الجمْـــهوريّة الجزائريـــّة الديمقْراطيّـــة الشّعْبيّـة
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

DR. YAHIA FARES UNIVERSITY OF MEDEA

Faculty of Technology

Final thesis

Presented for the Master's Degree in Electrical Engineering

Specialty: Instrumentation

Theme

Implementation of a remote control of a mobile robot via Arduino

---

Submitted by:
Ahmed Messaoud Rania
gurine Amina

Proposed and Led by:

Dr. Kherkhar abdelkrim

2019/2020

THANK YOU

Thanks to God the Almighty and Merciful to endow us with health, will, patience, during all these long years.

We also express our heartfelt thanks to our promoter

Mr.kherkhar Abdelkrim who guided this work; his enthusiasm, his many tips, his patience and his effective help for the realization of this memory.

We would like to extend our thanks to the jury members for the honour and pleasure of accepting and judging our work.

Finally, we would like to thank all those who have contributed directly or indirectly to the smooth running of our work and our studies.

DEDICACES

I thank Allah the Almighty for giving me the strength and courage to do this modest work.I have the honour to dedicate this modest work to:

To my dearest parents who have greatly encouraged and supported me throughout their lives to see that day and for their valuable advice and recommendations

to my dear brothers

to all my family, big and small without exception.

To all my friends

Rania

DEDICACES

I thank Allah the Almighty for giving me the strength and courage to do this modest work.I have the honour to dedicate this modest work to:

To my dearest parents who have greatly encouraged and supported me throughout their lives to see that day and for their valuable advice and recommendations and my dearest sister.

To all my family,

Big and small without exception.

To all my friends.

Amina

ملخص

يتعلق هذا المشروع بتنفيذ جهاز تحكم عن بعد لروبوت متنقل باستخدام لوحة إلكترونية أردوينو.

الغرض من هذه الدراسة هو التحكم في الروبوت عن طريق البلوتوث عبر تطبيق أندرويد

كما استخدمنا برمجة ماتلاب لتحقيق هدف المشروع.

في النهاية نحصل على إحدى الطرق العامة لتوجيه الروبوتات.

الكلمات الرئيسية :الروبوت المحمول ، اردوينو ، بلوتوث

Résumé

Ce projet est lié à la mise en œuvre d'une télécommande robotisée mobile utilisant une carte électronique Arduino.

L'objectif de cette étude est de contrôler le robot via Bluetooth via l'application Android.

Comme on a utilisé une programmation Matlab pour atteinte notre but de projet Enfin, nous obtenons un des moyens publics d'acheminer les robots

Mots clés : robotmobile, arduino, Bluetooth.

Abstract

This project is related to the implementation of a mobile robot remote control using a electronic board.

The purpose of this study is to control the robot via Bluetooth via the Android app

As we used Matlab programming to achieve our project goal finally, we get one of the public ways to route robots.

Keywords: mobile robot, arduino, Bluetooth.

List of continues

General Introduction……..…..01

CHAPTER I: GENERAL ON MOBILE ROBOTS

I.1.Introduction…….……..03
I.2Robotics……..……...…03
I.2.1. Definition of a mobile robot……....……03
I.2.2. Synoptic diagram of robot……...…….…..……04
I.2.3. Historical overview……..…..…...04
I.3.type of robots…….……...05
I.3.1. Manipulator robots…….……..…05
I.3.2. Mobile robots……..…..….06
I.4. Mode of operation……...……06
I.5. mobile bases…….…….07
I.5.1. Holonomy…….…....….07
I.5.2. differential platforms……...…07
I.5.3. Omnidirectional platforms……..……08
I.5.4. Non-holonomic platforms…….…….....08
I.5.5. Legged platforms…….…….……...09
I.6.Wheeled robots……..…09
I.6.1. Unicycle mobile robot…….…….09
I.6.2. Tricycle robot……...…….…..….10
I.6.3. Robot Car …….……10
I.6.4. Omnidirectional robot……...…..….11
I.7.field of application……..……...….12
I.8.Sensors used in mobile robotics……..……13
I.9.Advantages and disadvantages of robots……..…….……14
I.10.Conclusion……...…….14

CHAPTER 2: ARDUINO CAMMANDER SYSTEM

II.1. Introduction…….……...…15
II.2. the Arduino board…….……16

II.2.1. Brief history of the Arduino……..……..……16

II.2.2. Definition……..…16

II.2.3. the Applications……..……..……..17

II.2.4.Types of Arduino Boards…….……17

II.2.5. the range of the Arduino board……..……...……17

II.2.6. Arduino Uno…….....……19

II.2.7. Presentation……...……19

II.2.8. Arduino Uno Board Structure…….……..…20

II.2.9. Summary of Characteristics……..……...…22

II.2.10.Good networks to choose arduino……..…….....……23

II.2.11. Arduino Board Management software……..…24

II.2.12. the advantages of the arduino…….……...……...25

II.3. Conclusion…….....……...…26

CHAPTER III: SLIDING MODE CONTROL FOR ROBOT

III.1. Introduction…….…….....…27

III.2. Model of Robotic Joints……..…27

III.2.1.Model Description……...…27

III.2.2. Model Description Example……28

III.3.Sliding Mode Control Based on Input-Output Stability……..29

III.3.1.System Description…….……29

III.3.2. Design of Controller…….……30

III.3.3 Simulation Example……...……32

III.4.Sliding Mode Control Based on Computed Torque Method…….34

III.4.1Design of Controller……34

III.4.2 Simulation Example……...……35

III.5. Adaptive Sliding Mode Control for Manipulator……..….36

III.5.1.Adaptive Sliding Mode Controller…….....…36

III.5.2 Simulation Example…….....…….…38

III.6.Conclusion:……...39

CHAPTER IV: CONTROL OF AERIAL MANIPULATOR

IV.1.Introduction:……..……40

IV.2.Dynamics of combined system:……40

IV.3.Control problem:……...……41

IV.4.Control Approach……41

IV.4.1.Cartesian Impedance Control:……41

IV.4.2Adaptive Sliding Mode Control…….…….44

IV.4.3. Adaptive Cartesian Impedance Control……..…..…...44

IV.6.Conclusion and Future Work ……..……...…...47

LISTS OF FIGURES

CHAPTER I GENERAL INFORMATION ON MOBILE ROBOTS

Figure I.1 The first mobile robot Shakey (SRI, 1966-1972)**……..……03**

Figure I.2 Synoptic diagram of a robot.**…….……04**

Figure I.3 A left: "Beast" robot from John Hopkins University in the 1960s. Right: The Stanford Shakey robot in 1969 was a demonstration platform forartificial intelligence research**…….……05**

Figure I.4 manipulator robot (arm). Bosso Midi 4-axis loading and unloading robot**.....05**

Figure I.5 Humanoid robot.**……...…06**

Figure I.6 Example of an omnidirectional platform with Swedish wheels.**…..…….……08**

FigureI.7 walking mobile robot **……..…….…….09**

Figure I.8 unicycle robot**…….……..…..……09**

Figure I.9 Tricycle robot**……....….……10**

Figure I.10 robot car.**…….…….....11**

Figure I.11 Omnidirectional robot**……..……11**

Figure I.12 artificial robot.**…….…13**

CHAPTER II:ARDUINO CAMMANDER SYSTEM

Figure II.1 different Arduino boards.**……..…18**

Figure II.2 The compositions of the arduino board UNO **……..……19**

Figure II.3 Structure of the Arduino Uno board.**……...…20**

Figure II.4 microcontroller**……..20**

Figure II.5 Structure of the Arduino Uno board**……22**

Figure II.6 Arduino software interface.**……24**

Figure II.7 Button bar Detail.**……25**

CHAPTER III: SLIDING MODE CONTROL FOR ROBOT

Figure III.1 A classical dual-joint rigid manipulator**…….28**

Figure III.2 Tracking of position and velocity for the first joint**……33**

Figure III.3 Tracking of position and velocity for the second joint**……33**

Figure III.4 Control input**……34**

Figure III.5 The position tracking of a dual-joint manipulator**……...…36**

Figure III.6 The control input of a dual-joint manipulator**…….36**

CHAPTER IV: CONTROL OF AERIAL MANIPULATOR

Figure IV.1 UAV/Arm system with corresponding reference frame **……..…40**

Figure IV.2 Time histories simulation for Rigid Case.**……...…42**

Figure IV.3 Time histories simulation for Compliant Case**……43**

Figure IV.4 Time histories simulation for Compliant Rigid Case.**……43**

Figure IV.5 Time histories simulation for adaptive sliding mode controller.**……45**

Figure IV. 6 Position history of the quadrotor.**……...…46**

Figure IV.7 Attitude history of the quadrotor**……..…46**

Figure IV.8 Torque History for both links**……..……47**

TABLE LIST

Table I.1: Advantages and disadvantages of different types of wheeled robots**…….12**

Table III.1 Parameters of a dual-joint manipulator**…….29**

LIST OF ACRONYMS

AC: Alternating Current.

CCD: Charge Coupled Device.

DC: Direct Current.

EEPROM: Electrically-Erasable Programmable red –only memory

GND : Grounds (la masse).

GPS: Global Positioning System.

ICSP : In-Circuit Serial Programming ou (Programmation Série Dans le circuit).

IDE : Integrated Development Environment.

LED: light-emitting diode.

LX: Reception data.

PC: personal computer.

PWM : Pulse Width modulation (Modulation de la largeur des pics).

RAM: Random Access Memory (mémoire vive).

ROM: Read-Only Memory.

S.M.A : système mécanique articulé

SRAM : StaticRandom Access Memory (mémoire vive statique).

TW : TwoWire Interface ou (Interface "2 fils").

TX: emission data.

USB: Universal Serial Bus.

VIN: Voltage Input

GENERAL INTRODUCTION

General Introduction

Since 1970, the scientific and industrial world has been increasingly involved in the study and construction of mobile machinery. They are able to make movements without mechanical connection to a fixed point in the space in which they operate, hence the name of mobile robots.

Over the years, robotics has been a dominant contributor to the development of human society where robots are built to improve our daily lifestyles they areable to perform dangerous tasks in place of human being

The robots of the first generation were simply automata capable of performing repetitive tasks in well-known environments. The second generation saw the appearance of robots equipped with sensors allowing them to perceive changes in the environment and act in a simple way. Since then, robots have developed and become widespread. They are endowed with the means of perception and decision allowing them to understand and evolve in different types of environment. Some are even used for exploration of distant planets (Moon, Mars). The latest generation of robots should have a high degree of autonomy, it is the subject of intensive research. It has become increasingly clear that for many years to come cooperation between man and machine is inevitable. [1]

Robotics has two main areas of interest: handling robotics, which are fixed robots, and mobile robotics. One of the major problems of mobile robotics is movement planning. Around this problem of motion planning many studies have been carried out with the aim of developing general methods to guide robots. [2]

In this brief, we will present the implementation of a remote control of a mobile robot via Arduino, the principle of this study is to have a robot controlled by a Bluetooth via an Android application (App_robot) and programming will be done in Arduino language that looks like C++, plus Matlab programming to achieve our project goal.

The first chapter presents an introduction to mobile robotics its definitions , these types and applications.

The second chapter is devoted to the embedded systems used in our project, such as the arduino board.

In the third chapter, we assigned the main study to carry out our project, concern Sliding Mode Control for Robot.

In chapter 4: Control of Aerial Manipulator, we have to use Inspection and maintenance of power lines,Rescue operations and Construction in inaccessible sites.

Conclusion that summarizes the essence of our work, as well as the perspectives that can be envisaged to make improvements in the future.

CHAPTER I

GENERAL ON MOBILE ROBOTS

I.1. Introduction:

Robotics is a very good example of a multidisciplinary field that involves many topics such as mechanics, electronics, automation, computer science or artificial intelligence. It is divided into two types: industrial robots and mobile robots.

This chapter will provide a number of definitions and a brief history of robotics. The different types of robots and their components will be introduced.

We will conclude with a presentation of the advantages and disadvantages.

I.2 Robotics:

I.2.1. Definition of a mobile robot:

A robot is a mechatronic device, it combines mechanics, electronics and computer science. It automatically performs either tasks that are generally dangerous, painful, repetitive or impossible for humans, or tasks that are simpler but do them better than what a human being would do.**[3]**

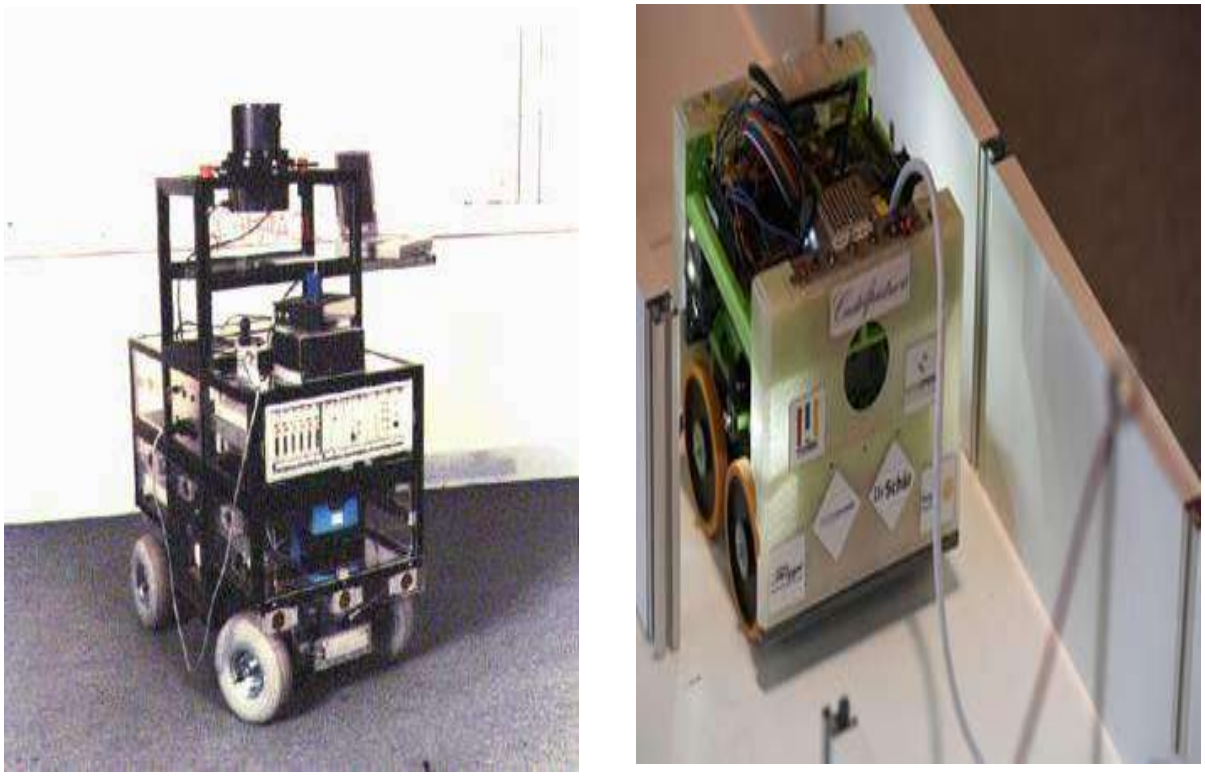


Figure I.1 The first mobile robot Shakey (SRI, 1966-1972)

I.2.2. Synoptic diagram of a robot[3]:

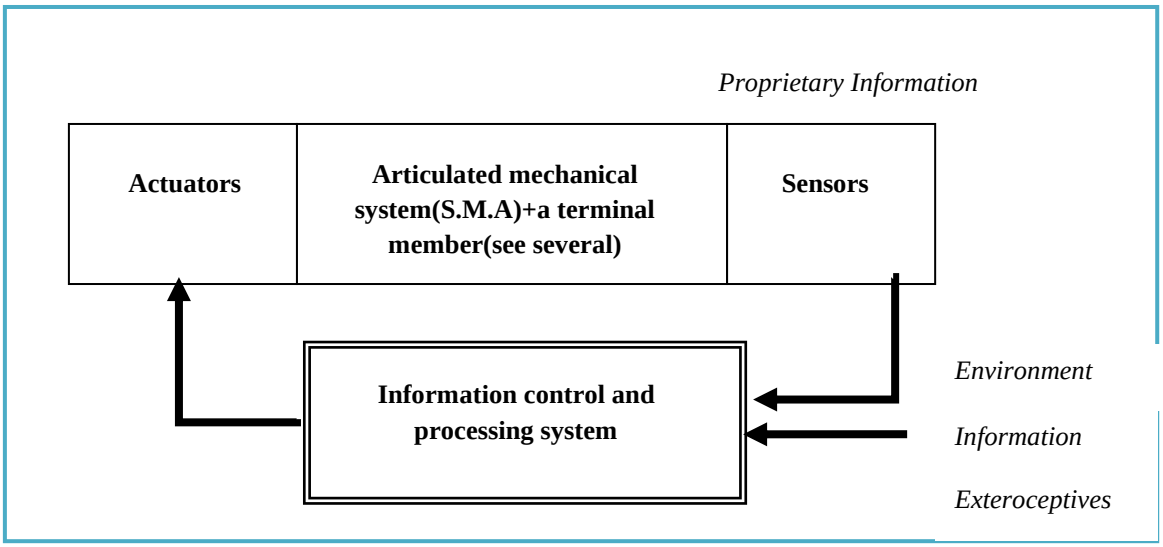


Figure I.2 Synoptic diagram of a robot.

✓ **The articulated mechanical system (MLS)** is a mechanism with a structure more or less similar to that of the human arm. It makes it possible to replace, or prolong, its action. Its role is to bring the final organ into a given situation (position and orientation). Its architecture is a chain of bodies, usually rigid, assembled by bonds called joints.

✓ **The motorization** is carried out by electric, pneumatic or hydraulic actuators that transmit their movements to the joints.

✓ **Sensors are sensory organs**: proprioceptive when measuring the internal state of the robot (joint positions and velocities) and external when collecting information on the environment (presence detection, contact, distance measurement, artificial vision).

✓ **The terminal component** means a device intended to manipulate objects (clamping devices, magnetic devices, etc.), or to transform them (tools, welding torch, paint gun, etc.). It is an interface that allows the robot to interact with its environment.

✓ **The control part** synthesizes the instructions of the controls controlling the actuators, starting from the function of perception and orders of the user.

I.2.3. Historical overview

It was not until the very beginning of the 20th century that robots appeared, following the work of engineers who wanted to test hypotheses made by biologists and psychologists. The electric dog designed by Hammond and Meissner in 1915 was attracted by a light, according to the animal phototropism highlighted by Loeb in 1918. The machines of Russell (1913) and

Stephens (1929), the cybernetic turtles of Grey Walter (1950), the electronic fox of Ducrocq (1953) or the homeostat of Ashby (1952) were endowed with learning abilities directly derived from the work of the Thorndike psychologists (1911), Hull (1943) and physiologist Pavlov (1903) on Man and Animal.

These realizations are robots, because they no longer behave as simple automata whose motor organs - their mechanisms - obey a predetermined program. Unlike automatons, they have sensory organs - the sensors - that collect information from the environment that will, in turn, influence the activity of their motor organs - the actuators.[4]

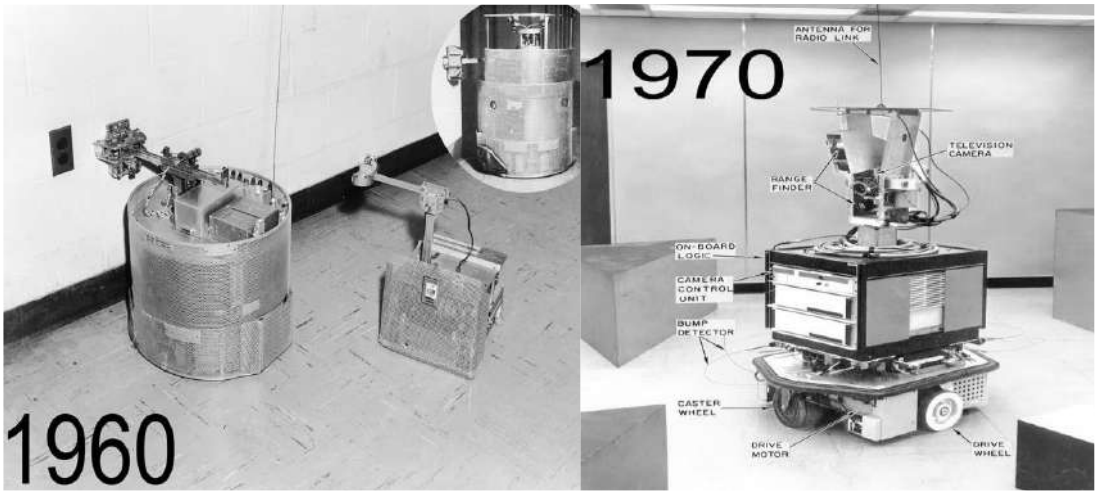


Figure I.3A left: "Beast" robot from John Hopkins University in the 1960s.Right: The Stanford Shakey robot in 1969 was a demonstration platform for artificial intelligence research.

I.3.type of robots:

I.3.1. Manipulator robots: Manipulator robots are robots integrated into a clean space for the transfer of semi-finished products between production lines. The robot is usually suspended due to space constraints and siting constraints. [5]



Figure I.4 manipulator robot (arm). Bosso Midi 4-axis loading and unloading robot

I.3.2. Mobile robots: robots capable of moving in an environment. They may or may not be equipped with manipulators depending on their use.

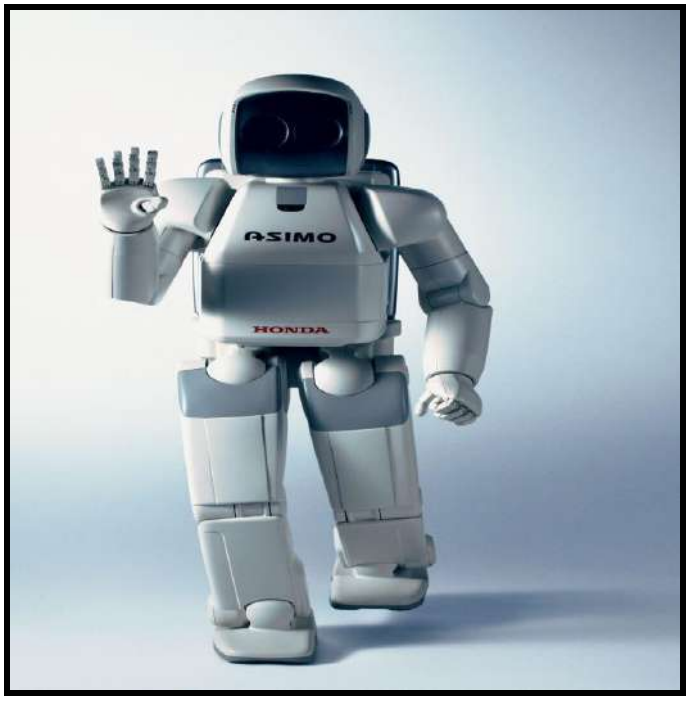


Figure I.5 Humanoid robot.

When we talk about mobile robotics, we often mean mobile wheeled robots. They are indeed the most studied systems, because they are simpler to realize than other types of mobile robots, which makes it possible to come to the study of their navigation faster. This type of robot is very often used for the study of autonomous systems. Next comes mobile robotics with legs, including humanoid robotics, but also robots with a higher number of legs that offer good properties for locomotion in difficult environments (forest and agricultural environments). Finally there are also many other types of mobile robots (marine robots, submarines, flying drones, micro and nano robots), generally the study of this type of robots is done in specific themes with individuals to the intended application. [6]

I.4. Mode of operation:

There are two main modes of operation for a mobile robot: teleoperated and autonomous. In the teleoperated mode, a person drives the robot remotely. It gives its orders via a command interface (joystick, keyboard/mouse, etc.), and these are sent to the robot via a communication link (internet, satellite, etc.). The robot must therefore obey the orders of the operator who perceives the environment around the robot. Conversely, in autonomous mode the robot must make its own decisions. This means that he must be able to perceive his

environment correctly, but also know how to react accordingly, depending on the level of autonomy. [6]

I.5. mobile bases:

We quickly present the different types of mobile bases used in robotics, focusing on mobile terrestrial platforms for the indoor environment.

I.5.1. Holonomy:

In robotics, a platform is called a holonome when the number of controllable degrees of freedom equals the total number of degrees of freedom. For a robot moving on a plane, there are 3 degrees of freedom (two translations and a rotation). From a given position, a holonomic platform must therefore be able to move forward, to the side and turn on itself. This capability makes it possible to control the robot very simply because all imaginable movements are feasible, which simplifies the problem of trajectory planning. Many simple platforms are not holonomic. This is for example the case of cars, which requires maneuvering to achieve certain trajectories. For example, it is necessary to make a niche to achieve lateral displacement. These constraints must therefore be taken into account when planning trajectories. We will, however, see some mechanisms for obtaining or approaching holonomic platforms. [7]

I.5.2. differential platforms:

One of the most commonly used configurations for indoor mobile robots is the differential configuration with two independently controlled wheels. One or more crazy wheels are added to the front or rear of the robot to ensure its stability. This platform is very easy to control, since you just need to specify the speeds of both wheels, and also allows the robot to turn on the spot. This possibility allows the robot to be treated in some cases as a holonomic robot, which will simplify the planning of movement and the control of the robot.

The estimation of the displacement by odometry is also very simple from the measurement of the rotational speeds of the two wheels ω1 and ω2. The travel and rotation speeds ω are given by:

$$v = \frac{\omega_1 r + \omega_2 r}{2} \qquad 1$$

$$\omega = \frac{\omega_1 r - \omega_2 r}{2l} \qquad 2$$

This type of platform can also be used with crawlers which provides an interesting small obstacle clearance capability . These platforms can therefore be used outdoors or in rubble. However, the use of tracks leads to a very noisy odometry because of the poorly defined contact between the tracks and the ground, which slide a lot, especially during rotations. Therefore, odometry-based direction estimation on these types of platforms is generally quickly unusable. [7]

I.5.3. Omnidirectional platforms:

Omnidirectional platforms make it possible to decouple more clearly the control of the rotation and translation of a robot and are therefore almost holonomic. There are different types of omnidirectional platforms. The first uses three or four wheels that rotate at the same speed to provide a translation and a mechanism that allows the wheels to be steered simultaneously in the direction of the desired motion (Figure I.6). The body of the robot itself does not rotate but only translations. This system allows a very simple and relatively fast control because the change of direction only concerns the wheels and can therefore be done very quickly. On the other hand, these platforms are relatively limited in crossing capacity and require very flat ground. A second category of platforms uses so-called "Swedish" wheels, which do not offer resistance to lateral movement. The platform consists of three

wheels with fixed axes. Motion in all directions and in rotation is achieved by individually varying the wheel speeds. The platformrotates on the spot when the three

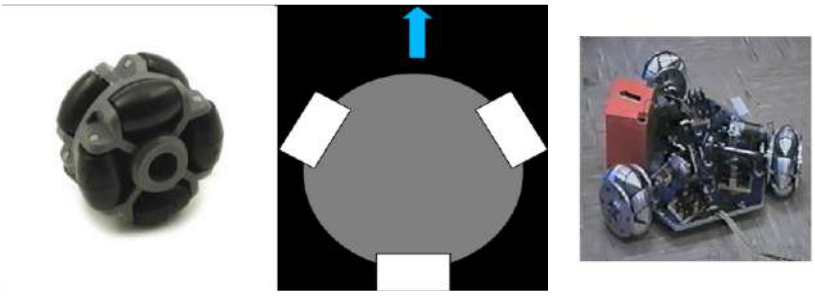


FIGURE I.6 Example of an omnidirectional platform with Swedish wheels.

Wheels turn in the same direction, at the same speed. When one wheel is fixed, and the other two rotate in the opposite direction, the platform moves in the direction of the fixed wheel. There are different speed combinations that allow for some movement. [7]

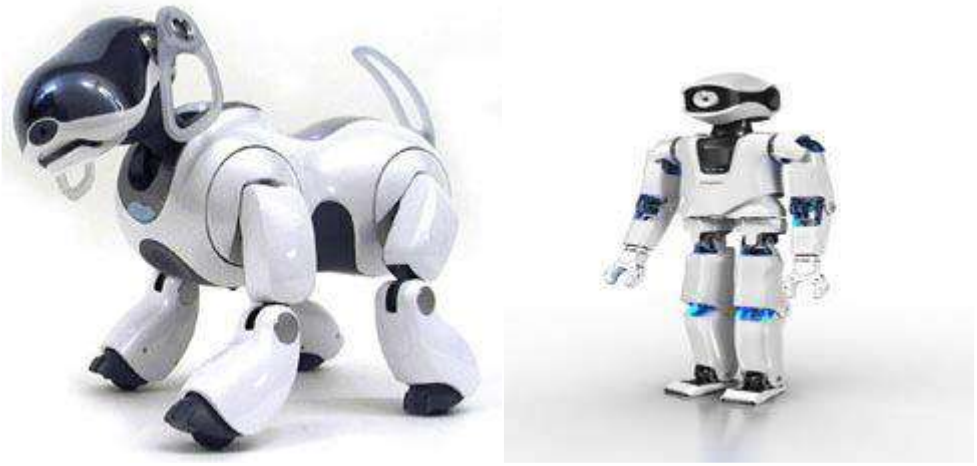
I.5.4. Non-holonomic platforms:

Non-holonomic platforms, such as cars, are also used in mobile robotics. This is particularly the case in the area of intelligent vehicles. However, these platforms are more difficult to control because they cannot rotate on site and must maneuver, which can be difficult in cluttered environments. Ordering these platforms to perform a particular move is a separate issue that we will not address in this course. However, these constraints can be taken into account in planning in a relatively simple manner. [7]

I.5.5. Legged platforms:

Two-, four- or six-legged platforms may also be used. They have the theoretical advantage of being able to move on fairly complex terrain, even if in practice most of these platforms only work on flat soils. [7]

There are walking robots (figure with two legs (humanoids), four legs (horse type), and six legs (spider type).



figureI.7 walking mobile robot

I.6.Wheeled robots:

I.6.1. Unicycle mobile robot: this model is driven by two independent wheels, it has possibly crazy wheels to ensure its stability. Its center of rotation is located on the axis connecting the two driving wheels. It is a non-holonomic robot, indeed it is impossible to move it in a direction perpendicular to the locomotor wheels. Its control can be very simple, it is indeed quite easy to move it from one point to another by a series of simple rotations and straight lines. [8]



Figure I.8 unicycle robot.

I.6.2. Tricycle robot:

A three-cycle robot is made up of two fixed wheels placed on the same axis and a wheel centered which can be rotated on the longitudinal axis. The movement of the robot is given by the speed of the two fixed wheels and by the orientation of the swivel wheel. Its centre of rotation is located at the intersection of the axis containing the fixed wheels and the axis of the swivel wheel. It's a non-holonomic robot. Indeed, it is impossible to move it in a direction perpendicular to the fixed wheels. His order is more complicated. Simple rotations are generally not possible due to a limited turning radius of the steerable wheel.[8]



Figure I.9 Tricycle robot

I.6.3. Robot Car:

A car-like robot is similar to a tricycle, consisting of two fixed wheels placed on the same axis and two swivel centered wheels placed on the same axis. The car-type robot is however more stable since it has an additional support point. All the other properties of the car robot are identical to the tricycle robot, the second being able to be brought back to the first by replacing the two front wheels by a single one placed in the center of the axis, in order to leave the center of rotation unchanged. [8]



Figure I.10 robot car.

I.6.4. Omnidirectional robot:

An omnidirectional robot is a robot that can move freely in all directions. It is usually made up of three pivoting off-center wheels placed in an equilateral triangle. The huge advantage of the omnidirectional robot is that it is holonomic since it can move in all directions. But this is at the expense of a much greater mechanical complexity. [8]



Figure I.11 Omnidirectional robot

Table I.1: Advantages and disadvantages of different types of wheeled robots.

type of robot	Advantage	disadvantage
• Unicycle	• Stable Rotation on oneself Low mechanical complexity	• Non-holonome
• Tricycle	• Moderate mechanical complexity	• Non-holonome • Not stable • No rotation on oneself
• Car	• Stable	• Non-holonome • Significant mechanical complexity
• Omnidirectional	• Holonoma Stable • Rotation on oneself	• Significant mechanical complexity

I.7.field of application: [9]

➢ **The industry.** The robotization of the industry began in the 1960s, in the automotive sector, then spread until we know today in the form of automated arms or a production line (bagging of fertilizer in bags).

➢ **The military domain.**Robots are increasingly used in the military domain. They are remotely controlled weapons In fact, miniaturization makes it possible today to create discrete robots but equipped with many sensors, which is ideal for espionage or illuminance missions.
.

➢ **Health.** Robots seem to have a future in the hospital.Robodoc, for example, helps with certain surgical operations. The nurse robot is still being planned. The HAL cyber skeleton helps people to move around.Thus; the patient robot allows future dental surgeons to learn how to treat without doing any damage.



Figure I.12artificial robot.

I.8.Sensors used in mobile robotics:

In mobile robotics, sensors are traditionally classified into two categories according to whether they measure the state of the robot itself or the state of its environments In the first case, has the image of perception in living beings, We are talking about proprioception and therefore proprioceptive sensors. For example, this category includes wheel position or speed sensors and battery charge sensors. The sensors that provide information about the state of the environment, and therefore about what is external to the robot itself, are called external sensors. These are sensors giving the distance of the robot to the environment, the temperature, signaling the placing of the robot in contact with the environment, etc. [7]

a. Telemetry sensors

Telemetry is any technique of measuring distance by acoustic, optical or radio processes. The device for measuring distances is called a rangefinder. There are different technologies for making a rangefinder. In the following we present the most widely used rangefinders in mobile robotics systems, giving an idea of their range of measurement and application. [4]

b. GPS

There are very few systems giving the absolute position of a point in a fixed fix give.The Global Positioning System (GPS), originally developed for US military applications, is currently available to the general public. However, it can be considered that its use in this context is not guaranteed. The implementation of the European civil project Galileo [ESA 04] should offer an alternative to GPS. The GPS works with a set of satellites, which perform time-synchronized broadcasts.By cross-checking the arrival times of the signals and the

position of the transmitting satellites, the receivers can calculate their position. The principle of position calculation is based on triangulation, using four signals received simultan.

I.9.Advantages and disadvantages of robots:

A robotic system consists not only of robots but also of other devices and systems that are used with the robot to perform the necessary task.

❖ **Les avantages des robots sont[11] :**

- Robotics and automation can in many situations increase the productivity, safety, efficiency, quality and consistency of products.
- Robots can work in a hazardous environment, without the need for life support, or safety concerns.
- Robots do not need lighting, air conditioning, ventilation or noise protection.
- Robots work continuously, without feeling tired or bored, and do not require medical or holiday insurance.
- Robots are repeatable precision at all times.
- Robots can be much more precise than humans. Linear precision of a robot typically is 10 to 20 microns.
- They are constantlyimproving.

❖ **The disadvantage of robots**

- They lack the ability to respond to unforeseen events, unless the situations are understood and the responses are included in the system. Safety measures are necessary to ensure that they do not injure operators or damage machines. Otherdisadvantagesinclude: [11]
- Inadequate response.
- Lack of decision making
- Consumption of energy.
- They can cause damage to other devices, and injury to man.
- Robots are expensive because of the initial cost of equipment, installation, the need for peripherals, the need for training and the need for programming (robots need humans).
- Fear of robots stealing our jobs
- Robot capabilities are limited.

I.10.Conclusion:

The main objective of our study is the remote control of a mobile robot For this we have presented in this chapter important generalities about mobile robots and its types we have also seen its field of application and sensors used in the field of robotics In the next chapter we will present the electronic cards that are used to manage and control the mobile robot.

CHAPTER II:

ARDUINO CAMMANDER SYSTEM

II.1. Introduction:

Since electronics has existed; its growth has been rapid and continues to this day. So much so that making electronics has become accessible to anyone who wants to. But, the lack of simple courses on the net or in bookseller prevents the satisfaction of future amateur or professional electronics and sometimes prevents some geniuses to reveal themselves what we're going to learn is a mix of electronics and programming. We are going to talk about embedded electronics, which is a sub-domain of electronics and which has the ability to combine the power of programming with the power of electronics. We will first look at what electronics and programming are. Then we will follow on the grip of the Arduino system

II.2. the Arduino board:

II.2.1. Brief history of the Arduino:

Arduino was established in 2005 in the Institute of Design Interaction of Ivrea in Italy, as an open source platform of. The founders of the project, Massimo Banzi and David Curatelles.Hernando Barra Gan, for his graduation thesis had undertaken the development of an electronic map called Waring, accompanied by a free and open programming environment. For this work, Hernando Barragan used the sources of the Processing project. Based on an easy-to-access programming language and adapted to the development of designer projects, the Wiring board inspired the Arduino project (2005). As for Wiring, the objective was to arrive at a device that would be simple to use, with low costs, the codes [11]

II.2.2. Definition:

Arduino is a platform based on easy-to-use hardware and software.It consists of a printable circuit board, which can be programmed (called microcontroller) and a ready-to-use software called Arduino IDE (Integrated Development Environment), which is used to write and load the computer code on the board.Arduino provides a standard form factor that breaks the functions of the microcontroller into a more accessible package. [12].

Over the years, Arduino has been the mastermind of thousands of projects, from everyday objects to complex scientific instruments.A global community of decision makers - students, amateurs, artists, programmers and professionals - gathered around this open source platform, their contributions [4]

II.2.3. the Applications:

The Arduino system allows us to achieve a lot of things, which have an application in all areas, we can give some examples:

• Control household appliances

• Play a game of lights

• Communicate with the computer

• Remote control of a mobile device (model building) etc.

• Build your own robot.

With Arduino, we will make electronic systems such as an electronic candle, a simplified calculator, a synthesizer, etc.All of these systems will be designed with an Arduino board and a wide range of electronic components. [13]

II.2.4.Types of Arduino Boards:

There are three types of cards:

a. the “official” that are manufactured in Italy by the official manufacturer (Smart Projects).
b. “Compatible” that is not manufactured by (Smart Projects), but is fully compatible with the official Arduino.

c. “Others” manufactured by various companies and marketed under a different name (Freeduino, Seeduino, Femtoduino ...). [5]

II.2.5. the range of the Arduino board:

currently, there are more than 20 versions of Arduino module, and we quote some to clarify the evaluation of this scientific and academic product:
• The Arduino NG, with a USB interface to program and use anATmega8.

• The Arduino end, with a USB interface to program and use an ATmega8 microcontroller.

• The Arduino Mini, a miniature version of the Arduino using a microcontroller ATmega168.
• The Arduino Nano, a small program board using this version USB port using an ATmega168 microcontroller (ATmega328 for a newer version).

•The LilyPad Arduino, a minimalist design for the wearable application using an ATmega168 microcontroller.

• The Arduino plus NG, with a USB interface to program and use anATmega168.

• The Arduino Bluetooth, with a Bluetooth interface to program using an ATmega168 microcontroller.

• The Arduino Decimal, with a USB interface and uses a microcontroller ATmega168.

• The Arduino Duemilanove ("2009"), using a microcontroller ATmega168 (ATmega328 for a newer version) and powered via USB/DC power.

• The Arduino Mega, using an ATmega1280 microcontroller for memory I/O.

• Arduino UNO, ATmega328 microcontroller uses.

• The Arduino Mega2560 uses an ATmega2560 microcontroller, and has all the memory at 256 KBS. It also incorporates the new ATmega8U2 (ATmega16U2 in revision 3 USB chipset).

• The Arduino Leonardo, with an ATmega3tâ4 piece that eliminates the need for USB connection and can be used as a keyboard.

• The Arduino Esplora: looks like a visual game controller, with a handle and integrated dampers for noise, light, temperature, and acceleration.

Among these types, we chose an Arduino UNO board (Basic board). The main interest of this map is to facilitate the implementation of such a command which will be subsequently cut [14].

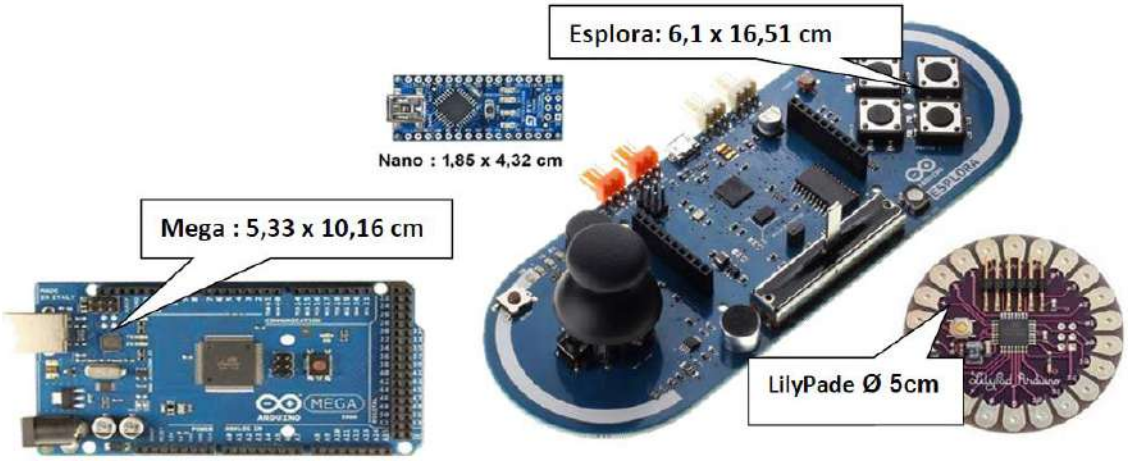


Figure II.1different Arduino boards.

II.2.6. Arduino Uno:

The Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power outlet, an ICSP header and a reset button.It contains everything needed to support the microcontroller; just connect it to a computer with a USB cable or power with an adapter or AC-DC battery to start. [1]

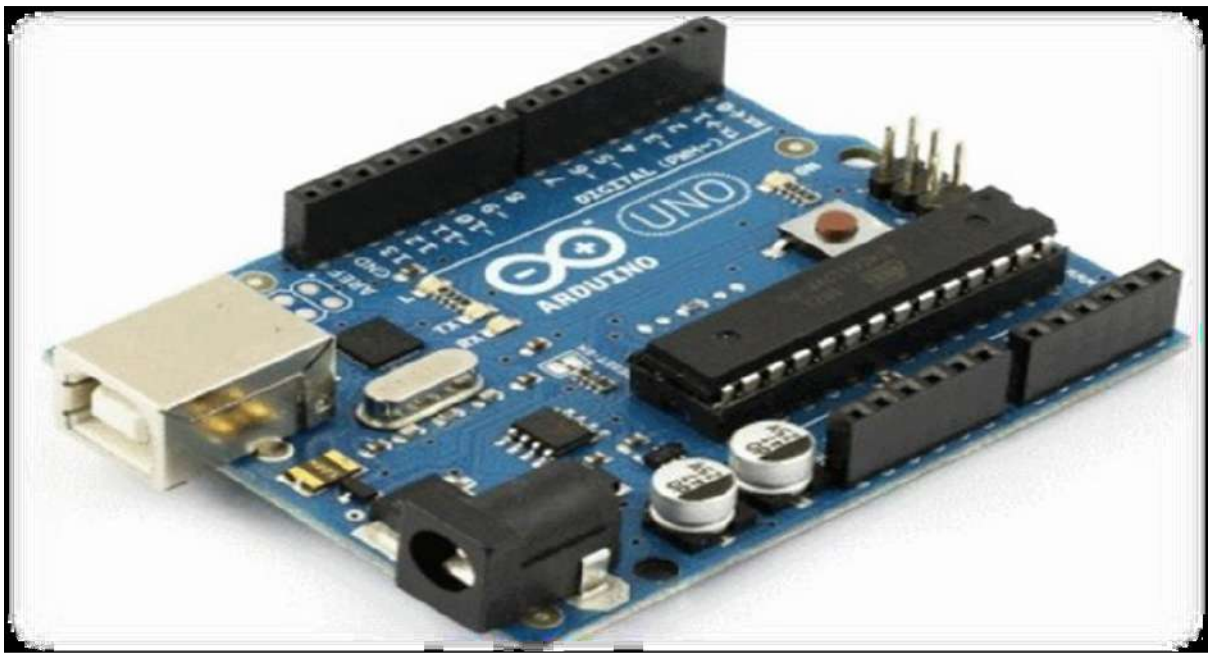


Figure II.2 The compositions of the arduino board UNO [1].

II.2.7. Presentation:

it is an electronic board based on an Atmega microcontroller from the manufacturer Atmel, whose price is relatively low for the possible scope of applications. That's what themap we're going to use in figure II.2 looks like

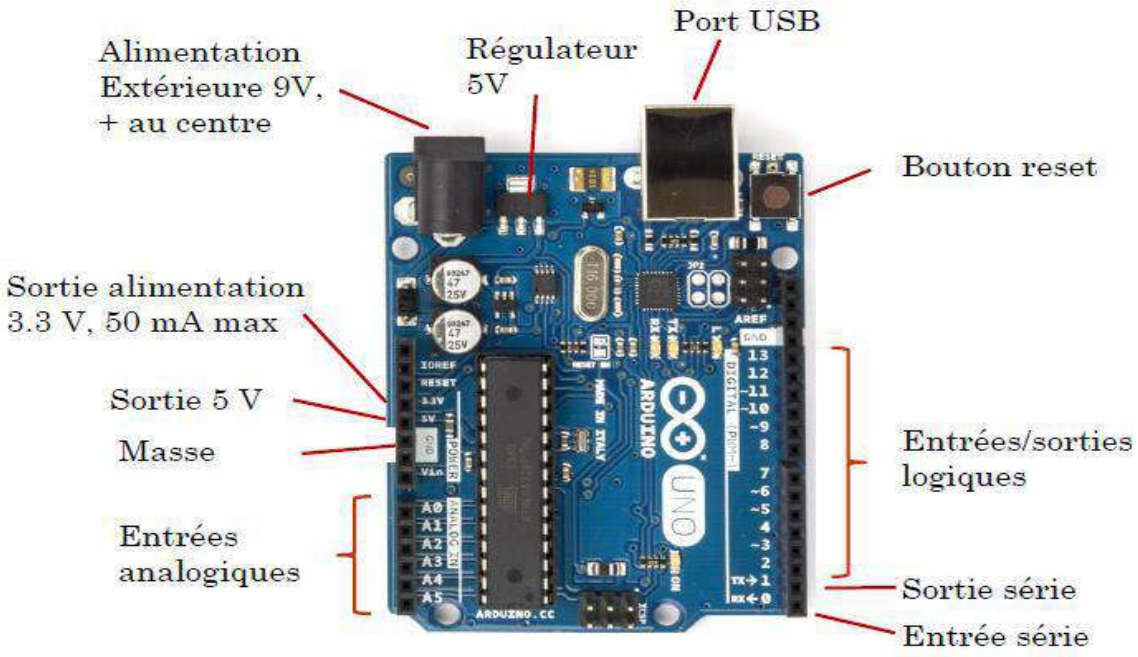


Figure II.3 Structure of the Arduino Uno board.

II.2.8. Arduino Uno Board Structure:

inwhat follows, we will detail the structure of this system and their interests.

• The microcontroller

themicrocontroller matches the brain of our card, it's the one that will process from the sensors in the input and give the answer to the output through the program you created and that will store it in its memory. [4]



Figure II.4 microcontroller.

•Digital pins

you use the digital pins, which are located along the top of the board in figure II.2.6, to send and receive digital signals. Numeric means that the signal has only one of the two up or down states.In termelectric terms, this means a voltage of 0 volts or 5 volts, without inter mediate tensionization. [15]

• Analog pins

the analog pins are located at the bottom of the board on Figure II.2.7. They are intended to receive analog values. A value is called analog when it can take a value among all those of a predefined range. In this case, the range is from 0 to 5 volts, and the value can be anywhere between the two terminals: 0.1 volt, 1.2 volt, 4.9 volt, etc. [15]

- **Analog outputs**

the most attentive of you will have noticed that there does not seem to be any analog output pins. In fact, there are, but they are simulated by the numeracy pins marked as PWM with the symbol "~". PWM is an abbreviation for Pulse with Modulation (Pulse Width Modulation), a technique you can use to simulate an analog output with a digital output. The ~symbol appears next to pins 3, 5, 6, 9, 10 and 11, so you have 6 pins that can transmit PWM modulation. [15]

- **The power pins**
- you first need to know the specifications of the Arduino board you own. First of all, they all use direct current.
- If we look at the different possible pins, we will usually find a pin labeled 5V, a pin labeled 3,3V and a pin labeled VIN or RAW.
- VIN or RAW is the input of the voltage regulator on the board. To power the Arduino via this pin, you need a power supply with a voltage between 7 and 12V.

•5V does not have the same role as the microcontroller on the Arduino with 5V power supply

or 3.3V. In the first case, it is directly the power supply voltage of the microcontroller. In the second case, the microcontroller is powered via a 3.3V regulator whose input is connected to the 5V.

• Similarly, the 3.3V will be the microcontroller supply voltage for those powered with 3.3V

and a service voltage to connect external devices requiring 3.3V if the microcontroller requires 5V.

Can feed the Arduino board by:

- ✓ USB Power Supply
- ✓ Power supply via Jack connector
- ✓ Two-pin power supply for the Arduino: VIN and GND

• Visualization

we have three LEDs that we find on the map given to two things:
The first (L): it is connected to a pin of the microcontroller and will be used to test the hardware.
Note: When the card is connected to the PC, it flashes for a few seconds.
Both LEDs: are used to visualize the activity on the serial track (one for the emission (TX) and the other for the reception (LX)).

As the program is downloaded into the microcontroller through this channel, they can be seen flashing when loaded. [4]

• The Reset button

The Uno board finally has a button next to the USB connector. It is the deréinitialisation button (RESET). It resets the Arduino or stops it completely when it is held pressed for a while. You get the same result by tying the bottom pin marked Reset to the ground, one of the two GND pins more to the right in the same row. The board contains many other components, all of which provide important functions, but the ones that have just been described are the ones that are important to you right now. [14]

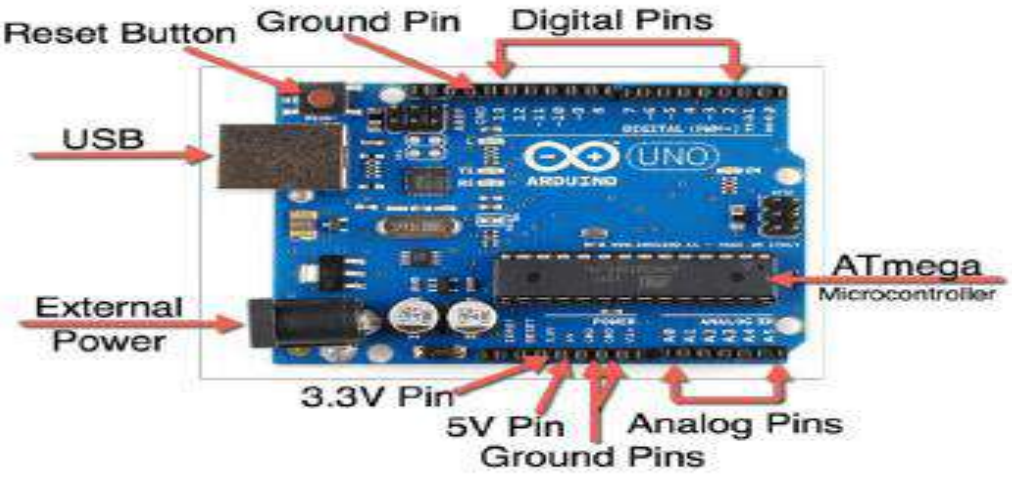


Figure II.5 Structure of the Arduino Uno board.

II.2.9. Summary of Characteristics:

- Microcontroller: ATmega328
- Operating voltage: 5V
- Supply voltage (recommended): 7-12V
- Supply voltage (limits): 6-20V
- Digital I/O pins: 14 (6 of which have PWM output)
- Analog input pins: 6 (can be used in digital I/O pins)
- Max current available per pin I/O (5V): 40 mA (CAUTION: 200mA cumulated for all I/O pins)
- Maximum current available for 3.3V output: 50 mA
  Max current available for 5V output: Power function used – 500 mA max if USB port used alone
  Flash Memory Program: 32 KB (ATmega328) of which 0.5 KB are used by the Bootloader
  SRAM memory (volatile memory): 2 KB (ATmega328)
- EEPROM memory (non-volatile memory): 1 KB (ATmega328)

Clock speed: 16 MHz

II.2.10.Good networks to choose arduino:

However, there are a multitude of platforms in commerce that make it possible to do the same thing.Especially the microcontrollers «PIC» of the manufacturer Microchip. We will choose the Arduino for these reasons

• **Price**: In terms of performance, Arduino boards are relatively low

Knives, which is a major criterion for the beginner.

• **Freedom**: That's a big word, but it's a pretty concise definition of the spirit of the Arduino. Its two things in and of itself:

• **The software**: free and open source, developed in Java, the simplicity of its use depends on the click on the mouse.

• **Hardware**: electronic maps with diagrams that are freely available on the Internet. This freedom has one condition: the name «Arduino» must be used only for official maps». • **Unofficial cards** can be found and purchased on the Internet and are almost entirely compatible with the official Arduino cards.

•**Compatibility**: The software, like the card, is compatible on the most common platforms (Windows, Linux and Mac), unlike other commercial programming tools which are generally only compatible with Windows.

•**The community**: The Arduino community is impressive and the number of resources about it is constantly evolving on the internet. The references to the Arduino language and a full page of tutorials can be found on the website arduino.cc (in English) and arduino.cc (in French). [5]

II.2.11.Arduino Board Management software:

Arduino Having a Special Environment (IDE) is a java writing application inspired by the Processing language, IDE allows writing, modifying a program and converting it into a series of instructions that are understandable to the card. [4]

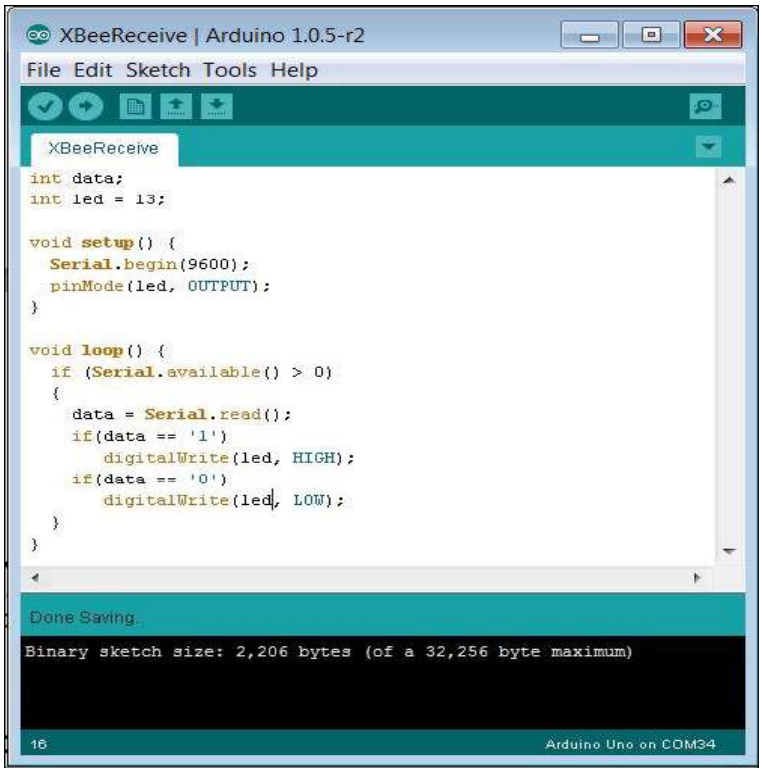


Figure II.6 Arduino software interface.

This blank window will be filled with words and numbers and other texts. This window is like any other software you have used. It contains menus, buttons, special alerts, and all kinds of controls: Open, Save, and the serial monitor button on the far right. New: creates a new sketch.

Open: presents a menu of all the sketches in your sketch folder. Click on one will open in the current window.

Record: Record your sketch.

Check: this button allows you to make sure that green software is example syntax error sending this button can both check and download a sketch to the Arduino if no spelling or formatting error is found.

Serial Monitor: This button allows you to open the serial monitor and to display the information from the serial port on arduino. The serial monitor is like a tool to talk with the Arduino and you use it to see things of interest [11]



Figure II.7 Button bar Detail.

Serial monitor

the serial monitor is used to display the information that is sent from the Arduino board to the application (usually via the USB cable). It also allows you to send information to the Arduino board. To do this, simply write text in the field at the top of the window and press the "Send" button. Of course, before you can do this you need to code an appropriate program, ready to receive this information. It is also possible to set the serial monitor baud rate, which specifies how fast the data transfer will take place. It is also possible to establish a communication.

Serial between the Arduino board and other peripherals or software.This function will be discussed in more detail in the "Glossy Text" and "Oscilloscope" projects. [4]

II.2.12. the advantages of the arduino [13]:

• it is cheap!

• Clear and simple programming environment.

• Cross-platform: runs on Windows, Macintosh and Linux.

• Many libraries available with various functions implemented.

• Open source and extensible software and hardware.

• Many tips, tutorials and online examples (forums, personal site etc...)

• Existence of "shield": these are additional boards that connect to the Arduino module to increase the possibilities such as color graphic display, Ethernet interface, GPS, etc...

II.3. Conclusion:

In this chapter we presented the Arduino board that we used to design the hardware architecture needed to control the mobile robot.

Chapter III

SLIDING MODE CONTROL FOR ROBOT

III.1. Introduction:

This chapter introduces three kinds of sliding mode controllers

Design for robot, including sliding mode controller design based on inputoutput

Stability, sliding mode controller design based on computed torque method and adaptive sliding mode controller design for manipulator.

III.2. Model of Robotic Joints:

III.2.1.Model Description:

In engineering, robots not only can improve productivity but also can achieve High-strength and hazardous jobs. Manipulators are the usual plants in robotics.

Consider an *n*-joint manipulator as follows:

$$\mathbf{H(q)\ddot{q} + c(q,\dot{q})\dot{q} + G(q) + f(\dot{q}) + \tau_d = \tau} \quad (11.1)$$

Where $\mathbf{q} \in \mathbf{R^n}$is the angle vector, $\mathbf{H(q)} \in \mathbf{R^{n\times n}}$is the inertia matrix,$\mathbf{C(q,\dot{q})} \in \mathbf{R^n}$

Denotes the centrifugal and coriolis forces$\mathbf{G(q)} \in \mathbf{R^n}$ is the gravity, $\mathbf{F(\dot{q})} \in \mathbf{R(n)}$ I

is the frictional force,$\tau \in R^n$is the control moment, and $\boldsymbol{\tau_d} \in \mathbf{R^n}$is the disturbance

moment.

The characteristics of the kinetic model of a manipulator [16]:

(1) Kinetic model contains more number of items. The number of items included

In the equation increases with the increase in the number of robot joints.

(2) Highly nonlinearity: Each item of the equations contains non-linear factors

Such as sine and cosine, et al.

(3) High degree of coupling.

(4) Model uncertainty and time-variant: The load will vary when the robot moves the objects because the objects are not similar. Also, the Joint frictiontorque will also change over time.

The characteristics of the kinetic model of a manipulator:

(1) $H(q)$Is a positive-definite symmetrical and bounded matrix, i.e., thereexist positive constants $m_1$and m_2 such that $\mathbf{m_1 I \leq H(q) \leq m_2 I}$;

(2) $\mathbf{c(q,\dot{q})}$is bounded, i.e., there exists known $\mathbf{c_b(q)}$such that$|\boldsymbol{c(q,\dot{q})}| \leq \boldsymbol{c_b(q)||\dot{q}||}$;

(3) Matrix $\boldsymbol{\dot{H} - 2C}$ isa skew-symmetric matrix, i.e.,$\mathbf{x^t(\dot{H} - 2C)x = 0}$, where$\boldsymbol{x}$ is a vector;

(4) The known disturbance is satisfied with $\mathbf{||\tau_d||} \leq \boldsymbol{\tau_m}$where $\boldsymbol{\tau_m}$is a knownpositive constant.

A classical dual-joint rigid manipulator is shown in **Fig.III.1.**

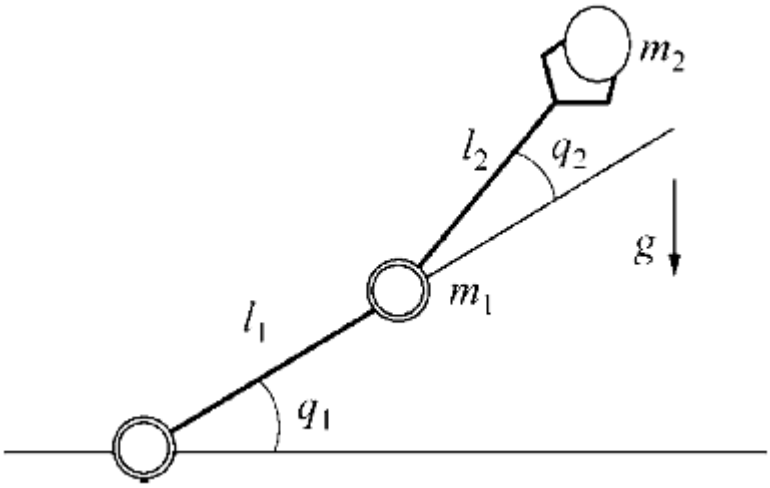


Figure III.1 A classical dual-joint rigid manipulator

III.2.2.Model Description Example:

Using the kinetic equation of a two-joint manipulator [18] as follows:

$$\boldsymbol{H(q)\ddot{q} + c(q,\dot{q})\dot{q} + G(q) = \tau} \qquad (11.2)$$

Where$\mathbf{q} = [q_1\ q_{2]}]^T$,$\boldsymbol{\tau} = [\tau_1\ \tau_2]^T$, and

$$\mathbf{H} = \begin{bmatrix} \alpha + 2\square\cos(q_2) + 2\eta\sin(q_2) & \beta + \square\cos(q_2) + \eta\sin(q_2) \\ \beta + \square\cos(q_2) + \sin(q_2) & \beta \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} (-2\,\square\sin(q_2) + 2\eta\cos(q_2))\dot{q}_2 & (-\,\square\sin(q_2) + \eta\cos(q_2))\dot{q}_2 \\ (\square\sin(q_2) - \eta\cos(q_2))\dot{q}_1 & 0 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \square e_1\cos(q_1+q_2) + \eta e_2\sin(q_1+q_2) + (\alpha-\beta+e_1)e_2\cos(q_1) \\ \square e_2\cos(q_1+q_2) + \eta e_2\sin(q_1+q_2) \end{bmatrix}$$

Where α, β, Ɛ and η are constants, α=$I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2$, β=$I_e + m_e l_{ce}^2$, Ɛ=$m_e l_1 l_{ce}\cos(\delta_2)$, η = $m_e l_1 l_{ce} sin(\delta_2)$.

As an illustration, the practical parameters of a manipulator are shown in TableIII.1[19]

Table III.1 Parameters of a dual-joint manipulator

m_1	l_1	l_{c1}	I_1	m_e	l_{ce}	I_e	δ_e	e_1	e_2
1 kg	1 m	1/2 m	1/12 kg	3 kg	1 m	2/5 kg	0	–7/12	9.81

Let **a**=$[\alpha \quad \beta \quad \varepsilon \quad \eta]^T$, and **â**is the estimation of **a**, **â**=$\left[[\hat{\alpha}\hat{\beta}\hat{\varepsilon}\hat{\eta}]^T\right]$.

We assume $\tilde{\mathbf{a}} = \hat{\mathbf{a}} - \mathbf{a}$, since **a** is a contant vector,$\dot{\tilde{\mathbf{a}}} = \dot{\hat{\mathbf{a}}}$.The estimated matrices of the above three matrices can be written respectively as :

$$\hat{\mathbf{H}} = \begin{bmatrix} \hat{\alpha} + 2\hat{\varepsilon}cos(q_2) + 2\hat{\eta}sin(q_2) & \hat{\beta} + \hat{\varepsilon}cos(q_2) + \hat{\eta}sin(q_2) \\ \hat{\beta} + \hat{\varepsilon}cos(q_2) + \hat{\eta}sin(q_2) & \hat{\beta} \end{bmatrix}$$

$$\hat{\mathbf{C}} = \begin{bmatrix} (-2\hat{\varepsilon}\sin(q_2) + 2\hat{\eta}\cos(q_2))\dot{q}_2 & (-\hat{\varepsilon}\sin(q_2) + \cos(q_2))\dot{q}_2 \\ (\hat{\varepsilon}\sin(q_2) - \hat{\eta}\cos(q_2))\dot{q}_1 & 0 \end{bmatrix}$$

$$\hat{\mathbf{G}} = \begin{bmatrix} \hat{\varepsilon}e_1\cos(q_1+q_2) + \hat{\eta}e_2\sin(q_1+q_2) + (\hat{\alpha}-\hat{\beta}+e_1)e_2\cos(q_1) \\ \hat{\varepsilon}e_2\cos(q_1+q_2) + \hat{\eta}e_2\sin(q_1+q_2) \end{bmatrix}$$

III.3.Sliding Mode Control Based on Input-Output Stability:

III.3.1.System Description:

For the system (11.2), we assume α,β, Ɛ and η are the unknown constant parameters.

Let$\mathbf{d_q(t)}$ denote the desired trajectory. The tracking error is

$$\mathbf{e}=\boldsymbol{q}_d-\boldsymbol{q}$$

Defline

$$\dot{\mathbf{q}}_\mathbf{r}=\dot{\mathbf{q}}_\mathbf{d}+\boldsymbol{\Lambda}(\mathbf{q}_\mathbf{d}-\mathbf{q})$$

Where$\boldsymbol{\Lambda}$is a positive diagonal matrix.

Let $\mathbf{p}=[\alpha \quad \beta \quad \varepsilon \quad \eta]^T$, $\hat{\mathbf{p}}$ is the estimation of **p**. Therefore, the vector **p**is a constant vector. Let $\tilde{\mathbf{p}}=\hat{\mathbf{p}}-\mathbf{p}$, therefore we can get $\dot{\tilde{\boldsymbol{p}}}=\dot{\hat{\boldsymbol{p}}}$**.**

According to the linear characteristic of robotics [16], we have

$$\mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r+\boldsymbol{G}(\boldsymbol{q})=\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\mathbf{p} \qquad (11.3a)$$

$$\tilde{\boldsymbol{H}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r+\tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r+\boldsymbol{G}(\boldsymbol{q})=\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{P}} \quad (11.3b)$$

Where $\tilde{\boldsymbol{H}}(\boldsymbol{q})=\boldsymbol{H}(\boldsymbol{q})-\tilde{\boldsymbol{H}}(\boldsymbol{q}),\tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$**=** $\mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ **-**$\tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$**,**$\tilde{\boldsymbol{G}}(\boldsymbol{q})=\boldsymbol{G}(\boldsymbol{q})-\tilde{\boldsymbol{G}}(\boldsymbol{q})$**,**

$\mathbf{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{q}_r,\dot{\boldsymbol{q}}_r)=$

$$\begin{bmatrix} \ddot{q}_{r1}+e_2\cos(q_1) & \ddot{q}_{r2}-e_2\cos(q_1) & \begin{matrix}2\cos(q_2)\ddot{q}_{r1}+\cos(q_2)\ddot{q}_{r2}\\ -2\sin(q_2)\dot{q}_2\dot{q}_{r1}\\ -\sin(q_2)\dot{q}_2\dot{q}_{r2}\\ +e_2\cos(q_1+q_2)\end{matrix} & \begin{matrix}2\sin(q_2)\ddot{q}_{r1}+\sin(q_2)\ddot{q}_{r2}\\ +2\cos(q_2)\dot{q}_2\dot{q}_{r1}\\ +\cos(q_2)\dot{q}_2\dot{q}_{r2}\\ +e_2\sin(q_1+q_2)\end{matrix} \\ 0 & \ddot{q}_{r1}+\ddot{q}_{r2} & \begin{matrix}\cos(q_2)\ddot{q}_{r1}+\sin(q_2)\dot{q}_1\dot{q}_{r1}\\ +e_2\cos(q_1+q_2)\end{matrix} & \begin{matrix}\sin(q_2)-\cos(q_2)\dot{q}_1\dot{q}_{r1}\\ +e_2\sin(q_1+q_2)\end{matrix} \end{bmatrix}$$

The expression of $\mathbf{Y}\ (\mathbf{q},\dot{\mathbf{q}},\mathbf{q}_\mathbf{r},\dot{\mathbf{q}}_\mathbf{r})$ is called the dynamic regression matrix, which

is deduced in Eq. (11.20).

III.3.2.Design of Controller:

The sliding variable is selected as

$$s=\dot{e}+\mathbf{\Lambda e}$$

Select the Lyapunov function as

$$V(t)=\frac{1}{2}s^{T}\mathbf{H(q)s}$$

Therefore,

$$\dot{V}(t)=\mathbf{s^{T}H(q)\dot{s}}+\frac{\mathbf{1}}{\mathbf{2}}\mathbf{s^{T}\dot{H}(q)s}=\mathbf{s^{T}H(q)\dot{s}}+\mathbf{s^{T}C(q,\dot{q})s}$$

$$=s^{T}[H(q)(\ddot{q}_r-\ddot{q})+C(q,\dot{q})(\dot{q}_r-\dot{q})]$$

$$=s^{T}[H(q)\ddot{q}_r+C(q,\dot{q})\dot{q}_r+G(q)-\tau] \quad (11.5)$$

The following methods can be adopted to realize the sliding mode control.

Method 1: Sliding mode control based on the estimated model

The controller is designed as

$$\boldsymbol{\tau}=\hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_{\mathbf{r}}+\hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_{\mathbf{r}}+\hat{\mathbf{G}}(\mathbf{q})+\boldsymbol{\tau}_{s} \quad (11.6)$$

Where τ_s is the robustness element to be designed

From Eqs. (11.5) and (11.6), we have:

$$\dot{V}(t)=s^{T}\left[H(q)\ddot{q}_r+C(q,\dot{q})\dot{q}_r+G(q)-\hat{H}(q)\ddot{q}_r-\hat{C}(q,\dot{q})\dot{q}_r-\hat{G}(q)-\tau_s\right]$$

$$=s^{T}\left[\tilde{H}(q)\ddot{q}_r+\tilde{C}(q,\dot{q})\dot{q}_r+\tilde{G}(q)-\tau_s\right]=S^{T}[Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\tilde{p}-\tau_s]$$

Where

$$\tilde{\mathbf{P}}=[\tilde{P_1}\quad\tilde{P_2}\quad\tilde{P_3}\quad\tilde{P_4}]^{T},\qquad |\tilde{P}_i|\le\overline{\tilde{q}_i},\qquad i=1,2,3,4$$

$$\mathbf{Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)}=[Y_{ij}],\qquad |Y_{ij}|\le\overline{Y_{ij}},\qquad i=1,2\qquad j=1,2,3,4$$

Therefore, we select

$$\boldsymbol{\tau_s} = \mathbf{K}\, sgn(\mathbf{s}) + \mathbf{s} = \begin{bmatrix} k_1\, sgn(s_1) + s_1 \\ k_2\, sgn(s_2) + s_2 \end{bmatrix} \qquad (11.7)$$

Where $k_i = \sum_{j=1}^{4} \overline{Y_{ij}}\,\overline{\tilde{p}_j}, i = 1,2.$

Therefore, we have

$$\dot{V}(t) = \sum_{i=1}^{2}\sum_{j=1}^{4} s_i\, Y_{ij}\tilde{p}_j - \sum_{i=1}^{2} s_i k_i sgn(s_i) - \sum_{i=1}^{2} s_i^2$$

$$= \sum_{i=1}^{2}\sum_{j=1}^{4} s_i\, Y_{ij}\tilde{p}_j - \sum_{i=1}^{2}\sum_{j=1}^{4} |s_i|\,\overline{Y_{ij}}\,\overline{\tilde{p}_j} - \sum_{i=1}^{2} s_i^2 \le -\sum_{i=1}^{2} s_i^2 \le 0$$

Method 2: Sliding mode control based on the bound of modelEquation (11.5) can be written as:

$$\dot{\mathbf{v}}(\mathbf{t}) = -\mathbf{S^T}[\boldsymbol{\tau} - (\mathbf{H(q)}\ddot{\mathbf{q}}_\mathbf{r} + \mathbf{C(q,\dot{q})}\dot{\mathbf{q}}_\mathbf{r} + \mathbf{G(q)})]$$

$$= -\boldsymbol{S^T}[\boldsymbol{\tau} - \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{P}]$$

The controller can be designed as:

$$\boldsymbol{\tau} = \bar{\mathbf{k}} sgn(\mathbf{s}) + \mathbf{s} = \begin{bmatrix} \overline{k_1}\, sgn(s_1) + s_1 \\ \overline{k_2}\, sgn(s_2) + s_2 \end{bmatrix} \qquad (11.8)$$

Where $\bar{k}_i = \sum_{j=1}^{4} \overline{Y_{ij}}\,\bar{p}_j \quad , \qquad i = 1,2.$

Therefore, we have

$$\dot{v}(t) = -\left[\sum_{i=1}^{2} s_i \bar{k}_i\; sgn(s_i) + \sum_{i=1}^{2} s_i^2 - \sum_{i=1}^{2}\sum_{j=1}^{4} s_i\, Y_{ij} p_j\right]$$

$$= -\left[-\sum_{i=1}^{2}\sum_{j=1}^{4} |s_i|\,\overline{Y_{ij}}\,\overline{P_j} + \sum_{i=1}^{2} s_i^2 - \sum_{i=1}^{2}\sum_{j=1}^{4} s_i\, Y_{ij}\, P_j\right] \le -\sum_{i=1}^{2} s_i^2 \le 0$$

From the switching gain $\bar{k}_i$in controller (11.8) and the switching gain $k_i$in Eq. (11.7), $\bar{k}_i$is larger than. k_iTherefore, the chattering generated by controller (11.8) is larger than that generated by controller (11.6).

III.3.3.Simulation Example:

We consider the plant as Eq. (11.2). The desired trajectories are $q_{d1} = sin(2\pi t)$ and $q_{d2} = sin(2\pi t)$. In sliding mode control, select$\mathbf{\Lambda} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$. When *M*_1, thecontroller is given in Eq. (11.6) where $\bar{\bar{P}}_I = |\tilde{P}_I| + 0.50$.the saturated function isused instead of the switch function where $\Delta = 0.05$.the simulation results areshown in Figs. 11.2 -11.4. Simultaneously, when *M*= 2, the controller is given in

Eq. (11.8) where $\bar{\bar{P}}_i = |p_i| + 0.50$.the same simulation results are obtained.

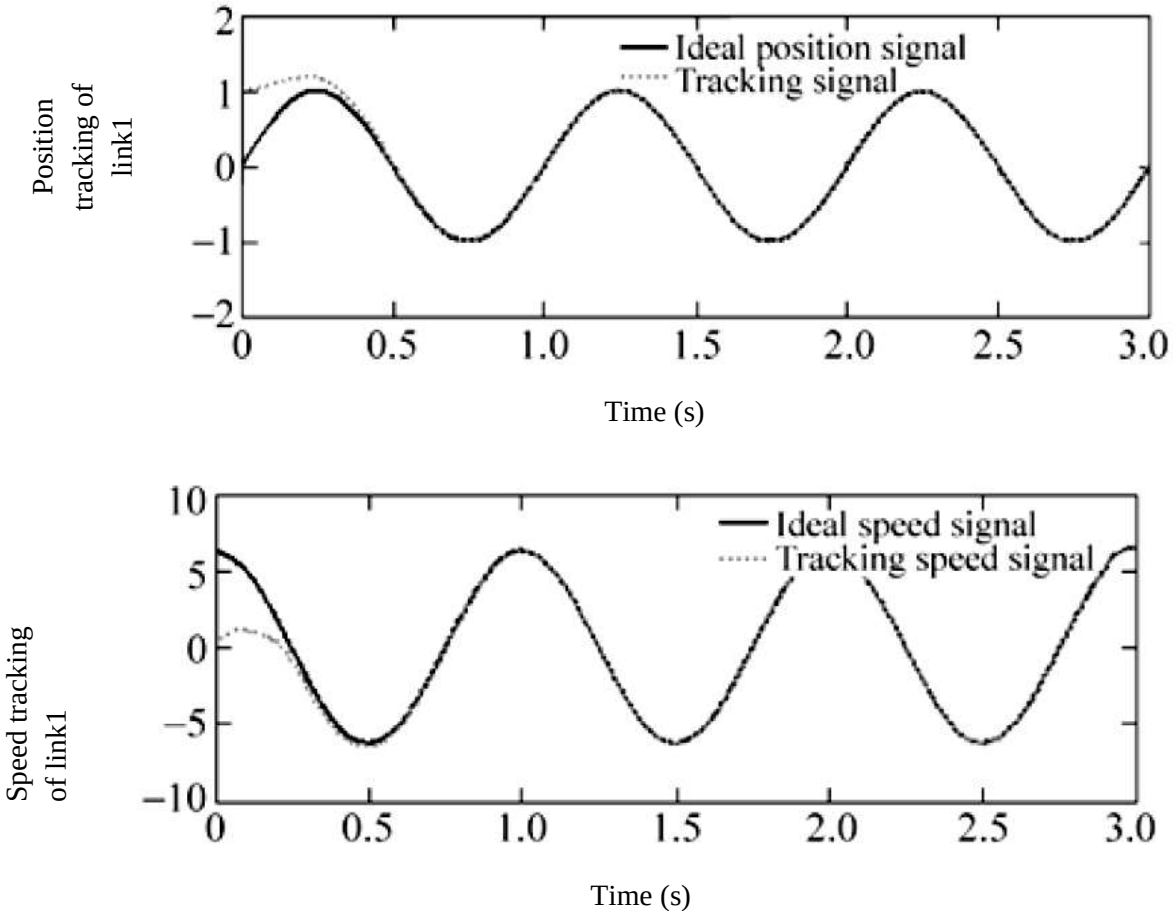
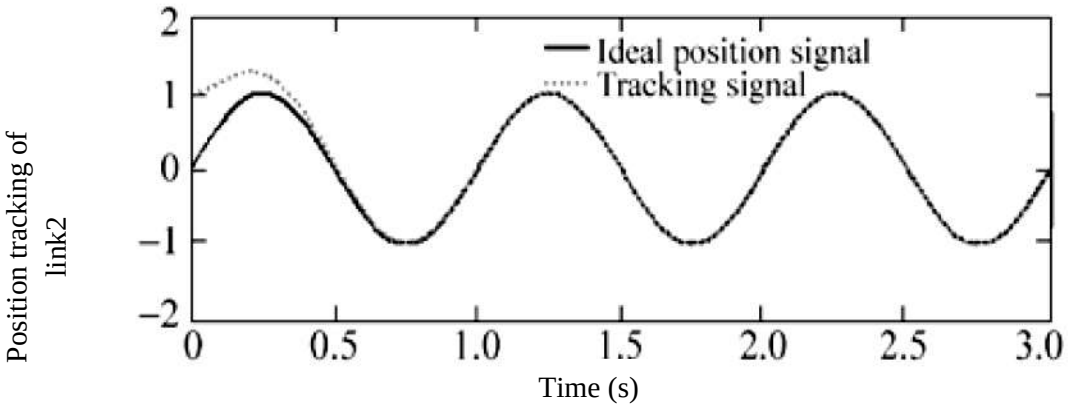


Figure III.2 Tracking of position and velocity for the first joint



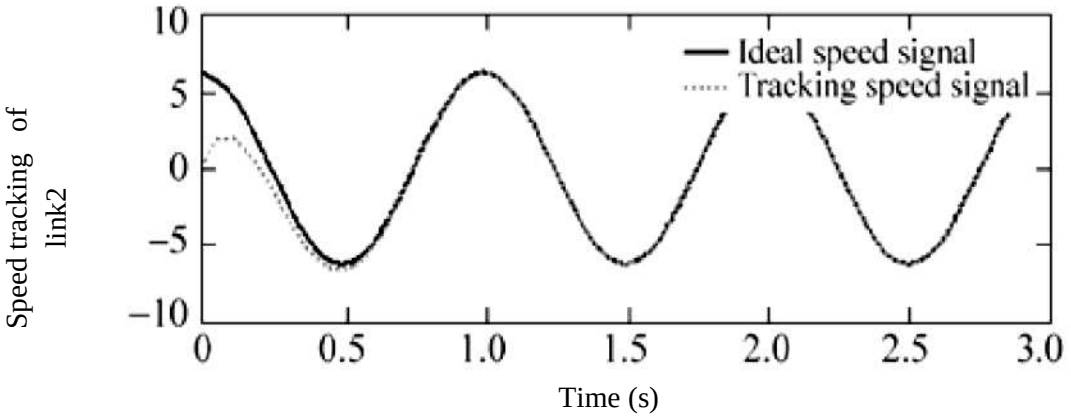
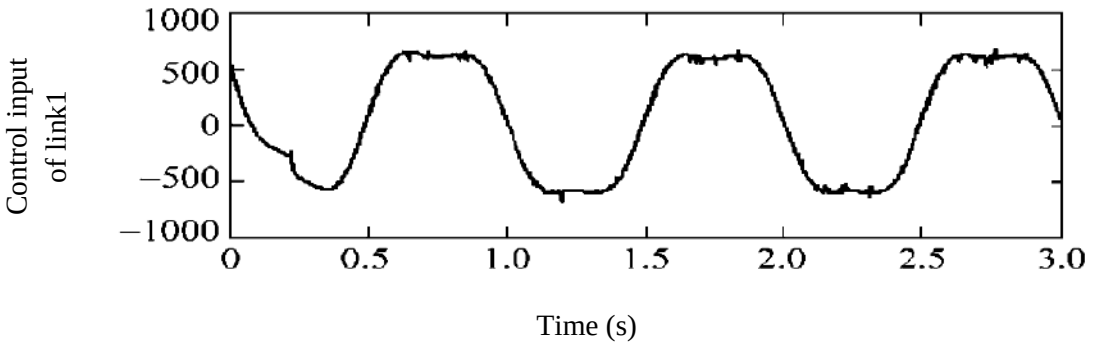


Figure III.3 Tracking of position and velocity for the second joint



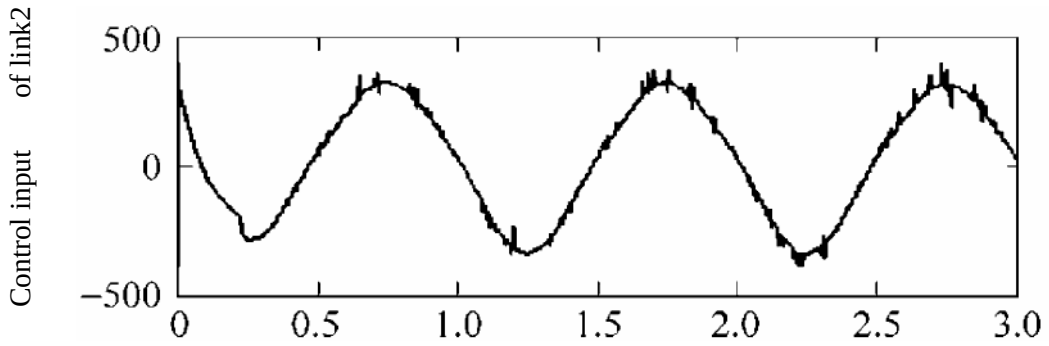


Figure III.4 Control input

III.4.Sliding Mode Control Based on Computed Torque Method:

III.4.1.Design of Controller:

Using the computed torque method, the control law can be designed for system

(11.2) as follows:

$$\tau = \hat{H}(q)v + \hat{c}(q\dot{q})\dot{q} + G(q) \qquad \textbf{(11.9)}$$

Where **v**is the auxiliary control input,$\hat{\mathbf{H}}(\mathbf{q})$,$\mathbf{C}(\mathbf{q}\ \dot{\mathbf{q}})$ and $\hat{G}(q)$are the respective estimations of ***H, C*** and **G**by using inertial parameter $\hat{\mathbf{P}}$ of manipulator.

Substituting Eq. (11.9) into Eq. (11.1), neglecting $F(\dot{q})$ and τ_d , considering

Eq. (11.3b), we have

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \hat{H}(q)v + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) \qquad \textbf{(11.10)}$$

i.e.

$$\hat{H}\ddot{q} = \hat{H}(q)v - \left[\tilde{H}(q)\ddot{q} + \tilde{C}(q,\dot{q})\dot{q} + \tilde{G}(q)\right] = \hat{H}(q)v - Y(q,\dot{q},\ddot{q})\tilde{P}$$

Where

$$\tilde{H} = H - \hat{H}, \tilde{C} = C - \hat{C}, \tilde{G} = G - \hat{G}, \tilde{P} = P - \hat{P}$$

If the parameter estimation $\hat{\mathbf{P}}$ makes $\hat{\mathbf{H}}(\mathbf{q})$ invertible, then the above equation

CanbeWritten as:

$$\ddot{q} = v - (\hat{H}(q))^{-1}Y(q,\dot{q},\ddot{q})\tilde{p} = v - \tilde{d}$$

Where$\varphi(q,\dot{q},\ddot{q},\hat{p}) = (\hat{H}(q)^{-1}Y(q,\dot{q},\ddot{q}), \varphi(q,\dot{q},\ddot{q},\hat{p})\tilde{p} = \tilde{d}.$

The sliding variable can be selected as:

$$\mathbf{s = \dot{e} + \Lambda e} \qquad \textbf{(11.11)}$$

Where$\mathbf{e = q_d - q}, \dot{\mathbf{e}} = \dot{\mathbf{q}}_\mathbf{d} - \dot{\mathbf{q}}, \boldsymbol{s} = [s_1 s_2 \dots s_n]^T$,**Λ**is a positive diagonal matrix.

Therefore,

$$\dot{\mathbf{s}} = \ddot{\mathbf{e}} + \boldsymbol{\Lambda}\dot{\mathbf{e}} = \ddot{\mathbf{q}}_\mathbf{d} - \ddot{\mathbf{q}} + \boldsymbol{\Lambda}\dot{\mathbf{e}} = \ddot{\mathbf{q}}_\mathbf{d} - \mathbf{v} + \tilde{\mathbf{d}} + \boldsymbol{\Lambda}\dot{\mathbf{e}}$$

Select

$$\mathbf{v} = \ddot{\mathbf{q}}_\mathbf{d} + \mathbf{\Lambda}\dot{\mathbf{e}} + \mathbf{d} \qquad \textbf{(11.12)}$$

Where $\mathbf{d} = \left(\bar{\tilde{\mathbf{d}}} + \square\right)\mathbf{sgn}(\boldsymbol{s}), ||\tilde{\mathbf{d}}|| \leq \bar{\tilde{\mathbf{d}}}, \square > \mathbf{0}.$

Therefore, we have

$\dot{\boldsymbol{s}} = \dot{\tilde{\mathbf{d}}} - \mathbf{d}$

The Lyapunov function is selected as $\mathbf{v} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{s}^\mathbf{T}\mathbf{s}$, therefore, we have

$$\dot{v} = \boldsymbol{S}^T\dot{\boldsymbol{S}} = \boldsymbol{S}^T\left(\tilde{\boldsymbol{d}} - \boldsymbol{d}\right) = \boldsymbol{S}^T\tilde{\boldsymbol{d}} - \bar{\tilde{\boldsymbol{d}}}\boldsymbol{S}^T\mathrm{sgn}(\boldsymbol{s}) - \square\boldsymbol{S}^T\mathrm{sgn}(\boldsymbol{s}) \leq -\square||\boldsymbol{S}|| \leq \mathbf{0}$$

From the controller, the more precise the estimation of $\hat{p}$ is, the smallerare $||\tilde{p}||$ and$\bar{\tilde{\boldsymbol{d}}}$, and smaller is the chattering of the sliding mode.

III.4.2 Simulation Example:

We consider the plant as Eq. (11.2) the initial state vector is$[0\ 0\ 0\ 0]^T$. Thesliding mode controller is given in Eqs. (11.9) and (11.12). The desired trajectoryis

$q_{d1} = \sin t \quad , q_{d2} = \sin t \quad , \hat{H} = 0.6H \quad , \hat{C} = 0.6C, \hat{G} = 0.6G, \bar{\tilde{d}} = 30, \square = 0.10,$

$\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{25} & \mathbf{0} \\ \mathbf{0} & \mathbf{25} \end{bmatrix}$The simulation results are shown in Fig. 11.5 and Fig. 11.6.

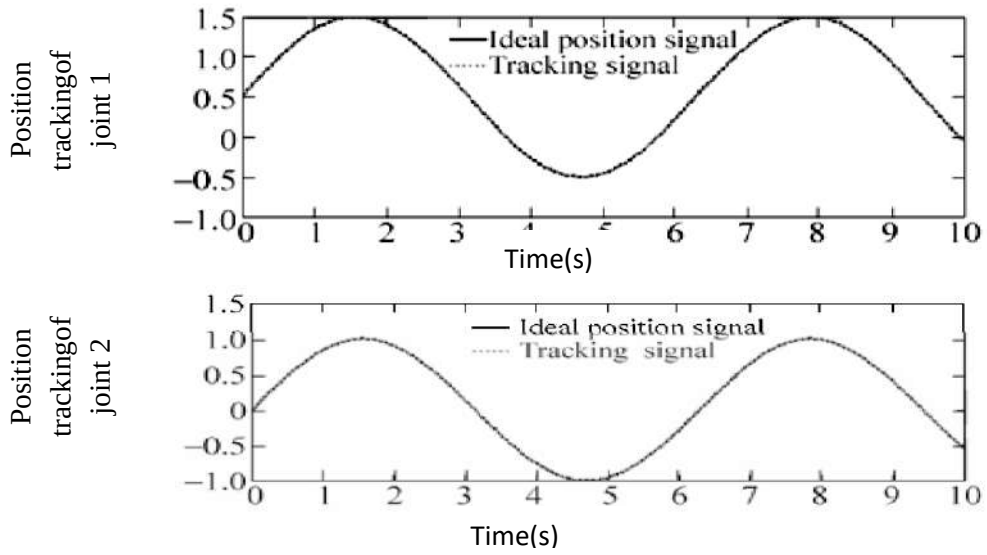


Figure III.5 The position tracking of a dual-joint manipulator

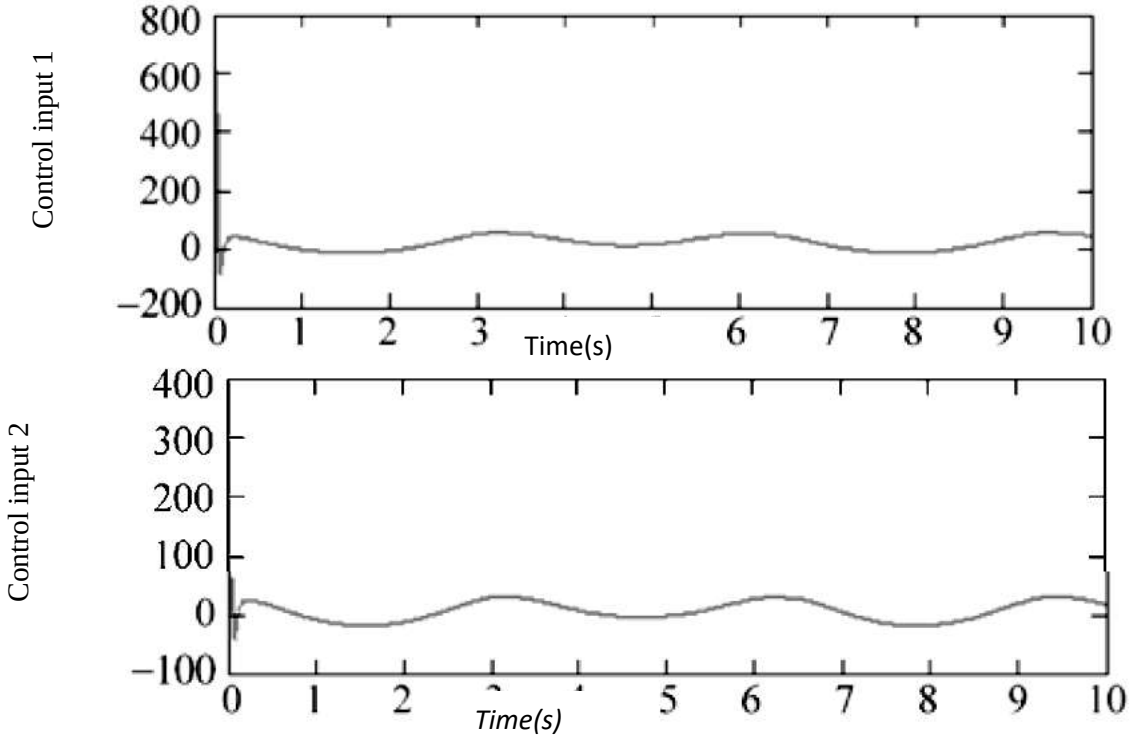


Figure III.6 The control input of a dual-joint manipulator

III.5. Adaptive Sliding Mode Control for Manipulator

We discussed an adaptive sliding mode controller design method for the manipulator.

Refer to paper [17].

III.5.1Adaptive Sliding Mode Controller

For the system (11.2), we assume $\alpha, \beta, \varepsilon$and $\eta$are the unknown constantParameters.

Let $\tilde{q}(t) = \mathbf{q}\ (\mathbf{t})\ - q_d(\mathbf{t})$be the tracking error, and denote

$$\dot{q}_r = q_d - \wedge\tilde{q}, \ddot{q}_r = \dot{q}_d - \wedge\dot{\tilde{q}}$$

Where $\Lambda=\begin{bmatrix} 0 & \lambda_1 \\ \lambda_2 & 0 \end{bmatrix}, \lambda_i > 0, i = 1.2.$

Define the sliding variable as:

$$\mathbf{S} = \dot{\tilde{q}} - \wedge\tilde{q}$$

Slotine et al. proposed the controller [17][18] as:

$$\mathbf{T} = \widehat{H}(\mathbf{q})\,\ddot{q}_r + \widehat{C}(\mathbf{q},\dot{q})\,\dot{q}_r + \widehat{G}\mathbf{p} - K_d s$$

Where$\mathbf{K_d} = \begin{bmatrix} \mathbf{K_{d_1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{d_2}} \end{bmatrix}, \mathbf{K_{d_i}} > 0, i = 1.2.$

Because ***H*** is a positive-definite matrix, we select the Lyapunov function as

$$\mathbf{V(t)} = \frac{\mathbf{1}}{\mathbf{2}} s^T \mathbf{H} s + \frac{\mathbf{1}}{\mathbf{2}} \tilde{a}^t \mathbf{T} \tilde{a}$$

Where$= \boldsymbol{T} = \begin{bmatrix} \gamma_1 & 0 & 0 & 0 \\ 0 & \gamma_2 & 0 & 0 \\ 0 & 0 & \gamma_3 & 0 \\ 0 & 0 & 0 & \gamma_4 \end{bmatrix}, \gamma_i > 0$,i=1,2,3,4.

Therefore, we have

$$v(t) = s^T H \dot{s} + \frac{1}{2} s^T \dot{H} s + \tilde{a}^t \mathbf{T} \dot{\tilde{a}} = \dot{s}^T (\mathbf{H}\ddot{q} - \mathbf{H}\ddot{q}_r) + \frac{1}{2} s^T \dot{H} s + \tilde{a}^t \mathbf{T} \dot{\tilde{a}}$$

$$= s^T (\tau - C\dot{q} - G - \mathbf{H}\ddot{q}_r) + \frac{1}{2} s^T \dot{H} s + \tilde{a}^t \mathbf{T} \dot{\tilde{a}}$$

$$= s^T (\tau - C(s - \dot{q}) - G - \mathbf{H}\ddot{q}_r) + \frac{1}{2} s^T \dot{H} s + \tilde{a}^t \mathbf{T} \dot{\tilde{a}}$$

From Eq. (11.15), we can get

$$\dot{v}(t) s^T (\tilde{\mathbf{H}}\ddot{q}_r + \tilde{C}\dot{q_r} + \tilde{G} + K_d s - C(s - \dot{q}) - G - \mathbf{H}\ddot{q}_r) = + \frac{1}{2} s^T \dot{H} s + \tilde{a}^T T \dot{\tilde{a}}$$

$$= s^T (\tilde{H}\ddot{q}_r + \tilde{C}\dot{q_r} + \tilde{G} + K_d s - Cs) \frac{1}{2} s^T \dot{H} s + \tilde{a}^T + T\dot{\tilde{a}}$$

From the linear characteristic of the kinetic equation of the manipulator [18], just

Like Eq. (11.3b), it follows:

$$\tilde{\mathbf{H}}\ddot{\mathbf{q}}_\mathbf{r} + \tilde{\mathbf{C}}\dot{\mathbf{q}}_\mathbf{r} + \tilde{\mathbf{G}} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{q}_\mathbf{r}, \dot{\mathbf{q}}_\mathbf{r}, \ddot{\mathbf{q}}_\mathbf{r})\tilde{\mathbf{a}}$$

Therefore,

$$v(T) = s^T (Y\tilde{a} + K_d s - \dot{C}s) + \frac{1}{2} s^T \dot{H} s + \tilde{a}^T \dot{\tilde{T}} \tilde{a}$$

$$= s^T (Y\tilde{a} + K_d s) \frac{1}{2} s^T (\dot{H} - 2C) s + \tilde{a}^T + T\dot{\tilde{a}}$$

$$= s^T (Y\tilde{a} + K_d s) + \tilde{a}^T T \dot{\tilde{a}} = \tilde{a}^T Y^T s - s^T K_d s - \tilde{a}^T T \dot{a}$$

$$= \tilde{a}^T (Y^T s + T\dot{\tilde{a}}) - s^T K_d s$$

Adaptive rule [17][18] was designed by slotine et al. as follows:

$$\dot{\hat{a}} = -T^{-1}Y^{T}s \qquad \textbf{(11.18)}$$

Therefore,

$$\dot{v}(T) = s^{T}K_{d}s \leq 0$$

Then$\tilde{\mathbf{q}} \to 0$as $t \to \infty$.

III.5.2 Simulation Example

We consider the plant as Eq. (11.2). The desired trajectories of the two arms are

$q_{d1} = \sin(2\pi t)\, and\ q_{d2} = \sin(2\pi t)$ Respectively. The controller and the adaptive rule

Are shown in Eqs.(11.15) and (11.18) respectively.$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{q_r}, \dot{\mathbf{q}}_\mathbf{r})$ Is given in

$$Eq.(11.20).Select\wedge = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, K_d = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(See the appendix). The simulation results are shown in Fig. 11.7 _ Fig. 11.10. We can see

From Fig. 11.9 and Fig. 11.10 that the effects of identification to, $\beta, \varepsilon\ and\ \eta$are not satisfied. The convergence of trajectories tracking can be guaranteed bythe adaptive rule. However, the estimations cannot converge to the desired values,

The reason is explained in Section 6.1. The information of desired trajectories must

Be abundant [17] to improve the estimation precision.

III.6.Conclusion:

At the end of this module we have learned three types of Sliding Mode Control Modules.

Chapter IV

CONTROL OF AERIAL MANIPULATOR

IV.1.Introduction:

Aerial Manipulator: Rotorcraft Equipped with a robotic arm

Current research in Aerial Manipulator:

Installing a gripper at the bottom of UAV to hold a payload

Suspending payload with cables

Autonomous navigation

Use:

Inspection and maintenance of power lines

Rescue operations

Construction in inaccessible sites

IV.2.Dynamics of combined system:

- Based on Euler-Lagrangian formulation

$$M(q)\ddot{q} + c(q,\dot{q})\dot{q} + G(q) = \tau + \tau_{ext} \qquad (1)$$

- Where, q is the generalized joints vector of 6+ n components. Inthis case, the arm has 2 DOF and n = 2.
- M is the inertia matrix, C is the centripetal/Coriolis matrix and G isthe gravity.

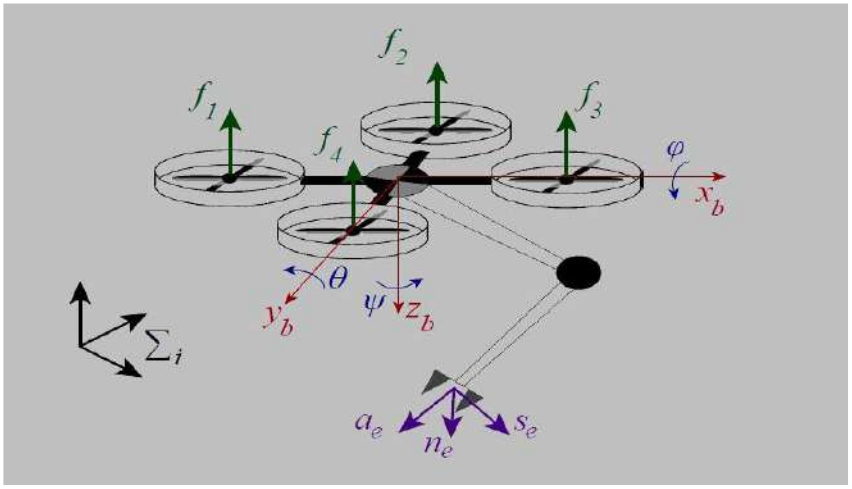


Figure IV.1 UAV/Arm system with corresponding reference frame [19].

IV.3.Control problem:

- Complicated coupled dynamics
- Displacement of center of mass
- Variation of mass distribution, moment of inertia

IV.4.Control Approach

IV.4.1.Cartesian Impedance Control:

Control Law:

$$\dot{X} = J_a \dot{q} \quad (2)$$

$$\ddot{X} = J_a \ddot{q} + \dot{J}_a \dot{q} \quad (3)$$

$$\tilde{X} = X_d - X \quad (4)$$

$$\tau = J_a^T f \quad (5)$$

From (1), (2), (3),(3),(4) & (5),the joint space dynamics can betransformed into manipulator end-effector Cartesian coordinates(x).

Here, $J_a$is the Jacobian of the system. τIs the generalized torquerelated to four motor thrust and joint actuator?

$$M_X \ddot{X} + C_X(X, \dot{X})\dot{X} + G_X(X) = f + f_{est} \quad (6)$$

Where,$M_X = J_a^{-T} M_q J_a^{-1}$,$C_X = J_a^{-T}(c_q - M_q J_a^{-1} j_a) J_a^{-1}$,

$G_X = J_a^{-T} G_q$,$f = J_a^{-T}\tau$, f_eXT=$J_a^{-T}\tau_{ext}$

- Following control law is defined whereK_P and $K_D$are$n_q * n_q$symmetric and positive definite matrices.

$$\tau = G + G_a^T(M_X \ddot{X}_d) + C_X \dot{X}_d + K_D \dot{\tilde{X}} + K_P \dot{X} \quad (7)$$

- Closed loop dynamics of the system is:

$$M_X \ddot{\tilde{X}} + (C_X + K_D)\dot{\tilde{X}} + K_P \tilde{X} = f_{ext} \quad (8)$$

StabilityAnalysis:

- Positive definite candidate Lypanuv function

$$V(X, T) = \frac{1}{2}\dot{\tilde{X}}^T M_X \dot{\tilde{X}} + \frac{1}{2}\tilde{X}^T K_P \tilde{X} \quad (9)$$

Time derivative of V:

$$\dot{V} = -\dot{\tilde{X}}^T K_D \dot{\tilde{X}} + \dot{\tilde{X}}^T f_{ext} \quad \textbf{(10)}$$

- If $f_{ext} = 0$for free motion case, Using Theorem 8.5, from Khalil, wecan conclude that $\dot{\tilde{X}} \to 0 as t \to \infty$.

- In case of constraint motion,$f_{ext} \neq 0$ only stability can beensured.

Simulation Results:

- ASCTEC PELLICAN,$m_b = 2k, I_b = diag(1.24, 1.24, 2.48)$
- 2-DOF robotic arm, $I_1 = 15cm I_2 = 5cm$with 3 revolute joints,
- $m_1 = 0.05\ kg, I_1 = 0.0019 m^2 kg, m_2 = 0.05\ kg, I_1 = 0.0011 m^2 kg$
- Focus on hovering control
  - initial conditions,$q = [0,0,0,0,0,0,0,\frac{\pi}{2}]^T$
  - desired position,$q_d = [0,0,6,0,0,0,0,0,0.5,0.5]$

- A force of 1 N is modelled through of sine wave of $\frac{\pi}{4}$rad/s whichacts along the $X_b$axis and another force of 0.5 N magnitude actsalong the x-direction of end-effector of 20 s to model the windysituation.

Simulation Results:$k_p = 80 I_8, K_D = 16 I_8$

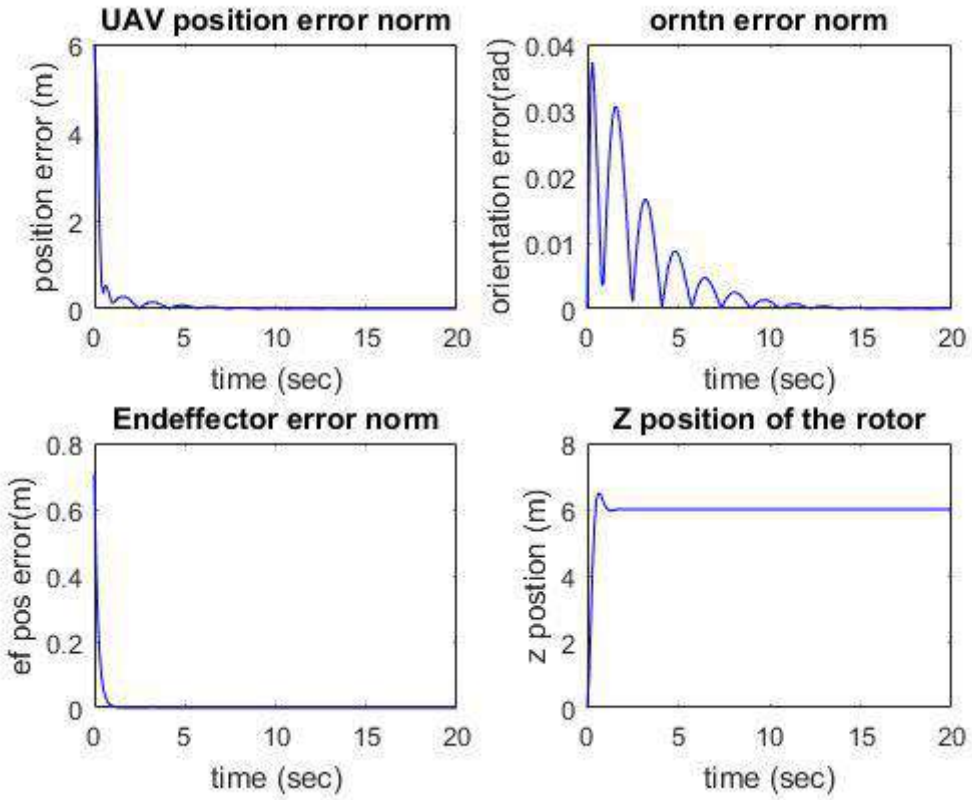


Figure IV.2 Time histories simulation for Rigid Case.

Simulation Results: $\mathbf{K_P} = 5I_8$;$\mathbf{K_D} = 2I_8$

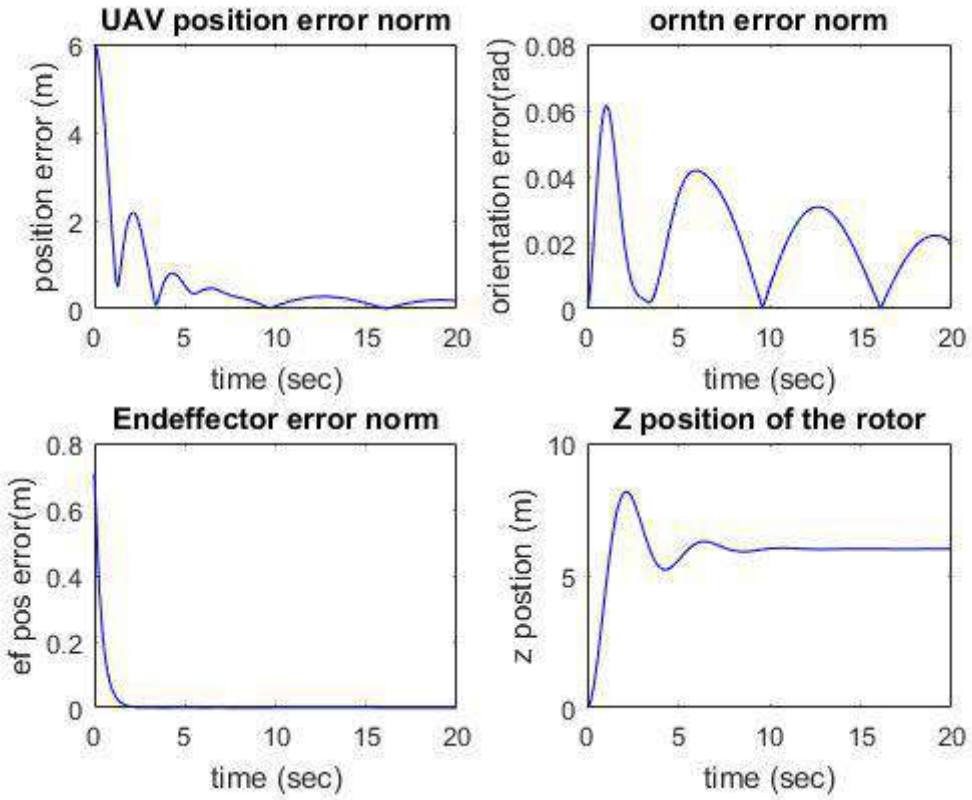


Figure IV.3 Time histories simulation for Compliant Case.

Simulation Results: $K_p = diag(10I_6, 100I_2)$, $K_D = diag(3I_6, 50I_2)$

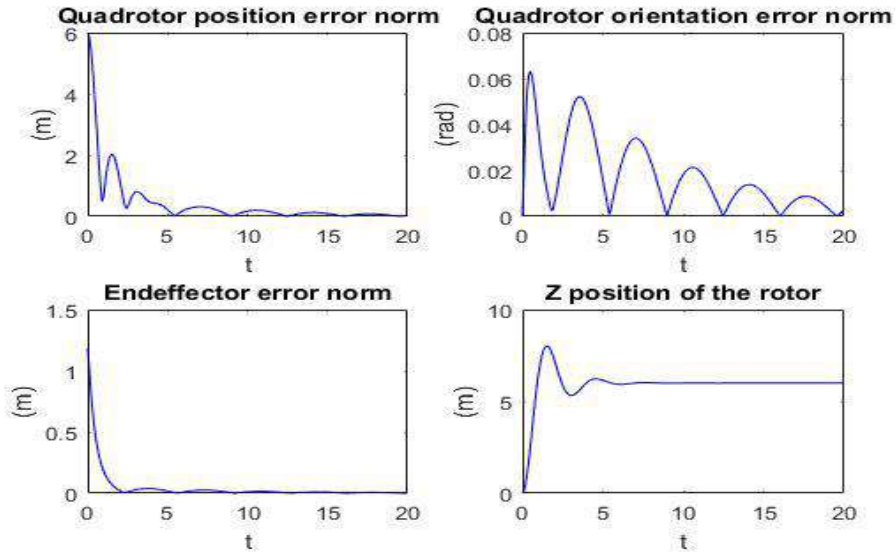


Figure IV.4 Time histories simulation for Compliant Rigid Case.

IV.4.2.Adaptive Sliding Mode Control

Control Law:

$$e = q - q_d \quad (11)$$

$$s = \dot{e} - \wedge e \quad (12)$$

$$q_r = q - \dot{S} = \dot{q}_d - \wedge e \quad (13)$$

To design the controller, we define error e and sliding surface s. Now the control law becomes:

$$\tau = \widehat{M}\,\ddot{q}_r + \hat{C}\dot{q}_r + \hat{\Delta} - As - Ksgn(s) \quad (14)$$

Here, $\widehat{M}, \hat{C}, \hat{G}$represents estimation of each matrix. A and K are Positive gain matrices.

$$M\dot{s} + Cs + \wedge s = -\Delta - \hat{\Delta} - Ksgn(s) \quad (15)$$

Where

$$\Delta = \tilde{M}\ddot{q}_r - \tilde{C}\dot{q}_r - \tilde{G} \quad (16)$$

Stability Analysis: Let a Lyapunov candidate function be

$$V = \frac{1}{2}s^T Ms + \frac{1}{2}\tilde{\Delta}^T\tilde{\Delta} > 0 \quad (17)$$

It time derivative is:

$$\dot{V} = \frac{1}{2}s^T(-\wedge s + \tilde{\Delta} - Ksgn(s) + \tau_{ext}) + \tilde{\Delta}^T\dot{\tilde{\Delta}} = -s^T\wedge s - S^T(Ksgn(s) - \tau_{ext}) \leq 0 \quad (18)$$

IV.4.2.Adaptive Cartesian Impedance Control

Adaptation law:

$$\dot{\hat{\Delta}} - \dot{\Delta} = -s \quad (19)$$

With the assumption that _ changes very slow compared with the adaptation rate, we get

$$\dot{\hat{\Delta}} = -s \quad (20)$$

Free motion: If we assume that$\tau_{ext} = 0$, from (19) and (20), we can conclude that

$s \to 0$As$t \to 0$, so $e \to 0$ exponentially.

If $\tau_{ext} \neq 0$, we can conclude that s is bounded, $so \Rightarrow e$ is bounded.

Simulation Results:

- Design Parameters: $\mathbf{K} = \mathbf{I}_{8*8}, \wedge = 2I_{8*8}, A = 10I_{8*8}$

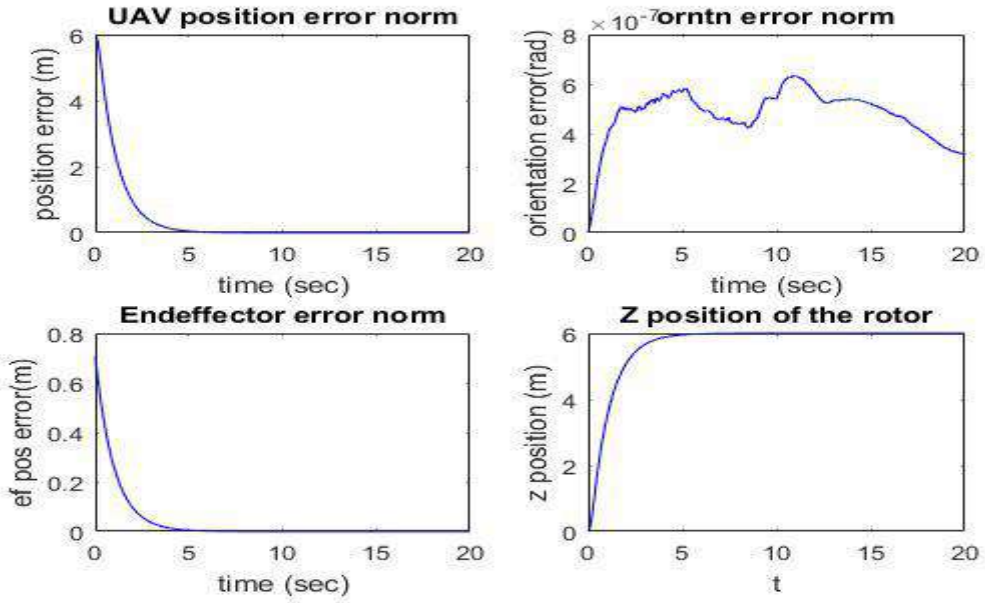


Figure IV.5 Time histories simulation for adaptive sliding mode controller.

Simulation Results: Position History:

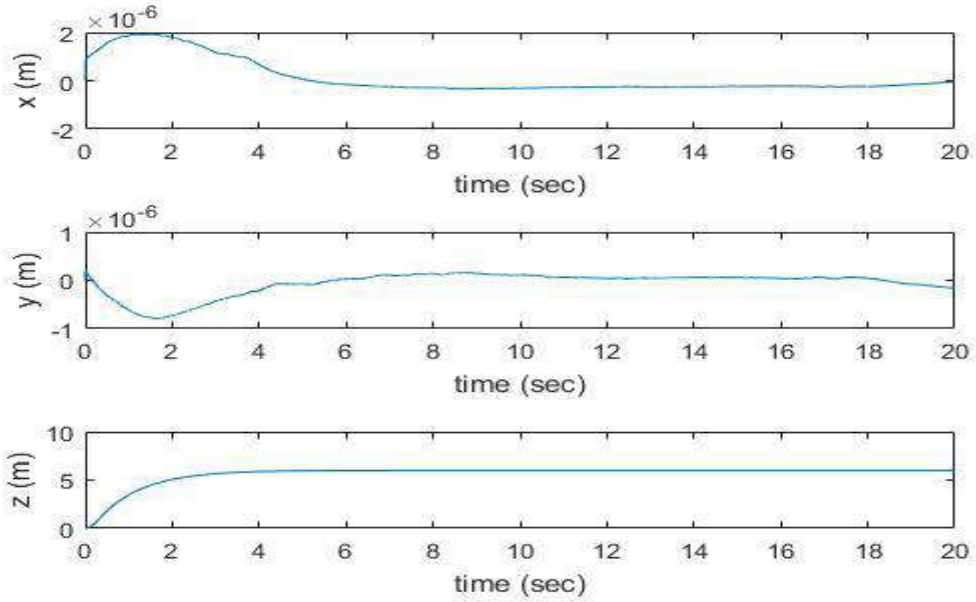


Figure IV. 6 Position history of the quadrotor.

Simulation Results: Attitude History:

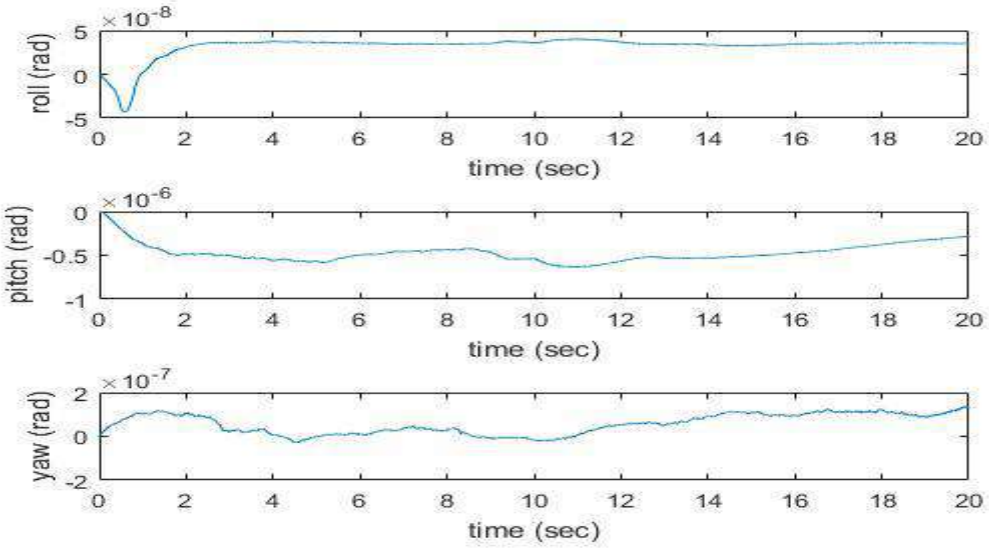
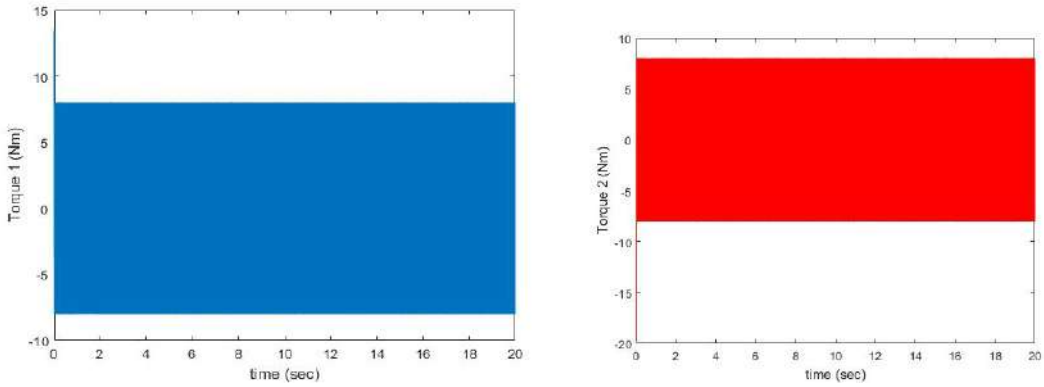


Figure IV.7 Attitude history of the quadrotor.

Simulation Results: Torque plot



(a) Torque1 History. (b) Torque2 History

Figure IV.8 Torque History for both links

IV.6.Conclusion and Future Work

- Coupled dynamics of aerial manipulator makes the controlProblem challenging
- Computation of C matrix
- Can implement a hierarchical control separately for both arm andthe rotor.
- Future work on other applicable control strategies and platform forTesting

GENERAL CONCLUSION

General Conclusion:

In recent years, electronics has been an integral part of our lives and tends to become widespread in all areas where miniaturization, mobility, computing and computer power are needed.

In this context, The objective of our work is the remote control of a mobile robot via Arduino, The study is based on a three-wheel omnidirectional robot.

A control strategy was implemented to find a remote robot control solution, For the moving system, the Arduino UNO was used, whose special features have helped us to facilitate the tasks especially regarding its programming, and an android app. Due to lack of time and material we used a Matlab program to achieve our project goal This work allowed us to deal with practical problems and to verify theoretical knowledge acquired throughout our research.

This work is only a start in the field of mobile robotics.The robot lacks important features such as travel autonomy, artificial intelligence, camera, etc. etc.In the near future, we want to address these concepts in order to manufacture a high-quality mobile robot.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] E. D. M. Professionnel, "Voice command of a rover via Bluetooth à base d'Arduino," 2018.

[2] “THE GESTUELLE CONTROL OF A MOBILE ROBOT (ARDUINO).“ .

[3] M. En and G. Electonique, “Option: Electronics of Embedded Systems Realisation and control of a Robot detector of obstacles Dédicaces,” 2019.

[4] F. A. Des, S. C. ET, and D. E. L. A. T. Echnology, “Department of Electrical Engineering and Informatics Implementation of a neuronal control for the visual control of a mobile robot Proposed and directed by;”

[5] P. Minist, E. Sup, R. S. Universit, M. Facult, A. Sp, and I. Industrielle, “Mémoire de F in d’études s,” 2019.

[6] M. D. E. F. I. N. D. Etudes, “Remerciements,” 2012.

[7] D. Filliat, R. Mobile, R. Mobile, and E. Paristech, “To cite this version: HAL Id: cel-00655005 Robotique Mobile,” 2011.

[8] R. Algerienne and D. ET, “Remerciements,” 2019.

[9] Mr. Saadi, N. Guersi, and N. Debbache, “Acknowledgements,” 2019.

[10] D. Filliat, “Robotics Mobile,” 2012.

[11] Mr. D. E. Master, B. Ahmed, and C. Safouane, “Theme Design and Construction of a Mobile Robot based on Arduino Merci,” pp. 2014–2015.

[12] A. Ide, I. D. Environment, and A. Arduino, “Programmer l ’ Arduino – Introduction.”

[13] Mr. Amira, “Option: Mechanical Modelling and Simulation,” 2017.

[14] Mr. Academique, “Study and realization of a control board by Arduino via the Android system,” 2015.

[15] J. Nussey, Arduino.

[16] Sadao kawamura, Mikhail Svinin. eds., Advances in Robot Control: From Everyday physicsto Human-like Movements. Springer, 2006

[17] Slotine JE, Li WP. On the adaptive control of robot manipulators. The Interational Journal of Robotics Research, 1987, 6(3): 49 _ 59

[18] Slotine JE, Li WP. Applied Non Linear Control. Prentice Hall, 1991

[19] V. Lippiello and F. Ruggiero,Cartesian impedance control of a UAVwith a robotic arm. In 10th International IFAC Symposium on RobotControl, Dubrovnik, Croatia, Sep 2012

[20] Kim S. et al. Aerial Manipulation Using a Quadrotor with a Two DOF Robotic Arm, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013) November 3-7, 2013, Tokyo, Japan.

ANNEX

Annex

(2) Program of the controller: chapIII_1ctrl.m

```
function [sys,x0,str,ts] = control_strategy(t,x,u,flag)
switch flag,
case 0,
[sys,x0,str,ts]=mdlInitializeSizes;
case 3,
sys=mdlOutputs(t,x,u);
case {2,4,9}
sys=[];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end

function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumOutputs = 2;
sizes.NumInputs = 10;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [];
functionsys=mdlOutputs(t,x,u)
q1_d=u(1);dq1_d=u(2);ddq1_d=u(3);
q2_d=u(4);dq2_d=u(5);ddq2_d=u(6);
q1=u(7);dq1=u(8);
q2=u(9);dq2=u(10);
dq=[dq1;dq2];
p=[6.7 3.4 3.0 0]; %Practical p
ep=0.95*p; %Estimated p

alfa_p=ep(1);
beta_p=ep(2);
epc_p=ep(3);
eta_p=ep(4);
m1=1;l1=1;
lc1=1/2;I1=1/12;
g=9.8;
e1=m1*l1*lc1-I1-m1*l1^2;
e2=g/l1;
dq_d=[dq1_d,dq2_d]';
ddq_d=[ddq1_d,ddq2_d]';
e=[q1_d-q1,q2_d-q2]';
de=[dq1_d-dq1,dq2_d-dq2]';

Y=[ddqr(1)+e2*cos(q1),ddqr(2)-e2*cos(q1),2*cos(q2)*ddqr(1)+cos(q2)*ddqr(2)
-2*sin(q2)*dq2*dqr(1)-sin(q2)*dq2*dqr(2)+e2*cos(q1+q2),2*sin(q2)*ddqr(1)+
```

```
sin(q2)*ddqr(2)+2*cos(q2)*dq2*dqr(1)+cos(q2)*dq2*dqr(2)+e2*sin(q1+q2);
0,ddqr(1)+ddqr(2),cos(q2)*ddqr(1)+sin(q2)*dq1*dqr(1)+e2*cos(q1+q2),sin(q2)
*ddqr(1)-cos(q2)*dq1*dqr(1)+e2*sin(q1+q2)];
Y_max=abs(Y)+0.10;

M=1;
if M==1
ep_up=abs(p-ep)+0.50; %Upper p-ep
k=Y_max*ep_up';
% tols=[sign(s(1)) 0;0 sign(s(2))]*k+s;
tols=[sats(1) 0;0 sats(2)]*k+s;
tol=H_p*ddqr+C_p*dqr+G_p+tols;
elseif M==2
p_up=abs(p)+0.50; %Upper p value
k_up=Y_max*p_up';
% tol=[sign(s(1)) 0;0 sign(s(2))]*k_up+s;
tol=[sats(1) 0;0 sats(2)]*k_up+s;
end
sys(1)=tol(1);
sys(2)=tol(2);
```

(3) Program of the plant: chap III_1plant.m

```
function [sys,x0,str,ts]=s_function(t,x,u,flag)
switch flag,
case 0,
[sys,x0,str,ts]=mdlInitializeSizes;
case 1,
sys=mdlDerivatives(t,x,u);
case 3,
sys=mdlOutputs(t,x,u);
case {2, 4, 9 }
sys = [];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 4;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 4;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys=simsizes(sizes);
x0=[1.0,0,1.0,0];
str=[];
ts=[];
function sys=mdlDerivatives(t,x,u)
tol=[u(1);u(2)];
q1=x(1);
```

```
dq1=x(2);
q2=x(3);
dq2=x(4);
p=[6.7 3.4 3.0 0];
alfa=p(1);
beta=p(2);
epc=p(3);
eta=p(4);
m1=1;l1=1;
lc1=1/2;I1=1/12;
g=9.8;
e1=m1*l1*lc1-I1-m1*l1^2;
e2=g/l1;
H=[alfa+2*epc*cos(q2)+2*eta*sin(q2),beta+epc*cos(q2)+eta*sin(q2);
beta+epc*cos(q2)+eta*sin(q2),beta];
C=[(-2*epc*sin(q2)+2*eta*cos(q2))*dq2,(-epc*sin(q2)+eta*cos(q2))*dq2;
(epc*sin(q2)-eta*cos(q2))*dq1,0];
G=[epc*e2*cos(q1+q2)+eta*e2*sin(q1+q2)+(alfa-beta+e1)*e2*cos(q1);
epc*e2*cos(q1+q2)+eta*e2*sin(q1+q2)];
%robot dynamic equation as
S=inv(H)*(tol-C*[dq1;dq2]-G);
sys(1)=x(2);
sys(2)=S(1);
sys(3)=x(4);
sys(4)=S(2);
functionsys=mdlOutputs(t,x,u)
sys(1)=x(1);
sys(2)=x(2);
sys(3)=x(3);
sys(4)=x(4);
```

(4) Program of the input command: chap III_1input.m

```
function [sys,x0,str,ts] = input(t,x,u,flag)
switch flag,
case 0,
[sys,x0,str,ts]=mdlInitializeSizes;
case 3,
sys=mdlOutputs(t,x,u);
case {2,4,9}
sys=[];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumOutputs =6;
sizes.NumInputs = 0;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
```

```
x0 = [];
str = [];
ts = [];
functionsys=mdlOutputs(t,x,u)
q1_d=sin(2*pi*t);
q2_d=sin(2*pi*t);
dq1_d=2*pi*cos(2*pi*t);
dq2_d=2*pi*cos(2*pi*t);
ddq1_d=-(2*pi)^2*sin(2*pi*t);
ddq2_d=-(2*pi)^2*sin(2*pi*t);
sys(1)=q1_d;
sys(2)=dq1_d;
sys(3)=ddq1_d;
sys(4)=q2_d;
sys(5)=dq2_d;
sys(6)=ddq2_d;
```

(2) Program of control law: chapIII_2ctrl.m

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
[sys,x0,str,ts]=mdlInitializeSizes;
case 3,
sys=mdlOutputs(t,x,u);
case {2,4,9}
sys=[];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end

function [sys,x0,str,ts]=mdlInitializeSizes

globalnmn
nmn=25*eye(2);
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 2;
sizes.NumInputs = 6;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0 0];
functionsys=mdlOutputs(t,x,u)
globalnmn
qd1=u(1);
dqd1=cos(t);
ddqd1=-sin(t);
```

```
qd2=u(2);
dqd2=cos(t);
ddqd2=-sin(t);
ddqd=[ddqd1;ddqd2];
dqd=[dqd1;dqd2];
ddqd=[ddqd1;ddqd2];
q1=u(3);dq1=u(4);
q2=u(5);dq2=u(6);
dq=[dq1;dq2];
e1=qd1-q1;
e2=qd2-q2;
e=[e1;e2];
de1=dqd1-dq1;
de2=dqd2-dq2;
de=[de1;de2];
alfa=6.7;beta=3.4;
epc=3.0;eta=0;
m1=1;l1=1;
lc1=1/2;I1=1/12;
g=9.8;
e1=m1*l1*lc1-I1-m1*l1^2;
e2=g/l1;
H=[alfa+2*epc*cos(q2)+2*eta*sin(q2),beta+epc*cos(q2)+eta*sin(q2);
beta+epc*cos(q2)+eta*sin(q2),beta];
C=[(-2*epc*sin(q2)+2*eta*cos(q2))*dq2,(-epc*sin(q2)+eta*cos(q2))*dq2;
(epc*sin(q2)-eta*cos(q2))*dq1,0];
G=[epc*e2*cos(q1+q2)+eta*e2*sin(q1+q2)+(alfa-beta+e1)*e2*cos(q1);
epc*e2*cos(q1+q2)+eta*e2*sin(q1+q2)];
H0=0.6*H;
C0=0.6*C;
G0=0.6*G;
316
s=de+nmn*e;
d_up=30;
xite=0.10;
M=2;
if M==1
d=(d_up+xite)*sign(s);
elseif M==2 %Saturated function
delta=0.05;
kk=1/delta;
if abs(s)>delta
sats=sign(s);
else
sats=kk*s;
end
end
d=(d_up+xite)*sats;
v=ddqd+nmn*de+d;
tol=H0*v+C0*dq+G0;
```

```
sys(1)=tol(1);
sys(2)=tol(2);
```

(3) Program of the plant: chap III_2plant.m

```
function [sys,x0,str,ts]=s_function(t,x,u,flag)
switch flag,
case 0,
[sys,x0,str,ts]=mdlInitializeSizes;
case 1,
sys=mdlDerivatives(t,x,u);
case 3,
sys=mdlOutputs(t,x,u);
case {2, 4, 9 }
sys = [];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 4;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 4;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys=simsizes(sizes);
x0=[0;0;0;0];
str=[];
ts=[];
function sys=mdlDerivatives(t,x,u)
q1=x(1);dq1=x(2);
```



```
q2=x(3);dq2=x(4);
dq=[dq1;dq2];
% The model is given by Slotine and WeipingLi(MIT 1987)
alfa=6.7;beta=3.4;
epc=3.0;eta=0;
m1=1;l1=1;
lc1=1/2;I1=1/12;
g=9.8;
e1=m1*l1*lc1-I1-m1*l1^2;
e2=g/l1;
H=[alfa+2*epc*cos(q2)+2*eta*sin(q2),beta+epc*cos(q2)+eta*sin(q2);
beta+epc*cos(q2)+eta*sin(q2),beta];
C=[(-2*epc*sin(q2)+2*eta*cos(q2))*dq2,(-epc*sin(q2)+eta*cos(q2))*dq2;
(epc*sin(q2)-eta*cos(q2))*dq1,0];
G=[epc*e2*cos(q1+q2)+eta*e2*sin(q1+q2)+(alfa-beta+e1)*e2*cos(q1);
epc*e2*cos(q1+q2)+eta*e2*sin(q1+q2)];
tol(1)=u(1);
```

```
tol(2)=u(2);
ddq=inv(H)*(tol'-C*dq-G);
sys(1)=x(2);
sys(2)=ddq(1);
sys(3)=x(4);
sys(4)=ddq(2);
functionsys=mdlOutputs(t,x,u)
sys(1)=x(1);
sys(2)=x(2);
sys(3)=x(3);
sys(4)=x(4);
```