People's Democratic Republic of Algeria
Ministry of Higher Education
and Scientific Research
Tissemsilt University

Faculty of Science and Technology
Department of Science and Technology

Final dissertation for graduation

Of Academic Master in

Sector: Mechanical Construction

Specialty: Mechanical and Production Manufacturing

Presented by: BENSMAIL Ahmed

Theme

Modeling, design and simulation of a manipulator robot

Supported on, 13/06/2023.

Before the Jury:

KHERRAB Mohamed	Chairman	M.A.A.	Univ-Tissemsilt
SATLA Zouaoui	Supervisor	M.C.B.	Univ-Tissemsilt
BENDIABE TALEB Hicham	Examiner	M.A.A.	Univ-Tissemsilt

College year: 2022-2023

THANKS AND APPRECIATION

I would particularly like to thank:

- my Assistant Master Supervisor ***Dr. SATLA Zouaoui*** *Department science and Technology of the University Tissemsilet for guiding me throughout my work, for their contribution to our training and the help provided for the realization of this thesis of Masters. It is thanks to their human and scientific qualities that we have been able to continue and carry out this research work. May they find here the expression of our deep gratitude.*

- We would like to thank all the professors of mechanical engineering ***KHERRAB Mohamed BENDIABE TALEB*** *In order to accept the discussion of a graduate thesis of Masters, and* ***MARGHACH sidi_Mohamed****,* ***HAMDI amine****,* ***El Hennani abdelhak*** *and* ***Hicham*** *.for framed us throughout our formative years*

Our thanks also go to:

- A big thank you to all Teachers and Heads of the Department science and technology and to all our Sincere colleagues from the Faculty of Technology as well as to all those who have contributed from near or far to our training, to the advancement and realization of this research work.

Nevertheless, I wish to express my sincere appreciations and love to all of Bouhalli Mohamed Amine and Abd el Moumenne Briniss whom been a big support to me

Abstract

The aim of this study is to learn how design and simulate a very complicated industrial robot. This study is divided into two parts:

The first part of this study: creating and designing an industrial robot using SolidWorks and converting it to MATLAB for simulations, using the Sim Mechanical system.

The second part of the study: a direct design of an industrial robot consisting of three arms and three joints, the use of MTLAB/SIMULINK in order to simulate all types of different levels of studies (forwards kinematics, inverse kinematics, Jacobian) to prove the quality and applicability of a class performed at the level of Conducting a dynamic study, the mismatch of the selected study and the response of the robot forced us to use a PID controller in order to improve the speed and quality of the response.

Keywords:

Robot, MATLAB, Solid works, Assembly, forward kinematics, inverse kinematics, dynamics model, Jacobian model

Resume

Le but de cette étude et apprend comment concevoir et de simuler un robot industriel très complique. Cette étude est divisée en deux parties :

Première partie de cette étude : créer et concevoir un robot industriel en utilisant SolidWorks et le convertir dent le MATLAB pour les simulations, en utilisant Sim Mechanics.

La deuxième partie de l'étude : une conception directe d'un robot industriel composé de trois bras et trois articulations, l'utilisation de MTLAB/SIMULINK afin de simuler tous types de niveaux d'études différents (Model géométrique directe, Model géométrique inverse, jacobien) pour prouver La qualité et l'applicabilité d'un cours effectué au niveau de la réalisation d'une étude dynamique, l'inadéquation de l'étude sélectionnée et la réponse du robot nous ont obligés à utiliser un contrôleur PID afin d'améliorer la rapidité et la qualité de la réponse.

Mots clés:

Robot , MATLAB, SolidWorks, Assemblage, model géométrique direct, model géométrique inverse, dynamique, jacobien .

الملخص

الهدف من هذه الدراسة هو تصميم و محاكاة لربوت صناعي و تنقسم هذه الدراسة من جزئين:

الجزء الاولى من هذه الدراسة: انشاء و تصميم الربوت الصناعي ب إسعمال سوليدواركس وتحويله الى الماتلاب من اجل المحاكات، بلاستعانة بنضام سيم مكانيك

الجزء الثاني من الدراسة :تصميم مباشر لربوت الصناعي المكون من ثلات اذرع (روابط) و ثلات مفاصل , استعمال متلاب سميليك من اجل محكاة جميع انواع مستويات مختلفة من دراسات (حركية إلى الأمام , حركية العكسية, جاكوبين) لاثبات جودة و تطباق صنف منجز على مستوى متلاب و اجراء دراسة ديناميكية. عدم تطابق الدراسة مختارة و استجابة الربوت ارغمنا على استعمال متحكم من نوع PID من اجل تحسين سرعة و تحسين نوعية استجابة

كلمات مفتاحية

روبوت ، ماطلاب، سوليدواركس، الحركية إلى الأمام ، الحركة العكسية ، نموذج الديناميكيات ، نموذج الجاكوبين

TABLLE OF CONTENTS

Chapter I : Definitions and outs on a robots

Chapter II Mathematical modeling

Chapter III simulation and result

General conclusion

LIST OF FIGURES

List of tables

Abreviations:

RRR : rotation rotation rotation

FK : forward kinematics et en français **(MGD**) : modelé géométrique direct

I K : inverse kinematics et en français **MGI** : modelé géométrique direct

q: joint Angels (represented the X Y Z position in trapezoidal velocity)

qd: represent the velocity

qdd*:* represent the accelerations

CAD: Computer Aide Design, et en français Conception Assistée par Ordinateur (CAO)

FK: forward kinematics et en français (MGD) model géométrique direct .

Ik : inverse kinematics et en français (MGI) model géométrique inverse .

(PID): A proportional–integral–derivative .

DH: Denavit-Hartenberg Method

UAV: unmanned aircraft vehicle.

ISO :International Organization for Standardization

PLS: Programmable logic controller

SMA: Articulated Mechanical Systems

General Introduction

The first robots appeared in the sixties of the twentieth century. The research disciplines associated with this theme are articulated around perception (pattern recognition), decision (artificial intelligence) and action (automatic). Initially, the researchers focused on manipulation robotics and its applications, particularly in the manufacturing industry. From the 1980s, due to the increased development of on-board computing and the improvement of sensors, mobile robotics became a major research topic aimed at the autonomy of the machine, whether to explore distant worlds such as the surface of Mars, perform various tasks or to intervene in dangerous areas.

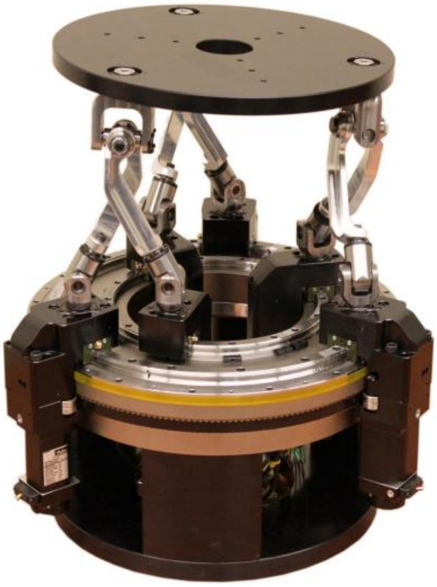
Since the appearance of the first industrial robots, their growth has been considerable, and each year new industrial sectors are opening up to robotization as equipment becomes more diversified, more adaptable and above all cheaper. Robots have become so essential in certain industrial sectors (the automobile industry for example) that their use determines the economic survival of companies. It therefore becomes important to master their technology. The term robot can designate a wide variety of technological achievements, ranging from simple mechanical devices performing repeated movements, to machines morphologically analogous to human arms and possessing a certain intelligence. as an anecdote, the firm Animation, which built the first robot around 1960, was practically the only one on the American market fifteen years later, and had marketed only 1000 robots. While the JIRA (Japanese Industrial Robotics Association) announced in 1975 65,000 robots?! It later turned out that what some referred to as simple automation, others called it robot. Today there are more than 15,000 industrial robots in Japan, or about nearly a third of the world's fleet.

The manuscript of this dissertation is divided into three chapters:

In the first chapter, definition some fundamental concepts related to robotics in general. And presentation their components, some demonstration their main axes, general aspects, categories, areas of application of robotics and the general structure of the robot manipulator.

In the second chapter, this part is devoted to a mathematical study of the equations of the direct and inverse geometric models and the direct and inverse kinetic models by a general and structured approach based on linear algebra

In the third chapter creation a geometric model of the robot mechanism, by using Solid works in combination with Sim-Mechanics and MATLAB Simulink. The design of robot in SolidWorks is done in two stages: first stage, Creation of the robot; Each part of the robot has been built independently, The Assembly of the robot is made by the connectivity of: - fixed base-with base of the robot by a rotational articulation - First link with the base of the robot by rotating articulation-shoulder with second link by a rotational articulation and lastly wrist with shoulder by a rotational joint. The simulation of our robot in Simulink is done using Sim Mechanics. The interfacing of Solid Work with Sim-Mechanics is done using 'Sim-Mechanic slink'. And this method we used in case if the robot is complex like delta robot to facilitate the study. In the second part of this chapter, present the development of the forwards kinematics (direct geometrical model) by using the homogeneous transformations, one calculates for each reference Ri the matrix of transformation Ai, allowing the passage of the reference Ri-1 to the reference Ri. By using the Denavit-Hartenberg method then presentation the development of the inverse kinematics (inverse geometric model) using the Paul method. In what follows the simulation results of the movements of the robot in three dimensions using PID regulators or PID corrector (proportional, integral, derivative). These regulators are control algorithms that improve the performance of the movement for each element of our robot in this Section; we present numerical results and discussions. Finally, conclusions are drawn in last Section.



Chapter I:

Definitions and outs on a robots

1.1Introduction:

The concept of robot dates back several centuries, but the term robot was coined by the Czech Karel Capek in a play written in 1920: "RUR or the universal robots of Rossum". This term is derived from the Czech verb "robota" meaning forced labor or drudgery. It is certain that for a very long time, developers of automatons have sought to be able to infuse their machines with behaviors adapted to the circumstances. Unfortunately, until the twentieth century, the techniques were too primitive to allow such achievements. It was not until 1959 that Georges Devol invented an original, versatile and reprogrammable machine, which allowed the robot to acquire an industrial reality. But in fact it was only towards the end of the 1970s that the first generation of industrial robots saw the light of day.[1].

1.2 Historied:

The history of robotics is one that is highlighted by a fantasy world that has provided the inspiration to convert fantasy into reality. It is a history rich with cinematic creativity, scientific ingenuity, and entrepreneurial vision. Quite surprisingly, the definition of a robot is controversial, even among roboticists. At one end of the spectrum is the science fiction version of a robot, typically one of a human form — an android or humanoid — with anthropomorphic features. At the other end of the spectrum is the repetitive, efficient robot of industrial automation. In ISO 8373, the International Organization for Standardization defines a robot as “an automatically controlled, reprogrammable, multipurpose manipulator with three or more axes.” The Robot Institute of America designates a robot as “a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks.” A more inspiring definition is offered by MerriamWebster, stating that a robot is “a machine that looks like a human being and performs various complex acts (as walking or talking) of a human being. [2]

2 Generalities of robotics:

Until recently, robots were mainly used in factories for automating production processes. In the 1970s, the appearance of factory robots led to much debate on their influence on employment. Mass unemployment was feared. Although this did not come to pass, robots have radically changed the way work is done in countless factories. This article focuses on how the use of robotics outside the factory will change our lives over the coming decades. New robotics no longer concerns only factory applications, but also the use of robotics in a more complex and unstructured outside world, that is in the industrial field, robotics produces automatons performing specific functions on assembly lines. Robotics also produces machines capable of moving in different environments: dangerous (polluted, radioactive, etc.), aerial, submarine, space, etc. In addition to industry, robotics now concerns scientific research, space exploration and military defense or law enforcement activities. It is also of interest to the medical sector, for prostheses, assistance to surgeons or nurses.

In order to understand the possibilities and impossibilities of the new robotics, it is therefore important to realise that robots are usually supported by a network of information technologies. [3]..

2.1 Generalities of drone:

Drones are flying robots that include unmanned aerial vehicles (UAVs) that fly thousands of miles and small drones that fly in confined spaces. Aerial vehicles that do not carry a human operator, fly remotely or autonomously, and carry lethal or non-lethal payloads are considered drones. A ballistic or semi-ballistic vehicle, cruise missiles, artillery projectiles, torpedoes, mines and satellites cannot be considered drones [4]. Advances in manufacturing, navigation, remote control and energy storage systems have enabled the development of a wide range of drones that can be used in a variety of situations where human presence is difficult, impossible or dangerous [5,6]. Flying robots for military surveillance, planetary exploration, and search and rescue have received more attention in recent years. Depending on the drone flight missions, the size and type of equipment installed are different [6]. The tremendous benefits of drones have led to a myriad of studies to focus on optimizing and improving the performance of these drones. According to the characteristics mentioned, drones benefit from the possibility of carrying out various operations, including reconnaissance, patrol, protection, transport of loads, and aerology [7]

3.1 Definition of a robot:

A robot is a mechatronic device (combining mechanics, electronics and computing) that automatically performs either tasks that are generally dangerous, painful, repetitive or impossible for humans or simpler tasks but performing them better than a human being would do. The most advanced robots are able to move and recharge themselves, like the ASIMO robot made by Honda.

Despite their high cost at the time (due to the lack of powerful mass-produced microprocessors), robots imposed themselves from the beginning of the 1970s, for certain tasks such as painting car bodies, in an atmosphere of toxic vapours. Since then, the evolution of electronics and computing has enabled robots to perform increasingly complex tasks, with more and more autonomy, and more and more quickly.

The science of robots is called robotics.

The term "robot" is also used to designate a device that is not automatic, to evoke the high technicality of the device, like the robot surgeon Da Vinci.

Microrobots exist, which can for example move on water like geris, and the first nanocomponents and nanomotors seem to suggest the creation of nanorobots in the years or decades to come.

Figure 1: robot arm[8]

3.2 Different component of a robotic:

3.2.1 Mechanism:

An assembly of moving parts performing a full functional motion, often forming part of a large machine such as a Bra de Humane

3.2.2 Perception:

Which manages the relationship between the robot and its environment. The organs of perception are called proprioceptive sensors when they measure the internal state of the robot (positions and speeds of the joints)

3.2.3-The order:

Which synthesizes the setpoints of the servocontrols controlling the actuators? Based on the perception function and user orders, it generates robot actions. The human-machine interface through which the user programs the tasks that the robot must perform. The workstation and the devices that constitute the environment in which the robot evolves.

3.2.4 Automaton:

A PLC is a machine programmed to perform a specific task in a given environment.

4. Composition of a robotic system:

A robot is an autonomous device composed of articulated mechanical elements and electronic elements: a controller, the programming unit, the brain of the robot. The programming tells the robot which movements it should perform.

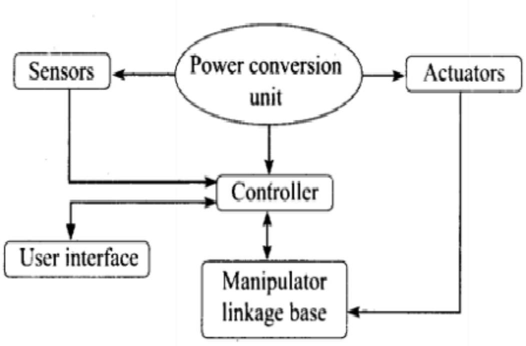


Figure 2: Compositions of robot [9].

4.1 Actuator:

The actuator is a hardware device for transforming digital information into a physical phenomenon; hence its name. It can modulate the behavior or change the state of a system. It can be alarms or switch from the word “action” was born the word “actuator”. An actuator is the part of the robot allowing it to act on its environment. It is thanks to him that the robot can perform work. ". Actuators can achieve high precision but the design of these actuators for such applications comes up against difficulties. One solution lies in the use of systematic optimization algorithms. It is still necessary to have, to use such algorithms, flexible models, both precise and fast.[10]

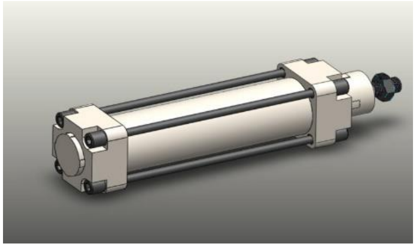


Figure 3: Actuator [11]

4.2 Sensors:

A sensor makes it possible to perceive a physical signal. A great number of sensors are available for different physical quantities in order to study them. This is for a robot to sense its environment or for the main controller of the robot to get information about the internal workings..**[12]**

Figure 4: Captor [13]

4.3 Mechanisms:

Some successful mechanical devices function smoothly however poorly they are made, while others do this only by virtue of the accurate construction and fitting of their moving parts. Device consisting of parts assembled or connected to each other and fulfilling a specific function (drive, braking, locking, etc.).[14]

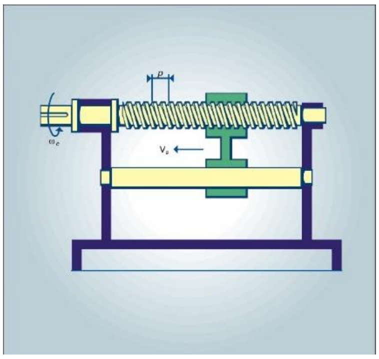


Figure 5: A mechanism [15]

4.4 End-Effecter:

In robotics and mechanical engineering, an effector is the tool set in motion by actuators. Robots control their effectors, also called end effectors. Effectors include legs, wheels, arms, fingers, wings, and fins. The orientation of the end-effector about the tool line is included in the analysis to completely describe the six degree-of-freedom motion of the end-effector. The linear and angular properties of motion of the end-effector, determined from the differential properties of the ruled surface, are utilized in the trajectory planning. [16]

Figure 6: end effector [17]

5 some application on Robot:

Automobile industry

- Medicine
- Nuclear energy
- Agriculture
- Space exploration
- Underwater inspection

5.1 Types of Tash:

Depending on the type of task and the sector of application, the robot will have a different mechanical architecture and control-command system among these tasks can be in normal or hostile environments:

- Parts handling

Welding (ensure good precision of the weld path).

Painting (must be animated at high speed)

- Assembly (the requirement for very high positioning accuracy)
- Assist the surgeon in an operation (medical robotics)
- Mobile robot evolving in an unknown environment and on uneven ground (military robotics).

6 Types of SMA Architecture in Robotics:

The motorized connections are those with which the joint control variables qi are associated. This is a way to distinguish active links from passive links.

6.1 Series architecture (or open kinematic chain):

There is only one possible path to go from the frame to the gripper; the segments of the robot as well as the links are indeed put in series.

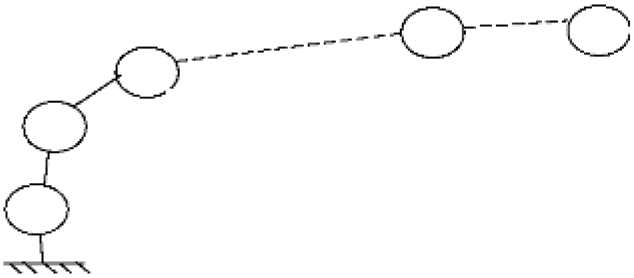


Figure 7: Serial architecture [18]

6.2 Parallel architecture (or multi-loop kinematic chain):

To go from the gripper to the frame, there are six different paths and on each path, a single prismatic active link

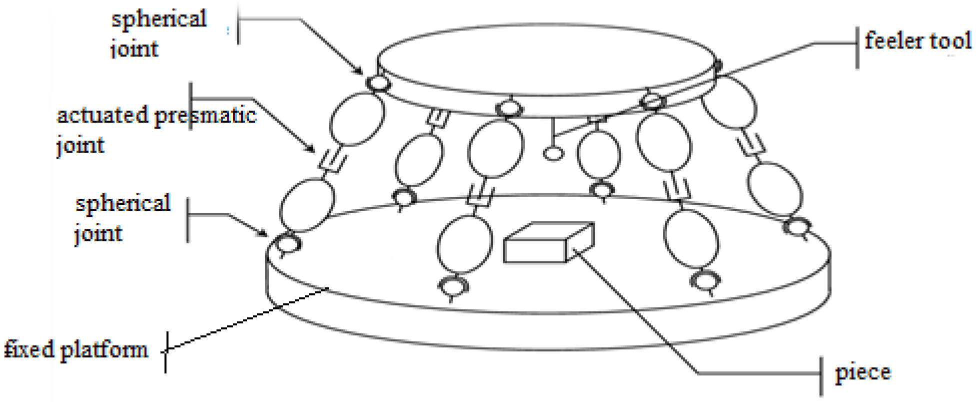


Figure 8: Parallel architecture [19].

7.1 Vocabulary of robotics

The vocabulary of robotics is defined as follows:

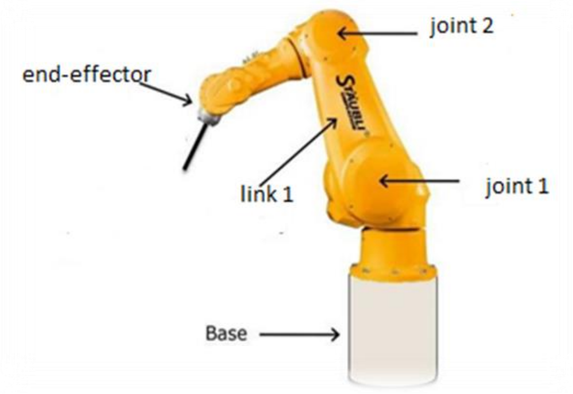


Figure 9: Arm manipulator [20]

7.2 The basic functions of robotics:

They are multiple, to which are added: handling, assembly, control and surveillance, intelligence and inspection, packaging, implants, agriculture, playful robots, etc.

a. Automaticity:

It gives the possibility of accomplishing a more or less complex task without having to resort to a human operator.

b. Adaptability:

It allows a system to perform a specific task in a variable, partially or totally unknown environment.

c. Versatility:

The system can achieve different objectives while maintaining the same structure.

a) For the manipulators, the change implies a modification of the program.

b) In the case of advanced robots, the modification of the program is carried out according to the adaptation to the multiplicity of tasks, the environment, the taking of information which is analyzed and processed by the computer, which orders the new behavior to be adopted by the robot.

8 Type of robots:

Assembly lines are flow-oriented production systems which are still typical in the industrial production of high quantity standardized commodities and even gain importance in low volume production of customized products (Scholl & Becker, 2006). In the past decades, robots have been widely used in assembly systems as called robotic assembly lines. An assembly robot can work 24 h a day without worries of fatigue. Goals for robot implementation include high productivity, quality of product, manufacturing flexibility, safety, less demand for skilled labor, etc.[21] Robots can also perform tasks in work environments that are too dangerous for humans. Robots can perform automatic tasks, but some are also endowed with certain intelligence. Their faculties of adaptation require qualities of perception in order to interact with their environment

8.1 Industrial robot arms:

A robotic arm is simply a programmable mechanical arm that is very similar to a human arm in functionality and sometimes even appearance. It is the combination of various other mechanisms and smaller subassemblies. Manipulators and linkages are usually connected by joints that allow either translational motion also called linear motion or rotational motion.[22]

Figure 10: industrial robot arm [23]

8.2 Robots in a hostile environment:

Japanese firm Toshiba, which originally supplied reactors to the power plant, including the one that was the worst hit, has been heavily involved in the cleanup. Between 2012 and 2013, it used quadruped robots, fitted with cameras and able to walk up and down stairs, to investigate the vent pipe at Unit 2. It is now working on improving them so they can be deployed in further applications. In 2016, Toshiba also unveiled an amphibious remote-controlled robot designed to remove debris and nuclear fuel rods.

Figure 11: robot hostile [24]

he model will be able to clean the reactors, remove unnecessary debris and fuel rods. He can carry out two actions at the same time. The prototype is also equipped with several cameras so that the men follow the course of the operation. It is controlled remotely (from another building) by an engineer.[25]

8.3 Exploration sous-marine:

A team from Japan's Okayama University is working on an underwater robot capable of navigating and performing tasks completely independently. The machine combines a stereoscopic 3D vision system and sonar that allow it to locate and track objects with an accuracy of a few millimeters. This robot could be useful in particular for monitoring missions of underwater or environmental installations.[26]


iStock by Getty Images

Fugure 12 : Represents underwater exploration robot [27]

8.4 Health:

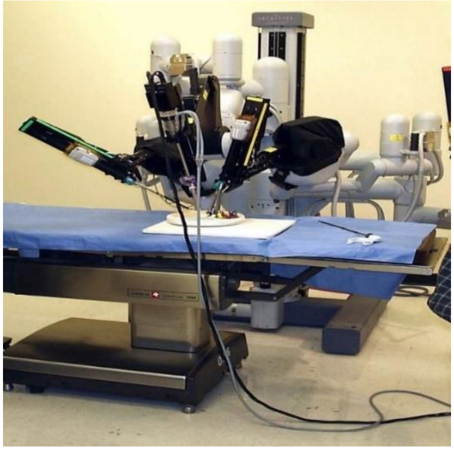


Figure 13: Represents a medical robot [28]

Surgical robotics is a new technology that holds significant promise. Robotic surgery is often heralded as the new revolution, and it is one of the most talked about subjects in surgery today. Up to this point in time, however, the drive to develop and obtain robotic devices has been largely driven by the market. There is no doubt that they will become an important tool in the surgical armamentarium, but the extent of their use is still evolving**..**[29]

8.5 The Military Domain:

A military robot is a robot, autonomous or remotely controlled, designed for military applications. Drones are a subclass of military robots.

Systems are already currently in service in a number of armed forces, with remarkable success, such as the Predator drone, which is capable of taking surveillance photographs, and even launching missiles on the ground, for combat drones . Studies are continuing because this type of device offers promising possibilities.

Figure 14: Robot military [30]

Nowadays, researchers design and manufacture different types of helicopters without pilot for vertical grips, landing and hovering. There are four types of helicopters, drones, single rotor, coaxial rotor, tandem rotor and quadrotor [38]. Heli-wing drones are other types of drones that use a wing, They can fly the helicopter vertically and also the drone fixed-wing.[31]

8.6 Humanoid:

Human beings have an “anthropomorphic” empathic relationship with robots in the sense that they attribute particular human characteristics to them. Robots in particular are capable of credibly simulating feelings and emotions. When they do not look like us morphologically (as is the case with a comforter for a child), we can easily form a beneficial and harmonious empathic relationship with them. When, however, the robots resemble us too much (as is the case for a humanoid), the relationship is transformed and can become repulsive or "pathological": this is what the first Humanoid robot foreshadows. [32]

Figure 15: Represents a humanoid robot [32]

8.7Agriculture:

Autonomous agricultural robots offer ways to improve production by automating tasks and increasing data processing and planning capabilities. Autonomous navigation functionality is central to the design of such intelligent systems for automating agricultural operations.[34]. It is a question of developing navigation strategies allowing a mobile robot to evolve and intervene autonomously and in complete safety in a farm. This type of farming environment is highly scalable and has many static (buildings, storage areas, etc.) and dynamic (cars, farm machinery, human operators, animals, etc.) obstacles. The proposed navigation strategy must therefore be both reactive and adaptive. [33]

fugure 16: Agriculture robot [35]

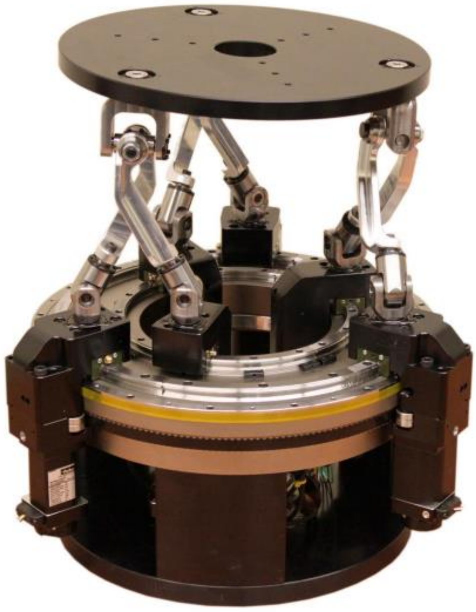
8.8 Domestic use:

For robots to be truly useful to humans, we need the robot to be equipped with proactive behaviors; because it can help humans achieve their object us algorithms.[36]. Today, it seems normal to see a robot at home vacuuming, washing the windows or cleaning the floor in complete autonomy. These chores can be delegated to machines, and comfort is greatly improved on a daily basis.

Figure 17 : Household robot [37]

9 Conclusion:

In this chapter, we have presented some fundamental concepts related to robotics in general. And we present their components; we presented their main axes, general aspects, categories, areas of application of robotics and the general structure of the manipulator robot

Chapter II : Mathematical modeling


1.2 Introduction:

This chapter is devoted to the derivation of the direct kinematics equation by a systematic, general approach based on linear algebra. This allows the position and orientation of the effector (pose) to be expressed as a function of the articular variables of the mechanical structure with respect to the reference frame. This chapter ends with the derivation of the solutions to the inverse kinematics problem, which consists in the determination of the joint variables corresponding to the position of the given effector.

2.2 Geometric model of a robot:

Performance. The necessary mathematical models are

- The direct and inverse geometric models which express the speed and the situation of the terminal organ according to the joint variables and vice versa.

- The direct and inverse kinematic models which express the speed as a function of the joint variables and vice versa.

- The dynamic models defining the equations of motion of the robot which make it possible to establish the relationships between the torques and forces exerted by the and the positions, speeds, accelerations of the joints. To control or simulate the behavior of an articulated mechanical system (robot), one must have a model. Several levels of modeling are possible depending on the objectives, the constraints of the task and the desired

We have a matrix relation of the type:

$$X = f(q) \text{ with } X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, q = \begin{pmatrix} q1 \\ q2 \\ q3 \end{pmatrix} \ldots\ldots\ldots(2.1)$$

Where f is a static vector function (the temporal variable does not intervene)

3.2 Operational contact details:

Several possibilities exist for the definition of the vector X depending on the methods used to specify the position and the orientation. The orientation is specified using the direction cosines (described below), we will have:

$$X = [Mx \;\; My \, Mz \;\; a11 \;\; a21 \;\; a31 \;\; a12 \;\; a22 \;\; a32 \;\; a13 \;\; a23 \;\; a33\,]^t \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.2)$$

In general, the position of a reference point linked to the end device is defined via Cartesian (3 lengths), cylindrical (2 lengths + 1 angle) or spherical (1 length + 2 angles) coordinates.

There are different possibilities to define the orientation of the end device:

Euler angles,

Other types of angles (Bryant, …),

Direction cosines. Their use amounts to considering all the projections of unit vectors carried by the axes of the reference R1, on the axes of the reference R0 (orthonormal reference points). This results in 3 × 3 parameters, in effect:

Six relations are needed to indicate that the basis is orthonormal (3 to indicate unit norms + 3 to indicate the orthogonality of the basis),

Three parameters to describe the orientation of the marker

The vector (a_11 a_21 a_31) represents the unit vector x1 (of the reference R1) along the axes x0, y0, z0 (of the
mark R0).

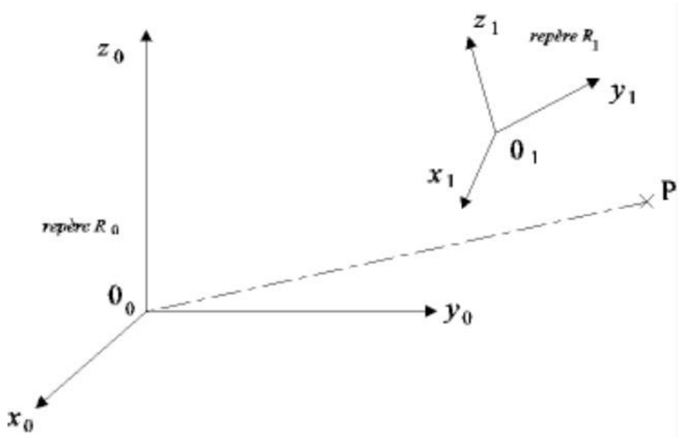


Figure 18 : :Reference R_0 R_1. [38]

The purpose is to express in the base frame R0 the coordinates of the point P having as coordinates (X1, Y1, Z1) in the base frame R1 (i.e.: O1P /1 = (X1, Y1 , Z1)-1), knowing that the origin of the reference R1, that is to say O1, has for coordinates (a, in the reference b,c) R0.

O1p/1 = {x1 y1 z1}, i.e. O1P /1= x1 x1+y1 y1+z1 z1The rotation matrix is

$R01_{,}\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ contains the base deb vectors of R1, c is

That is to say x1, y1, z1, expressed in the base R0 , i.e. according to the vectors x0, y0, z0. The position of the base frame R1 can be deduced from that of the base frame R0 through:[38]

- A translation O0 O1,

Rotations around the 3 axes (x0, y0, z0) of the reference R0 , with which the following elementary rotation matrices can be associated respectively: R01(x0 , θ0 x), R01 (y0 , θ0y)y and R01 (z0 , θ0z), defined later in this section. We then have: R01 = R01 (x0 , θ0 x) R01 (y0 , θ0y) R01 (z0 , θ0z)-

4.2 Case of a simple translation:

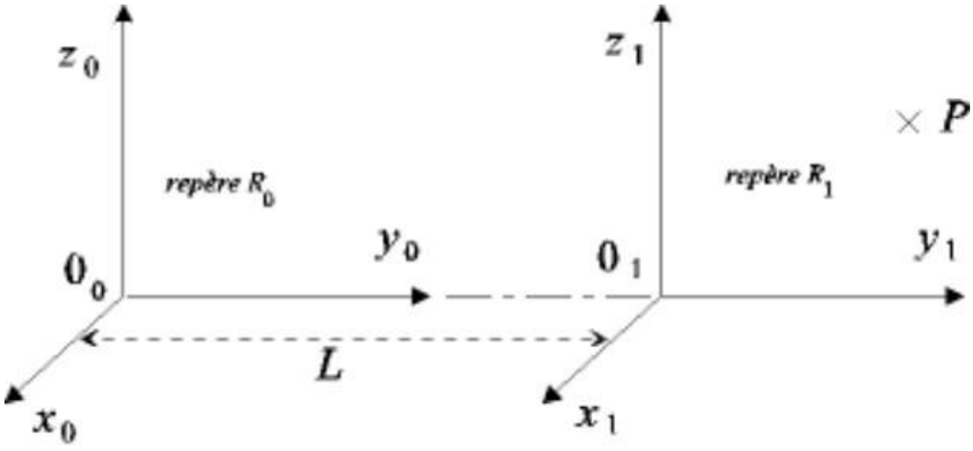


Figure. 19: Case of a simple translation [38]

Where:

$$\overrightarrow{O_0P_{/0}} = \overrightarrow{O_0O_{1/0}} + R_{01} * \overrightarrow{O_1P_{/1}} \ldots\ldots\ldots\ldots (2.3)$$

$$\overrightarrow{op} = \begin{pmatrix} 0 \\ l \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \begin{pmatrix} X_1 \\ L_1 + Y_1 \\ Z_1 \end{pmatrix}$$

The translation matrix (vector) operates along the y0 axis. The rotation matrix (of zero angle) is such that: $x0 = x1, y0 = y1, z\,0 = z1$.

Rotation matrices: Special case of rotation around an axis

By convention, it is considered that the rooted articulation of a robot takes place around one of the 3 axes of an orthonormal frame. Consider, for example, a rotation of angle θ0x around the axis O . [38]

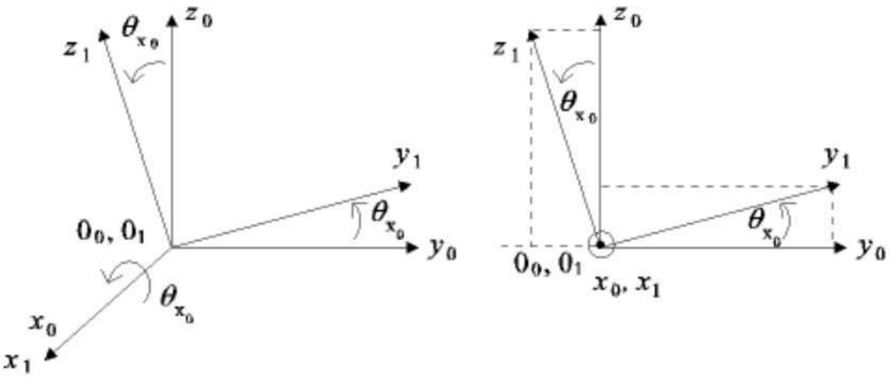


Figure 20: Case of a simple rotation [38]

Where:

$$R01(x0, \theta o) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{pmatrix}$$

$\overrightarrow{x_1} = \overrightarrow{x_0}$, $\overrightarrow{y_1} = \theta\cos(\theta_x)\overrightarrow{y_0} + \sin(\theta_x)\overrightarrow{z_0}$ $\overrightarrow{z_1} = -\sin(\theta_x)\ \overrightarrow{y_0} + \cos(\theta_x)\overrightarrow{z_0}$ ……………….(2.4)

Let us express in the frame R0 the coordinates (x1,y1,z1) of the point P in the frame R1 .

Knowing that the origin O1 of the reference R1 has for coordinates R0

$$\overrightarrow{o_0 p_{10}} = \begin{bmatrix} a + x_1 \\ b + \cos(\theta)\, y_1 - \sin(\theta)\, z_1 \\ c + \sin(\theta) y_1 + \cos(\theta)\, z_1 \end{bmatrix}$$

5.2 Homogeneous transformation matrix:

The figuration of homogeneous coordinates consists of providing any vector annotation with a scale element of introducing a complementary coordinate only moment M for the revealed space of triple rectangular axes distribution in the correlation [38]

$$\boldsymbol{OM} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

Then the representation of the point M using homogeneous coordinates is done with a quaternion,

$$\overrightarrow{OM} = \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}, \text{ with } a = \frac{x}{w}, \ b = \frac{y}{w}, \ c = \frac{z}{w}$$

Consider the homogeneous transformation matrix T01 corresponding to the following partitioned matrix: Rotation

$$\mathbf{T_{01}} = \begin{bmatrix} \overset{\text{Rotation}}{\mathbf{R01}(3,3)} & \overset{\text{Translation}}{\mathbf{t01}(3,1)} \\ \mathbf{ooo} & \mathbf{1} \end{bmatrix}$$

The scale factor is unity (w = 1).

The matrix T01 represents the transformation making it possible to pass from the reference R1 to the reference R0. Transformation of the R1 to the R0 .

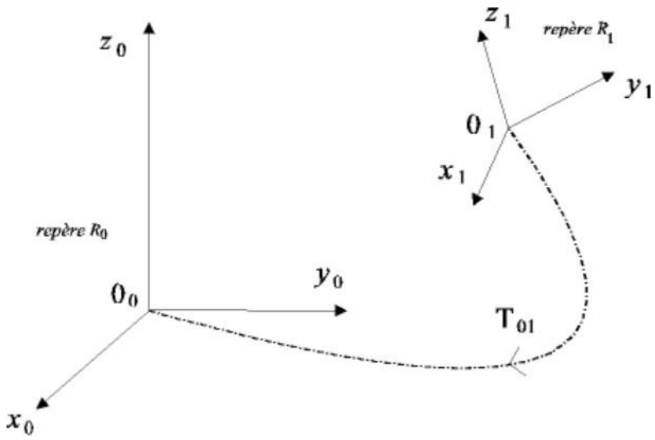


Figure 21: Transformation of the R1 frame to the R0 frame.

it possible to express in the reference R0 the coordinates of a vector expressed in the reference R1. In other Word, we have

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix} = \mathrm{T01}\begin{pmatrix} X_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$

5.2.1 Case of a simple translation:

We note Trans(x, a) the homogeneous transformation matrix corresponding to a translation of a along the axis

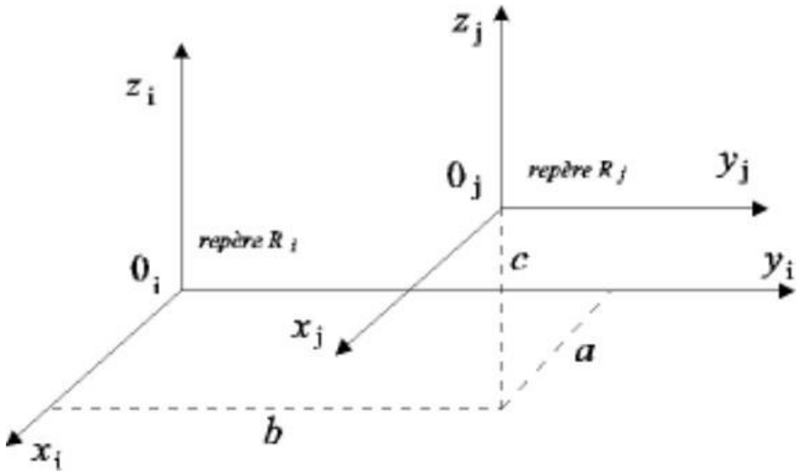


Figure 22: Translation of along the x axis.

Where:

$$T_{ij} = Trans(x_i, a) * Trans(y_i, b) * Trans(z_i, c) \dots\dots\dots\dots\dots\dots\dots.. (2.5)$$

$$Tij=\begin{bmatrix} 1 & 0 & 0 & \mathrm{a} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathrm{b} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let be the coordinates$(X\ Y\ Z)$ of a point M in the frame Rj (i.e.: Oj $Mij = \{X\ Y\ Z\}$ then The coordinates of the point Mans the reference Ri

$$Tij * Mij = \begin{bmatrix} 1 & 0 & 0 & \mathrm{a} \\ 0 & 1 & 0 & \mathrm{b} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{x+a} \\ \mathrm{y+b} \\ \mathrm{z+c} \\ 1 \end{bmatrix}$$

5.2.2 Case of a simple rotation:

We are note Rot(x, θ) the rotation matrix of an angle θ around the x

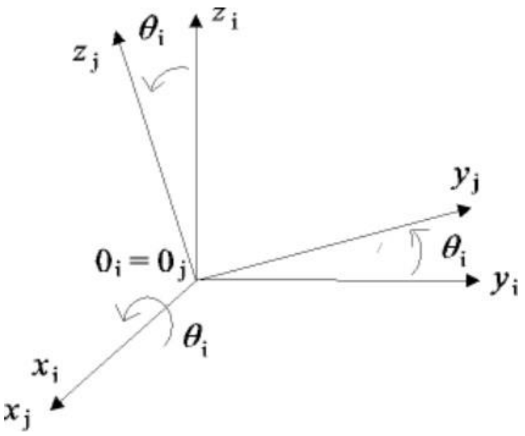


Figure 23: Rotation of angle θ around the x axis.[38]

$$Tij = Rot(xi, \theta_{\mathrm{i}})=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$(x\ X\ Y\ Z\ )$ the coordinates of a point M in the Rj frame then the coordinates of the point M in the frame Ri are

$$Tij = rot\ (xi, yi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ \cos(\theta)\,y - \sin(\theta)\,z \\ \sin(\theta)\,y + \cos(\theta)\,z \\ 1 \end{bmatrix}$$

Let (x y z) be the coordinates of a point M in the location R then the coordinates of the point M in the part R; are transformation matrix can be decomposed into 2 transformation matrices

$$Tij * MIJ \ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ \cos\theta\ y - \sin\theta\ z \\ \sin\theta\ y + \cos\theta\ z \\ 1 \end{pmatrix}$$

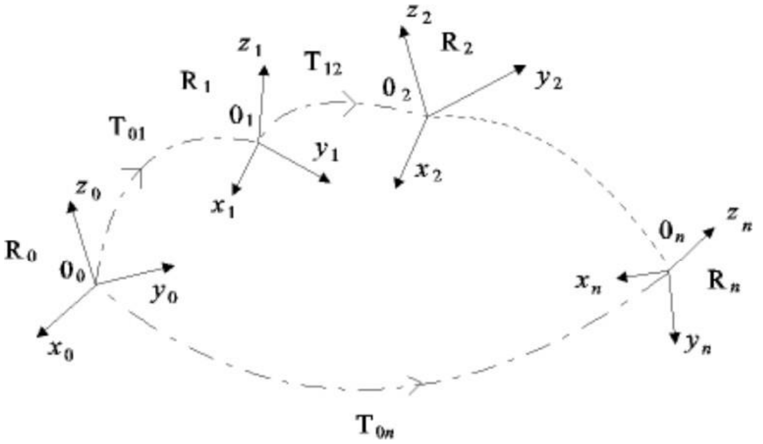


Figure 24: Rotation of an angle θ around the x axis. [38]

$$T0n = T01 * T02 * \ldots\ldots\ldots..Tn\ldots\ldots\ldots\ldots(2.7)$$

Let OM=(x y z), then T0n*(x y z) expresses the coordinates of the point M in the mark R0

$$Oo\ M/_{\mathrm{n}} = T0n * On\ Min\ \ldots\ldots\ldots(2.8)$$

6.2 Obtaining the direct geometric model:

The proposed method uses homogeneous transformation matrices. A marker is associated with each solid of the robot, starting with the base. If a joint has several degrees of freedom (d.d.l.), we introduce fictitious solids (of zero mass and length). The situation of the end device with respect to the base corresponds to the product of the homogeneous transformation matrices of the various reference marks associated with the solid of the robot. Note that the writing of homogeneous transformation matrices is not unique (there is an infinity of ways to link a reference to a solid.[39]

6.2.1 Denavit-Hartenberg Parameters:

The parameters of modified Denavit-Hartenberg make it possible to have a parameter setting of the connections such that the matrices of passage all have the same literal form, which facilitates calculations. The following method applies when the robot corresponds to a simple open chain and its joints are rotoidal, or prismatic (which is the case in general). The bodies making up the robot are assumed to be perfectly rigid and connected by ideal joints (no mechanical play, no elasticity).

The solids are numbered in ascending order starting from the base. Thus the robot is composed of n +1 bodies, denoted C0. Cn, and n joints ($n \geq 1$). The body C0 designates the base (base) of the robot, the body Cn the body carrying the end device. The Rj frame is linked to the Cj body of the robot. The joint variable j, which links the body Cj to the body Cj-1 , is denoted q j [20]. So that the transition matrices T all have the same literal form, we use Denavit Hartenberg's parameterization. It was established by the latter two in the 1950s and then modified by W. Khalil under The name: Modified Denavit-Hartenberg Parameterization. 1. Modified Denavit-Hartenberg parameters The reference Ri (Oi , xi ,yi ,zi) is fixed on the solid Si , it is defined as follows: zi: axis of the Si/Si-1 bond xi: common perpendicular between zi and zi+1 yi=zi ^ x

Note DH uses two groups of parameters:

Location parameters of the zi axis in Si-1;

- αi=angle (zi-1, zi) measured around xi-1
- di= distance (zi-1, zi) measured along xi-1

Motion parameters (rotation and/or translation) of Si/Si-1

- θi: angle (xi-1, xi) measured around zi
- ri: distance (xi-1, xi) measured along zi De

1. Ri-1 rotates around its x axis by an angle αi
2. it undergoes a translation of length di along x
3. a rotation around Z of angle θi
4. a translation along Z of length ri

The joint parameter Q_i is equal to:

Ri in case of translation

Oi in case of rotation.[39]

Ratings

The solids are numbered in ascending order starting from the base. Thus the robot is composed of n+1 body noted c0……Cn and of n articulation (n>1) the body c0 designates the base (the base) of the robot, the body Cn the body carrying the terminal organ

The Rj starts again and binds to the Cj blows of the robot

The variable of the articulation j which links the body Cj to the body Cj-1.[38]

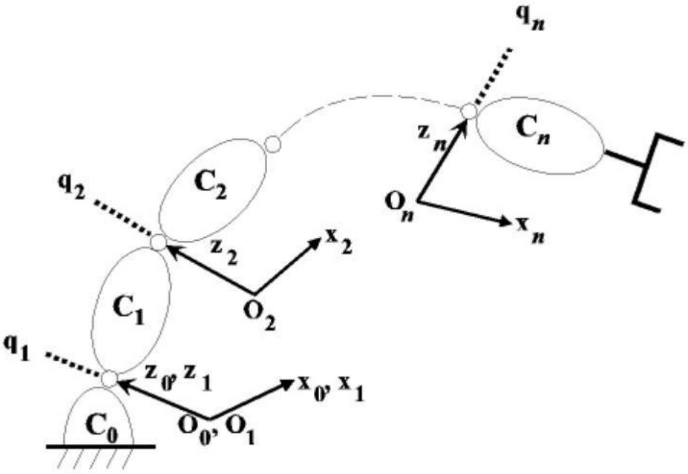


Figure 25: Modified Denavit-Hartenberg parameters [38]

DH Homogeneous Passage Matrix

From relation (1), we can calculate the passage matrix DH.

$$T^{i-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\propto) & -\sin(\propto) & 0 \\ 0 & \sin(\propto) & \cos(\propto) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{i-1} = \begin{bmatrix} \cos(\theta 1) & -\sin(\vartheta 1) & 0 & d1 \\ \cos(\alpha 1)\sin(\theta 1) & \cos(\alpha 1)\cos(\theta 1) & -\sin(\alpha 1) & -r1\sin\alpha 1 \vdots \\ \sin(\alpha 1)\sin(\theta 1) & \sin(\alpha 1)\cos(\vartheta 1) & \cos(\alpha 1) & r1c\alpha 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the definition of the reference frame R0, the simplest choice consists in taking R0 confused with the frame R1 when 0 q1. This means that 0 z is confused with 1 z, moreover O0 = O1 when joint 1 is rotoid, and 0 x is parallel to 1 x when joint 1 is prismatic. - For a prismatic joint j, the axis j z is parallel to the axis of the joint but the position of this axis in space can be arbitrary.

- Very often, the two consecutive axes of a robot are orthogonal or parallel,

- this results in an angle α equal to 0°, ± 90°, 180°.

- The inverse transformation is obtained by the following formula:

$$T_{j,j-1} = \mathrm{Trans}(z_j, -r_j) * \mathrm{Rot}(z_j, -\theta_j) * \mathrm{Trans}(x_{j-1}, -d_j) * \mathrm{Rot}(x_{j-1}, -\theta_j),$$

7.2 Obtaining direct and inverse geometric models:

The direct geometric model is the set of relations which make it possible to express the situation of the terminal organ, the operational coordinates, of the robot according to its articular coordinates. In the case of a simple open chain, it can be represented by the matrix T ,0 n which is calculated by:

$$T_{0,n} = T_{0,1}(q_1 * T1) ; 2\, q_2) \cdots T_{n1,n}(q_n) \text{…….(2.9)}$$

The direct geometric model of the robot can also be represented by the relation:

$$X = f(q) \text{………………………..(2.10)}$$

Where X is the vector of the operational coordinates expressed in the reference frame R0 and q the joint variables. The matrix T ,0 n represents the position and the orientation, expressed in the reference frame R0, of the end device of the robot. The inverse geometric model is the inverse problem

Which makes it possible to know the articular variables according to the situation of the terminal organ, which can be represented by the relation

$$q = g(X) \text{ …………………….(2.11)}$$

7.2.1 Inversion of the geometric model:

It is a question of determining the articular coordinates q allowing to obtain a desired situation for the terminal organ and specified by the operational coordinates X.

There is no systematic method of inverting the geometric model. When it exists, the explicit form, resulting from a mathematical inversion, which gives all the possible solutions to the inverse problem (there is rarely uniqueness of the solution) constitutes the inverse geometric model. There are a number of methods to calculate the inverse geometric model, including Paul's method which treats each particular case separately and is suitable for most industrial robots.[38]

Realization of trajectories between two points I and F

Let the points be:$11 \in (X) - (X_1), F \in (X) - (X_2), I_2\ et\ F_2 \in (X_1) \cap (X_2)$.
Consider the trajectory 1F1 I , it is impossible to join 1 I and F1 without reconfiguration. The trajectory necessarily passes through a point of the common border, it is in particular impossible to carry out the rectilinear trajectory between the points 1 I and F1 .

Let us consider the trajectory 2F2 I , the rectilinear trajectory between the points 2 I and F2 is realizable, as well in posture elbow high as in posture elbow low.

Consider the trajectory 2F1 I , the rectilinear trajectory between points 2 I and F1 is achievable if at point 2 I the manipulator is in a low elbow posture. Othe Wise There Will bé reconfiguration.[39]

7.2 .2 Paul's méthode

In the case of robots with simple geometry (for which most of the distances dj and rj are zero and the angles θ j and α j are equal to 2 .0 ± π), the inverse geometric model (M.G.I.) can be obtained analytically via Paul's method.
Consider the robot described by the following transformation matrix:[38]

$$\mathrm{T}_{0,\mathrm{n}} = \mathrm{T}_{0,1}(\mathrm{q}_1) * \mathrm{T}_{1,2}(\mathrm{q}_2) \cdots \mathrm{T}_{\mathrm{n}-1,\mathrm{n}}(\mathrm{q}_\mathrm{n})$$

U_0 the location of the Rn (linked to the terminal organ) described by:

$$\mathrm{U}_{0s} = \begin{bmatrix} \mathrm{s}_\mathrm{x} & \mathrm{N}_\mathrm{x} & \mathrm{A}_\mathrm{x} & \mathrm{p}_x \\ \mathrm{s}_\mathrm{y} & \mathrm{N}_\mathrm{y} & \mathrm{A}_\mathrm{y} & \mathrm{p}_y \\ \mathrm{s}_\mathrm{z} & \mathrm{N}_\mathrm{z} & \mathrm{A}_\mathrm{z} & \mathrm{p}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3 parameters (independent) to define the orientation of the reference Rn compared to the basic reference).

$$OoOn = pxx0 + pyy0 + pzz0$$

The I K (MGI). is obtained by solving the following matrix equation:

$$U_0 = T_{0,1}(q_1) * T_{1,2}(q_2) \cdots T_{n-1,n}(q_n) \text{ ……(2.12)}$$

Paul's method allows the determination of 1 q, then 2 q and so on until n q. This involves moving each of the joint variables (n q , , q 1 L) one after the other into the left-hand side of the equation. To do this, we multiply by Tj, j−1 (taking successively j = ,1 L,n) on both sides of the equation.[38]

Consider a robot with 6 d.d.l. (6n =). Let's apply the method:

- Pre-multiply the previous equation by T, 1 0 ,

$$T1.0\ (q1) * U0\ =\ T1{,}2\ (q2)\cdots T5{,}6\ (q6).\,h \ldots\ldots\ldots\ldots..(2.13)$$

- The elements located in the left member are either independent or functions of 1 q. The elements located in the right-hand side are either constants or functions of 2 6 q, L, q .

- Deduce 1 q from the previous equation.

- Pre-multiply the previous equation by T 1.2

$$T_{2,1}\,(q)_2 * T_{1.0} * (q)_1 * U_0\ =\ T_{2,3}(q)_3 *\cdots T_{5,6}\ (q_6 \ldots\ldots(2.14)$$

- To deduce q_2.

- Continue this procedure to deduce q3…. q6 ,

$$U_0 = T_{0,1}(q_1)\times T_{1,2}(q_2)\times T_{2,3}(q_3)\times T_{3,4}(q_4)\times T_{4,5}(q_5)\times T_{5,6}(q_6)$$

$$T_{1,0}(q_1)\times U_0 = T_{1,2}(q_2)\times T_{2,3}(q_3)\times T_{3,4}(q_4)\times T_{4,5}(q_5)\times T_{5,6}(q_6)$$

$$T_{2,1}(q_2)\times U_1 = T_{2,3}(q_3)\times T_{3,4}(q_4)\times T_{4,5}(q_5)\times T_{5,6}(q_6)$$

$$T_{3,2}(q_3)\times U_2 = T_{3,4}(q_4)\times T_{4,5}(q_5)\times T_{5,6}(q_6)$$

$$T_{4,3}(q_4)\times U_3 = T_{4,5}(q_5)\times T_{5,6}(q_6)$$

$$T_{5,4}(q_5)\times U_4 = T_{5,6}(q_6)$$

Avec $U_j = T_{j6} = T_{j.j}^{-1} U_{j-\&}$

The resolution of these equations is intuitive, but involves (in principle) some types of equations whose analytical solution is known; for example of the type:

$$-Xr_i\ =\ Y\ ,\ldots\ldots..(2.15)$$

$$-\ \ X\sin(\theta_i) + Y\cos(\theta_i) = Z\ \ ,$$

$$X1\sin(\theta_i) + Y1\cos(\theta_i) = Z1$$

$$x2\sin(\theta_i) + Y2\cos(\theta_i) = z2$$

7.2.3 Motion Generation:

A robot's motion task is specified by defining a path for the robot to follow. A path is a sequence of points defined either in the task space (operational) (in order to locate the terminal organ), or in the configuration space (articular) of the robot (in order to indicate the values of the parameters of connection).

There are some steps on a researcher that he must control at least in order to be able to enter this important field:

a) Programmed by learning, (BYTHON, MATLAB,…).

b) From a CAD system database, the trajectories of a robot can be classified as follows:

c) Movement between 2 points with free movements between points,

d) Movement between 2 points via a sequence of desired intermediate points, specified in particular to avoid obstacles; the trajectory is free between the intermediate points,

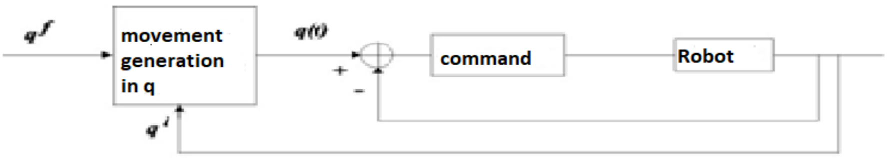
e) Movement between 2 points, the trajectory being constrained between the points (straight trajectory for example),

f) Movement between 2 points via intermediate points, the trajectory being constrained between the intermediate points

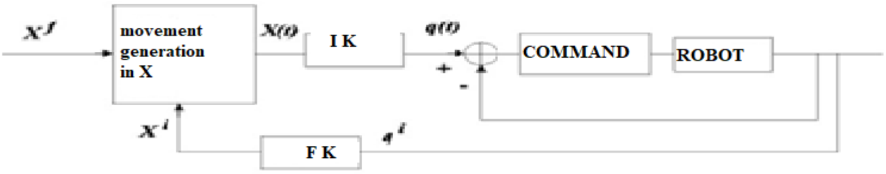
In the first two cases, the generation of movement can be done directly in the space of the configurations: it results in a sequence of articular positions constituting the instructions of the servo-controls.

In the last two cases, the trajectory being fixed at all times in the operational space, it is preferable to reason in this space. The control law generated must then be transformed into joint instructions by the coordinate changer.

These figure represent– generation of movement in the articular space and generation of movement in the operational space – are schematized



Motion generation in the joint space



motion generation in the operational space

Figure 26: Motion generation in the joint space and motion generation in the operational space

The Generating motion in the joint space has several advantages - motion is minimal over each joint,

-it requires less online calculation (in the sense that there is no coordinate changer),

- The movement is not affected by the passage on the singular configurations,

- The maximum speed and torque constraints are known with precision since they press at the figure limits of the actuators.

On the other hand, the geometry of the trajectory in the operational space cannot be imposed. Between 2 given points, the terminal organ moves in an unpredictable but repetitive way (which can cause collisions when the robot evolves in a cluttered environment

This type of movement is therefore suitable for making rapid movements in an open space.

The generation of movement in the operational space makes it possible to control the geometry of the trajectory (rectilinear movement for example). However

-It involves the metamorphosis into articular coordinates of each point of the trajectory,

- It can be defeated when the calculated trajectory passes through a singular position,

It can be defeated each time the points of the trajectory generated are not in the accessible volume of the robot or each time the trajectory could reconfigure the mechanism (change of appearance during the trajectory) [38]

8.2Kinematic modeling:

The kinematics complete the geometric modeling by establishing the relation between the the velocities of the articulation parameters (q) and the velocities of the articulation variables. These relations are designated by the notation $X = j(q)\,.\,q$ is a matrix that boosts the configuration of the SMA. The obvious property of the kinematic model is its linearity with respect to velocities.

It is therefore a priori easier to handle than the geometric model,[39]

8.2.1 Forword kinematic modeling (MCD):

The direct kinematic model, determined by the time derivatives of the

Operational coordinates $\ddot{p}$ and $\ddot{q}$ of the articlier coordinates expresses the operational velocities as a function of the articulate velocities

P And w are speeds of translation and rotation of the probe in operational space

Given by the following relation$\begin{pmatrix}\mathbf{p}\\ \mathbf{w}\end{pmatrix}$= j(q) *dq

P: represents the absolute linear velocity of the probe relative to R0

W: representing the absolute rotation vector of the probe with respect to R0.

$J(q)$: The Jacobian matrix as a function of joint variables of dimension (m x n) With:

- n: the number of degrees of freedom of the articulated structure

- m ≤ 6 degrees of freedom of the TO. If m=6 i.e. 3 for the speed of translation and 3 for the speed of rotation[39]

Relationships will give

$$\boldsymbol{p} = \frac{\mathrm{dp}}{\mathrm{dt}} = \frac{\partial \mathrm{p}}{\partial \mathrm{ti}} \times \frac{\partial \mathrm{p}}{\partial \mathrm{t}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.16)$$

The rotational speeds are determined by the following relation:

$$W = \begin{pmatrix} wx \\ wy \\ wz \end{pmatrix} = [\partial 1 * A10 * a1 \ \ldots\ldots\ldots\ldots\ldots\ldots..an] * \begin{pmatrix} q.1 \\ q.j \\ q.n \end{pmatrix}\ldots\ldots\ldots.(2.17)$$

${}^{0}Aj$: it is a rotation matrix expressed by the reference Ri in the base R0.aj =(0, 0, 1)T: unit vector carried by the ‘articulation qi .

σi: binary coefficient of the joint (σj =1: prismatic et$\overline{\sigma \mathrm{j}}$ =1rotoïde)

The equations and give the product of the Jacobean and the velocities articular

That's to say

$$\begin{pmatrix} p \\ w \end{pmatrix} = J(q) \times \begin{pmatrix} q, \\ qj \\ q,n \end{pmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.18)$$

8.2.2 Inverse kinematic modeling (MCI);

The objective of the inverse differential model is to calculate, from the given configuration q, the joint differential dq to be controlled to satisfy an imposed operational coordinate differential dX:

$$q^{\cdot} = J^{-1} * \dot{X} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.19)$$

The inverse kinematic model of a manipulator robot, are to calculate the velocities of the joint coordinates (q'), according to the operational velocities (X').[39]

It is noted :

$$q^{\cdot} = j\hat{\ }1\dot{X}^{\cdot} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2,20)$$

8.2.3 Dynamic Modeling (MDD) and (MDI):

The dynamic model is the relation between the torques (and/or forces) applied to the actuators and the joint positions, velocities and accelerations.

With This model is useful for trajectory control and planning because it allows calculation of the joint forces required to follow a given trajectory. The MDI can also be used to write the descriptor model for certain control laws.[39]

The dynamic model is represented by a relation of the form:

$$\Gamma = f(q, \dot{q}, \ddot{q}, f(e) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.21)$$

With:

Γ: vector of the torques/forces of the actuators, depending on whether the joint is rotary or prismatic. In the following, we will simply write couples [39]

q: Vector of joint positions

$\dot{q}$: Vector of joint velocities

q̈: Vector of joint accelerations

f(e): vector representing the external force (forces and moments) exerted by the robot on the environment.

We agree to call inverse dynamic model, or simply dynamic model,.

The direct dynamic model is the one that expresses the accelerations as a function of the positions, speeds and torques of the joints. it is then represented by the relation: [39]

$$q^{\cdot\cdot} = g(q, q', r, f(e)) \dots\dots\dots\dots\dots\dots\dots\dots (2.22)$$

The dynamics is systems are governed by the laws of motion introduced by newton's and most of us have used these laws in high school and college, limiting our attention to the dynamics of robots, only the second law movement is all we need. That the force applied on the body of mass M will cause the body to move with an acceleration a and these quantities are related by the famous equation of motion of Newton's second law

$$a^{\cdot\cdot} = f / m \dots\dots\dots\dots\dots\dots\dots\dots\dots(2.32)$$

In reality the force applied is not only used to give acceleration but it is also consumed in a number of other things, this force is consumed in three things

Thirst to impart acceleration to robotic lengths

Second in overcoming centripetal and centrifugal Coriolis forces

Third to overcome the gravitational forces acting on the robot for any robotic manipulator can be written in this form

$$T = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \dots\dots\dots\dots\dots\dots\dots\dots\dots(2.24)$$

M(q)q̈ : these terms are nothing but the accelerating mass, , that is to say. The force consumed to produce the acceleration

C(q,q̇)q̇: these terms group together represent the forces needed to overcome centripetal and centrifugal Coriolis effects on the robot and the last term gives the forces needed to overcome gravitational attraction

G(q) the last term gives the forces necessary to overcome the gravitational attraction

This equation will consist of coupled nonlinear are differential equations of the second order

Lagrange formalism:

This formalism is. Approached because

Based on the expression of the kinetic and potential energies of the manipulator as a function of the generalized variables.

Using n coordinates $q1q2 \ldots\ldots qn,$ the dynamic model Om will take the corresponding algebraic constraint equations as well as the ordinary differential equations from the Lagrange differentiation as follows [38]

$$\text{Lagrange} \qquad F = \frac{d}{dt}\left(\frac{dl}{d\dot{q}}\right) - \frac{dl}{dq} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.25)$$

$L = Ec - Ep$ Is the Lagrange, Ec represents the kinematic energy Ep represents the potential energy [39]

9.2 Newton-Euler formalism:

Based on the equations of Newton (second law) and Euler (corresponding to the kinetic moment theorem), this formalism is well suited to the calculation of the dynamic model (in inverse or direct form), it uses relative coordinates. The generation algorithms of the MDI, like the MDD, use a double recurrence (from the base to the effector and vice versa); the writing of the dynamic torsor in a reference linked to a connection of the solid makes it possible not to show the reaction forces in the links. The implementation by iterative symbolic calculation with the parameterization of Denavit Hartenberg modified and the basic inertial parameters makes it possible to obtain in a systematic way[38]

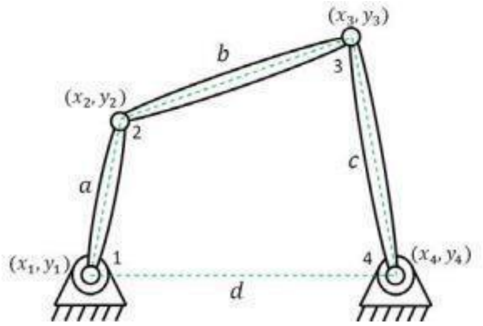


Figure 27 natural coordinates on a closed kinematic chain.

10.2 Navigation Procedure:

The problem of controlling a manipulator arm in the presence of an obstacle is to find a method to bring the arm from a starting point (source) to an end point (target) while avoiding the obstacle encountered.[39]

11. Conclusion:

This chapter is devoted to the derivation of the direct kinematics equation by a systematic, general approach based on linear algebra. This allows the position and orientation of the effector (pose) to be expressed as a function of the articular variables of the mechanical structure with respect to the reference frame. This chapter ends with the derivation of the solutions to the inverse kinematics problem, which consists in determining the joint variables corresponding to the position of the effect

Chapter III : Simulations And Results

I Introduction

The objective of this chapter is to create a geometric model of the robot mechanism, using Solid works in combination with Sim-Mechanics and MATLAB Simulink. The design of our robot in SolidWorks is done in two stages: Creation of the parts; Each part of the robot was built independently, The assembly of the robot is made by the connectivity of: - fixed base-robot base through a rotational joint - the first arm with robot base through a rotational joint-and the second arm through a rotational joint .The simulation of our robot in Simulink is done using Sim-Mechanics. The interfacing of Solid Work with Sim-Mechanics is done using 'Sim-Mechanics-link'. In the second part of this chapter, we will present the development of the model direct geometrical by using the homogeneous transformations, one calculates for each reference Ri the matrix of transformation Ai , allowing the passage of the reference Ri-1 to the reference Ri . We will use the Denavit-Hartenberg method then we will present the development of the inverse geometric model using the Paul method. In what follows we will present the simulation results of the movements of the robot in three dimensions using PID regulators or PID corrector (proportional, integral, derivative). These regulators are control algorithms that improve the performance of the movement for each element of our robot.

2. Modeling with SolidWorks

The modeling of robot in SolidWorks is done in two steps:

2.1 Creation of the parts

Each part of the robot was built independently. The model has three parts
A fixed base, the first link, and the second link, the **figure 28** represents built using SolidWorks.

2.2 Assembly of parts

The assembly of parts is made using ‘Solid works-Assembly’, in which one can build a complex assembly consisting of several components which can be parts or assemblies. The assembly of the robot is made by the connectivity of: -fixed link1, with base of the robot by a rotational joint – and the link2 with a link 1by a rotational joint

After assembling all the parts, the robot model shown in **Figure 28**

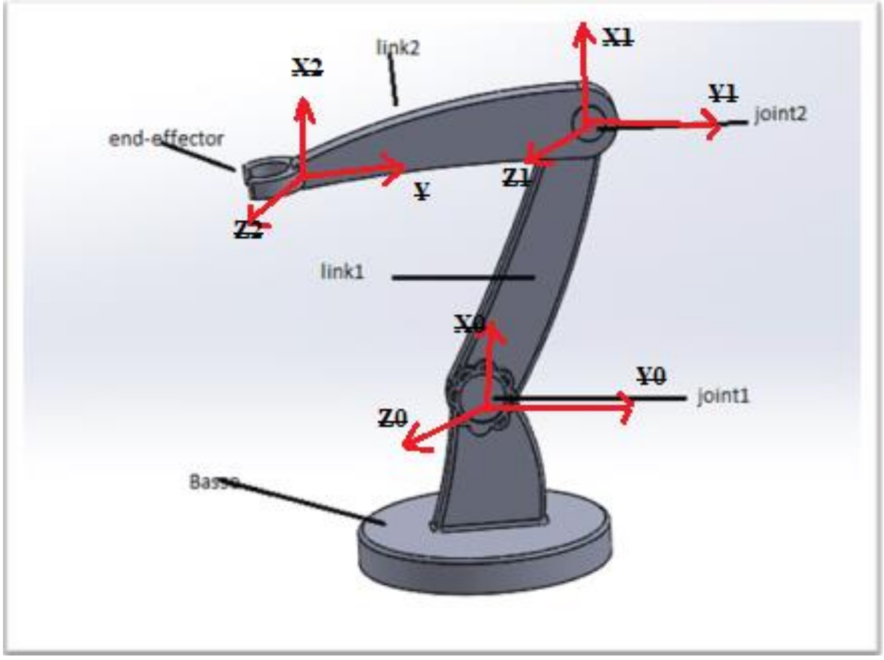
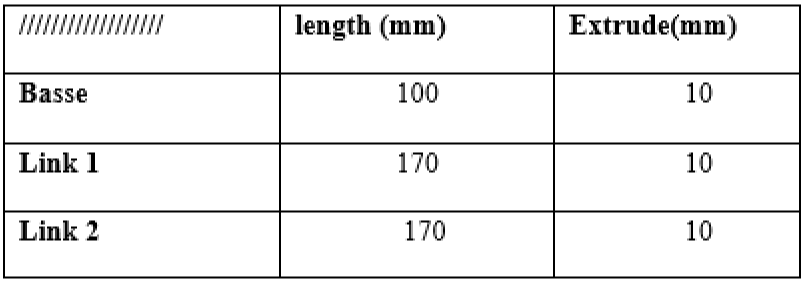


Figure28 model assembles of the robot



//////////////////	**length (mm)**	**Extrude(mm)**
Basse	100	10
Link 1	170	10
Link 2	170	10

Table3.1 Dimension tables robot 1

2.3 Simulation of the robot

The simulation of our robot in Simulink is made using Sim-Mechanics:

Sim-Mechanics is a MATLAB software tool that is used to model 3D mechanical systems within the Simulink environment. This tool is used to build a model composed of bodies, joints, constraints, and force elements that reflect the structure of the system. Sim-Mechanics gives a 3D animation, generated automatically and allows to visualize the dynamics of the system

3. Interfacing Solid Work with Sim-Mechanics

'Sim-Mechanics link' is a necessary intermediate tool for allowing the user to transfer CAD assemblies to Sim-Mechanics models, The intermediate step between the CAD assembly and the Sim-Mechanics model is the export of the XML file of the assembled model. The export process automatically creates

STL files that contain information about robot geometry needed for visualization of system elements. Then, Sim-Mechanics Importer converts these files to XML files and which are references to STL files in order to visualize the robot. Models in Sim-Mechanics with

the physical structure systematically can be produced from XML them. The XML file contains the mechanical structure, DOF and body geometry. The procedure of exploiting the robot design from Solid works to Sim-Mechanics is shown in **Figure30**

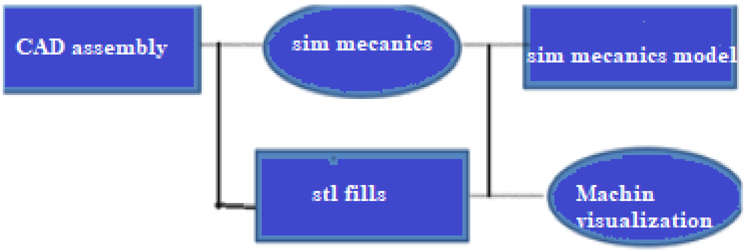


Figure29: Assembly CAD conversion scheme for Sim-Mechanics model

The following figure presents a flowchart of our study:

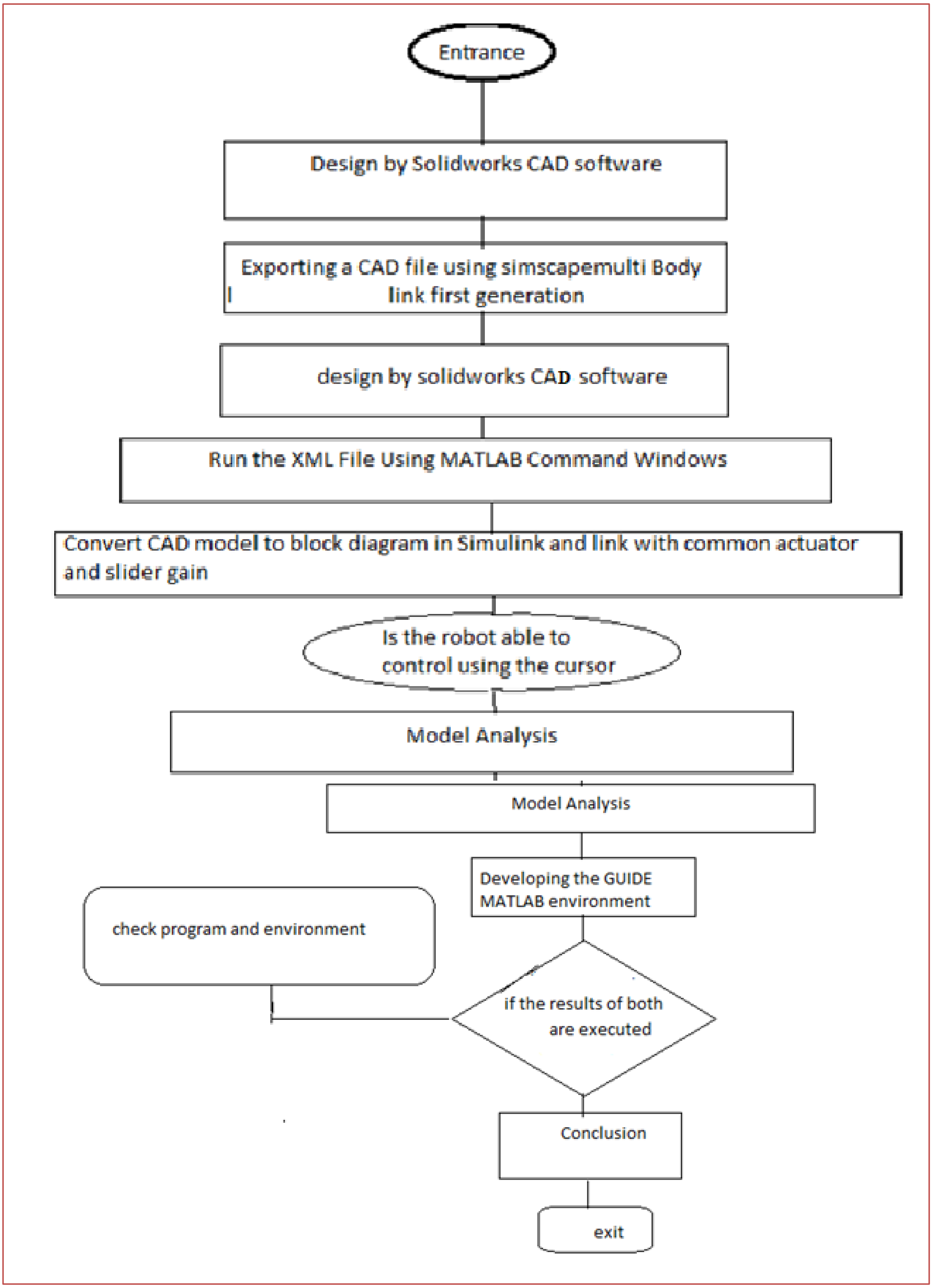


Figure 30: schema of works

3.1 simulation block 3D

The transfer of CAD assemblies from the robot to Sim-Mechanics models is shown in the following **Figure 32,** As you can see this is a sewing model of our Simulink model of a robot, Bat now, the robot is going crazy, so we need to control this, using a P I D and some other mechanism.

This Rivoli joint is the joint of the robot which must provide by input the actuation and the torque is automatically calculated and we must detect the position of the robot.

Double-click on the empty space and type Simulink to physical converter (in put), and physical converter to Simulink (output), then connect in put and output.

Select all this and create a subsystem and we name this robot mechanical model

4.1 Model robot 1

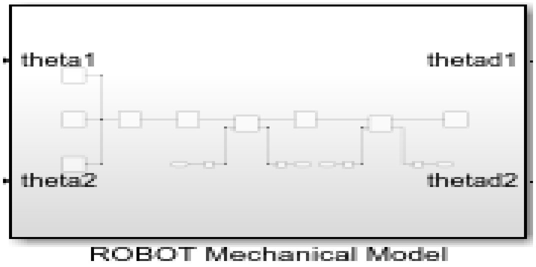


Figure 31: robot mechanical

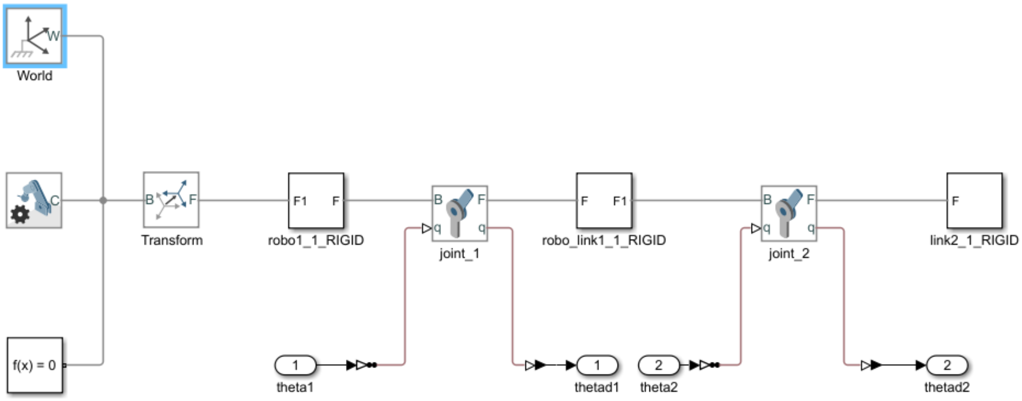


Figure 32: CAD assembly conversion for Sim Mechanics model

The results of study aims to propose a new PID controller with optimally selected again values. The control algorithm has been used for the case of robot or drones' type quadcopter.[40]

To do this we need to go to the Simulink library , and we add the (functions block, clock block ,PID controller, the sum to subtract the error) ,we have to assemble this block, and we have to configurate this blocks (**Appendices 01**).

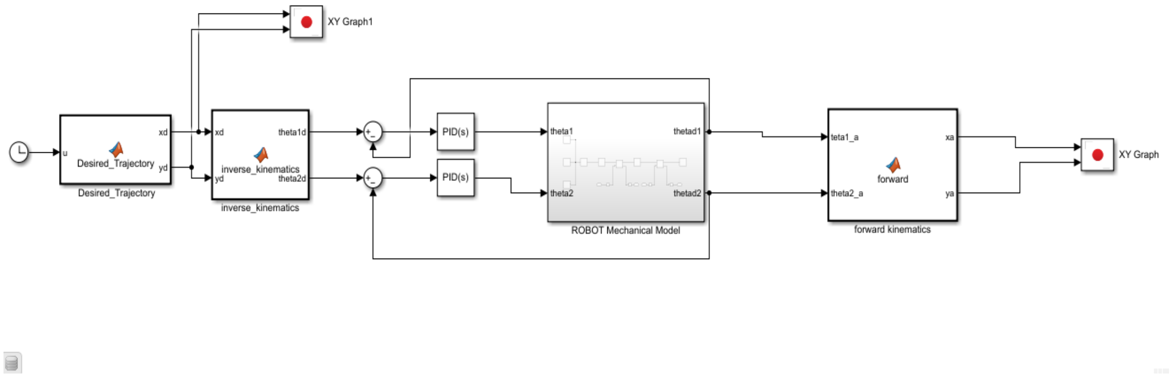


Figure33: model of robot 1with inverse kinematics and forward kinematics

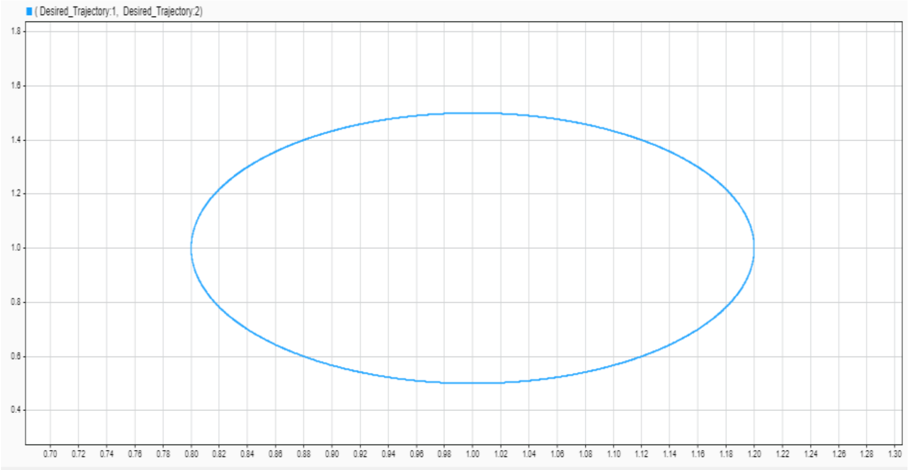


Figure 34: graph 1 represents the trajectory

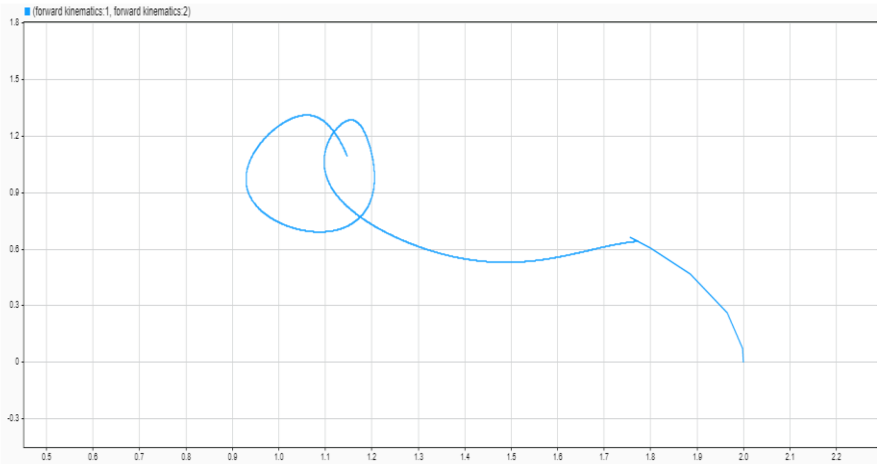
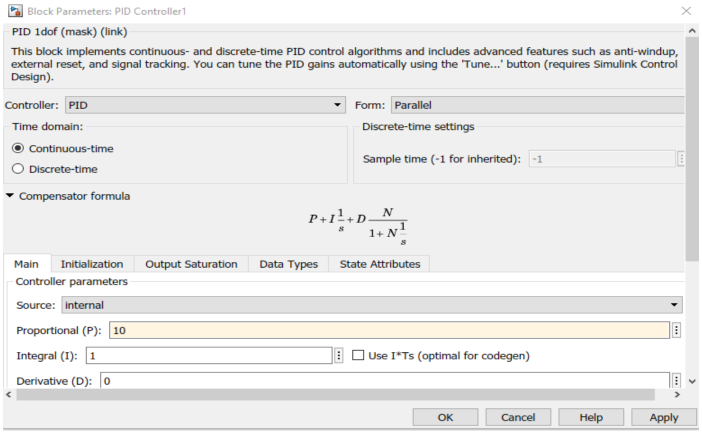


Figure35: graph 2 represented link according to angle

We have to configurated this PID controller to improve the desired trajectory., we have to change some parameters (a proportional–integral–derivative).


Block Parameters: PID Controller1

PID 1dof (mask) (link)

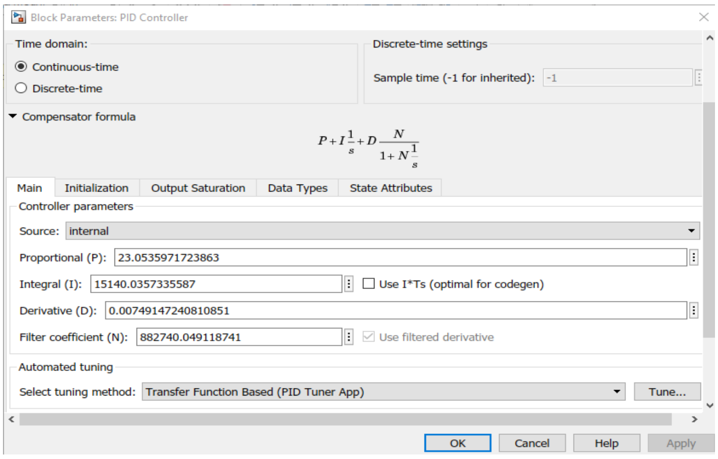
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PID
Form: Parallel
Time domain: ● Continuous-time ○ Discrete-time
Sample time (-1 for inherited): -1

$$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$$

Source: internal
Proportional (P): 10
Integral (I): 1
☐ Use I*Ts (optimal for codegen)
Derivative (D): 0

Figure 36:PID Controller

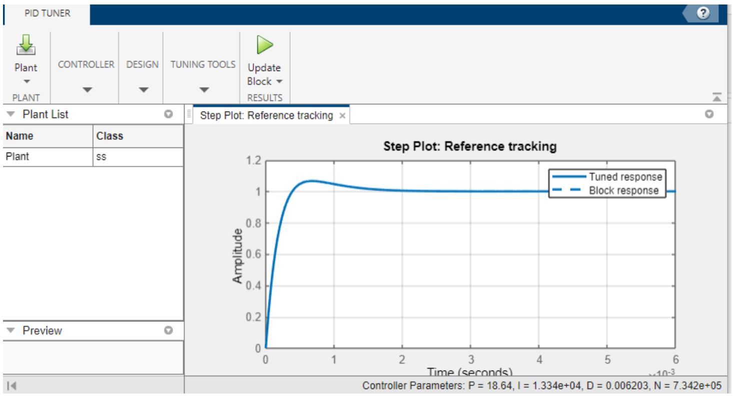

Block Parameters: PID Controller

Time domain: ● Continuous-time ○ Discrete-time
Sample time (-1 for inherited): -1

$$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$$

Source: internal
Proportional (P): 23.0535971723863
Integral (I): 15140.0357335587
☐ Use I*Ts (optimal for codegen)
Derivative (D): 0.00749147240810851
Filter coefficient (N): 882740.049118741
☑ Use filtered derivative
Select tuning method: Transfer Function Based (PID Tuner App)

Figure 37: Configuration PId paramètre


Name	Class
Plant	ss

Controller Parameters: P = 18.64, I = 1.334e+04, D = 0.006203, N = 7.342e+05

Figure 38 : Tuner 1

The results shown in **Figure 39,40** after the configuration of PID controller

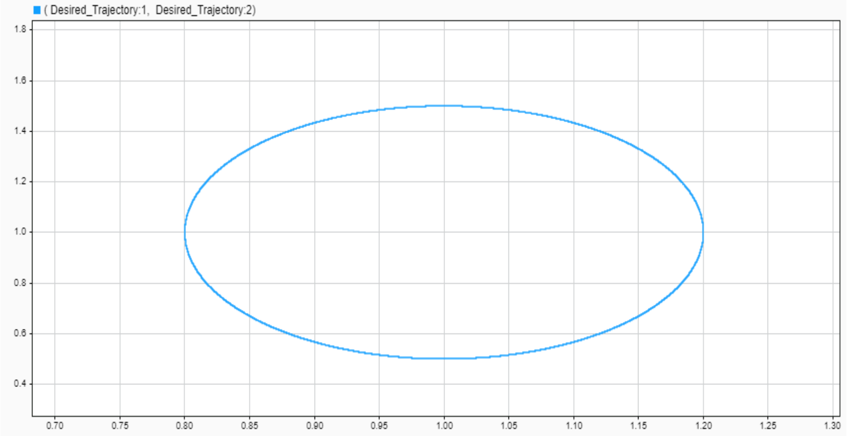


Figure 39: graph 1 represents the trajectory of this robot

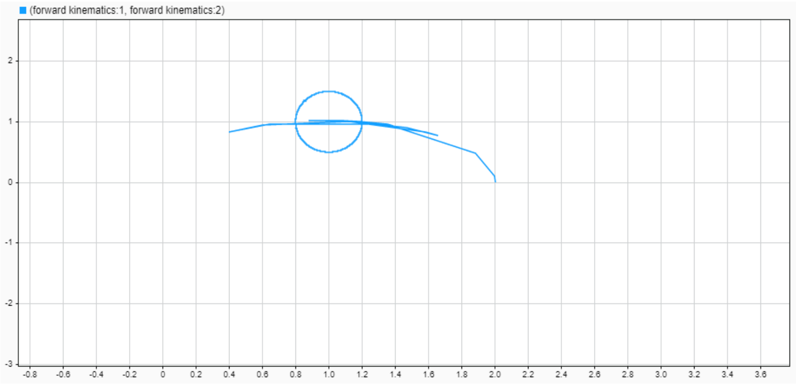


Figure40: graph 2 represents link at the angle Q

SECOND CASE

4.2 Robot Model 02

The robot will have three joints. To implement this robot in the simulated environment we need a tool that will build the required CAD model and insert the joints in the correct location and orientations this will be achieved by the Simscape toolbox after that we need drive this robot into his workspace

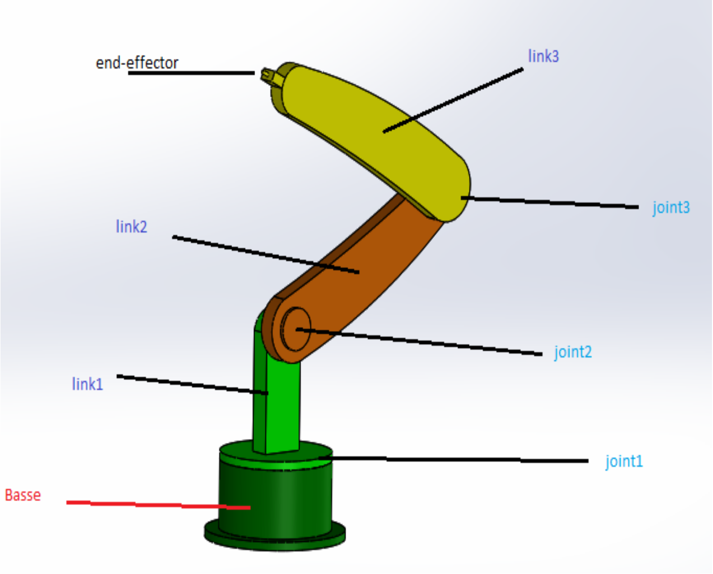


Figure41: représente the robot2

Table 3.2 : Dimension tables

	length (mm)	Extrude(mm)
basse	45	5
Link1	110	10
Link2	150	10
Link3	170	10

4.2.1Forward kinematics

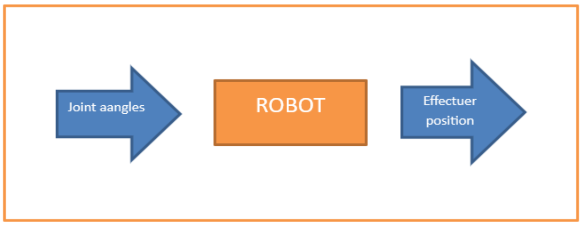


figure 42: Forward kinematics

We can command the joints to rotate at a certain angle which is within the rotational limit of the joint and when the joints have rotated at those certain angles the end-effector would have reached a corresponding point in space working this is what we call forward kinematics algorithm

It is therefore the Simulink environment on which we will work

Then after the base we have a joint which has a frame here which is just above the base and then after

joint one we have a solid which is link one and which is link one and this link one to his frame right in the middle of it, after that we need to add joint 2 at the end of link one, so we need frame transform which will transform the frame straight from the center of link one to the far edge of link one. And same way with link two and link three **Figure43** ,so i will create a subsystem and name these as link two

RRR model

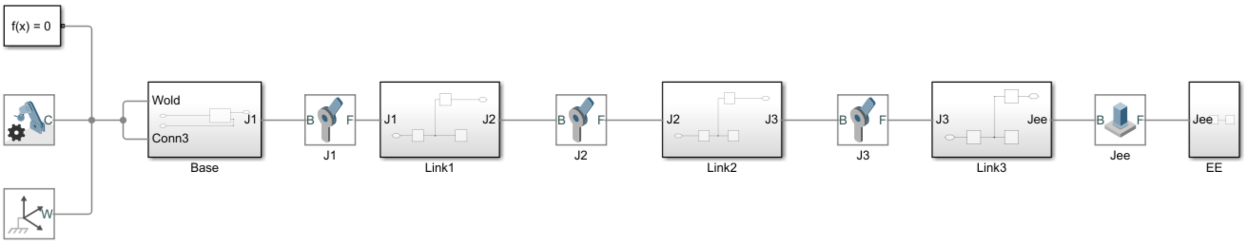


Figure43: Model of Robot RRR

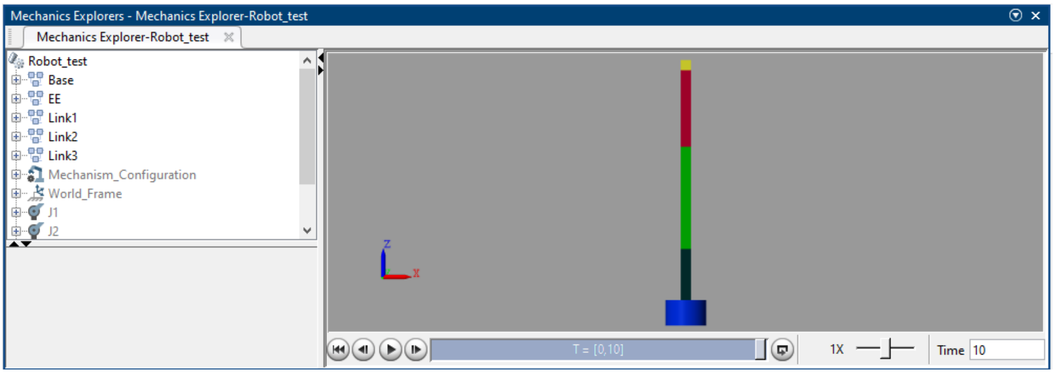


Figure 44: Model of Robot RRR

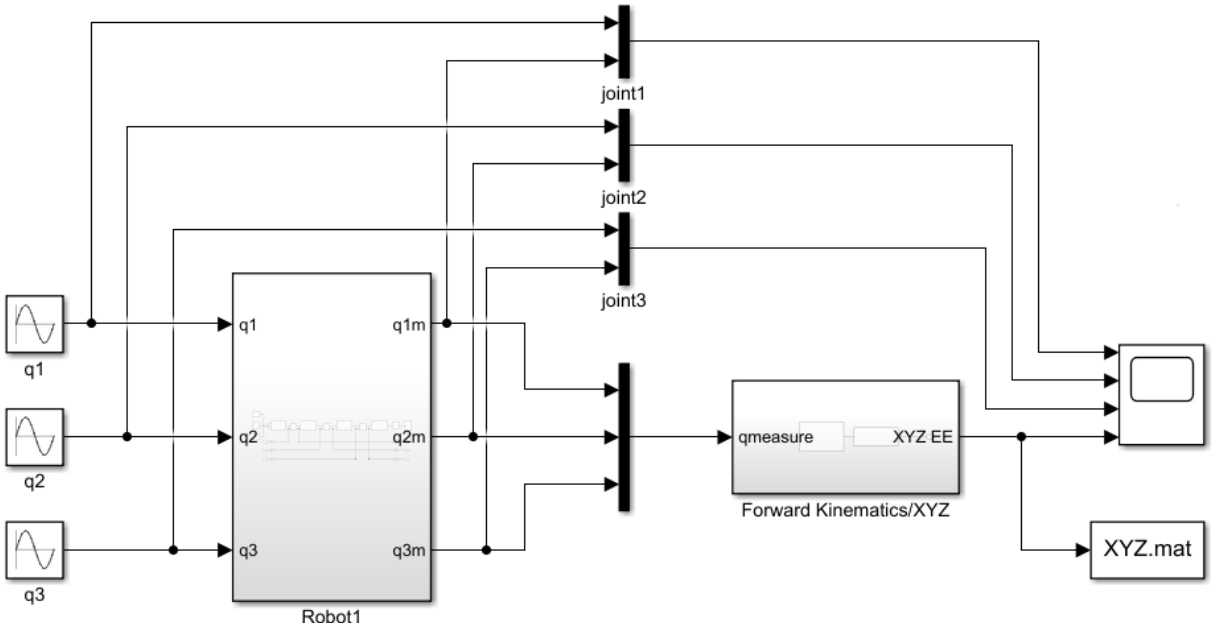


Figure45: Forwarde kinematics RRR

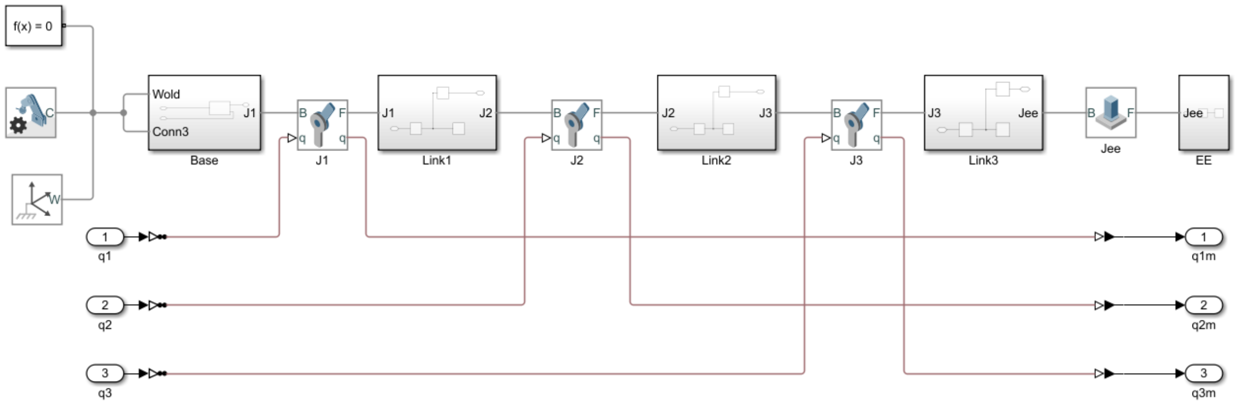


Figure46: robot for position study

Now we want to drive this robot using the robotic system toolbox, so let's first look at this robotic system toolbox,. we can see different options in this toolbox, so we are going to be working on manipulator algorithms. Inside manipulator algorithm we have various algorithms. I need a get transform block, so this block requires two ting numbers

First, it will need the rigid body tree, so it will need to be provided with the rigid body tree that we have already created

And second, the configuration means that what joints or what frames are at what angles

Now this will give me the homogeneous transformation matrix which will contain the orientation and position of the end- effector, so to extract the position of the end-effector to extract the position information from the homogeneous transformation matrix, we need a coordinate transformation conversation block, in side this block ill use the input as homogeneous transformation and we need the output in form of translation vector, so it well gives me the translation vector or the x y z position

So, I hide into one subsystem and name it as forward kinematics. Because it is taking the angles of the robot and giving the position of the end effector

So we hide into one subsystem and name it as forward kinematics. because it is taking the angles of the robot and giving the position of the end effector

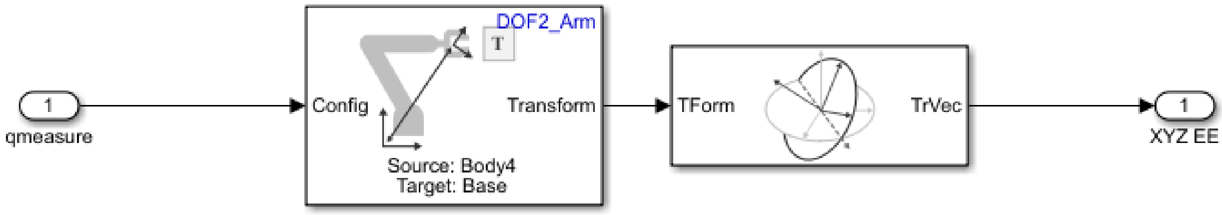


Figure 47: forward Kinematics

we use scope to show all these graphs, and inside the scope we are going chose three different graphs:

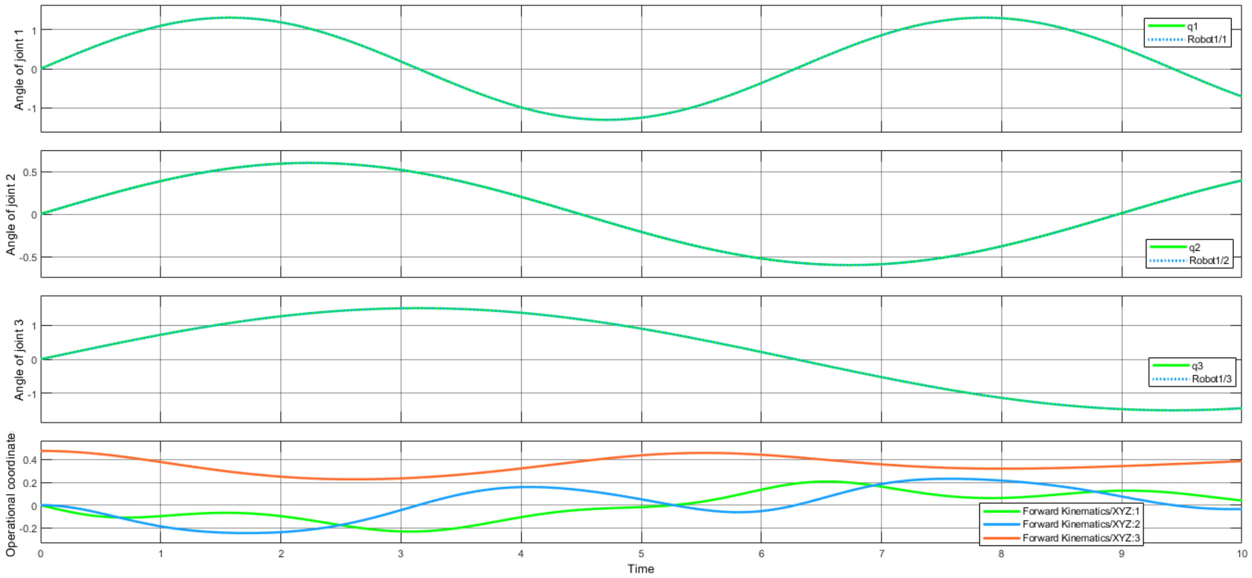


Figure48: presentation of MGD and angles of joints of robot.

Last graph we are shown the X Y Z position of the end effector depending by the time the upper three graph, we are showing the results of deference between joint q and the repines qim

The **Figure49** shous the trajectoiry X Y Z of end effectors in 3D

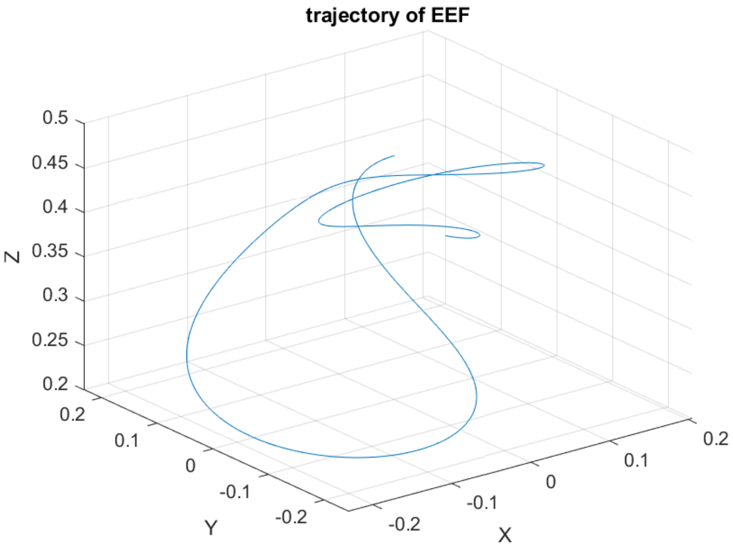


Figure49: Trajectory of EEF of robot

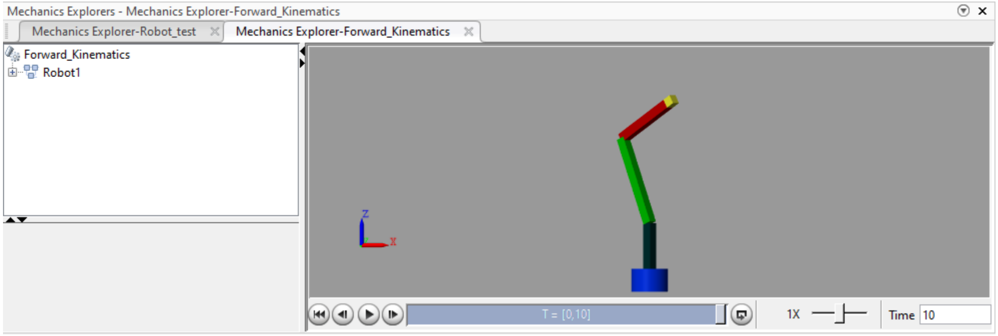


Figure50: Mechanics explorer of Robot in simulation under test of forward kinematics

4.2.2 Inverse kinematics

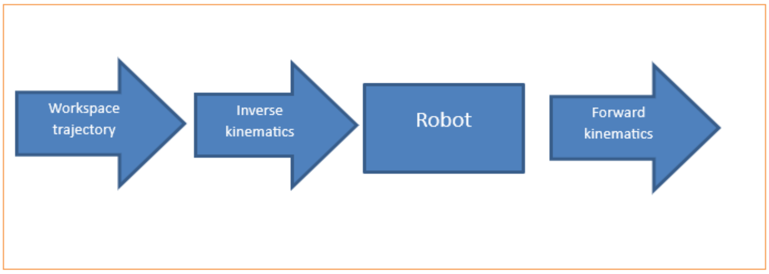


Figure51: schema of work for inverse kinematics

We are providing the trajectory and we know that this trajectory will move the end effector of the robot, we are converting that trajectory into homogeneous transformation Metrix that matrix will be called pose ,the robot will move the movement of the robot will be measured by the sensors which are placed on the joints and then those joint angle measurements . the forward kinematics block will calculate the effectors position,

We can use a signal builder that will generated trajectory.so Inside this signal builder we have different option Of signals, right now we want a custom signal because we want to define the *X Y* Z of the end effector .

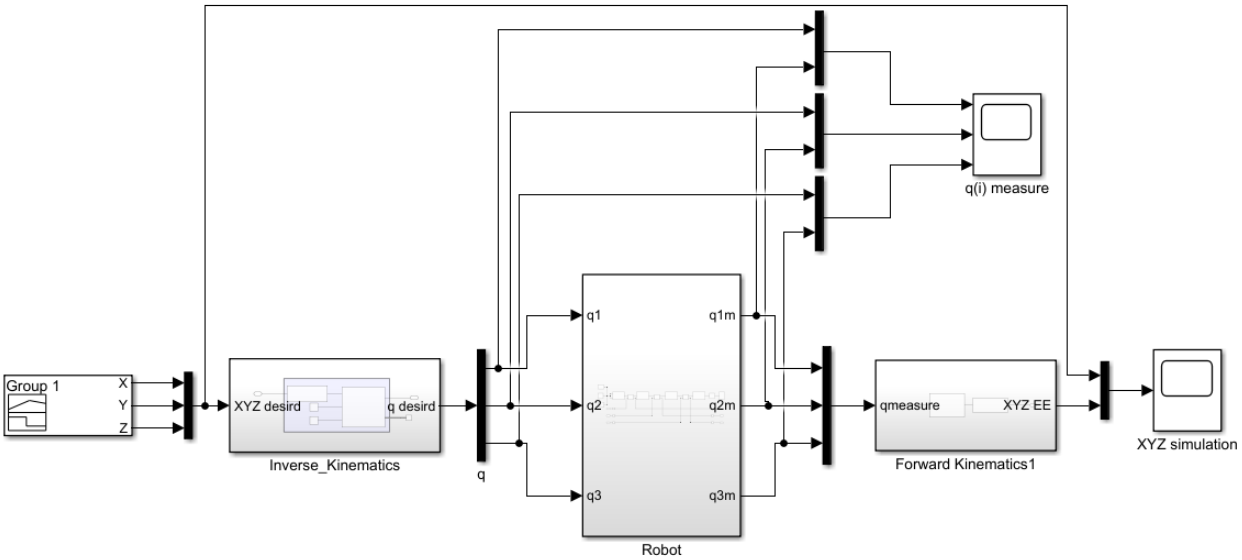


Figure52: Model simulink of Robot for invercs kinematics

So let provide these signals to inverse kinematics block, it can give us the joint angles which will be provided to the robot over here. So for inverse kinematics block I can get in library and it requires threes imports and generates two out puts. The first impute of this block is pose .it men's the end-effector position and orientation end we called with a homogeneous transformation.so to provide the pose to this inverse kinematics block, we need to convert his translation vector in to a homogenous transformation matrix, and for that we need to configuration this block, so in the impute side we will have a transformation vector and on the output side we would have a homogenous matrixes. The second input which is of weights so we need a constant, and in that Constantin going to pass an array it will have their zeros for rotation and that for translation.

So this constant called whit the weights and the third input for initial guess is required so we need another constant that will provide initial guess, In another aside (output) we need a demux for a configuration port and the second output, we blocked because we don't need this port for anything so I am going to terminate it using a terminator **Figure53**.

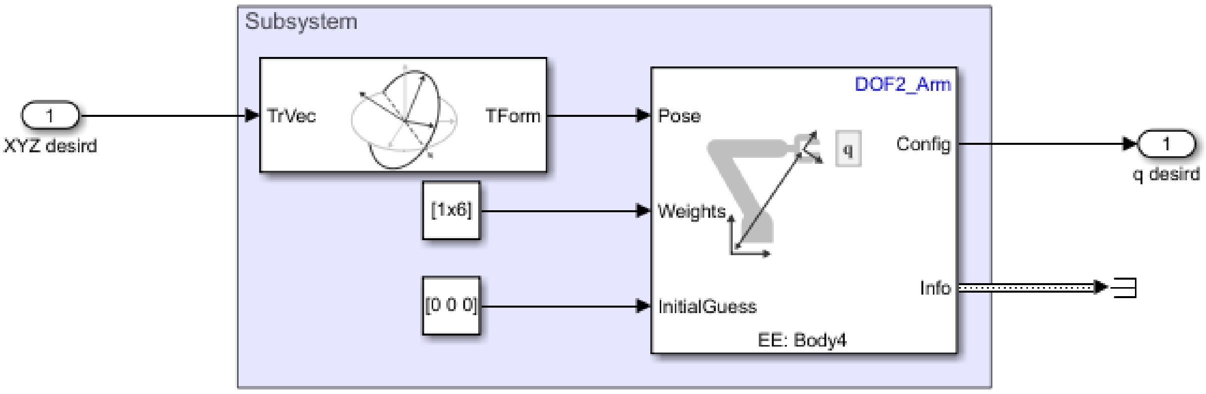


Figure53: Inverse Kinematics Model

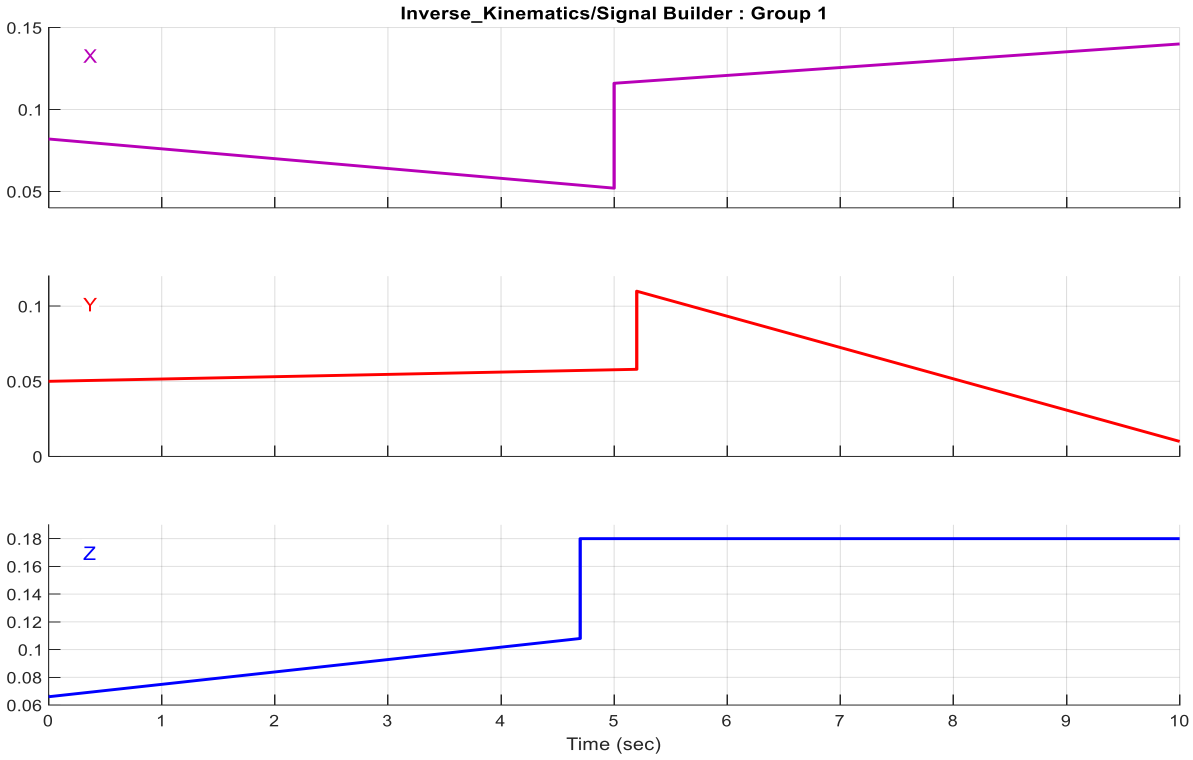


Figure 54: Desired trajectories chosen for X, Y, and Z.

The following **Figure 54** shous the changes of desired trajectories chosen X, Y, and Z.in terms of time.

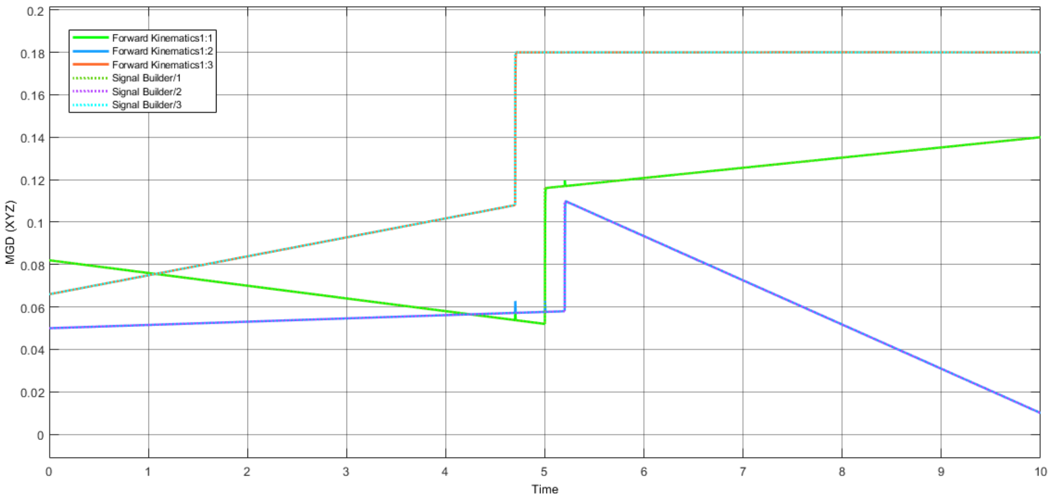


Figure55: responses of X, Y, and Z under simulation.

The following figure shows the results between the trajectory x y z represented by the dashed line and the q1m line represent by continuous line under simulation .

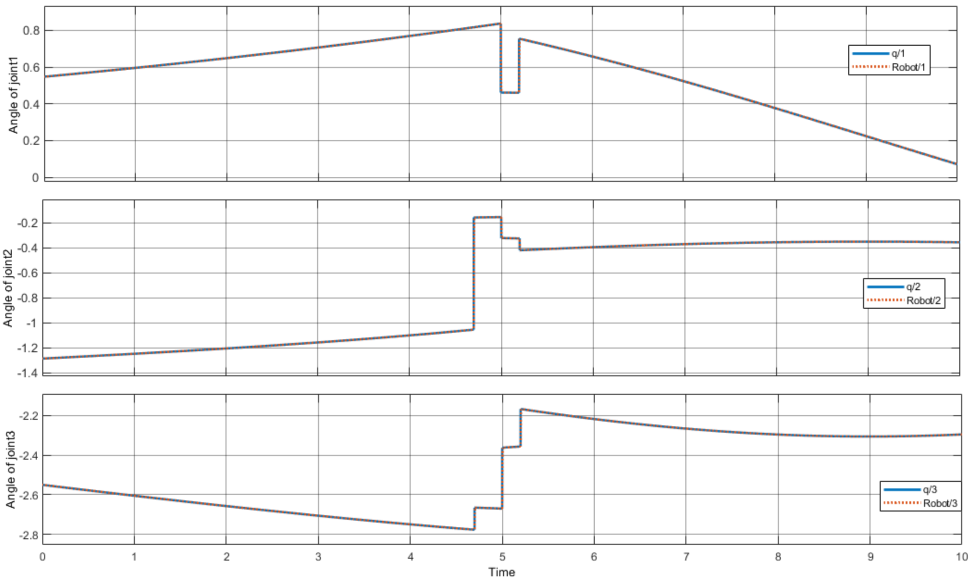


Figure56: invers kinematics of joints coordinates.

The following **figure56** shows the results between the q_i represented by the dashed line and the q_im line represent by continuous line under simulation depending on the time .so the q_i represent the output of the invers kinematics block. And the qim represent the output of robot under simulation.

4.2.2.1Trapezoidal velocity

The trapezoidal velocity is very useful if your robotic manipulate wants to move in linear motion that is it wants to traverse in straight lines and in our case we want our manipulator to move in a straight line.

The input for this block is time, and we connected the output of this block with the inverse kinematics for calculate the joint angels. This block generate three output, thirst the positions the $X\ Y\ Z$ trajectory positions ,and the second output velocities, and the last output its accelerations

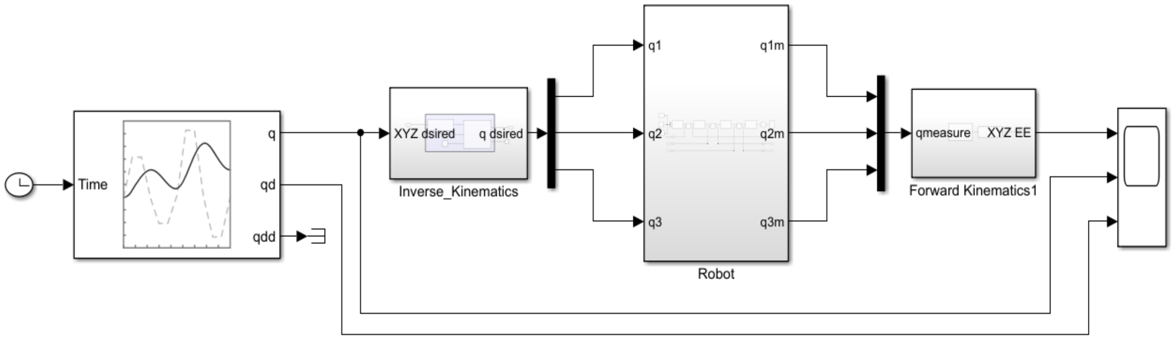


Figure 57: Generate trajectories through multiple waypoints using trapezoidal velocity profiles

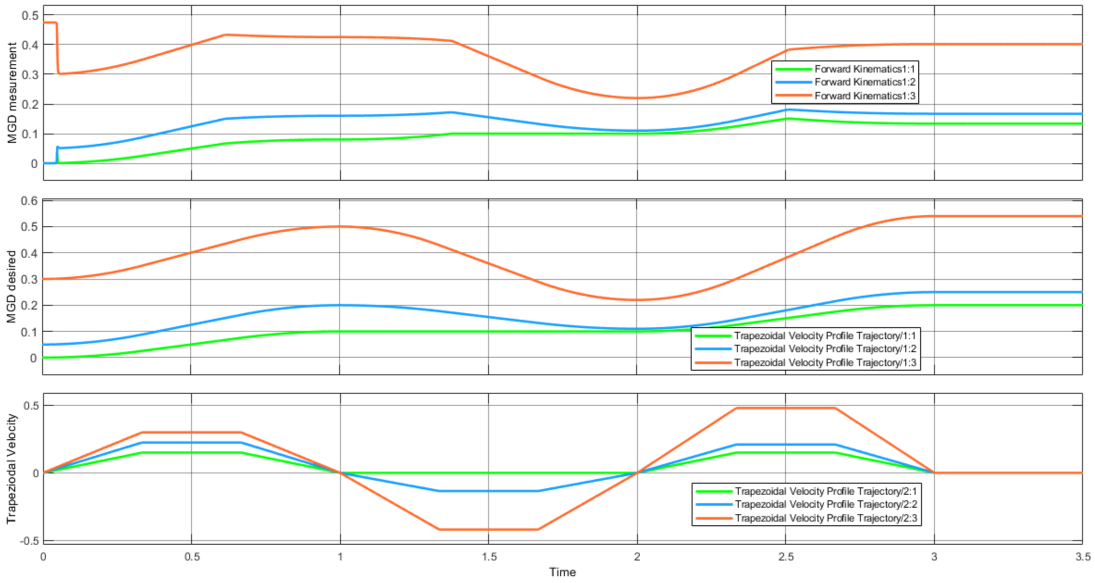


Figure58: simulation of trajectories through multiple waypoints using trapezoidal velocity

The results are shown in figure **Figure58**

Lasted three segments are being completed in 0.5 seconds then in one second and then once a gene in one second so these two straight lines of these square [1,2] are being completed in one second and then the next

The first graph represent the results of forward kinematics depending on the time, so we notice that the red line at the beginning was at the position 0,5and on the same time fill to the point 0.3 that because the trajectory chosen it was from the point 0,3.

4.2.2.2Polynomial trajectories

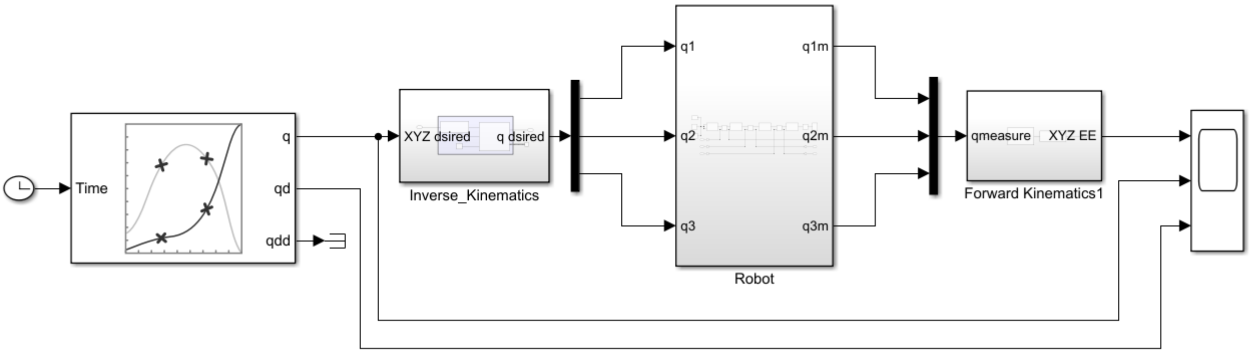


Figure59: Generate polynomial trajectories through multiple waypoints.

We are going to choose the polynomial trajectory block, this block require three trajectory in output :

✓ The first is the trajectory coordinates.

✓ The second is the velocities.

✓ And the third is the accelerations.

q here doesn't men's joint angles **q** represent the **X Y Z** position of the trajectory, **Qd** represent the velocity and the **qdd** represent the accelerations, this is all represent the output, in input that require the time, I'm going to place a clock block in input of the block.

In the first graph, and the second graph it showed the positions which are generated by the trajectory block, and the last graph show the velocities graph

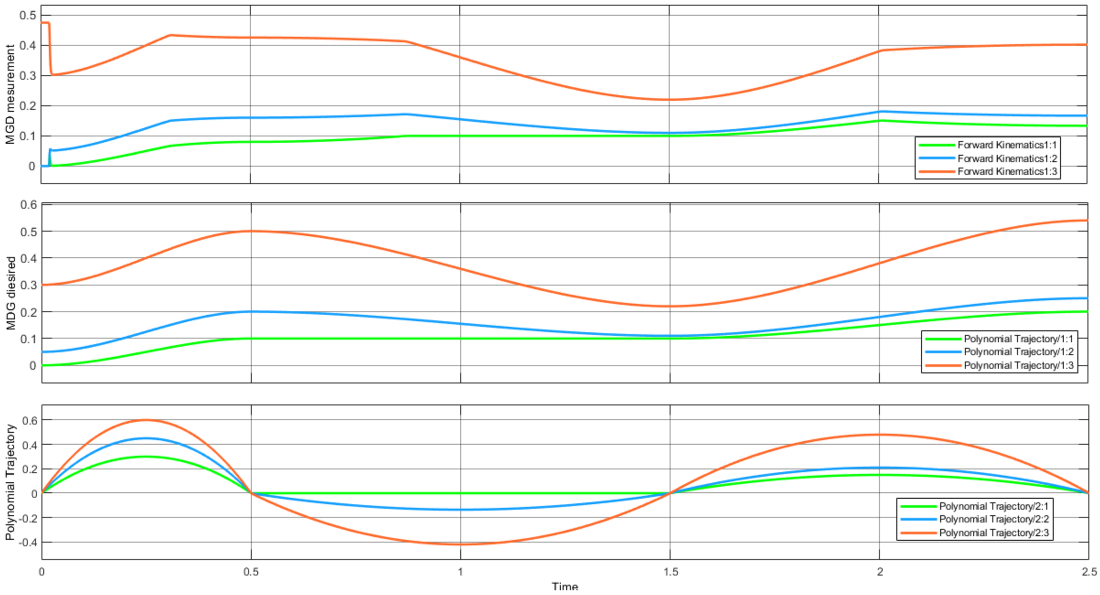


Figure60: Simulation polynomial trajectories through multiple waypoints

The first graph represent change of forward kinematics depending on the time. Its show the position of the X Y Z axes of the end effector which is the forward kinematics block was measuring.

The second graph represent the trajectory depending by time. Its shows the trajectory that was generated by the trajectory generation block

Third graphs shows the velocities in X Y Z directions yellow is the velocity in **X** direction, blue is the velocity in **y** direction and red is the velocity in **z** direction depending by time .

4.2.3. Jacobian

At the start we were generation the trajectory and providing to the inverse kinematics block, this block was converting that trajectory into joint angles which were being provided to the robot, then we were measuring the joint angles of the robot providing those joint angles to the forward kinematics and forward kinematics will convert the joint angles into the end effector positions or the trajectory. Then we were comparing the trajectory on which the robot was moving

The forward velocity kinematics block will transform joint velocities into end effector velocities

The inverse Jacobin kinematics can be used for effector velocities

In the robotic system toolbox, we have a block name get Jacobian, this Block is going to provide the Jacobian matrix, and in the other side it is going to give me a Jacobian. We need to detect joint velocities so for that we need to enter the robot and we need to detect joint velocities exactly the same way we detect joint position

First we need to maxi that we can have a matrix of velocities, and for the matrix multiplication I will use a matrix multiplication block, the first matrix will be the Jacobin and the second matrix will be that of my final effector vector of velocities so that's all now I'm going get six velocities related to my end effector, out of these first three will be the angular velocities of the end effector and the last three velocities will be the linear velocities of my end effector, I'm going to compare these velocities linear with the linear velocities of the trajectory, so I split them into two parts the upper half and the lower half the lower half are the linear velocities and the upper half is the angular velocity, I am using a terminator and will provide the coating velocities got the scope

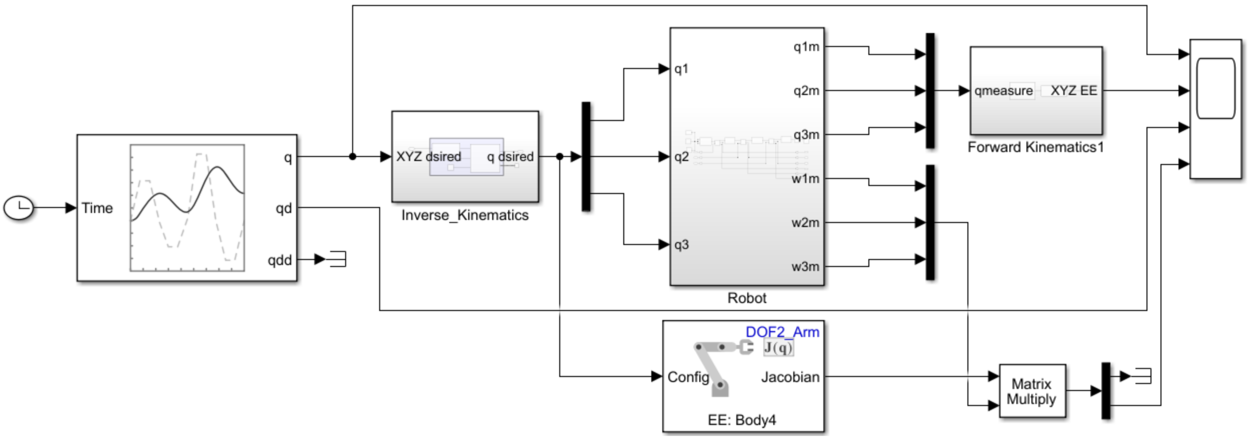


Figure61: Modal Jacobian of robot with Trapezoidal Velocity

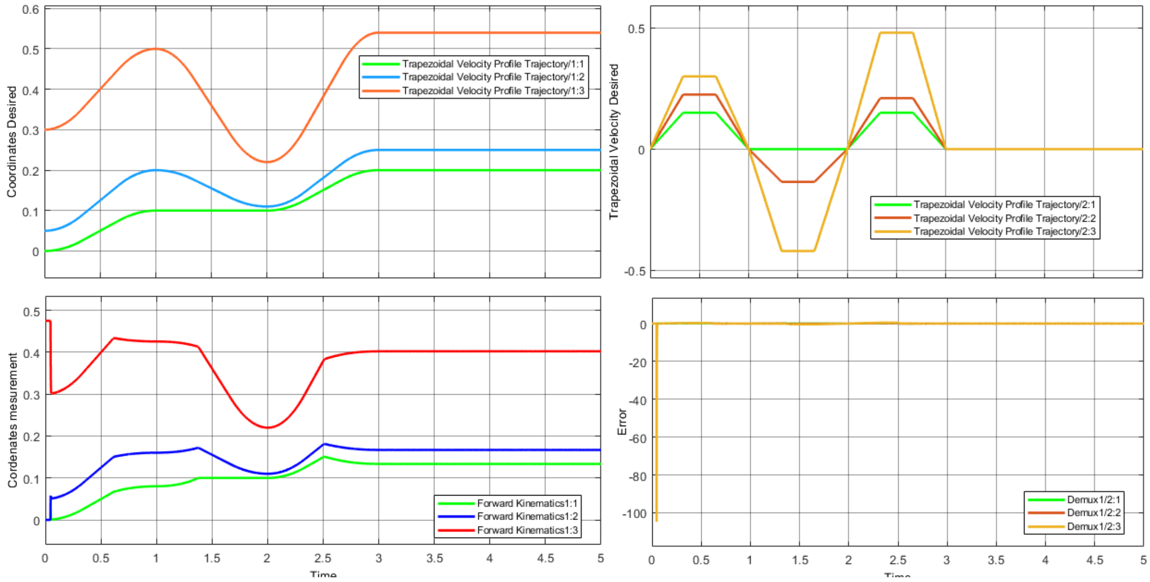


Figure62: Results of simulation with jacobian.

The result depicted in **figure 62:**

The first graph represent change coordinates desired depending on the time the second graph represent the change of the trapezoidal velocity in function of time.

The third graph represent the forward kinematics under simulation, we remark the red line at the beginning was at the position 0.5and on the same time fill to the point 0.3 that because the trajectory chosen it was from the point 0.3.

In the last graph shown the (ERROR).

4.2.4. Dynamics

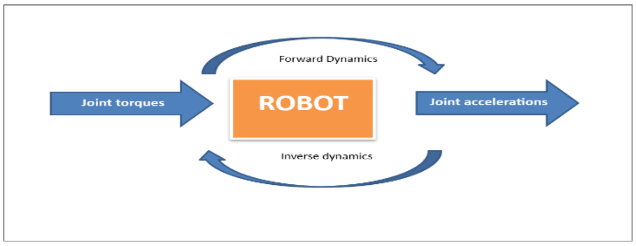


Figure 63: Robot dynamic diagram

This figure36.3 exposes the way of working of the system dynamics. To generate a trajectory so we need a trajectory generator block and we have configured this block the next thing we need inverse dynamics block from system toolbox composed of four inputs (configuration, joint velocities, joint accelerations and external forces acting on the robot).

In the external forces we used a constant that we will configurate .There will be four bodies plus the force and the torques, they will move the robot into X Y Z directions and evolves around the X Y Z axes

Now this block will generate the joint torques necessary for the robotic manipulator to move on the desired trajectory, we can now provide these torques to the forward dynamic block or the original robot or the physics system robot and see how these things are moving so if we provide these torques to the forward dynamic block, now this block will generate the joint accelerations, this joint acceleration can be integrated to get the joint velocities and the joint angles,

We see the forward dynamics block is exactly the opposite of this inverse dynamics block, so whatever trajectory we are providing over a trajectory generator block we are going to get that trajectory out over these scops, The forward dynamics block required three other things one is the same as this (external forces, the configuration which is coming from trajectory generator and the joint velocity which coming for joint velocity)

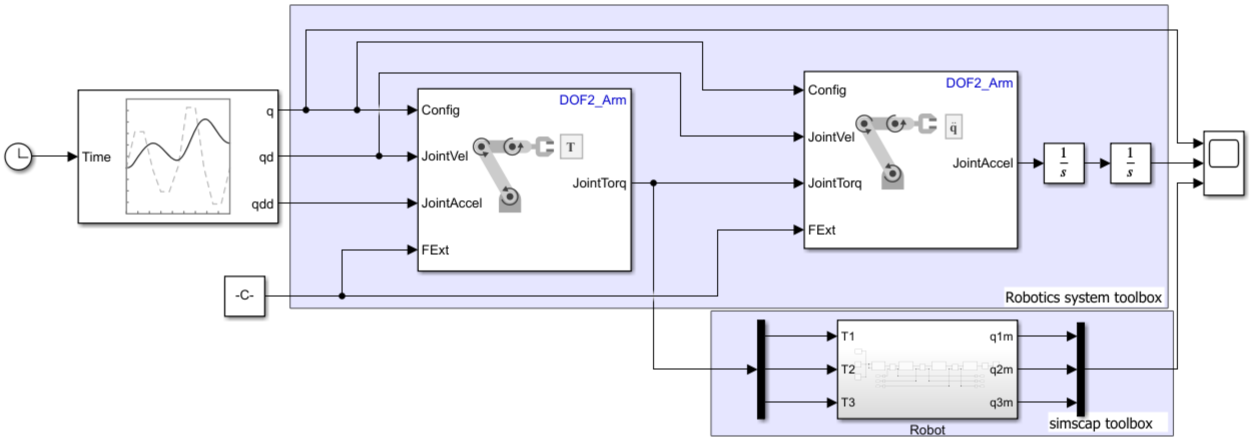


Figure64: Block Simulink of Robot under dynamics study.

Now, on the other side, I can also feed this conversation to my physical robot and see that this torque is enough for my robot to follow the desired trajectory. For This I need a robots physiques system. This robot has three joints, and end effectors as well and there links .there is the base the link but this robot is not torque operated, now movement this time because we are going to provide torques to the joints so for that I have to go inside that joint and into the actuation section select provide per input for torque and for motion you can calculate it automatically.

In the scope, the first input would be the trajectory or the joint configuration which we are providing as a reference, the second one is being generated from the forward dynamics block the last one is being generated from a physical system robot.

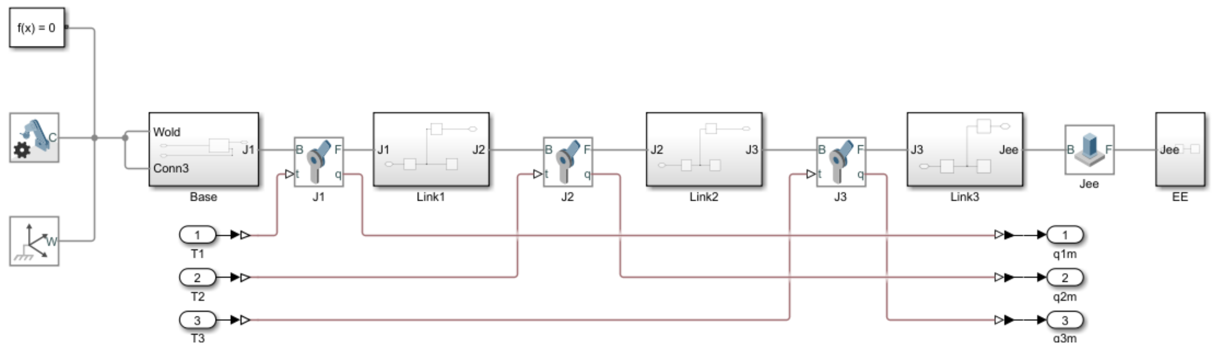


Figure65: model of Robot

As you now the forward dynamics block is exactly the opposite of inverse dynamics so first tow graphs should be similar if this physical model is exactly what is being represented by this inverse dynamics, then whatever torque is required for this model to move.

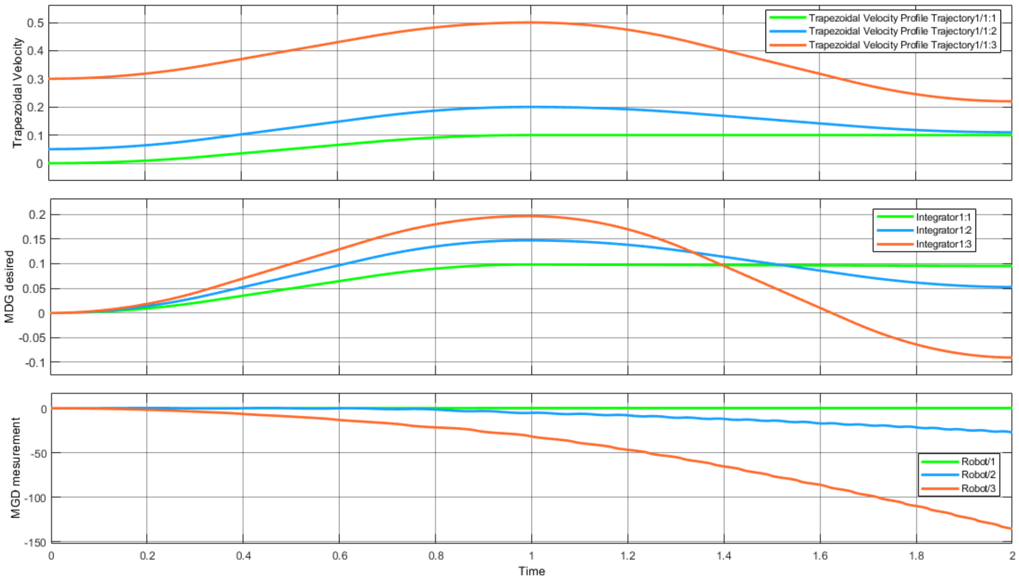


Figure66: Dynamics response of three joint.

So in this figure 66, explained that: the first graphs shows the trapezoidal velocity depending on the tim which we wanted our robot to move. The second graph represent the forward dynamics block generated that is provided the torques produced by the inverse dynamics block to the forward dynamics block and it generated these trajectories you can see that this these two trajectories are exactly the same whereas .and the and the last is the trajectory followed by the physical system robot and it is quite clear that robot was not moving on our desired trajectory

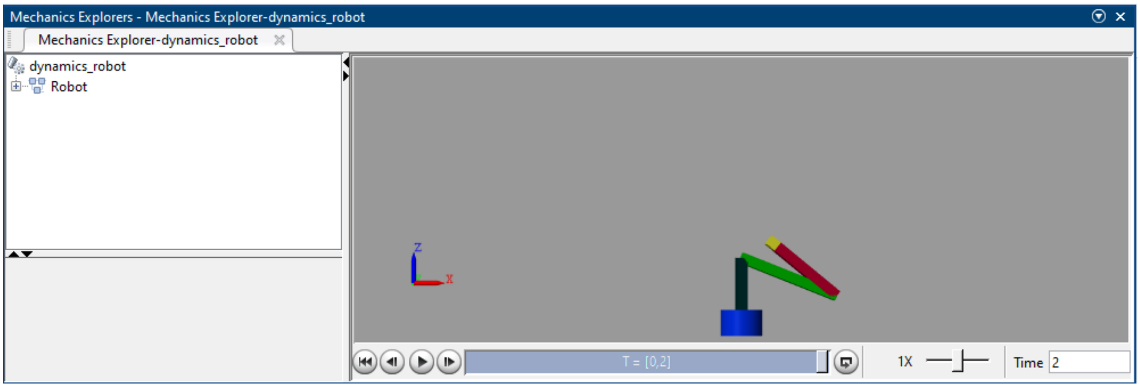


Figure67: Mechanics explorers of robot under Dynamics study.

Here comes the need for a controller, a controller will be amelioration the trajectory from the trajectory generator block will be using of dynamics of the robot

5. PID controller

Fractional calculus has been studied for over three centuries, and it has multifarious applications in science and engineering. This review investigates its progress since the first reported use of control systems, covering the fractional PID proposed by Podlubny in 1994, and is presenting a state-of-the-art fractional PID controller, incorporating the latest contributions in this field. It highlights developments in the field of fractional PID controllers, including their design and tuning, as well as explores their various versions.[41, 42, 43]

In this regard, the present study aims to propose a new PID controller with optimally selected gain values. The control algorithm has been used for the case robot [42, 44, 45].

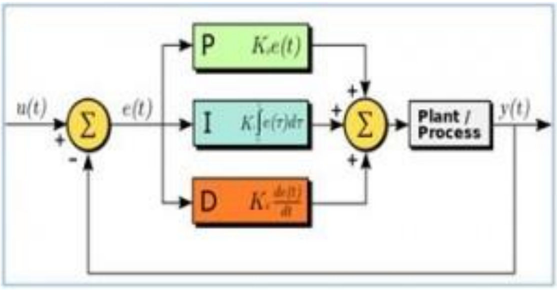


Figure 68: PID controller.

This **Figure 69** explainers the block of PID controller so this block require a reference signal which is coming from signal builder (**Appendices 02**). In this block signal builder that is creation three signals and we are going to provide these step signals as joint angles to the robot, this robot it will require three outputs, the first output for positions, the second output for velocities and the last output for torques. I don't us these two outputs for that I used the terminated

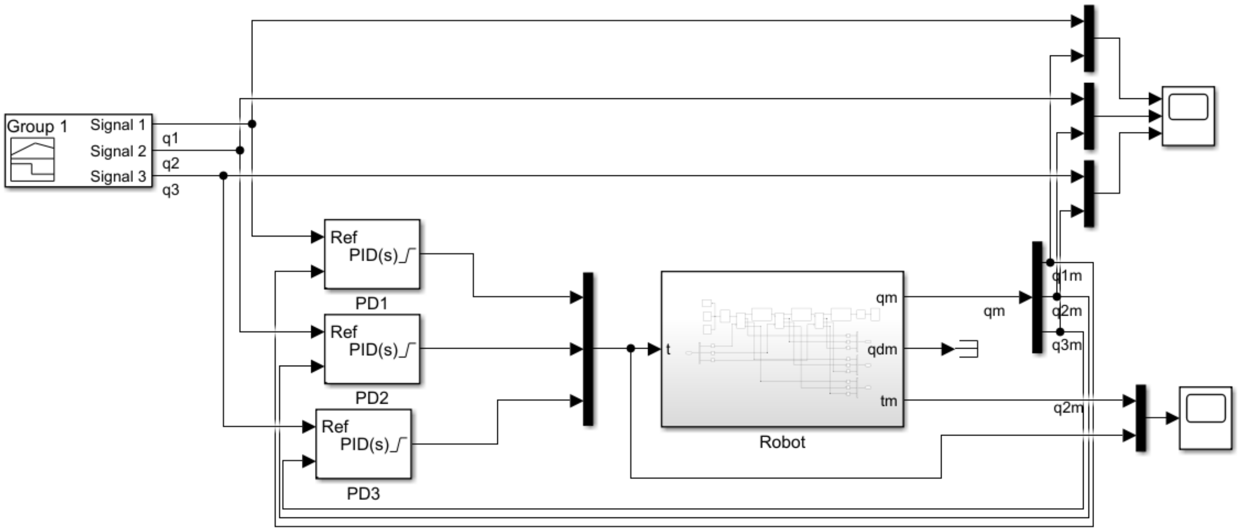


Figure 69: block of PID controller

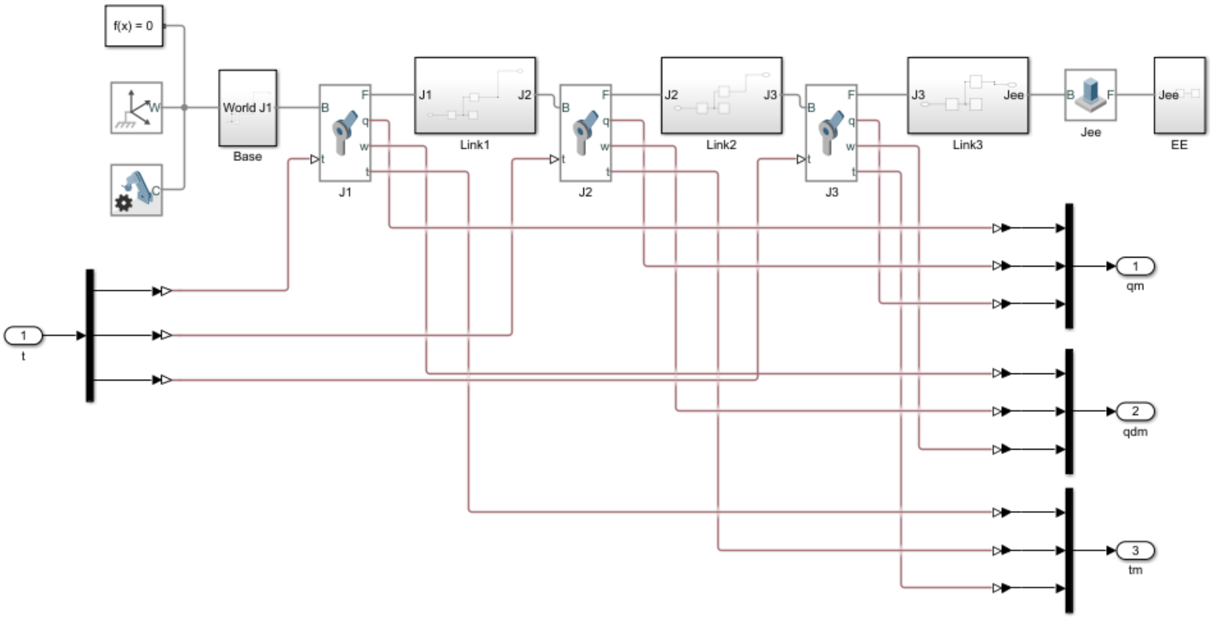


Figure70: block Simulink of Robot.

The results are shown in the figure70:

Thes three graphs represent the joint angels q_i whith the repense of PID controler depending on the time, the red linereprisent q_i,the blue line represent the reponse of PID controler, in the real same time

The first graph for the joint one .The second graph for the joint two . The therd graph for the joint thre

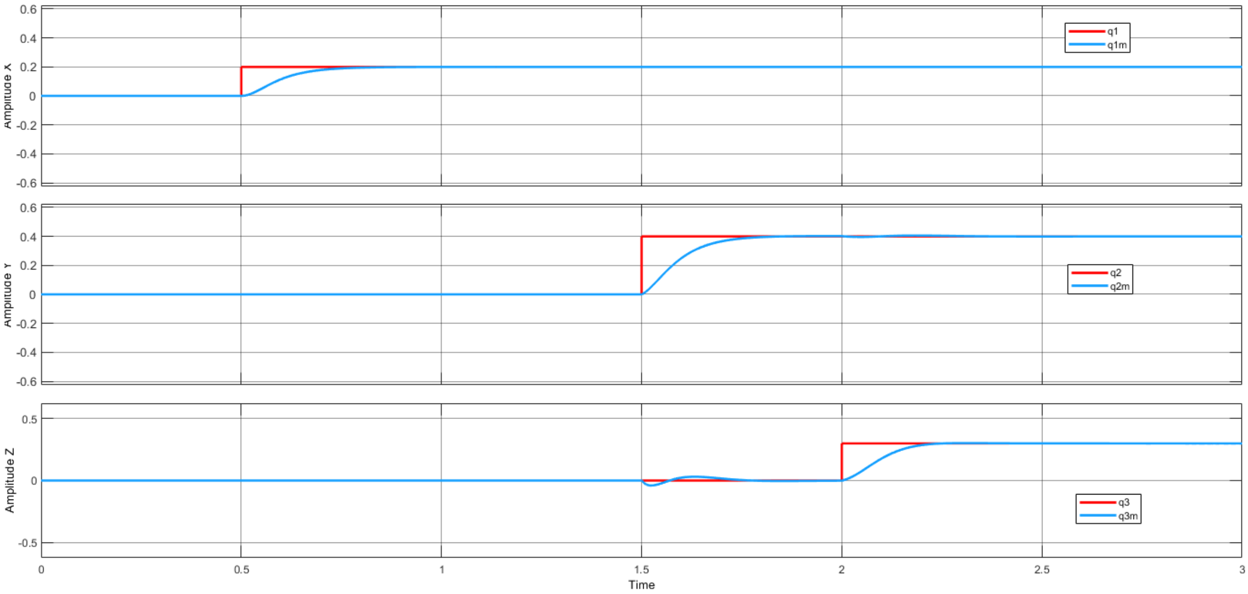


Figure71: response of MGD with PID controller.

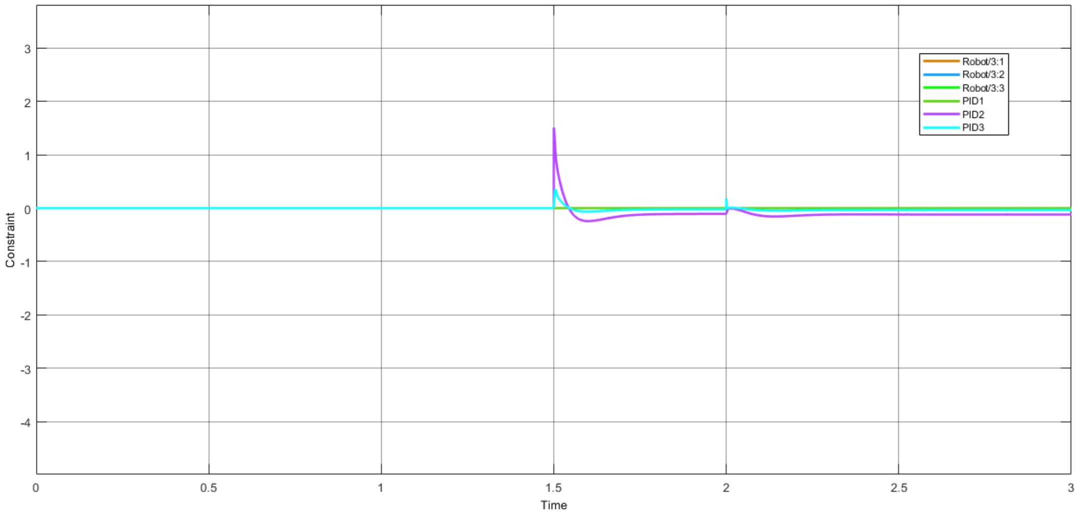


Figure72: Constraint response of joints.

This figure 72 explains the repines for physical system of robot and the PID controller depending on the time we note this graphs lead to zero because the PID controller improves the error.

General Conclusion

The main object of this memory was the creation of a model with a graphic interface in order to present a mechanical structure of our robot and in the end Simulink facilitates us the connection of all the parameters necessary to make the simulation of the robot possible by using the SolidWorks software in combination with Sim-Mechanics and MATLAB Simulink. This interface of Solid Work with Sim-Mechanics is done using ‘Sim-Mechanics-link’. In our study we presented a forward kinematics model and inverse kinematics model and dynamics model and Jacobian. The establishment of a forward kinematics model and inverse kinematics model and dynamic and Jacobian model constitute the first step.

The resulting equations express the relationship between the joint configuration of the robot and the position (in the Cartesian reference frame) of the extremity of the wrist using the Denavit-Hartenberg method. The inverse problem which allows knowing the articular variables according to the situation of the terminal organ by using the method of Paul.

This graphic interface allows us to obtain the movement resulting from our robot which is obtained by composition of the elementary movements of each link compared to the previous one.

This graphical interface allows us to obtain the geometric configuration of the robot in three dimensions.

PID regulators allow us to improve the motion performance for each element of our robot.
This study ends with the derivation of the solutions to the inverse problem, which consists in the determination of the articular variables corresponding to the position of the given effector. First, it should be noted that the modeling does not stop there. As part of the work, the electric motor of one of the joints was modeled.

The results obtained in both cases demonstrate the quality of the efficacy of both options, either to study the robot directly through MATLAB or to use SolidWorks for its installation, and this latter is very practical in case of difficulty in designing the robot using MATLAB or complexity of model.

Appendices 01

Robot1

Desired trajectory

```
function [xd,yd] = Desired_Trajectory(u)
xd=1+0.2*sin ((2*pi/5)*u+pi/2);
yd=1+0.5*cos((2*pi/5)*u+pi/2);
```

Inverse kinematics

```
function [theta1d,theta2d] = inverse kinematics(xd,yd)
l1=1;
l2=1;
theta2d=acos((xd^2+yd^2-l1^2-l2^2)/(2*l1*l2));
theta1d=atan(yd/xd)-atan((l2*sin(theta2d))/(l1 + l2*cos(theta2d)));
```

Forward kinematics

```
function [xa,ya] = forward(teta1_a,theta2_a)
l1=1;
l2=1;
xa=l1*cos(teta1_a)+l2*cos(teta1_a+theta2_a);
ya=l1*sin(teta1_a)+l2*sin(teta1_a+theta2_a);
```

Appendices 02

PID controller

```
%% Load Rigid Body Tree from SimScape model
[DOF3_RRR,ArmInfo] = importrobot('DOF3_RRR');


%% Creating slTuner and configuring it
% Create slTuner interface
TunedBlocks = {'PID1', 'PID2', 'PID3'};
ST0 = slTuner('DOF3_PD',TunedBlocks);
```

```
% Mark outputs of PID blocks as plant inputs
addPoint(ST0,TunedBlocks)

% Mark joint angles as plant outputs
addPoint(ST0,'Robot/qm');

% Mark reference signals
RefSignals = {...
  'DOF3_PID/Signal Builder/q1', 'DOF3_PID/Signal Builder/q2', 'DOF3_PID/Signal Builder/q3'};
addPoint(ST0,RefSignals)

%% Defining Input and Outputs and Tuning the system
Controls = TunedBlocks;      % actuator commands
Measurements = 'DOF3_PID/Robot/qm';  % joint angle measurements
options = looptuneOptions('RandomStart',80','UseParallel',false); % Optimization
% routine will restart 80 times from random locations, and if 'UseParallel'
% is true, then parallel processing will be used to speed up the tuning process
TR = TuningGoal.StepTracking(RefSignals,Measurements,0.005,0);
ST1 = looptune(ST0,Controls,Measurements,TR,options);

%% Update PID Block

writeBlockValue(ST1)

%% Simulating the model
 sim('DOF3_PID.slx',3)
```

Bibliography

[1] HAMDI, H. (2002-2003). *Introduction à la robotique*. Les éditions de l'université Mentouri_Constantine.https://telum.umc.edu.dz/pluginfile.php/10400/mod_resource/content/1/intro%20robotique_ver_etudiant.pdf

[2] Stone,W.L.(2018).*The history of robotics*. In Robotics and automation handbook (PP.8-19).CRC Press.

[3] Royakkers, L., & van Est, R. (2015). *A literature review on new robotics: automation from love to war*. *International journal of social robotics*, *7*, 549-570. https://link.springer.com/article/10.1007/s12369-015-0295-x

[4] U.K. MoD, Joint Doctrine Note 2/11 *the UK Approach to Unmanned Aircraft Systems, UK MoD The Development*, Concepts and Doctrine Centre, SWINDON, Wiltshire, 2011.

.[5] R.J. Bachmann, F.J. Boria, R. Vaidyanathan, P.G. Ifju, R.D. Quinn, *A biologically inspired micro-vehicle capable of aerial and terrestrial locomotion*, Mech. Mach. Theory 44 (2009) 513–526.

.[6] M. Hassanalian, H. Khaki, M. Khosrawi, *A new method for design of fixed wing micro air vehicle*, Proc. Inst. Mech. Eng. J. Aerosp. Eng. 229 (2014) 837–850.

.[7] Satla, Z. & Encadreur Elajrami, M. (2019). « *Contribution À La Modélisation Et À La Commande D'un Drone Miniature* » [Thèse de Doctorat, Université Djillali Liabès - Sidi Bel Abbès].https://www.theses-algerie.com/2519265048435545/these-de-doctorat/universite-djillali-liabes---sidi-bel-abbes/contribution-%C3%A0-la-mod%C3%A9lisation-et-%C3%A0-la-commande-d-un-drone-miniature.

[8]EL BAZ ,O.(2019), *CATIA rendering 3Dprinting,Eductional, Machine design* https://grabcad.com/library/bras-robotique-5

[9]Deep, A.(2020) more on Sarthaks.com - https://www.sarthaks.com/885684/discuss-the-functions-of-key-components-in-robots

[10] par Olivier de La Barrière (De la Barrière) sous la direction de Mohamed Khémis Gabsi (2010).*étude théorique et expérimentale des pertes magnétiques dans les matériaux granulaires*.[Cachan, Ecole normale supérieure] .

[11]Darrell ,J .(2016) .*Senior Technologies Advisor*.

https://www.allworldmachinery.com/blog/75/what-is-an-actuator-

[12] Pallas-Areny, R., & Webster, J. G. (2012). *Sensors and signal conditioning*. John Wiley & Sons.

[13] Julien,H.(2006) .*description et utilisation des micro rupteurs* https://pobot.org/Interrupteur-Microswitch.html.

[14] Phillips, J. (2007). *Mechanism and the mobility of mechanism. In Freedom in Machinery* (pp. 6-25). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511751745.003

[15] Robert , B . *Les principaux mécanismes". L'École nationale supérieure des arts et métiers* https://www.universalis.fr/encyclopedie/mecanique-mecanismes/3-les-principaux mécanismes/

[16] Ryuh, B. S., & Pennock, G. R. (1988). *Accurate motion of a robot end-effector using the curvature theory of ruled surfaces*. https://asmedigitalcollection.asme.org/mechanicaldesign/article abstract/110/4/383/433383/Accurate-Motion-of-a-Robot-End-Effector-Using-the

[17] _Chakraborty, E . (2023*) Geeks Lambda* | https://fr.lambdageeks.com/robot-end-effector/

[18] BENMISRA, Abd . (2007) *Programmation des robots industriel et application sur le robot manipulateur Algérie machines*.Université de Saad Dahleb de Blida (Algérie) - Magistère en Génie Mécanique https://www.memoireonline.com/07/08/1351/m_programmation-robots-industriels-application-robot-manipulateur-algerie6.html

[19] Stéphane Leleu , J . (2000) ." *Machine à architecture parallèle à 6 chaînes cinématiques"*. Ecole Nationale Supérieure d'Arts et Métiers | ENSAM · Département de Conception, Industrialisation, Risque et Décision (CIRD) - Lille PhD https://www.researchgate.net/figure/Machine-a-architecture-parallele-a-6-chaines-cinematiques_fig13_259562166

[20]https://www.freepik.com/premium-psd/industrial-robotic-arm-isolated-equipment-used-automotive-industry_6653856.htm

[21] Gao, J., Sun, L., Wang, L., & Gen, M. (2009). *An efficient approach for type II robotic assembly line balancing problems.* Computers & Industrial Engineering, *56*(3), 1065-1080. https://www.sciencedirect.com/science/article/abs/pii/S0360835208002222

[22] Robots, A.(2020) ."*Electromik*" Général. https://www.electromike.com/blog/post/les-bras-de-robot..html

[23]POLLY, (2021). https://roboticsandautomationnews.com/2021/04/22/robot-applications-in-auto-manufacturing/42632/

[24] Juliette , R .(2016) *.lusin nouvelle* , https://www.usinenouvelle.com/article/video-toshiba-invente-un-robot-pour-le-demantelement-de-fukushima.N1840147

[25] Saini, A., & Koyama, T. (2016). *Cleanup technologies following Fukushima*. MRS Bulletin, 41(12), 952-954.

[26] Mamoru ,M .(2014). *Intelligent deep sea robotics*. Natural Science and Technology, Okayama University https://www.okayamau.ac.jp/user/kouhou/ebulletin/feature/vol9/feature_001.html

[27]https://www.istockphoto.com/fr/vectoriel/autonome-sous-marin-rov-avec-manipulateurs-ou-bras-robotiques-v%C3%A9hicule-sous-marin-gm904146808-249351405

[28] CC BY-SA 3.0 ,(2006). https://fr.wikipedia.org/wiki/Usage_des_robots

[29] Lanfranco, A. R., Castellanos, A. E., Desai, J. P., & Meyers, W. C. (2004). *Robotic surgery: a current perspective*. *Annals of surgery*, *239*(1), 14. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1356187/

[30]AirForcephoto ,U.S. (2008)https://fr.wikipedia.org/wiki/Robot_militaire#/media/Fichier:Twuav_13_02.jpeg

[31] Zouaoui, S., Mohamed, E., Kouider, B. (2019) "*Easy Tracking of UAV Using PID Controller*", Periodica Polytechnica Transportation Engineering, 47(3), pp. 171–177. https://doi.org/10.3311/PPtr.10838

[32]Emmanuel, P. (2021) *la relation empathique de l'humain avec le robot*, Université de Bordeaux, GREThA ; https://www.implications-philosophiques.org/la-relation-empathique-homme-robot-normale-ou-pathologique/

[33] Leca, D. (2021). *Navigation autonome d'un robot agricole* (Doctoral dissertation, Université Paul Sabatier-Toulouse III).

[34] Nehme, H. (2022). *Perception de l'environnement de navigation pour un robot autonome en milieu agricole* (Doctoral dissertation, Normandie Université).

[35](2009), https://fr.wikipedia.org/wiki/Robot_agricole#/media/Fichier:Amazone_BoniRob_Feldroboter-Entwicklungsprojekt.jpg

[36] Buyukgoz, S. (2022). *Generating Proactive Communicative Robot Behaviors* (Doctoral dissertation, Sorbonne Université).

[37] https://technplay.com/lhistoire-fascinante-des-robots-domestiques-de-1985-a-aujourdhui/

[38]Boimond,J.(2019) . Université ISTIA, Angers.*La robotique peut être définie comme l'ensemble ... 4) Cours de robotique*, J.-L. Ferrier, DESS ASC. 5) Robots.
85 pages https://www.ensta-bretagne.fr/jaulin/mastersds_cours_robot_boimond.pdf

[39]BORSALI,S.(2012) . *Modélisation des Robots* .université de Tlemcen .https://ft.univ-tlemcen.dz/assets/uploads/pdf/departement/gee/Cours_modelisationdes_robots.pdf

[40] Elajrami, M., Satla, Z., & Bendine, K. (2021). *Drone Control using the Coupling of the PID Controller and Genetic Algorithm*. Communications - Scientific letters of the University of Zilina, 23(3), C75-82. doi: 10.26552/com.C.2021.3.C75-C82

[41] Shah, P., & Agashe, S. (2016). *Review of fractional PID controller*. Mechatronics, *38*, 29-41.

[42] Satla, Z., Bendine, K, & All *"P, PI, PID controller designed for UAV Quadrotors trajectory"*https://scholar.google.com/citations?view_op=view_citation&hl=en&user=gDPIoTYAAAAJ&citation_for_view=gDPIoTYAAAAJ:roLk4NBRz8UC.

[43] Satla, Z., Bendine, K, & All, May *2023 NEW DESIGN FOR AN ORNITHOPTER DRONE USING SIMPLE MECHANISM BASED ON PID CONTROLLER* Conference: 4th INTERNATIONAL ISTANBUL CURRENT SCIENTIFIC RESEARCH CONGRESS At: Istanbul, TÜRKİYE. https://www.researchgate.net/publication/371076917_NEW_DESIGN_FOR_AN_ORNITHOPTER_DRONE_USING_SIMPLE_MECHANISM_BASED_ON_PID_CONTROLLER/references

[44]Rzi Abbas (2023). Autotuning of PID Controller for a Non-planar Manipulator (https://www.mathworks.com/matlabcentral/fileexchange/114195-autotuning-of-pid-controller-for-a-non-planar-manipulator), MATLAB Central File Exchange. Retrieved June 27, 2023.

[45] Satla, Z., Bendine, K, & All, May *2023 A NEW DRONE DESIGN USING SIMPLE MECHANISMS BASED ON PD CONTROLLER* Conference: 4. INTERNATIONAL ANTALYA SCIENTIFIC RESEARCH AND INNOVATIVE STUDIES CONGRESS DATE AND PLACE May 9-10, 2023 / Antalya, Turkiye https://www.researchgate.net/publication/371577126_A_NEW_DRONE_DESIGN_USING_SIMPLE_MECHANISMS_BASED_ON_PD_CONTROLLER