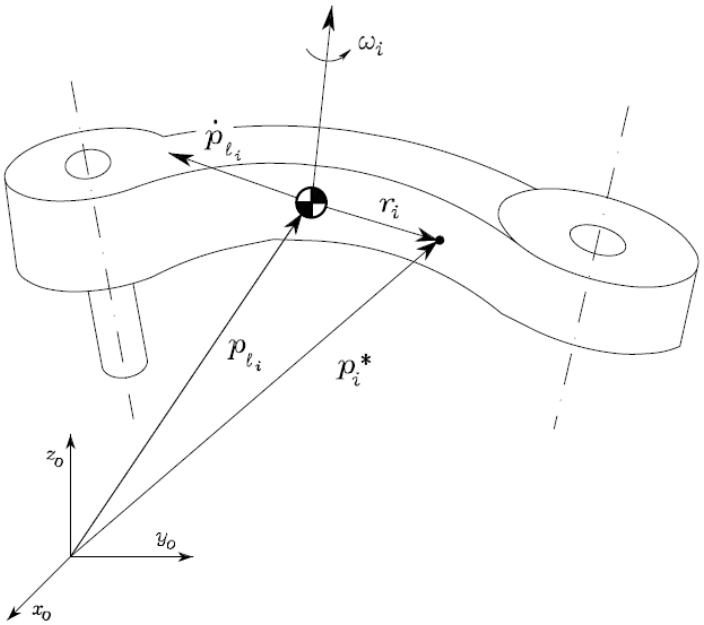

Figure 2.1: Kinematic description of Link i for Lagrange formulation(from [36])

referring to 2.7 it is

$$\int_{V_{\ell_i}} \dot{p}_i^* \rho \, dV = p_{\ell_i} \int_{V_{\ell_i}} \rho \, dV \tag{2.11}$$

Rotational

The contribution is

$$\frac{1}{2} \int_{V_{\ell_i}} r_i^T S^T(\omega_i) S(\omega_i) r_i \rho \, dV = \frac{1}{2} \omega_i^T \left(\int_{V_{\ell_i}} S^T(r_i) S(r_i) \rho \, dV \right) \omega_i \tag{2.12}$$

where the property $S(\omega_i)r_i = -S(r_i)\omega_i$ has been exploited. in the view of

the matrix operator $S(.)$

$$S(r_i) = \begin{bmatrix} 0 & -r_{iz} & r_{iy} \\ r_{iz} & 0 & -r_{ix} \\ -r_{iy} & r_{ix} & 0 \end{bmatrix} \tag{2.13}$$

it is

$$\frac{1}{2}\int_{V_{\ell_i}} r_i^T S^T(\omega_i) S(\omega_i) r_i \rho \, dV = \frac{1}{2}\omega_i^T I_{\ell_i} \omega_i \tag{2.14}$$

The matrix

$$I_{\ell_i} = \begin{bmatrix} \int (r_{iy}^2 + r_{iz}^2)\rho \, dV & -\int r_{ix} r_{iy} \rho \, dV & -\int r_{ix} r_{iz} \rho \, dV \\ * & \int (r_{ix}^2 + r_{iz}^2)\rho \, dV & -\int r_{iy} r_{iz} \rho \, dV \\ * & * & \int (r_{ix}^2 + r_{iy}^2)\rho \, dV \end{bmatrix} \tag{2.15}$$

$$= \begin{bmatrix} I_{\ell_i xx} & -I_{\ell_i xy} & -I_{\ell_i xz} \\ * & I_{\ell_i yy} & -I_{\ell_i yz} \\ * & * & I_{\ell_i zz} \end{bmatrix} \tag{2.16}$$

The tensor is symmetric and describes the inertia relative to the center of mass of Link i when expressed in the base frame. It's important to note that the position of Link i depends on the manipulator's configuration, making the inertia tensor configuration-dependent when expressed in the base frame. If the angular velocity of Link i is represented with reference to a frame attached to the link (as in the Denavit-Hartenberg convention), the tensor takes on a specific form.

$$\omega_i^i = R_i^T \omega_i \tag{2.17}$$

where R_i is the rotation matrix from Link i frame to the base frame. When referred to the link frame, the inertia tensor is constant. Let $I_{\ell_i}^i$ be such tensor; then the following relation is easily verified

$$I_{\ell_i} = R_i I_{\ell_i}^i R_i^T \tag{2.18}$$

If the axes of Link i frame coincide with the central axes of inertia, then the inertia products are null and the inertia tensor relative to the centre of mass is a diagonal matrix.

By summing the translational and rotational contributions 2.9 and 2.14 the kinetic energy of Link i is

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell_i} \dot{p}_{\ell_i}^T \dot{p}_{\ell_i} + \frac{1}{2} \omega_i^T R_i I_{\ell_i}^i R_i^T \omega_i. \tag{2.19}$$

Now we express the kinetic energy as a function of the generalized coordinates of the system, that are the joint variables. To do that, the geometric method for Jacobian computation is applied to the intermediate link other than the end-effector, yielding

$$\dot{p}_{\ell_i} = j_{P_1}^{(\ell_i)} \dot{q}_1 + \ldots + j_{P_i}^{(\ell_i)} \dot{q}_i = J_P^{(\ell_i)} \dot{q} \tag{2.20}$$

$$\omega_i = j_{O_1}^{(\ell_i)} \dot{q}_1 + \ldots + j_{O_i}^{(\ell_i)} \dot{q}_i = J_O^{(\ell_i)} \dot{q} \tag{2.21}$$

where the contributions of the Jacobian columns relative to the joint velocities have been taken into account up to current Link i. The Jacobians to consider are then:

$$J_P^{(\ell_i)} = \begin{bmatrix} j_{P_1}^{(\ell_i)} & \ldots & j_{P_i}^{(\ell_i)} & 0 & \ldots & 0 \end{bmatrix} \tag{2.22}$$

$$J_O^{(\ell_i)} = \begin{bmatrix} j_{O_1}^{(\ell_i)} & \ldots & j_{O_i}^{(\ell_i)} & 0 & \ldots & 0 \end{bmatrix} \tag{2.23}$$

the columns of the matrices in 2.22 and 2.23 can be computed according to the Jacobian partitioning that is explained in the third chapter of [36], we get

$$j_{P_j}^{(\ell_i)} = \begin{cases} z_{j-1} & \text{for a prismatic joint} \\ z_{j-1} \mathrm{x} (p_{\ell_i} - p_{j-1}) & \text{for revolute joint} \end{cases}$$

$$j_{O_j}^{(\ell_i)} = \begin{cases} 0 & \text{for a prismatic joint} \\ z_{j-1} & \text{for revolute joint} \end{cases}$$

where p_{j-1} is the position vector of the origin of Frame $j-1$ and $z-1$ is the unit vector of axis z of Frame $j-1$ It follows that the kinetic energy of Link i in 2.19 can be re written as

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell_i} \dot{q}^T J_p^{(\ell_i)T} J_P^{(\ell_i)} \dot{q} + \frac{1}{2} \dot{q}^T J_O^{(\ell_i)T} R_i I_{\ell_i}^i R_i^T J_O^{(\ell_i)} \dot{q} \tag{2.24}$$

The kinetic energy contribution of Joint i's motor can be computed in a manner similar to that of the link. This is particularly applicable to rotary electric motors, which can actuate both revolute and prismatic joints through appropriate transmissions. When considering such motors, we can assume that the contribution of the fixed part (stator) is already accounted for in the kinetic energy of the link on which the motor is situated. Thus, we only need to compute the contribution of the rotor.

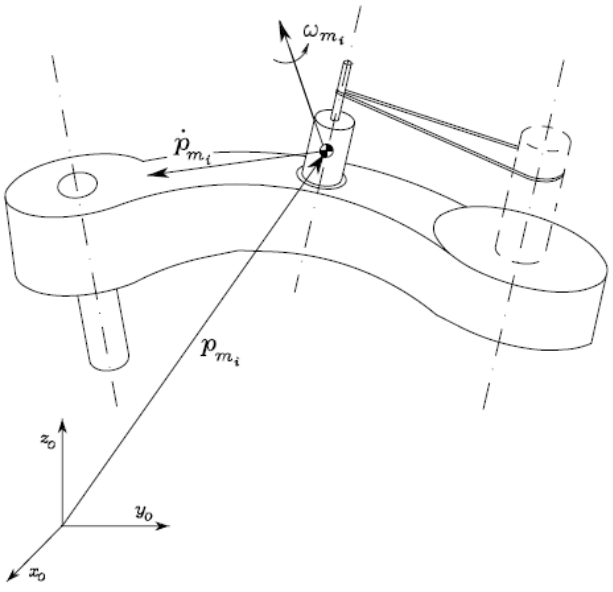

Figure 2.2: Kinematic description of Motor i (from[36])

In Fig. 2.2, we assume that the motor of Joint i is located on Link $i-1$

. The joint actuator torques are transmitted to the motors via mechanical transmission gears, and the contribution of the gears to the kinetic energy can be appropriately included in that of the motor.

It is also assumed that no induced motion occurs, meaning the motion of Joint i does not affect the motion of other joints[36].

The kinetic energy of Rotor i can be written as

$$\mathcal{T}_{m_i} = \frac{1}{2} m_{m_i} \dot{p}_{m_i}^T \dot{p}_{m_i} + \frac{1}{2} \omega_{m_i}^T I_{m_i} \omega_{m_i} \tag{2.25}$$

where m_{m_i} is the mass of the rotor, $\dot{p}_{m_i}$ denotes the linear velocity of the centre of mass of the rotor, I_{m_i} is the inertia tensor of the rotor relative to its centre of mass, and ω_{m_i} denotes the angular velocity of the rotor. We denote the angular position of the rotor (ϑ_{m_i}). On the assumption of a rigid transmission we get

$$k_{ri} \dot{q}_i = \dot{\vartheta}_{m_i} \tag{2.26}$$

where k_{ri} is the gear reduction ratio. We notice that, in the case of actuation of a prismatic joint, the gear reduction ratio is a dimensional quantity.

According to the angular velocity composition rule that is explained in the third chapter of [36]. and the relation in(2.26), the total angular velocity of the rotor is given by :

$$\omega_{m_i} = \omega_{i-1} + k_{ri} \dot{q}_i z_{m_i} \tag{2.27}$$

where ω_{i-1} is the angular velocity of Link $i-1$ on which the motor is located, and z_{m_i} denotes the unit vector along the rotor axis. expressing the rotor kinetic energy as a function of the joint variables, it is worth expressing the linear velocity of the rotor centre of mass similar to 2.20

$$\dot{p}_{m_i} = J_P^{(m_i)} \dot{q} \tag{2.28}$$

The Jacobian is

$$J_P^{(m_i)} = [J_{P_1}^{(m_i)} \dots J_{P,i-1}^{(m_i)} \quad 0 \dots 0] \tag{2.29}$$

where the columns of this Jacobian defined by

$$J_{P_j}^{(m_i)} = \begin{cases} z_{j-1}, & \text{for a prismatic joint} \\ z_{j-1}\mathrm{x}(p_{m_i} - p_{j-1}), & \text{for a revolute joint} \end{cases}$$

We have p_{j-1} is the position vector of the origin of frame $j-1$. looking back at (2.29) $J_{p_i}^{(m_i)} = 0$ because the center of mass of the rotor has been taken along its axis of rotation. Refering to(2.27) it's possible to express the angular velocity as a function of the joints variables in this way:

$$\omega_{m_i} = J_O^{(m_i)} \dot{q} \tag{2.30}$$

In this case the Jacobian

$$J_O^{(m_i)} = [J_{O1}^{(m_i} \dots J_{O,i-1}^{(m_i)} J_{O_i}^{(m_i)} \quad 0 \dots 0] \tag{2.31}$$

by referring to (2.27) and (2.21) the columns of that Jacobian are respectively given by

$$J_{O_j}^{(m_i)} = \begin{cases} J_{O_i}^{(l_i)}, & j = 1, \dots, i-1 \\ k_{r_i} z_{m_i}, & j = i \end{cases}$$

Hence, the kinetic energy of Rotor i can be expressed as follows:

$$\mathcal{T}_{m_i} = \frac{1}{2} m_{m_i} \dot{q}^T J_P^{(m_i)T} J_p^{(m_i)} \dot{q} + \frac{1}{2} \dot{q}^T J_O^{(m_i)T} R_{m_i} I_{m_i}^{m_i} R_{m_i}^T J_O^{(m_i)} \dot{q} \tag{2.32}$$

By summing (2.24) and(2.4) we get the total kinetic energy in a quadratic form

$$\mathcal{T} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij}(q) \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{q}^T B(q) \dot{q} \tag{2.33}$$

where

$$B(q) = \sum_{i=1}^{n} \left(m_{\ell_i} J_P^{(\ell_i)T} J_P^{(\ell_i)} + J_O^{(\ell_i)T} R_i I_{\ell_i}^i R_i^T J_O^{(\ell_i)} + m_{m_i} J_P^{(m_i)T} J_P^{(m_i)} + J_O^{(m_i)T} R_{m_i} I_{m_i}^{m_i} R_{m_i}^T J_O^{(m_i)} \right) \tag{2.34}$$

where $B(q)$ is an (n x n) symmetric, positive definite configuration-dependent (in general),It's referred to as the inertia matrix[36] or the mass matrix[22].

2.1.2 Potential Energy Computation

Similar to the kinetic energy, the potential energy stored in the manipulator is obtained by summing up the contributions from each link and each rotor:

$$\mathcal{U} = \sum_{i=1}^{n} \left(\mathcal{U}_{\ell_i} + \mathcal{U}_{m_i} \right) \tag{2.35}$$

Assuming rigid links, the contribution of gravitational forces, and only gravitational forces, is expressed as follows:

$$\mathcal{U}_{\ell_i} = - \int_{V_{\ell_i}} g_0^T p_i^* \rho \, dV = -m_{\ell_i} g_0^T p_{\ell_i} \tag{2.36}$$

In(2.36) g_0 is the acceleration vector of the gravity in the base frame. The contribution of rotor i :

$$\mathcal{U}_{m_i} = -m_{m_i} g_0^T p_{m_i} \tag{2.37}$$

Now we substitute (2.36) and (2.37) into (2.35) we get the potential energy

$$\mathcal{U} = - \sum_{i=1}^{n} \left(m_{\ell_i} g_0^T p_{\ell_i} + m_{m_i} g_0^T p_{m_i} \right) \tag{2.38}$$

Where we have the potential energy, by the vectors p_{ℓ_i} and p_{m_i} is a function of the joint variables q and not of the joint velocities $\dot{q}$.

2.1.3 Equations of Motion

After calculating the total kinetic and potential energy of the system, as given in equations (2.33) and (2.38), the Lagrangianin(2.1) for the manipulator can be expressed as:

$$\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) \tag{2.39}$$

By taking the derivatives required by the Lagrange equations as described in 2.3and considering that $\mathcal{U}$ does not depend on $\dot{q}$, we obtain the following result:

$$B(q)\ddot{q} + n(q, \dot{q}) = \xi \tag{2.40}$$

where

$$n(q, \dot{q}) = \dot{B}(q)\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}(\dot{q}^T B(q)\dot{q})\right)^T + \left(\frac{\partial \mathcal{U}(q)}{\partial q}\right)^T \tag{2.41}$$

Taking into account the fact that $\mathcal{U}$ in equation (2.38) does not depend on $\dot{q}$. and considering the expression provided in equation (2.33) , we can derive the following result:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) = \frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{q}_i}\right) = \sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n} \frac{db_{ij}(q)}{dt}\dot{q}_j = \sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial b_{ij}(q)}{\partial q_k}\dot{q}_k\dot{q}_j \tag{2.42}$$

and

$$\frac{\partial \mathcal{T}}{\partial q_i} = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial b_{jk}(q)}{\partial q_i}\dot{q}_k\dot{q}_j \tag{2.43}$$

In the derivation, we have conveniently switched the indices of summation. Moreover in (2.20) and (2.28)

$$\frac{\partial \mathcal{U}}{\partial q_i} = -\sum_{j=1}^{n}\left(m_{\ell_i} g_0^T \frac{\partial p_{\ell_j}}{\partial q_i} + m_{m_j} g_0^T \frac{\partial p_{m_j}}{\partial q_i}\right) \tag{2.44}$$

$$= -\sum_{j=1}^{n}\left(m_{\ell_j} g_0^T J_{P_i}^{(\ell_j)}(q) + m_{m_j} g_0^T J_{P_i}^{(m_j)}(q)\right) = g_i(q) \tag{2.45}$$

where also the summation index has been changed. Hence the equations of motion can be expressed as

$$\sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}(q)\dot{q}_k\dot{q}_j + g_i(q) = \xi_i \quad i = 1, \ldots, n \tag{2.46}$$

where

$$h_{ijk} = \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i} \tag{2.47}$$

The equations of motion(2.46) have a physical interpretation: The coefficients b_{ii} represent the moment of inertia at Joint i axis in the current manipulator configuration when the other joints are fixed, while b_{ij} accounts for the effect of the acceleration of Joint j on Joint i. The Term $h_{ij}q_j\dot{q}_j^2$ epresents the centrifugal effect induced on Joint i by the velocity of Joint j (noting that $h_{iii} = 0$ since $\frac{\partial b_{ii}}{\partial q_i} = 0$). Furthermore, the term $h_{ijk}q_j\dot{q}_jq_k\dot{q}_k$ captures the Coriolis effect induced on Joint i by the velocities of Joints j and k.Additionally, the term g_i represents the moment generated at Joint i axis of the manipulator in the current configuration due to the presence of gravity. During the structure's design, some joint dynamic couplings, such as b_{ij} and h_{ijk},can be reduced or set to zero to simplify the control problem. This approach helps manage complexity and improve control efficiency.

The equations of motion (2.40) can be expressed in a more concise matrix form, representing the joint space dynamic model:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + F_s sgn(\dot{q}) + g(q) = \tau - J^T(q)h_e \tag{2.48}$$

In (2.48) the nonconservative forces that act at the joints can be expressed as the difference between the actuation torques and the combined effects of viscous friction torques $F_v\dot{q}$ and simplified static friction torques $f_s(q,\dot{q})$. Here, F_v is a diagonal matrix of viscous friction coefficients, and $F_s sgn(\dot{q})$ can be represented as the Coulomb friction torques . The matrices F_s and Fv have dimensions (nxn), and $sgn(\dot{q})$represents the (nx1) vector containing the sign functions of the individual joint velocities. When the manipu-

lator's end-effector makes contact with the environment, a portion of the actuation torques is utilized to counterbalance the torques arising from the contact forces at the joints. This balancing effect is given by the product of the transpose of the Jacobian matrix $J^T(q)h_e$, where h_e represents the force and moment exerted by the end-effector on the environment.

we also have that C is a suitable $(n \times n)$ matrix such that its elements c_{ij} satisfy the equation

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_k\dot{q}_j \tag{2.49}$$

This dynamic model has two properties that are useful for both dynamic parameter identification and the derivation of control algorithms. The first property is *the skew-symmetry of Matrix* $\dot{B} - 2C$ The choice of matrix C is not unique, since we can have more than one matrix that satisfies (2.49).A specific choice can be achieved by expanding the term on the right-hand side of (2.49) and taking into account the expressions of the coefficients h_{ijk} as given in (2.47) we get

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial b_{ij}}{\partial q_k}\dot{q}_k\dot{q}_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j$$

which gives us the relation to calculate the generic element of C

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k \tag{2.50}$$

where the coefficients

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} + \frac{\partial b_{jk}}{\partial q_i}\right) \tag{2.51}$$

The Terms in(2.51) are referred to as *Christoffel symbols* of the first type. It

is essential to note that, due to the symmetry of B it results:

$$c_{ijk} = c_{ikj} \tag{2.52}$$

This particular choice for the matrix C reveals a notable property of the equations of motion (2.48). The matrix

$$N(q, \dot{q}) = \dot{B}(q) - 2C(q, \dot{q}) \tag{2.53}$$

is skew-symmetric, By substituting (2.51) into (2.50) we get the expression of the generic element of the matrix N in (2.53) as

$$n_{ij} = b_{ij} - 2c_{ij} = \sum_{k=1}^{n} \left(\frac{\partial b_{jk}}{\partial q_i} - \frac{\partial b_{ik}}{\partial q_j} \right) \dot{q}_k \tag{2.54}$$

The skew symmetry of the matrix N gives us this interesting result:

$$\dot{q}^T N(q, \dot{q}) \dot{q} = 0 \tag{2.55}$$

It can be shown that The relation (2.55) holds for any matrix C choice as it directly follows from the physical properties of the system. In contrast, the general skew symmetry (for any vector) valid only for the specific choice of C elements stated in (2.50) and (2.51). The second property is *the linearity in the dynamics parameters*. A crucial characteristic of the dynamic model is its linearity concerning the dynamic parameters that define the manipulator's links and rotors. By associating the kinetic and potential energy contributions of each rotor with those of the link on which it is located, we can consider the augmented Link i as the union of Link i and Rotor $i+1$. In this context, the kinetic energy contribution of the augmented Link i is given by...

$$\mathcal{T}_i = \mathcal{T}_{\ell_i} + \mathcal{T}_{m_{i+1}} \tag{2.56}$$

by calculating the linear velocity of the link and rotor with reference to the center of mass of the augmented link,and by the virtue of steiner theorem we get the matrix that represents the inertia tensor relative to the overall

centre of mass the inertia tensor incorporates an additional contribution resulting from the translation of the reference point (pole) relative to which the tensor is evaluated. and by Assuming the rotor has a symmetric mass distribution about its axis of rotation, its inertia tensor is expressed in a frame R_{mi} with the center of mass at the origin, and the z_{mi} axis aligned with the rotation axis. we get that the inertia tensor remains unchanged (invariant) with respect to any rotation around the z_{mi} axis and remains constant when referred to any frame attached to Link i_1. To achieve the objective of determining a set of dynamic parameters that are independent of the manipulator joint configuration, it is beneficial to refer the inertia tensor of the link $\bar{I}_i$to frame R_i attached to the link and the inertia tensor $I_{m_{i+1}}$ to frame $R_{m_{i+1}}$. This transformation ensures that the inertia tensor becomes diagonal. By applying the linear velocity composition rule for Link i and referring all the vectors to Frame i, we obtain:

$$\mathcal{T}_i = \frac{1}{2} m_i \dot{p}_i^{iT} \dot{p}_i^i + \dot{p}_i^{iT} S(w_i^i) m_i r_i^i c_i + \frac{1}{2} w_i^{iT} \hat{I}_i^i w_i^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} z_{m_{i+1}}^{iT} w_i^i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 I_{m_{i+1}} \tag{2.57}$$

where

$$\hat{I}_i^i = \bar{I}_i^i + m_i S^T(r_{i,C_i}^i) S(r_{i,C_i}^i) \tag{2.58}$$

The result represents the inertia tensor with respect to the origin of Frame i, using the Steiner theorem. If we make $r_{i,C_i}^i = [\ell_{C_i x} \quad \ell_{C_i y} \quad \ell_{C_i z}]^T$ The first moment of inertia becomes

$$m_i r_{i,C_i}^i = \begin{bmatrix} m_i \ell_{Cix} \\ m_i \ell_{Ciy} \\ m_i \ell_{Ciz} \end{bmatrix} \tag{2.59}$$

From (2.58) the inertia tensor of augmented Link i is

$$\hat{I}_i^i = \begin{bmatrix} \bar{I}_{ixx} + m_i(\ell_{C_i y}^2 + \ell_{C_i z}^2) & -\bar{I}_{ixy} - m_i \ell_{C_i x} \ell_{C_i y} & \bar{I}_{ixz} - m_i \ell_{C_i x} \ell_{C_i z} \\ * & \bar{I}_{iyy} + m_i(\ell_{C_i x}^2 + \ell_{C_i z}^2) & -\bar{I}_{iyz} - m_i \ell_{C_i y} \ell_{C_i z} \\ * & * & \bar{I}_{izz} + m_i(\ell_{C_i x}^2 + \ell_{C_i y}^2) \end{bmatrix} \tag{2.60}$$

$$= \begin{bmatrix} \hat{I}_{ixx} & -\hat{I}_{ixy} & -\hat{I}_{ixz} \\ * & \hat{I}_{iyy} & -\hat{I}_{iyz} \\ * & * & \hat{I}_{izz} \end{bmatrix} \tag{2.61}$$

Hence, the kinetic energy of the augmented link shows linearity concerning the dynamic parameters. These dynamic parameters include the mass, the three components of the first moment of inertia as given in (2.59), the six components of the inertia tensor as stated in (2.61), and the moment of inertia of the rotor. Concerning potential energy, by referring all the vectors to Frame i, we can express the single contribution of potential energy:

$$\mathcal{U}_i = -g_0^{iT}(m_i p_i^i + m_i r_i^i c_i) \tag{2.62}$$

Which confirm that the potential energy of the augmented link is linear with respect to the mass and the three components of the first moment of inertia in (2.59) The Lagrangian of the system (7.1) can be formulated by summing the contributions of kinetic energy and potential energy for all augmented links in the following manner:

$$\mathcal{L} = \sum_{i=1}^{n} (\beta_{\mathcal{T}i}^T - \beta_{\mathcal{U}i}^T)\pi_i \tag{2.63}$$

where $_i$ is the (11 x 1) vector of dynamic parameters

$$\pi_i = [\ m_i\ m_i\ell_{C_ix}\ m_i\ell_{C_iy}\ m_i\ell_{C_iz}\ \hat{I}_{ixx}\ \hat{I}_{ixy}\ \hat{I}_{ixz}\ \hat{I}_{iyy}\ \hat{I}_{iyz}\ \hat{I}_{izz}\ I_{m_i}\ ]^T \tag{2.64}$$

where the expression includes the moment of inertia of Rotor i, which has been linked with the parameters of Link i to simplify the notation. In equation (2.63), $\beta_{\mathcal{T}i}$ and $\beta_{\mathcal{U}i}$ are two (11 x 1) vectors that enable the representation of the Lagrangian as a function of π_i. These vectors are dependent on the generalized coordinates of the mechanical system, including their derivatives concerning $\beta_{\mathcal{T}i}$. At this stage, it is essential to note that the derivations involved in the Lagrange equations (2.2) do not affect the linearity in the parameters. As a result, the generalized force at

Joint i can be expressed as follows:

$$\xi_i = \sum_{j=1}^{n} y_{ij}^T \pi_j \tag{2.65}$$

where

$$y_{ij} = \frac{d}{dt}\frac{\partial \beta_{\mathcal{T}j}}{\partial \dot{q}_i} - \frac{\partial \beta_{\mathcal{T}j}}{\partial q_i} + \frac{\partial \beta_{\mathcal{U}j}}{\partial q_i} \tag{2.66}$$

As the partial derivatives of $\beta_{\mathcal{T}j}$ and $\beta_{\mathcal{U}j}$ appearing in (2.66) become zero for$j < i$, we arrive at the following significant outcome:

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} y_{11}^T & y_{12}^T & \dots & y_{1n}^T \\ 0^T & y_{22}^T & \dots & y_{2n}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \dots & y_{nn}^T \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix} \tag{2.67}$$

This results in the manipulator model exhibiting linearity concerning a suitable set of dynamic parameters. In the scenario where there are no contact forces (i.e., $h_e = 0$), it might be beneficial to incorporate the viscous friction coefficient F_{vi} and Coulomb friction coefficient F_{si} as part of the parameters represented by the vector $_i$. This addition would result in a total of 13 parameters per joint. In conclusion, Equation (2.67) can be concisely expressed as:

$$\tau = Y(q, \dot{q}, \ddot{q})\pi \tag{2.68}$$

2.2 Two-link Planar Arm Dynamic Model

Consider the two-link planar arm shown in Fig.(2.3), with the vector of generalized coordinates $q = [\vartheta_1 \quad \vartheta_2]^T$. The distances of the centers of mass of the two links from their respective joint axes are represented as ℓ_1, ℓ_2. Additionally, we havem_{ℓ_1} and m_{ℓ_2} as the masses of the two links, andm_{m_1} and m_{m_1} as the masses of the rotors of the two joint motors. The moments of inertia with respect to the axes of the two rotors are denoted as I_{m_1} and I_{m_2}, while $I_{\ell_1}$and I_{ℓ_2}represent the moments of inertia relative to

the centers of mass of the two links, respectively. We assume that$p_{m_i} = p_{i1}$ and$z_{m_i} = z_{i1}$ for $i = 1, 2$. In other words, the motors are positioned on the joint axes, with centers of mass located at the origins of their respective frames. From (2.34) we compute the inertia matrix

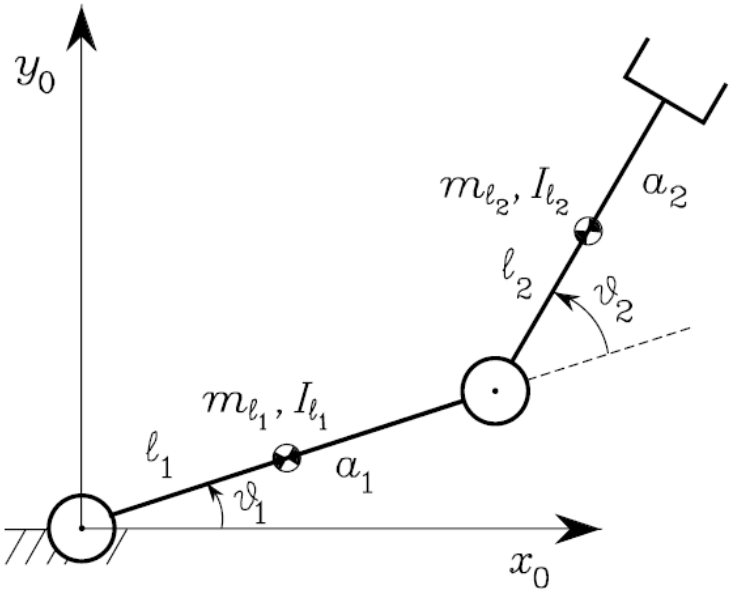

Figure 2.3: Two-link planar arm (from[36])

$$B(q) = \begin{bmatrix} b_{11}(\vartheta_2) & b_{12}(\vartheta_2) \\ b_{21}(\vartheta_2) & b_{22} \end{bmatrix}$$

$$b_{11} = I_{\ell_1} + m_{\ell_1}\ell_1^2 + k_{r1}^2 I_{m_1} + I_{\ell_2} + m_{\ell_2}(a_1^2 + \ell_2^2 + 2a_1\ell_2 c_2) + I_{m_2} + m_{m_2}a_1^2$$

$$b_{12} = b_{21} = I_{\ell_2} + m_{\ell_2}(\ell_2^2 + a_1\ell_2 c_2) + k_{r2} I_{m_2}$$

$$b_{22} = I_{\ell_2} + m_{\ell_2}\ell_2^2 + k_{r2}^2 I_{m_2}$$

Computing *Christoffel* symbols in (2.51):

$$c_{111} = \frac{1}{2}\frac{\partial b_{11}}{\partial q_1} = 0$$

$$c_{112} = c_{121} = \frac{1}{2}\frac{\partial b_{11}}{\partial q_2} = -m_{\ell_2} a_1 \ell_2 s_2 = h$$

$$c_{122} = \frac{\partial b_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial b_{22}}{\partial q_1} = h$$

$$c_{211} = \frac{\partial b_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial b_{11}}{\partial q_2} = -h$$

$$c_{212} = c_{221} = \frac{1}{2}\frac{\partial b_{22}}{\partial q_1} = 0$$

$$c_{222} = \frac{1}{2}\frac{\partial b_{22}}{\partial q_2} = 0$$

from which we build the C matrix as

$$C(q, \dot{q}) = \begin{bmatrix} h\dot{\vartheta}_2 & h(\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ -h\dot{\vartheta}_1 & 0 \end{bmatrix}$$

Now we need to compute N matrix given by (2.53)

$$N(q, \dot{q}) = \dot{B}(q) - 2C(q, \dot{q})$$

$$= \begin{bmatrix} 2h\dot{\vartheta}_2 & h\dot{\vartheta}_2 \\ h\dot{\vartheta}_2 & 0 \end{bmatrix} - 2\begin{bmatrix} h\dot{\vartheta}_2 & h(\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ -h\dot{\vartheta}_1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -2h\dot{\vartheta}_1 - h\dot{\vartheta}_2 \\ 2h\dot{\vartheta}_1 + h\dot{\vartheta}_2 & 0 \end{bmatrix}$$

From witch we see that the skew symmetry property is confirmed. Now we need to compute the gravitational term, since $g_0 = [0 \quad -g \quad 0]^T$ we get

$$g_1 = (m_{\ell_1}\ell_1 + m_{m_2} a_1 + m_{\ell_2} a_1) g c_1 + m_{\ell_2} \ell_2 g c_{12}$$

$$g_2 = m_{\ell_2} \ell_2 g c_{12}$$

At this point we can state the resulting equations of motions which computes τ_1 and τ_2 that are the torques applied to the joints. these equations

are computed in the absence of friction and tip contact forces :

$$\begin{aligned}\tau_1 = &(I_{\ell_1} + m_{\ell_1}\ell_1^2 + k_{r1}^2 I_{m_1} + I_{\ell_2} + m_{\ell_2}(a_1^2 + \ell_2^2 + 2a_1\ell_2 c_2) + I_{m_2} + m_{m_2}a_1^2)\ddot{\vartheta}_1 \\ &+ (I_{\ell_2} + m_{\ell_2}(\ell_2^2 + a_1\ell_2 c_2) + k_{r2}I_{m_2})\ddot{\vartheta}_2 \\ &- 2m_{\ell_2}a_1\ell_2 s_2\dot{\vartheta}_1\dot{\vartheta}_2 - m_{\ell_2}a_1\ell_2 s_2\dot{\vartheta}_2^2 \\ &+ (m_{\ell_1}\ell_1 + m_{m_2}a_1 + m_{\ell_2}a_1)gc_1 + m_{\ell_2}\ell_2 g c_{12}. \\ \tau_2 = &(I_{\ell_2} + m_{\ell_2}(\ell_2^2 + a_1\ell_2 c_2) + k_{r2}I_{m_2})\ddot{\vartheta}_1 \\ &+ (I_{\ell_2} + m_{\ell_2}\ell_2^2 + k_{r2}^2 I_{m_2})\ddot{\vartheta}_2 \\ &+ m_{\ell_2}a_1\ell_2 s_2\dot{\vartheta}_1^2 + m_{\ell_2}\ell_2 g c_{12}\end{aligned} \tag{2.69}$$

The parameterization of the dynamic model with reference to (2.68) Upon careful examination of the expressions for the joint torques, we can identify the corresponding parameter vector as follows:

$$\begin{aligned}\pi_1 &= m_{\ell_1} + m_{m_2}, \\ \pi_2 &= m_{\ell_1}(\ell_1 - a_1), \\ \pi_3 &= I_{\ell_1} + m_{\ell_1}(\ell_1 - a_1)^2 + I_{m_2}, \\ \pi_4 &= I_{m_1}, \\ \pi_5 &= m_{\ell_2}, \\ \pi_6 &= m_{\ell_2}(\ell_2 - a_2), \\ \pi_7 &= I_{\ell_2} + m_{\ell_2}(\ell_2 - a_2)^2, \\ \pi_8 &= I_{m2}.\end{aligned}$$

$$\pi = [\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4 \quad \pi_5 \quad \pi_6 \quad \pi_7 \quad \pi_8]^T \tag{2.70}$$

$$
\begin{aligned}
y_{11} &= a_1^2\ddot{\vartheta}_1 + a_1 g c_1,\\
y_{12} &= 2a_1\ddot{\vartheta}_1 + g c_1,\\
y_{13} &= \ddot{\vartheta}_1,\\
y_{14} &= k_{r1}^2\ddot{\vartheta}_1,\\
y_{15} &= (a_1^2 + 2a_1a_2c_2 + a_2^2)\ddot{\vartheta}_1 + (a_1a_2c_2 + a_2^2)\ddot{\vartheta}_2\\
&\quad - 2a_1a_2s_2\dot{\vartheta}_1\dot{\vartheta}_2 - a_1a_2s_2\dot{\vartheta}_2^2 + a_1 g c_1 + a_2 g c_{12}\\
y_{16} &= (2a_1c_2 + 2a_2)\ddot{\vartheta}_1 + (a_1c_2 + 2a_2)\ddot{\vartheta}_2 - 2a_1s_2\dot{\vartheta}_1\dot{\vartheta}_2\\
&\quad - a_1s_2\dot{\vartheta}_2^2 + g c_{12},\\
y_{17} &= \ddot{\vartheta}_1 + \ddot{\vartheta}_2,\\
y_{18} &= k_{r2}\ddot{\vartheta}_2,
\end{aligned}
$$

It is evident that the count of non-null parameters is lower than the maximum permissible number of twenty-two parameters in this situation. The regressor in equation (2.68) is then computed:

$$
\begin{aligned}
y_{21} &= 0,\\
y_{22} &= 0,\\
y_{23} &= 0,\\
y_{24} &= 0,\\
y_{25} &= (a_1a_2c_2 + a_2^2)\ddot{\vartheta}_1 + a_2^2\ddot{\vartheta}_2 + a_1a_2s_2\dot{\vartheta}_1^2 + a_2 g c_{12},\\
y_{26} &= (a_1c_2 + 2a_2)\ddot{\vartheta}_1 + 2a_2\ddot{\vartheta}_2 + a_1s_2\dot{\vartheta}_1^2 + g c_{12}\\
y_{27} &= \ddot{\vartheta}_1 + \ddot{\vartheta}_2,\\
y_{28} &= k_{r2}\ddot{\vartheta}_1 + k_{r2}^2\ddot{\vartheta}_2.
\end{aligned}
$$

$$
Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} & y_{17} & y_{18} \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} & y_{26} & y_{27} & y_{28} \end{bmatrix} \tag{2.71}
$$

Chapter 3

Motion Control (Inverse Dynamics Control)

The primary objective in controlling a manipulator is to determine the time history of generalized forces (forces or torques) applied by the joint actuators. This task ensures the successful execution of the desired task while meeting specific transient and steady-state requirements.

Various techniques can be employed for controlling a manipulator, and the choice of technique and its implementation significantly influence the manipulator's performance and its potential applications. For example, the requirement for trajectory tracking control in operational space may lead to different hardware/software implementations compared to point-to-point control, which focuses solely on reaching the final position.

Additionally, the mechanical design of the manipulator plays a vital role in determining the suitable control scheme. For instance, the control problem for a Cartesian manipulator differs substantially from that of an anthropomorphic manipulator.

The type of driving system used for the joints also affects the control strategy. If a manipulator is actuated by electric motors with high gear ratios, the presence of gears can linearize the system dynamics and decouple the joints, reducing nonlinearity effects. However, this may introduce joint friction, elasticity, and backlash, which can limit system performance. In

contrast, a robot actuated with direct drives eliminates drawbacks related to friction, elasticity, and backlash, but it introduces weighty nonlinearities and couplings between the joints, necessitating different control strategies for achieving high performance.

Irrespective of the specific type of mechanical manipulator, it is essential to note that task specifications (end-effector motion and forces) are typically defined in operational space, while control actions (joint actuator generalized forces) are carried out in the joint space. This naturally leads to considering two kinds of general control approaches: the joint space control approach and the operational space control approach. Both approaches incorporate closed-loop control structures to leverage the benefits of feedback, such as robustness to modeling uncertainties and disturbance reduction. We can make the following general considerations:

The joint space control problem can be broken down into two subproblems. Firstly, we solve the manipulator's inverse kinematics to transform the desired motion x_d from the operational space into the corresponding motion q_d in the joint space. Subsequently, a joint space control approach is designed to ensure that the actual motion q tracks the reference inputsq_d .However, this approach has the limitation that it does not directly influence the operational space variables x_e, which are controlled in an open-loop fashion through the manipulator's mechanical structure. Consequently, uncertainties in the structure (e.g., construction tolerance, lack of calibration, gear backlash, elasticity) or inaccuracies in knowledge about the end-effector pose relative to the object being manipulated can result in reduced accuracy in the operational space variables.

On the other hand, the operational space control problem adopts a more comprehensive approach with greater algorithmic complexity. In this approach, the inverse kinematics is embedded into the feedback control loop. A key conceptual advantage is the ability to directly influence operational space variables x_e. However, practical measurement of these variables is often indirect and involves evaluating direct kinematics functions based on measured joint space variables. In this chapter, we will focus on inverse dynamics control in joint space. We will address the issue

of uncertainties in the dynamical model and how they can affect control efficiency. To tackle this challenge, we introduce robust control as a solution. Our approach involves a novel method to calculate a parameter that relies on feedback about the error, rather than relying on prior knowledge of the uncertainties. By adopting this feedback-based approach, we aim to enhance the system's ability to cope with uncertainties, leading to more efficient and reliable control of the manipulator.

3.1 Inverse Dynamics Control

As demonstrated in the previous chapter , the equations of motion of a manipulator, in the absence of external end-effector forces and, for the sake of simplicity, static friction (as it is challenging to accurately model), can be expressed as follows:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v\dot{q} + g(q) = \tau \tag{3.1}$$

To control the motion of the manipulator in free space, the objective is to determine the n components of generalized forces "torques for revolute joints and forces for prismatic joints "that enable the execution of a motion $q(t)$ such that the manipulator moves according to the specified trajectory $q_d(t)$. After considering the influence of various factors, such as the transmissions, backlash, and armature currents of the motors, as well as the vector of armature voltages, armature resistances, and voltage constants of the motors, we also take into account the matrix of gains of the amplifiers and the vector of control voltages of the servomotors. The dynamic model of the system, encompassing the manipulator and its drives, is expressed by:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F\dot{q} + g(q) = u \tag{3.2}$$

let's consider the problem of tracking a joint space trajectory. We approach this within the context of controlling nonlinear multivariable systems. The dynamic model of an n-joint manipulator is represented by equation (3.2),

which can be rephrased as follows:

$$B(q)\ddot{q} + n(q, \dot{q}) = u \tag{3.3}$$

where

$$n(q, \dot{q}) = C(q, \dot{q})\dot{q} + F\dot{q} + g(q) \tag{3.4}$$

The approach presented here is based on the concept of determining a control vector u as a function of the system state that achieves a linear input/output relationship. Rather than seeking an approximate linearization, the goal is to achieve an exact linearization of the system dynamics through nonlinear state feedback. The potential to find such a linearizing controller is ensured by the specific structure of the system dynamics. The equation in (3.3) exhibits linearity in the control u and involves a full-rank matrix $B(q)$, which is invertible for any manipulator configuration. By taking the control u as a function of the manipulator state in the form:

$$u = B(q)y + n(q, \dot{q}) \tag{3.5}$$

This approach leads to a system described by:

$$\ddot{q} = y$$

where y results in a new input vector whose expression is yet to be determined; the resulting block scheme is illustrated in Fig. 3.1.

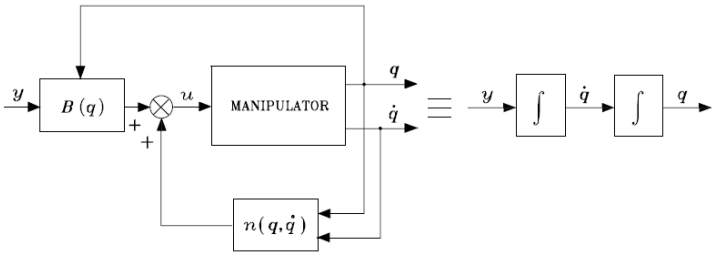

Figure 3.1: (Exact linearization performed by inverse dynamics control from[36])

The control law in (3.5) is termed inverse dynamics control because it is based on computing the manipulator's inverse dynamics. The system under control, exhibits linearity and decoupling with respect to the new input y. In simpler terms, each component y_i influences only the joint variable q_i through a double integrator relationship, independently of the motion of other joints. With the adoption of this control law, the manipulator control problem is simplified to the task of determining a stabilizing control law y. To achieve this objective, the selection of

$$y = -K_P q - K_D \dot{q} + r \tag{3.6}$$

we get a system of second-order equations:

$$r = \ddot{q} + K_D \dot{q} + K_P q \tag{3.7}$$

which, under the assumption of positive definite matrices K_P and K_D, ensures asymptotic stability. Selecting K_P and K_D as diagonal matrices of the form:

$$K_P = diag\{\omega_{n1}^2, \ldots, \omega_{nn}^2\}$$

$$K_D = diag\{2\zeta_1\omega_{n1}, \ldots, 2\zeta_n\omega_{nn}\}$$

We obtain a decoupled system, each reference component r_i only influences the joint variable q_i. with a second-order input/output relationship characterized by a natural frequency ω_{ni} and a damping ratio ζ_i. Given any desired trajectory $q_d(t)$, tracking this trajectory for the output $q(t)$ is ensured be selecting

$$r = \ddot{q}_d + K_D \dot{q}_d + K_P q_d \tag{3.8}$$

If we substitute (3.8) into (3.7) we get the homogeneous second-order differential equation

$$\ddot{\bar{q}} + K_D \dot{\bar{q}} + K_P \bar{q} = 0 \tag{3.9}$$

The dynamics of the position error $\bar{q} = q_d - q$ while tracking the given trajectory.This error occurs only if $q(0)$ and/or $\dot{q}(0)$ are different from zero and it converges to zero with a speed that depends on the matrices K_P and

K_D as shown in the simulation results in figures (3.2 and 3.3).

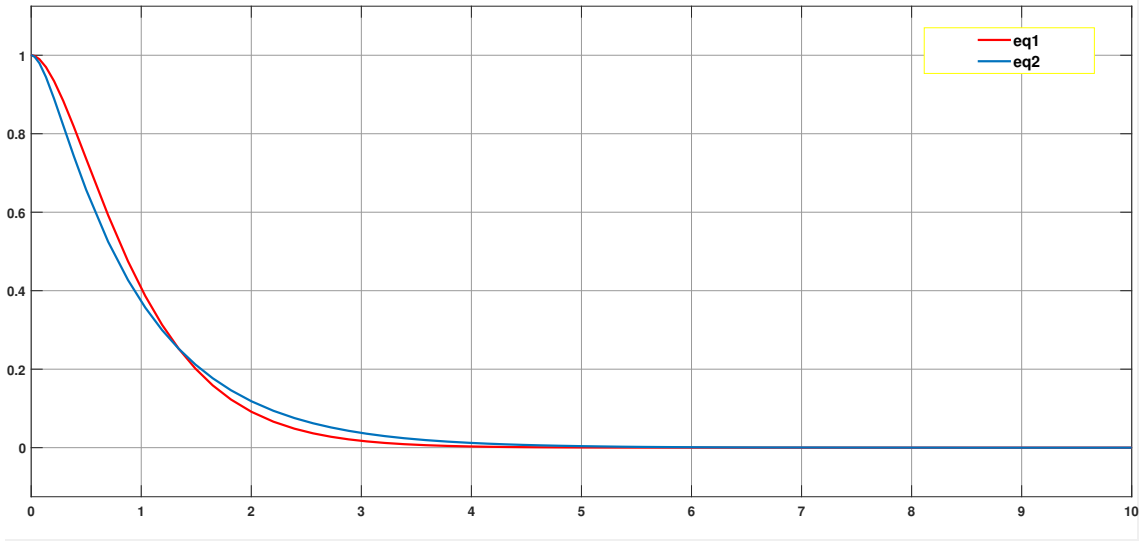

Figure 3.2: (Error convergence with setting small gains $k_{p1} = 4$, $k_{p2} = 9$ and $k_{d1} = 4$, $k_{d2} = 9$)

We can see the simulation results shown in figures (3.2 and 3.3) on the 2R Robot that we got it's dynamical model in the previous chapter where we set the parameters to:

$$a_1 = 1 \quad a_2 = 1 \quad \ell_1 = 0.5 \quad \ell_2 = 0.5$$

$$m_{\ell_1} = 50 \quad m_{\ell_2} = 50 \quad I_{\ell_1} = 10 \quad I_{\ell_2} = 10$$

$$g = 9.81 \quad k_{r1} = k_{r2} = 0 \quad m_{m1} = m_{m2} = 0 \quad I_{m1} = I_{m2} = 0$$

Where $a_{1,2}$ are the lengths of links,$\ell_{1,2}$ are the distances of the center of mass of the two links,$m_{\ell_{1,2}}$ are the masses of the links,$I_{\ell_{1,2}}$ are the moments of inertia of the links,g is the gravity,$k_{r1,2}$ are the gears reduction ratios of Motors,$m_{m1,2}$ are the masses of the motors,and $I_{m_{1,2}}$ are the moments of inertia of the motors.

Figure (3.4) depicate the resulting block scheme. The block scheme illustrates two feedback loops: an inner loop based on the manipulator's dynamic model and an outer loop operating on the tracking error. The ob-

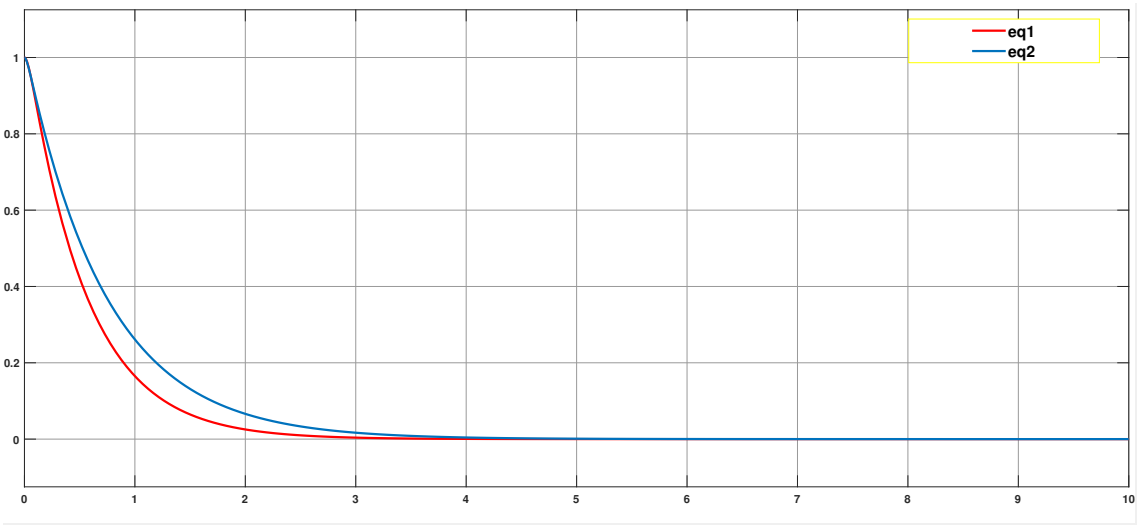

Figure 3.3: (Error convergence setting big gains $k_{p1} = 49, k_{p2} = 64$ and $k_{d1} = 28, k_{d2} = 48$)

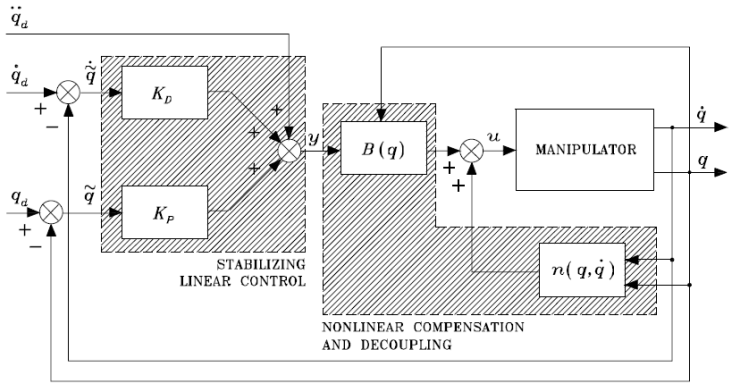

Figure 3.4: (Block scheme of joint space inverse dynamics control from[36])

jective of the inner loop is to achieve a linear and decoupled input/output relationship. On the other hand, the outer loop is responsible for stabilizing the entire system. The controller design for the outer loop is simplified as it operates on a linear and time-invariant system. It is important to note that the implementation of this control scheme involves computing the inertia matrix $B(q)$ and the vector of Coriolis,centrifugal,gravitational,and damping terms $n(q, \dot{q})$ in (3.4). These terms must be computed online since the control now relies on nonlinear feedback of the current system state. As a result, it is not possible to precompute these terms offline,The control technique of nonlinear compensation and decoupling is highly appealing because it transforms the nonlinear and coupled manipulator dynamics into n linear and decoupled second-order subsystems.

However, the technique relies on the assumption of perfect cancellation of dynamic terms, leading to concerns about sensitivity and robustness due to inevitable imperfect compensation. The implementation of inverse dynamics control laws necessitates accurate knowledge of the system's dynamic model parameters and real-time computation of the complete equations of motion.

However, these conditions can be challenging to meet in practical situations. On one hand, the model is typically known with some degree of uncertainty, owing to imperfect knowledge of manipulator mechanical parameters, presence of unmodeled dynamics, and model dependence on end-effector payloads, which are not precisely known and, therefore, not perfectly compensated. Additionally, the computation of inverse dynamics needs to be executed at sampling intervals on the order of a millisecond to maintain the assumption of operating in the continuous time domain. This requirement can impose significant constraints on the hardware and software architecture of the control system. Consequently, in certain situations, it may be prudent to reduce the computational burden by calculating only the dominant terms of the inverse dynamics. Based on the observations mentioned above, it becomes apparent that from an implementation perspective, compensation can be imperfect due to both model uncertainty and approximations made during on-line computation of inverse dynamics

simulation results are shown in figure (3.5) .

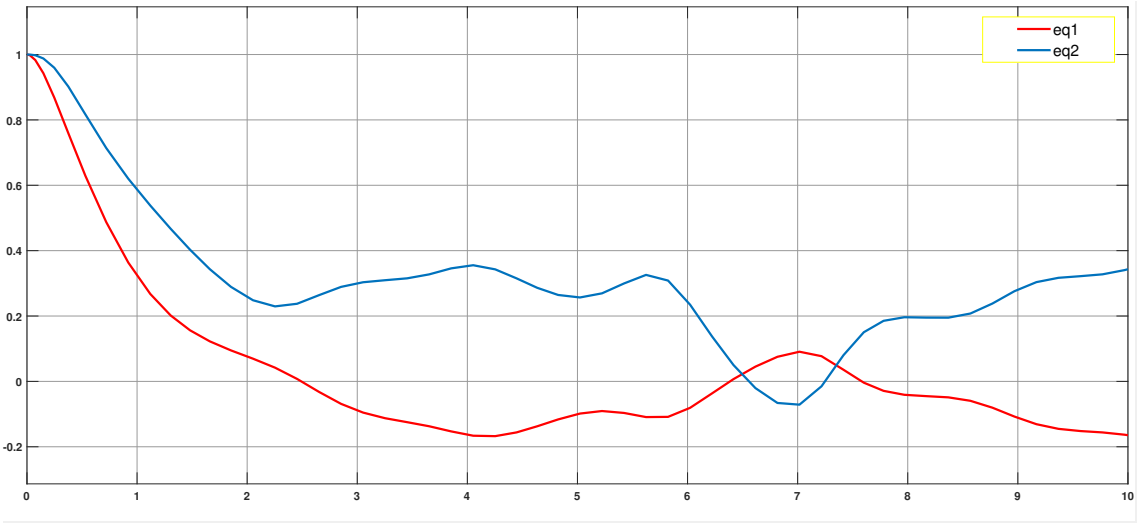

Figure 3.5: Error behavior with the presence of uncertainties

In the following section we will introduce control techniques that aim to counteract the effects of imperfect compensation. In [36], the authors propose an approach that involves introducing an additional term to the inverse dynamics controller. This inclusion enhances the system's robustness by compensating for the approximations made during on-line computation of inverse dynamics. However, this approach relies on three assumptions, two of which are practically guaranteed, while the third requires prior knowledge to be applicable. To overcome the requirement for previous knowledge and ensure robust control, we will introduce an alternative approach in the following sections. This new approach calculates the robustness term based on feedback from the system, making it independent of prior knowledge.

3.2 Robust Control

The control problem is to create a control strategy that guarantees tracking of desired joint angle trajectory $q_d(t)$, characterized by bounded first derivative $\dot{q}_d(t)$ and second derivative $\ddot{q}_d(t)$. This control law should en-

sure that the actual position and velocity trajectories $q(t)$ and $q(t)$ follow the reference trajectories $q_d(t)$ and $\dot{q}_d(t)$ with both a specified convergence rate and a desired level of accuracy. This control approach should remain effective regardless of initial conditions and variations within permissible uncertainties. Additionally, the feedback loop will utilize available measurements of joint angles and rates to achieve these tracking objectives.[31]

In scenarios which involves imperfect compensation and uncertainties it's reasonable to consider $\hat{B}$ and $\hat{n}$ which represent the adopted computational model expressed in terms of estimates of the terms in the dynamic model that arises from imperfect model compensation and intentional simplification in inverse dynamics computation. The uncertainty is represented by :

$$\tilde{B} = \hat{B} - B \qquad \tilde{n} = \hat{n} - n \tag{3.10}$$

Taking these estimates into account our control vector expressed by equation (3.3) becomes:

$$u = \hat{B}(q)y + \hat{n}(q, \dot{q}) \tag{3.11}$$

by making use of (3.11) as a non linear control law we get :

$$B\ddot{q} + n = \hat{B}y + \hat{n} \tag{3.12}$$

The functional dependence has been omitted in the above representation. Since the inertia matrix B is invertible,we can write:

$$\ddot{q} = y + (B^{-1}\hat{B} - I)y + B^{-1}\tilde{n} = y - \eta \tag{3.13}$$

where:

$$\eta = (I - B^{-1}\hat{B})y - B^{-1}\tilde{n} \tag{3.14}$$

by using feedback and PD gains that will be defined later we can take y as :

$$y = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q)$$

substituting this into (3.13) we get η as

$$\ddot{\tilde{q}} + K_D\dot{\tilde{q}} + K_P\tilde{q} = \eta \tag{3.15}$$

at this point the system described by (3.15) remains nonlinear and coupled because η is a nonlinear function of q and $\dot{q}$. Merely relying on the term on the left-hand side does not guarantee error convergence to zero. In order to achieve error convergence to zero while tracking a trajectory, especially in the presence of uncertainties, a simple linear PD control is no longer adequate. To address this, the Lyapunov direct method can be employed to design an outer feedback loop on the error, which must be robust to the uncertainty η in (3.14).

Since our desired trajectory is assigned in the joint space where $\tilde{q} = q_d - q$ is the position error,we get it's first time-derivative as $\dot{\tilde{q}} = \dot{q}_d - \dot{q}$ and it's second time-derivative relying on (3.13) as

$$\ddot{\tilde{q}} = \ddot{q}_d - y + \eta \tag{3.16}$$

at this point we can introduce the system state as ξ which can be written as:

$$\xi = \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} \tag{3.17}$$

By the derivation of ξ we get the first-order differential matrix equation as:

$$\dot{\xi} = H\xi + D(\ddot{q}_d - y + \eta) \tag{3.18}$$

in which H is a block matrix of dimension ($2n$ x $2n$),and D is also a block matrix of dimension ($2n$ x n) that has the values :

$$H = \begin{bmatrix} O & I \\ O & O \end{bmatrix} \quad D = \begin{bmatrix} O \\ I \end{bmatrix} \tag{3.19}$$

After reaching this point our problem of tracking a given trajectory can be addressed by finding a control law y . The objective of the control law y is

to ensure asymptotic stability of (3.18) for any variation of η .

with reference to (3.13) we choose:

$$y = \ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q} + w \tag{3.20}$$

In the proposed control law, the PD term ensures the stabilization of the error dynamic system matrix, $\ddot{q}_d$ provides a feedforward term, and the term w is chosen to ensure robustness to the effects of uncertainty described by η in (3.14). In (3.20) if we set $K = [K_P \quad K_D]$ we get

$$\dot{\xi} = \tilde{H}\xi + D(\eta - w) \tag{3.21}$$

in (3.21) $\tilde{H}$ is the subtraction of H minus D where D is multiplied by K :

$$\tilde{H} = (H - DK) = \begin{bmatrix} O & I \\ -K_P & -K_D \end{bmatrix}$$

that is a matrix with all its eigenvalues have negative real parts,given that K_P and K_D are positive definite.

$$K_P = \begin{bmatrix} k_{p_{11}} & 0 & \dots & 0 \\ 0 & k_{p_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_{p_{nn}} \end{bmatrix}, K_D = \begin{bmatrix} k_{d_{11}} & 0 & \dots & 0 \\ 0 & k_{d_{22}} & 0\dots & 0 \\ \vdots & 0\vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_{d_{nn}} \end{bmatrix}$$

By selecting K_P and K_D as a diagonal matrix with the diagonal elements $(k_{p_{11}}, k_{p_{22}}, \dots, k_{p_{nn}})$,$(k_{d_{11}}, k_{d_{22}}, \dots, k_{d_{nn}})$are positive and greater than zero, and all off diagonal elements are zeros.This choice allows the desired error system dynamics to be prescribed. the system can be decoupled into n independent equations as regards the linear part. If the uncertainty term vanishes, i.e., $w = 0$, the result with an exact inverse dynamics controller is recovered ($\hat{B} = B$ and $\hat{n} = n$).

In order to determine the value of w, we can consider the following

positive definite quadratic form as a candidate for the Lyapunov function:

$$V(\xi) = \xi^T Q \xi > 0 \quad \forall \xi \neq 0 \tag{3.22}$$

where Q is a positive definite matrix of dimensions ($2n$ x $2n$), if we compute the derivative of V along the trajectories of the error system (3.21) we get

$$\dot{V} = \dot{\xi}^T Q \xi + \xi^T Q \dot{\xi} = \xi^T (\tilde{H}^T Q + Q\tilde{H})\xi + 2\xi^T QD(\eta - w) \tag{3.23}$$

Given that $\tilde{H}$ has eigenvalues with all negative real parts, it is a well-known fact that for any symmetric positive definite matrix P, the equation

$$\tilde{H}^T Q + Q\tilde{H} = -P \tag{3.24}$$

gives a unique solution Q that is symmetric positive definite. Now (3.23) is

$$\dot{V} = -\xi^T P \xi + 2\xi^T QD(\eta - w) \tag{3.25}$$

The first term on the right-hand side of equation (3.25) is negative definite, which implies that the solutions converge if $\xi \in \mathcal{N}(D^T Q)$. but instead if $\xi \notin \mathcal{N}(D^T Q)$, the control w must be chosen in a way that makes the second term in equation (3.25) less than or equal to zero.

If we introduce $z = D^T Q \xi$, the second term in (3.25) now is $z^T(\eta - w)$ by making :

$$w = \frac{\rho}{\|z\|} z \qquad \rho > 0 \tag{3.26}$$

where It is essential to divide z by its norm to achieve a linear dependence on z in the term containing the control $z^T w$. By doing so, it effectively counteracts,for $z \to 0$ the term containing the uncertainty $z^T \eta$, which is linear in z. from(3.26) we get :

$$z^T(\eta - w) = z^T \eta - \frac{\rho}{\|z\|} z^T z \leq \|z\|\|\eta\| - \rho\|z\| = \|z\|(\|\eta\| - \rho) \tag{3.27}$$

so now the main focus to make sure of the convergence we need to choose

ρ such that :

$$\rho \geq \|\eta\| \qquad \forall q, \dot{q}, \ddot{q}_d \tag{3.28}$$

if we make so the control law (3.26) guarantees that $\dot{V}$ is less than zero along all trajectories of the error system.

To achieve this goal the writers in[36] propose some assumptions were the control design is reliant on the assumption that, while the uncertainty η is unknown, an estimate of its range of variation is accessible to ensure asymptotic stability.

At this point they introduced three assumptions considering the η is a a function of q , $\dot{q}$, and $\ddot{q}$ that are:

$$sup_{t\geq 0}\|\ddot{q}_d\| < Q_M < \infty \quad \forall \ddot{q}_d \tag{3.29}$$

$$\|I - B^{-1}(q)\hat{B}(q)\| \leq \alpha \leq 1 \quad \forall q \tag{3.30}$$

$$\|\tilde{n}\| \leq \phi < \infty \quad \forall q, \dot{q} \tag{3.31}$$

Assumption (3.29) is practically fulfilled since any planned trajectory cannot demand infinite accelerations. Regarding assumption (3.30), since B is a positive definite matrix with upper and lower bounded norms, the following inequality holds:

$$0 < B_m \leq \|B^{-1}(q)\| \leq B_M < \infty \quad \forall q \tag{3.32}$$

so the choice for $\hat{B}$ always exists which satisfies (3.30). In fact, by setting

$$\hat{B} = \frac{2}{B_M + B_m} I$$

and from (3.30) it is

$$\|B^{-1}\hat{B} - I\| \leq \frac{B_M - B_m}{B_M + B_m} = \alpha < 1 \tag{3.33}$$

If $\hat{B}$ is a better estimate of the inertia matrix, the inequality can be satisfied with values of α that can be arbitrarily small (in the limit, it is $\hat{B} = B$ and $\alpha = 0$).

Regarding assumption (3.31), it is observed that $\tilde{n}$ is a function of q and $\dot{q}$. For revolute joints, a periodic dependence on q is obtained, while for prismatic joints, a linear dependence is obtained. However, the joint ranges are limited, and thus the contribution mentioned above is also limited. Additionally, concerning the dependence on $\dot{q}$, unbounded velocities for an unstable system may arise in theory, but in practice, there are saturations on the maximum velocities of the motors. In total, assumption (3.31) can be realistically satisfied.

To fulfill the condition in (3.28), and by considering the definition of η in (3.14) along with assumptions (3.29)(3.31),and being $\|w\| = \rho$ it's

$$\|\eta\| \leq \|I - B^{-1}\hat{B}\|(\|\ddot{q}_d\| + \|K\|\|\xi\| + \|w\|) + \|B^{-1}\|\|\tilde{n}\|$$

$$\leq \alpha Q_M + \alpha\|K\|\|xi\| + \alpha\rho + B_M\Phi$$

Then by setting :

$$\rho \geq \frac{1}{1-\alpha}(\alpha Q_M + \alpha\|K\|\|\xi\| + B_M\Phi) \tag{3.34}$$

we have :

$$\dot{V} = -\xi^T P\xi + 2z^T\left(\eta - \frac{\rho}{\|z\|}z\right) < 0 \quad \forall\xi \neq 0 \tag{3.35}$$

The resulting block scheme is illustrated in Fig(3.6) and the simulation results for error convergence are shown in Fig(3.7)

However, obtaining such estimates can be challenging in certain situations. This difficulty arises due to the presence of significant flexibility resulting from variation in payload, unanticipated disturbances, and the evolution of time-dependent factors like friction and parameters associated with the robot's aging(wear and tear).[50][51].

Given the circumstances, it is now more practical to seek a solution that ensures error convergence without relying on any assumptions regarding

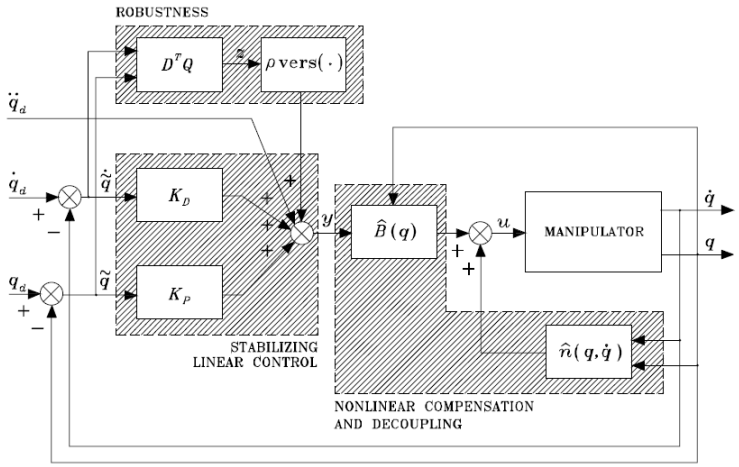

Figure 3.6: (Block scheme of joint space robust control from[36])

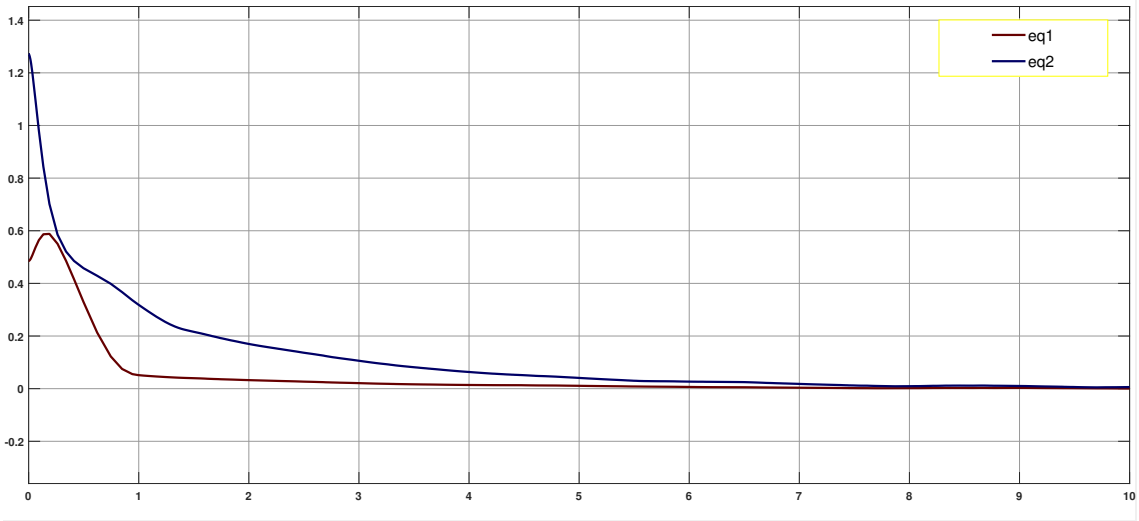

Figure 3.7: (Error Convergence RAW=16)

prior knowledge or estimates of the range of variations in uncertainties η.

To initiate this approach, we begin by defining (u), which represents the input to the manipulator as:

$$\hat{B}(q)y + \hat{n}(q, \dot{q}) = u \tag{3.36}$$

Furthermore, we maintain consistency by selecting the same control law:

$$y = K_P\tilde{q} + K_D\dot{\tilde{q}} + \ddot{q}_d + \omega \tag{3.37}$$

In this control law, the PD terms play a crucial role to insure stability of the error dynamic system matrix. Additionally, $\ddot{q}_d$ serves as the feed-forward term, And the important parameter, ω, that plays a pivotal role in accounting for the robustness against the effects of η, which represents the uncertainties.

To determine the value of ω and given that our candidate for the Lyapunov function is:

$$V(\xi) = \xi^T Q \xi \tag{3.38}$$

This Lyapunov function in (3.38) is greater than zero for all ξ except when $\xi = 0$. ie:

$$V(\xi) > 0 \quad \forall \xi \neq 0$$

Indeed, the first condition for the Lyapunov function, which involves the matrix Q being positive definite (as computed and explained previously), is satisfied. This positive definiteness is a crucial requirement to ensure error convergence in the context of Lyapunov stability analysis.[18]

The second condition pertains to radial unboundedness, indicating that the Lyapunov function should exhibit unbounded growth as the state vectors move away from the origin. In mathematical terms, this condition can be expressed as follows: $V(\xi) \to \infty$ as $\xi \to \infty$. It is evident that our chosen candidate satisfies this condition.[18]

Concerning the condition of having the derivative of the Lyapunov function less than zero for all ξ not equal to zero($\dot{V}(\xi) < 0 \quad \forall \xi \neq 0$)][18],

and by examining the derivative of our Lyapunov candidate which is:

$$\dot{V} = \dot{\xi}^T Q \xi + \xi^T Q \dot{\xi} \tag{3.39}$$

By performing the same calculations as outlined previously we get :

$$\dot{V} = -\xi^T P \xi + 2z^T (\eta - \frac{\rho}{\|z\|} z) \tag{3.40}$$

As previously proposed, by setting

$$\omega = \frac{\rho}{\|z\|} z$$

, our task now simplifies to choosing a value for ρ that ensures $\dot{V} < 0$ for all ξ not equal to 0. This is achieved without requiring prior knowledge about the uncertainties.

It's important to recall that ρ, as discussed earlier, must be greater than or equal to the norm of η for all $q, \dot{q}, \ddot{q}_d$.

Examining equation (3.39), we notice that Q is a constant matrix determined by $\tilde{H}$ and P, which are also constants. Given that we have access to ξ and $\dot{\xi}$ through online feedback, we can contemplate treating ρ as a variable and defining its value online. This can be done by continuously monitoring the value of $\dot{V}$ and updating the value of ρ dynamically. The goal is to adjust ρ such that it becomes sufficiently large to ensure $\dot{V}(\xi) < 0 \quad \forall \xi \neq 0$, thereby achieving the desired stability and convergence properties.

By adopting this approach and considering that the previously discussed assumptions (3.29) and (3.31) are realistically met, there's no requirement for estimates regarding the range of variations in the unknown uncertainty η in order to compute ρ as described in (3.34). Instead, this method allows $\rho$to be dynamically updated online based on the value of $\dot{V}$. This dynamic update ensures that ρ adapts in real-time to maintain the desired stability properties without relying on prior knowledge of η's variations.

By implementing this approach, we have determined the parameter

(ω), which serves as the robust component designed to counteract the effects of uncertainties present in ($\hat{B}$) and ($\hat{n}$) when calculating the nonlinear components that depend on the manipulator's state. Crucially, this calculation is performed without the need for any prior knowledge about the characteristics or specifics of these uncertainties.

To determine ρ, we require a mechanism that incrementally increases its value until ($\dot{V}(\xi) < 0$) is attained. Once this goal is reached, it becomes crucial to sustain the value of ρ as long as the condition remains satisfied. In cases where the condition is no longer met, it becomes necessary to recalibrate ρ once again to ensure the continuous fulfillment of the condition.

To achieve this, we can make use of the derivative of ρ. By setting the derivative equal to a constant multiplied by the current time value, we can systematically boost ρ as time elapses when our condition is unfulfilled. Conversely, when the condition is met, setting the derivative of ρ to zero will maintain its current value.

The function of $\dot{\rho}$ can be written as :

$$\dot{\rho}(t) = \begin{cases} K & \text{If} \quad \dot{V} > 0 \\ 0 & \text{If} \quad \dot{V} < 0 \end{cases}$$

By integrating ($\dot{\rho}$) and incorporating it into the controller, while utilizing the relationship $\omega = \frac{\rho}{\|z\|} z$, we effectively determine the value of ω. This approach allows us to dynamically compute and adjust ω in response to the system's behaviour, which contributes to the robustness against uncertainties and helps achieve the desired control objectives.

In summary, the approach presented has resulted in the development of a control law comprised of three distinct contributions:

- The term $\hat{B}y + \hat{n}$ serves to approximately compensate for nonlinear effects and achieve joint decoupling.
- The term $\ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q}$ introduces both linear feedforward action ($\ddot{q_d} + K_D\dot{q_d} + K_P q_d$) and linear feedback action ($-K_D\dot{q} - K_P q$). These actions collectively serve to stabilize the error system dynamics.

- The term $w = \frac{\rho}{\|z\|} z$ signifies the robust component designed to counterbalance the uncertainties in $\tilde{B}$ and $\tilde{n}$ when computing the nonlinear terms linked to the manipulator's state. Notably, the magnitude of the scalar ρ increases with greater uncertainty. This control law can be described as a unit vector type because it is represented by a vector with a magnitude of ρ aligned along the unit vector of $z = D^T Q \xi, \quad \forall \xi$.

3.2.1 Overview of Sliding Mode Control (SMC)

In this section, we will provide a general definition and overview of Sliding Mode Control (SMC). Sliding Mode Control is a critical concept in our study, and by grasping its fundamentals, we can better interpret and analyze the outcomes of our simulations.

Sliding mode control (SMC) is a robust control technique used to generate control inputs for achieving desired trajectories in a given system. These inputs may take on a discontinuous form depending on the system's state. Robustness is a fundamental characteristic of SMC, meaning that it can effectively achieve its control objectives even in the presence disturbances or uncertainties affecting the system. This robustness is a key feature of SMC, as demonstrated in various studies (e.g., [7] [40] [48] and related references).[11]

Sliding mode control falls under the broader category of Variable Structure Control Systems (VSCS). In SMC, VSCS are purposefully designed to drive and maintain the system state within a defined region determined by a switching function. This approach offers several advantages. Firstly, it allows for tailoring the dynamic behavior of the system by selecting an appropriate switching function. Secondly, it provides a high degree of robustness to the closed-loop response, rendering it highly insensitive to specific types of system uncertainties. Moreover, the analysis of discontinuous signals applied to the system can be utilized to model the signal activity required for achieving optimal system performance.

The concept of Variable Structure Control Systems originated in the 1960s in the Soviet Union and has since evolved into a well-established and

mature approach for achieving robust control and estimation in a wide range of applications.[32]

The SMC design process comprises two fundamental steps:

1- Crafting a stable surface within the system's state space in Rn.

2- Formulating a control input that constrains the trajectory of the system defined in step (1) within a predetermined vicinity of the established stable surface. This restriction ensures that the trajectory converges towards the origin while remaining within the specified neighborhood.

In practical SMC design, for the sake of simplicity and ease of implementation, the mentioned "surface" is typically a subspace within the system's state space. The selection of a stable subspace ensures that every trajectory, when confined to the vicinity of this subspace, ultimately converges to the origin of the state space [7] [40] [42] [48]. The SMC theory hinges on the existence of these stable sliding subspaces, and various systematic and ad hoc methods for identifying such stable subspaces are available in the literature [7] [40] [42] [48].

Under the robust control scheme described before , all resulting trajectories converge to the subspace defined by $z = D^T Q\xi = 0$. This subspace, referred to as the sliding subspace, is determined by the matrix Q used in the Lyapunov function V. Within this attractive sliding subspace, the control input w ideally switches at an infinite frequency, leading to the convergence of all error components to zero. The speed of this convergence depends on the characteristics of the matrices Q, K_P, K_D,and K which is the variable used to compute ρ (how fast ρ grow in time).

In a two-dimensional scenario, the behavior of an error trajectory is illustrated in Figure 3.8 . Notably, when the initial condition $\xi(0) \neq 0$ and $\xi(0) \notin \mathcal{N}(D^T Q)$, the trajectory is attracted to the sliding hyperplane (a line) $z = 0$. It then approaches the origin of the error state space, and this evolution is governed by the value of ρ.

In practice, physical limitations on the controller's components lead to a control signal that switches at a finite frequency. Consequently, the trajectories oscillate around the sliding subspace with an amplitude that decreases as the frequency of switching increases.

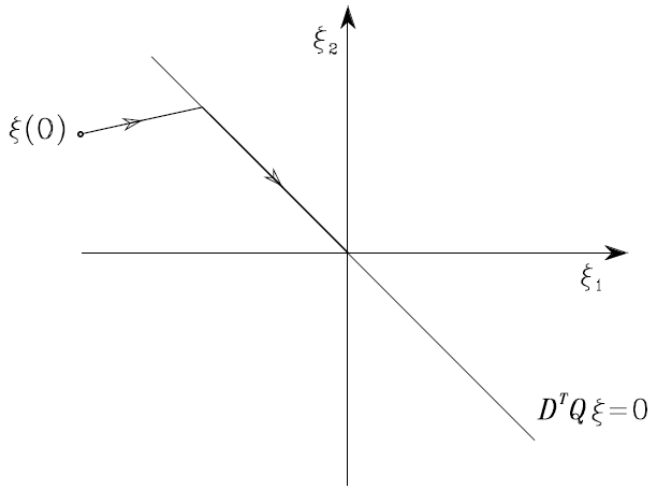

Figure 3.8: (Error trajectory with robust control from[36])

To eliminate the undesired high-frequency components (chattering) in control systems, one approach is to adopt a robust control law that, while it may not guarantee error convergence to zero, does ensure that the errors remain bounded within a specified norm. A control law of this type can be designed as follows:

$$\omega = \begin{cases} \frac{\rho}{\|z\|} z & \text{if} \quad \|z\| \geq \epsilon \\ \frac{\rho}{\epsilon} z & \text{if} \quad \|z\| < \epsilon \end{cases} \tag{3.41}$$

To offer an intuitive interpretation of this control law, let's consider that equation (3.41) results in a zero control input when the error exists in the null space of the matrix D^TQ. Conversely, equation (3.26) features a gain that tends to infinity as z approaches the null vector, resulting in a control input with a limited magnitude. These control inputs, despite their differing behaviors, operate at an infinite switching frequency, compelling the error system dynamics to remain confined to the sliding subspace.

In the example mentioned earlier, control law (3.41) establishes a hyperplane $z = 0$ that is no longer attractive, permitting the error to fluctuate

within a boundary layer whose thickness depends on ϵ (see Fig.3.9).

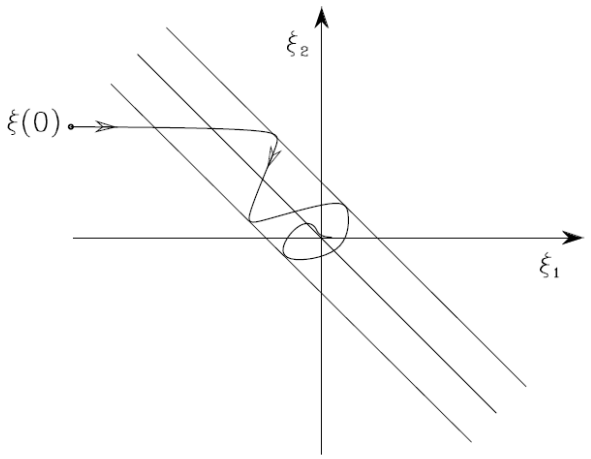

Figure 3.9: (Error trajectory with robust control and chattering elimination from[36])

3.2.2 SIMULATION

In this section, we present a comprehensive set of results to provide a detailed insight into the performance of our robust controller, which incorporates the online computation of ρ, in the context of tracking control for a two-link rigid manipulator. We go beyond a singular perspective by presenting a range of scenarios that involve variations in the value assigned to constant K and see how ρ change over time and the effects on the error convergence.

Moreover, our presentation extends to an investigation to apply the previously explained solution to eliminate the undesired high-frequency components (chattering) in control systems,

The simulation was done using Matlab and Simulink,

The prescribed joint trajectories are defined as follows:

For the first joint: $q_{d1} = 0.25\pi + 0.5(1 - \cos(0.5\pi t))$

For the second joint: $q_{d2} = 0.5\pi + 0.25(1 - \cos(\pi t))$

where the 2R robot whose dynamical model was derived previously and for our purpose of analysis, we assigned specific parameter values to the system as follows:

Link Lengths: - $a_1 = 1$ - $a_2 = 1$

Center of Mass Distances: - $\ell_1 = 0.5$ - $\ell_2 = 0.5$

Link Masses: - $m_{\ell_1} = 50$ - $m_{\ell_2} = 50$

Link Moments of Inertia: - $I_{\ell_1} = 10$ - $I_{\ell_2} = 10$

Gravity: - $g = 9.81$

Motor Gear Reduction Ratios: - $k_{r1} = k_{r2} = 0$

Motor Masses: - $m_{m1} = m_{m2} = 0$

Motor Moments of Inertia: - $I_{m1} = I_{m2} = 0$

These parameters define the physical characteristics and properties of our 2R robot, which are essential for the subsequent analysis and control design.

To simulate uncertainties in the system, the controller was provided with altered parameter values, introducing the following modifications:

Modified Link Lengths: - $\hat{a}_1 = 0.5$ - $\hat{a}_2 = 0.7$

These modified parameters ($\hat{a}_1$ and $\hat{a}_2$) represent the variations in the lengths of the robot's links, introducing uncertainty into the system for the purpose of analysis and robust control evaluation.

We have also configured the following values in our MATLAB code:

$$K_p = \begin{bmatrix} 49 & 0 \\ 0 & 68 \end{bmatrix}$$

$$K_d = \begin{bmatrix} 15 & 0 \\ 0 & 28 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By utilizing MATLAB's Lyapunov solver with the command `Q = lyap(H_TILDA',`

P), we obtained the matrix Q:

$$Q = \begin{bmatrix} 1.8197 & 0 & 0.0102 & 0 \\ 0 & 1.4380 & 0 & 0.0074 \\ 0.0102 & 0 & 0.0340 & 0 \\ 0 & 0.0074 & 0 & 0.0181 \end{bmatrix}$$

The effect of the constant K

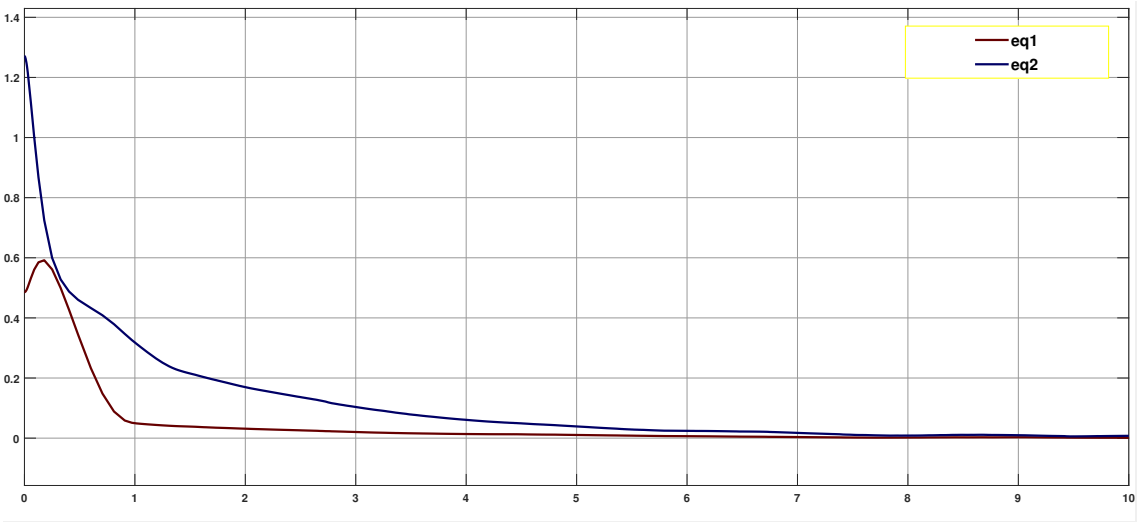

Figure 3.10: (Error Convergence K=1000)

As illustrated in Figure (3.11), where we have chosen the value of the constant $K = 1000$ our ρ grows rapidly, reaching a magnitude significant enough to make ω able to counteract the uncertainties throughout the entire trajectory.

The error can be observed in the figure (3.10), where the error converges towards the sliding hyperplane (a line) $z = 0$. and ultimately approaches to zero.

While in Figure (3.13), where we observe the behavior of ρ with $K = 15$, it becomes apparent that ρ increases multiple times. This increase is necessary to effectively adapt to the effects of uncertainties that impact the convergence of the error throughout the trajectory. We can closely examine

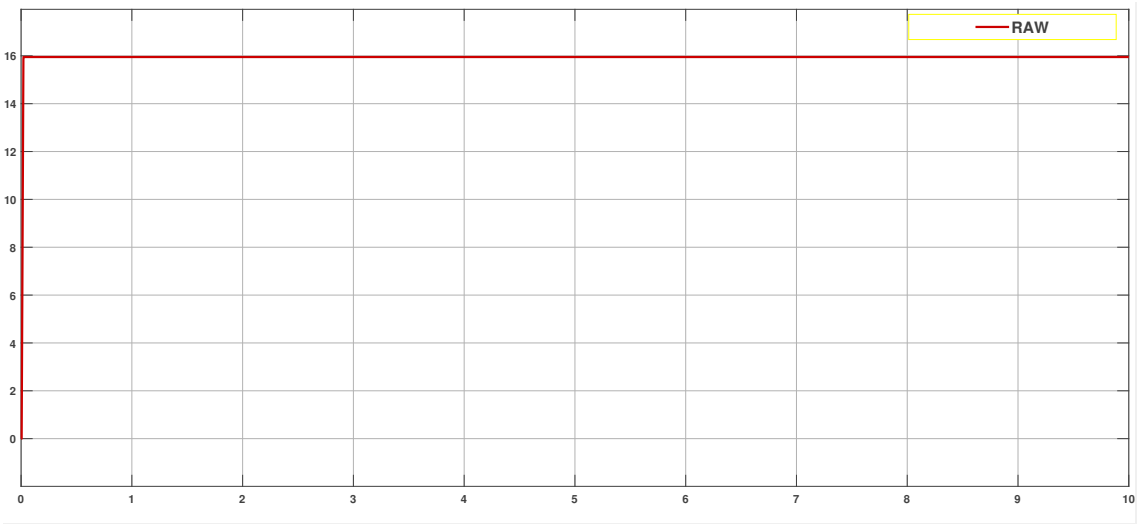

Figure 3.11: (Raw evolution with K=1000)

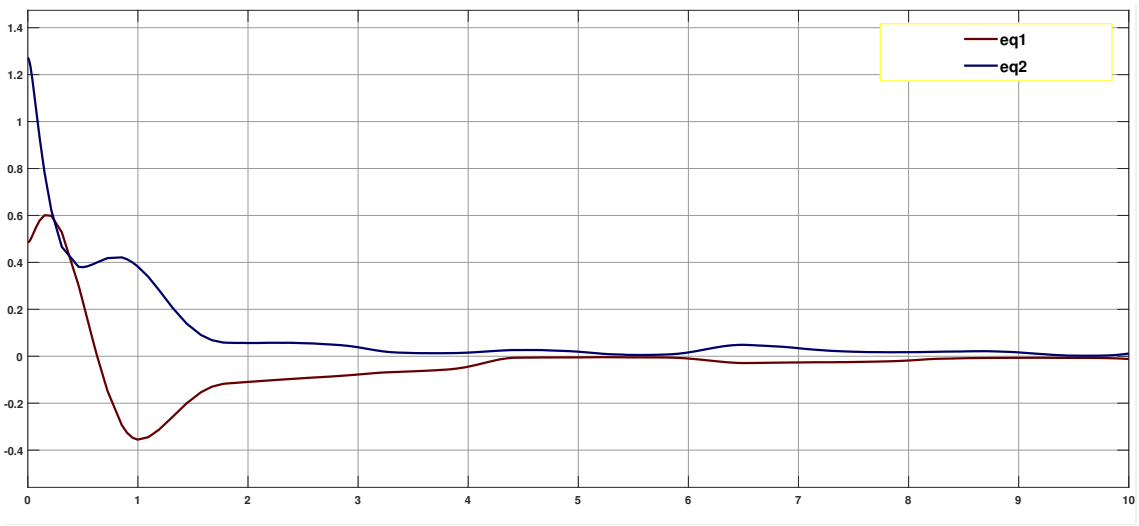

Figure 3.12: (Error Convergence K=15)

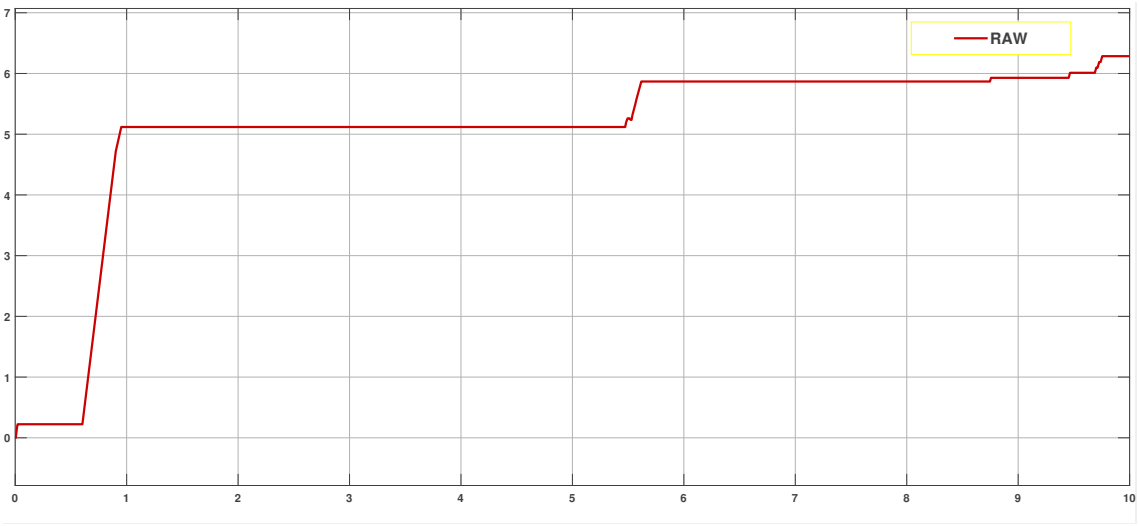

Figure 3.13: (Raw evolution with K=15)

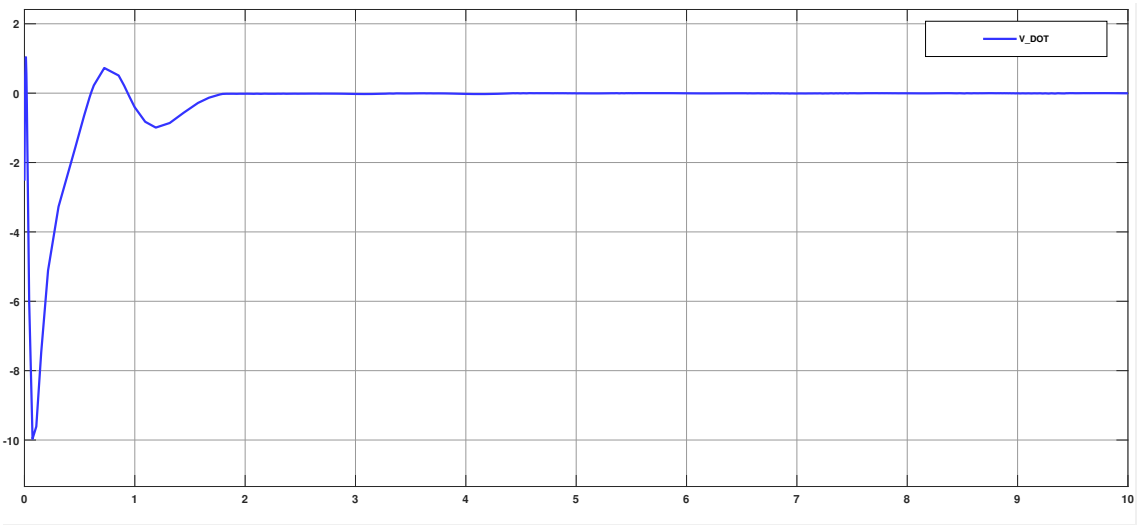

Figure 3.14: (The derivative of Lyaponove function when K=15)

the behavior of the error in Figure (3.12).

Specifically, we see that the value of ρ rises when the derivative of the Lyapunov function, as depicted in Figure (3.14), exceeds zero. This scenario causes the error to depart from the sliding hyperplane. By augmenting the value of ρ, we actively prevent $\dot{V}$ from becoming positive, successfully guiding the error back onto the sliding hyperplane.

ELIMINATING THE UNDESIRED HIGH-FREQUENCY COMPONENTS (CHATTERING)

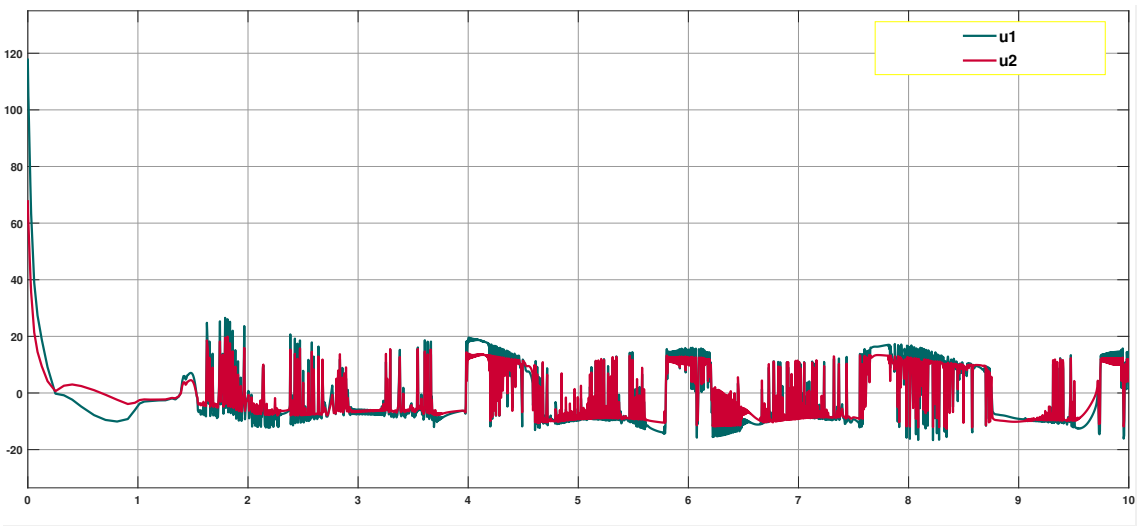

Figure 3.15: (u1,u2 with chattering)

In Figure (3.15), we observe a practical aspect of the control system. in which, the control signal exhibits a finite switching frequency.

In Figure (3.16), we can observe the control signal behavior when adopting the robust control law as defined in Equation (3.41). While this control law may not guarantee error convergence to zero, it does ensure that the errors remain bounded within a specified norm. This control strategy establishes a hyperplane $z = 0$ that is no longer an attractive equilibrium point, allowing the error to fluctuate within a boundary layer. The thickness of this boundary layer is denoted as ($\epsilon = 0.07$), as shown in Figure (3.17).

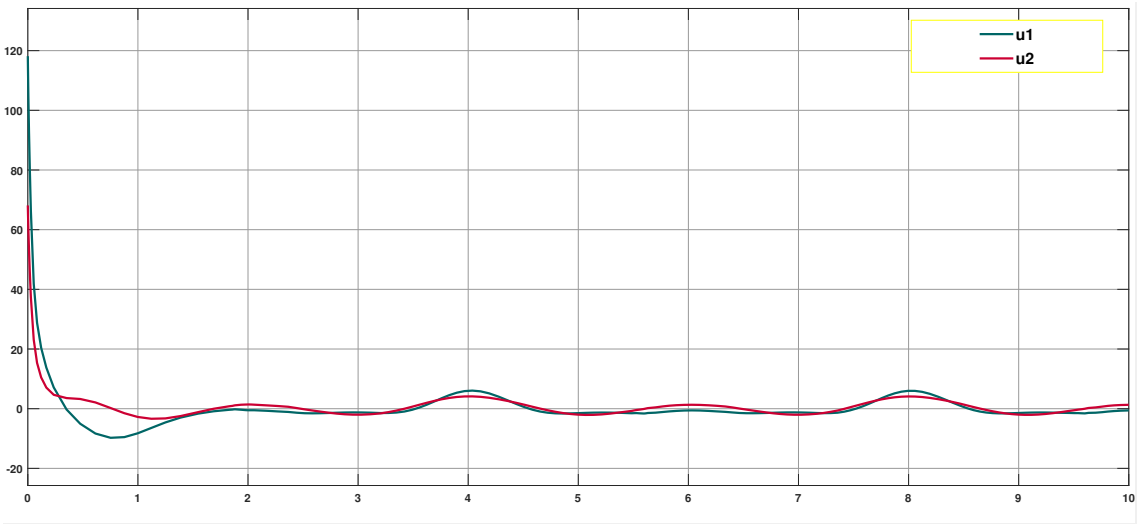

Figure 3.16: (u1,u2 without chattering)

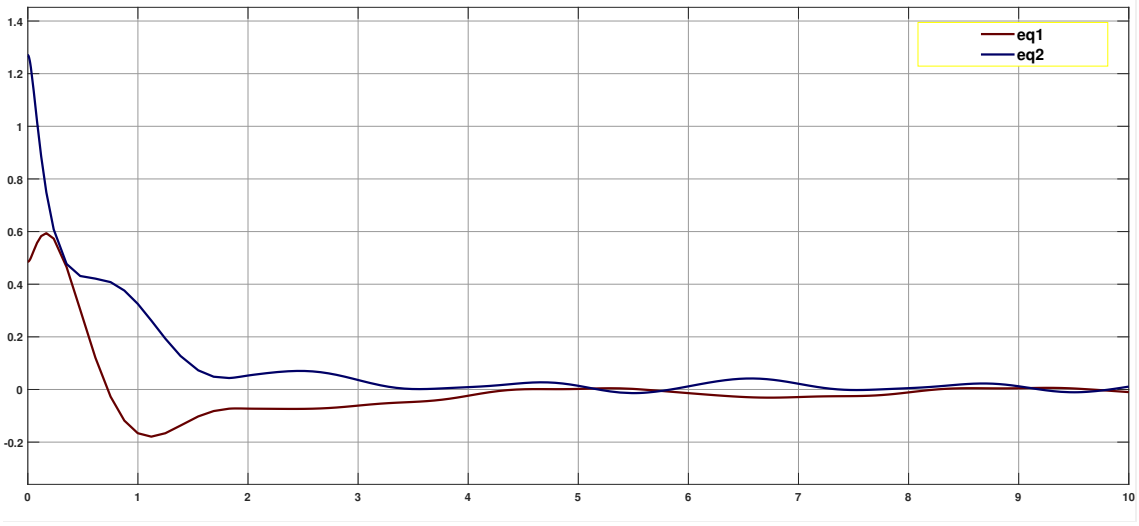

Figure 3.17: (Error Convergence,K=1000,epsilon=0.07)

3.3 Non linear PD+ controller

PD+ control strategies typically consist of two main components: a feedback component for ensuring closed-loop stability and a feedforward component designed to enhance tracking performance [43]. In many cases, the feedback component employs a linear PD controller with constant gains, while the feedforward part can take various forms, including full or partial manipulator dynamics or even be absent altogether.

When the feedforward term encompasses the complete manipulator dynamics, stability analysis of linear PD+ control methods has demonstrated global asymptotic and exponential stability, particularly in the context of tracking control for rigid manipulators [43], [41], [29], [44]. However, in scenarios where the feedforward term does not encompass the entire manipulator dynamics, the equilibrium point of the closed-loop system is no longer guaranteed to be at the origin, and tracking errors may not diminish over time. In such cases, stability and robustness analyses of linear PD+ controllers have indicated exponential convergence and uniform ultimate boundedness of tracking errors for manipulator control [43].

While the use of PD+ control laws guarantees closed-loop system stability, the overall performance is heavily influenced by the selection of controller gains. Conventional PD+ controllers with fixed gains often demand substantial initial actuator torques, and this can be problematic, especially when dealing with limitations imposed by the size of the actuators. To enhance the performance of closed-loop systems, a solution has been introduced in the form of independent joint nonlinear gain PD controllers [45], [15], [21], [34]. These controllers feature proportional and derivative gains that are nonlinear functions of the errors in joint position and velocity.

In the context of the control design described below, it is assumed that certain factors, including friction effects, disturbance terms, and joint angle velocities, have bounded characteristics as follows:

$$0 \leq k_{fd1} = \lambda_m(F_d) \quad \leq \quad \lambda_M(F_d) = k_{fd2} \tag{3.42}$$

$$k_{fs} = \sup_{\dot{q} \in R^n} \|f_s(\dot{q})\|, \quad k_{ud} = \sup_{t \in R_+} \|u_d(t)\| \tag{3.43}$$

$$k_{qd1} = \sup_{t \in R_+} \|\dot{q}_d(t)\|, \quad k_{qd2} = \sup_{t \in R_+} \|\ddot{q}_d(t)\| \tag{3.44}$$

Utilizing the equations of motion derived in prior sections, we adopt the following general control framework:

$$u = u_{fb}(q, \dot{q}, q_d, \dot{q}_d) + u_{ff}(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) \tag{3.45}$$

In this context, the control structure consists of a feedback component denoted as u_{fb} and a feedforward term represented as u_{ff}. Our focus is on examining the stability robustness of a particular category of nonlinear controllers that incorporate the feedforward component as:

$$u_{ff} = B_n(q)\ddot{q}_d + C_n(q, \dot{q})\dot{q}_d + g_n(q) \tag{3.46}$$

Here, $B_n(q)$, $C_n(q, \dot{q})$, and $g_n(q)$ represent the nominal or estimated values for $B(q)$, $C(q, \dot{q})$, and $g(q)$, respectively, and these estimates are constrained by the following bounds:

$$\delta_M = \sup_{q \in R^n} \|B(q) - B_n(q)\| \tag{3.47}$$

$$\delta_C = \sup_{q \in R^n, \|\dot{q}_d\| \leq k_{qd1}} \|C(q, \dot{q}_d) - C_n(q, \dot{q}_d) \tag{3.48}$$

$$\delta_g = \sup_{q \in R^n} \|g(q) - g_n(q)\| \tag{3.49}$$

While subsequent stability and robustness analysis relies on the existence of the mentioned bounds, only $_C$ needs to be explicitly and beforehand known to guarantee stability and robustness. In scenarios involving manipulators with known uncertainty limited to the inertia parameters, the bound $_C$ can be determined using The property of the dynamic model of

the manipulator which states that There exist a positive constant K_C such that :

$$\|C(x,y)z\| = \|C(x,z)y\| \leq K_C\|y\|\|z\| \quad \forall x,y,z \tag{3.50}$$

Where K_C satisfies the following bound [16]:

$$K_C \geq n^2 \left(\max_{i,j,k,q} |c_{ijk(q)}| \right) \tag{3.51}$$

where $c_{ijk}(q)$ is the (i,j,k)-th Christoffel symbol used in the definition of the matrix $C(q,\dot{q})$

However, in many practical applications, the primary source of uncertainty is an unknown payload, resulting in a single uncertain inertia parameter affecting the computation of δ_C. The general structure of U_{fb} is designed based on the characteristic that rigid manipulators fall into a category of mechanical systems that can be stabilized using PD-type control laws. The suggested feedback term takes the form of a general affine function of the tracking errors, as described by the following equation:

$$u_{fb} = K_p(q,\dot{q},q_d,\dot{q}_d)\tilde{q} + K_d(q,\dot{q},q_d,\dot{q}_d)\dot{\tilde{q}} \tag{3.52}$$

The matrices $K_p(q,\dot{q},q_d,\dot{q}_d)$ and $K_d(q,\dot{q},q_d,\dot{q}_d)$ are chosen as function matrices to meet stability and performance criteria. In their basic configuration, Kp and Kd are diagonal matrices with positive definite constant values, resulting in the widely used independent joint linear PD control law. However, to enhance specific performance aspects or adhere to torque constraints, the representation of u_{fb} has been expanded to include nonlinear gain PD controllers [15], [21], [34].

In order to broaden the scope of representations for u_{fb} in the context of trajectory tracking control for n-joint rigid robotic manipulators with nonlinear and interconnected dynamics, a more comprehensive structure for the gain matrices is introduced. The nonlinear controllers falling under this category are formulated as follows:

$$u_{fb} = K_p(\tilde{q})\tilde{q} + K_d(q,\dot{q},q_d,\dot{q}_d)\dot{\tilde{q}} \tag{3.53}$$

The derivative gain matrix, denoted as $K_d(q, \dot{q}, q_d, \dot{q}_d)$, is assumed to be symmetric, positive definite, and bounded for all $q, \dot{q}, q_d, \dot{q}_d$. Throughout the following discussion, we use the notation $K_d(\tilde{q}, \dot{\tilde{q}})$ to represent the symmetric, positive definite derivative gain matrix. It's important to note that the subsequent developments apply equally to the more general case of $K_d(q, \dot{q}, q_d, \dot{q}_d)$. In previous studies [15], [21], [34], to guarantee global asymptotic stability of nonlinear gain PD controllers for position control of manipulators, they have typically employed a positive definite diagonal proportional gain matrix $K_p(\tilde{q})$ with $K_{pii}(\tilde{q}_i) > 0, i = 1 \dots n$ as it's diagonal elements. In this context, the proportional gain matrix $K_p(\tilde{q})$ is assumed to be symmetric with structure:

$$K_p(\tilde{q}) = \begin{bmatrix} k_{p11}(\tilde{q}_1) & k_{p12} & \dots & k_{p1n} \\ k_{p21} & k_{p22}(\tilde{q}_2) & \dots & k_{p2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{pn1} & k_{pn2} & \dots & k_{pnn}(\tilde{q}_n) \end{bmatrix} \tag{3.54}$$

The diagonal elements of $K_p(\tilde{q})$ have both an upper bound and a positive lower bound, meaning they satisfy the following condition:

$$k_{pii}^M \geq k_{pii}(\tilde{q}_i) \geq k_{pii}^m > 0, \quad \forall \tilde{q}_i \in R \tag{3.55}$$

Defining constant symmetric positive definite matrices K_p^m and K_p^M for $i = 1, \dots, n$ as:

$$K_p^m = K_p(\tilde{q}) - diag[K_p(\tilde{q})] + diag(k_{p11}^m, \dots, k_{pnn}^m) \tag{3.56}$$

and

$$K_p^M = K_p(\tilde{q}) - diag[K_p(\tilde{q})] + diag(k_{p11}^M, \dots, k_{pnn}^M) \tag{3.57}$$

When $k_{pii}^M > k_{pii}^m > 0, i = 1, \dots, n$, it is always possible to construct K_p^m and K_p^M as defined in (3.56) and (3.57). For instance, K_p^m and K_p^M can be defined as diagonally dominant symmetric positive definite matrices that meet the

following conditions:

$$k_{pii}^{M} > k_{pii}^{m} > \sum_{j=1, j\neq i}^{n} |k_{pij}$$

with the potential for non-zero off-diagonal elements.

The stability analysis of nonlinear controllers relies on the following properties of the aforementioned gain matrices:

1. Given that the proportional gain matrix $K_p(\tilde{q})$ and the constant matrices K_p^m and K_p^M are symmetric and positive definite, there exist positive constants k_{p1} and k_{p2} such that $\forall \tilde{q}, x \in R^n$:

   $$k_{p1}\|x\|^2 \leq x^T K_p^m x \leq x^T K_p(\tilde{q}) x \leq x^T K_p^M x \leq k_{p2}\|x\|^2$$

   where k_{p1} is equal to the smallest eigenvalue of K_p^m and k_{p2} is equal to the largest eigenvalue of K_p^M.

2. As the derivative gain matrix $K_d(\tilde{q}, \dot{\tilde{q}})$ is symmetric and positive definite, there exist positive constants k_{d1} and k_{d2} such that $\forall \tilde{q}, \dot{\tilde{q}}, x \in R^n$,

   $$k_{d1}\|x\|^2 \leq x^T K_d(\tilde{q}, \dot{\tilde{q}}) x \leq k_{d2}\|x\|^2$$

   where k_{d1} is equal to ∞ of the smallest eigenvalue of $K_d(\tilde{q}, \dot{\tilde{q}})$,k_{d2} is equal ti the sup of the largest eigenvalue of $K_d(\tilde{q}, \dot{\tilde{q}})$.

Regarding the stability of the closed-loop system when employing the controller (3.53) for trajectory tracking in the context of the dynamics . The closed-loop system is given as :

$$B(q)\ddot{\tilde{q}} + C(q, \dot{q})\dot{\tilde{q}} + K_p(\tilde{q})\tilde{q} + [K_d(\tilde{q}, \dot{\tilde{q}}) + F_d]\dot{\tilde{q}} = \Delta u \tag{3.58}$$

(3.58) is considered a non-autonomous differential equation because q_d and $\dot{q}_d$ are time-varying trajectories.

$$\Delta u = [B(q) - B_n(q)]\ddot{q}_d + [C(q, \dot{q}) - C_n(q, \dot{q})]\dot{q}_d + [g(q) - g_n(q)] + F_d\dot{q}_d + f_s(\dot{q}) + u_d \tag{3.59}$$

Prior to presenting the stability results, we introduce the following lemmas.

Lemma 1: Considering a dynamic system

$$\dot{x}_i = f_i(x_1, \ldots, x_m, t) \tag{3.60}$$

where $x_i \in R^{n_i}, fori + 1, \ldots, m$ and $t \geq 0$. Letting f_i be locally Lipschitz with respect to $x_i, \ldots, x_m$ uniformly in t on bounded intervals and continuous in t for $t \geq 0$. Now suppose a scalar function $V(x,t) : R^N \times R_+ \rightarrow R_+$ that is :

$$V(x,t) \geq c_i \|x_i\|^2 \tag{3.61}$$

in which we have $x^T = [x_1^T, \ldots, x_m^T], N = n_1 + \ldots + n_m, c_i > 0$ for $i = 1, \ldots, m$, and along the solution trajectories of (3.60)

$$\dot{V}(x,t) \leq -\sum_{i \in I_1} \left(\gamma_i - \sum_{j \in I_{2i}} \gamma_{ij} \|x_j\|^{r_{ij}} \right) \|x_i\|^2 + \epsilon$$

where $\gamma_i, \gamma_{ij}, r_{ij} > 0, \epsilon \geq 0$ and $I_{2i} \subseteq I_1 \subseteq \{1, \ldots, m\}$.

If $\forall i \in I_i$ (with reference to (3.61))

$$\gamma_i > \sum_{j \in I_{2i}} \gamma_{ij} \left(\frac{V_0}{c_j} \right)^{r_{ij}/2} \tag{3.62}$$

where $V_0 = V(x_1(0), \ldots, x_m(0), 0)$ then

$$\forall \beta_i \in \left(0, \gamma_i - \sum_{j \in I_{2i}} \gamma_{ij} \left(\frac{V_0}{c_j} \right)^{r_{ij}/2} \right)$$

The subsequent inequality is valid:

$$\dot{V}(x,t) \leq -\sum_{i \in I_1} \beta_i \|x_i\|^2 + \epsilon \tag{3.63}$$

for $\|x\| > R$ where $R = \sqrt{\epsilon/(min\beta_i)}$ The proof is based on Lemma 2.1 from [44]. **Definition 1:** Uniform Ultimate Boundedness (U.U.B.) [17]: A solution $x(t) : [t_0, \infty)^n$of (3.60) with the initial condition $x(t_0) = x_0$ is

considered uniformly ultimately bounded if there exist positive constants b and c. For every (0, c), there exists a positive constant b and c, and for every $\sigma \in (0, c)$ there is a positive constant $T(\sigma)$ such that $\|x_0\| < \sigma$ implies that $\|x(t)\| \leq b$ for all $t \geq t_0 + T(\sigma)$. The constant b is referred to as the ultimate bound. Uniform ultimate boundedness means that the solution trajectory of the system (3.60), starting at x_0 at time t_0, will ultimately enter and remain within the closed ball $B(b)$. If $B(b)$ is a small region around the equilibrium, then U.U.B. represents a practical notion of stability, often referred to as practical stability. The next lemma provides conditions that ensure U.U.B, and global exponential convergence (to a closed ball) of the solution trajectories of (3.60) [5].

Lemma 2: If exists a continuously differentiable scalar function $V(x, t)$: $R^N \times R_+ \rightarrow R_+$ that has the following properties :

- There are positive constants $\underline{c}$ and $\overline{c}$ such that $\forall x \in R^N$ and $t \in R_+$,
$$\underline{c}\|x\|^2 \leq V(x, t) \leq \overline{c}\|x\|^2$$
- There are constants $\mu > 0$ and $\epsilon \geq 0$ such that along the solution trajectories of (3.60)
$$\dot{V}(x, t) \leq -\mu V(x, t) + \epsilon \tag{3.64}$$

for all x s.t $\mu V(x, t) > \epsilon$ and $t \in R_+$. Then the solution trajectories of (3.60) are uniformly ultimately bounded and globally exponentially convergent to the closed ball $B(r)$, where $r = \sqrt{\epsilon/(\mu c)}$. If, in addition,$\epsilon = 0$, then the system (3.60) is globally exponentially stable about its origin [5].

Theorem 1: Consider the robotic model described by the dynamics along with the nonlinear gain $PD+$ controller from equation (3.53). Assume that the derivative gain matrix $K_d(\tilde{q}, \dot{\tilde{q}})$ is symmetric and positive definite for all $\tilde{q}, \dot{\tilde{q}} \in R^n$ in the real numbers with $k_{d1} > \delta_C$. Additionally, consider the symmetric proportional gain matrix$K_p(\tilde{q})$ with the structure defined in equation (3.54) and diagonal elements satisfying equation (3.55). If the symmetric matricesK_p^m and K_p^M as defined in equations (3.56) and (3.57) are positive definite, then the solution $\overline{q}$ of the closed-loop system described in equation (3.58) will exhibit both uniform ultimate boundedness and exponential convergence towards the closed ball $B(r)$, as defined

below.[31] ***Proof*** Considering the following scalar function:

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2}\dot{\tilde{q}}^T B(q)\dot{\tilde{q}} + \int_0^{\tilde{q}} z^T K_p(z)dz + \alpha \tilde{q}^T B(q)\dot{\tilde{q}} \tag{3.65}$$

We can rewrite the integral term as :

$$\sum_{i=1}^{n}\left(\int_0^{\tilde{q}_i} z_i k_{pii}(z_i)dz_i\right) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} k_{pij}\tilde{q}_i\tilde{q}_j$$

and α in (3.65 is a sufficiently small positive constant :

$$min\left\{\frac{k_{p1}}{m_2}, \frac{m_1}{m_2}, \frac{2(k_{d1}+k_{fd1}-\delta_C)}{3\omega k_C + 2m_2 + \rho k^*}\right\} > \alpha > 0 \tag{3.66}$$

where $k^* = k_{d2} + k_C k_{qd1} + k_{fd2} + \delta_C$ and ρ is defined as:

$$\rho > \frac{k_{d2} + k_C k_{qd1} + k_{fd2} + \delta_C}{2k_{p1}} > 0 \tag{3.67}$$

we also have the constant $\omega > 0$ whose value will be defined later. The first term of $V(\tilde{q}, \dot{\tilde{q}})$ is a positive definite function with respect to $\dot{\tilde{q}}$ due to the positive definiteness of $B(q)$. To demonstrate that the integral term in equation (3.65) is also a positive definite function, it can be expressed as follows:

$$\begin{aligned}\int_0^{\tilde{q}} z^T K_p(z)dz &= \sum_{i=1}^{n}\left(\int_0^{\tilde{q}_i} z_i k_{pii}(z_i)dz_i\right) + \frac{1}{2}\tilde{q}^T K_p^m \tilde{q} - \frac{1}{2}\sum_{i=1}^{n} k_{pii}^m \tilde{q}_i^2 \\ &= \sum_{i=1}^{n}\left(\int_0^{\tilde{q}_i} z_i k_{pii}(z_i)dz_i - \frac{1}{2}k_{pii}^m\tilde{q}_i^2\right) + \frac{1}{2}\tilde{q}^T K_p^m \tilde{q}\end{aligned} \tag{3.68}$$

referring to (3.55) we have for $i = 1, \ldots, n$

$$\int_0^{\tilde{q}_i} z_i k_{pii}(z_i)dz_i \geq \int_0^{\tilde{q}_i} z_i k_{pii}^m dz_i = \frac{1}{2}k_{pii}^m \tilde{q}_i^2 \tag{3.69}$$

by using (3.68) and the first property of the gain matrix:

$$\int_0^{\tilde{q}} z^T K_p(z) dz \geq \frac{1}{2}\tilde{q}^T K_p^m \tilde{q} \geq \frac{1}{2} k_{p1} \|\tilde{q}\|^2 \tag{3.70}$$

so we get the lower-bound of (3.65) as:

$$V(\tilde{q}, \dot{\tilde{q}}) \geq \frac{1}{2} m_1 \|\dot{\tilde{q}}\|^2 + \frac{1}{2} k_{p1} \|\tilde{q}\|^2 + \alpha \tilde{q}^T B(q) \dot{\tilde{q}}$$

The cross term in (3.65) have the upper-bound:

$$|\alpha \tilde{q}^T B(q) \dot{\tilde{q}}| \leq \alpha m_2 \|\tilde{q}\| \|\dot{\tilde{q}}\| \leq \frac{1}{2} \alpha m_2 (\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2) \tag{3.71}$$

so we get that $\alpha \tilde{q}^T B(q) \dot{\tilde{q}} \geq -\frac{1}{2} \alpha m_2 (\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2)$

So our $V(\tilde{q}, \dot{\tilde{q}})$ can be lower bounded as:

$$V(\tilde{q}, \dot{\tilde{q}}) \geq \frac{1}{2} m_1 \|\dot{\tilde{q}}\|^2 + \frac{1}{2} k_{p1} \|\tilde{q}\|^2 - \frac{1}{2} \alpha m_2 (\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2) \geq c_1 \|\tilde{q}\|^2 + c_2 \|\dot{\tilde{q}}\|^2 \tag{3.72}$$

where

$$c_1 = \frac{1}{2}(k_{p1} - \alpha m_2) \quad \text{and} \quad c_2 = \frac{1}{2}(m_1 - \alpha m_2) \tag{3.73}$$

Given that α satisfies (3.66), which implies $min\{c_1, c_2\} > 0$, we have ensured that $V(\tilde{q}, \dot{\tilde{q}})$ in (3.65) is globally positive definite and radially unbounded. Additionally, it is zero at the equilibrium point($\tilde{q} = 0, \dot{\tilde{q}} = 0$). Therefore, the scalar function $V(\tilde{q}, \dot{\tilde{q}})$ in (3.65) serves as a Lyapunov function candidate.

To demonstrate the decrement property of the scalar function (3.65),

we can rewrite the integral term as follows:

$$\begin{aligned}\int_0^{\tilde{q}} z^T K_p(z)dz &= \sum_{i=1}^{n}\left(\int_0^{\tilde{q}_i} z_i k_{pii}(z_i)dz_i\right) \\ &+ \frac{1}{2}\tilde{q}^T K_p^M \tilde{q} - \frac{1}{2}\sum_{i=1}^{n} k_{pii}^M \tilde{q}_i^2 \\ &= \sum_{i=1}^{n}\left(\int_0^{\tilde{q}_i} z_i k_{pii}(z_i)dz_i - \frac{1}{2}k_{pii}^M \tilde{q}_i^2\right) + \frac{1}{2}\tilde{q}^T K_p^M \tilde{q}\end{aligned} \tag{3.74}$$

from (3.55)we have for $i = 1, \ldots, n$

$$\int_0^{\tilde{q}_i} z_i k_{pii(z_i)dz_i} \leq \int_0^{\tilde{q}_i} z_i k_{pii}^M dz_i = \frac{1}{2}k_{pii}^M \tilde{q}_i^2 \tag{3.75}$$

Which yields using (3.75) and the first property of the gain matrices to:

$$\int_0^{\tilde{q}} z^T K_p(z)dz \leq \frac{1}{2}\tilde{q}^T K_p^M \tilde{q} \leq \frac{1}{2}k_{p2}\|\tilde{q}\|^2 \tag{3.76}$$

so we can introduce an upper-bound on the Lyaponov candidate as

$$V(\tilde{q}, \dot{\tilde{q}}) \leq \frac{1}{2}m_2\|\dot{\tilde{q}}\|^2 + \frac{1}{2}\alpha m_2(\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2) + \frac{1}{2}k_{p2}\|\tilde{q}\|^2 \leq c_3\|\tilde{q}\|^2 + c_4\|\dot{\tilde{q}}\|^2 \tag{3.77}$$

in (3.77) we have that

$$c_3 = \frac{1}{2}(\alpha m_2 + k_{p2}) \quad \text{and} \quad c_4 = \frac{1}{2}(\alpha + 1)m_2 \tag{3.78}$$

So, the proposed Lyapunov function$V(\tilde{q}, \dot{\tilde{q}})$ in equation (3.65) is a function that possesses global positive definiteness, radial unboundedness, and decreasing properties while satisfying the following inequalities:

$$\underline{c}(\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2) \leq V(\tilde{q}, \dot{\tilde{q}}) \leq \overline{c}(\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2) \tag{3.79}$$

where $\underline{c} = min\{c_1, c_2\} > 0$ and $\overline{c} = max\{c_3, c_4\} > 0$

The time derivative of $V(\tilde{q}, \dot{\tilde{q}})$ along the solution trajectories of equation

(3.58) is expressed as:

$$\begin{aligned}\dot{V}(\tilde{q},\dot{\tilde{q}}) = &-\dot{\tilde{q}}^T[K_d(\tilde{q},\dot{\tilde{q}}) + F_d]\dot{\tilde{q}} + \alpha\dot{\tilde{q}}^T B(q)\dot{\tilde{q}} \\ &+ \alpha\tilde{q}^T C(q,\dot{q})^T\dot{\tilde{q}} - \alpha\tilde{q}^T K_p(\tilde{q})\tilde{q} \\ &- \alpha\tilde{q}^T[K_d(\tilde{q},\dot{\tilde{q}}) + F_d]\dot{\tilde{q}} + (\dot{\tilde{q}}^T + \alpha\tilde{q}^T)\Delta u\end{aligned} \tag{3.80}$$

In order to introduce an upper bound we start from :

$$\begin{aligned}\alpha\tilde{q}^T C(q,\dot{q})^T\dot{\tilde{q}} =&\alpha\tilde{q}^T C(q,\dot{q}_d - \dot{\tilde{q}})^T\dot{\tilde{q}} \\ &\leq \alpha k_C\|\dot{q}_d - \dot{\tilde{q}}\|\|\tilde{q}\|\|\dot{\tilde{q}}\| \\ &\leq \alpha k_C k_{qd1}\|\tilde{q}\|\|\dot{\tilde{q}}\| + \alpha k_C\|\tilde{q}\|\|\dot{\tilde{q}}\|^2\end{aligned} \tag{3.81}$$

Since $B(q)$, $K_P(\tilde{q})$ and $K_d(\tilde{q},\dot{\tilde{q}})$ are positive definite matrices we can write :

$$\alpha\dot{\tilde{q}}^T B(q)\dot{\tilde{q}} \leq \alpha m_2\|\dot{\tilde{q}}\|^2 \tag{3.82}$$

$$-\dot{\tilde{q}}^T[K_d(\tilde{q},\dot{\tilde{q}}) + F_d]\dot{\tilde{q}} \leq -(k_{d1} + k_{fd1})\|\dot{\tilde{q}}\|^2 \tag{3.83}$$

$$-\alpha\tilde{q}^T K_p(\tilde{q})\tilde{q} \leq -\alpha k_{p1}\|\tilde{q}\|^2 \tag{3.84}$$

$$|-\alpha\tilde{q}^T[K_d(\tilde{q},\dot{\tilde{q}}) + F_d]\dot{\tilde{q}}_d| \leq \alpha(k_{d2} + k_{fd2})\|\tilde{q}\|\|\dot{\tilde{q}}\| \tag{3.85}$$

Considering the bounds provided in equations (3.42) through (3.44) and (3.47) through (3.49), we can deduce the following:

$$\|[C(q,\dot{q}) - C_n(q,\dot{q})]\dot{q}_d\| = \|[C(q,\dot{q}_d) - C_n(q,\dot{q}_d)]\dot{q}\| \leq \delta_C\|\dot{q}_d - \dot{\tilde{q}}\| \tag{3.86}$$

$$\begin{aligned}\|\Delta u\| \leq&\delta_M k_{qd2} + \delta_C(k_{qd1} + \|\dot{\tilde{q}}\|) + \delta_g \\ &k_{fd2}k_{qd1} + k_{fs} + k_{ud} \\ &= \eta_1 + \eta_2\|\dot{\tilde{q}}\|\end{aligned} \tag{3.87}$$

where $\eta_1 = \delta_M k_{qd2} + (\delta_C + k_{fd2})k_{qd1} + \delta_g + k_{fs} + k_{ud}$ and $\eta_2 = \delta_C$ and

$$\begin{aligned}(\dot{\tilde{q}}^T + \alpha\tilde{q}^T)\Delta u \leq & \|\dot{\tilde{q}}\|\|\Delta u\| + \alpha\|\tilde{q}\|\|\Delta u\| \\ & \leq \alpha\eta_1\|\tilde{q}\| + \eta_1\|\dot{\tilde{q}}\| \\ & + \alpha\eta_2\|\tilde{q}\|\|\dot{\tilde{q}}\| + \eta_2\|\dot{\tilde{q}}\|^2\end{aligned} \tag{3.88}$$

Following the inequalities (3.81) through (3.85) and (3.88), we can now express the time derivative of $V(\tilde{q}, \dot{\tilde{q}})$ as:

$$\begin{aligned}\dot{V}(\tilde{q}, \dot{\tilde{q}}) \leq & -(k_{d1} + k_{fd1} - \alpha m_2 - \eta_2)\|\dot{\tilde{q}}\|^2 - \alpha k_{p1}\|\tilde{q}\|^2 \\ & + \alpha(k_{d2} + k_{fd2} + k_C k_{qd1} + \eta_2)\|\tilde{q}\|\|\dot{\tilde{q}}\| \\ & + \eta_1\|\dot{\tilde{q}}\| + \alpha\eta_1\|\tilde{q}\| + \alpha k_C\|\tilde{q}\|\|\dot{\tilde{q}}\|^2\end{aligned} \tag{3.89}$$

if we make $a = \alpha m_2 + \eta_2, b = k_{d2} + k_C k_{qd1} + k_{fd2} + \eta_2$,$v_1 = k_{d1} + k_{fd1} - a - \frac{1}{2}\alpha\rho b, v_2 = k_{p1} - \frac{b}{2\rho}$, where ρ is a positive constant that was defined in (3.67) we can rewrite(3.89) as:

$$\begin{aligned}\dot{V}(\tilde{q}, \dot{\tilde{q}}) \leq & -(k_{d1} + k_{fd1} - a)\|\dot{\tilde{q}}\|^2 - \alpha k_{p1}\|\tilde{q}\|^2 \\ & + \alpha b\|\tilde{q}\|\|\dot{\tilde{q}}\| + \eta_1\|\dot{\tilde{q}}\| + \alpha\eta_1\|\tilde{q}\| + \alpha k_C\|\tilde{q}\|\|\dot{\tilde{q}}\|^2 \\ & \leq -(k_{d1} + k_{fd1} - a)\|\dot{\tilde{q}}\|^2 - \alpha k_{p1}\|\tilde{q}\|^2 + \frac{ab}{2}\left(\frac{\|\tilde{q}\|^2}{\rho} + \rho\|\dot{\tilde{q}}\|^2\right) \\ & + \eta_1\|\dot{\tilde{q}}\| + \alpha\eta_1\|\tilde{q}\| + \alpha k_C\|\tilde{q}\|\|\dot{\tilde{q}}\|^2 \\ & \leq -v_1\|\dot{\tilde{q}}\|^2 - \alpha v_2\|\tilde{q}\|^2 + \eta_1\|\dot{\tilde{q}}\| + \alpha\eta_1\|\tilde{q}\| + \alpha k_C\|\tilde{q}\|\|\dot{\tilde{q}}\|^2\end{aligned}$$

Given that the conditions in (3.66) and (3.67) imply that v_1 and v_2 are both greater than 0, we can establish the following inequalities through the process of completing the square:

$$\eta_1\|\dot{\tilde{q}}\| \leq \left(\frac{\eta_1}{\sqrt{v_1}}\right)^2 + \left(\frac{\sqrt{v_1}}{2}\right)^2\|\dot{\tilde{q}}\|^2 \tag{3.90}$$

$$\eta_1\|\tilde{q}\| \leq \left(\frac{\eta_1}{\sqrt{v_2}}\right)^2 + \left(\frac{\sqrt{v_2}}{2}\right)^2\|\tilde{q}\|^2 \tag{3.91}$$

Now we can write the upper-bound of $\dot{V}(\tilde{q}, \dot{\tilde{q}})$ as:

$$\begin{aligned}\dot{V}(\tilde{q}, \dot{\tilde{q}}) \leq & -\frac{3}{4}v_1\|\dot{\tilde{q}}\|^2 - \frac{3}{4}\alpha v_2\|\tilde{q}\|^2 + \alpha k_C\|\tilde{q}\|\|\dot{\tilde{q}}\|^2 + \frac{\eta_1^2}{v_1} + \frac{\alpha\eta_1^2}{v_2} \\ & = \gamma_1\|\tilde{q}\|^2 - \gamma_2\|\dot{\tilde{q}}\|^2 + \gamma_{21}\|\tilde{q}\|\|\dot{\tilde{q}}\|^2 + \epsilon\end{aligned} \tag{3.92}$$

in which

$$\gamma_1 = 3\alpha v_2/4, \quad \gamma_2 = 3v_1/4 \quad \text{and} \quad \gamma_{21} = \alpha k_C \tag{3.93}$$

$$\epsilon = \frac{\eta_1^2}{v_1} + \frac{\alpha\eta_1^2}{v_2} \tag{3.94}$$

Define w in (3.66) as $\omega = \left(\frac{V_0}{c_1}\right)^{1/2} > 0$, where $V_0 = V(\tilde{q}, \dot{\tilde{q}})\big|_{t=0} > 0$ for a positive definite $V(\tilde{q}, \dot{\tilde{q}})$, and $c_1 > 0$ is given by (3.73). Since α satisfies (3.66), we have:

$$\gamma_2 > \gamma_{21}\omega \tag{3.95}$$

and by using Lemma 1 for $\beta_2 \in (0, \gamma_2 - \gamma_{21}\omega)$ with $\|\tilde{q}\| > \sqrt{\epsilon/min\{\gamma_1, \beta_2\}}$ the following inequality holds

$$\begin{aligned}\dot{V}(\tilde{q}, \dot{\tilde{q}}) \leq & -\gamma_1\|\tilde{q}\|^2 - \beta_2\|\dot{\tilde{q}}\|^2 + \epsilon \\ & \leq -min\{\gamma_1, \beta_2\}(\|\tilde{q}\|^2 + \|\dot{\tilde{q}}\|^2) + \epsilon \\ & \leq -\mu V(\tilde{q}, \dot{\tilde{q}}) + \epsilon\end{aligned} \tag{3.96}$$

in which we have $\mu = min\{\gamma_1, \beta_2\}/\overline{c}$. By applying Lemma 2, we can establish that the solution $\overline{q}$ of the closed-loop system (3.58) is both uniformly ultimately bounded and exponentially convergent to the closed ball $B(r)$, where the value of r is determined as $r = \sqrt{\epsilon/(\mu\underline{c})}$, and the constants $\overline{c}$ and $\underline{c}$ are provided by (3.79).

Given that the lower bound of α is zero, equation (3.95) can be fulfilled for any initial conditionsV_0 by selecting a very small α. As a result, the convergence domain covers the entire state space, and the solution $\overline{q}$ of the closed-loop system (3.58) experiences global exponential convergence towards the closed ball $B(r)$, albeit without a uniform rate[31].

3.3.1 SIMULATION

The presented results showcase the convergence behavior and ultimate upper bounds (U.U.B.) of tracking errors during the utilization of the nonlinear PD+ controller for tracking control in a two-link rigid manipulator. Furthermore, comparative results are provided, specifically focusing on tracking error convergence, to assess the performance of the nonlinear PD+ controller in contrast to the previously explained robust controller.

In the simulation, a planar elbow manipulator is employed. It's configured with the first revolute joint anchored to the ground and the second revolute joint connected to link 1. This manipulator bears a payload, specifically a mass of $m_1 = 1$ kg, positioned at the far end of link 2.

For illustrative purposes, the manipulator's dynamics are considered without accounting for friction and disturbance terms. The primary source of uncertainty is attributed to variations in the payload, which can range from 0 to 1 kg.

The prescribed joint trajectories are defined as follows:

For the first joint: $q_{d1} = 0.25\pi + 0.5(1 - \cos(0.5\pi t))$

For the second joint: $q_{d2} = 0.5\pi + 0.25(1 - \cos(\pi t))$

To track these trajectories,the non linear PD+ controller utilizes the feedforward component as described in equation (3.46). However, it's important to note that the nominal dynamic terms, including $B_n(q)$, $C_n(q, \dot{q})$, and $g_n(q)$, were calculated based on a nominal payload of $m_l = 0.5$ kg. This nominal payload value differs from the actual payload of 1 kg that was used for this specific simulation, and the feedback component of this controller :

$$u_{fb} = K_p(\tilde{q})\tilde{q} + K_d(\tilde{q}, \dot{\tilde{q}})\dot{\tilde{q}} \tag{3.97}$$

As previously mentioned, the key requirement for guaranteeing the stability of the controller discussed above is the selection of positive definite proportional and derivative gain matrices. Specifically, these matrices should be chosen in a way that $k_{d1} > \delta_c$. With the numerical values provided for the manipulator parameters, The positive constant k_C, which is calculated based on the difference between the Coriolis and centrifugal

terms of the actual system $C(q, q_d)$ and the nominal system$C_n(q, q_d)$, is determined to be $k_C = 2$.

Hence, we derive the bound $\delta_C = 2.2$ .The performance of the feedback controllers depends on the specific nonlinear gain functions and the selected parameter values within the gain matrices.

We utilized the following diagonal gain matrices for the PD+ non linear controller :

$$K_p(\tilde{q}) = \begin{bmatrix} \frac{194.2}{0.2+|\tilde{q}_1|} & 0 \\ 0 & \frac{74.3}{0.2+|\tilde{q}_2|} \end{bmatrix} \tag{3.98}$$

$$K_d(\tilde{q}, \dot{\tilde{q}}) = \begin{bmatrix} \frac{74.4}{(0.5+|\tilde{q}_1|)(1+0.07|\dot{\tilde{q}}_1|)} & 0 \\ 0 & \frac{7.6}{(0.1+|\tilde{q}_2|)(1+0.07|\dot{\tilde{q}}_2|)} \end{bmatrix} \tag{3.99}$$

while for the robust controller we set the values :

$$K_p = \begin{bmatrix} 49 & 0 \\ 0 & 68 \end{bmatrix}$$

$$K_d = \begin{bmatrix} 15 & 0 \\ 0 & 28 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and we get :

$$Q = \begin{bmatrix} 1.8197 & 0 & 0.0102 & 0 \\ 0 & 1.4380 & 0 & 0.0074 \\ 0.0102 & 0 & 0.0340 & 0 \\ 0 & 0.0074 & 0 & 0.0181 \end{bmatrix}$$

with

$$K = 1000 \quad \text{and} \quad \epsilon = 0.001$$

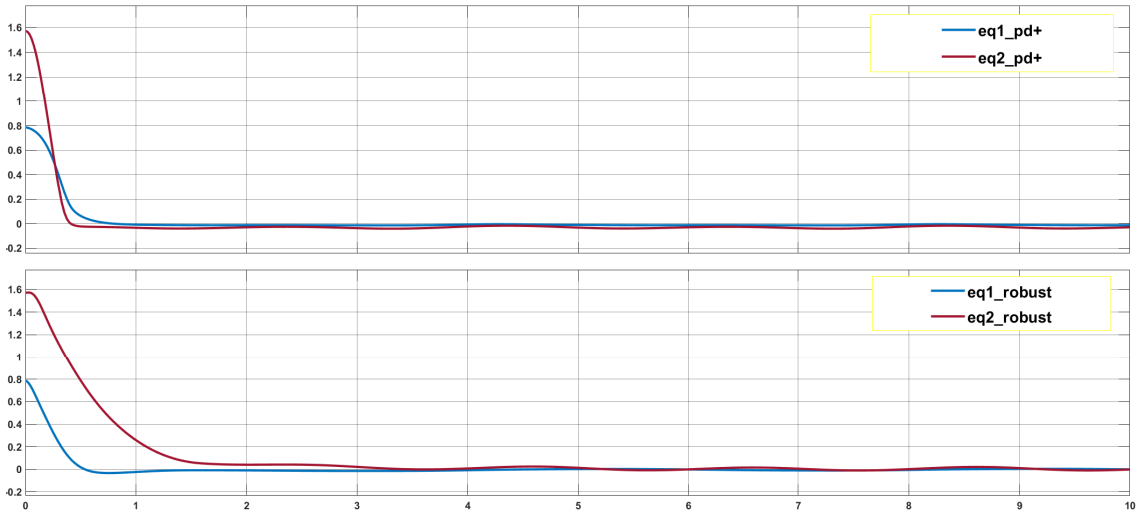

Figure 3.18: (Error convergence)

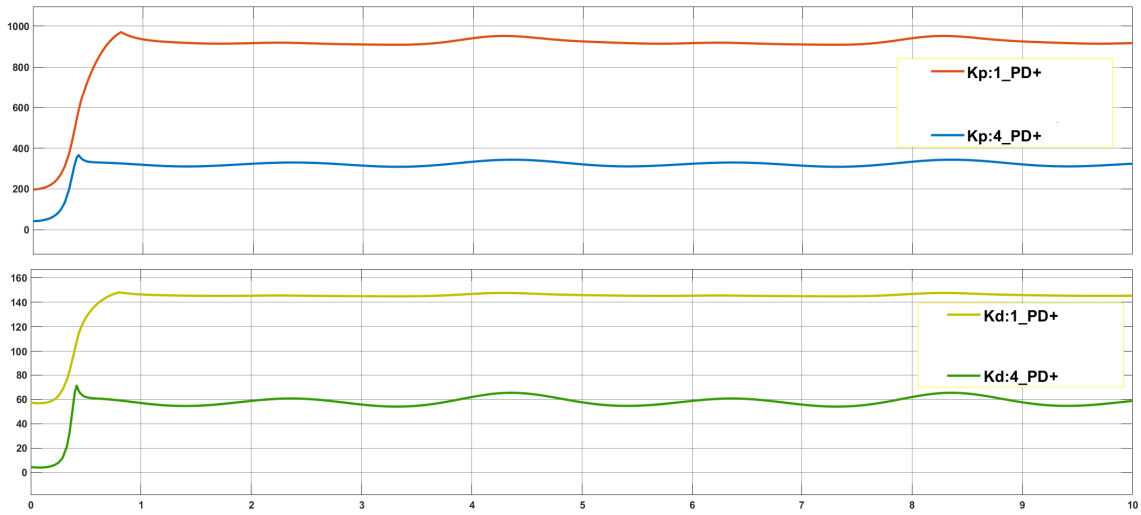

Figure 3.19: (Kp,Kd of the PD+ controller)

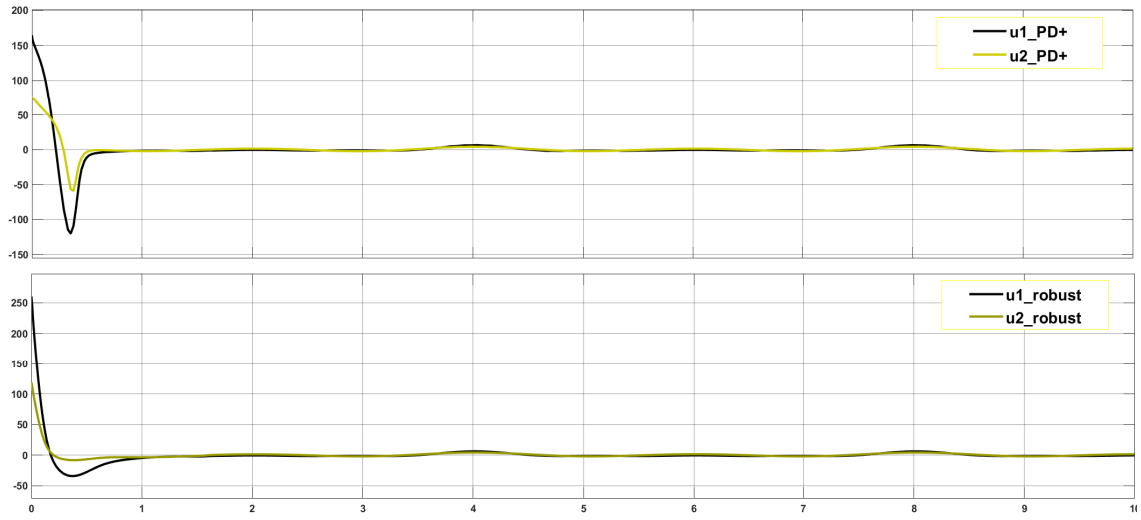

Figure 3.20: (Controllers outputs)

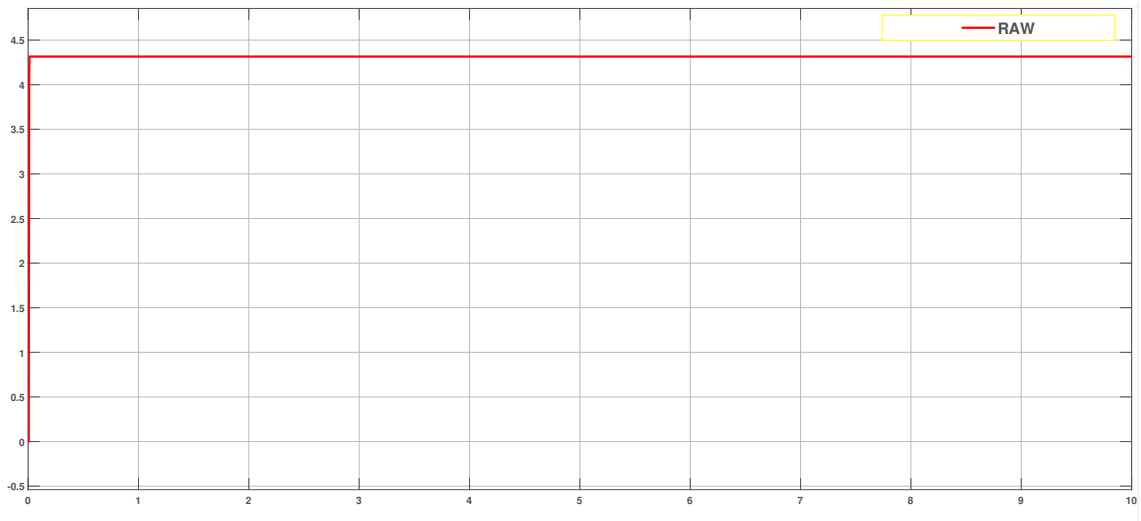

Figure 3.21: (Raw)

As evident from figure (3.18), it's apparent that the PD+ controller exhibits a notably faster convergence rate. This characteristic is a result of its primary objective, which is to rapidly escalate the gain values to very high levels, as illustrated in figure (3.19).

Conversely, our robust controller operates effectively with substantially lower gain values, resulting in a smoother controller output, as depicted in figure (3.20),keeping in mind that faster convergnce can be achieved by higher values for K_p, K_d, Q, and K.

It's important to note that achieving faster convergence with the PD+ controller necessitates prior knowledge regarding the extent of uncertainties, which is vital for computing its gain matrices. In contrast, our robust controller demands no prior knowledge, as it dynamically computes the required value of ρ online, as demonstrated in figure (3.21).

Chapter 4

Extending Controller Efficacy: Complex Robot Validation

In the previous chapters, our efforts were primarily focused on the development, analysis, and simulation of our control strategy. We established a solid groundwork using a simple Two-link Planar Arm as our testing ground.

However, as we progress into this chapter, we embark on a transformative journey to assess the real-world applicability of our controller on more intricate robotic platforms. Introducing SCARA robot into our research endeavors represents a substantial leap in complexity and challenges.

This chapter signifies a pivotal phase in our research, where we shift our emphasis towards the practical validation of our control strategies. Our objective is to evaluate the true effectiveness and adaptability of our control methodologies in real-world scenarios.

The SCARA robot, with it's multiple degrees of freedom, intricate dynamics, and diverse operational demands, present a profound testing ground. Our assessment on this platform goes beyond a mere formality; it is a crucial step towards real-world implementation.

Through a meticulously planned series of experiments and comprehensive analyses, our aim is to showcase the capability of our controller in navigating the intricacies of this multi-dimensional robot. This journey

begins with a detailed introduction to the robot, highlighting it's unique attributes and functionalities.

We delve into the adaptation and implementation of our control algorithms, tailoring them to meet the specific requirements and dynamics of this complex robotic system. Our evaluation encompasses various facets, including trajectory tracking precision, system stability, and adaptability to dynamic uncertainties.

By validating our control strategy across such diverse platforms, we take a significant stride towards its practical deployment in complex real-world scenarios. This chapter serves as the bridge between theory and application, setting the stage for the eventual integration of our control algorithms into a wide array of intricate robotic systems.

4.1 SCARA Robot

At this juncture in our research journey, it becomes highly pertinent to put our controller to the test on a robotic platform that closely mirrors real-world applications. To accomplish this goal, we set our sights on the SCARA robot, a widely employed and versatile robot in various industrial domains.

What sets the SCARA robot apart from our previous testing platform is its unique configuration. Beyond the complexities of operating in a three-dimensional space, the SCARA robot introduces an additional layer of intricacy with the inclusion of a prismatic joint.

The addition of a prismatic joint significantly expands the robot's operational capabilities, enabling it to perform tasks and maneuvers that are not achievable by traditional rotational joints alone.

By applying our controller to the SCARA robot, we aim to evaluate its adaptability to a wider spectrum of robotic systems and operational scenarios. This step represents a crucial move towards ensuring that our control strategy is not only effective in theory but also capable of thriving in the practical, real-life environments where robots like the SCARA are extensively utilized.

The SCARA, which stands for "Selective Compliance Assembly Robot Arm", see Figure(4.1) was firstly designed in 1979 by Hiroshi Makino, a professor of Yamanashi University (Japan)[27][24]. Since that time, it became one of the most used robotic arms in the world. It has gained widespread acclaim and utilization owing to its multitude of advantageous features. Its success can be attributed to several key factors, which include its remarkable precision, compact dimensions, straightforward structure, minimal backlash between components, and ease of assembly.

Figure 4.1: Example of SCARA Robot

This robotic arm's versatility shines in the realm of factory assembly lines, primarily due to its exceptional capability to execute movements not only along a horizontal plane but also seamlessly transitioning to the vertical dimension. This distinctive attribute empowers the SCARA robot to perform a wide range of tasks within the workspace.

Typical industrial applications where the SCARA robot excels include, but are not limited to, pick and place operations, palletizing tasks, intricate assembling processes, and efficient packaging activities. Its adaptability and efficiency make it a valuable asset in various manufacturing scenarios, enhancing productivity and precision[9].

While various companies produce SCARA robots with unique features and capabilities tailored to specific applications, the fundamental structural design remains relatively consistent across the board.

The core structure typically consists of two primary links and a total of four axes, providing the robot with four degrees of freedom. Among these axes, two are parallel rotary joints that enable rotational movement in the horizontal plane. Additionally, a linear vertical joint provides freedom of movement along the X-Y-Z coordinate space. The fourth degree of freedom is facilitated by the rotational motion of the end effector along the vertical axis as seen in the Figure(4.2)

This standardized configuration offers a versatile platform that can be adapted and customized to meet the diverse needs of different industries and applications, making SCARA robots a valuable asset in the world of automation and robotics.[10].

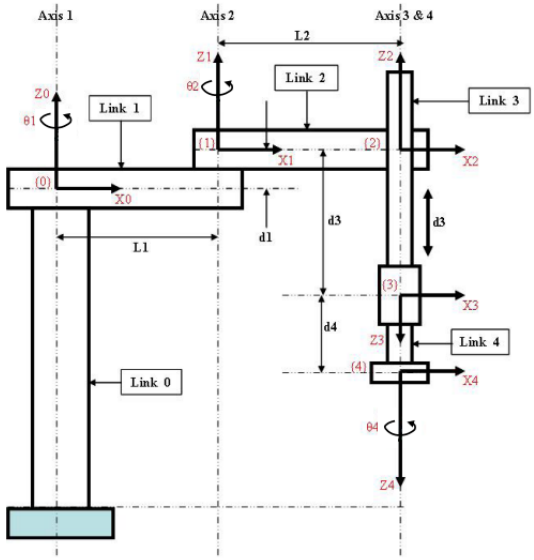

Figure 4.2: Schematic Diagram of the SCARA Robot[28]

4.2 Simulation

We have conducted experiments to validate the performance of our controller on a SCARA Manipulator. These experiments specifically aim to assess the controller's robustness in the face of dynamic model uncertainties. Additionally, we have investigated error convergence using a robust control approach, where the online computation of the ρ value eliminates the need for any prior knowledge of uncertainty characteristics.

In the context of our experimental setup, we harnessed the capabilities of the MATLAB Simulink SimScape[26] library to create two distinct Rigid Body Trees. These trees played a pivotal role in our investigation, each serving a unique purpose.

The first Rigid Body Tree was meticulously configured with what we consider as real parameters in our experiment. Subsequently, we employed the powerful Robotics Toolbox library within Simulink to compute essential matrices, namely B, C, and G, which govern the dynamics of the robotic system.These matrices were calculated using the Robotics Toolbox library and subsequently utilized as inputs for the Robot function in our simulation.

On the other hand, the second Rigid Body Tree took on a different role in our experiment. It was intentionally endowed with intentionally inaccurate parameters, intentionally introduced to emulate uncertainties in the system's dynamic model.This approach allowed us to simulate a scenario where the system's true characteristics were not perfectly known,Just as in the case of the first tree, we leveraged the Robotics Toolbox library within Simulink to compute alternative matrices, denoted as $\hat{B}$, $\hat{C}$, and $\hat{G}$. These matrices were crafted using the intentionally erroneous parameters, mirroring the uncertainties we aimed to address in our study. Notably, the computed $\hat{B}$, $\hat{C}$, and $\hat{g}$ matrices derived from the second Rigid Body Tree served a distinct purpose. They were utilized as inputs to our robust control strategy (the controller function in our simulation). The essence of this approach lay in its ability to adapt to the dynamic uncertainties present in the system, making real-time adjustments to ensure precise trajectory

tracking and system stability. By incorporating these matrices into the controller within Simulink, we could effectively test the performance and robustness of our control strategy under conditions where prior knowledge of uncertainties was not available.

In the process of constructing our first rigid body tree, which represents the robot with real parameters, in Simscape, the following parameters were configured:

- Base radius: 0.04 meters
- Length: 0.05 meters
- Links lengths:
  - a_1: 0.1 meters
  - a_2: 0.2 meters
  - a_3: 0.15 meters
  - a_4: 0.01 meters

For the second rigid body tree, which models the robot with incorrect parameters to simulate uncertainties, the parameters were set as follows:

- Base radius: 0.04 meters
- Length: 0.03 meters
- Links lengths:
  - $\hat{a}_1$: 0.088 meters
  - $\hat{a}_2$: 0.3 meters
  - $\hat{a}_3$: 0.2 meters
  - $\hat{a}_4$: 0.02 meters

By setting the desired trajectory as:

$$q_{d1} = \cos(t)$$

$$q_{d2} = \cos(2t)$$

$$q_{d3} = \cos\left(\frac{5t}{30}\right)$$

$$q_{d4} = \cos(3t)$$

and by setting:

$$Kp = \begin{bmatrix} 49 & 0 & 0 & 0 \\ 0 & 68 & 0 & 0 \\ 0 & 0 & 49 & 0 \\ 0 & 0 & 0 & 68 \end{bmatrix}$$

$$Kd = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

$$P = \text{Identity matrix of size(8)}$$

$$\epsilon = 0.07$$

$$K = 1000$$

4.2.1 Results and comments

The simulation results for the SCARA robot reaffirm the findings presented in the earlier chapters, where we extensively discussed the dynamics and control strategies for the 2R robot,

In Figure (4.3), we observe the convergence of the error towards the subspace defined by $z = D^T Q\xi = 0$. The rate of this convergence is influenced by various factors such as the matrices Q, K_P, K_D, and a constant K which controls the growth rate of the variable ρ over time. Figure (4.4) provides insights into the behavior of the variable ρ.

This convergence of the error towards zero is primarily guided by the control input ω, which, ideally switches at an infinite frequency, a phenomenon known as chattering, as depicted in Figure (4.5).

By employing a robust control law, as described in Equation (3.41), we can observe the corresponding control input in Figure (4.6). Here, the parameter ϵ plays a crucial role, representing a trade-off between achieving a smooth controller output and ensuring that the error remains bounded.

The thickness of the boundary layer within which the error fluctuates depends on the value of ϵ, as illustrated in Figure (4.7).

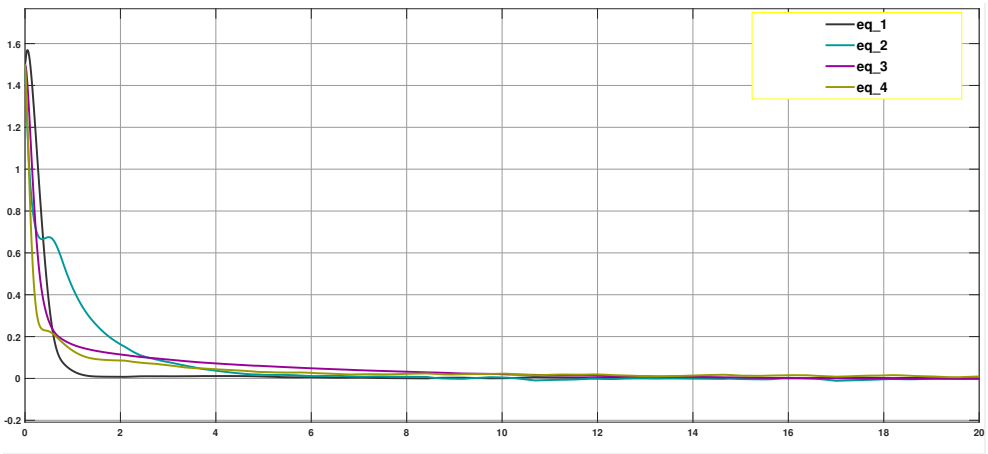

Figure 4.3: (Error convergence with chattering)

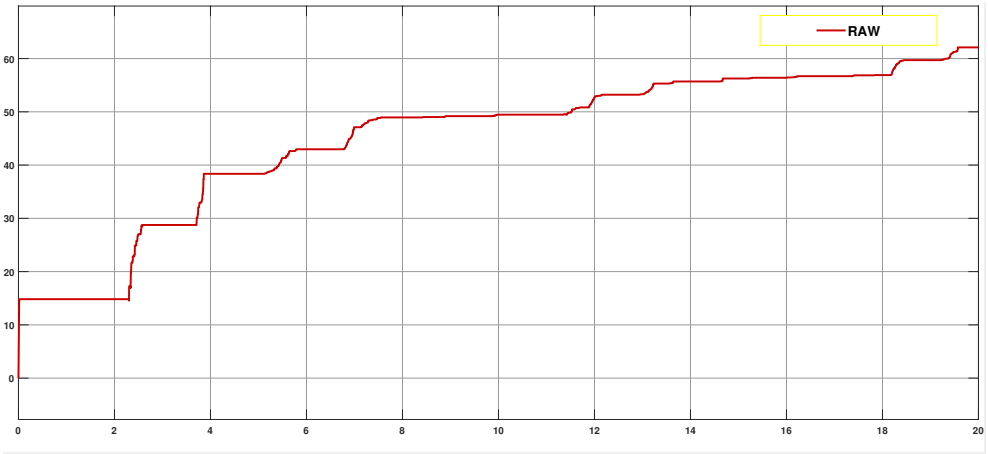

Figure 4.4: (Raw with chattering)

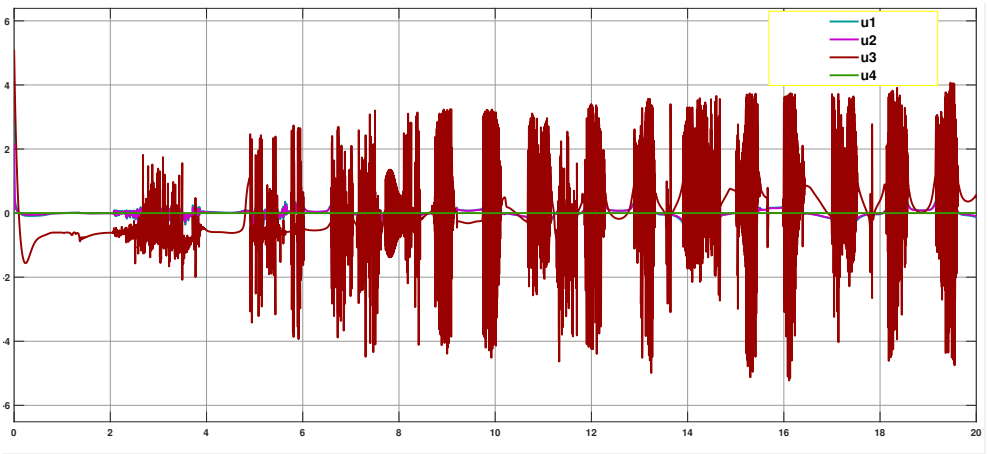

Figure 4.5: (Controller output with chattering)

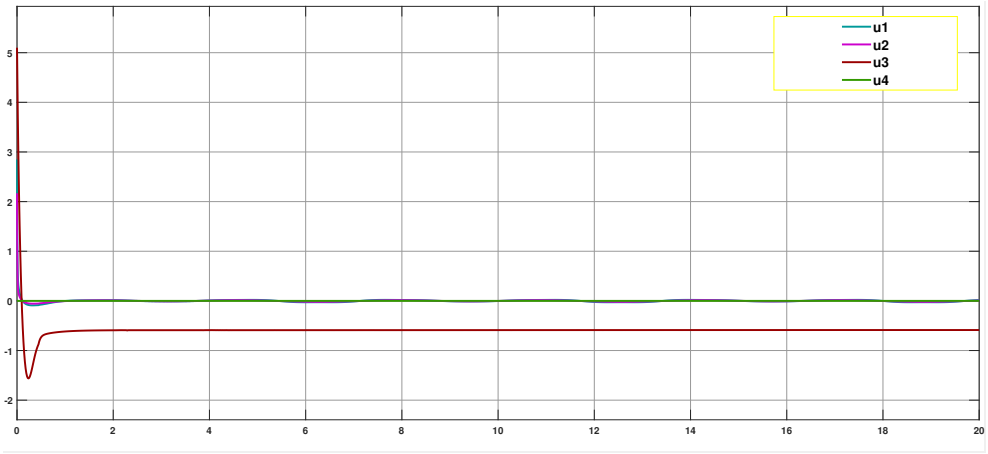

Figure 4.6: (Controller output without chattering)

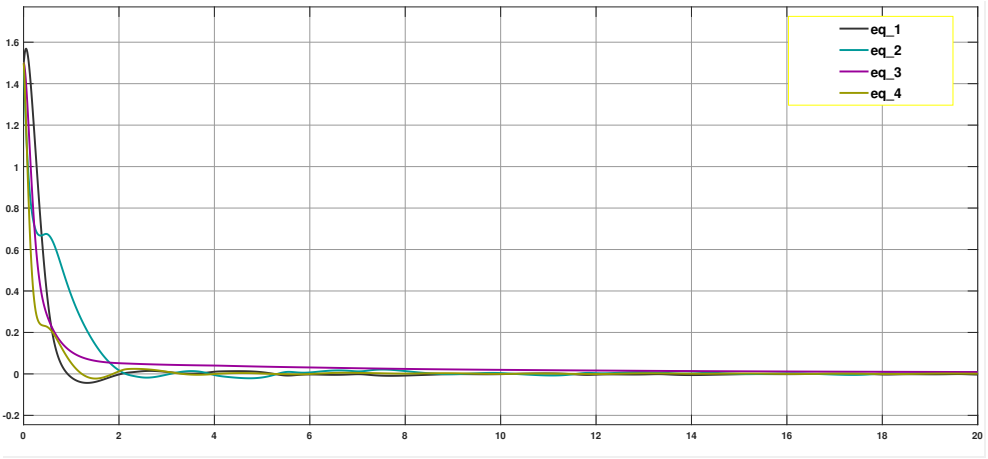

Figure 4.7: (Error convergence without chattering)

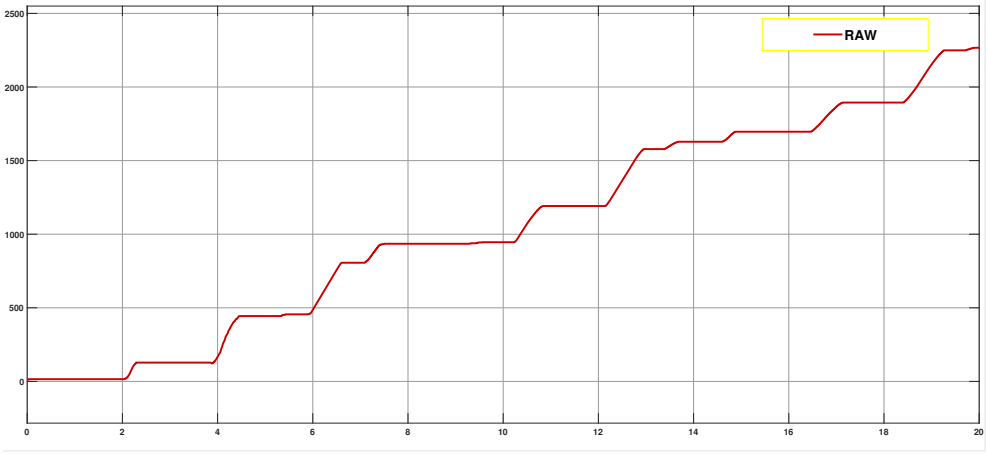

Figure 4.8: (Raw without chattering)

Chapter 5

Conclusion

The control of rigid manipulators with tracking objectives has seen significant advancements through the implementation of robust controllers featuring real-time computation for the robustness term. These controllers also incorporate estimates of dynamic model terms, accounting for both the challenges posed by imperfect model compensation and intentional simplifications made during inverse dynamics computation. As a result of these innovative approaches, it has been demonstrated that all resulting trajectories tend to converge effectively within the desired subspace.

To ensure stability in this context, a novel method has been introduced to calculate the value of the term "" dynamically. This computation occurs in real time, primarily by monitoring the derivative of the Lyapunov function. The primary objective is to adjust the value of "" to ensure that the derivative of the Lyapunov function remains non-positive, thereby guaranteeing stability.

One key component in this control strategy is the term ω defined as $\omega = \frac{\rho}{\|z\|} z$ This term plays a crucial role in countering uncertainties in $\tilde{B}$ and $\tilde{n}$ when computing the nonlinear terms linked to the manipulator's state. Notably, the magnitude of ρ increases in response to higher uncertainty levels, ensuring robustness in the face of dynamic variations.

An additional important aspect of this control approach is the elimination of undesired high-frequency components, often referred to as chat-

tering, which can plague control systems. To address this issue, a robust control law has been adopted, emphasizing that while it may not guarantee error convergence to zero, it does maintain the errors within a specified norm. Consequently, this law results in a control input characterized by a limited magnitude.

What distinguishes this control strategy is its ability to operate at an infinite switching frequency, a feature that compels the error system dynamics to remain confined to the sliding subspace. This control law establishes a hyperplane that no longer attracts the error, allowing it to fluctuate within a boundary layer whose thickness is contingent on the parameter ϵ.

Simulation results have been employed to provide an in-depth illustration of the stability and robustness analysis inherent in this control scheme. These results serve to underscore the promising potential for enhanced controller performance in practical applications.

References

[1]C. Abdallah et al. *Survey of robust control for rigid robots*. IEEE Control Syst. Mag. 11 (2), 1991.

[2]K.J. Astrom and B. Wittenmark. *Adaptive Control*. New York: Addison-Wesley, 1995.

[3]J.K. Åström and B. Wittenmark. *A survey of adaptive control applications*. New Orleans,Louisiana: Proceedings of 34th IEEE Conference on Decision and Control, 1995.

[4]O. Barambones and V. Etxebarria. *Robust adaptive control for robot manipulator with un-modeled dynamics*. Int. J. Cybernet. Syst. 31 (1), 2000.

[5]M. Corless. *Guaranteed rates of exponential convergence for uncertain systems*. J. Opt. Theory and Approx., vol. 64, no.3, 1990, pp. 481–494.

[6]J.J. Craig, P. Hsu, and S.S. Sastry. *Adaptive control of mechanical manipulators*. Int. J. Robotics Res. 6 (2), 1987.

[7]R.A. DeCarlo, S.H. Zak, and G.B. Matthews. *Variable structure control of nonlinear multivariable systems: a tutorial*. Proceedings of IEEE 76,3, 1988, pp. 212–232.

[8]H. Elmali and N. Olgac. *Theory and implementation of sliding mode control with perturbation estimation*. IEEE Int. Conf. Robotics Autom. 3, 1992.

[9]D. Fadanelli, L. Schenato, and R. Kavanagh. *Design of a workcell incorporating a gripper-based SCARA robot*. Padova,Italy: Università degli studi di Padova,University College Cork,Ireland,master thesis, 2016.

[10]L. E. Ferrari, A. Beghi, and R. Kavanagh. *Matlab-based Control of a SCARA Robot*. Padova,Italy: Universit'a degli studi di Padova,master thesis, 2015.

[11]R.N. Gasimov, A. Karamanciolu, and A.Yazici. *A nonlinear programming approach for the sliding mode control design*. Applied Mathematical Modelling 29, 2005, pp. 1135–1148.

[12]J.Q. Gong and B. Yao. *Adaptive robust control without knowing bounds of parameter variations*. Proc. IEEE Conf. Decision Control 4, 1999.

[13]D.T. Greenwood. *Advanced Dynamics*. Cambridge university Press, 2006.

[14]L.R. Hunt, R. Su, and G. Meyer. *Design for multi-input nonlinear systems in Differential Geometric Control Theory*. R.W. Brockett et al., eds. Boston-Basel-Stuttgart: Birkhauser, 1983, pp. 268–298.

[15]R. Kelly and R. Carelli. *A class of nonlinear PD-type controllers for robotic manipulators*. J. Robotic Sys., vol. 13, no. 12, 1996, pp. 793–802.

[16]R. Kelly and R. Salgado. *PD control with computed fcedforcard of robot manipulators: a design procedure*. IEEE Trans. on Robotics and Automation, vol. 10, no. 4, 1994, pp. 566–571.

[17]H. Khalil. *Nonlinear Systems*. Prentice Hall, Englewood Cliffs, 1996.

[18]H.K. Khalil. *CONTROL SYSTEMS, ROBOTICS, AND AUTOMATION, Volume 12.* Encyclopedia of Life Support Systems (EOLSS), n.d.

[19]N. Kim, C.W. Lee, and P.H. Chang. *Sliding mode control with perturbation estimation: application to motion control of parallel manipulators*. Control Eng. Pract. 6 (11), 1998.

[20]Y. Landau. *Adaptive Control: The Model Reference Approach*. Marcel Decker Inc., 1979.

[21]M. A. Llama et al. *Stable fuzzy self-tuning computed-torque control of robot manipulators*. Proc. Int. Conf on Robotics and Automation, 1998, pp. 2369–2374.

[22]K.M. Lynch and F.C. Park. *MODERN ROBOTICS,MECHANICS, PLANNING, AND CONTROL*. Cambridge University Press, 2019, pp. 269–320.

[23]G. Leitmann M.J. Corless. *Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems*. IEEE Trans. Autom. Control 26 (5), 1981.

[24]H. Makino, A. Kato, and Y. Yamazaki. *Research and Commercialization of SCARA Robot The Case of Industry-University Joint Research and Development*. Int. J. of Automation Technology, Vol.1 No.1, 2007.

[25]B.R. Markiewicz. *Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator*. NASA-JPL Technical Memo, 1973, pp. 33–61.

[26]"MATLAB and Simulink for Technical Computing. The MathWorks Inc., USA. [Online]: http://www.mathworks.com/." In: ().

[27]Shimon Y. Nof. *Handbook of Industrial Robotics, Second Edition*. John Wiley Sons, Inc.

[28]Edeh O.D.F. and Ossia C.V. *Journal of Automation and Control, 2019, Vol. 7, No. 1*. 2019, pp. 7–14.

[29]B. Paden and R. Panja. *Globally asymptotically stable PD+ controllers for robotic manipulators*. Int. J. of Control, vol. 4 7, no. 6, 1988, pp. 1697–1712.

[30]Z. Qu and Y. Jin. *Robust control of nonlinear systems in the presence of unknown exogenous dynamics*. Proceedings of IEEE Conference on Decision and Control, 2001.

[31]T. Ravichandran, D. W. L. Wang, and G. R. Heppler. *Stability and Robustness of a Class of Nonlinear Controllers for Robot Manipulators*. Boston,Massachusetts,American Control Conference, 2004, pp. 5262–5267.

[32]A. Sabanovic, L.M. Fridman, and S. Spurgeon. *Variable Structure Systems from principles to implementation*. Institute of Engineering and Technology,London,United Kingdom, 2004, pp. 3–16.

[33]T. Samad and A.M. Annaswamy. *The Impact of Control Technology*. www.ieeecss.org, 2011, pp. 566–571.

[34]V. Santibanez, R. Kelly, and M. A. Llama. *Fuzzy PD+ control for robot manipulators*. Proc. of the 2000 IEEE Int. Cmif. on Robotics and Automation, 2000, pp. 2112–2117.

[35]H. Seraji. *A new approach to adaptive control of manipulators*. ASME J. Dynam. Syst. Measur. Control 109 (3), 1987.

[36]B. Siciliano et al. *Robotics Modeling,Planning and Control*. Springer-Verlag,London, 2010, pp. 247–338.

[37]J.J. E Slotine and J. Coetsee. *Adaptive sliding controller synthesis for non-linear systems*. Int. J. Control 43, 1986.

[38]J.J. E Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall Inc., 1991.

[39]J.J.E. Slotine and S.S. Sastry. *Tracking control of non-linear systems using sliding surface, with application to robot manipulators*. Int. J. Control 38 (2), 1983.

[40]J.J.E. Slotine and S.S. Sastry. *Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators*. International Journal of Control 38,2, 1983, pp. 465–492.

[41]M. Takegaki and S. Arimoto. *A new feedback method for dynamic control of manipulators*. J. Dyn. Sys., Meas. and Control, vol. 103, 1981, pp. 119–125.

[42]V.I. Utkin. *Variable structure systems with sliding modes*. IEEE Transactions on Automatic Control 22,2, 1977, pp. 212–222.

[43]J. T. Wen. *A unified perspective on robot control: the energy Lyapunov function approach*. Int. J. of Adaptive Control and Signal Processing, vol. 4, 1990, pp. 487–500.

[44]J. T. Wen and D. S. Bayard. *A new class of control laws for robotic manipulators - Part I: Non-adaptive case*. Int. J. of Control, vol. 47, no. 5, 1988, pp. 1361–1385.

[45]Y. Xu, J. M. Hollerbach, and D. Ma. *A nonlinear PD controllers for force and contact transient control*. IEEE Control Systems Magazine, vol. 15, no. 1, 1995, pp. 15–21.

[46]B. Yao and M. Tomizuka. *Robust adaptive motion and force control of robot manipulators in unknown stiffness environments*. Proc. IEEE Conf. Decision Control 1, 1993.

[47]B. Yao et al. *Adaptive robust control of single-rod hydraulic actuators: theory and experiments*. Proceedings of American Control Conference, 1999.

[48]K.D. Young, V.I. Utkin, and U.O zgu ner. *A control engineer's guide to sliding mode control*. IEEE Transactions on Control Systems Technology 7,3, 1983, pp. 328–342.

[49]K.D. Young, V.I. Utkin, and U. Ozguner. *A control engineers guide to sliding mode control*. IEEE Trans. Control Syst. Technol. 7 (3), 1999.

[50]M. Zeinali and L. Notash. *Adaptive sliding mode control with uncertainty estimator for robot manipulators*. Mechanism and Machine Theory, 2010, pp. 80–90.

[51]K. Zhou, C.J. Doyle, and K. Glover. *Robust Optimal Control*. Prentice-Hall,Inc., 1996.