Politecnico di Torino
Department of Electronics and Telecommunications

Institute for Engineering and Architecture

Thesis

For the Degree of

Master of Science in Mechatronics Engineering

Modeling and Control of a Four Axes CNC machine

By

Joud Basso (Matr.-Nr: S262607)

Submission Date: July, 2021

Mentor:	Prof. Alessandro Rizzo
Supervisor:	Ing. Giovanni Guida – Brain Technologies

Abstract

The principal goal of this report is to introduce the implementation of the MOREPRO project.

This project would not have been possible without the assistance of:

- Brain Technologies s.r.l.
- Machining Centers Manufacturing S.p.A.
- CAMS di Corda Fausto & C. S.a.s.
- ALMEC S.r.l.

The MOREPRO project's primary objective is to build predictive algorithms capable of estimating an end-usury effector's and state of health. To accomplish this, a precise model of the machine and an effective control system are needed. Several chapters address the aspects of the work on this thesis.

The first chapter serves as a brief introduction to the project and the thesis work's objectives. The second defines CNC machines. The third and fourth chapters outline the kinematical and dynamical models of the CNC machine.

The fifth chapter discusses two distinct types of control algorithms. Some additional experiments are then carried out in the sixth chapter to determine which control would be more suitable to control the machine motion.

Acknowledgements

A special thanks to my advisor at Politecnico di Torino, Prof. Alessandro Rizzo, for being such an excellent coordinator and providing me with helpful advice.

Additionally, I'd like to express my gratitude to the workers at Brain Technologies for their assistance and invaluable support. I would like to express my gratitude to my advisor, Ing. Giovanni Guida, for his invaluable support in the process of comprehending, preparing, and developing the research work.

Finally, I'd like to express my gratitude to my family and friends for their encouragement and support during this work.

Contents

LIST OF FIGURES

LIST OF ABBREVITIONS

HW hardware

SW software

HIL hardware in the loop

SIL software in the loop

PI proportional integrator

PD proportional derivative

CNC computer numerical control

SCADA supervisory control and data acquisition

ERP enterprise resource arranging

MES manufacturing execution system

IDK inverse differential kinematics

HT homogeneous transformation

MCU machine control unit

CHAPTER 1: INTRODUCTION

This chapter introduces the idea of Industry 4.0, which includes our MOREPRO project, and defines its objectives. Additionally, the thesis work's objective is presented.

1.1 Industry 4.0

Industry 4.0 is a holistic automation, business data, and assembling execution engineering. It aims to improve the industry by mixing all the parts of the production and trade across the organization limits for more prominent proficiency. The term industry 4.0 began in Germany, yet the ideas are in congruity with the overall activities, including smart processing plants, Industrial Internet of Things, intelligent and progressed manufacturing [1]. Industry 4.0 includes an entire arrangement of inventive forms and advancements that are combining modern innovations with industry measures within the manufacturing sector in arrange. This is done in order to serve the progressively fast-moving markets. The internet has on a very basic level changed communication and shopper conduct. Nowadays, much appreciated to web and cloud innovation, data can be shared around the globe. Web-based information sets are not limited to specific devices or computer program and that can be downloaded and adjusted in an instant, no matter where you're within the world. This modern playing field will alter the face of industry and manufacturing until the end of time. Industry 4.0 is superior defined not by its basic innovations, but by the worldview move that's moving us away from centralized control to a world of decentralized, cleverly process units [2]. Whereas the third industrial revolution centered on mechanization and presented computer numerical control (CNC), supervisory control and data acquisition (SCADA), enterprise resource arranging (ERP), and manufacturing execution system (MES), the developments of Industry 4.0 spin around keywords such as connected, real-time, integrated, measured, artificial intelligence, and intelligent manufacturing as shown in figure (1). Smart manufacturing is the method of minimizing human interaction and to use the human brain only when it is needed. In spite of the fact that a few of the objectives for Industry 4.0 are very ambitious, the hope is that it can advance into artificial intelligence and automated decision-making, close idealize mechanical automation and advantageous human integration, and have manufacturing facilities that are totally interconnected and 'smart', from crude materials to wrapped up products [3].

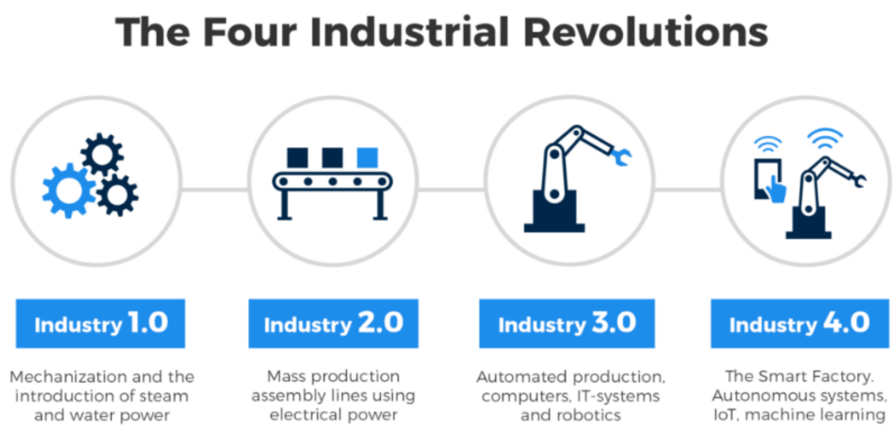

Figure 1: The revolution of the industries [10]

1.2 MOREPRO

The creation of the MOREPRO project served as the foundation for the thesis work. MOREPRO was born in the industry 4.0 space to address the need to improve the efficiency and management of manufacturing production systems. It is important to reduce the probability of accidental stops and product quality drift. In addition to the need to manage maintenance interventions in order to keep the production system in good working order, the plant's output must be maximized. Herein lies the significance of MOREPRO, which seeks to create an advanced distributed intelligence system capable of enhancing the management of manufacturing plants through simulation, monitoring, and prediction of the state of health and usury of a CNC machine's end-effector. It is looking to create a modern digital system architecture that integrates machine controllers and plant managers to maintain the manufacturing line in both peak condition, as well as overall control through its specific procedures (monitoring and forecast of the residual life of the tools) and of the productivity of the systems (predictive and opportunistic management maintenance interventions). As a result, the MOREPRO project aims to bring a digital approach to the creative area of modular production systems that is currently lacking in the national and international production systems. It is built on a multi-level distributed logic architecture and includes the following features:

1. Monitor the state of health of critical components of the machine and system.
2. Monitor the usury of the end-effector.
3. Create models to predict the state of health and usury of the end-effector.

From a hardware and software design perspective, the prototype has the following attributes:

4. A first level of field on the computer (Edge) for local monitoring of the machine's health. In order to avoid anomalies and respond to alarm conditions, the signal is sent to a second level (Server) for asynchronous processing and to the single machine for potential synchronous actuations.
5. A second level (Server) in the private or public cloud to produce statistics and a database of diagnostics and maintenance policies, to facilitate the implementation of digital twin methodologies, and to measure the necessary parameters for reconfiguration.

These final two levels must be structured in such a way that interoperability and operational stability are maintained even in the case of communication failure. Additionally, the second level (Server) must be capable of exchanging data with other digital systems usually found in manufacturing plants, such as pre-existing line supervision tools, to ensure an appropriate level of transparency and security.

The project possesses completely unique characteristics that are not currently available on the market. In reality, the MOREPRO project will allow the development of a completely novel approach to monitoring the health of machine components. Depending on the degree of automation and organization already present in Italian manufacturing, the MOREPRO solution will provide an improvement in overall production productivity of at least 20% and a reduction in current maintenance costs of at least 25%.

The MOREPRO project can be placed in the context of predictive and opportunistic management of systems flexible production. The project presents a series of highly innovative aspects, which can be identified in the following points:

1. Distributed intelligence architecture for the intelligent control and implementation of flexible production systems.
2. Innovative techniques for monitoring and forecasting the state of health of the end-effector and the machine.

From the architectural and HW and SW implementation level, there are two elements that distinguish MOREPRO's proposed solution from what is currently available on the market:

1. The distributed intelligence architecture that allows you to monitor the health status of the machine.
2. Edge computing capacity implemented in the "edge devices" installed in the machine, which results in faster and more effective responses and reactions to degraded states identified in real time, as well as in the optimization of data transfers relating to locally pre-processed information.

The edge devices reflect the machine's local intelligence nodes; they are capable of transmitting a set of signals that the system can use to produce useful information for the project's objectives.

The project is intended to accomplish two distinct objectives:

1. **State of health and usury of the machine components**. This is done by the usage of predictive algorithms to define the health status and usury of the machine components.
2. **State of health and usury of the end-effector**. This is done by the usage of algorithms to define the state of health and the usury of the end-effector.

1.3 State of the Art

Modeling a robot, a manipulator or even a CNC machine is an important task to be performed accurately. A real mechanical system can always be replaced with a simulated one allowing to obtain reliable control systems without expensive experimental tests. Therefore, the accuracy and precision in controlling every mechanical system is based on mathematical models. For these reasons, the problem of finding accurate models for mechanical systems in order to achieve stable control systems is an important task.

"Adaptive control system" tries to learn the uncertain parameters of the system and achieves the best performance [14]. But the disadvantage of this type of control is that stability is not treated rigorously. Also, the high gain observes is needed to avoid full state measurement. Other than that, the system relatively slows convergence and high cost is produced and the process is very complex [15].

"Decentralized control system" controls each joint independently and all coupling effects like inertial, coriolis, centrifugal, and gravitational forces are considered as disturbances into the system. The advantage of this type of control is that it is easy to be implemented.

"Robust control system" are centralized control type where all the joints are controlled by a single controller. It is a fixed structure that guarantees stability and performance in bounded uncertainties.

The machine considered as case study does not have coriolis and centrifugal forces because it is composed of four joints where the first three are prismatic and only the last one is revolute, therefore, no coriolis or centrifugal forces are present. Moreover, the dynamic parameters of the machine studied are estimated. So, it is appropriate to develop robust and decentralized controls and perform tests to choose the appropriate control from these two to control the action of the machine and have a stable performance.

1.4 Partnership

Different companies have been participated in the development of the MOREPRO project. Each company has its own role and different skills in the project, which can be used to create a good predictive monitoring system solution. Below is presented the list of all the companies that have been participated in the development of the MOREPRO project and the role of each one of them.

- Brain Technologies: The role of Brain Technologies within the project is as lead partner and project manager. It also provides technical and scientific services for the industrial project. The company's main task is to develop the prediction system and computer peripherals and the edge-computing devices. It also participated in the implementation and verification of the final prototype on the production line.
- Machining Centers Manufacturing S.p.A. (MCM): It participates in defining the requirements and limitations of the algorithm. Its main function is to collect machine data, assist in installing the MOREPRO solution on the CAMS machine and finally test and verify the developed algorithm.
- AL.MEC: It is a manufacturer of electromechanical components. His main task at MOREPRO is to develop and manufacture electronic circuit boards that can collect and process data from sensors.
- CAMAS: it is a precision machining company involved in defining requirements for predictive monitoring systems, defining use cases, testing implementation, and verifying monitoring systems on the production line.

1.5 Team Organization

The MOREPRO initial work flow was accomplished by a teamwork of six mechatronics engineering students studying at Politecnico di Torino under the guidance of Brain Technologies engineer and Politecnico di Torino professor from the department of Control and Computer Engineering (DAUIN). After explaining the purpose of the project and the first idea for the first implementation, the team needs to be divided.

The work flow of MOREPRO is shown in figure (2). This figure shows all the requirements of the project. The first step in the project is to perform models of the machine used for prediction purposes. After these, translation of everything comes into executable code where all the needed softwares should run on edge devices. At the end, several tests of the device are done (hardware in the loop and software in the loop validations).

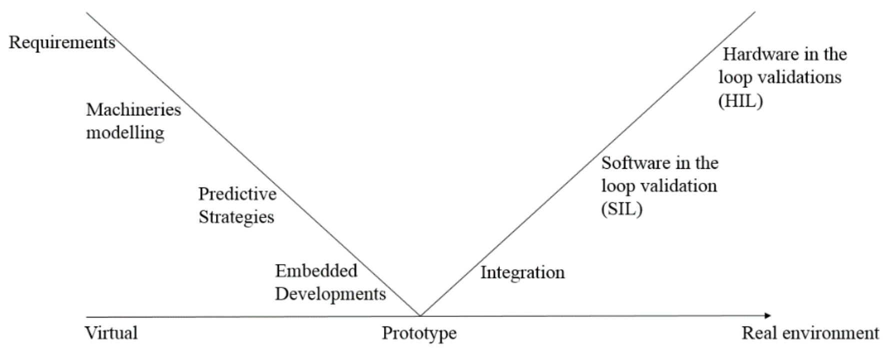

Figure 2: The work flow in the MOREPRO project

The group subdivision shown in figure (3) is as follows:

- Requirements team: This group was responsible in identifying the requirements to assign the duties of the other teams. Another task assigned to the members of this group was to run experiments to run some implementation tests.
- Modeling and control team: This second group is responsible of developing the kinematical, differential and dynamical models of the machine. Afterwards, an appropriate control is performed to control the axis of the machine.
- Prediction team: this team executes the core of the MOREPRO project; in fact the main goal is to perform prediction algorithms in order to estimate the usury and state of health of the machine. The first step carried out by this group is to develop a model of the machine. After that, a multi-model approach is carried out to achieve the main goal which is the estimation of the wear and state of health of the machine.

  In sum, the work of all the groups was at the first three phases of the project which is in the virtual environment. Throughout the work, the original team division was not adhered to; after the first few months, some team members began to work alone, and another started new tasks and stayed in teams.

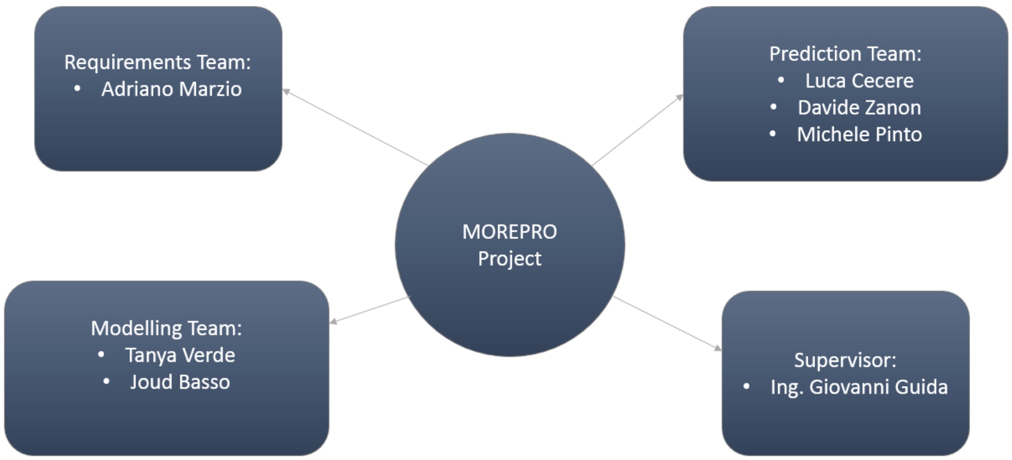

Figure 3: the tasks assigned to each group of students in the project

1.6 Objective of the thesis

An essential phase in MOREPRO project is the modeling phase. It is essential due to the need of a machine model used to predict the state of health and usury of the end-effector. The achieved models are the kinematical and dynamical models of the machine. After finding the models, two control strategies of the machine axes are performed. These two control algorithms allow having the best control of the machine. The first control algorithm is the decentralized control where the movement of each joint was controlled alone. The second control algorithm is the robust centralized control where all the machine axes were controlled together based on the dynamics of the machine. When both of these controls are done, their simulation results are compared to determine which control is more appropriate for the machine.

The machine to be modelled and controlled is a CNC machine. It is shown in the figure below:

II. Kinematic Model of Machine

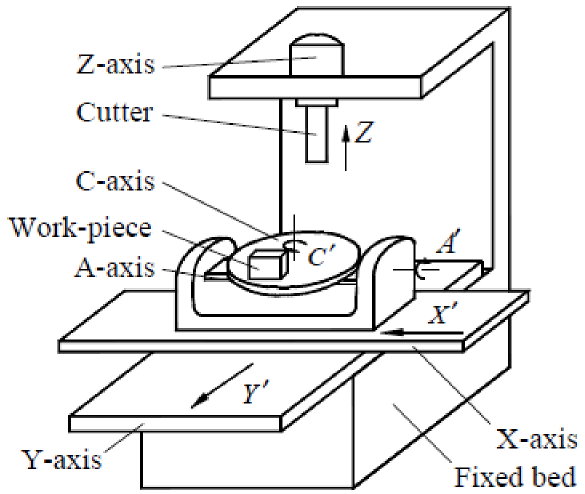

Figure 4: the machine model

This machine is a milling machine composed of five axes, the first three are prismatic around the x, y and z axis and the last two are revolute. In our study, we will consider only the first four axis, the last one is not considered because it is always rotating at a constant speed.

CHAPTER 2: CNC MACHINE

2.1 What is a CNC machine

A CNC machine is a subtractive production process that usually uses machine tools to extract layers from the workpiece and creates a custom-designed component [16]. The work piece to be shaped can be made of plastic, wood, metals…etc. the common machining operations are turning, drilling, milling and surface grinding as shown in the figures below:

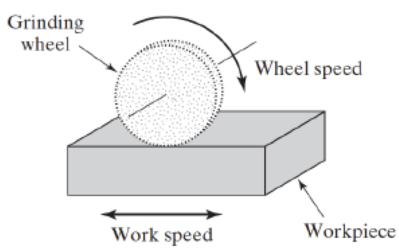

Figure 5: Grinding

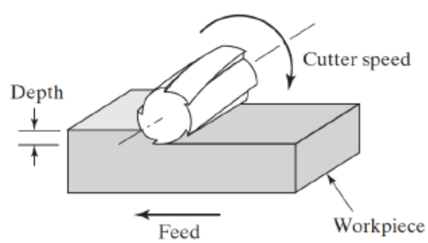

Figure 6: Peripheral milling

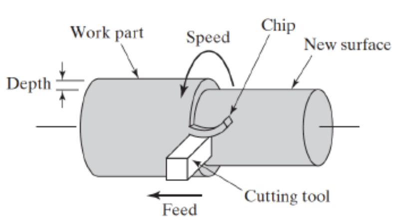

Figure 7: Turning

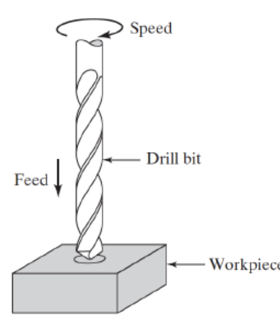

Figure 8: Drilling

2.2 CNC Basic Components

The basic components of a CNC machine are shown in figure (9):

- Part program of instructions: it is the detailed set of step-by-step instructions. It is a code written in a certain language.
- Machine control unit (MCU): it is the interface between the software part and the hardware part. It is a microcomputer and related control hardware that performs the following actions:
    1. Saves the instructions of the program.
    2. Executes the program instructions.
    3. Convert each instruction into a mechanical action of the processing equipment, one instruction at a time.
- Processing equipment: it executes the productive work. Process equipment usually consists of a worktable and a spindle that makes the workpiece revolving around itself, as well as motors and controllers that drive them.

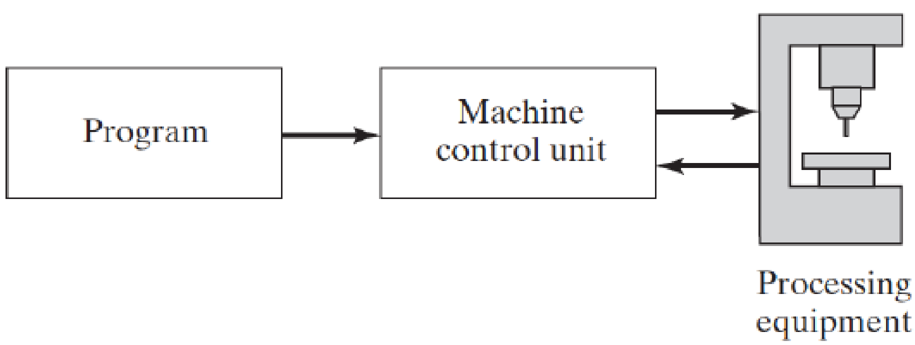

Figure 9: CNC basic components

2.3 Motion Control Systems

In CNC machines, there are two kinds of motions:

1. Point-to-point systems: It is also called positioning systems. They move the worktable to a programmed location regardless of the route to that location. After the movement is over, the working head performs processing actions at this time, such as drilling or punching a hole. Therefore, the program consists of a series of point positions for performing operations. This motion control system is shown in the figure below:

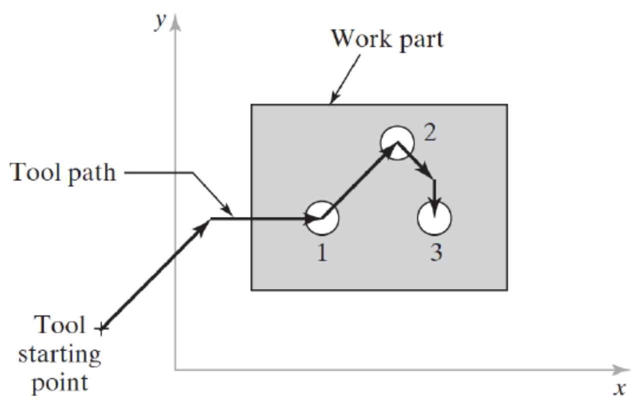

Figure 10: Point-to-point system

   In this motion control system, the machine is provided with a sequence of points. Then, the machine will decide how to go there. Most of the time, if the machine does not need to machine along the path, the path will be a straight line.

2. Continuous path systems: The possibility of continuous simultaneous control of two or more axes is guaranteed in this motion control system. This allows you to control the path of the tool relative to the workpiece. In this case, the tool performs the process while the worktable is moving, which allows the system to create corners, 2D curves, or 3D contours on the workpiece. Many milling and turning operations require this type of control. This motion control system is shown in the figure below:

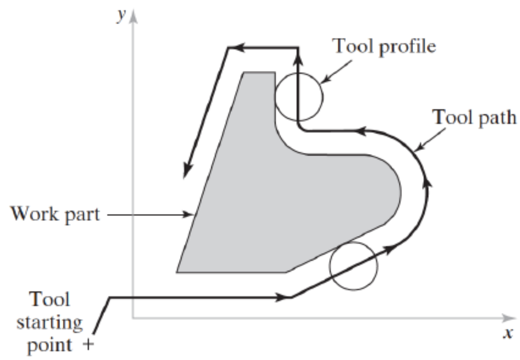

Figure 11: Continuous path systems

In this type of motion control system, the path matter because the piece is processed while the tool is moving. When the tool travels, it machines the piece. Therefore, the trajectory of the cutting tool matters.

CHAPTER 3: INTRODUCTION TO KINEMATICS

Modeling is a very important and necessary step in Robotics; the obtained mathematical models can be used to perform some simulations that can allow to better understand the robot movement and behaviors. The most important benefit in modelling and simulating a system is that using environment of simulation, like MathWorks (MATLAB, Simulink) which is less expensive than performing real test with the system and also helps to collect data. The initial models, normally, are as simple as possible in order to explain the main characteristics of the system. But in order to have good information from the simulation performed, it is better to create a model closer enough to reality. This means increasing the level of detail and accuracy of the model.

In the following paragraphs, kinematical model is described. After that, the kinematical model of the machine is studied in details. The kinematic model is used in order to implement movement control strategies and is a description of the motion without considering dynamics components (like force and torque).

3.1 What is a Kinematical Model

A common manipulator, robot, or in our case study a CNC machine, can be easily and schematically represented as a succession of links (that represent the arms of the robot), connected between them with joints (which are the actuators of each arm). The succession of links and joints constitutes what is called kinematic chain. There are two kinds of joints:

- Prismatic (translational) joints: they allow a translation between the connected links.
- Revolute (rotational) joints: they allow a rotation between the connected links.

In addition, two other important parts can be distinguished in the kinematic chain, the end-effector and the base. The base is fixed while the end-effector is attached to the last link and is able to perform tasks or hold a tool. The end-effector is the interface of the robot with the environment. The robot movements can be described precisely by finding the position and orientation. The position and orientation of the robot are described generally with respect to the base reference frame which is fixed.

From a kinematical point of view, it can be distinguished the following classes:

- Direct Kinematics: it describes the position and orientation of the end-effector based on the joints variables.
- Inverse Kinematics: it allows obtaining the joint variables based on the position and orientation of the end-effector.

3.2 The Kinematical Model of the Machine

There are different approaches that can be used to find the direct kinematics of the machine. The position and orientation of the robot is described generally with respect to the base reference frame which is fixed. One of the approaches is the “Denavit-Hartenberg” approach. It is a systematic approach to find the pose (position and orientation) of the end-effector with respect to the base reference frame. This approach is normally used when having difficult geometric configurations of the robot. Another approach that can be used to describe the pose of the end-effector is the application of the “Geometric” approach. This latter can be used when having easy robot configurations.

The CNC machine axis considered as case study in MORPRO project is shown in the figure (12). It can be seen that the machine is composed of two open kinematic chains. On one side, there is the cutter while on the other side there is the plate where the workpiece is positioned.

In the thesis work, the open chain of the cutter is not taken into considerations, but a manipulator composed of 5 joints, of which the first three are prismatic and the last two are rotational joints is studied. In addition, the last revolute joint is not considered in the dynamic and kinematic analysis (joint 5) because it is used for machining and rotate at a certain speed to shape the material. Therefore, it does not perform any other movement and does not affect the kinematical and dynamical models.

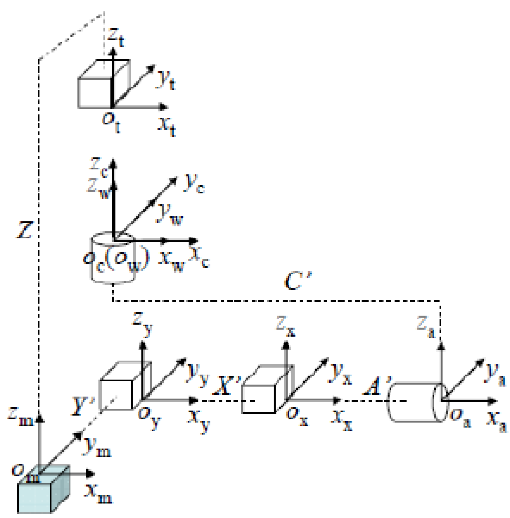

Figure 12: CNC machine chain

In the figure below, there is a representation of the scheme adopted for the modelling:

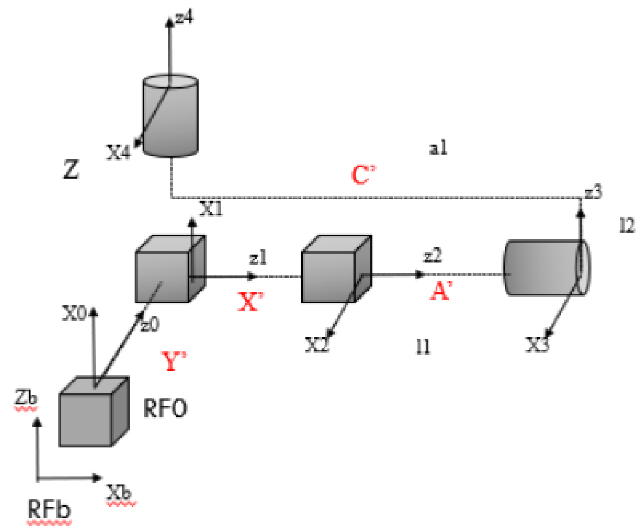

Figure 13: Considered model of the CNC machine

Due to the easiest machine geometric configuration, the pose of the end-effector (direct kinematics) can be found easily using the geometric approach. To do so, the homogeneous transformation matrices are calculated between each base frame and the base reference frame.

$$T_1^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & a_1 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & a_2 + d_2 \\ 1 & 0 & 0 & a_1 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^0 = \begin{bmatrix} 0 & 0 & 1 & a_3 + d_3 \\ 0 & -1 & 0 & a_2 + d_2 \\ 1 & 0 & 0 & a_1 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^0 = \begin{bmatrix} 0 & 0 & 1 & a_4 + a_3 + d_3 \\ sin\vartheta & -cos\vartheta & 0 & a_2 + d_2 \\ cos\vartheta & sin\vartheta & 0 & a_1 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Denavit-Hartenberg approach is used also to find the pose of the end-effector. This approach is used normally for complicated manipulators like anthropomorphic because the level of complexity is high and it is difficult to find geometrically the manipulator pose. Using this approach, there is a rule to position the reference frames and find the pose between one joint and the other. The rule is described as follows and can be seen in figure (14):

- Choose axis z_i along the axis of joint i+1.
- Locate the origin O_i at the intersection of axis z_i with the common normal to axes z_{i-1} and z_i. Also, locate $O_{i'}$ at the intersection of the common normal with axis z_{i-1}.
- Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from joint i to joint i + 1.
- Choose axis y_i so as to complete a right-handed frame.

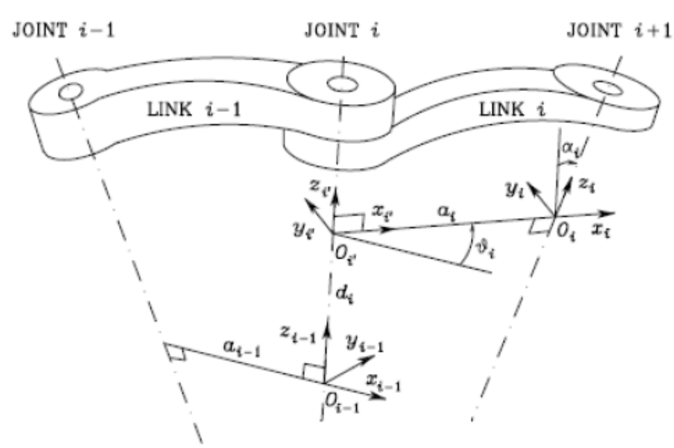

Figure 14: Denavit-Hartenberg convention

According to Denavit-Hartenberg convention, the most important parameters describing two consecutive reference frames are as follows:

- a_i: is the "link length". It is the distance between two origins $O_i\ O_{i'}$.
- d_i: is the "link offset". It is the coordinate of $O_{i'}$ along z_{i_1}. In case the joint is prismatic, so d_i will be the joint variable.
- α_i: is the "link twist". It is angle between z_{i1} and z_i along x_i to be taken positive when the rotation is counter-clockwise.
- ϑ_i: is the "joint angle". It is the angle between the axes $x_{i-1}$and x_i along the axis z_{i-1}. In case the joint is revolute, so ϑ_i will be the joint variable.

a_i and α_i are constant parameter that depends only on the geometry of the manipulator. After defining how to position the reference frames on each joint and what are the parameters used in Denavit-Hartenberg convention, the parameters of our configuration can be found and they are as follows:

DH parameters	a_i	d_i	α_i	ϑ_i
Joint 1	0	$a_1 + d_1$	-90°	0°
Joint 2	0	$a_2 + d_2$	-90°	0°
Joint 3	0	$a_3 + d_3$	0°	0°
Joint 4	0	a_4	0°	t_4

The prismatic joint variables are d_1, d_2 and d_3 while t_4 is the revolute joint variable. Using the Denavit-Hartenberg parameters found before and putting them in the roto-translation matrix below, the homogeneous transformation matrix relating two consecutive reference frames can be obtained.

$$A_i^{i-1}(q_i) = A_{i'}^{i-1} A_i^{i'} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} c_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By multiplying all the homogeneous transformation matrices, the direct kinematics is obtained:

$$A_0^n(q) = A_1^0 A_2^1(q) A_3^2$$

CHAPTER 4: DYNAMICAL MODEL AND PARAMETRES IDENTIFICTIONS

A mathematical model is an algorithmic rule or set of conditions that is combined with a group of data values represents the crucial behavior of a system, process, or phenomenon [4]. In this chapter, the dynamical model of the machine is found. The dynamical model is useful for the control of the axes of the machine. The machine has four axes, the first three are translational axes around the x, y and the z axes and the last one is rotational. After finding the dynamical model of the machine, it is linearized. Then, "Least-Squares" minimization technique is used to find the dynamical parameters of the dynamical model.

4.1 Dynamical Model

The determination of the dynamic model of a manipulator plays a critical role for simulation of movement and design of control algorithms. The "Lagrange formulation" is used to derive the equations of motion of the machine in the joint space. The Lagrange method is conceptually simple and systematic. First of all, the generalized coordinates have to be chosen $q_i \quad i = 1 \dots n$ where n is the number of joints and q_i describes the link position. In case the joint is revolute q_i is the angle ϑ_i or if the joint is prismatic q_i is the distance d_i

The Lagrangian of a mechanical system is defined as:

$$L = T - U$$

Where T is the kinetic energy and U the potential energy of the system.

The Lagrange's dynamical equation is equal to:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \varepsilon_i \qquad i = 1 \dots n$$

Where ε_i denotes the generalized force associated with the generalized coordinates q_i. The generalized forces are given by the joint actuator torque, the joint friction torque and the torques induced by the end-effector when it is in contact with the environment. In order to find the dynamical equation describing the movement of each joint, the kinetic energy and the potential energy of each joint and link should be found. In the following paragraphs, the kinetic energies and the potential energies are found in order to obtain the final dynamical equation describing the motion of each joint in the joint space.

4.1.1 Computation of the Kinetic Energy

The total kinetic energy is equal to the sum of the kinetic energy of all links and motors that actuate each joint:

$$T = \sum_{i=1}^{n} (T_{li} + T_{mi})$$

The expression of the kinetic energy of link i is equal to:

$$T_{li} = \frac{1}{2} m_{li} \dot{p}_{li}^T \dot{p}_{li} + \frac{1}{2} w_i^T I_{li} w_i$$

Where m_{li} denotes the mass of link i and $\dot{p}_{li}$ the translational velocity of link i as it can be seen in figure (15). w_i is the angular velocity of link i expressed in the base reference frame. I_{l_i} is the matrix of the inertia tensor relative to the center of mass of link i when expressed in the base reference frame.

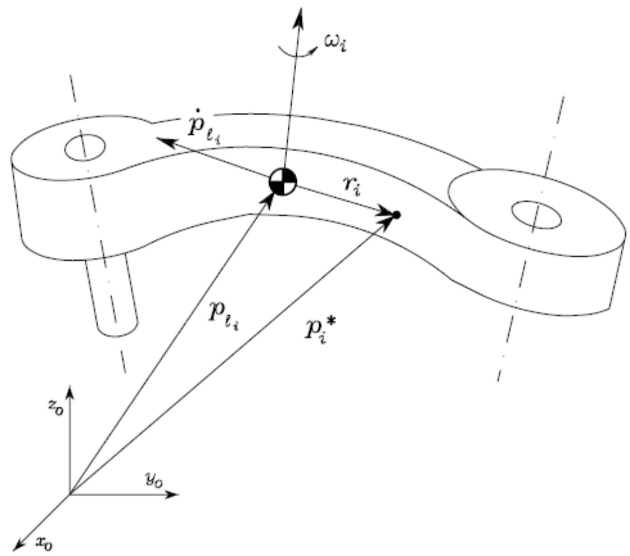

Figure 15: Kinematic description of link i for the Lagrange formulation

Referring to the link reference frame the inertia tensor $I_{l_i}^i$ is constant, so the following relation can be written: $I_{li} = R_i I_{l_i}^i R_i^T$. R_i is the rotation matrix between the frame attached to link i and the base reference frame. Therefore, the expression of the kinetic energy becomes:

$$T_{li} = \frac{1}{2}\, m_{li}\, \dot{p}_{li}^T \dot{p}_{li} \; + \frac{1}{2}\, w_i^T R_i I_{li}^i R_i^T w_i$$

The first term in the expression of the kinetic energy of link i represents the translational contribution and the second one represents the rotational contribution. $\dot{p}_{li}$ and w_i can be found from the jacobians as follows: $\dot{p}_{l_i} = J_p^{(l_i)} \dot{q}$ *and* $w_i = J_O^{(l_i)} \dot{q}$. $J_p^{(l_i)}$ *and* $J_O^{(l_i)}$ have the following forms:

$$J_p^{(l_i)} = \left[Jp_1^{(l_i)} \ldots \; Jp_i^{(l_i)} \;\; 0 \ldots 0 \right] \text{ where } Jp_j^{(l_j)} = \begin{cases} z_{j-1} & \text{for a prismatic joint} \\ z_{j-1} * (p_{li} - p_{j-1}) & \text{for a revolute joint} \end{cases}$$

$$J_O^{(l_i)} = \left[Jo_1^{(l_i)} \ldots \; Jo_i^{(l_i)} \;\; 0 \ldots 0 \right] \text{where } JO_j^{(l_j)} = \begin{cases} 0 & \text{for a prismatic joint} \\ z_{j-1} & \text{for a revolute joint} \end{cases}$$

z_{j-1} is the unit vector of the axis z of frame j-1and p_{j-1} is the position vector of the origin of reference frame j-1. Hence, the final expression of the kinetic energy of link i in function of the generalized coordinates can be written as:

$$T_{li} = \frac{1}{2}\, m_{li}\, \dot{q}^T J_p^{(l_i)T} J_p^{(l_i)} \dot{q} + \frac{1}{2}\, \dot{q}^T J_O^{(l_i)T} R_i I_{li}^i R_i^T J_O^{(l_i)} \dot{q}$$

The computation of the kinetic energy of the motor that actuates the joint can be done in similar way as the link. The kinetic energy of motor i can be written as:

$$T_{mi} = \frac{1}{2} m_{m_i}\, \dot{p}_{m_i}^T \dot{p}_{m_i} + \frac{1}{2}\, w_{m_i}^T I_{m_i} w_{m_i}$$

Where m_{m_i} denotes the mass of the rotor, $\dot{p}_{m_i}$ is the linear velocity of the center of mass of the rotor, I_{m_i} denotes the inertia tensor of the rotor relative to its center of mass and w_{m_i} is the angular velocity of the rotor.

The expression of $\dot{p}_{m_i}$ can be written as a function of the joint variables as:

$\dot{p}_{m_i} = J_p^{(m_i)} \dot{q}$ where $J_p^{(m_i)} = \left[Jp_1^{(m_i)} \ldots Jp_{i-1}^{(m_i)}\ 0 \ldots 0 \right]$

Each column of $J_p^{(m_i)}$ can be found as $Jp_j^{(m_i)} \begin{cases} z_{j-1} & \text{for a prismatic joint} \\ z_{j-1} * \left(p_{mi} - p_{j-1}\right) & \text{for a revolute joint} \end{cases}$

The angular velocity w_{m_i} expression in function of the joint variable is:

$w_{m_i} = J_O^{(m_i)} \dot{q}$ where $J_O^{(m_i)} = \left[Jo_1^{(m_i)} \ldots\ Jo_i^{(m_i)}\ 0 \ldots 0 \right]$

Each column of $J_O^{(m_i)}$ can be found as $Jo_j^{(m_j)} = \begin{cases} Jo_j^{(l_i)} & j = 1 \ldots . i-1 \text{ for a prismatic joint} \\ k_{ri} * z_{mi} & j = i \quad \text{for a revolute joint} \end{cases}$

The final expression of the kinetic energy of the motor actuating each joint as a function of the joint variables is as follows:

$$T_{mi} = \frac{1}{2}\, m_{m_i}\, \dot{q}^T J_p^{(m_i)T} J_p^{(m_i)} \dot{q} + \frac{1}{2}\, \dot{q}^T J_O^{(m_i)T} R_{m_i} I_{m_i}^{m_i} R_{m_i}^T J_O^{(m_i)} \dot{q}$$

Now, the final expressions of the kinetic energy of the links and the joints as function of the joint variables can be found. Summing the two expressions, the total kinetic energy is obtained and has a quadratic form as:

$$T = \sum_{i=1}^{n} (\, T_{li} + T_{mi}) = \frac{1}{2}\, \dot{q}\, B(q) \dot{q}$$

Where $B(q)\ \epsilon\ R^{n*n}$ is the inertia matrix. It is symmetric, positive definite and configuration dependent matrix. The expression of $B(q)$ is:

$$B(\mathrm{q}) = \sum_{i=1}^{n}(m_{li} J_p^{(l_i)^T} J_p^{(l_i)} + J_O^{(l_i)^T} R_i I_{li}^i R_i^T J_O^{(l_i)} + m_{mi} J_P^{(m_i)^T} J_P^{(m_i)} + J_O^{(m_i)^T} R_{mi} I_{mi}^{mi} R_{mi}^T J_O^{(m_i)})$$

In order to find $B(\mathrm{q})$, the inertia matrix of our machine, $J_p^{(l_i)}, J_O^{(l_i)}, J_p^{(m_i)}$ and $Jo_j^{(m_j)}$ should be found firstly. In our case study, the machine has four axes three prismatic and one revolute, so $J_p^{(l_i)}, J_O^{(l_i)}, J_p^{(m_i)}$ and $Jo_j^{(m_j)}$ are matrices of the following forms:

$$J_p^{(l_1)} = \left[J_{p_1}^{(l_1)}\ J_{p_2}^{(l_1)}\ J_{p_3}^{(l_1)}\ J_{p_4}^{(l_1)}\right] = [z_0\ 0\ 0\ 0]$$

$$J_p^{(l_2)} = \left[J_{p_1}^{(l_2)}\ J_{p_2}^{(l_2)}\ J_{p_3}^{(l_2)}\ J_{p_4}^{(l_2)}\right] = [z_0\ z_1\ 0\ 0]$$

$$J_p^{(l_3)} = \left[J_{p_1}^{(l_3)}\ J_{p_2}^{(l_3)}\ J_{p_3}^{(l_3)}\ J_{p_4}^{(l_3)}\right] = [z_0\ z_1\ z_2\ 0]$$

$$J_p^{(l_4)} = \left[J_{p_1}^{(l_4)}\ J_{p_2}^{(l_4)}\ J_{p_3}^{(l_4)}\ J_{p_4}^{(l_4)}\right] = [z_0\ z_1\ z_2\ z_3 * (p_{l_4} - p_3)]$$

$$J_O^{(l_1)} = \left[J_{O_1}^{(l_1)}\ J_{O_2}^{(l_1)}\ J_{O_3}^{(l_1)}\ J_{O_4}^{(l_1)}\right] = [0\ 0\ 0\ 0]$$

$$J_O^{(l_2)} = \left[J_{O_1}^{(l_2)}\ J_{O_2}^{(l_2)}\ J_{O_3}^{(l_2)}\ J_{O_4}^{(l_2)}\right] = [0\ 0\ 0\ 0]$$

$$J_O^{(l_3)} = \left[J_{O_1}^{(l_3)}\ J_{O_2}^{(l_3)}\ J_{O_3}^{(l_3)}\ J_{O_4}^{(l_3)}\right] = [0\ 0\ 0\ 0]$$

$$J_O^{(l_4)} = \left[J_{O_1}^{(l_4)}\ J_{O_2}^{(l_4)}\ J_{O_3}^{(l_4)}\ J_{O_4}^{(l_4)}\right] = [0\ 0\ 0\ z_3]$$

$$J_p^{(m_1)} = \left[\ J_{p_1}^{(m_1)}\ \ J_{p_2}^{(m_1)}\ \ J_{p_3}^{(m_1)}\ \ J_{p_4}^{(m_1)}\right] = [0\ 0\ 0\ 0]$$

$$J_p^{(m_2)} = \left[\ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\right] = [z_0\ 0\ 0\ 0]$$

$$J_p^{(m_2)} = \left[\ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\right] = [z_0\ z_1\ 0\ 0]$$

$$J_p^{(m_2)} = \left[\ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\ \ J_{p_1}^{(m_2)}\right] = [z_0\ z_1\ \ z_2 * \left(p_{m_2} - p_1\right)\ \ 0]$$

$$J_O^{(m_1)} = \left[J_{O_1}^{(m_1)}\ J_{O_2}^{(m_1)}\ J_{O_3}^{(m_1)}\ \ J_{O_{41}}^{(m_1)}\right] = [k_{r_1} * z_0\ \ 0\ \ 0\ \ 0]$$

$$J_O^{(m_2)} = \left[J_{O_1}^{(m_2)}\ J_{O_2}^{(m_2)}\ J_{O_3}^{(m_2)}\ J_{O_{41}}^{(m_2)}\right] = [J_{O_1}^{(l_2)}\quad k_{r_2} * z_1\ \ 0\ \ 0]$$

$$J_O^{(m_3)} = \left[J_{O_1}^{(m_3)}\ J_{O_2}^{(m_3)}\ J_{O_3}^{(m_3)}\ J_{O_{41}}^{(m_3)}\right] = [J_{O_1}^{(l_3)}\ J_{O_2}^{(l_3)}\ k_{r_3} * z_2\ \ 0]$$

$$J_O^{(m_4)} = \left[J_{O_1}^{(m_4)}\ J_{O_2}^{(m_4)}\ J_{O_3}^{(m_4)}\ J_{O_{41}}^{(m_4)}\right] = [J_{O_1}^{(l_4)}\ J_{O_2}^{(l_4)}\ J_{O_3}^{(l_4)}\ k_{r_4} * z_3]$$

Using MATLAB, the expression of the inertia matrix $B(\mathrm{q})$ of the machine is obtained. $B(\mathrm{q})$

$$= \begin{bmatrix} I_{m1} * kr_1^2 + ml_1 + ml_2 + ml_3 + ml_4 & 0 & 0 & 0 \\ 0 & I_{m2} * kr_2^2 + ml_2 + ml_3 + ml_4 & 0 & 0 \\ 0 & 0 & I_{m3} * kr_3^2 + ml_3 + ml_4 & 0 \\ 0 & 0 & 0 & I_{m4} * kr_4^2 + I_{l4} \end{bmatrix}$$

4.1.2 Computation of the Potential Energy

As for the kinetic energy, the potential energy can be found as the sum of the potential energies of the links and the motors that actuate the links:

$$U = \sum_{i=1}^{n} (U_{l_i} + U_{m_i})$$

The kinetic energy of link i U_{l_i} is equal to: $U_{l_i} = -m_{l_i} g_0^T p_{l_i}$ where g_0 is the gravity acceleration vector referred to the base reference frame and p_{l_i} is the position of the center of mass.

The kinetic energy of motor i is: $U_{m_i} = -m_{m_i} g_0^T p_{m_i}$.

Summing the expressions of the kinetic energy of the link and the motor, the final expression of the potential energy is given by:

$$U = -\sum_{i=1}^{n} (m_{l_i} g_0^T p_{l_i} + m_{m_i} g_0^T p_{m_i})$$

From the final expression of the potential energy, it can be seen that the potential energy depends only on the joint variables q and not on the joint velocities $\dot{q}$. This dependency on the joint variables is due to the dependency of p_{l_i} and p_{m_i} only on the joint variables.

4.1.3 Equation of Motion

After computing the kinetic energy and the potential energy, the Lagrangian can be written as:

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

Taking the derivatives required by Lagrange's dynamical equation yield to:

$$B(\mathrm{q})\ddot{q} + n(q,\dot{q}) = \varepsilon$$

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

As a consequence, the equation of motion becomes:

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

Where $h_{ijk} = \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}$ and $g_i(q) = -\sum_{j=1}^{n}\left(m_{lj} g_0^T J_{Pi}^{(l_j)}(q) + m_{m_j} g_0^T J_{Pi}^{(m_j)}(q)\right)$

Each term of the equation of motion has a physical interpretation. As regards to the non-conservative ε_i forces doing work at the joints, they are given by:

Non-conservative forces = actuated joint torque τ- viscous friction torque $F_v\dot{q}$ - static friction torque $F_s sgn(\dot{q})$ - actuated torques to balance external contact forces $J^T(q)h$

To summarize, the equation of motion can be written in matrix compact in "the joint space dynamical model" as:

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

Where $C(\mathrm{q},\dot{q})$ matrix represents the Centrifugal Effect induced on joint i by the velocity of joint j and the Coriolis Effect induced on joint i by the velocities of joints j and k. $C(\mathrm{q},\dot{q})$ is a (n*n) matrix whose elements are:

$$c_{ij} = \sum_{k=1}^{n} c_{ijk} q_k \text{ where } c_{ijk} = \left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)$$

b_{ij}: are the elements of the inertia matrix $B(\mathrm{q})$.

In order to find the dynamical model of each joint of the machine, MATLAB was used to perform all the calculations and find the matrices. The matrix $C(\mathrm{q},\dot{q})$ obtained is:

$$C(\mathrm{q},\dot{q}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

From the form of the $C(\mathrm{q},\dot{q})$ matrix, it can be seen that there are no coriolis and centrifugal forces. This is true because the first three axes of the machine are translational and only the last one is revolute.

The matrix of viscous friction coefficients F_v is:

$$F_v = \begin{bmatrix} F_{m1} * Kr_1^2 & 0 & 0 & 0 \\ 0 & F_{m2} * Kr_2^2 & 0 & 0 \\ 0 & 0 & F_{m3} * Kr_3^2 & 0 \\ 0 & 0 & 0 & F_{m4} * Kr_3^2 \end{bmatrix}$$

The vector $g(\mathrm{q})$ is the following:

$$g(\mathrm{q}) = \begin{bmatrix} g * (ml_1 + ml_2 + ml_3 + ml_4) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The equation of motion obtained for each joint is as follows:

$$\tau_1 = (I_{m1} * kr_1^2 + ml_1 + ml_2 + ml_3 + ml_4)\ddot{q}_1 + g * (ml_1 + ml_2 + ml_3 + ml_4) + F_{m1} * kr_1^2 * \dot{q}_1$$

$$\tau_2 = (I_{m2} * kr_2^2 + ml_2 + ml_3 + ml_4)\ddot{q}_2 + F_{m2} * kr_2^2 * \dot{q}_2$$

$$\tau_3 = (I_{m3} * kr_3^2 + ml_3 + ml_4)\ddot{q_3} + F_{m3} * kr_3^2 * \dot{q}_3$$

$$\tau_4 = (I_{m4} * kr_4^2 + I_{l4})\ddot{q_4} + F_{m4} * kr_4^2 * \dot{q}_4$$

4.2 Dynamic Parameters Identification

The information provided by robot makers concerning the dynamic parameters of robotic systems (the inertial properties of the links and friction parameters at the kinematic joints) is limited and even nonexistent. As an example, friction parameters are usually not provided. Thus, it's necessary to develop efficient procedures for their measure. The direct measurement of those parameters isn't easy since it might imply disassembling the robot [12]. So, it is needed to find methods to identify them because they are essential for solving simulation and control problems demands. It is not straightforward to compute such parameters from the design data of the mechanical structure.

There are a lot of methods to discover them like “CAD displaying technique” which permits the computation of the values of the inertial parameters of the different components (links, actuators and transmissions) on the premise of their geometry and sort of materials utilized.
But, the disadvantage of this method is that some components could not be fully modeled in detail and parameters that depend on operational conditions, like friction, could not be determined. In any case, the estimates obtained by such procedures are imprecise due to the simplification in the geometric modelling.
Another strategy that can be utilized is “A heuristic approach” that might be to disassemble the different components of the manipulator and perform a series of estimations to evaluate the inertial parameters. Such strategy isn't simple to execute and may be troublesome to measure the significant quantities. So to find precise estimates of dynamic parameters, it is worth resorting to “identification techniques”.
Before resorting to such techniques, the dynamical model of the machine should be linearized. The next two paragraphs contain the linearization of the dynamical model of the machine and the procedure to identify the parameters using Least Square technic.

4.2.1 Linearization of the Dynamical Model

The Lagrangian can be rewritten in linear form as:

$$L = \sum_{i=1}^{n} (\beta_{T_i}^{T} - \beta_{U_i}^{T})\pi_i$$

π_i is a (11*1) vector of dynamic parameters. β_{T_i} and β_{U_i} are functions of the generalized coordinates of the mechanical system and their derivatives ($\beta_{T_i}(q_1, q_2, \dots, q_i, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_i)$ and $\beta_{U_i}(q_1, q_2, \dots, q_i)$). Performing the derivatives does not alter the linearity properties:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \varepsilon_i \qquad => \qquad \varepsilon_i = \sum_{j=1}^{n} y_{ij}^{T}\pi_j$$

$$\text{With } y_{ij}^{T} = \frac{d}{dt}\frac{\partial \beta_{T_j}}{\partial \dot{q}_i} - \frac{\partial \beta_{T_j}}{\partial q_i} + \frac{\partial \beta_{U_j}}{\partial q_i}$$

Therefore, the following result is obtained:

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} = \begin{bmatrix} y_{11}^{T} & y_{12}^{T} & \cdots & y_{1n}^{T} \\ 0^{T} & y_{22}^{T} & \cdots & y_{2n}^{T} \\ \vdots & \vdots & \ddots & \vdots \\ 0^{T} & 0^{T} & \cdots & y_{nn}^{T} \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix}$$

The final linear expression can be written in compact form as:

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

Where $Y(q, \dot{q}, \ddot{q})$ is the regression matrix of dimension (n*p) called the regressor and π is a (p*1) vector of constant parameters.

Based on the linear expression obtained, the dynamical model of our machine can be written in this form: $\tau = Y(q, \dot{q}, \ddot{q})\pi$ where the elements of the regressor matrix $Y(q, \dot{q}, \ddot{q})$ and the elements of the vector π are as follows:

$\pi_1 = I_{m1} * kr_1^2 + ml_1 + ml_2 + ml_3 + ml_4$

$\pi_2 = F_{m1} * kr_1^2$

$\pi_3 = g * (ml_1 + ml_2 + ml_3 + ml_4)$

$$\pi_4 = I_{m2} * kr_2^2$$

$$\pi_5 = F_{m2} * k_{r2}^2$$

$$\pi_6 = ml_2 + ml_3 + ml_4$$

$$\pi_7 = I_{m3} * kr_3^2$$

$$\pi_8 = F_{m3} * {k_{r}}_3^{\,2}$$

$$\pi_9 = ml_3 + ml_4$$

$$\pi_{10} = I_{m4} * kr_4^2 + I_{l4}$$

$$\pi_{11} = F_{m4} * kr_4^2$$

$$y_{1,1} = \ddot{q}_1$$

$$y_{1,2} = \dot{q}_1$$

$$y_{1,3} = 1$$

$$y_{1,4} = y_{1,5} = y_{1,6} = y_{1,7} = y_{1,8} = y_{1,9} = y_{1,10} = y_{1,11} = 0$$

$$y_{2,1} = y_{2,2} = y_{2,3} = y_{2,7} = y_{2,8} = y_{2,9} = y_{2,10} = y_{2,11} = 0$$

$$y_{2,4} = y_{2,6} = \ddot{q}_2$$

$$y_{2,5} = \dot{q}_2$$

$$y_{3,1} = y_{3,2} = y_{3,3} = y_{3,4} = y_{3,5} = y_{3,6} = y_{3,10} = y_{3,11} = 0$$

$$y_{3,7} = y_{3,9} = \ddot{q}_3$$

$$y_{3,8} = \dot{q}_3$$

$$y_{4,1} = y_{4,2} = y_{4,3} = y_{4,4} = y_{4,5} = y_{4,6} = y_{4,7} = y_{4,8} = y_{4,9} = 0$$

$$y_{4,10} = \ddot{q}_4$$

$$y_{4,11} = \dot{q}_4$$

4.2.2 Least Square Technic

Least squares methodology, additionally referred to as least squares approximation, is a technique for estimating the true value of some quantity based on a consideration of errors in observations or measurements [13]. It is a form of mathematical regression analysis utilized to decide the line of best fit for a set of data, giving a visual demonstration of the relationship between the data points. Each point of data represents the relationship between a known independent variable and an unknown dependent variable [7].

On the assumption that the kinematic parameters are known, for example the joint positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$ can be measured. The torque also can be measured using sensors (but because I do not have access to the machine, I tried to do simulations to find such values). If the measurements of joint torques, positions, velocities and accelerations have been obtained at given time instants $t_1 \ldots t_N$, it can be written:

$$\bar{\tau} = \begin{bmatrix} \tau(t_1) \\ \vdots \\ \tau(t_N) \end{bmatrix} = \begin{bmatrix} Y(t_1) \\ \vdots \\ Y(t_N) \end{bmatrix} \pi = \bar{Y}\pi$$

Where the matrix $\bar{Y}$ is a matrix whose elements are the positions, velocities and accelerations of each joint. π is a vector whose elements are the dynamical parameters to be estimated and $\bar{\tau}$ is a vector of the joint torques.

In order to obtain good numerical conditioning, the number of time moment that should be considered should be high enough. In this way, solving the equation using least square leads to solution in the form:

$$\pi = (\overline{Y^T}\,\overline{Y})^{-1}\bar{Y}^T\bar{\tau}$$

Where $(\overline{Y^T}\,\overline{Y})^{-1}\bar{Y}^T$ is the left pseudo-inverse of matrix$\bar{Y}$.

CHAPTER 5: CONTROL

The primal objective of a control system is that of making a dynamic system to act in an ideal way. The design and plan of such a control system to give a requested behavior is normally done by utilizing a mathematical model of the dynamic system. This model is picked to address the major dynamical features of the process. For the reason that the mathematical model is an idealization of the real process, it is inaccurate and this inaccuracy involves the presence of model uncertainty [8].

In this chapter two types of control are presented. The first type of control is the “centralized control”. In particular, “robust control” is presented. This type of control has many advantages. The first one is its capacity to drop the measured and unmeasured disturbances. The second advantage of this type of control is its ability to deal with uncertain model parameters and external disturbances. The last advantage is that prior knowledge of uncertain inputs is not required [9]. Therefore, due to the ability of this type of control to deal with uncertain model parameters, it is chosen to control the machine. The uncertainty in the dynamical parameters is present because all the parameters are estimated using least square technic.

The second type of control that is discussed in this chapter is the “decentralized control”. This is the simplest type of control of a manipulator because each joint axis is controlled independently from the other.

5.1 Centralized Control

“Robust control” is a critical branch of control theory that explicitly deals with uncertainty in its control design. Over the past few decades, there has been an increasing interest in robust control. On one hand, uncertain parameters or disturbances exist in advanced manufacturing, and other complex engineering systems. On the other hand, uncertain parameters or disturbances seriously influence the stability, accuracy, and reliability of control systems. Hence, robust control has become a challenging problem in international control field [5]. Therefore, robust control is chosen to control the machine due to the presence of uncertainty in the dynamical parameters. Uncertainty exists because all the masses, inertias … etc. are estimated using least square technic presented in the previous chapter.

This type of control is not only based on a linear PD control but also on a design of an outer loop on the error that should be robust to the uncertainty in the estimates. The control law is:

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

Where the PD term ensures stabilization of the error dynamic system matrix. $\ddot{q}_d$ provides a feed forward term. The term w has to be chosen to guarantee robustness to the effects of uncertainty in the estimates. If the uncertainty term w disappears, so w becomes equal to zero, and the above control returns to a normal PD controller. The block scheme of the control is shown in the figure (16).

The block scheme presented in figure (16) contains four main blocks. The first block “Stabilizing Linear Control” and the second block “robustness” contain the controller and provide as output the control input y to the “Non Linear Compensation and Decoupling” block. The term $\ddot{q}_d + K_D\dot{\tilde{q}} + K_p\tilde{q}$ introduces a linear feedfroward action and a linear feedback action which stabilizes the error system dynamics.

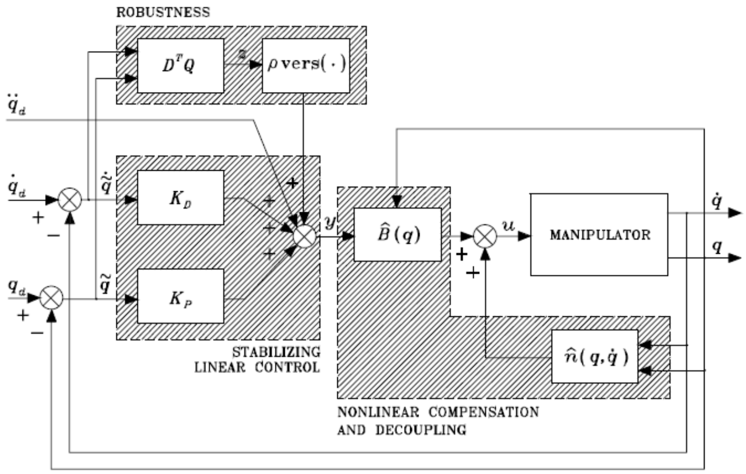

Figure 16: The block scheme of the robust control in the joint space

The term $\hat{B}(q) + \hat{n}(q, \dot{q})$ in the "Non Linear Compensation and Decoupling" block ensures an approximate compensation of nonlinear effects and joint decoupling. The last "manipulator" block contains the dynamical model of the machine.

The matrices K_D and K_P should be diagonal matrices of the form:

$K_D = diag\{2\varepsilon_1 w_{n_1}, \dots, 2\varepsilon_n w_{n_n}\}$ and $K_p = diag\{w_{n_1}^2, \dots, w_{n_n}^2\}$

The term w that represents the robust contribution that counter act the uncertainty in the dynamical parameters should be as follows:

$$w = \begin{cases} \frac{\rho}{||z||} z, & ||z|| \geq \epsilon \\ \frac{\rho}{\epsilon} z, & ||z|| < \epsilon \end{cases}$$

Where $z = D^T Q \varepsilon$ where D is a block matrix of dimension (2n*n), Q is a (2n*2n) positive definite matrix and ε represents the width of the boundary layer.

The matrix D has to be chosen like this: $D = \begin{bmatrix} O \\ I \end{bmatrix}$

A great value of ρ leads to saturation of the input and consequently bad behavior of the whole system while a reasonable value gives a reasonable output values. Therefore, the value of ρ should be chosen carefully when designing the control [11].

The Elimination of high-frequency components can be achieved by adopting this robust control law. But the limitation of this type of control is that error convergence to zero is not guaranteed, it only ensures bounded-norm errors.

The block scheme developed on MATLAB to perform this type of control of our machine is shown below:

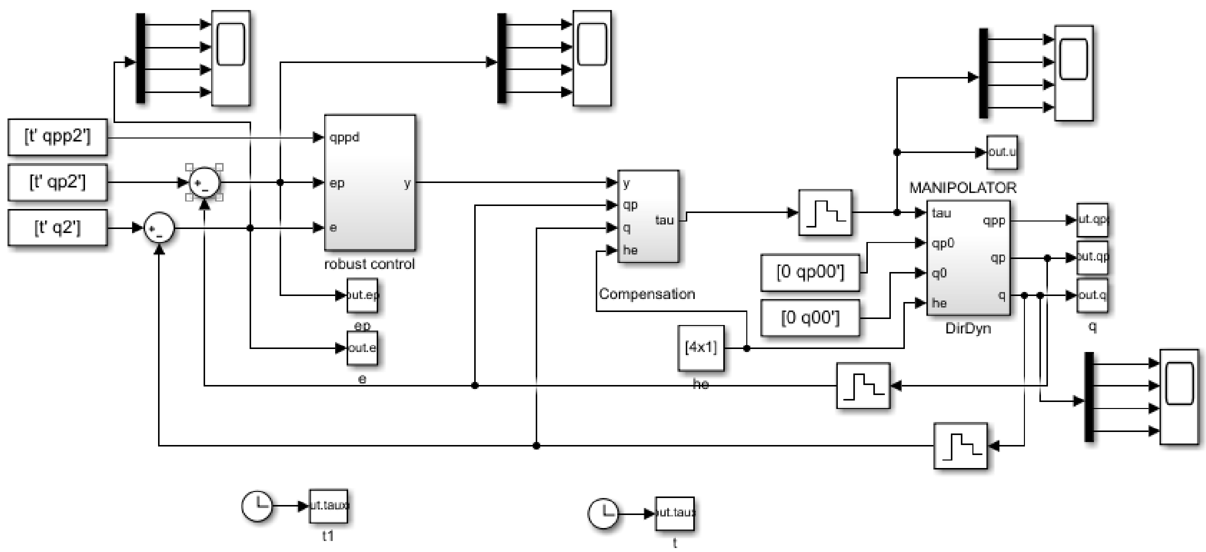

Figure 17: The block scheme of robust control implemented on MATLAB

The "robust control" block contains the control law. The parameters K_P, K_D, D, Q, ρ and ϵ are chosen as follows:

$K_P = diag([100\ 10\ 2\ 100])$

$K_D = diag([300\ 100\ 30\ 100])$

$D = [zeros(4,4); eye(4,4)]$

$Q = [diag([10\ 10\ 10\ 10]);\ \ diag([10\ 10\ 10\ 10]); K_P * 10\ \ \ K_D];$

$\rho = 10;$

$\epsilon = 1;$

The "Compensation" block contains the approximated dynamical model which is chosen as:

$$u = \bar{B}y + \bar{g}$$

Where $\bar{B}$ is a constant diagonal matrix relative to the average inertias of the axes of each joint. The diagonal elements of the matrix $\bar{B}$ are:

$$\bar{b}_{11} = I_{m1} * kr_1^2 + ml_1 + ml_2 + ml_3 + ml_4$$

$$\bar{b}_{22} = I_{m2} * kr_2^2 + ml_2 + ml_3 + ml_4$$

$$\bar{b}_{33} = I_{m3} * kr_3^2 + ml_3 + ml_4$$

$$\bar{b}_{44} = I_{m4} * kr_4^2 + I_{l4}$$

$\bar{g}$ vector is chosen equal to: $\bar{g} = \begin{bmatrix} g * (ml_1 + ml_2 + ml_3 + ml_4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

The DirDyn block contains the dynamical of the machine found in the previous chapter who was:

$$\tau_1 = (I_{m1} * kr_1^2 + ml_1 + ml_2 + ml_3 + ml_4)\ddot{q}_1 + g * (ml_1 + ml_2 + ml_3 + ml_4) + F_{m1} * kr_1^2 * \dot{q}_1$$

$$\tau_2 = (I_{m2} * kr_2^2 + ml_2 + ml_3 + ml_4)\ddot{q}_2 + F_{m2} * kr_2^2 * \dot{q}_2$$

$$\tau_3 = (I_{m3} * kr_3^2 + ml_3 + ml_4)\ddot{q}_3 + F_{m3} * kr_3^2 * \dot{q}_3$$

$$\tau_4 = (I_{m4} * kr_4^2 + I_{l4})\ddot{q}_4 + F_{m4} * kr_4^2 * \dot{q}_4$$

5.2 Decentralized Control

Decentralized control is the simplest type of control that can be chosen to control a manipulator. In this type of control system, the plant is not controlled by a single controller. Each joint is controlled independently from the others and all the coupling effects between the joints are considered as disturbances. This control is based on the error between the desired and actual output. The general block scheme describing this control is shown in the figure below:

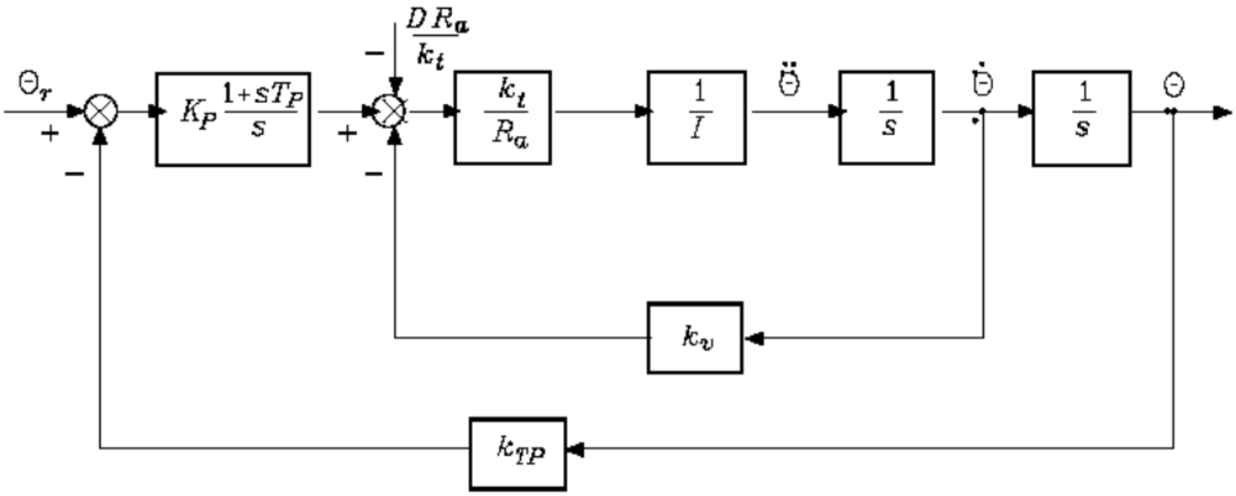

Figure 18: The block scheme of the decentralized control of each joint

The block containing the term $K_p \frac{1+sT_P}{s}$ represents the transfer function of a PI controller. A PI controller is chosen for two reasons. The first one is that a large value of amplifier gain K_p before the point of intervention of the disturbance guarantees an effective rejection of the disturbance on the output. The second reason is that the integral term in the controller cancels the effect of the gravitational component on the output at steady-state.

The term $\frac{DR_a}{K_t}$ denotes the disturbance due to the coupling between the joints. R_a is the armature resistance of the motor actuating the joint and k_t is the torque constant which is numerically equal to K_v.

K_{TP} represents the transducer that measures the output position θ.

The transfer function of the motor is:

$$M(s) = \frac{\frac{K_t}{R_a I\, S}}{1+K_v\frac{K_t}{R_a I s}}\frac{1}{S} = \frac{1}{K_v}\frac{1}{S(1+\frac{R_a I}{K_t K_v}S)} = \frac{K_m}{S(1+T_m S)}.$$

With $K_m = \frac{1}{K_v}$ and $T_M = \frac{R_a I}{K_t K_v}$.

I is the equivalent average inertia at the motor shaft. K_v is the constant voltage term that depends on the construction details of the motor as well as on the magnetic flux of the coil.

This type of control is dedicated to the control of the positions of the joints. Each joint position is controlled alone. The block scheme shown in figure (18) is called "Position Feedback" control.

In order to know the precise values of the control parameters that shall be chosen, a root locus analysis can be performed as a function of the position loop: $\frac{K_m K_p K_{TP} T_p}{T_M}$.

Three situations can be illustrated for the poles of the closed-loop system with reference to the relation between T_M and T_P:

1. If $T_P < T_M$, the system is intrinsically unstable. The root locus are as shown in the figure below:

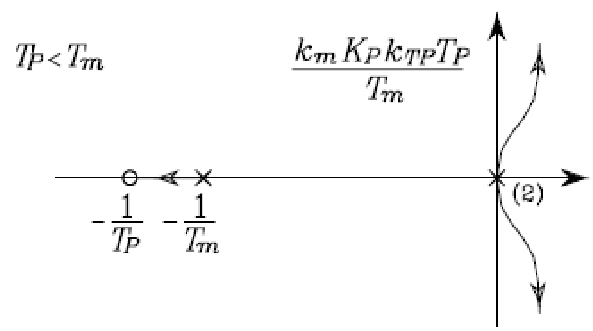

Figure 19: The root locus in the case $T_P < T_M$

2. If $T_P > T_M$, the system is stable. The root locus are as follows:

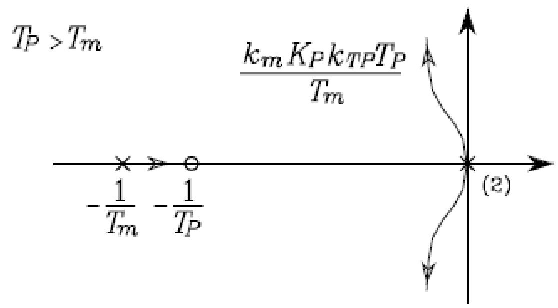

Figure 20: The root locus in the case $T_P > T_M$

3. If $T_P \gg T_M$, the system remarkably improves its swiftness features. The root locus are like these:

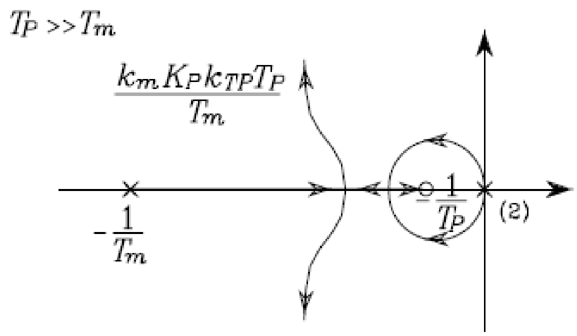

Figure 21: The root locus in the case $T_P \gg T_M$

The control scheme developed on MATLAB to perform the control of each joint is shown below:

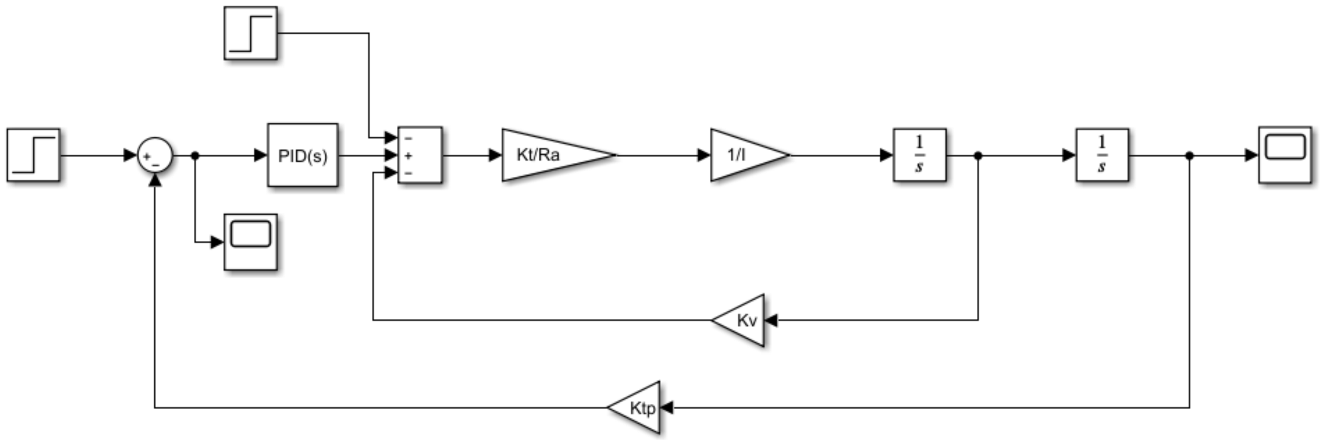

Figure 22: The block scheme of the decentralized control of each joint implemented on MATLAB

In order to guarantee the stability and error convergence to zero in our machine model, the control parameters for each joint are chosen as follows:

- For each joint, the $T_M = \frac{R_a I}{K_t K_v}$ term is calculated. After performing the computation, the T_P term of the PI controller is chosen greater than T_M ($T_P > T_M$). This choice guarantees stability of the system.
- The term K_p for each joint controller is chosen greater enough to reject the effect of the disturbances.

CHAPTER 6: TESTS

Several control tests are performed. Each test is done in specific conditions. These tests are performed in order to conclude which control is more appropriate for the machine action.

6.1 Tests on the Centralized Control

For this control type, three tests have been performed. The first test is done in nominal conditions. The second test is done in the presence of high uncertainty in the dynamical parameters. The last one is performed with low uncertainty in the dynamical parameters where some dynamical parameters have values different from their real ones.

6.1.1 Test in nominal conditions

This test is done in nominal conditions. The value of the uncertainty is chosen as nominal. So, the value of the ρ value was equal to one. The values of the position, velocity and acceleration errors of each joint are shown in the figures below:

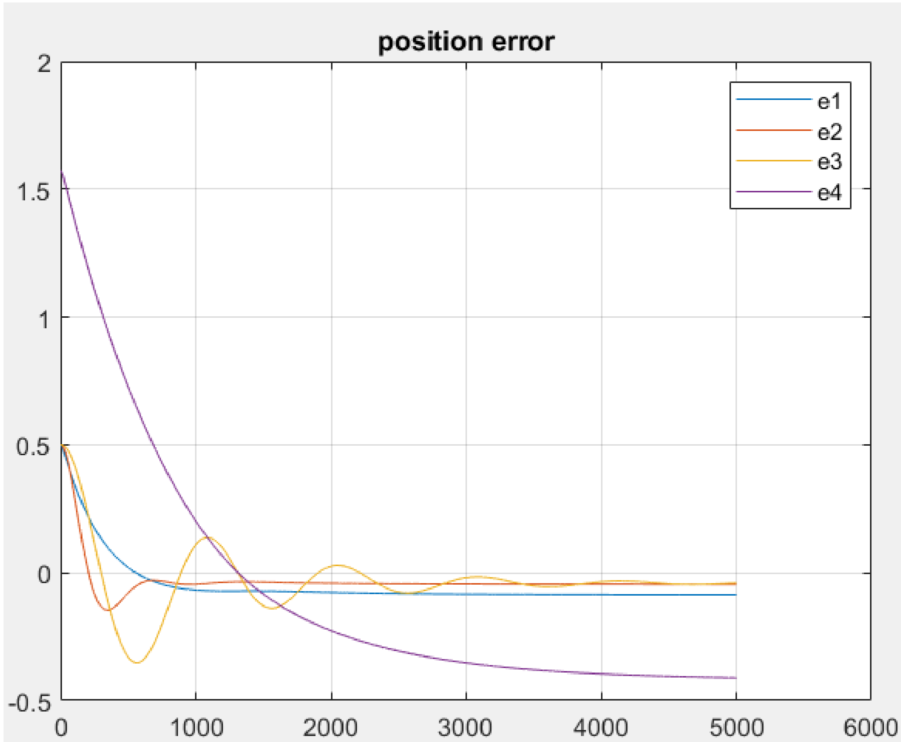

Figure 23: The position errors of all the joints in nominal test condition

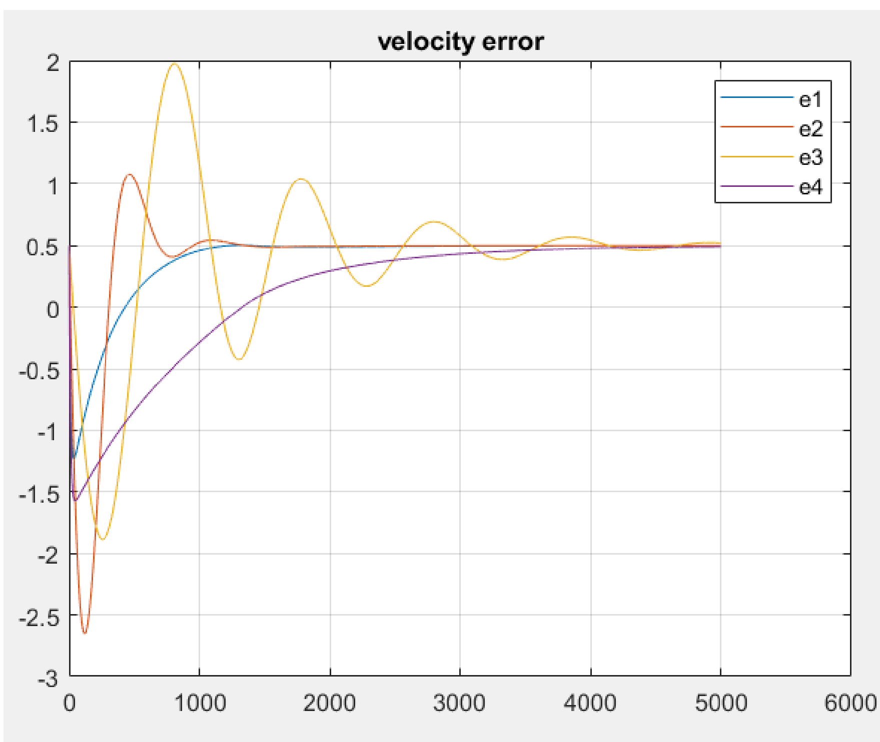

Figure 24: The velocities errors of all the joints in nominal test condition

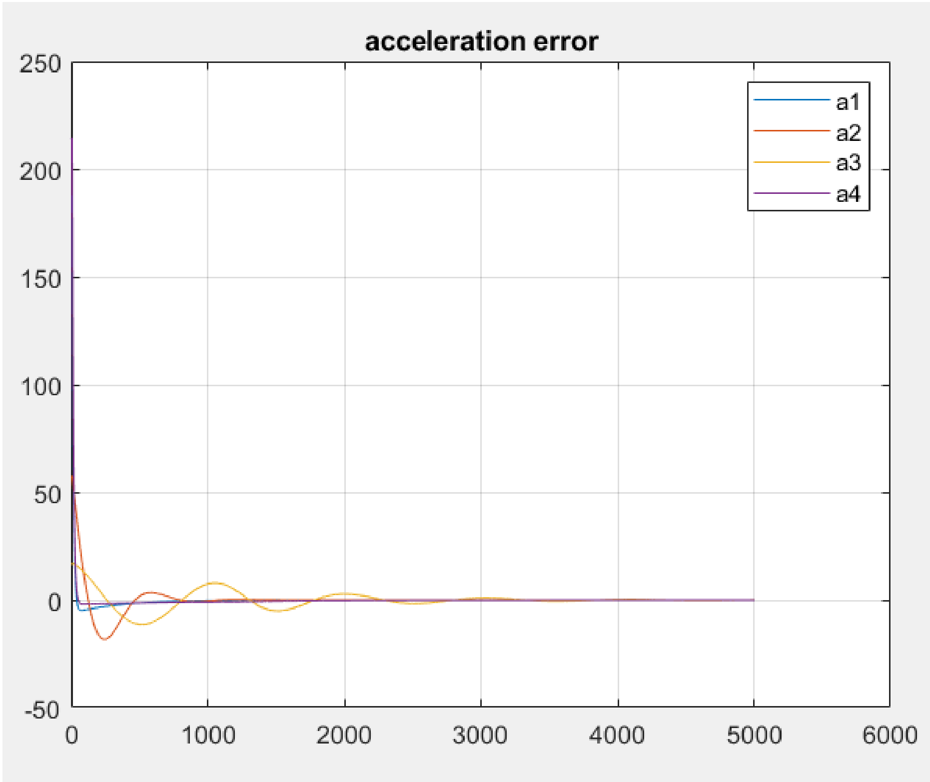

Figure 25: The acceleration errors for all the joints in nominal test conditions

From the graphs, it can be seen that the error values are good because all the errors tends to zero or closely to zero after a certain time. So, it can be seen that in nominal conditions the controller works efficiently.

6.1.2 Test with high uncertainty

In this test, the uncertainty is high. The ρ value is chosen high different from one. In that case, it should be seen bad behaviors of the whole system because the robust control works properly with bounded norm uncertainty. Looking at the graphs of the velocity and acceleration errors in figures (27) and (28), the bad behavior of the system can be seen. The errors are oscillating a lot and they did not tend to zero.

Figure 26: The position errors of all the joints in high uncertainty test condition

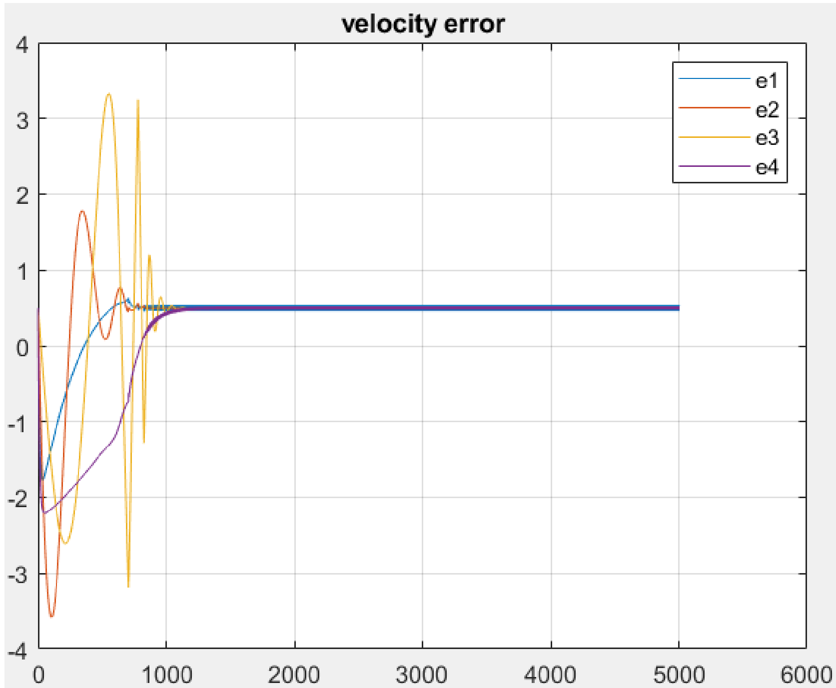

Figure 27: The velocity errors of all the joints in high uncertainty test condition

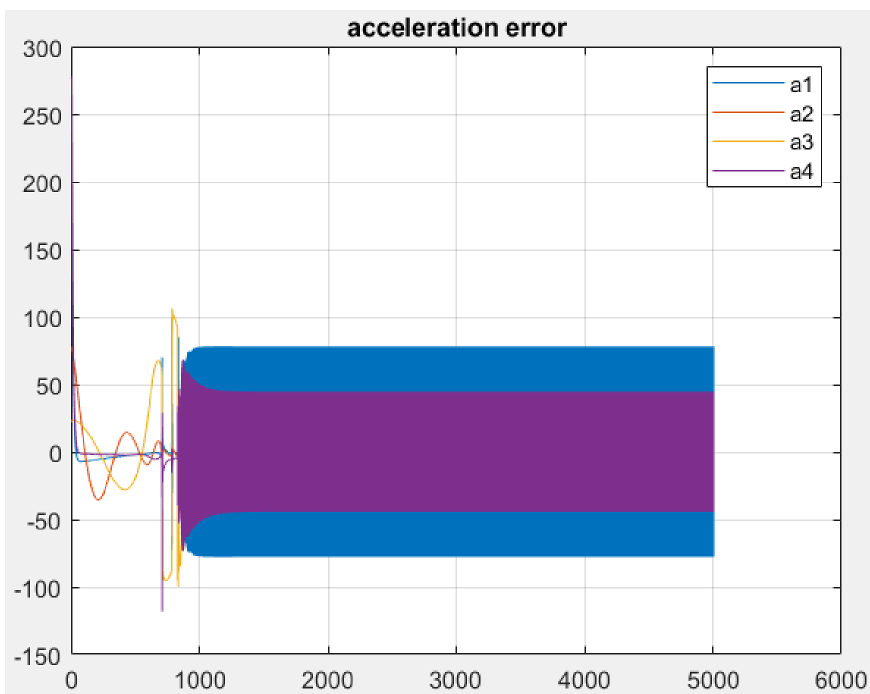

Figure 28: The acceleration errors of all the joints in high uncertainty test condition

6.1.3 Test with low uncertainty

This control test is done with low uncertainty. In this test, the ρ value was equal to one (as in the nominal conditions). Comparing the graphs of the position, velocity and accelerations errors shown in figures (29), (30) and (31) in this control test to the one in nominal conditions, it can be seen that they are higher than the ones in nominal conditions. This is for sure true because in this test the dynamics of the system is different from the real One.

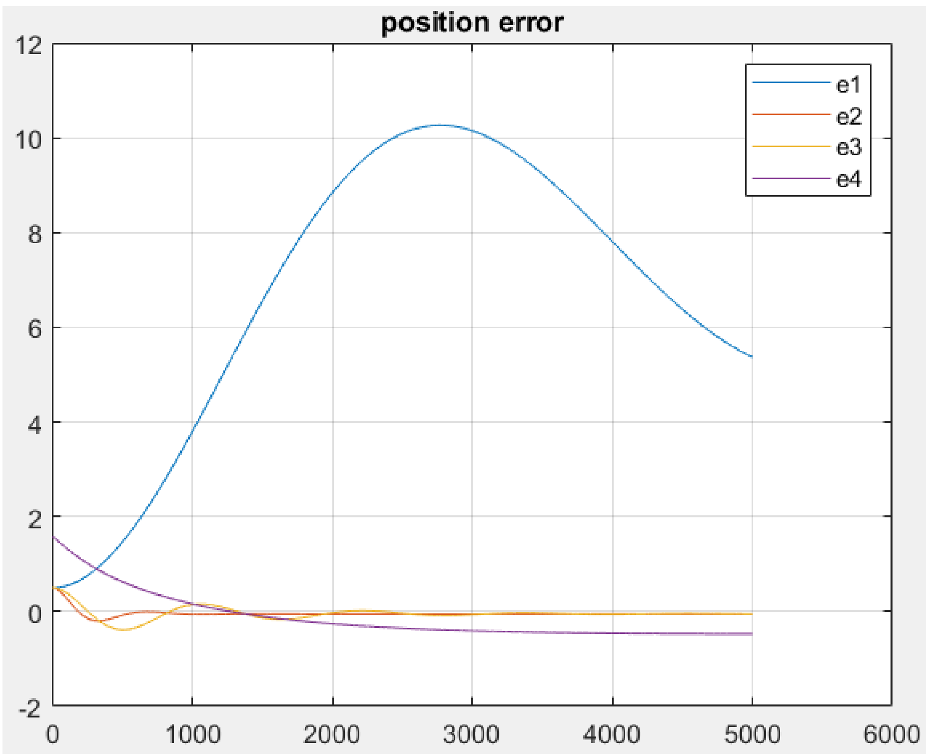

Figure 29: The position errors of all the joints in low uncertainty test condition

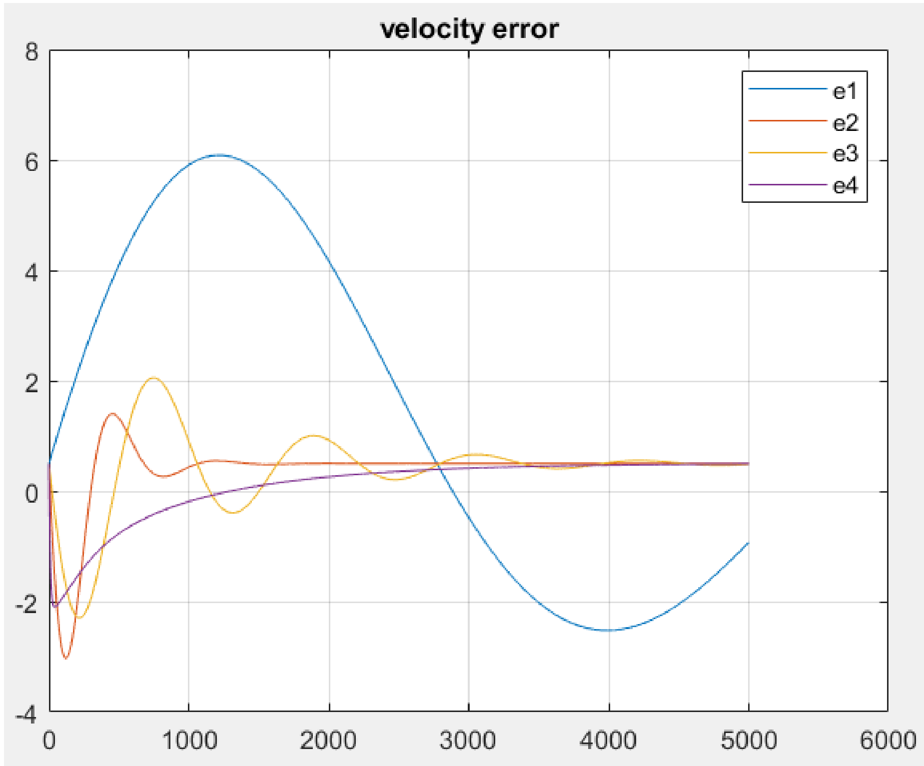

Figure 30: The velocity errors of all the joints in low uncertainty test condition

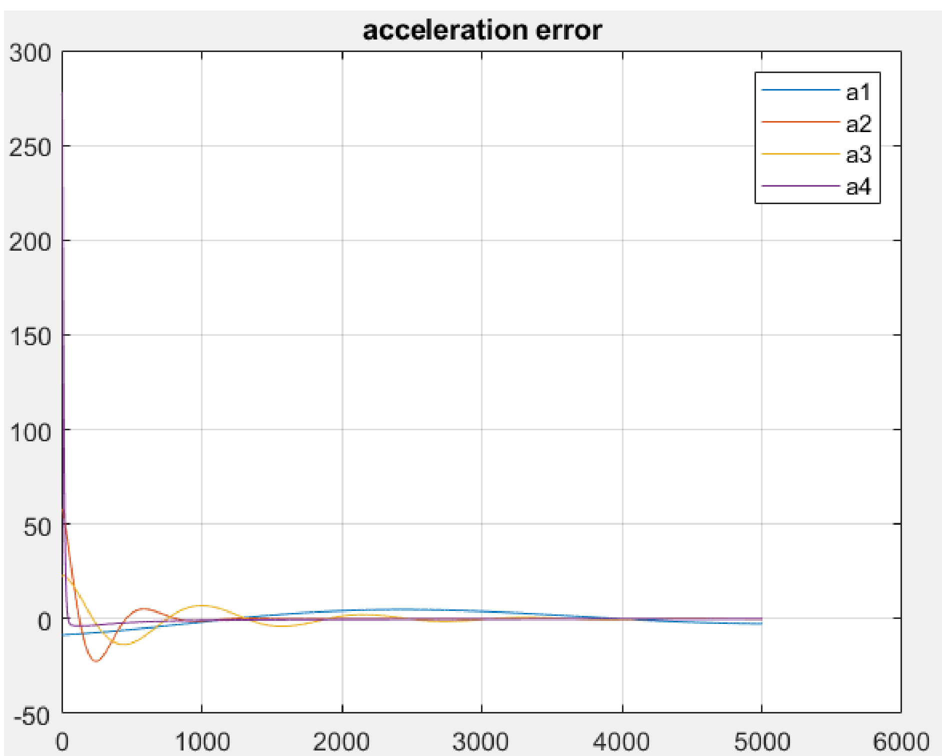

Figure 31: The acceleration errors of all the joints in low uncertainty test condition

6.2 Tests on the Decentralized Control

In this control type, two control tests has been done. One test is done in nominal conditions and the other one in the absence of disturbances.

6.2.1 Test in nominal conditions

In this test, all the position errors for the four joints are close to zero. The value of the position error for each joint is presented in the figure shown below:

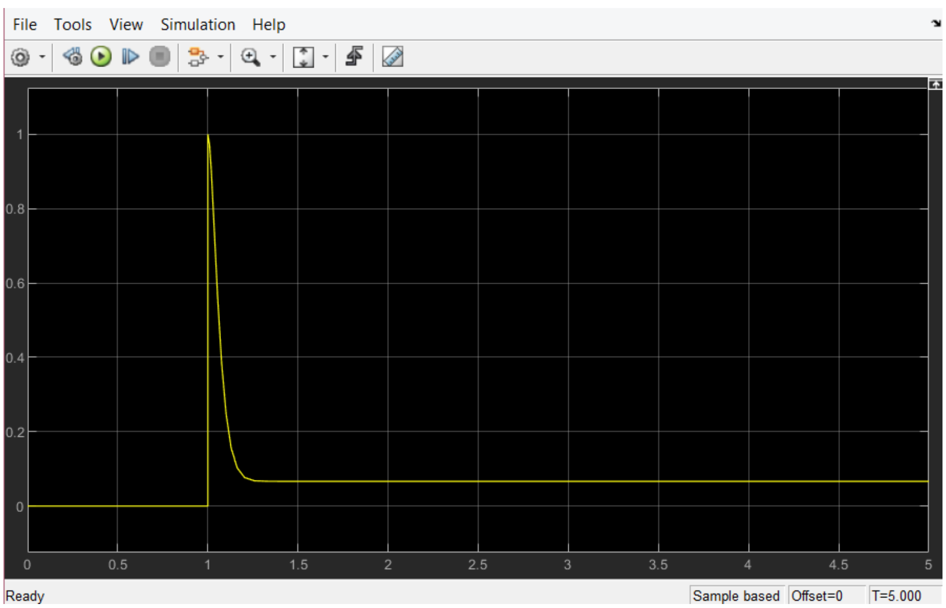

Figure 32: The position error for the first prismatic joint in nominal test condition

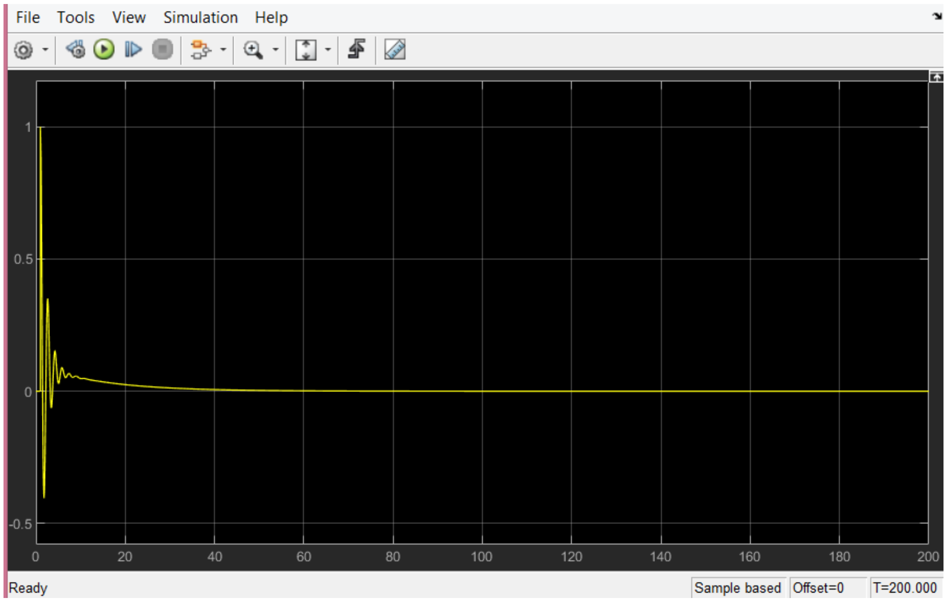

Figure 33: The position error for the second prismatic joint in nominal test condition

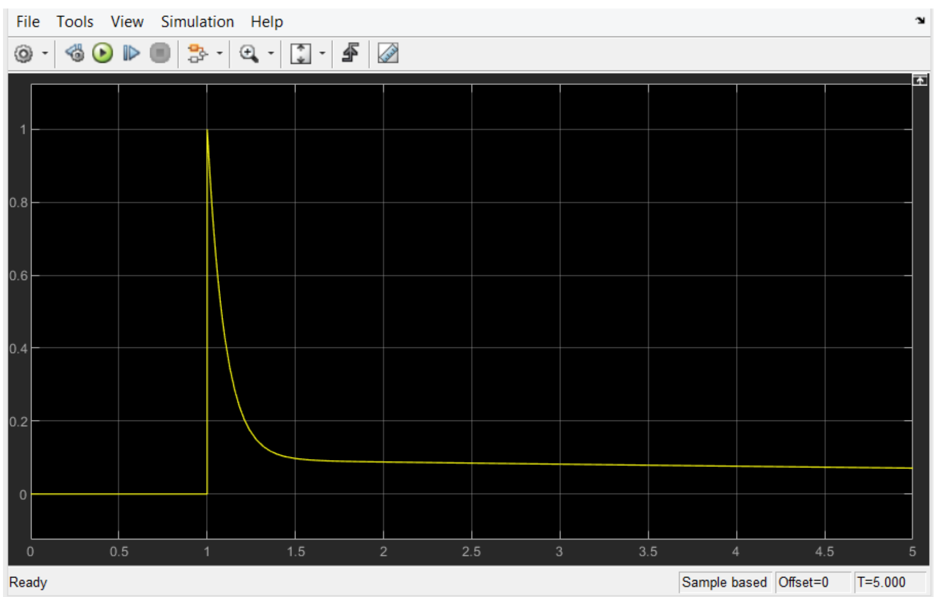

Figure 34: The position error for the third prismatic joint in nominal test condition

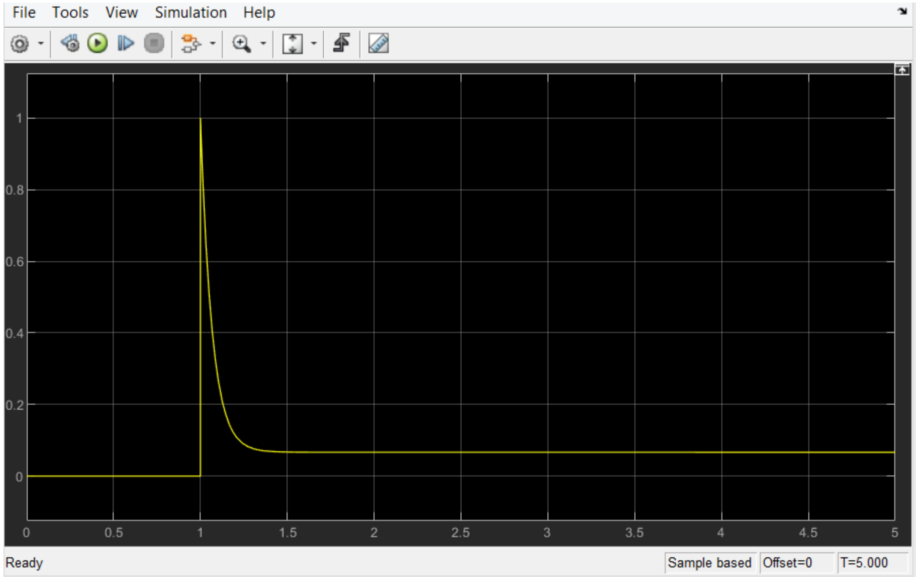

Figure 35: The position error for the fourth revolute joint in nominal test condition

6.2.2 Test in the absence of disturbances

In this control test, the disturbances affecting each joint are considered to null. All the position errors tends to zero. This is true because in that case the system is not affected by any disturbance. The position error for each joint is shown in the figures below:

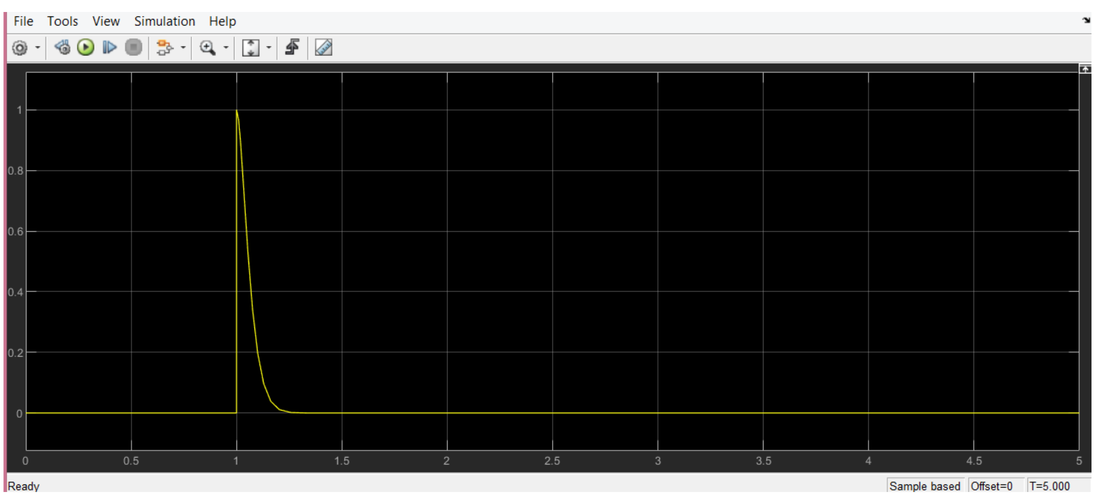

Figure 36: The position error for the first prismatic joint in the absence of disturbances

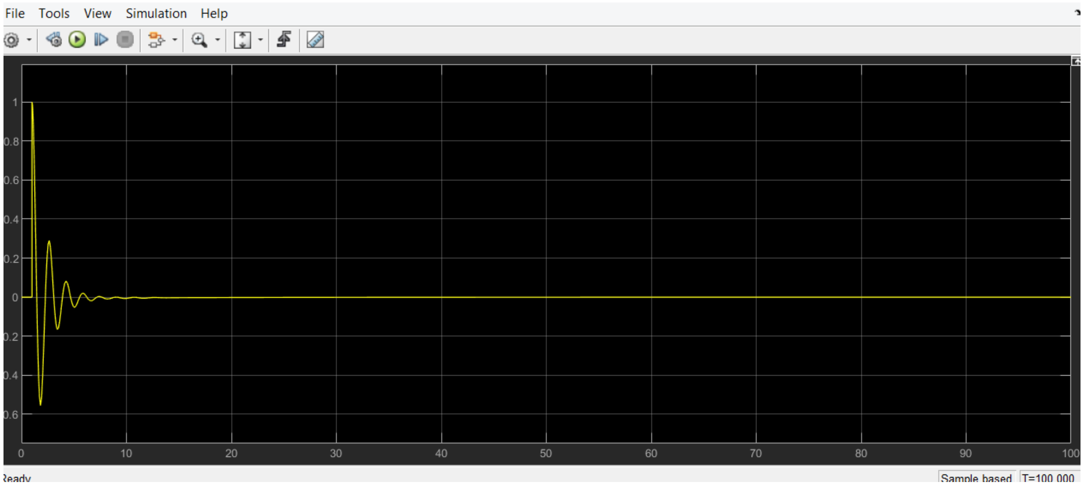

Figure 37: The position error for the second prismatic joint in the absence of disturbances

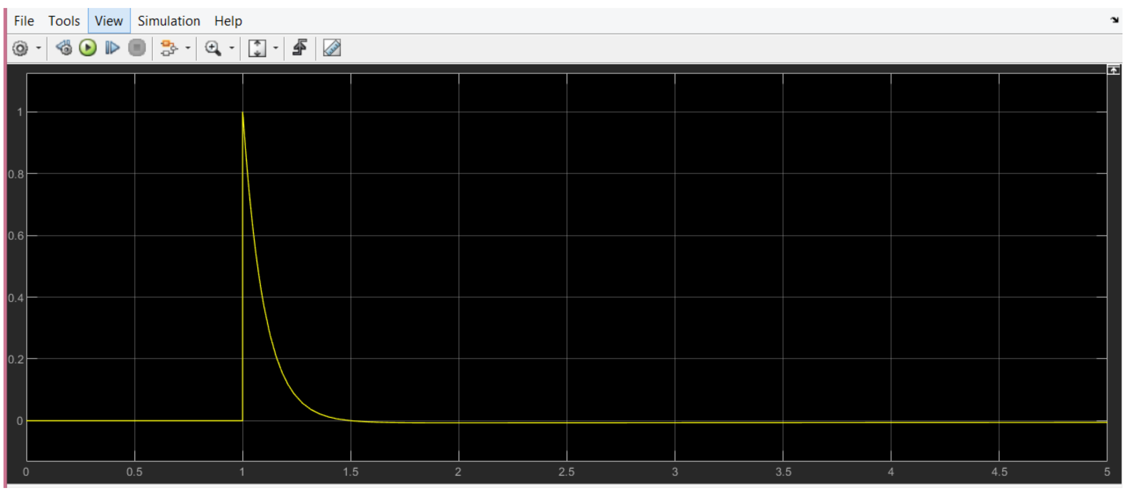

Figure 38: The position error for the third prismatic joint in the absence of disturbances

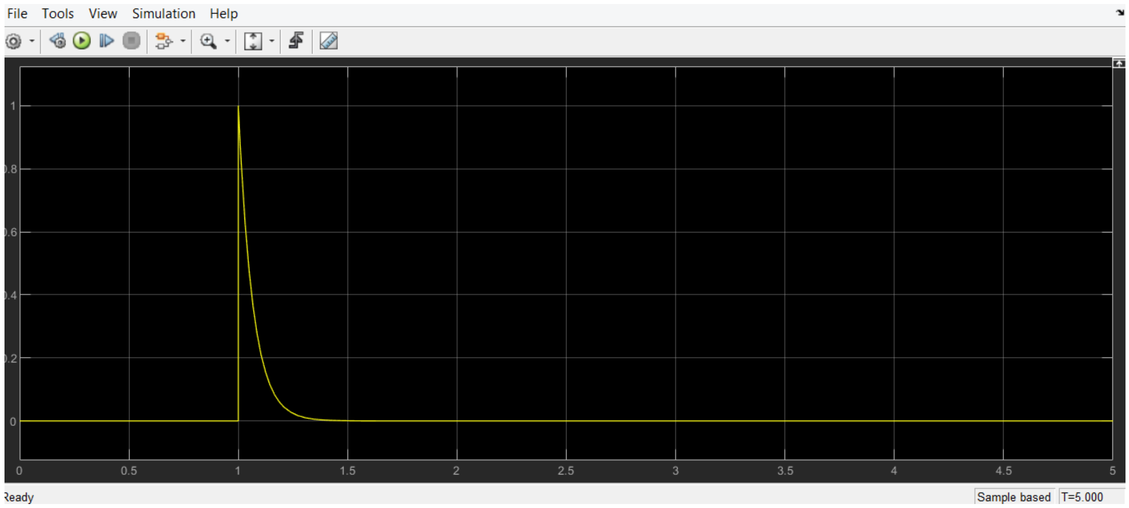

Figure 39: The position error for the fourth revolute joint in the absence of disturbances

6.3 Comparison of the Two Controls

From the centralized and decentralized control tests in nominal conditions, it can be concluded which control type is more appropriate to control the machine. Comparing the positions errors for the centralized and decentralized control, one can decide which control type is the best for the machine. The position errors in the centralized control in nominal conditions does not tends to zero. The errors for the first three joints are close to zero while the error for the fourth joint is slightly high (as it can be seen in figure (23)). In the decentralized control, all the positions errors in nominal conditions tend to a value close to zero. Therefore, it may be concluded that the decentralized control is more appropriate to control the machine where each joint in the machine is controlled separately by a PI controller. This PI controller is characterized by a large gain and an integral action able to cancel the effects of the disturbances and the gravity at the output.

CHAPTER 7: CONCLUSION

In summary, we can define the steps followed for the achievement of an appropriate control of the CNC machine for the MOREPRO project. These steps are the following:

1. Analysis of the machine kinematics:
    a. The kinematical model of the machine is found.
2. Finding the machine dynamics and estimating the dynamical parameters:
    a. Finding the dynamics of the machine in the joint space using Lagrange formulation.
    b. Linearizing the dynamical model.
    c. Using the linearized model and least square technic to estimate the dynamical parameters of the machine.
3. Designing two types of control systems:
    a. The first one is the centralized robust control where the whole machine is controlled based on a single controller.
    b. The second one is the decentralized control where each joint is controlled independently from the others and the coupling between the joints is considered as disturbances into the system.
4. Performing several control tests:
    a. In the centralized robust control, three tests have been performed, one in nominal conditions, one in the presence of high uncertainty in the system and the other in the presence of low uncertainty.
    b. In the decentralized control, two test have been performed, one in nominal conditions and the other in the absence of disturbances.
5. Comparing the results of the two control tests in nominal conditions in order to choose the appropriate control type to controls the machine action.

At the end, from the control tests performed, it can be seen that the decentralized control is more appropriate to control the machine that we have.

BIBLIOGRAPHY

[1] Lydon, B. (2020, July 7). How Industry 4.0 and Digitization Improves Manufacturing Responsiveness, Quality, and Efficiency. *InTech magazine.* Retrieved from: https://blog.isa.org/industry-40-digitization-improve-manufacturing-responsiveness-quality-efficiency-iot

[2] Bartevyan, L. (2016, July). Industry 4.0 – Summary report. *DLG Expert report 7/2016.* Retrieved from: https://www.dlg.org/fileadmin/downloads/lebensmittel/themen/publikationen/expertenwissen/lebensmitteltechnologie/e_2016_7_Expertenwissen_Industrie_4_0.pdf

[3] Deng, W. (2016, October). Industry 4.0: Digitization, sensorization and optimization. *tmforum.* Retrieved from: https://inform.tmforum.org/internet-of-everything/2016/10/industry-4-0-digitization-sensorization-optimization/

[4] Ledin, J. (2000, August 1). Modeling Dynamic Systems. *Embedded Systems Programming.* Retrieved from: https://www.embedded.com/modeling-dynamic-systems/

[5] Li, T. Zhang, B. Feng, Z. Zheng, B. (2014, January 2014). Robust Control with Engineering Applications. *Mathematical Problems in Engineering.* Retrieved from: https://www.hindawi.com/journals/mpe/2014/567672/

[6] Siciliano, B. Sciavicco, L. Villani, L. Oriolo, G. Robotics, Modelling, Planning and Control.doi:10.1007/978-1-84628-642-1.

[7] Kenton, W. Kindness, A. (2020, June 10). Least Square Method Definition. *Investopedia.* Retrieved from: https://www.investopedia.com/terms/l/least-squares-method.asp#:~:text=The%20least%20squares%20method%20is%20a%20statistical%20procedure%20to%20find,the%20behavior%20of%20dependent%20variables

[8] Ferre, C. (2003, May 5). Robust Control through Robustness Enhancement. *Univeritat Antonoma de Barcelona.* Retrieved from:

https://www.tdx.cat/bitstream/handle/10803/5340/cpf1de1.pdf?sequence=1&isAllowed=y

[9] Behrooz, F. Mariun, N. Marhaban, M. Radzi, M. Ramli, A. (2018, February 27). Review of Control Techniques for HVAC Systems—Nonlinearity Approaches Based on Fuzzy Cognitive Maps. *Energies.* Retrieved from: https://www.researchgate.net/publication/323441631_Review_of_Control_Techniques_for_HVAC_Systems-Nonlinearity_Approaches_Based_on_Fuzzy_Cognitive_Maps

[10] (2018, February 2018). Industry 4.0 and how smart sensors make the difference. *Spectral Engines, Member of the Nynomic Group.* Retrieved from: https://www.spectralengines.com/articles/industry-4-0-and-how-smart-sensors-make-the-difference

[11] Torres, S. Mendez, J.A. Acosta, L. Becerra, V.M. Adaptive Robust Controller for Robot Manipulators: Experiments on a PUMA 560 Robot. *IFAC Proceedings Volume.* Retrieved from: https://www.sciencedirect.com/science/article/pii/S1474667016385536

[12] Mata, V. Farhat, N. Rodriguez, M. Valera, A. Page, A. (2008, April 1). Parallel Manipulators, towards New Applications. doi: 10.5772/5424. Retrieved from: https://www.intechopen.com/books/parallel_manipulators_towards_new_applications/dynamic_parameter_identification_for_parallel_manipulators

[13] Routledge, R. Least squares method. *Britannica.* Retrieved from: https://www.britannica.com/topic/least-squares-approximation

[14] Khan, M. Islam, R. Iqbal, J. Control strategies for robotic manipulators. *Robotics and Artificial Intelligence (ICRAI), 2012 International Conference.* Retrieved from: https://www.researchgate.net/publication/261451087_Control_strategies_for_robotic_manipulators

[15] IPL. Retrieved from: https://www.ipl.org/essay/Disadvantages-Of-Adaptive-Control-PCCJS8RYKXG

[16] Staff Writer. Understanding CNC Machining. *Thomas.* Retrieved from:

https://www.thomasnet.com/articles/custom-manufacturing-fabricating/understanding-cnc-machining/