



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of El-Oued



Faculty of Science and Technology
Second Year Master's

MEMORANDUM ENTITLED

Performance study of a four-pole cable robot

Prepared by the student

***Haroun Abdeljabar**

***Maamra Salah**

Supervisor

***Aroudj Mohammed**

Academic year: 2025/2026

الإهداء

إلى من كان لهما الفضل الكبير في حياتي،
إلى من سهر الليالي وضحيًا من أجلي،
إلى أعز الناس على قلبي... والدَيَّ الكريمين،
أهدي هذا العمل المتواضع، راجيًا من الله أن يحفظهما ويديمهما تاجًا على رأسي
إلى كل من دعمني وساندني من قريب أو بعيد،
أهدي ثمرة جهدي

الشكر والتقدير

الحمد لله الذي وفقنا لإتمام هذا العمل

نتقدم بجزيل الشكر وعظيم الامتنان إلى الأستاذ عروج محمد على إشرافه القيّم، وتوجيهاته السديدة، ونصائحه التي كان لها الأثر الكبير في إنجاز هذه المذكرة

كما نتقدم بخالص الشكر والتقدير إلى لجنة المناقشة المحترمة على تفضلهم بقبول مناقشة هذا العمل، وعلى ملاحظاتهم القيمة التي ستساهم في تحسينه

ولا يفوتنا أن نتقدم بجزيل الشكر والعرفان إلى والدينا الكريمين على دعمهما المتواصل وتشجيعهما الدائم، فلهما كل الفضل بعد الله فيما وصلنا إليه

كما نشكر كل من ساهم من قريب أو بعيد في إنجاز هذا العمل

ملخص:

تركز هذه الدراسة على أداء روبوت متوازي مستوي رباعي الأقطاب، يعمل بنظام الكابلات، ومصمم لإعادة تأهيل الأطراف العلوية والمساعدة في تمارين الكتابة. يتميز هذا الروبوت بمساحة عمل واسعة، وهيكـل خفيف الوزن ومرن، ومستوى أمان معزز في حالة حدوث عطل.

يتضمن العمل المقدم في "دراسة أداء روبوت الكابلات رباعي الأقطاب" نمذجة حركية وديناميكية للروبوت لتحليل سلوكه وتحسين شد الكابلات وعزم دوران المشغلات. تُظهر الدراسة أن هذا التحسين يُحسن استقرار النظام، ويمنع ارتخاء الكابلات، ويضمن دقة حركة أكبر.

كما تم استخدام استراتيجية تحكم بالنمط الانزلاقي ومقارنتها بالأساليب التقليدية مثل PID. تُظهر النتائج المتحصل عليها تحسناً في زمن الاستجابة، والمتانة، ودقة الروبوت أثناء تمارين إعادة التأهيل والكتابة.

الكلمات المفتاحية: الروبوت متواز، الروبوت المتوازي مع أربعة كابلات، النمذجة، التحكم الأمثل، إعادة تأهيل الأطراف العلوية.

Abstract:

This study focuses on the performance of a four-pole, cable-operated parallelogram robot designed for upper limb rehabilitation and typing exercises. The robot features a large working area, a lightweight and flexible structure, and enhanced safety in case of failure.

The work presented in the "Performance Study of the Four-Pole Cable Robot" includes kinematic and dynamic modeling of the robot to analyze its behavior and optimize cable tension and actuator torque. The study demonstrates that this optimization improves system stability, prevents cable slack, and ensures greater motion accuracy.

A glide-type control strategy was also employed and compared with traditional methods such as PID. The results show improvements in response time, robustness, and accuracy of the robot during rehabilitation and typing exercises.

Keywords: parallel robot, parallel robot with four cables, modeling, optimization control, rehabilitation of the upper limbs.

Table of Contents

List of figures

GENERAL INTRODUCTION

Chapter 1 : GENERAL INFORMATION ON CABLE PARALLEL ROBOTS

1.1Description.....	1
1.2Classification.....	1
1.3 Advantage Disadvantages.....	2
1.3.1Advantages.....	2
1.3.2Disadvantages.....	3
1.4Applications.....	3
1.4.1Skycam.....	3
1.4.2IPanema.....	4
1.4.3 Pyramidal cable driven robot for child writing assistance.....	4
1.4.4 The Adiutor upper limb device.....	5
1.4.5 CaTraSys.....	6
1.4.6 STRING-MAN.....	7
1.4.7 NeReBot.....	7
1.5 Some Issues and State of the Art.....	8
1.5.1 Analysis, Design, and Workspa.....	8
1.5.2 Modeling and Cable Effects Modeling.....	9
1.5.3 Control and Tension Distribution.....	9
1.6 Conclusion.....	10

Chapter 2: Modeling of the 4-Cable Planar Robot

2.1Introduction.....	12
2.2 The Procedure for Design.....	12
2.3 Interface to Control the System.....	14
2.3.1 Graphical user interface GUI.....	14
2.3.2 Strategy the Control Used.....	15
2.4 Geometric Modeling.....	15
2.4.1 Inverse Geometric Modeling.....	16
2.4.2 Direct Geometric Model.....	16
2.5 Kinematic Modeling.....	17
2.5.1 Inverse Kinematic Model.....	17
2.5.2 Direct Kinematic Model.....	18
2.6 Dynamic Model.....	19
2.6.1 The Lagrange Method.....	19
2.6.1.1 Calculation of the angular velocities and accelerations of the pulleys.....	20
2.6.1.2 Kinetic Energy of the System.....	20
2.6.1.3 The potential energy of the system.....	21
2.6.1.4 System Dissipation Energy.....	21
2.6.1.5 Calculation of Derivatives.....	21
2.6.1.6 Calculation of the Qd coefficients.....	22

2.6.1.7 Lagrange Equations in Matrix Form.....	23
2.6.2 Newton-Euler Method.....	24
2.6.2.1 Dynamic Model of the Effector.....	24
2.6.2.2 Mechanical Structure of Motors.....	24
2.6.2.3 Dynamic Model of the System.....	25
2.7 Conclusion.....	27
Chapter 3: Robot Control and Simulation	
3.1 Introduction.....	29
3.2 Simulation with MSC-ADAMS.....	29
3.3 Procedure de simulation Matlab-ADAMS.....	30
3.4 Validation and Simulation of the Robot Geometric Model.....	31
3.4.1 Simulation of the Inverse Geometric Model.....	31
3.4.2 Simulation of the End-Method Trajectory.....	34
3.4.2.1 A. Variation of cable lengths.....	35
3.4.2.2 B. Variation of cable angles with respect to the x-axis.....	36
3.5 Test continuous trajectories.....	37
3.6 Conclusion.....	40
3.7 OPEN LOOP COMMAND.....	41
3.7.1 Introduction.....	41
3.7.2 The State-Space Representation of the System.....	41
3.7.3 Simulation of the Robot's Response.....	43
3.7.3.1 Step Response.....	43
3.7.3.2 Impulse Response.....	43
3.7.3.3 Sinusoidal Response.....	44
3.7.4 Static Force Analysis.....	44
3.7.5 Calculation of Optimized Actuator Torques.....	49
3.7.6 Implementation of the torque optimization algorithm in MATLAB.....	51
3.7.6.1 Application of the optimization algorithm to the robot.....	52
3.7.7 Generating a trajectory for the robot in open loop.....	54
3.7.8 Conclusion.....	57
3.8 Closed-loop control.....	58
3.8.1 Introduction.....	58
3.8.2 Sliding Mode Control Approach.....	58
3.8.2.1 Introduction to Sliding Mode.....	58
3.8.2.2 Basic Principle of Sliding Mode Control.....	58
3.8.2.3 Structure of Sliding Mode Control.....	59
3.8.3 Sliding Control of the 4-Cable Robot.....	60
3.8.3.1 Establishing the Sliding Control Law.....	60
3.8.3.2 Simulation of the control with constraints in sliding mode for the 4-cable robot.....	63
3.8.3.2.1 Example of a 4-cable robot simulation.....	64
3.8.4 Conclusion.....	68

Chapter 4: PRACTICAL SECTION

4.1 Introduction.....	70
4.2 Hardware Section.....	70
4.2.1 The Base.....	70
4.2.2 Pulley and Guides.....	71
4.2.3 The Effector.....	71
4.2.4 Cables.....	72
4.2.4.1 Massless, Indeformable Cable.....	72
4.2.4.2 Massless Elastic Cable.....	73
4.2.4.3 Elastic Cable with Non-Zero Mass.....	74
4.2.5 Arduino Control Board.....	74
4.2.6 Motor.....	75
4.2.7 Driver Circuit.....	76
4.3 Software Section.....	77
4.3.1 The Arduino IDE Interfac.....	77
4.3.2 Position Control of Motors.....	78
4.3.3 Program Implementation.....	78
4.4 Problems Encountered During the Practical Session.....	79
4.4.1 Motors.....	79
4.4.2 Cables.....	79
4.4.3 Structure.....	79
4.4.4 The Pulley.....	80
4.5 Upper Limb Rehabilitation Using a Four-Cable Planar Robot.....	80
4.5.1 Upper Limb Movements.....	81
4.5.2 Pre-programmed Mode (Passive Assistant Mode).....	82
4.5.3 Active Assistant Mode.....	82
4.5.4 Progressive Resistance Mode.....	83
4.5.5 Virtual Rehabilitation Therapy Game.....	84
4.6 Conclusion.....	84
General Conclusion.....	85
BIBLIOGRAPHY.....	87
Appendix 1.....	90
Appendix 2.....	91
Appendix 3.....	94

Table of Figures

Figure 1.1: Components of a parallel robot operating with 4 cables.....	1
Figure 1.2: Camera controlled by a cable mechanism with an example of positioning in a stadium.....	3
Figure 1.3 :IPA-nema, 6DOF parallel mechanism driven by cables	4
Figure 1.4: Pyramidal cable driven robot.....	5
Figure 1.5: Adiator home assitive device.....	6
Figure 1.6 : Representation of a force sensor for cable robots (CaTraSys) 2.....	6
Figure 1.7 :The mechanical prototype of STRING-MAN 6.....	7
Figure 1.8: NeReBot is a cable-based robot for upper limb rehabilitation with 3 degrees of freedom	7
Figure 2.1: Mechanical structure of the 4-cable planar robot.....	12
Figure 2.2: Proposed design cable-driven planar robot using solidworks software.....	13
Figure 2.3 :The motors and pulleys in our system.....	13
Figure 2.4 Upper part of the graphical control interface.....	14
Figure 2.5 :Lower part of the graphical control interface.....	14
Figure 2.6 :Open loop control architecture for the robot.....	15
Figure 2.7 :Geometric model of the four-cable robot.....	16
Figure 2.8 : Change of reference frame for calculating the kinematic model.....	17
Figure 2.9: Free-body diagram for the <i>ime</i> pulley / shaft	24
Figure 3.1 : MSC-ADAMS Graphical Interface.....	29
Figure 3.2 : Virtual model of the 4-cable robot under MSC-ADAMS.....	30
Figure 3.3 : Simulink for ADAMS.....	31
Figure 3.4 : Flowchart of the lengths and angles of the 4 robot cables.....	32
Figure 3.5 : Terminal organ in initial position.....	32
Figure 3.6 : Terminal organ in position (14.7).....	33
Figure 3.7 : Terminal organ in position (10,-17).....	33
Figure 3.8 : Flowchart tracing the 4-cable geometry with the circular trajectory.....	34
Figure 3.9 : circular trajectory of radius $R = \frac{R_{max}}{2}$	35
Figure 3.10 : Flowchart showing the variations in cable lengths and anglesand writing.....	35
Figure 3.11 :Variations in the lengths of the 4 robot cables for $R = R_{max}$	36
Figure 3.12 : Variations in cable lengths in polar form.....	36
Figure 3.13 :Variations in cable angles.....	37
Figure 3.14:(a) drawing a triangle in LabView. (b) output valuesafter processing (c) The cable lengths(L1, ...L5).....	37
Figure 3.15:(a) drawing a square in LabView. (b) output values (c) The cable lengths (L1,.,L4).....	37
Figure 3.16:(a) writing a letter 'A' in LabView. (b) output valuesafter processing (c) The cables lengths (L1, ...L5).....	39
Figure 3.17:(a) writing a letter 'E' in LabView. (b) output valuesafter processing (c) The cables lengths (L1, ...L5).....	39
Figure 3.7.1: Step response of the robot, Profile of $x(m)$	43
Figure 3.7.2: Step response of the robot, Profile of $y(m)$	43

Table of Figures

Figure 3.7.3: Robot impulse response, $x(m)$ profile.....	44
Figure 3.7.4: Robot impulse response, $y(m)$ profile.....	44
Figure 3.7.5: Sinusoidal response of the robot, $x(m)$ profile.....	44
Figure 3.7.6: Sinusoidal response of the robot, $y(m)$ profile.....	44
Figure 3.7.7: Static force diagram	45
Figure 3.7.8: Sectors of the workspace of the 4-cable robot	47
Figure 3.7.9: Flowchart for calculating optimal pairs.....	51
Figure 3.7.10: Minimum torques of the actuators.....	53
Figure 3.7.11: Optimal actuator torques.....	53
Figure 3.7.12: Optimal cable tension.....	53
Figure 3.7.13: Virtual generalized Cartesian forces.....	53
Figure 3.7.14: Flowchart of the simulation of the planar robot with 4 cables in BO.....	55
Figure 3.7.15: Response of the robot with 4 cables to a circle for 1s.....	56
Figure 3.7.16: Response of the robot with 4 cables to a circle for 5 seconds.....	56
Figure 3.7.17: Static error for the circle during 1 s.....	56
Figure 3.7.18: Static error for the circle during 5 s.....	56
Figure 3.7.19: Response of the 4-cabl robot to a Lissajous curve for 1 s.....	57
Figure 3.7.20: Response of the 4-cable robot to a Lissajous curve for 5 s.....	57
Figure 3.8.1 : Different modes for the trajectory in the phase plane	59
Figure 3.8.2 : Representation of the SMC control law.....	63
Figure 3.8.3 : The implementation diagram corresponding to the closed-loop control expressions by SMC according to x and y	64
Figure 3.8.4 : Profile of the (x) position by sliding mode.....	65
Figure 3.8.5 : The profile of the position (y) by the sliding mode.....	65
Figure 3.8.6 : The velocity (V_x) profile by the sliding mode.....	66
Figure 3.8.7 : Le profil de la vitesse (V_y) par le mode glissant.....	66
Figure 3.8.8 : Trajectory of the system represented in phase plane along (x)	67
Figure 3.8.9 : Trajectory of the system represented in phase plane along (y)	67
Figure 3.8.10 : The trajectory followed by the terminal effector.....	68
Figure 4.1 : the base.....	70
Figure 4.2: Pulleys and guides.....	71
Figure 4.3: The effector.....	71
Figure 4.4: Cable modeling.....	72
Figure 4.5: Non-deformable cable of zero mass.....	73
Figure 4.6: Zero-mass elastic cable.....	73
Figure 4.7: Elastic cable with non-zero mass.....	74
Figure 4.8: Arduino MEGA 2560 board.....	75
Figure 4.9: DC motor.....	76
Figure 4.10 : Circuit L298N.....	77
Figure 4.11: The Arduino IDE programming environment.....	77
Figure 4.12: Motor position control.....	78
Figure 4.13: Flowchart of the robot control program.....	78
Figure 4.14: Stepper motor.....	79
Figure 4.15: Proposed structure.....	80

Table of Figures

Figure 4.16: The pulley proposed for this type of robot.....	80
Figure 4.17: The application of the robot for upper limb rehabilitation.....	81
Figure 4.18: Movements of the upper limbs	81
Figure 4.19: Passive assistant mode	82
Figure 4.20: Active assistant mode	83
Figure 4.21: Mode of progressive resistance	83
Figure 4.22: Billiards game for virtual rehabilitation through the robot	84

General Introduction

General Introduction:

Robotics encompasses all the technologies used in the design and construction of automated machines or robots. A robot is a mechatronic device (combining mechanics, electronics, and computer science) designed to perform tasks that mimic or reproduce human actions within a defined domain. The design of these systems is the subject of a scientific discipline, a branch of automation called robotics.

Robots have numerous applications. They have been integrated into various industries, enabling them to perform repetitive tasks with extreme precision. Thanks to technological advancements, robots are now used in advanced sectors such as aerospace, aviation, medicine, and the military, and in recent years, even in homes.

Cable robots (also called cable-suspended or wire robots) are a type of parallel robot that uses flexible cables as joints. One end of each cable is wound around a rotor operated by a servo, while the other end is connected to a terminal device. These cables are significantly lighter than the rigid joints found in series or parallel robots. The dynamic analysis of cable robots differs from that of other parallel robots because the cables can only pull the terminal tool, not push it. Therefore, a cable robot can only perform a task if the force across all the cables is non-negative. Redundancy plays a crucial role in the design of cable robots; to ensure the robot follows predetermined paths, the cables must remain taut.

Within the framework of ongoing work on a four-cable planar robot in the Process Control Laboratory, this work aims to study and develop a prototype of this robot. This study includes modeling, optimizing, and controlling the four-cable planar robot. This prototype is intended to assist specialists in upper limb rehabilitation.

This thesis is organized into four chapters.

Chapter 1: contains general information about robotics and the characteristics and application areas of cable-operated parallel robots.

In Chapter 2: we present a model of a four-cable planar robot. This model incorporates three types: direct and inverted geometric models, direct and inverted kinematic models, and a dynamic model. In studying the final model...

We used two methods: the Lagrange method and the Newton-Euler method.

Chapter 3: is dedicated to studying the open loop by creating a state representation, examining the different responses of the four-cable planar robot, and studying the optimization of the actuator torque using . It also covers the closed loop using slip patterns and the validation of the dynamic model using MATLAB.

Chapter 4: The final chapter is dedicated to building a prototype of the four-cable planar robot to attempt to apply the theoretical studies begun in the previous chapters, followed by the initial use of the robot in the functional rehabilitation of the upper limbs

Chapter : 1

GENERAL INFORMATION ON CABLE PARALLEL ROBOTS

1.1 Description

A parallel cable robot is composed of a generally fixed base (or structure) and a mobile actuator (or platform or nacelle in parallel robotics) connected in parallel by cables. The movement or effort is imposed on the actuator by driving the cables through actuators generally composed of rotary motors coupled to rollers (or drums) intended to store the cables, and often equipped with return poles for guiding the cables, allowing the same quantity of cables to be wound, thus avoiding their tangling (Figure 1.1). Other action techniques have been proposed, using, for example, linear motors coupled to rollers [1].

Actuators can be remote on the base, or more rarely mounted on the end effector. In the classic case of remote actuation, where the actuators and their possible cable return pulleys are fixed to the base, each cable is attached at one end to the end effector and wound at the other end via an actuator through its return pulley. The point of contact between the cable and the platform is called the attachment point, and the point of contact between the cable and the return pulley is called the exit point (Figure 1.1).

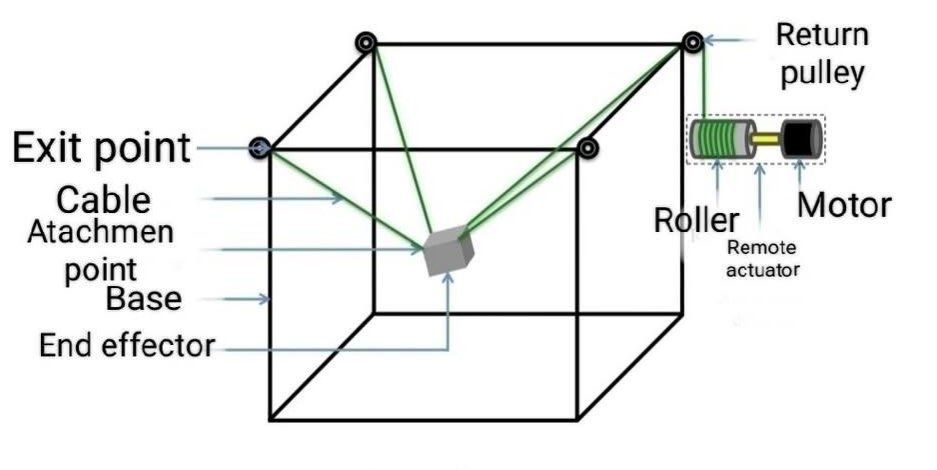


Figure 1.1 Components of a parallel robot operating with 4 cables

1.2 Classification

For a given geometric shape of the base and end effector, the geometry of a cable-driven parallel robot is defined by the positions of the attachment and exit points. The number of cables n and their arrangement allows us to deduce the number of degrees of freedom of the end effector m , and therefore the degree of actuation redundancy $n-m$. Depending on their geometry and the presence or absence of actuation redundancy, cable-driven parallel robots are found in different configurations, giving rise to specific classifications that characterize their control technology.

A condition for controlling all m degrees of freedom of the effector is to use at least $m+1$ cables. A fundamental classification then depends on the degree of actuation redundancy and defines three types of configurations:

Incompletely constrained configuration : that uses nm cables to control the m degrees of freedom of the effector.

Fully constrained configuration : that uses $nm+1$ cables to control the m degrees of freedom of the effector.

Redundantly constrained configuration: that uses $nm+1$ cables to control the m degrees of freedom of the effector.

The fully constrained and redundantly constrained configurations, which concern redundant robots, can be grouped into the same class. In such configurations, the end effector pose is completely defined from the cable lengths (geometric problem). On the other hand, even if the incompletely constrained configuration, which concerns non-redundant robots, was initially defined with $n=m$ cables, it is possible to achieve such a configuration with $n < m$ cables. In such a configuration, the end effector pose is not completely defined from the cable lengths and requires additional forces (e.g., gravity) to determine the resulting pose (geometric-static problem).

The stated mechanical controllability condition is necessary but not sufficient, since it does not take into account the geometry of the robots and the cable arrangement. Thus, this fundamental classification can be encapsulated in the following classification, which defines two types of configurations.

Fully constrained configuration where the robot is capable of generating forces in all directions. This configuration requires more cables than degrees of freedom of the end effector and therefore includes fully constrained and redundantly constrained robots that manage to generate arbitrary forces.

1.3 Advantages and Disadvantages

1.3.1 Advantages

The use of cables for actuation in cable-driven parallel robots offers several significant advantages over rigid parallel robots, including.

- reduced risk of interference due to the small size of the cables.
- a very large workspace, since in theory it depends only on the maximum length of the cables used[1].
- the possibility of relocating the actuators outside the workspace[2].
- high load capacity[3].
- Transparency of the cables, which do not obstruct the field of vision of the workspace[3].

- adaptability to different workspace scales[4].
- configurability, since the positions of the attachment and exit points can be quickly changed[5].
- very high accelerations thanks to the low mass and inertia of the moving parts[6].
- portability, and ease of assembly and disassembly for operations at different sites.
- Ease and low cost of design and maintenance.

1.3.2 Disadvantages

Despite all these interesting properties, cable-driven parallel robots have certain disadvantages closely linked to the unilateral (or unidirectional) and flexible behavior of the cables. Indeed.

- The cables can only produce tensile forces (positive tensions) in the direction of the cable, and therefore can only exert force on the end effector (and not push it).
- The cables can deform according to a law dependent on the characteristics of the materials they are made of, under the effect of external forces applied to them by intrinsic elements of the structure, such as the end effector and actuators, or extrinsic forces, such as gravity[7].

1.4 Applications

1.4.1 Skycam

Thanks to their numerous advantages, cable-driven parallel robots are currently being considered for a variety of applications, both in academic research and industry. While the first research dates back to the suspended robot presented in Landsberger , the first commercial prototype was the Skycam Cone suspended robot [8], designed to move a video camera in sports stadiums.

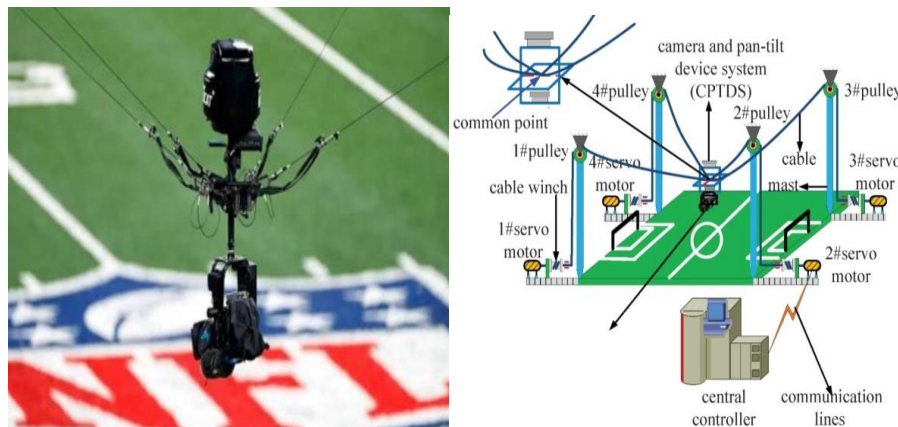


Figure 1.2: Camera controlled by a cable mechanism with an example of positioning in a stadium .

1.4.2 IPAnema

the group at Fraunhofer IPA has been developing a family of robots called IPA-nema. These include energy-efficient systems and equipment for manufacturing products or installations such as solar power plants or wind turbines . The components, which until now have been large and/or heavy, are mostly handled using conventional cranes [8]. The robot is a 6DOF parallel mechanism driven by cables (Figure 1.3). One of the main objectives of the ongoing research projects is the development of an industrial-grade cable robot with high robustness and precision.

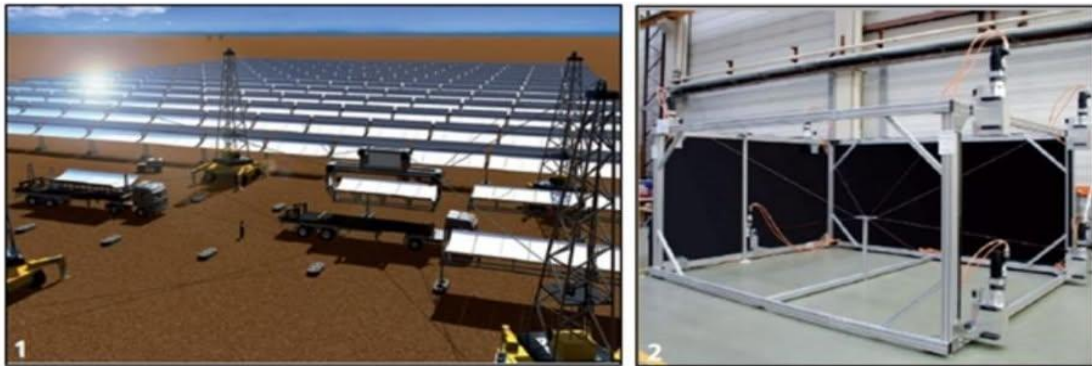


Figure 1.3 :IPA-nema, 6DOF parallel mechanism driven by cables

1.4.3 Pyramidal cable driven robot for child writing assistance

An innovative pyramidal cable driven robot has been designed to assist primary school children with difficulties in learning or regaining writing skills, as shown in Figure 1.4.

The robot's design features a pyramid-shaped structure with cables attached to the top point, which control a writing instrument held at the base. The child interacts with the robot by moving a stylus or using a modified pen grip. As the child moves the stylus, the robot translates those movements into precise and controlled writing actions.

This technology aims to provide personalised and adaptive support to children with fine motor difficulties, helping them develop and recover writing skills in a fun and engaging way. Personalisation is key as the system selects specific tasks for the child, which are then translated into executable trajectories by the robot, thereby tailoring the writing experience to their individual needs and therapeutic goals.

Through this assistive robotic solution, children can experience a positive and rewarding writing learning experience, enhancing their overall educational journey[9].



Figure 1.4: Pyramidal cable driven robot: (a) concept; (b) prototype at UNICAL.

1.4.4 The Adiutor upper limb device

The Adiutor upper limb rehabilitation device is an advanced assistive mechatronic solution designed for home-based exercising and rehabilitation of the upper limb, Figure 1.5. Based on a delta architecture, it is designed to offer a wide range of rehabilitation modalities, supporting functional recovery in a targeted and effective manner.

The Adiutor device not only allows users to perform a series of pre-programmed exercises, but also offers the possibility to record and replicate specific movements. These movements can be created and set by the therapist, who can customise the exercises to the patient's individual needs. After recording movement, the device allows the patient to repeat it, providing a score based on execution time and comparison with a healthy person's time. This scoring system helps monitor the patient's progress and motivate them, making the rehabilitation experience more engaging and goal-oriented.

In addition, the Adiutor device features an intuitive interface that facilitates patient interaction with the system and increases engagement during rehabilitation sessions. The integrated telemonitoring functionality allows health professionals to remotely follow and assess patients' progress, thus ensuring personalised care and effective remote support.

This integrated approach, combining user interface and telemonitoring, makes the Adiutor a valuable tool for efficient upper limb rehabilitation, all from the comfort of one's own home[10]



Figure 1.5: Adiutor home assistive device: a) concept; b) prototype at UNICAL.

1.4.5 CaTraSys

Another area of interest in biomedical applications is motion tracking. The example of CaTraSys (Cassino Tracking System) has been used for the identification of kinematic parameters and human mobility (2). CaTraSys is a measurement system that was designed and built at LARM (Laboratory of Robotics and Mechatronics) in Cassino, Italy[11]. It has been used to determine the position of the extremities of limbs during their movement and, in addition, it can measure the forces and torques exerted by the limb, as shown in Figure 1.6.

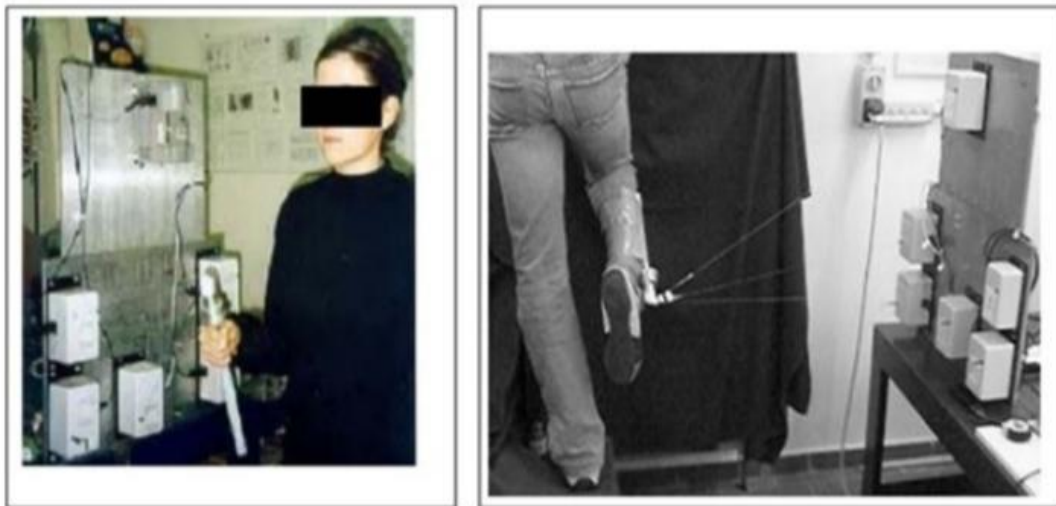


Figure 1.6 : Representation of a force sensor for cable robots (CaTraSys) 2

1.4.6 STRING-MAN

Cable system for rehabilitation of the jaw tube. This system is called STRING-MAN. It is a cable manipulator that supports the patient during therapy for recovery. Its design was inspired by the principle of puppet strings. The mechanical structure, in Figure (1.7), has a configuration with ten cables. There is a conveyor belt that allows the patient's movement. The human body constitutes the end organ of the robot[12].



FIGURE 1.7 :The mechanical prototype of STRING-MAN

1.4.7 NeReBot

NeReBot (Neuro Rehabilitation roBot) is a cable-based robot developed by the robotics laboratory of the Department of Mechanical Innovation and Management at the University of (PADUA) Bali. It has three degrees of freedom and was designed for treatment and rehabilitation for post-shock[13].

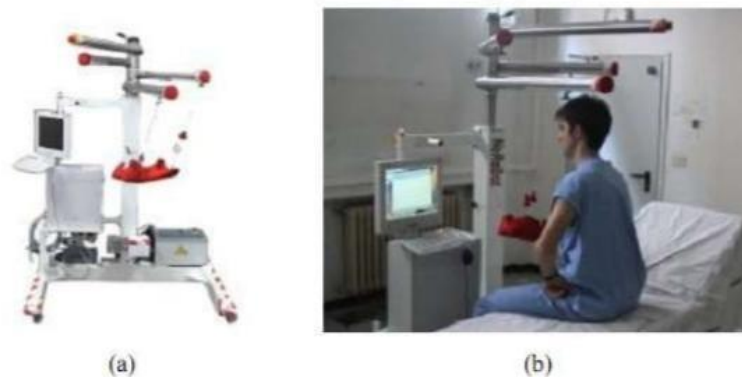


FIGURE 1.8: NeReBot is a cable-based robot for upper limb rehabilitation with 3 degrees of freedom a) the mechanical structure b) the system during clinical trials

The manipulator's mechanical structure consists of a central column and four aluminum connecting tubes, as shown in Figure 1.8. Three plastic cables are linked to a splint in which the patient's limb is positioned. Each cable is driven by a disconnected motor positioned on the base of the structure. The robot follows a spatial trajectory of the splint by pulling the cables.

The mechanical structure is adjusted by the therapist according to the specific needs of the patient and the therapy. This is a complex task requiring experience and technical skill. The angular position of each link and the linear position of the cable entry points through the links must be established.

The therapy consists of repetitive, passive, high-performance movements of the patient's upper limbs. There is a learning phase during which the therapist moves the limbs through a trajectory and saves the angular positions of each motor

Control is achieved using software that implements the entire motion control (a proportional-integral-derivative PID controller to separately control each motor) and a graphical user interface (Human-Machine Interaction).

1.5 Some Issues and State of the Art

The major drawbacks mentioned for cable-driven parallel robots make their study even more complex than in the case of rigid parallel robots. These limitations and technical particularities can indeed introduce problems in certain aspects described below, to mention only those directly related to our work:

1.5.1 Analysis, Design, and Workspace

The design phase of cable-parallel robots is more complex than that of traditional rigid robots. The primary objective is to determine the optimal geometry to ensure specific characteristics such as.

Accessible Workspace:

This is the area in which the effector can move.

Employable-Mode Workspace (EPP):

Also known as a wrench closure workspace, this is the set of modes in which any voltage can be generated on the effector due to unlimited positive cable tensions[14].

Educable-Generating-Tensile (ETG) Workspace:

This is the set of modes in which a specific range of voltages can be generated with limited cable tensions.

Optimization:

Mathematical criteria are used to determine the ideal dimensions of the robot, taking into account cable interference and collision avoidance.

1.5.2 Modeling and Cable Effects Modeling

is often simplified by considering cables as rigid, straight lines. However, reality presents challenges due to the physical properties of cables:

Cable Mass:

The mass of a cable causes sagging, changing its shape from a straight line to a curve. This directly affects the accuracy of responder positioning, especially over long distances.

Elasticity:

Cables under tension stretch, creating a duality between the geometric and static models. Cables are sometimes modeled as virtual springs to study these effects. Vibrational Stability:

The stiffness of a mechanism is related to cable tension. Low tensions can cause unwanted vibrations, requiring careful study to determine the optimal stiffness to minimize this phenomenon.

1.5.3 Control and Tension Distribution

Controlling these robots presents a very delicate problem due to the nature of the cables, which can only "pull" (cannot push):

The challenge of positive tensions:

The control system must ensure that tensions remain within certain limits (greater than zero to avoid slack, and less than the breaking point).

Fully Constrained Mechanisms:

These rely on redundancy to distribute tensions. Linear or quadratic optimization algorithms are used to distribute forces between the cables in a way that minimizes energy consumption or ensures continuous motion.

Suspended robots:

These rely partly on gravity to maintain cable tension. In this case, fewer cables can be used, but control becomes more difficult because weight is the primary factor in maintaining balance. Real-time implementation problems: Complex computational methods that take into account the

flexibility and mass of cables are often too slow for real-time application, forcing researchers to use simplified models or external sensors (such as cameras or lasers) to correct errors.

1.6 Conclusion

Robotics in general, and parallel manipulators in particular, have experienced significant growth in recent years, especially in industry where their use has become widespread. An increasing number of tasks are being assigned to cable-driven parallel robots, which are becoming ever more complex. Cable-driven robotic manipulators are used in a multitude of applications, ranging from welding and handling to machining, and even in the biomedical field. It is clear that each task requires a specific type of manipulator depending on the constraints to be met.

This chapter has discussed cable-driven parallel robots, their advantages and disadvantages, followed by some applications of these robots.

Chapter 2:

Modeling of the 4-Cable Planar Robot

2.1 Introduction

This chapter will present the modeling of a four-cable planar parallel robot, as modeling is the first step in establishing the robot's control. We will represent the forward and inverse geometric, kinematic, and dynamic models of the four-cable robot, enabling planar motion (2 degrees of freedom).

The mechanical structure of this robot consists of:

- A fixed, square base (frame); each cable is connected to the base by an actuator (motor/pulley).
- A mobile platform, which carries the end effector.
- The base and the mobile platform are connected by four cables.
- Each cable is connected to the base by an actuator (motor/pulley). Figure 2.1 illustrates the robot's mechanical structure.

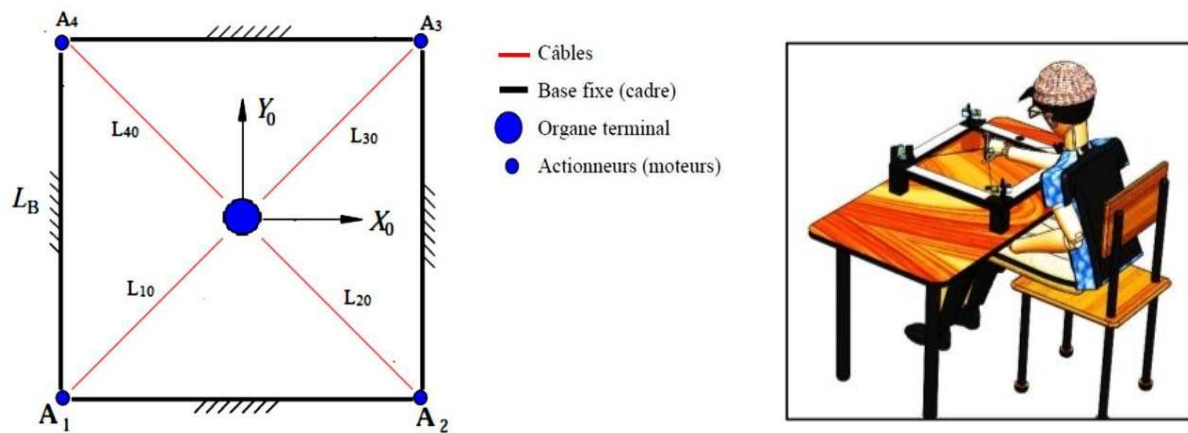


Figure 2.1: Mechanical structure of the 4-cable planar robot.

This model represents the transformations (or relationships) between the operational space (in which the position of the end effector is defined) and the joint space (in which the configuration of the robot's joints is defined).

To control or simulate the behavior of an articulated mechanical system (robot), a model is required. Several levels of modeling are possible depending on the objectives, task constraints, and desired performance.

2.2 The Procedure for Design

In this section, we meticulously crafted a comprehensive cable-parallel planar robot model using the advanced design and simulation capabilities of the Solidworks software. The primary aim of this undertaking was to provide a highly detailed and realistic visualization of the robot's

intricate structure and operational mechanisms. By harnessing the versatility of Solidworks, every component of the robot was painstakingly developed to adhere to precise specifications and engineering standards. This rigorous design process facilitated a thorough comprehension of the robot's intricacies, encompassing its kinematics, dynamics, and structural integrity. Additionally, Solidworks enabled us to conduct rigorous simulations, allowing us to validate the model's behavior under various conditions, pinpoint potential design optimizations, and fine-tune the robot's performance for improved efficiency and reliability. The software's visualization capabilities further enriched the project by furnishing stakeholders with vivid and insightful representations, thus making it more accessible for them to grasp the robot's intricate design and operational intricacies. Consequently, this effort promoted a deeper appreciation for the robot's potential applications and contributions within the field of robotics.

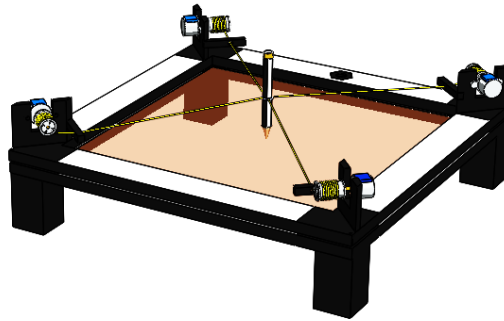


Figure 2.2 Proposed design cable-driven planar robot using solidworks software

Figure 2.2 illustrates a pivotal aspect of our system design, featuring a sturdy and dependable base as its foundation. Atop this stable base, we have strategically installed four motors, positioning them at the ends of the platform. These motors play a vital role in the operation of the system, as each one is specifically tasked with the responsibility of rotating a pulley.

The purpose of this rotation is to intricately control the winding and unwinding of the cable around the pulley mechanism, thereby dictating the precise movements of the end effector. For a visual representation of this mechanism, please

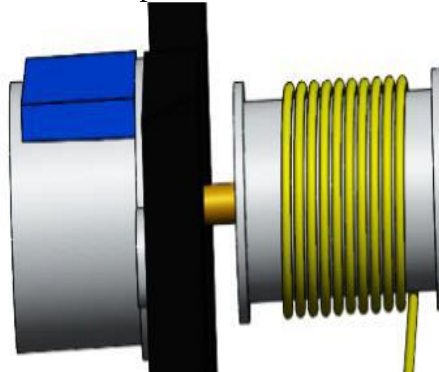


Figure2.3 The motors and pulleys in our system

2.3 Interface to Control the System

2.3.1 Graphical user interface GUI

In this section, we will introduce the user interface designed to control our robot. The development of this interface was accomplished through the utilization of LabView Software, chosen for its user-friendly nature and its proven track record of delivering excellent results in various applications. Specifically, LabView was employed as the control software for managing the operations of the pyramid robot, as previously discussed in Reference [15]. This carefully crafted interface serves as the gateway through which users can interact with and command the robot, ensuring a seamless and efficient control experience.

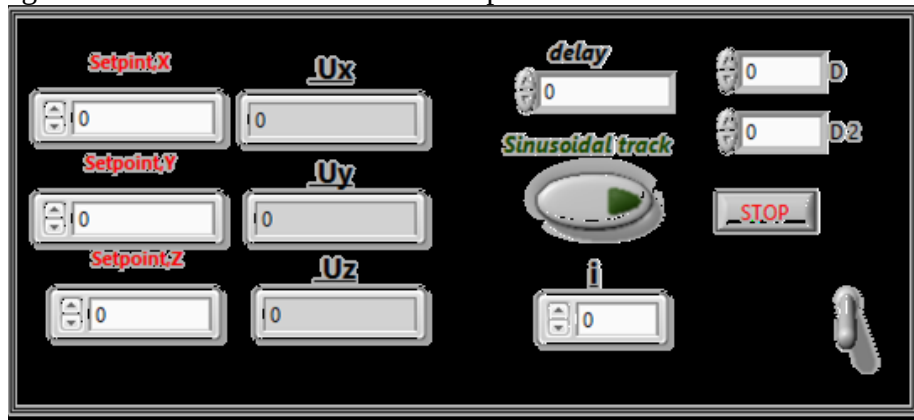


Figure 2.4 Upper part of the graphical control interface

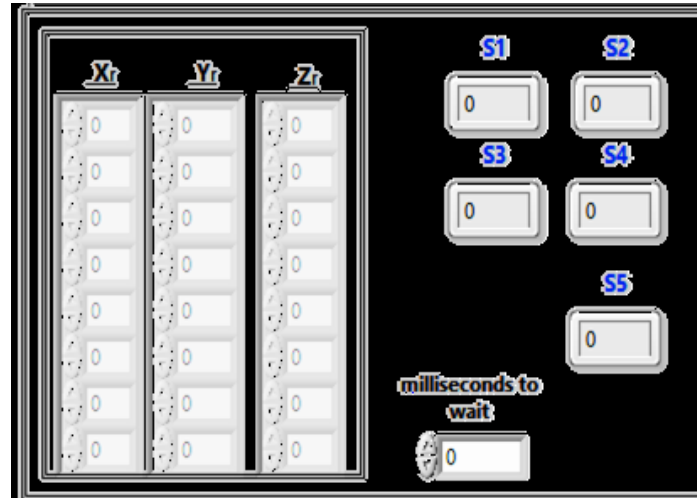


Figure 2.5 :Lower part of the graphical control interface.

Figure 2.4:depicts the upper segment of the graphical control interface, which plays a pivotal role in the incremental creation of a dashed path, point by point. This section of the interface offers users the capability to meticulously design the specific trajectory they desire for the robot's movement. On the other hand, Figure 2.5 provides insight into the lower section of the graphical control interface. This particular component is harnessed to generate intricate and uninterrupted paths that demand a multitude of data points for precise definition. The lower portion of the interface is instrumental in crafting complex, continuous paths, and is particularly advantageous

when the robot's motion requires a high level of detail and precision. Together, these graphical control elements empower users to tailor the robot's path precisely to their unique requirements, whether necessitating dashed or continuous trajectories.

2.3.2 Strategy the Control Used

In this work, an open-loop control strategy was adopted to control the robot's movement using a DC motor equipped with a gearbox and an encoder position sensor. This strategy is characterized by its ease of implementation and lower complexity compared to closed-loop control systems, as movement commands are sent directly to the motor without the need for continuous feedback to correct errors. As shown in the figure2.6, the controller receives the required reference values and then converts them into appropriate control signals, which are sent to the motor driver circuit. The motors are then supplied with the necessary voltage and current to produce the desired movement for the robotic system.



Figure 2.6 DC motor

The necessary mathematical models are:

- direct and inverse geometric models that express the position of the end effector as a function of the joint variables and vice versa.
- Direct and inverse kinematic models express the velocities of the end effector as a function of joint variables and vice versa.
- Dynamic models define the robot's motion equations, establishing the relationships between the torques or forces exerted by the robot and the positions, velocities, and accelerations of the joints.

2.4 Geometric Modeling

The direct and inverse geometric models express the position of the end organ as a function of the joint variables of the mechanism, and vice versa.

Figure (2.7) shows the end organ in the (x, y) position and the different lengths of the cables (L_i) as well as the angles that these cables make with the x-axis (θ_i).

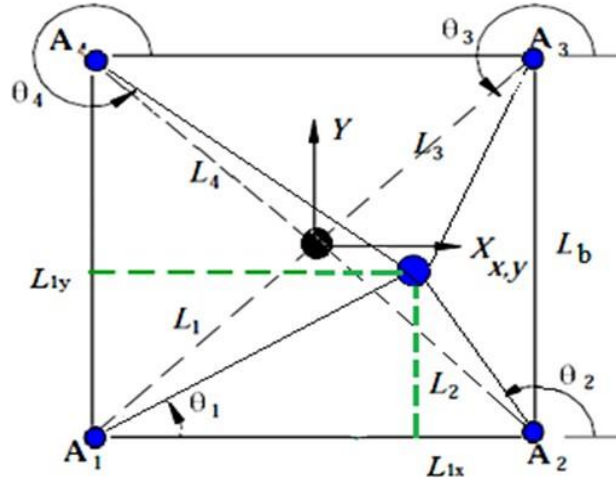


FIGURE 2.7 – Geometric model of the four-cable robot

2.4.1 Inverse Geometric Modeling

Inverse geometric modeling consists of calculating the joint coordinates corresponding to a given position of the end organ. When it exists, the explicit form that yields all possible solutions (there is rarely a unique solution) constitutes what is called the inverse geometric model (IGM) [16].

We seek to find the cable lengths (L_i) and angles (θ_i) as a function of the effector coordinates (x , y), expressed by the following equations:

$$L_i = \sqrt{(L_{ix}^2 + L_{iy}^2)} \quad i = 1,4 \quad (2.1)$$

Avec :

$$L_{ix} = x - A_{ix} \text{ et } L_{iy} = y - A_{iy} \quad i = 1,4 \quad (2.2)$$

Substituting into equation (2.1) we find:

$$L_i = \sqrt{(x - A_{ix})^2 + (y - A_{iy})^2} \quad i = 1,4 \quad (2.3)$$

For the angles we have:

$$\theta_i = \arctang\left(\frac{y - A_{iy}}{x - A_{ix}}\right) \quad i = 1,4 \quad (2.4)$$

2.4.2 Direct Geometric Model

The direct geometric model of a robot allows the calculation of the operational coordinates that define the position of the end effector as a function of the joint coordinates [16].

Therefore, the position of the effector (x , y) is expressed as a function of the cable lengths (L_i). The direct geometric model is difficult to solve due to its closed structure (the angles (θ_i) are related to the cable lengths (L_i)).

This problem can be simplified by moving the frame $R(O, X, Y)$ to point A_1 , which gives us new coordinates for points $A_1 = (O, O)$ and $A_2 = (L_b, O)$, as shown in Figure 2.2. Therefore, the solution to the direct geometric model is the intersection of two circles, one with center A_1 and radius L_1 , and the other with center A_2 and radius L_2 :

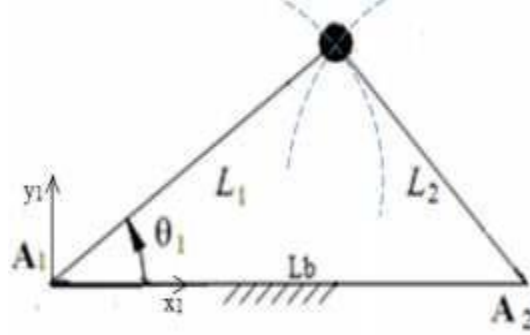


Figure 2.8 : Change of reference frame for calculating the kinematic model.

$$x_1^2 + y_1^2 = L_1^2 \quad (2.5a)$$

$$(L_b - x_1)^2 + y_1^2 = L_2^2 \quad (2.5b)$$

$$(L_b - x_1)^2 - x_1^2 = L_2^2 - L_1^2 \quad (2.5c)$$

$$2x_1L_b - L_b^2 = L_2^2 - L_1^2 \quad (2.5d)$$

Hence:

$$\begin{cases} x = \frac{L_b^2 + L_1^2 - L_2^2}{2L_b} \\ y = \pm \sqrt{(L_b^2 - (L_b - x)^2)} = \pm \sqrt{(L_1^2 - x^2)} \end{cases}$$

2.5 Kinematic Modeling

Kinematic modeling of a robot in general allows us to establish the relationships between the (generalized) joint velocities and the operational velocities of the end effector. In this section, we present the inverse and forward kinematic modeling of the 4-cable planar robot [16].

2.5.1 Inverse Kinematic Model

To calculate the inverse kinematic model, we consider the *iemme* vector obtained from equations (2.2):

$$x = A_{ix} + L_i \cos(\theta_i) \quad i = 1,4 \quad (2.7a)$$

$$y = A_{iy} + L_i \sin(\theta_i) \quad i = 1,4 \quad (2.7b)$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A_{ix} + L_i \cos(\theta_i) \\ A_{iy} + L_i \sin(\theta_i) \end{pmatrix} \quad (2.8)$$

If we differentiate $\begin{pmatrix} x \\ y \end{pmatrix}$ with respect to time, we obtain:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos(\theta_i) - L_i \sin(\theta_i) \\ \sin(\theta_i) + L_i \cos(\theta_i) \end{pmatrix} \begin{pmatrix} \dot{L}_i \\ \dot{\theta}_i \end{pmatrix} \quad i = 1,4 \quad (2.9)$$

By reversing equation (2.9) we obtain:

$$\begin{pmatrix} \dot{L}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\frac{\cos(\theta_i)}{L_i} & \frac{\sin(\theta_i)}{L_i} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad i = 1,4 \quad (2.10)$$

Since we are interested in the cable speeds as a function of the effector speed, we can extract the first line of (2.10) to obtain:

$$\begin{pmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (2.11)$$

$$M = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \quad M: \text{Inverted Jacobean} \quad (2.12)$$

2.5.2 Direct Kinematic Model

The direct kinematic model of a manipulator robot describes the velocities of the operational coordinates as a function of the joint velocities.

To obtain the direct kinematic model, we must invert equation (2.11), which gives us:

$$\dot{X} = M^{-1} \dot{L} \quad (2.13)$$

Therefore, the solution requires calculating the inverse Jacobian matrix. Due to actuation redundancy, M is not square but has dimension $(4 * 2)$, which creates a problem calculating M^{-1} . To solve this problem, we used the Moore-Penrose pseudoinverse:

$$X' = M^+ L' \quad \text{Ou} \quad M^+ = (M^T M)^{-1} M^T \quad (2.14)$$

Such as :

M^+ : Pseudo-inverse of matrix M .

$\dot{X} = (\dot{x}, \dot{y})$: The Cartesian velocities of the effector.

2.6 Dynamic Model

Dynamic models define the robot's equations of motion, which establish the relationships between the torques or forces exerted by the actuators and the positions, velocities, and accelerations of the joints.

The dynamic analysis of the system can be performed using various methods, including Newton-Euler, virtual power, and Lagrange's method. We will use two of these methods to establish the equations of motion for the end effector.

2.6.1 The Lagrange Method

For the dynamic model derived in this section, we assume that the cables of the four-cable planar parallel robot are massless and perfectly rigid, so we do not consider their inertias or elasticities. We also ignore Coulomb friction and instead model linear viscous friction to account for friction losses. Despite these simplifications, the resulting model is coupled and nonlinear.

The Lagrange formalism describes the equations of motion in terms of the work and energy of the system, which translates, when the external force on the end effector is assumed to be zero, into the following equation:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = Q_d \quad (2.15)$$

Or:

L is the Lagrangian of the system defined by:

$$L = T - U \quad (2.16)$$

T : Total kinetic energy of the system.

U : Total potential energy of the system.

D : Total dissipation energy of the system.

Q_d : Is the vector of given forces related to the torques of the motors.

2.6.1.1 Calculation of the angular velocities and accelerations of the pulleys

To calculate the kinetic and dissipation energies of the system, we must first know the angular velocities and accelerations of the pulleys. Equation (2.17) gives the angles of rotation β_i of the pulleys:

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} L_{10} - L_1 \\ L_{20} - L_2 \\ L_{30} - L_3 \\ L_{40} - L_4 \end{pmatrix} \quad (2.17)$$

By differentiating the equation with respect to time, we obtain the angular velocities of the pulleys:

$$\dot{\beta} = -\frac{1}{r} \begin{pmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \end{pmatrix} = -\frac{1}{r} \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (2.18)$$

By differentiating equation (2.18) with respect to time, we obtain the angular accelerations of the pulleys:

$$\ddot{\beta} = \frac{1}{r} \begin{pmatrix} -\dot{\theta}_1 \sin(\theta_1) & \sin(\theta_1) \\ -\dot{\theta}_2 \sin(\theta_2) & \sin(\theta_2) \\ -\dot{\theta}_3 \sin(\theta_3) & \sin(\theta_3) \\ -\dot{\theta}_4 \sin(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} - \begin{pmatrix} \sin(\theta_1) & \cos(\theta_1) \\ \sin(\theta_2) & \cos(\theta_2) \\ \sin(\theta_3) & \cos(\theta_3) \\ \sin(\theta_4) & \cos(\theta_4) \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} \quad (2.19)$$

2.6.1.2 Kinetic Energy of the System

This is the sum of the kinetic energy of the effector and the kinetic energy of the pulleys, and is expressed as follows:

$$T = \frac{1}{2} m V^2 + \frac{1}{2} J_1 \dot{\beta}_1^2 + \frac{1}{2} J_2 \dot{\beta}_2^2 + \frac{1}{2} J_3 \dot{\beta}_3^2 + \frac{1}{2} J_4 \dot{\beta}_4^2 \quad (2.20)$$

Such that:

V_e : is the velocity of the effector, expressed as:

$$V_e^2 = \dot{x}^2 + \dot{y}^2 \quad (2.21)$$

m : Is the mass of the effector.

J_1, J_2, J_3, J_4 : are the moments of inertia of the pulleys.

$$\dot{\beta}_1, \dot{\beta}_2, \dot{\beta}_3, \dot{\beta}_4, : \text{ are the angular velocities of the pulleys expressed using equation } \quad (2.22)$$

Substituting these equations into equation (2.20), we find:

$$T = \frac{1}{2} m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2r^2} J_1(\cos(\theta_1) \dot{x} + \sin(\theta_1)\dot{y})^2 + \frac{1}{2r^2} J_2(\cos(\theta_2) \dot{x} + \sin(\theta_2)\dot{y})^2 + \frac{1}{2r^2} J_3(\cos(\theta_3) \dot{x} + \sin(\theta_3)\dot{y})^2 + \frac{1}{2r^2} J_4(\cos(\theta_4) \dot{x} + \sin(\theta_4)\dot{y})^2 \quad (2.23)$$

2.6.1.3 The potential energy of the system

We know that the effector moves in a horizontal plane, the pulleys are fixed, and the mass of the strings is negligible, therefore the potential energy of the system is zero; thus:

$$U = 0 \quad \text{et} \quad L = T \quad (2.24)$$

2.6.1.4 System Dissipation Energy

Dissipation energy, primarily due to friction in engines and polishing, is defined as follows:

$$D = \frac{1}{2} C_1 \dot{\beta}_1^2 + \frac{1}{2} C_2 \dot{\beta}_2^2 + \frac{1}{2} C_3 \dot{\beta}_3^2 + \frac{1}{2} C_4 \dot{\beta}_4^2 \quad (2.25)$$

C_1, C_2, C_3, C_4 : The viscous damping coefficients due to the motors. Substituting $\dot{\beta}_i$ with their expressions, we obtain:

$$D = \frac{1}{2} C_1 (\cos(\theta_1) \dot{x} + \sin(\theta_1)\dot{y})^2 + \frac{1}{2} C_2 (\cos(\theta_2) \dot{x} + \sin(\theta_2)\dot{y})^2 + \frac{1}{2} C_3 (\cos(\theta_3) \dot{x} + \sin(\theta_3)\dot{y})^2 + \frac{1}{2} C_4 (\cos(\theta_4) \dot{x} + \sin(\theta_4)\dot{y})^2 \quad (2.26)$$

2.6.1.5 Calculation of Derivatives

We differentiate the kinetic energy equation (2.22) by adding $\dot{x}$ and we find:

We differentiate this last equation (2.26) by bringing the time into, obtaining:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{x}} \right) = m\ddot{x} + \frac{J_1}{r^2} [\cos^2(\theta_1) \ddot{x} + \cos(\theta_1) \sin(\theta_1) \ddot{y} - 2\dot{\theta}_1 \cos(\theta_1) \sin(\theta_1) \dot{x} + \dot{\theta}_1 (2 \cos^2(\theta_1) - 1) \dot{y}] \\ + \frac{J_2}{r^2} [\cos^2(\theta_2) \ddot{x} + \cos(\theta_2) \sin(\theta_2) \ddot{y} - 2\dot{\theta}_2 \cos(\theta_2) \sin(\theta_2) \dot{x} + \dot{\theta}_2 (2 \cos^2(\theta_2) - 1) \dot{y}] + \frac{J_3}{r^2} [\cos^2(\theta_3) \ddot{x} + \cos(\theta_3) \sin(\theta_3) \ddot{y} - 2\dot{\theta}_3 \cos(\theta_3) \sin(\theta_3) \dot{x} + \dot{\theta}_3 (2 \cos^2(\theta_3) - 1) \dot{y}] \\ + \frac{J_4}{r^2} [\cos^2(\theta_4) \ddot{x} + \cos(\theta_4) \sin(\theta_4) \ddot{y} - 2\dot{\theta}_4 \cos(\theta_4) \sin(\theta_4) \dot{x} + \dot{\theta}_4 (2 \cos^2(\theta_4) - 1) \dot{y}] \end{aligned} \quad (2.27)$$

OneDrive, the energy dissipation equation (2.25) by $\dot{x}$ brings us:

$$\left(\frac{\partial D}{\partial \dot{x}} \right) = \frac{1}{r} C_1 \cos(\theta_1) (\cos(\theta_1) \dot{x} + \sin(\theta_1)\dot{y}) + \frac{1}{r} C_2 \cos(\theta_2) (\cos(\theta_2) \dot{x} + \sin(\theta_2)\dot{y}) + \frac{1}{r} C_3 \cos(\theta_3) (\cos(\theta_3) \dot{x} + \sin(\theta_3)\dot{y}) + \frac{1}{r} C_4 \cos(\theta_4) (\cos(\theta_4) \dot{x} + \sin(\theta_4)\dot{y}) \quad (2.28)$$

We also have:

$$\frac{\partial T}{\partial x} = 0 \quad (2.29)$$

We differentiate the kinetic energy equation (2.22) by contributing y' and find:

$$\begin{aligned} \frac{\partial T}{\partial \dot{y}} = & m\dot{y}' + \frac{1}{r^2} J_1 \sin(\theta_1)(\cos(\theta_1) \dot{x}' + \sin(\theta_1)\dot{y}') + \frac{1}{r^2} J_2 \sin(\theta_2)(\cos(\theta_2) \dot{x}' + \sin(\theta_2)\dot{y}') \\ & + \frac{1}{r^2} J_3 \sin(\theta_3)(\cos(\theta_3) \dot{x}' + \sin(\theta_3)\dot{y}') + \frac{1}{r^2} J_4 \sin(\theta_4)(\cos(\theta_4) \dot{x}' + \sin(\theta_4)\dot{y}') \end{aligned} \quad (2.30)$$

Deriving this last equation (2.30) by bringing the time into, we obtain:

$$\begin{aligned} \frac{\partial T}{\partial \dot{y}} = & m\ddot{y} + \frac{J_1}{r^2} [\sin^2(\theta_1)\ddot{y} + \cos(\theta_1)\sin(\theta_1)\ddot{x} + 2\dot{\theta}_1 \cos(\theta_1)\sin(\theta_1)\dot{y} + \dot{\theta}_1^2 \cos^2(\theta_1) - \sin^2(\theta_1)\dot{x}'] + \\ & \frac{J_2}{r^2} [\sin^2(\theta_2)\ddot{y} + \cos(\theta_2)\sin(\theta_2)\ddot{x} + 2\dot{\theta}_2 \cos(\theta_2)\sin(\theta_2)\dot{y} + \dot{\theta}_2^2 \cos^2(\theta_2) - \sin^2(\theta_2)\dot{x}'] + \\ & \frac{J_3}{r^2} [\sin^2(\theta_3)\ddot{y} + \cos(\theta_3)\sin(\theta_3)\ddot{x} + 2\dot{\theta}_3 \cos(\theta_3)\sin(\theta_3)\dot{y} + \dot{\theta}_3^2 \cos^2(\theta_3) - \sin^2(\theta_3)\dot{x}'] + \\ & \frac{J_4}{r^2} [\sin^2(\theta_4)\ddot{y} + \cos(\theta_4)\sin(\theta_4)\ddot{x} + 2\dot{\theta}_4 \cos(\theta_4)\sin(\theta_4)\dot{y} + \dot{\theta}_4^2 \cos^2(\theta_4) - \sin^2(\theta_4)\dot{x}'] \end{aligned} \quad (2.31)$$

OneDrive's dissipation equation (2.25) per y' provides:

$$\begin{aligned} \frac{\partial D}{\partial \dot{x}} = & \frac{1}{r^2} C_1 \sin(\theta_1)(\cos(\theta_1) \dot{x}' + \sin(\theta_1)\dot{y}') + \frac{1}{r^2} C_2 \sin(\theta_2)(\cos(\theta_2) \dot{x}' + \sin(\theta_2)\dot{y}') \\ & + \frac{1}{r^2} C_3 \sin(\theta_3)(\cos(\theta_3) \dot{x}' + \sin(\theta_3)\dot{y}') + \frac{1}{r^2} C_4 \sin(\theta_4)(\cos(\theta_4) \dot{x}' + \sin(\theta_4)\dot{y}') \end{aligned} \quad (2.32)$$

We also have:

$$\frac{\partial T}{\partial y} = 0 \quad (2.33)$$

2.6.1.6 Calculation of the Qd coefficients

These are determined from the virtual power developed by the 4 motors, which is given by the relation (the ϕ_i denote the torques on the pulleys):

$$P^* = \tau_1 \dot{\beta}_1^* + \tau_2 \dot{\beta}_2^* + \tau_3 \dot{\beta}_3^* + \tau_4 \dot{\beta}_4^* \quad (2.34)$$

Such as :

$$\dot{\beta}^* = \frac{1}{r} \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x}^* \\ \dot{y}^* \end{pmatrix} \quad (2.35)$$

$$P^* = -\frac{1}{r} (\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4) \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x}^* \\ \dot{y}^* \end{pmatrix} \quad (2.36)$$

From equation (2.36) we derive the expressions for the given forces along x and y:

$$Q_{dx} = -\frac{1}{r} (\tau_1 \cos(\theta_1) + \tau_2 \cos(\theta_2) + \tau_3 \cos(\theta_3) + \tau_4 \cos(\theta_4)) \quad (2.37a)$$

$$Q_{dy} = -\frac{1}{r} (\tau_1 \sin(\theta_1) + \tau_2 \sin(\theta_2) + \tau_3 \sin(\theta_3) + \tau_4 \sin(\theta_4)) \quad (2.37b)$$

We can write:

$$Q_d = \frac{1}{r} S(X) \tau \quad (2.38)$$

Such as :

$$S(X) \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix} \quad (2.39)$$

And :

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} \quad (2.40)$$

2.6.1.7 Lagrange Equations in Matrix Form

After substituting the equations for the derivatives of kinetic energy and dissipation energy into the Lagrange formula, and after manipulating these equations, we can write them in the following matrix form:

$$M(X)X'' + N(X, X')X' = S(X)\tau \quad (2.41)$$

With :

$$Q = rQ_d = S(X)\tau \quad (2.42)$$

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad N = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} \quad (2.43)$$

Such as :

$$M_{11} = r * m + \frac{1}{r} (J_1 \cos^2(\theta_1) + J_2 \cos^2(\theta_2) + J_3 \cos^2(\theta_3) + J_4 \cos^2(\theta_4)) \quad (2.44a)$$

$$M_{12} = \frac{1}{r} (J_1 \cos(\theta_1) \sin(\theta_1) + J_2 \cos(\theta_2) \sin(\theta_2) + J_3 \cos(\theta_3) \sin(\theta_3) + J_4 \cos(\theta_4) \sin(\theta_4)) \quad (2.44b)$$

$$M_{22} = r * m + \frac{1}{r} (J_1 \sin^2(\theta_1) + J_2 \sin^2(\theta_2) + J_3 \sin^2(\theta_3) + J_4 \sin^2(\theta_4)) \quad (2.44c)$$

$$N_{11} = \frac{1}{r} [\cos(\theta_1) (C_1 \cos(\theta_1) - 2J_1 \dot{\theta}_1 \sin(\theta_1)) + \cos(\theta_2) (C_2 \cos(\theta_2) - 2J_2 \dot{\theta}_2 \sin(\theta_2)) + \cos(\theta_3) (C_3 \cos(\theta_3) - 2J_3 \dot{\theta}_3 \sin(\theta_3)) + \cos(\theta_4) (C_4 \cos(\theta_4) - 2J_4 \dot{\theta}_4 \sin(\theta_4))] \quad (2.44d)$$

$$N_{12} = \frac{1}{r} [C_1 \sin(\theta_1) \cos(\theta_1) + J_1 \dot{\theta}_1 (\cos^2(\theta_1) - \sin^2(\theta_1)) + C_2 \sin(\theta_2) \cos(\theta_2) + J_2 \dot{\theta}_2 (\cos^2(\theta_2) - \sin^2(\theta_2)) + C_3 \sin(\theta_3) \cos(\theta_3) + J_3 \dot{\theta}_3 (\cos^2(\theta_3) - \sin^2(\theta_3)) + C_4 \sin(\theta_4) \cos(\theta_4) + J_4 \dot{\theta}_4 (\cos^2(\theta_4) - \sin^2(\theta_4))] \quad (2.44e)$$

$$N_{21} = N_{12} \quad (2.44f)$$

$$N_{22} = \frac{1}{r} [\sin(\theta_1) (C_1 \cos(\theta_1) - 2J_1\dot{\theta}_1 \cos(\theta_1)) + \sin(\theta_2) (C_2 \cos(\theta_2) - 2J_2\dot{\theta}_2 \cos(\theta_2)) + \sin(\theta_3) (C_3 \cos(\theta_3) - 2J_3\dot{\theta}_3 \cos(\theta_3)) + \sin(\theta_4) (C_4 \cos(\theta_4) - 2J_4\dot{\theta}_4 \cos(\theta_4))] \quad (2.44g)$$

2.6.2 Newton-Euler Method

2.6.2.1 Dynamic Model of the Effector

The dynamic model of the effector is expressed by the following relation:

$$m\ddot{X} = F_R \quad (2.45)$$

Where:

m : is the mass of the effector.

$\ddot{X}$: is the acceleration of the effector.

F_R : is the resultant force of all the cable tensions applied to the effector. Therefore, equation (2.45) can be written in the following matrix form:

$$\begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} F_{Rx} \\ F_{Ry} \end{pmatrix} \quad (2.46)$$

2.6.2.2 Mechanical Structure of Motors

The dynamic behavior of the motors is expressed by equation (2.47) [17]:

$$J\ddot{\beta} + C\dot{\beta} = \tau - rT \quad (2.47)$$

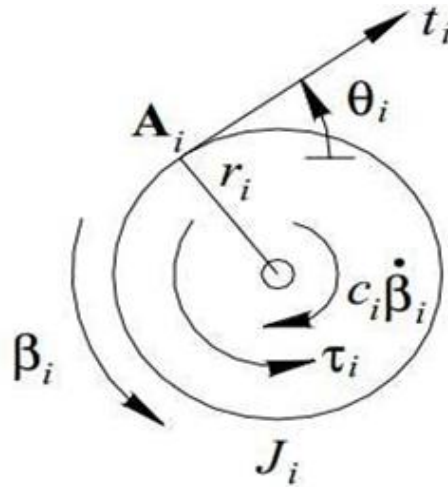


Figure 2.9: Free-body diagram for the i th pulley / shaft [17].

Such as :

$$J = \begin{pmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & J_3 & 0 \\ 0 & 0 & 0 & J_4 \end{pmatrix} C = \begin{pmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{pmatrix} \tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} \quad (2.47)$$

J : is the inertia matrix of the motors.

C : is the matrix of viscous damping coefficients.

r : The radius of the pulleys (they are identical).

τ : is the torque vector supplied by the motors.

T : is the tension vector of the cables. Therefore, we can extract the following relationship:

$$T = \frac{1}{r} (\tau - J\ddot{\beta} - C\dot{\beta}) \quad (2.49)$$

2.6.2.3 Dynamic Model of the System

The overall dynamic model of the system is obtained by combining the dynamic model of the end member and the dynamic model of the motors [18]. If we consider that the pulley angles are zero when the position of the end member is at the center of the square $X = (0 \ 0)^T$, we also have:

$$L_{10} = L_{20} = L_{30} = L_{40} = \frac{L_b \sqrt{2}}{2} \quad (2.50)$$

The relationship between the angles of rotation (β_i) of the pulleys of radius (r) and the variations in cable lengths (ΔL_i) is:

$$\beta_i r = \Delta L_i \quad \text{Avec } \Delta L_i = L_i - L_{i0} \quad (2.51)$$

Such as :

$$L_i = \sqrt{(x - A_{ix})^2 + (y - A_{iy})^2} \quad i = 1, 4 \quad (2.52)$$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} L_{10} - L_1 \\ L_{20} - L_2 \\ L_{30} - L_3 \\ L_{40} - L_4 \end{pmatrix} \quad (2.53)$$

By differentiating equation(2.53) with respect to time, we obtain the angular velocities of the pulleys:

$$\dot{\beta} = -\frac{1}{r} \begin{pmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \end{pmatrix} = -\frac{1}{r} \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (2.54)$$

By differentiating equation (2.54) with respect to time, we obtain the angular accelerations of the pulleys:

$$\ddot{\beta} = \begin{bmatrix} \begin{pmatrix} -\dot{\theta}_1 \sin(\theta_1) & \sin(\theta_1) \\ -\dot{\theta}_2 \sin(\theta_2) & \sin(\theta_2) \\ -\dot{\theta}_3 \sin(\theta_3) & \sin(\theta_3) \\ -\dot{\theta}_4 \sin(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} & \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} \end{bmatrix} \quad (2.55)$$

Hence:

$$T = \frac{1}{r} \left[\tau - J \left(\frac{d}{dt} \left(\frac{\partial \beta}{\partial \dot{X}} \right) \dot{X} + \frac{\partial \beta}{\partial X} \ddot{X} \right) - C \left(\frac{\partial \beta}{\partial \dot{X}} \dot{X} \right) \right] \quad (2.56)$$

With:

$$\dot{\beta} = \left(\frac{\partial \beta}{\partial \dot{X}} \right) \dot{X} \quad (2.57)$$

$$\ddot{\beta} = \frac{d}{dt} \left(\frac{\partial \beta}{\partial \dot{X}} \right) \dot{X} + \frac{\partial \beta}{\partial X} \ddot{X} \quad (2.58)$$

We have the static relationship $F_R = ST$ between the cable tensions and the effector forces at the end.

Such as:

$$S(X) = \begin{pmatrix} -\cos(\theta_1) - \cos(\theta_2) - \cos(\theta_3) - \cos(\theta_4) \\ -\sin(\theta_1) - \sin(\theta_2) - \sin(\theta_3) - \sin(\theta_4) \end{pmatrix} \quad (2.59)$$

Finally, by combining equations (2.46), (2.56) and the static relation $F_R = ST$, we obtain the global dynamic equations of motion, expressed in a standard Cartesian form for robotic systems:

$$M(X)\ddot{X} + N(X, \dot{X})\dot{X} = S(X)\tau \quad (2.60)$$

With:

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad N = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} \quad (2.61)$$

Such as :

$$M_{11} = r * m + \frac{1}{r} (J_1 \cos^2(\theta_1) + J_2 \cos^2(\theta_2) + J_3 \cos^2(\theta_3) + J_4 \cos^2(\theta_4)) \quad (2.62a)$$

$$M_{12} = \frac{1}{r} (J_1 \cos(\theta_1) \sin(\theta_1) + J_2 \cos(\theta_2) \sin(\theta_2) + J_3 \cos(\theta_3) \sin(\theta_3) + J_4 \cos(\theta_4) \sin(\theta_4)) \quad (2.62b)$$

$$M_{22} = r * m + \frac{1}{r} (J_1 \sin^2(\theta_1) + J_2 \sin^2(\theta_2) + J_3 \sin^2(\theta_3) + J_4 \sin^2(\theta_4)) \quad (2.62c)$$

$$N_{11} = \frac{1}{r} [\cos(\theta_1) (C_1 \cos(\theta_1) - 2J_1 \dot{\theta}_1 \sin(\theta_1)) + \cos(\theta_2) (C_2 \cos(\theta_2) - 2J_2 \dot{\theta}_2 \sin(\theta_2)) + \cos(\theta_3) (C_3 \cos(\theta_3) - 2J_3 \dot{\theta}_3 \sin(\theta_3)) + \cos(\theta_4) (C_4 \cos(\theta_4) - 2J_4 \dot{\theta}_4 \sin(\theta_4))] \quad (2.62d)$$

$$N_{12} = \frac{1}{r} [C_1 \sin(\theta_1) \cos(\theta_1) + J_1 \dot{\theta}_1 (\cos^2(\theta_1) - \sin^2(\theta_1)) + C_2 \sin(\theta_2) \cos(\theta_2) + J_2 \dot{\theta}_2 (\cos^2(\theta_2) - \sin^2(\theta_2)) + C_3 \sin(\theta_3) \cos(\theta_3) + J_3 \dot{\theta}_3 (\cos^2(\theta_3) - \sin^2(\theta_3)) + C_4 \sin(\theta_4) \cos(\theta_4) + J_4 \dot{\theta}_4 (\cos^2(\theta_4) - \sin^2(\theta_4))] \quad (2.62e)$$

$$N_{21} = N_{12} \quad (2.62f)$$

$$N_{22} = \frac{1}{r} [\sin(\theta_1) (C_1 \cos(\theta_1) - 2J_1 \dot{\theta}_1 \cos(\theta_1)) + \sin(\theta_2) (C_2 \cos(\theta_2) - 2J_2 \dot{\theta}_2 \cos(\theta_2)) + \sin(\theta_3) (C_3 \cos(\theta_3) - 2J_3 \dot{\theta}_3 \cos(\theta_3)) + \sin(\theta_4) (C_4 \cos(\theta_4) - 2J_4 \dot{\theta}_4 \cos(\theta_4))] \quad (2.62g)$$

2.7 Conclusion

At the start of the day, the team began assembling the four-cable planar robot (a four-cable planar robot) to precisely control this type of intelligent computer. The mathematical model began by analyzing the relationship between tensile forces and cable movement, which helps to optimize performance in various ways. This study also highlights the importance of selecting research findings (kinematic and dynamic) to avoid problems such as perplexing puzzles or the inability to control certain situations. Therefore, the model forms the basis for effective practical applications in fields such as industry, ready-to-use systems, and intelligent systems :

Chapter: 3

Robot Control and Simulation

3.1 Introduction

In this section, we developed a virtual dynamic model using MSC-ADAMS software to simulate the robot's real-world behavior. This simulation also allowed us to validate the robot's kinematic and dynamic models, as well as the Simplex algorithm for calculating optimal torques in real time.

3.2 Simulation with MSC-ADAMS

To validate the torque results obtained using Matlab, we used dynamic simulation software, specifically MSC-ADAMS, whose graphical interface is shown in Figure 3.1.

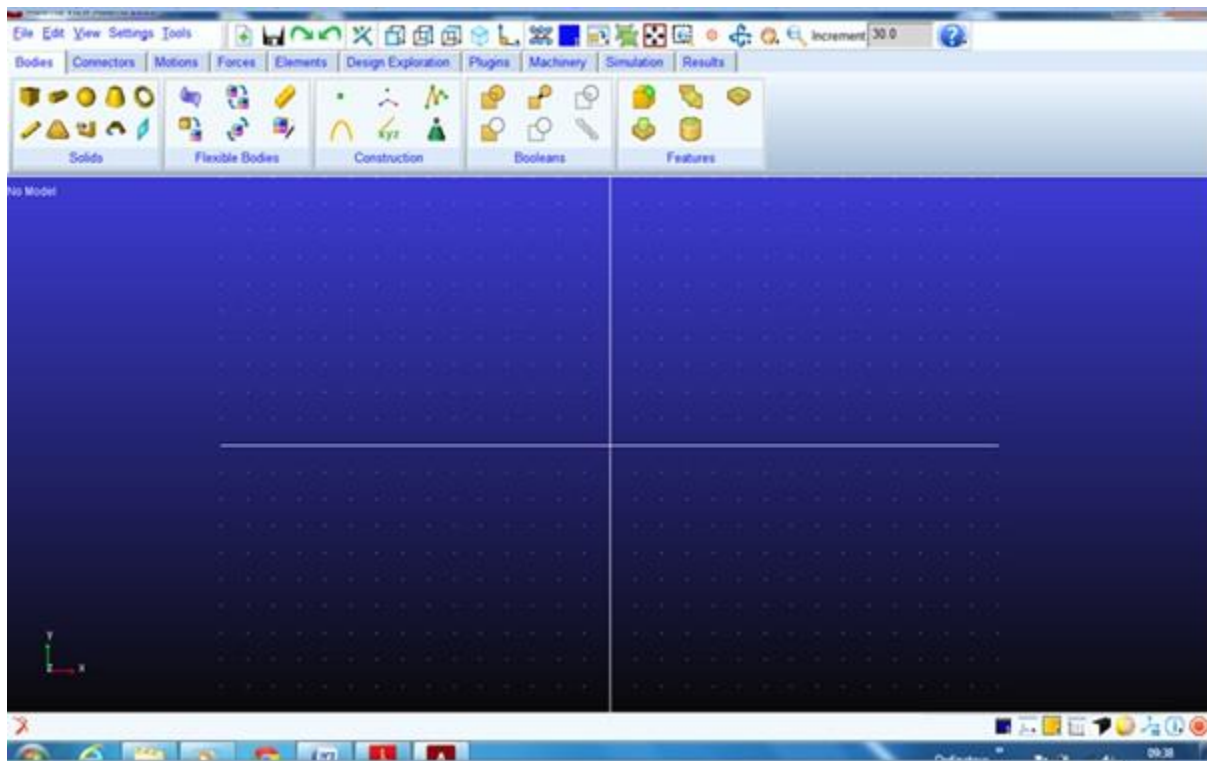


FIGURE 3.1 – MSC-ADAMS Graphical Interface

To perform our MATLAB-MSCADAMS simulation, we will create a virtual dynamic model. Throughout our project, we have assumed inextensible cables; therefore, each cable is modeled by two forces of the same direction but opposite senses, as shown in Figure (3.2). The tensions are represented in red between the effector and the fixed points (the vertices of the square workspace). We could have controlled the effector with a cable/pulley system, but this library is not available in the Student version of ADAMS. For this reason, we chose to control the effector using the tensions obtained in the MATLAB simulation.

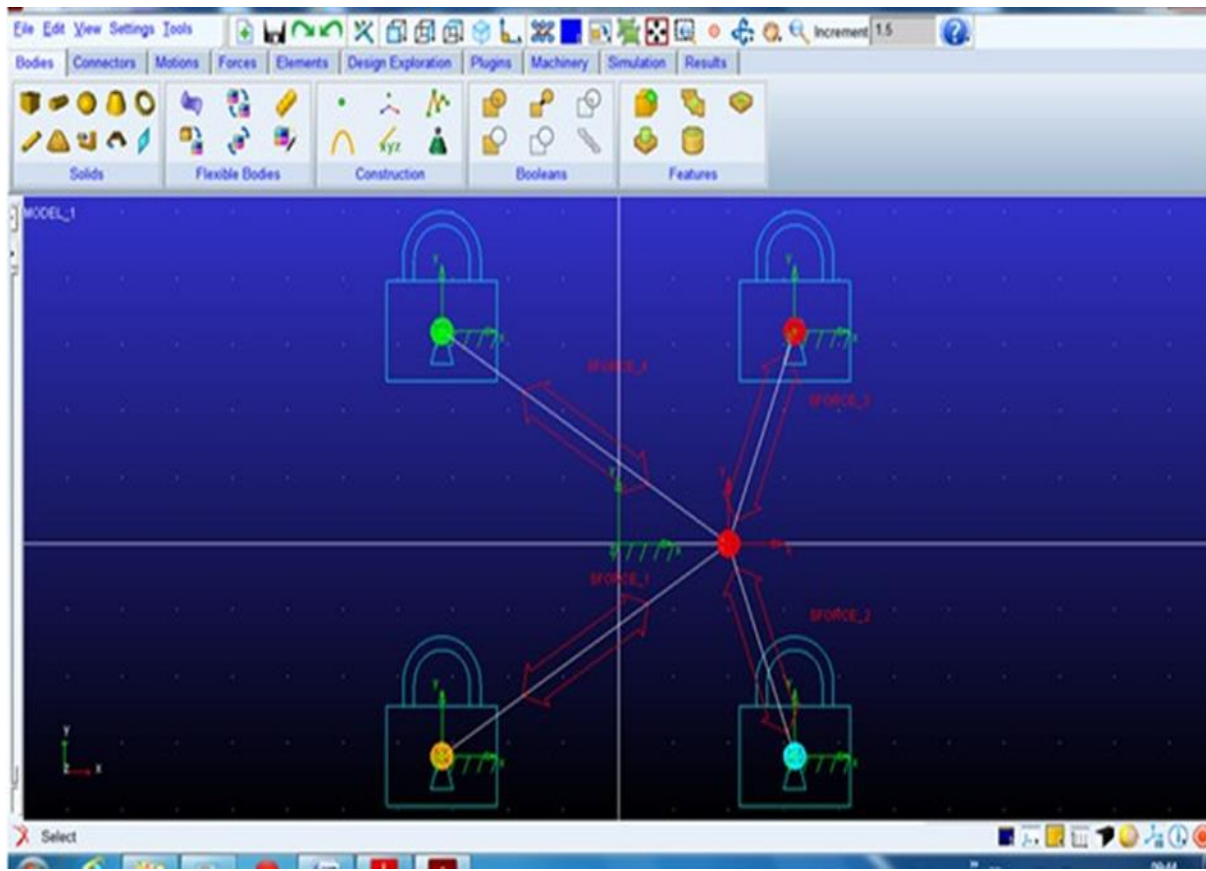


FIGURE 3.2 – Virtual model of the 4-cable robot under MSC-ADAMS

3.3 Procédure de simulation Matlab-ADAMS

Après calcul des couples par Matlab en utilisant les données adéquates au problème, on crée un modèle virtuel sous ADAMS en respectant toutes les données nécessaires : les dimensions de l'espace de travail, la masse, la position initiale de l'effecteur. . . Ensuite, on importe les couples de Matlab en utilisant Simulink, ce dernier permet de faire l'interaction entre les deux logiciels ADAMS et Matlab, le modèle Simulink est représenté sur la figure (3.3) où les tensions T_i sont les outputs de Matlab et les inputs de ADAMS, et les outputs de ADAMS dans ce cas sont les positions x , y ainsi que les erreurs de position $e(x)$, $e(y)$ [18].

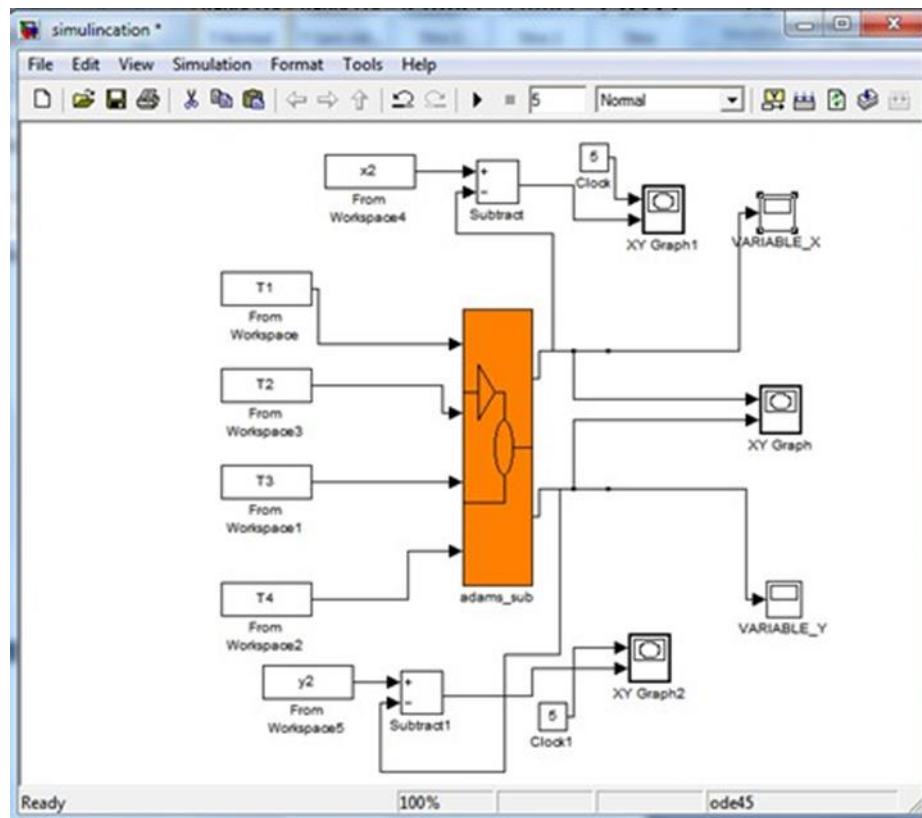


FIGURE 3.3 – Simulink for ADAMS

3.4 Validation and Simulation of the Robot Geometric Model

In this section, we attempt to develop programs using MATLAB to validate the geometric model of a 4-cable planar robot. We represent the variation of cable lengths and angles for different positions or trajectories of the end effector, and also enable a graphical visualization of the 4-cable planar robot.

3.4.1 Simulation of the Inverse Geometric Model

The values of (x, y) are given, and the cable lengths and angles are determined for each displacement of the end effector.

Figure (3.4) illustrates the flowchart of the program calculating the different lengths of the robot's four cables (L_1, L_2, L_3, L_4), as well as the angles ($\theta_1, \theta_2, \theta_3, \theta_4$) that these cables make with the x-axis.

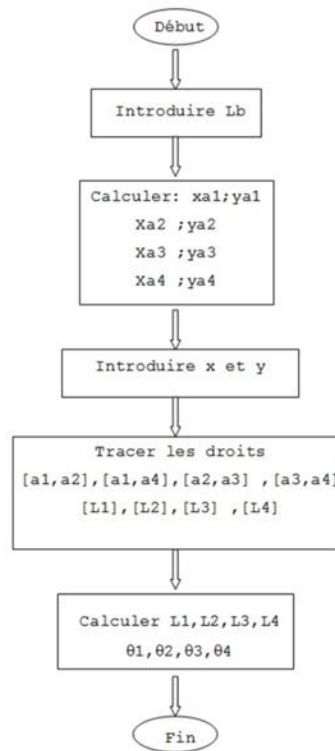


FIGURE 3.4 – Flowchart of the lengths and angles of the 4 robot cables

Figure (3.5) represents the 4-cable robot with the end effector in the rest position ($x_p = 0$, $y_p = 0$). The end effector is marked with a plus sign (+).

$X_p(cm)$	$Y_p(cm)$	$L_1(cm)$	$L_2(cm)$	$L_3(cm)$	$L_4(cm)$	$\theta_1(deg)$	$\theta_2(deg)$	$\theta_3(deg)$	$\theta_4(deg)$
0	0	28.28	28.28	28.28	28.28	45	135	255	315

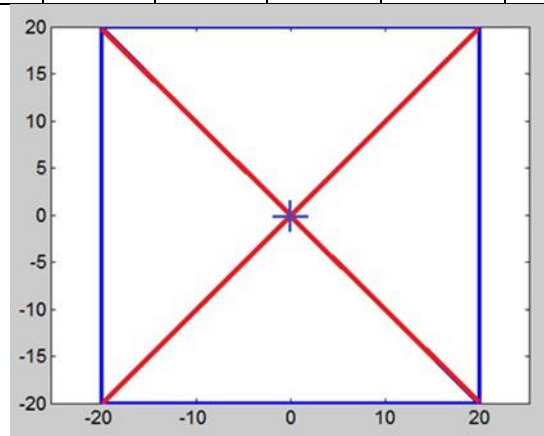


FIGURE 3.5 – Terminal organ in initial position

Figures 3.6 and 3.6 show some terminal organ positions obtained by the program.

Example 1: $x_p = 14$ cm, $y_p = 7$ cm

$x_p(cm)$	$y_p(cm)$	$L_1(cm)$	$L_2(cm)$	$L_3(cm)$	$L_4(cm)$	$\theta_1(deg)$	$\theta_2(deg)$	$\theta_3(deg)$	$\theta_4(deg)$
14	7	43.41	27.65	14.31	36.40	38.45	102.52	245.22	339.07

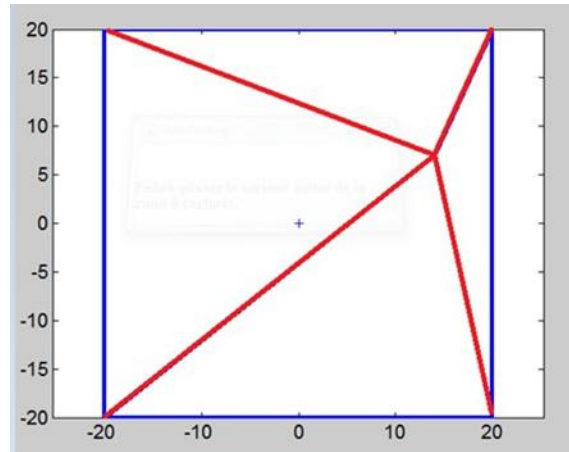


Figure 3.6 – Terminal organ in position (14,7)

Exemple 1 : $x_p = 10$ cm, $y_p = -17$ cm.

$x_p(cm)$	$y_p(cm)$	$L_1(cm)$	$L_2(cm)$	$L_3(cm)$	$L_4(cm)$	$\theta_1(deg)$	$\theta_2(deg)$	$\theta_3(deg)$	$\theta_4(deg)$
10	-17	30.14	10.44	38.32	47.63	5.71	163.30	254.87	309.03

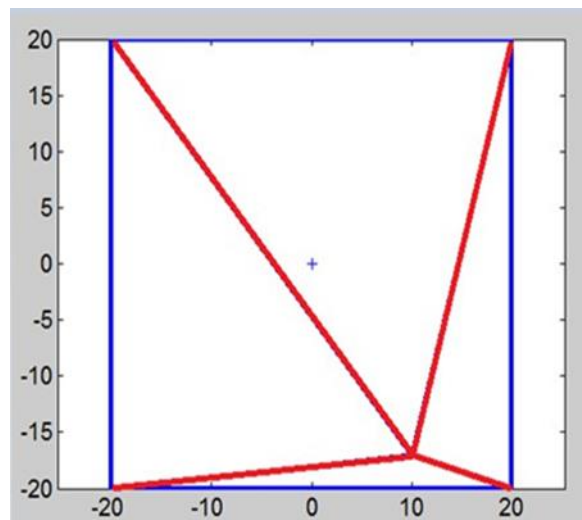


FIGURE 3.7 – Terminal organ in position (10,-17)

Comment:

We observe that the graphical representation of our 4-cable planar robot for the 2 positions of the end effector is very accurate, therefore the inverse geometric model is valid.

3.4.2 Simulation of the End-Method Trajectory

In this section, we attempt to simulate the variations in cable lengths, as well as the angles these cables make with the x-axis, as a function of the polar angle of the end-method.

We propose a circular trajectory of radius R centered at the origin O. The parametric equations of this trajectory are:

$$x = R \cos(\theta)$$

$$y = R \sin(\theta)$$

or`:

$$0 \leq \theta \leq 2\pi \quad \text{et} \quad 0 \leq R \leq R_{\max}$$

or`: $R_{\max} = \frac{L}{2}$ is the distance between the center of the square O and one of the four sides.

Figure (3.8) shows the flowchart of the program tracing the 4-cable geometry, with the circular trajectory followed by the terminal organ.

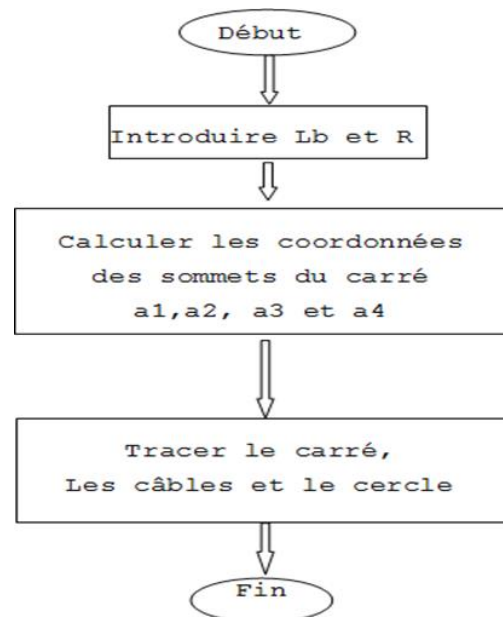


FIGURE 3.8 – Flowchart tracing the 4-cable geometry with the circular trajectory

For a circular trajectory of radius $R = \frac{R_{max}}{2}$ the problem geometry resembles Figure (3.9).

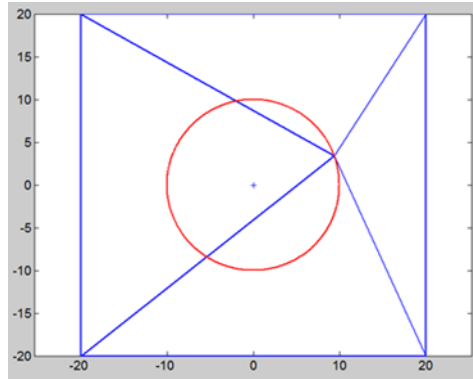


FIGURE 3.9 – circular trajectory of radius $R = \frac{R_{max}}{2}$

Figure (3.10) illustrates the program flowchart tracing the variations of the lengths, cables and their angles with respect to the x-axis, as a function of the polar angle made by the terminal organ when it reverses the circular trajectory and writing

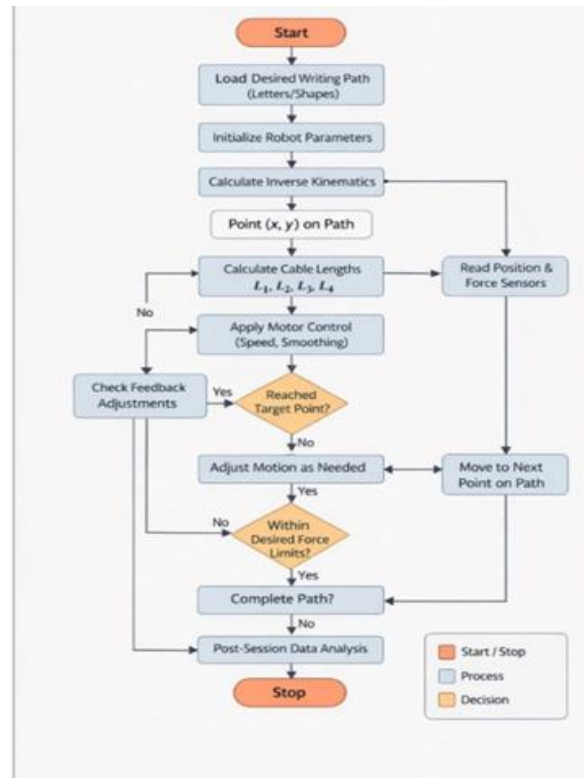


FIGURE 3.10 – Flowchart showing the variations in cable lengths and angles and writing

3.4.2.1 A. Variation of cable lengths

For a circular trajectory of radius $R = R_{max}$, the variations in the lengths of the 4 cables are represented by Figure (3.11).

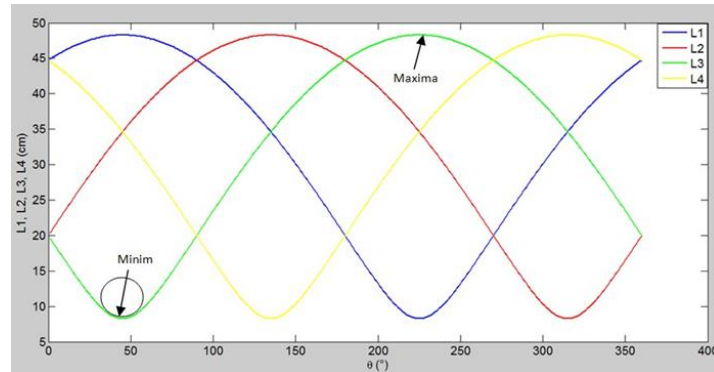


FIGURE 3.11 – Variations in the lengths of the 4 robot cables for $R = R_{\max}$

Comments:

1. We observe that the variations in cable lengths are 2π -periodic.
2. The variations in the lengths of the four cables are similar and repeat alternately every 90° .
3. We also note that the shapes of these curves are non-sinusoidal, which means that the radii of the curves at the minima are smaller than the radii of the curves at the maxima. This difference becomes more pronounced as the radius of the circular trajectory increases.

Figure (3.12) allows us to visualize the variations in cable lengths as a function of the polar angle of the terminal organ, presented in polar form.

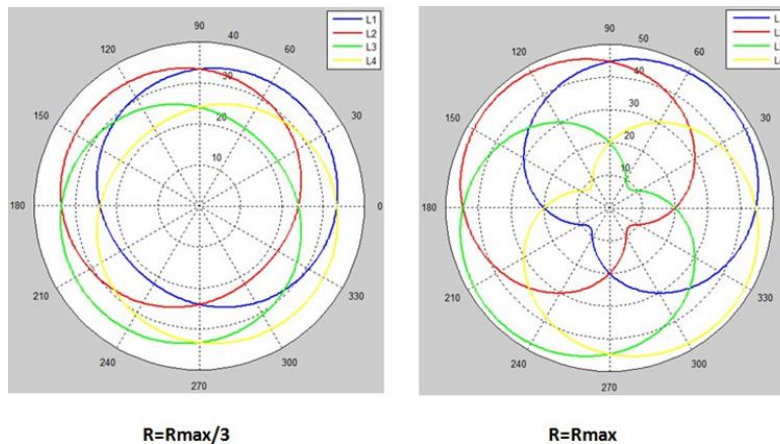


FIGURE 3.12 – Variations in cable lengths in polar form

3.4.2.2 B. Variation of cable angles with respect to the x-axis

Figure (3.13) shows the variations in the angles ($\theta_1, \theta_2, \theta_3, \theta_4$) that the cables make with respect to the x-axis as a function of the polar angle of the terminal organ.

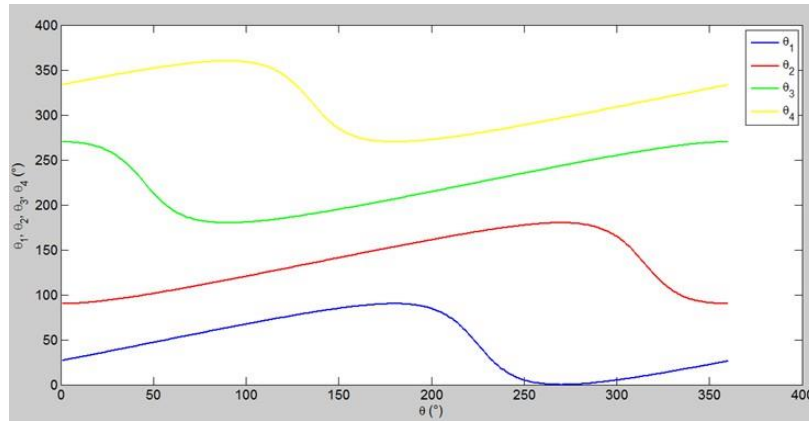


FIGURE 3.13 – Variations in cable angles

3.5 Test continuous trajectories

In this section, we carefully select a variety of letters and shapes to serve as the subject of our active test drawing and writing processes, as per the design proposed within the context of our robot (Cable-Driven parallel robot), as you can see in figures 13 to 18. This meticulous selection ensures that the chosen elements represent a comprehensive range of challenges and scenarios, allowing us to thoroughly evaluate the capabilities and performance of our design in real-world applications.

(a)

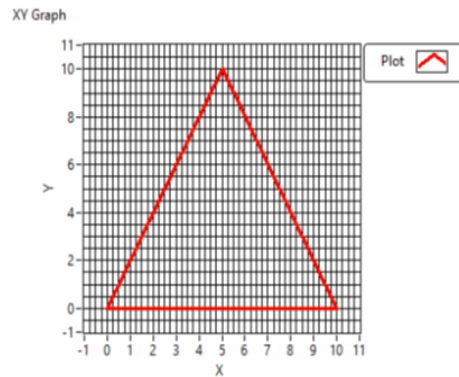


Figure 3.14

(a) drawing a triangle in LabView.

(b) output values after processing

(c) The cables lengths (L1, ...L4).

(a)

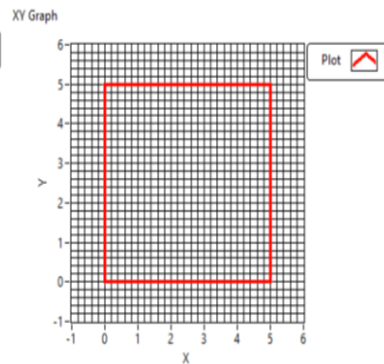


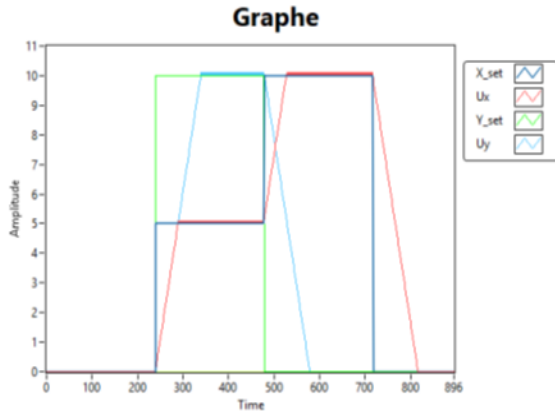
Figure 3.15

(a) drawing a square in LabView.

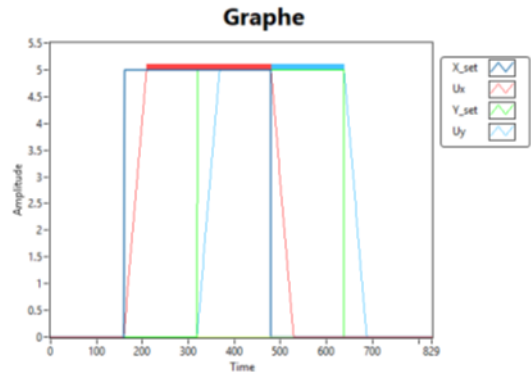
(b) output values

(c) The cables lengths (L1, ...L4).

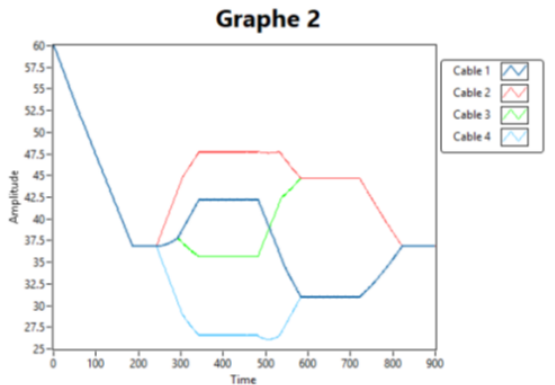
(b)



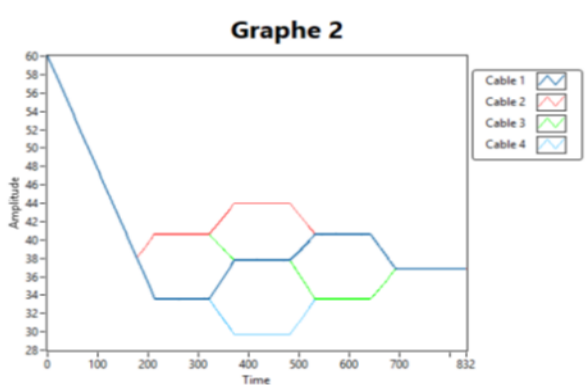
(b)



(c)



(c)



In Figure 3.14(a), we present the results of a validation test, which demonstrates the capability of our system to accurately draw a triangle. In Figure 3.14(b), we showcase the output values obtained after subjecting the system to the control strategy of an open-loop mechanism, and figure 3.14 (c) we showcase the varying the length of the cables over time. This experimental setup allows us to comprehensively assess how our system performs under dynamic conditions, providing valuable insights into its stability and accuracy in responding to changes in cable length during the drawing process.

Figure 3.15(a) reveals the outcomes of a validation test, providing compelling evidence of our system's precision in rendering a square shape. In Figure 3.15(b), we delve into the results derived from our system operating under an open-loop control strategy, while Figure 3.15(c) illustrates the dynamic variation in cable length over time. This multifaceted experimental configuration serves as a comprehensive means to evaluate the system's performance in dynamic

scenarios, offering valuable insights into its stability and precision when adapting to fluctuations in cable length throughout the drawing process.

In Figure 3.16(a), we display the outcomes of a validation test, illustrating our system's proficiency in precisely reproducing the letter "A." Figure 3.16(b) highlights the output data obtained when implementing an open-loop control strategy, while Figure 3.16(c) portrays the dynamic variations in cable length over time. This experimental arrangement offers a comprehensive framework for evaluating our system's performance under dynamic conditions, granting us invaluable insights into its stability and precision when adapting to cable length fluctuations during the writing process.

(a)

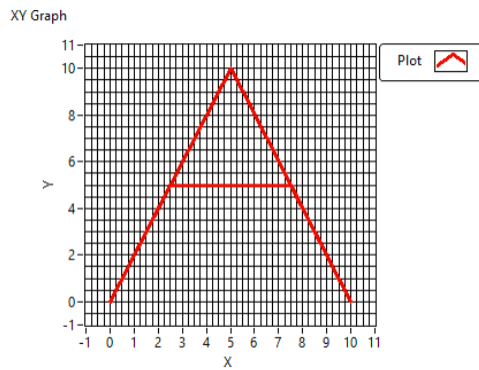


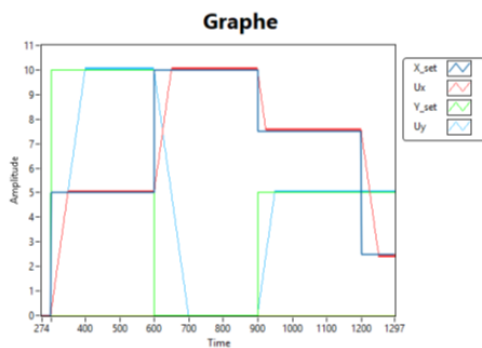
Figure 3.16

(a) writing a letter 'A' in LabView.

(b) output values after processing

(c) The cables lengths (L1, ...L4)

(b)



(a)

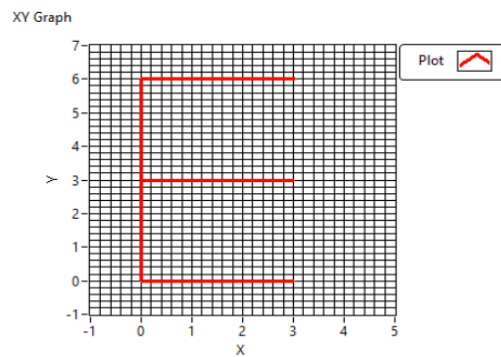


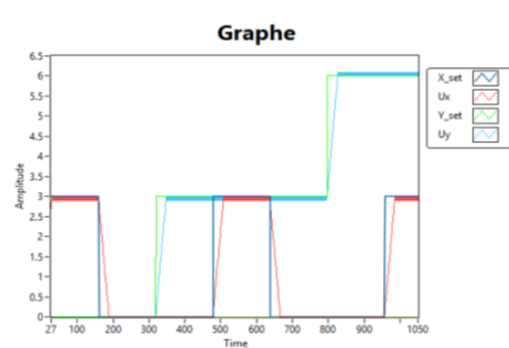
Figure 3.17

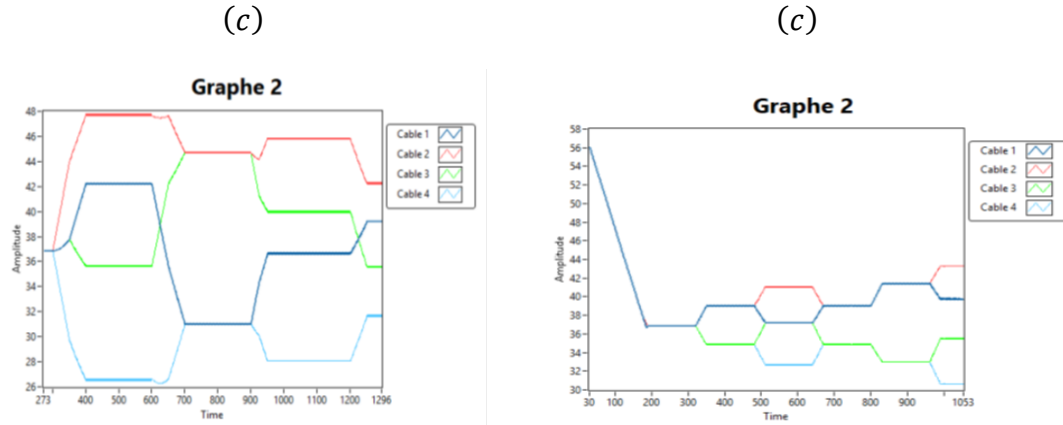
(a) writing a letter 'E' in LabView.

(b) output values after processing

(c) The cables lengths (L1, ...L4)

(b)





In Figure 3.17(a), we present the results of a validation test that demonstrates our system's ability to accurately reproduce the letter "E." Figure 3.17(b) showcases the output values obtained when the system is subjected to the control strategy of an open-loop mechanism, and in Figure 3.17(c), we illustrate the variation in cable length over time. This experimental setup provides a comprehensive assessment of how our system performs under dynamic conditions, offering valuable insights into its stability and accuracy when responding to changes in cable length during the drawing process.

3.6 Conclusion

This chapter summarizes the geometric, kinematic, and dynamic modeling calculations for a four-cable robot (moving in two directions). The modeling of this robot is followed by a simulation (using Matlab software) of the end effector displacements, length variations, and angle changes. And the simulation of paths, specifically character representation. The results of this simulation consistently show strong performance in all tests conducted.

3.7 OPEN LOOP COMMAND

3.7.1 Introduction

In this chapter, we represent the system as a state-space representation to validate the dynamic model and study the stability of the 4-cable planar robot by simulating the different responses (impulse and step) of the open-loop system. We then attempt to optimize the torques provided by each motor of this robot for use in the closed-loop control.

3.7.2 The State-Space Representation of the System

As we saw in section (2.4.1.7), the dynamic equation of the 4-cable planar robot is given by:

$$M(X)\ddot{X} + N(X, \dot{X})\dot{X} = S(X)\tau \quad (3.7.1)$$

Tel que :

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad N = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} \quad (3.7.2)$$

$$S(X) = \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix} \quad \tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} \quad (3.7.3)$$

So we define the following state variables:

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = \dot{x}(t) \\ x_3(t) = y(t) \\ x_4(t) = \dot{y}(t) \end{cases}$$

We differentiate the state variables with respect to time:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \ddot{x}(t) \\ \dot{x}_3(t) = \dot{y}(t) \\ \dot{x}_4(t) = \ddot{y}(t) \end{cases}$$

Using the robot's dynamic equation, we find:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ M_{11}\dot{x}_2(t) + M_{12}\dot{x}_4(t) + N_{11}x_2(t) + N_{12}x_4(t) = u_1(t) \\ \dot{x}_3(t) = x_4(t) \\ M_{21}\dot{x}_2(t) + M_{22}\dot{x}_4(t) + N_{21}x_2(t) + N_{22}x_4(t) = u_2(t) \end{cases} \quad (3.7.6)$$

Such as :

$$U(t) = F_R = S(X)\tau = \begin{pmatrix} F_{Rx} \\ F_{Ry} \end{pmatrix} \quad (3.7.7)$$

SO :

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} -\cos(\theta_1)\tau_1 - \cos(\theta_2)\tau_2 - \cos(\theta_3)\tau_3 - \cos(\theta_4)\tau_4 \\ -\sin(\theta_1)\tau_1 - \sin(\theta_2)\tau_2 - \sin(\theta_3)\tau_3 - \sin(\theta_4)\tau_4 \end{pmatrix} \quad (3.7.8)$$

Therefore, we can write the state representation of the robot as follows:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & M_{11} & 0 & M_{12} \\ 0 & 0 & 1 & 0 \\ 0 & M_{21} & 0 & M_{22} \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -N_{11} & 0 & -N_{12} \\ 0 & 0 & 0 & 1 \\ 0 & -N_{21} & 0 & -N_{22} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \quad (3.7.9)$$

We ask:

$$M_{2d} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & M_{11} & 0 & M_{12} \\ 0 & 0 & 1 & 0 \\ 0 & M_{21} & 0 & M_{22} \end{pmatrix} \quad N_{2d} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -N_{11} & 0 & -N_{12} \\ 0 & 0 & 0 & 1 \\ 0 & -N_{21} & 0 & -N_{22} \end{pmatrix} \quad (3.7.10)$$

$$B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \dot{X}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{pmatrix} \quad U(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \quad (3.7.11)$$

The state representation is then written as follows:

$$\dot{X}(t) = M_{2d}^{-1}N_{2d}X(t) + M_{2d}^{-1}BU(t) \quad (3.7.12)$$

SO :

$$\dot{X}(t) = f(X) + g(X) * U(t) \quad (3.7.13)$$

Such as :

$\dot{X}(t)$: is the state vector of the system.

$f(X)$ and $g(X)$: are nonlinear functions.

$U(t)$: is the control vector of the system.

3.7.3 Simulation of the Robot's Response

In this section, we attempt to simulate the response of a 4-cable planar robot using MATLAB, with the following conditions:

- The inertias of the rotor and pulley of each motor: $J = 0.0008 \text{ kg} \cdot \text{m}^2$ $i = 1:4$.
- The viscous damping coefficients of each motor shaft:

$$C = 0.001 \text{ N} \cdot \text{m} \cdot \text{s} \quad i = 1:4$$

- The mass of the end effector: $m = 1 \text{ kg}$.

- The radius of the pulley for each motor $r_i = 0.05 \text{ m}$ $i = 1:4$.

3.7.3.1 Step Response

The step response of the robot is simulated with a fixed system input ($u_1 = 1 \text{ N}$ and $u_2 = 1 \text{ N}$). Figures (3.7.1) and (3.7.2) represent the open-loop position profiles of the robot:

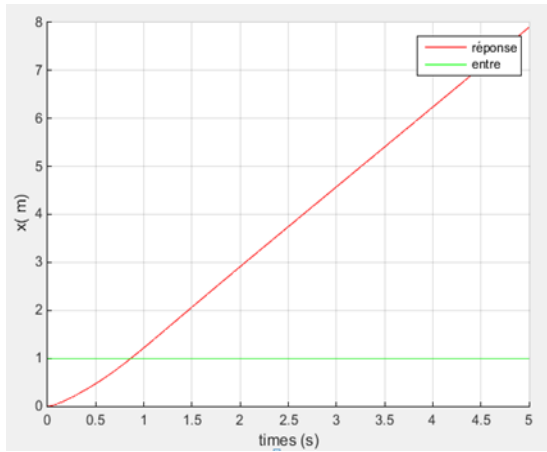


Figure3.7.1: Step response of the robot, Profile of $x(m)$.

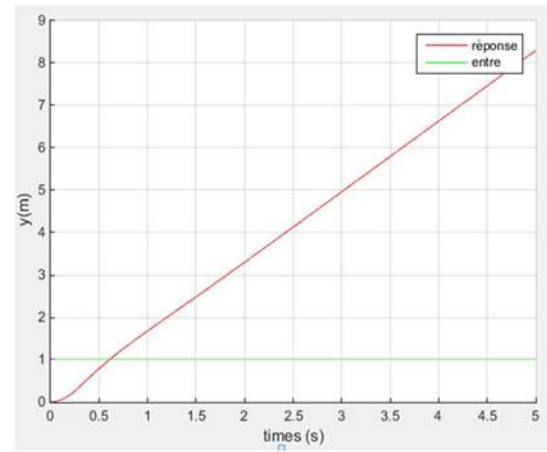


Figure3.7.2: Step response of the robot, Profile of $y(m)$.

3.7.3.2 Impulse Response

The robot's impulse response is simulated with a fixed system input ($u_1 = 1 \text{ N}$ and $u_2 = 1 \text{ N}$) for a very short duration. Figures (3.7.3) and (3.7.4) represent the open-loop position profiles of the robot:

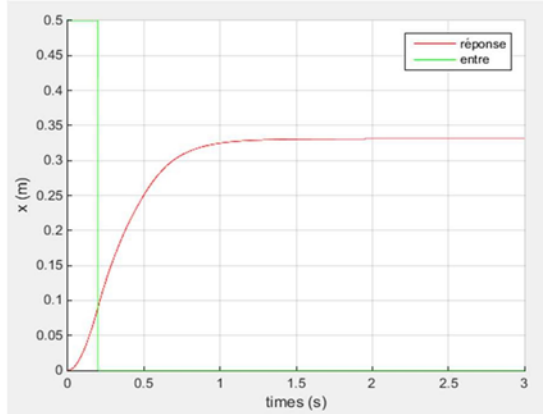


Figure3.7.3: Robot impulse response, $x(m)$ profile.

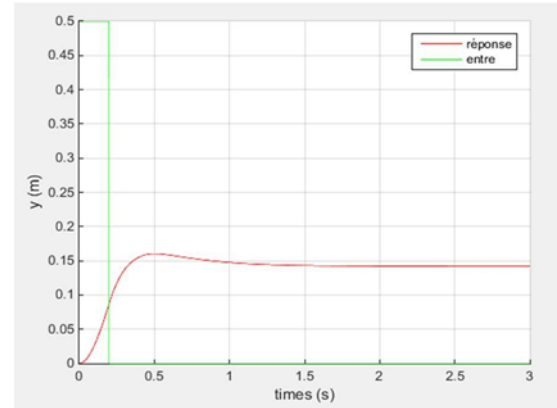


Figure3.7.4: Robot impulse response, $y(m)$ profile.

3.7.3.3 Sinusoidal Response

The simulation of the robot's sinusoidal response with a system input of: ($u_1 = \sin(t)$ and $u_2 = \sin(t)$)

For a very short duration, Figures (3.7.5) and (3.7.6) represent the open-loop position profiles of the robot:

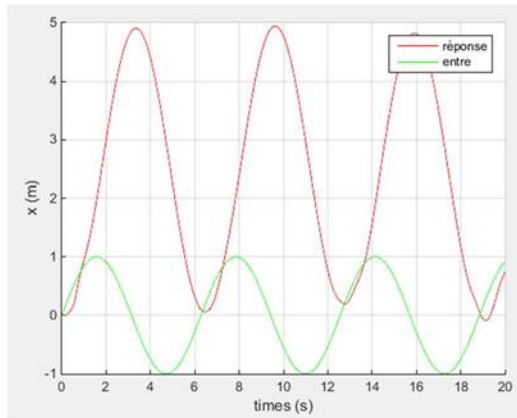


Figure 3.7.5: Sinusoidal response of the robot, $x(m)$ profile.

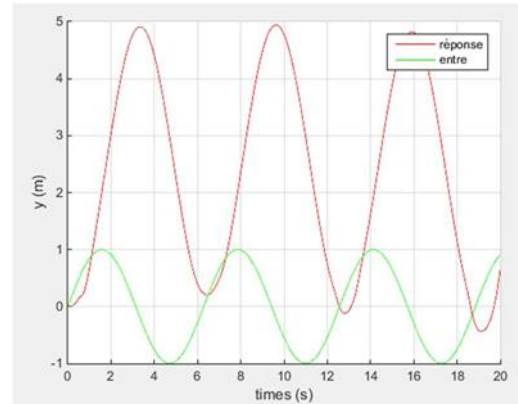


Figure3.7.6: Sinusoidal response of the robot, $y(m)$ profile.

Based on the simulation of different responses of the 4-cable planar robot, we conclude that the system is unstable, and that it is a system with a double integrator.

3.7.4 Static Force Analysis

The workspace in which all cables are under positive tension while exerting all possible Cartesian forces is called the static workspace. Static modeling and the attempt to maintain positive cable tension are presented in this section. We use a simple method to determine

the extent of the static workspace, that is, the workspace in which all possible forces can be applied with positive cable tensions.

For static equilibrium, the sum of the forces exerted on the displacement point by the cables must equal the resultant external force exerted on the environment [19].

Figure (3.7.7) shows the static free-body diagram for the 4-cable planar robot. The static equations are:

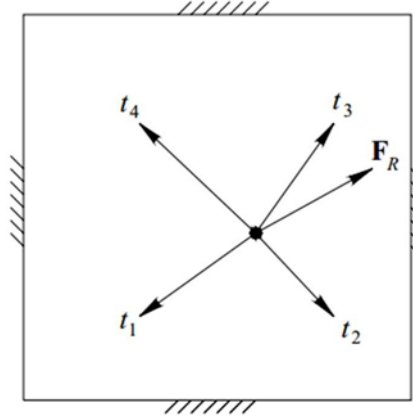


Figure3.7.7: Static force diagram [19].

$$\sum_{i=1}^4 t_i = - \sum_{i=1}^4 t_i L_i = F_R \quad (3.7.14)$$

Because the force of gravity is eliminated, the equation for the resulting force can be written as follows:

$$F_R = S * T \quad (3.7.15)$$

Such as :

$$L_i = \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} EtS = (-L_1 \quad -L_2 \quad -L_3 \quad -L_4) \quad (3.7.16)$$

SO

$$S = \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix} \quad (3.7.17)$$

So we can write:

$$F_R = \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix}^T T \quad (3.7.18)$$

With :

$$T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} \text{ and } F_R = \begin{pmatrix} F_{Rx} \\ F_{Ry} \end{pmatrix} \quad (3.7.19)$$

SO.

$$\begin{pmatrix} -\cos(\theta_1) - \cos(\theta_2) - \cos(\theta_3) - \cos(\theta_4) \\ -\sin(\theta_1) - \sin(\theta_2) - \sin(\theta_3) - \sin(\theta_4) \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} F_{Rx} \\ F_{Ry} \end{pmatrix} \quad (3.7.20)$$

For 4-cable planar robots with actuation redundancy, the force is constrained, meaning there are infinite solutions to the cable tension vector T to exert the given Cartesian force F_R . To invert equation (3.7.15) (Solution of the required cable tensions T given by the desired force FR), we adapt the homogeneous particular solution [19]:

$$T = S^+ F_R + (In - S^+ S)Z \quad (3.7.21)$$

With :

In : Is the identity matrix ($4 * 4$).

Z : is an arbitrary vector of dimension 4.

S^+ : is the Moore-Penrose pseudo-inverse of dimension ($4 * 2$).

$$S^+ = S^T (SS^T)^{-1} \quad (3.7.22)$$

The first term in (3.21) is the particular solution ($S^+ F_R$) for achieving the desired force, and the second term is the homogeneous solution ($(In - S^+ S)Z$).

For cable robots with two or more degrees of actuation redundancy (the 4-cable planar case), determining the static workspace and the method for maintaining positive cable tensions is more complex [19].

For second-degree actuation redundancy, an equivalent expression for (3.7.21) is:

$$T = \begin{pmatrix} t_{p1} \\ t_{p2} \\ t_{p3} \\ t_{p4} \end{pmatrix} + a \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} + b \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \quad (3.7.23)$$

Where the particular solution ($S+FR$) is again the first term in (3.7.23) and the homogeneous solution is expressed as the two kernel vectors of S ($N = (n_1 \ n_2 \ n_3 \ n_4)^T$ and $P = (p_1 \ p_2 \ p_3 \ p_4)^T$) multiplied by arbitrary scalars a and b .

The condition for a planar cable robot configuration to lie in the static workspace is:

$$a \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} + b \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} > \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.7.24)$$

Divide the workspace into four sectors as in Figure (3.7.7); we can construct a different null space basis for each sector. This is necessary to demonstrate that the entire workspace lies within the static workspace [20].

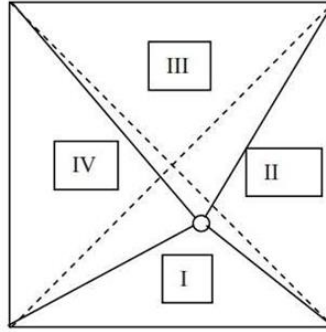


Figure 3.7.8: Sectors of the workspace of the 4-cable robot [20].

Case I: Suppose the endpoint of the effector is in the first sector. A possible basis for the kernel of S is:

$$N = \begin{pmatrix} \frac{\sin(\theta_4 - \theta_2)}{\sin(\theta_2 - \theta_1)} \\ \frac{\sin(\theta_1 - \theta_4)}{\sin(\theta_2 - \theta_1)} \\ 0 \\ 1 \end{pmatrix} \quad P = \begin{pmatrix} \frac{\sin(\theta_3 - \theta_2)}{\sin(\theta_2 - \theta_1)} \\ \frac{\sin(\theta_1 - \theta_3)}{\sin(\theta_2 - \theta_1)} \\ 1 \\ 0 \end{pmatrix} \quad (3.7.25)$$

If the effector is located in zone I, the permissible cable angle ranges are:

$$0^\circ \leq \theta_1 \leq 45^\circ \quad 135^\circ \leq \theta_2 \leq 180^\circ \quad 225^\circ \leq \theta_3 \leq 270^\circ \quad 270^\circ \leq \theta_4 \leq 315^\circ$$

The possible delta angle ranges are:

$$\begin{aligned} 90^\circ \leq \theta_4 - \theta_2 \leq 135^\circ \quad 90^\circ \leq \theta_2 - \theta_1 \leq 180^\circ \quad -270^\circ \leq \theta_1 - \theta_3 \leq -180^\circ \quad -315^\circ \leq \theta_1 - \theta_4 \leq -225^\circ \\ 45^\circ \leq \theta_3 - \theta_2 \leq 135^\circ \end{aligned} \quad (3.7.26)$$

Therefore, all sinusoidal functions in (3.7.25) are positive or zero, and any combination of N and P (with positive coefficients a and b) always has positive components as required in (3.7.24). In conclusion, the first sector belongs to the static workspace.

Case II: Suppose the endpoint is in the second sector. A possible basis for the kernel of S is:

$$N = \begin{pmatrix} 0 \\ \frac{\sin(\theta_4 - \theta_3)}{\sin(\theta_3 - \theta_2)} \\ \frac{\sin(\theta_2 - \theta_4)}{\sin(\theta_3 - \theta_2)} \\ 1 \end{pmatrix} \quad P = \begin{pmatrix} 1 \\ \frac{\sin(\theta_1 - \theta_3)}{\sin(\theta_3 - \theta_2)} \\ \frac{\sin(\theta_2 - \theta_1)}{\sin(\theta_3 - \theta_2)} \\ 0 \end{pmatrix} \quad (3.7.27)$$

If the effector is located in sector II, the permissible angle ranges of the cable are:

$$0^\circ \leq \theta_1 \leq 45^\circ \quad 90^\circ \leq \theta_2 \leq 135^\circ \quad 225^\circ \leq \theta_3 \leq 270^\circ \quad 315^\circ \leq \theta_4 \leq 360^\circ \quad (3.7.28)$$

The possible delta angle ranges are:

$$45^\circ \leq \theta_4 - \theta_3 \leq 135^\circ \quad -225^\circ \leq \theta_2 - \theta_4 \leq -180^\circ \quad -270^\circ \leq \theta_1 - \theta_3 \leq -180^\circ \quad (3.7.29)$$

$$45^\circ \leq \theta_2 - \theta_1 \leq 135^\circ \quad 90^\circ \leq \theta_3 - \theta_2 \leq 180^\circ \quad (3.7.30)$$

Therefore, all sinusoidal functions in (3.7.27) are positive or zero, and any combination of N and P (with positive coefficients a and b) always has positive components, as required in (3.7.24). In conclusion, the second sector belongs to the static workspace.

The last two cases are similar to the first two, so we can choose a suitable basis for the kernel of S for sector III:

$$N = \begin{pmatrix} 0 \\ 1 \\ \frac{\sin(\theta_2 - \theta_4)}{\sin(\theta_4 - \theta_3)} \\ \frac{\sin(\theta_3 - \theta_2)}{\sin(\theta_4 - \theta_3)} \end{pmatrix} \quad P = \begin{pmatrix} 1 \\ 0 \\ \frac{\sin(\theta_1 - \theta_4)}{\sin(\theta_4 - \theta_3)} \\ \frac{\sin(\theta_3 - \theta_1)}{\sin(\theta_4 - \theta_3)} \end{pmatrix} \quad (3.7.31)$$

And for sector IV:

$$N = \begin{pmatrix} \frac{\sin(\theta_3 - \theta_4)}{\sin(\theta_4 - \theta_1)} \\ 0 \\ 1 \\ \frac{\sin(\theta_1 - \theta_3)}{\sin(\theta_4 - \theta_1)} \end{pmatrix} \quad P = \begin{pmatrix} \frac{\sin(\theta_2 - \theta_4)}{\sin(\theta_4 - \theta_1)} \\ 1 \\ 0 \\ \frac{\sin(\theta_1 - \theta_2)}{\sin(\theta_4 - \theta_1)} \end{pmatrix} \quad (3.7.32)$$

The conclusion for each case is identical, namely that the third and fourth sectors also belong to the static workspace, including all internal triangular edges. The only point we did not consider is the center of the square, but in this case, the kernel has only one vector $N = (1 \ 1 \ 1 \ 1)^T$ because

the rank of S degenerates to a single vector. Clearly, this special case lies within the static workspace since it easily satisfies (3.7.24).

Therefore, we have shown that the fully allowed kinematic workspace of the base square is also the static workspace. The boundary of the base square (which corresponds to kinematic singularities) and all points outside the base square are outside the static workspace.

3.7.5 Calculation of Optimized Actuator Torques

This section presents a method for determining the optimal actuation torques for the controller architecture of a 4-cable planar robot with two degrees of actuation redundancy. The work also includes an algorithm for estimating the minimum actuation torques required to maintain positive cable tension.

For the control, we need to calculate the torques of the real actuators τ_i given the virtual forces F_v . The solution of the system under constraints $F_v = S(X)\tau$ is similar to (3.7.23) of section (3.7.4), the difference compared to (3.7.23) is that we can no longer make the pseudo-static assumption $\tau_i = r t_i$ (τ_i is the i th torque of the actuator, r is the radius of the cable pulley and t_i is the i th tension of the cable), due to the dynamics, this assumption no longer holds; We must calculate the actuation torques required for the control, while trying to dynamically maintain the positive tensions of the cables [21].

$$\tau = \begin{pmatrix} t_{p1} \\ t_{p2} \\ t_{p3} \\ t_{p4} \end{pmatrix} + a \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} + b \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \quad (3.7.33)$$

The particular solution $S^+ F_v$ is the first term in (3.7.33), and the homogeneous solution is expressed as the two kernel vectors of S ($N = (n_1 \ n_2 \ n_3 \ n_4)^T$ and $P = (p_1 \ p_2 \ p_3 \ p_4)^T$) multiplied by arbitrary scalars a and b .

The goal of the torque calculation is to find an optimal solution τ_{opt} with the following characteristics:

1. Each component of τ_{opt} must be greater than or equal to a specified minimum torque τ_{min} .
2. The norm of τ must be minimized.

The first condition ensures that the torque in each motor is always positive, in fact always greater than τ_{min} . If we do not consider dynamic effects, the voltage

The tension in each cable is found to be greater than $r\tau_i$. Under this pseudo-static condition, if τ_{min} is sufficiently high, the cables will never get wet.

Unfortunately, when high speed is used, due to dynamic effects, one or more cables may be slack despite a positive τ_{min} . In this dynamic case, the minimum value for τ_{min} must be estimated online for each cable in real time.

The online cable tension estimation algorithm is based on forcing each tension component to be positive at all times in the dynamic model.

$$T_i = \frac{1}{r} (\tau_i - J_i \beta'' i - C_i \beta' i) \geq 0 \quad (3.7.34)$$

The estimated torque solution for each actuator of (3.34) to maintain positive cable tension is:

$$\tau_{min_i} = \max\{J_i \beta'' i + C_i \beta' i; 0\} \quad (3.7.35)$$

The reason for the maximum function in (3.7.35) is that when dynamics are taken into account, the minimum torque required to guarantee the risk of cable strain may be negative for one or more components. In (3.7.35), we force all torque components to be at least zero. For the optimal torque solution, the norm is defined as the simple sum because no component can be negative:

$$\|\tau(a, b)\| = \sum_{i=1}^4 \tau_i \quad (3.7.36)$$

The solution form (3.33) gives a system of non-homogeneous linear inequalities in the scalar unknowns a and b .

$$\tau_{pi} + an_i + bp_i \geq \tau_{min_i} \quad (3.7.37)$$

Given this form, we have a typical problem in linear programming, where the objective function (3.7.36), the torque norm, must be minimized by a definite inequality constraint given by (3.7.37).

In the solution space $\langle a, b \rangle$, each equation of (3.7.37) represents a semiplane whose boundary is given by the line:

$$\tau_{pi} + an_i + bp_i = \tau_{min_i} \quad (3.7.38)$$

When the n lines intersect, we obtain up to ni points of intersection $\langle a_l, b_l \rangle$,

$l = 1, \dots, ni$ we denote this set of points by Θ Note that the

maximum number of intersections is $ni = \sum_{j=0}^3 j$ (in our case $ni = 6$). We can prove that

the optimal solution is one of these intersections $\langle a, b \rangle$

This is $\tau_{opt} \in \{\tau(a_l, b_l), l = 1, \dots, ni\}$. The subset of points in Θ that satisfies the minimum torque requirement (36) is given by:

$$\Theta_{pos} = \{(a, b) \in \Theta | \tau(a, b)_i \geq \tau_{min_i} \forall i = 1, \dots, 4\} \quad (3.7.39)$$

We choose the $(a, b) \in \Theta_{pos}$ that gives the minimum norm of the pair. In conclusion, we choose the points of intersection that satisfy:

1. All pairs greater than or equal to τ_{min_i} (construct the subset Θ_{pos}).
2. The smallest norm of the pair vectors ($\|\tau(a, b)\| = \sum_{i=1}^4 \tau_i$ is the smallest).

3.7.6 Implementation of the torque optimization algorithm in MATLAB

To validate this actuator torque optimization algorithm, we use MATLAB. Figure (3.7.8) shows the flowchart of the algorithm's execution to calculate the optimal torques for a virtual force F_V applied to the end effector and with the coordinates of the end effector point (x, y) :

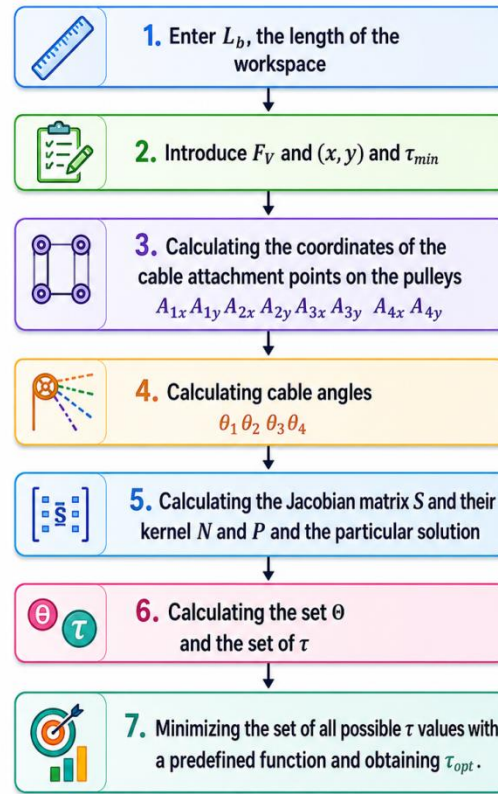


Figure 3.7.9: Flowchart for calculating optimal pairs.

The function that minimizes the set of possible τ is defined by the following two conditions:

1. All pairs greater than or equal to τ_{min_i} .
2. The smallest norm of the paired vectors.

Example: We revisit the example given in “Robert L. Williams II 2003 Ohio University” to compare it with our results. The calculation results are

$$F_V = (-1.3 \text{ N}; 1.05 \text{ N})^T (x; y) = (0.04; -0.23) m \tau_{min} = (0.1 \text{ N} 0.1 \text{ N} 0.1 \text{ N} 0.1 \text{ N})^T$$

SO :

$$S(X) = \begin{pmatrix} -0.9658 & 0.9460 & 0.4592 & -0.5509 \\ -0.2591 & -0.3241 & 0.8883 & 0.8346 \end{pmatrix} N = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} = \begin{pmatrix} 0.7339 \\ 0.3859 \\ 0.5278 \\ -0.1841 \end{pmatrix}$$

$$P = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} 0.0853 \\ 0.6492 \\ -0.3618 \\ 0.6636 \end{pmatrix} \tau_p = \begin{pmatrix} \tau_{p1} \\ \tau_{p2} \\ \tau_{p3} \\ \tau_{p4} \end{pmatrix} = \begin{pmatrix} 0.3541 \\ -0.6930 \\ 0.2899 \\ 0.7904 \end{pmatrix}$$

$$\Theta = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{pmatrix} = \begin{pmatrix} -0.5294 & -0.3482 & -0.2183 & 0.3393 & 2.5947 & -1.3247 \\ 1.5334 & 0.0168 & -1.1009 & 1.0199 & -0.3206 & -1.4078 \end{pmatrix}$$

After minimization, we find the vector of optimal torques τ_{opt} of the actuators:

$$\tau_{opt} = \begin{pmatrix} \tau_{opt1} \\ \tau_{opt2} \\ \tau_{opt3} \\ \tau_{opt4} \end{pmatrix} = \begin{pmatrix} 0.6902 \\ 0.1000 \\ 0.1000 \\ 1.4048 \end{pmatrix} \text{N}$$

We note that we obtained the same final result for the optimal actuator torques as in reference [21], but a difference in the intermediate calculation results for the various quantities; therefore, we can say that the actuator torque optimization algorithm is valid.

3.7.6.1 Application of the optimization algorithm to the robot

In this section, we attempt to visualize the optimal actuator torques (for a given minimum torque of 0.05 N), the optimal cable tensions, and the virtual generalized Cartesian forces. For this simulation, we use the previous algorithm, but we generate a circular trajectory and calculate the virtual generalized Cartesian forces by:

$$F_V = M(X_r) \ddot{r} + N(X_r, \dot{r}) \dot{r} \quad (3.7.40)$$

For the circular trajectory, we attempt to generate it using the following equations:

$$\begin{cases} x_r = R \cos(\varphi) \\ y_r = R \sin(\varphi) \end{cases} \quad (3.7.41)$$

Such as :

$$\varphi = 2\pi \left[10 \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5 \right] \quad (3.7.42)$$

For this simulation example, we attempt to generate a circle centered at O(0,0) with a radius

$R = 0.2165$ m and a time $t_f = 1$ s.

With the following conditions:

- The inertias of the rotor and pulley of each motor: $J_i = 0.0008 \text{ kg}\cdot\text{m}^2$ $i = 1:4$.

- The viscous damping coefficients of each motor shaft:

$C_i = 0.001 \text{ N}\cdot\text{m}\cdot\text{s}$, $i = 1:4$.

- The mass of the end effector: $m = 1 \text{ kg}$.

- The radius of the pulley of each motor $r_i = 0.05 \text{ m}$ $i = 1:4$

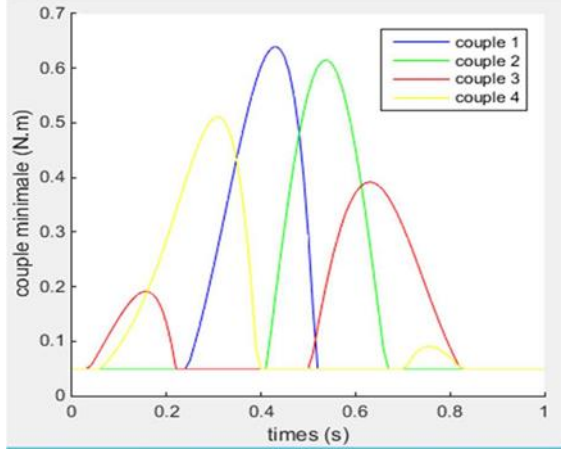


Figure 3.7.10: Minimum torques of the actuators.

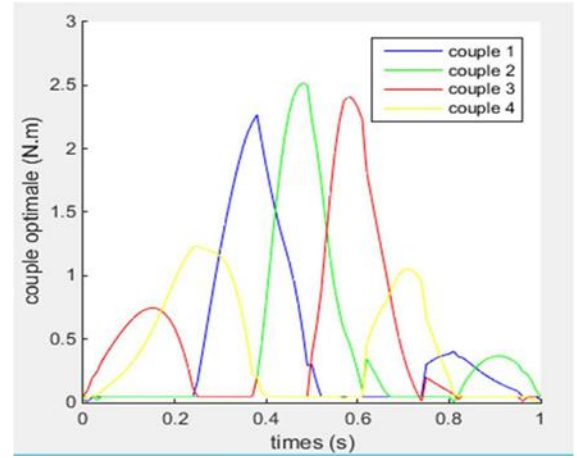


Figure 3.7.11: Optimal actuator torques.

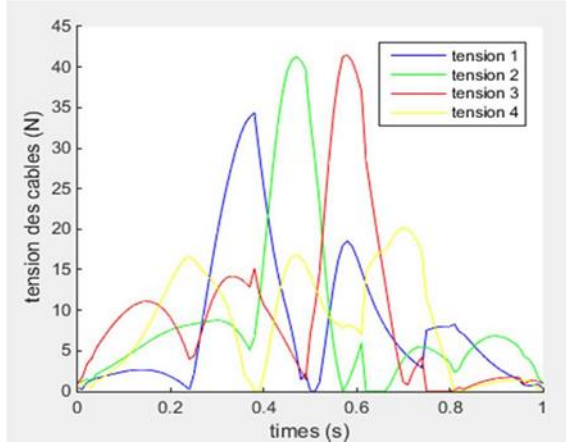


Figure 3.7.12: Optimal cable tension

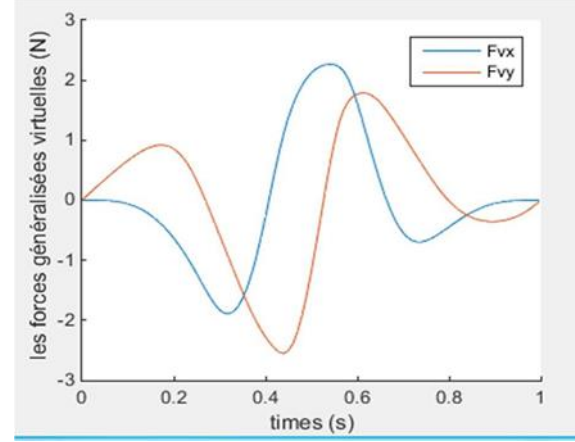


Figure 3.7.13: Virtual generalized Cartesian forces.

Comment:

During the motion, the four torques yield negative values in the calculation, but these are limited to a positive value, which is the minimum torque. The actual associated cable tensions resulting from the simulated motion as a function of the dynamic model are shown in Figure (3.7.11). For this simulated motion, the x and y components of the virtual generalized Cartesian forces are shown in Figure (3.7.12); these are not limited to being positive.

Without the dynamic minimum tension estimation algorithm, each convertible tension becomes negative and is therefore released at different times during the simulated motion. Obviously, this is unacceptable because control would be lost within these motion ranges. Thus, the constant minimum tension specification is only suitable for pseudo-static motion, not for dynamic motion with high velocities and accelerations.

3.7.7 Generating a trajectory for the robot in open loop

After validating the robot's state representation and simulating the different robot responses, and after calculating the optimal actuator torques, this section attempts to control the robot in open loop and generate the same circular trajectory used previously, as well as other trajectories. The circular trajectory is generated using the following equations: Figure (3.13) shows the flowchart of the simulation using MATLAB.

$$\begin{cases} x_r = R \cos(\varphi) \\ y_r = R \sin(\varphi) \end{cases} \quad (3.7.43)$$

Such as :

$$\varphi = 2\pi \left[10 \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5 \right] \quad (3.7.44)$$

For this simulation example we try to generate a circle with center O(0,0) and radius

R = 0.2165 m and during a $t_f = 1$ s.

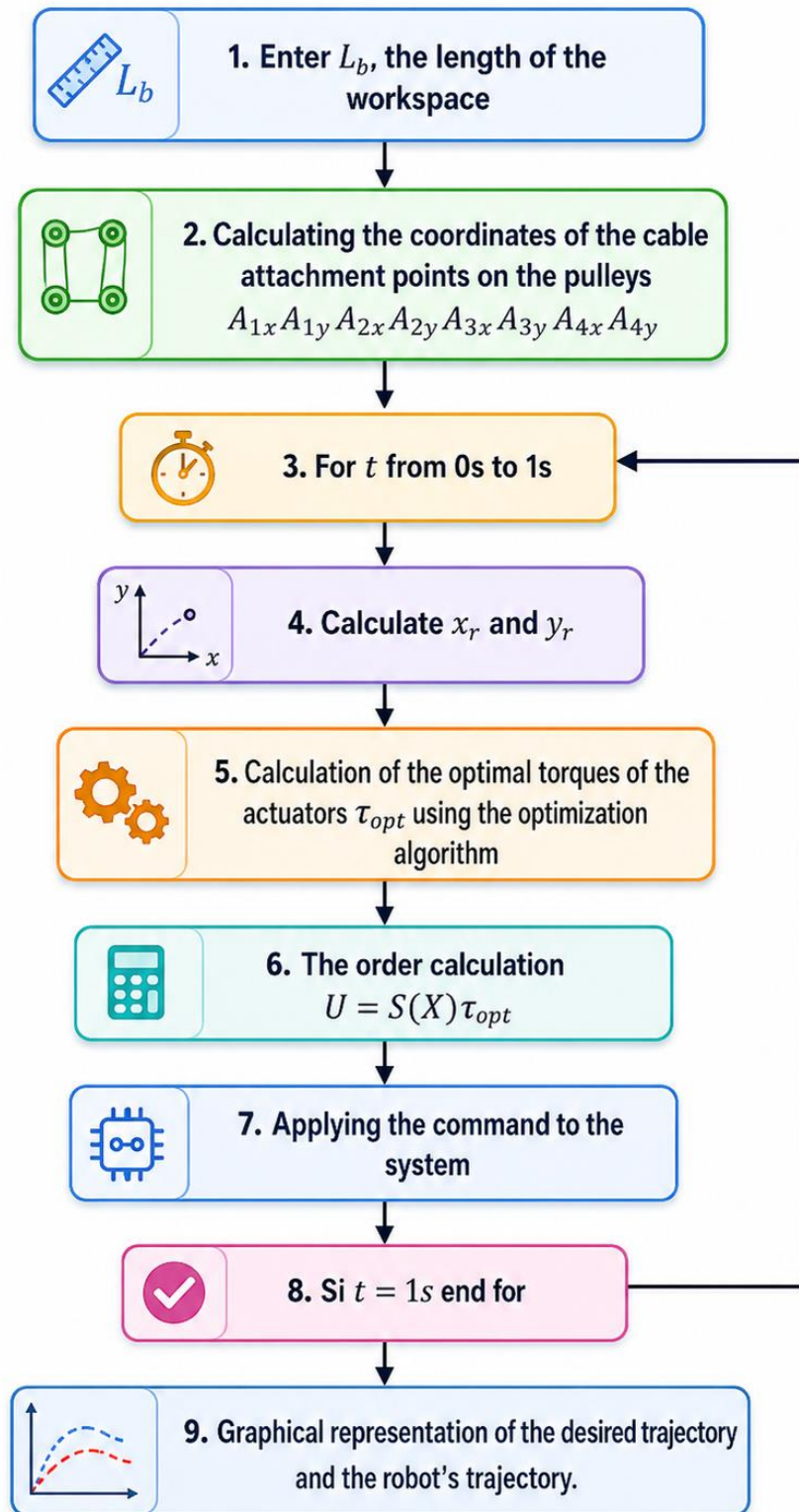


Figure3.7.14: Flowchart of the simulation of the planar robot with 4 cables in BO.

The robot's response is given in figures (3.7.15) and (3.7.16):

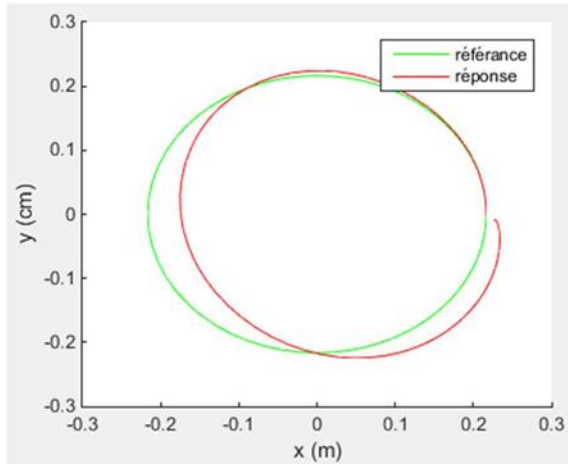


Figure3.7.15: Response of the robot with 4 cables to a circle for 1s

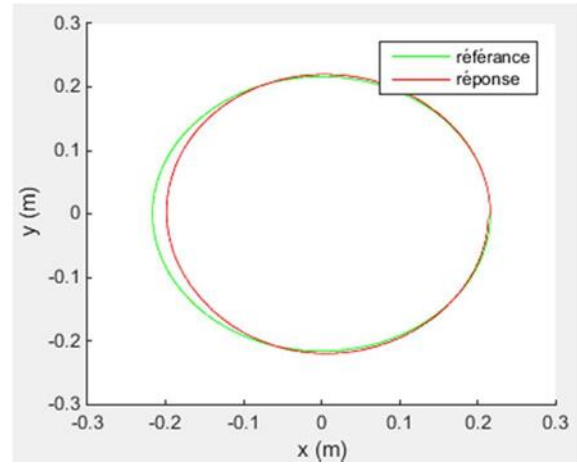


Figure3.7.16: Response of the robot with 4 cables to a circle for 5 seconds.

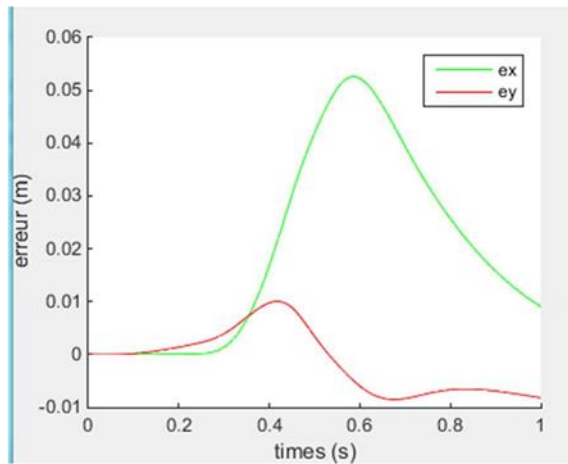


Figure 3.7.17: Static error for the circle during 1 s.

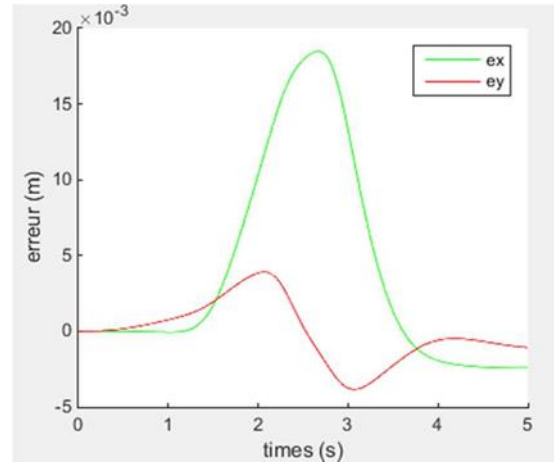


Figure3.7.18: Static error for the circle during 5 s.

We can also add other trajectories, and we define these trajectories as follows:

1. Lissajous trajectory: the parametric equations of the circle are:

$$\begin{cases} x_r = R \cos(2\varphi) \\ y_r = R \sin(\varphi) \end{cases}$$

Tel que : $R = 0.2 \text{ m}$ $\varphi = D \left[10 \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5 \right]$ $D = 2\pi$

For t_f we consider two cases: $t_{f1} = 1\text{s}$ $t_{f2} = 5\text{s}$

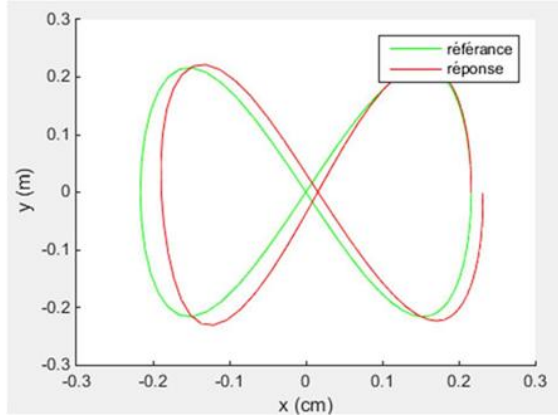


Figure3.7 .19: Response of the 4-cabl robot to a Lissajous curve for 1 s.

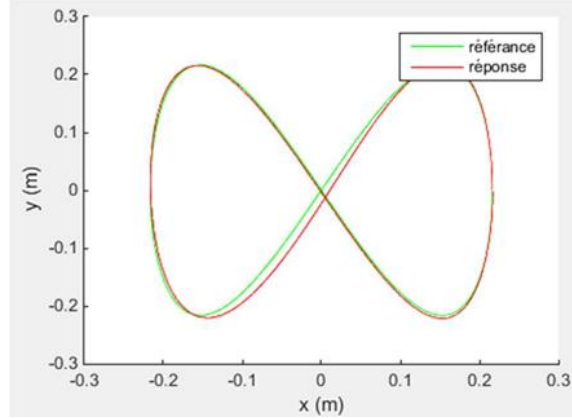


Figure37..20: Response of the 4-cable robot to a Lissajous curve for 5 s.

Comment:

Figures (3.7.15) and (3.7.16) show the robot's response over two different durations: 1 second and 5 seconds. According to the simulation, open-loop control with a circular trajectory requires a control system that ensures tracking of the reference point imposed by the trajectory. We also observe that increasing the trajectory generation time decreases the error between the reference point and the response (static error). Figures (3.7.17) and (3.7.18) illustrate this observation. Therefore, we can see that the trajectory execution time has a significant influence on the static error.

3.7.8 Conclusion

In this chapter, we validated the dynamic model of the 4-cable planar robot through the simulation of different responses (impulse, step, and sinusoidal) using MATLAB. We then calculated the optimal torques provided by the actuators using a real-time optimization algorithm to ensure positive cable tensions and prevent loss of robot control (because if the cable tensions are negative, the cables will slacken). These results justify the necessity of closed-loop control, as open-loop control contains significant steady-state errors.

3.8 Closed-loop control

3.8.1 Introduction

The control of wired robots is still an ongoing research topic. For this reason, there is very little literature on the control of these robots. However, the few studies in this area seem to mostly use the PID technique. In our case, we will explore a control technique called sliding mode.

3.8.2 Sliding Mode Control Approach

3.8.2.1 Introduction to Sliding Mode

The theory of variable structure systems (VSS) with sliding mode has been studied in detail over the last thirty years. It is based on the concept of changing the controller structure to obtain the desired response. There are several advantages to sliding mode control: high accuracy, good stability, simplicity, invariance, and robustness. This makes it particularly well-suited for systems with an imprecise model [21]. It is also relatively insensitive to certain parameter variations and disturbances.

Consequently, the VSS approach has been widely applied to the design of many practical control systems, such as servo systems, robotic manipulators, and flight control systems, etc. For these reasons, sliding mode control will be explored for the control of our system.

3.8.2.2 Basic Principle of Sliding Mode Control

In sliding mode control of variable structure systems, the state trajectory is brought towards a surface and then, using the switching law, it is forced to remain in the vicinity of this surface. This surface is called the sliding surface, and the movement that occurs along it is called the sliding motion [22].

1. The steady-state mode: this mode is added for the study of the system's response around its equilibrium point (origin of the phase plane), it is characterized by the quality and performance of the control [22].

The trajectory in the phase plane consists of three distinct parts [23]:

2. Convergence mode **CM** : This is the mode in which the system state moves from any initial point in the phase plane and tends towards the switching surface $s(x)=0$. This mode is characterized by the control law and the convergence criterion.

3. Sliding mode **SM** : This is the mode in which the state variable has reached the sliding surface and tends towards the origin of the phase plane. The dynamics of this mode are characterized by the choice of the sliding surface.

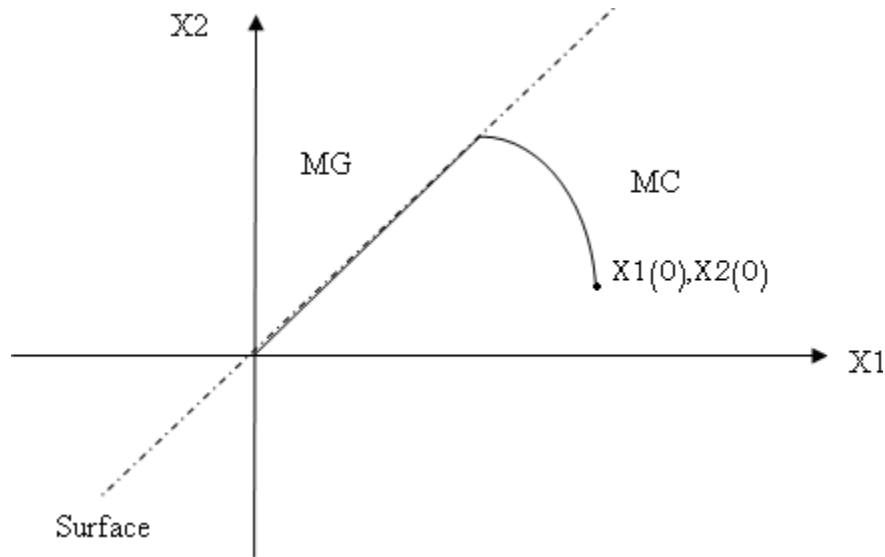


FIGURE 3.8.1 – Different modes for the trajectory in the phase plane [23]

The operation of a variable structure system is characterized by the following characteristics:

- Since the system's equilibrium state is the origin of the phase plane, the system's behavior in sliding mode is the same as its behavior in transient mode.
- During sliding mode, the system dynamics are determined solely by the parameters describing the switching line.

It should be noted that this sliding mode is often described as ideal because it requires an infinitely high switching frequency to exist. Indeed, every control system contains imperfections such as delays and hysteresis, which impose a finite switching frequency. The state trajectory then oscillates in the vicinity of the sliding surface, a phenomenon called chattering.

3.8.2.3 Structure of Sliding Mode Control

The structure of this type of controller comprises two parts: a continuous part representing the system dynamics during the sliding mode and a discontinuous part representing the system dynamics during the convergence mode. The latter is important in nonlinear control because its role is to eliminate the effects of inaccuracies and disturbances on the model [23].

The control design can be carried out in three main, highly interdependent steps [23]:

1. surface selection;

2. establishing the conditions of existence;
3. and determining the control law.

3.8.3 Sliding Control of the 4-Cable Robot

3.8.3.1 Establishing the Sliding Control Law

In the case of the 4-cable robot, two sliding surfaces must be established, one along x and the other along y.

According to the expressions obtained in section (4.3), the state representation of our robot results in a system of coupled nonlinear equations:

$$\dot{X}(t) = f(X) + g(X) * U(t) \quad (3.8.1)$$

The sliding surface along x can be expressed in the form:

$$s_{dx} = C_{1dx} * (x_{1d}(t) - x_{1ref}) + C_{2dx} * x_{2d}(t) \quad (3.8.2)$$

Similarly, the sliding surface along y can be expressed in the form:

$$s_{dy} = C_{1dy} * (x_{3d}(t) - x_{2ref}) + C_{2dy} * x_{4d}(t) \quad (3.8.3)$$

Or :

C_{1dx} , C_{2dx} , C_{1dy} and C_{2dy} : are parameters determined by the simulation;

x_{1ref} : This is the instruction according to x;

x_{2ref} : That's the instruction according to y.

Differentiating equation (3.8.2), we obtain:

$$\dot{s}_{dx} = C_{1dx} * \dot{x}_{1d}(t) + C_{2dx} * \dot{x}_{2d}(t) \quad (3.8.4)$$

As we have already reached the end of the state representation of the 4-cable system in section (4.3), therefore:

$$\dot{x}_{1d}(t) = \dot{x}_{2d}(t)$$

And :

$$\dot{x}_{2d}(t) = \frac{\alpha_1 * x_{2d}(t) + \alpha_2 * x_{4d}(t) + M_{22} u_1(t) - M_{12} u_2(t)}{\beta}$$

Then equation (3.8.4) becomes:

$$\dot{s}_{dx} = C_{1dx} * \dot{x}_{2d}(t) + C_{2dx} * \frac{\alpha_1 * x_{2d}(t) + \alpha_2 * x_{4d}(t) + \psi_x * u_{dx}(t)}{\beta} \quad (3.8.5)$$

Or :

$$\alpha_1 = M_{12} * N_{21} - M_{22} * N_{11}$$

$$\alpha_2 = M_{12} * N_{22} - M_{22} * N_{12}$$

$$\psi_x = M_{22} * M_{12}$$

$$\beta = M_{22} * M_{11} - M_{21} * M_{12}$$

$$u_{dx}(t) = u_1(t) - u_2(t)$$

To determine the control law, we worked with a new synthesis method, namely the arrival law approach of . Its general function is written:

$$\dot{s}_{dx} = -K_{dx} * s_{dx}(t) - Q_{dx} * \text{sign}(s_{dx}) \quad (3.8.6)$$

$$\dot{s}_{dx} = -K_{dx} * [C_{1dx} * (x_{1d}(t) - x_{1ref}) + C_{2dx} * x_{2d}(t)] - Q_{dx} * \text{sign}(s_{dx}) \quad (3.8.7)$$

We compare (3.8.5) with (3.8.7) and obtain the control law:

$$u_{dx}(t) = -K_{1dx} * x_{2d}(t) - \sigma_x x_{4d}(t) - K_{2dx} * (x_{1d}(t) - x_{1ref}) - Q_x * \text{sign}(s_{dx}) \quad (3.8.8)$$

Or :

$$\bullet K_{1dx} = \frac{1}{\psi_x} \left[\alpha_1 + \frac{\beta * C_{12dx}}{C_{22dx}} + K_{dx} * \beta \right]$$

$$\bullet K_{2dx} = \frac{\beta * K_{dx} * C_{12dx}}{\psi * C_{22dx}}$$

$$\bullet \sigma_x = \frac{\alpha_1}{\beta}$$

$$\bullet Q_x = \frac{\beta * Q_{dx}}{\psi_x * C_{22dx}}$$

Differentiating equation (3.8.3) we obtain:

$$\dot{s}_{dy} = C_{1dy} * \dot{x}_{3d}(t) + C_{2dy} * \dot{x}_{4d}(t) \quad (3.8.9)$$

with

$$\dot{x}_{3d}(t) = x_{4d}(t)$$

And :

$$\dot{x}_{4d}(t) = \frac{\rho_1 * x_{2d}(t) + \rho_2 * x_{4d}(t) + M_{12} u_1(t) - M_{11} u_2(t)}{\beta}$$

Then equation (3.8.9) becomes:

$$\dot{s}_{dx} = C_{1dy} * \dot{x}_{4d}(t) + C_{2dy} * \frac{\rho_1 * x_{2d}(t) + \rho_2 * x_{4d}(t) + \psi_y * u_{dy}(t)}{(-\beta)} \quad (3.8.10)$$

$$\rho_1 = M_{22} * N_{21} - M_{21} * N_{11}$$

$$\rho_2 = M_{11} * N_{22} - M_{21} * N_{12}$$

$$\psi_x = M_{21} * M_{11}$$

$$\beta = M_{22} * M_{11} - M_{21} * M_{12}$$

$$u_{dy}(t) = u_1(t) - u_2(t)$$

We can write to each other:

$$\dot{s}_{dy} = -K_{dy} * s_{dy}(t) - Q_{dy} * \text{sign}(s_{dy}) \quad (3.8.11)$$

$$\dot{s}_{dy} = -K_{2dy} * [C_{1dy} * (x_{3d}(t) - x_{2ref}) + C_{2dy} * x_{4d}(t)] - Q_{dy} * \text{sign}(s_{dy}) \quad (3.8.12)$$

We compare (6.11) with (6.10) and obtain the control law:

$$u_{dy}(t) = -K_{1dy} * x_{4d}(t) - \sigma_y x_{2d}(t) - K_{2dy} * (x_{3d}(t) - x_{2ref}) - Q_y * \text{sign}(s_{dy}) \quad (3.8.13)$$

Or :

$$\bullet K_{1dy} = \frac{1}{\psi_y} \left[\alpha_1 + \frac{(-\beta) * C_{1dy}}{C_{2dy}} + K_{dy} * (-\beta) \right]$$

$$\bullet K_{2dy} = - \frac{\beta * K_{dy} * C_{1dy}}{\psi * C_{2dy}}$$

$$\bullet \sigma_y = \frac{\alpha_1}{\beta}$$

$$\bullet Q_y = - \frac{\beta * O_{dy}}{\psi_x * C_{2dy}}$$

When constraints are applied to the command, chatter is observed due to the sign function; in order to reduce this chatter, the sign function in equations (3.8.8) and (3.8.13) is replaced by:

$$\frac{s_d}{|s_d| + v}$$

$$S_d = S_{dx} \text{ OR } S_{dy}$$

And :

$$v = v_x \text{ OR } v_y$$

3.8.3.2 Simulation of the control with constraints in sliding mode for the 4-cable robot

In accordance with the equations governing the sliding mode applied to our 4-cable robot, a simulation program was developed using Matlab and Simulink. It should be noted that the commands are the forces applied to the final effector in the x and y directions. These forces must be converted into tension using the Moore-Penrose equation. By their very nature, for our application, these tensions must always be positive. Similarly, they must be limited between t_{min} and t_{max} so that the cables remain taut. Furthermore, the end effector must not extend beyond the workspace. The simulation of our system's operation requires the integration of the system of equations under the aforementioned constraints. This was performed using the Runge-Kutta numerical method.

The principle diagram of the sliding mode control is given by:

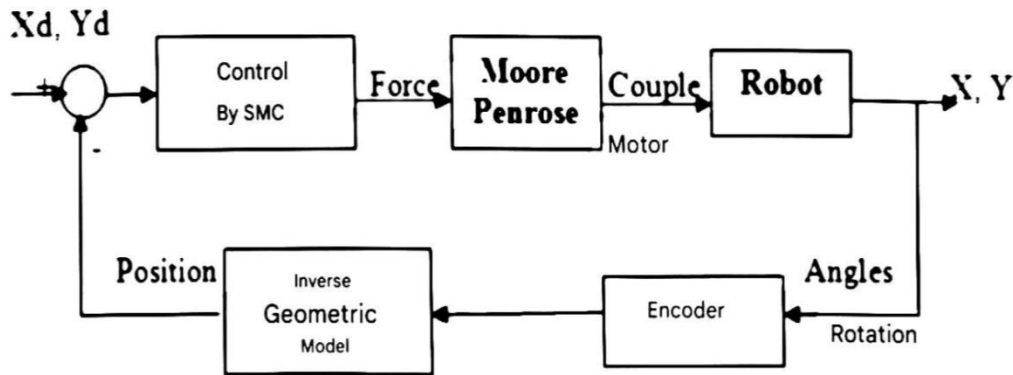


FIGURE 3.8 .2 – Representation of the SMC control law

In reality, two blocks must be added to the feedback chain to obtain the position. The first block is the encoder; its input is the system output, which is the rotation angle of the pulleys, and its output is the cable length values. The second block is for the direct geometric modeling, which gives us the position value.

The simulation diagram corresponding to the expressions for the closed-loop sliding mode control is given in the following figure:

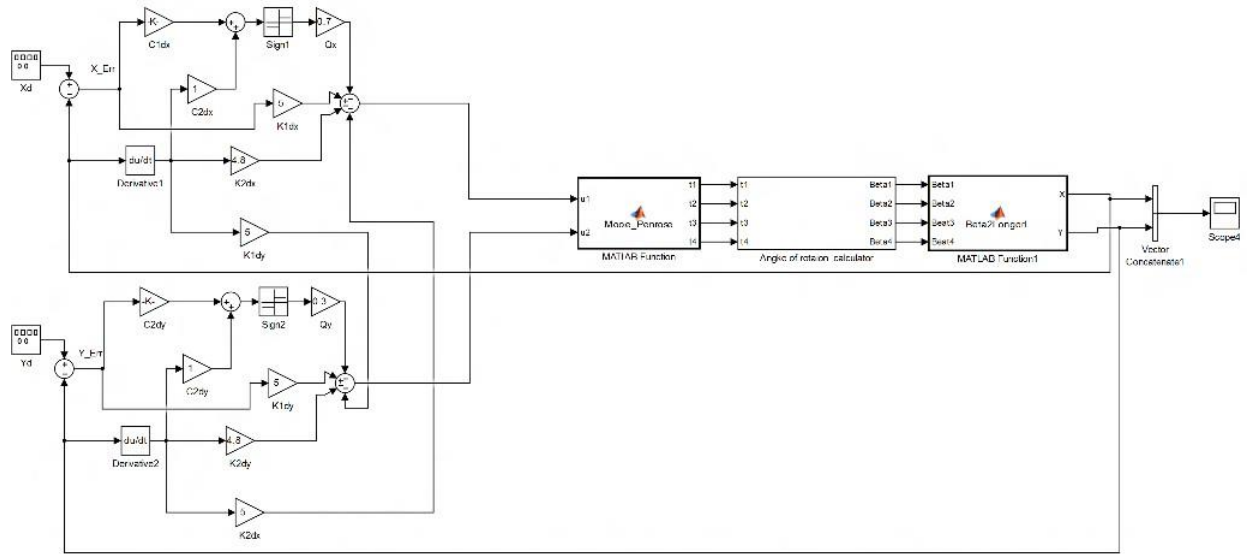


FIGURE 3.8.3 – The implementation diagram corresponding to the closed-loop control expressions by SMC according to x and y

The preceding control diagram comprises four stages. The first stage allows the input of the setpoints x_{ref} and y_{ref} . The second stage represents the SMC controller, which determines the forces F_x and F_y . These forces are defined as inputs to the third block, which determines the motor torques τ_1, τ_2, τ_3 , and τ_4 using the Moore-Penrose method. After determining the optimum torques, these are used to effectively control the positions and velocities of the final effector. To this end, the control parameters were adjusted to obtain good performance of the terminal element (stability, speed, precision).

Finally, the chosen parameters are:

$$C_{1dy} = C_{1dx} = 12.91$$

$$C_{2dx} = C_{2dy} = 1$$

$$Q_x = 0.7$$

$$Q_y = 0.3$$

3.8.3.2.1 Example of a 4-cable robot simulation

We consider our system at rest at the position:

$$x_{1d}(0) = 0 \text{ cm And } x_{3d}(0) = 0 \text{ cm.}$$

We want to transfer the terminal effector to the final position:

$$x_{1d}(0) = 1 \text{ cm And } x_{3d}(0) = 3 \text{ cm.}$$

The results of the positions along (x and y) obtained by the simulation are presented in Figures (3.8.4) and (3.8.5) respectively:

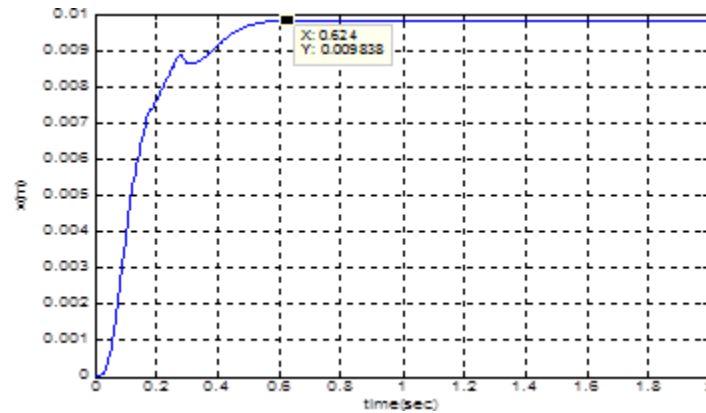


FIGURE 3.8.4 – Profile of the (x) position by sliding mode

By examining the position profile $x(t)$ represented in the figure above, we can see that the closed-loop sliding mode control does indeed send the end effector to the desired point, but with small perturbations caused by the control switching. The response time is determined by the choice of control parameters and the physical parameters of the robot; in our system, the response time is on the order of 0.6 sec.

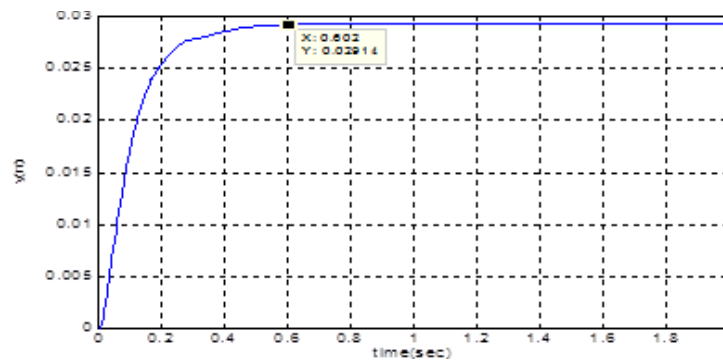


FIGURE 3.8.5 – The profile of the position (y) by the sliding mode

Similarly, by examining the position profile $y(t)$ represented in the figure below, we can see that the closed-loop sliding mode control effectively sends the terminal to the desired point. The response time is determined using the same method as before.

The velocity profiles (V_x) and (V_y) for the sliding mode are shown in Figures (3.8.6) and(3.8.7) respectively:

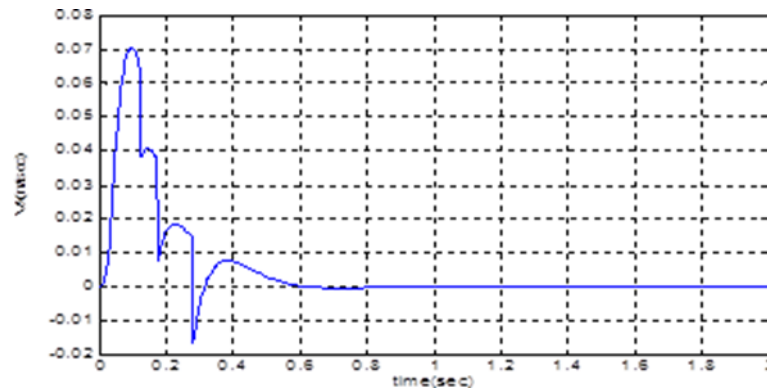


FIGURE 3.8.6 – The velocity (V_x) profile by the sliding mode

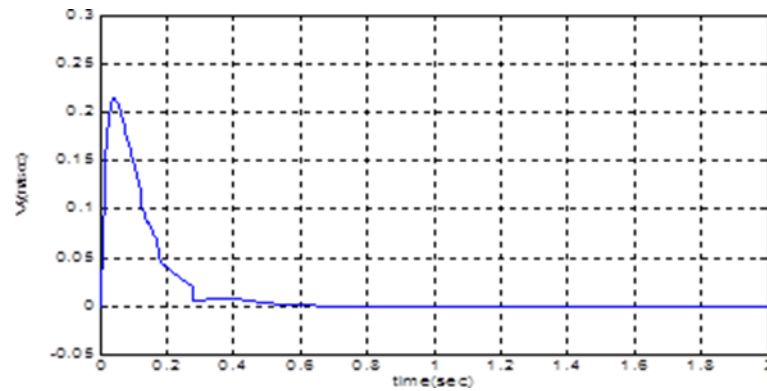


Figure 3.8.7 – Le profil de la vitesse (V_y) par le mode glissant

We observe that the velocity has an impulsive shape with a deceleration slower than the ascent. Small deformations are caused by the switching of the control.

The trajectories in the phase planes (x, V_x) and (y, V_y) for the sliding mode are shown in Figures (3.8.8) and (3.8.9) respectively:

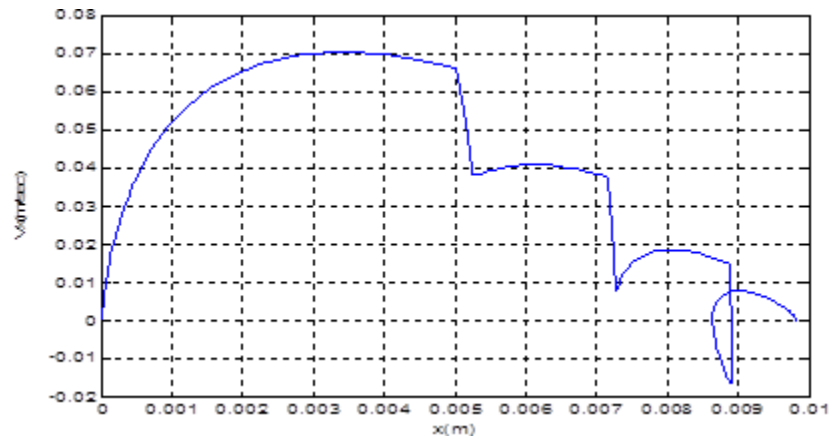


FIGURE 3.8.8 – Trajectory of the system represented in phase plane along (x)

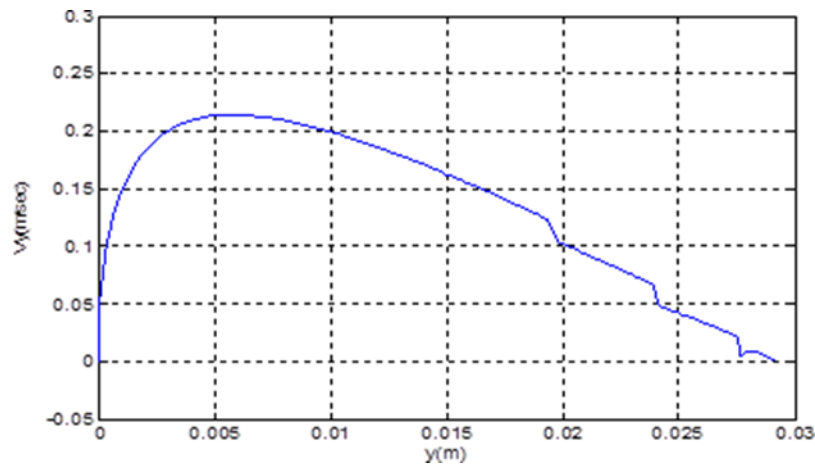


FIGURE 3.8.9 – Trajectory of the system represented in phase plane along (y)

We observe that Figure (3.8.8) shows that the chatter on x is greater than that on y , because the parameter $Q_x > Q_y$.

The control represented in the figure above allows the terminal effector to follow the trajectory shown in Figure (3.8.10):

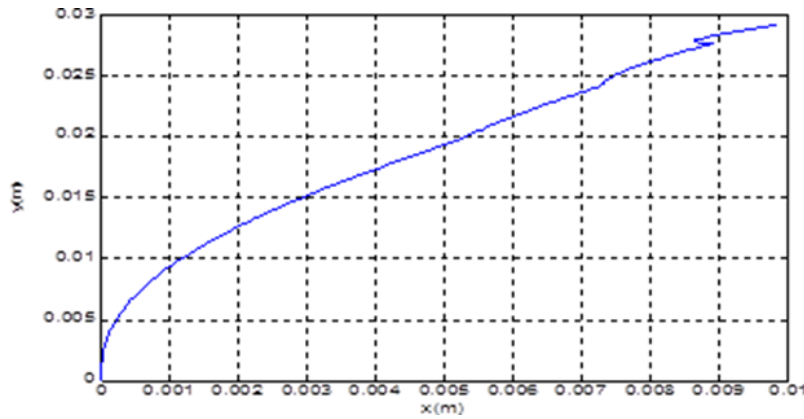


FIGURE 3.8.10 – The trajectory followed by the terminal effector

3.8.4 Conclusion

The sliding mode offers good performance (very good response time, stability and high precision) for controlling cable robots.

Chapter 4:

PRACTICAL SECTION

4.1 Introduction

The practical work consists of a hardware section describing the various physical parts, a software section covering all the software and programming, a section including suggestions for improving the robot's movement, and finally, the robot's application for upper limb rehabilitation.

4.2 Hardware Section

This section focuses on the mechanical and electrical structure of the 4-cable planar robot. First, we define the different mechanical components that form the proposed structure. Then, we define the different electrical components used to control the robot. Finally, we define the advantages and disadvantages of these components (mechanical structure and electrical components) for improving the prototype intended for upper limb rehabilitation.

4.2.1 The Base

The four-cable planar parallel robot is a fixed robot, so a smooth square surface was chosen that includes its workspace to facilitate the movement of the end effector. This base is a wooden plate; wood was chosen to avoid damaging the base. This square base is 550 mm long on each side and has four 8 mm diameter holes to house the motors and pulleys. These holes are spaced 500 mm apart.



Figure 4.1 : the base.

4.2.2 Pulley and Guides

The pulley transmits motion from the motor to the actuator via cables. Our pulley has a diameter of 70 mm.

The guides define the points of contact between the cable and the last pulley or eyelet before it attaches to the platform. We chose steel guides with a height of 5 mm. These guides limit the working area of the actuator.



Figure 4.2: Pulleys and guides.

4.2.3 The Effector

A cylindrical steel piece, 70 mm in diameter and 60 mm high, has a hole in its center to allow the pen to be inserted for viewing the movement results. Ball bearings are placed in its base to allow the pen to move smoothly within the workspace without friction.



Figure 4.3: The effector

4.2.4 Cables

Cables are the key specification of this type of robot because they replace rigid joints to provide a larger workspace and a high payload-to-mass ratio. Currently, most work on this type of robot uses a simplified cable model without mass or elasticity, as described in Figure 4.3. However, for large robots, self-mass and elasticity introduce cable deflection, causing the profile to deviate from the theoretical straight line. The conditions necessary to neglect this deflection and model the cables as straight segments have been presented. This is achieved by comparing the initial deflection of a realistic cable with the cable's direction if it were perfectly straight. This study is useful for determining the effect of cable gravity but does not consider the effect of elasticity.

If a cable connects a point A_i on the frame and a point B_i on the platform, several cable models can be defined depending on whether or not mass and elasticity are taken into account when calculating the cable length required to connect these two points.

In what follows, we will denote d_i as the distance between A_i and B_i , l_i as the cable fully extended (i.e., without load, at zero tension), and L_i as the actual cable length. We present different cable models.

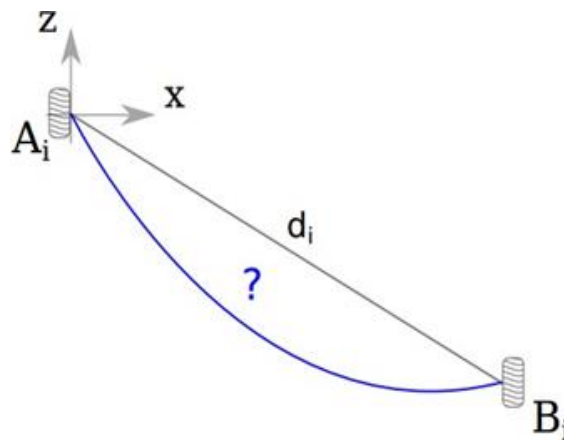


Figure 4.4: Cable modeling

4.2.4.1 Massless, Indeflexible Cable

This concept can be considered theoretical because all cables have mass and are therefore more or less elastic. This strong assumption allows for a simplified modeling of cable robots, and the error introduced by using this model is often considered negligible in the literature.

This model leads to considering the cable as a straight segment of length l_i , that is, the deployed length, connecting the exit point A_i and the attachment point B_i . We therefore write $d_i = l_i = L_i$

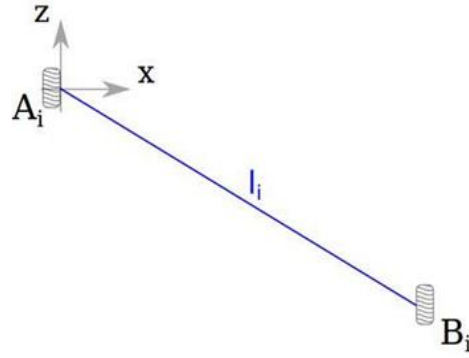


Figure 4.5: Non-deformable cable of zero mass.

4.2.4.2 Massless Elastic Cable

Cables are more or less elastic and can therefore stretch depending on the tension. It is possible to account for this elasticity while considering the cable as a straight segment, i.e., with its mass considered negligible. However, taking elasticity into account makes the cable lengths a function of the tension in each cable [24].

The relationship between the length under stress L_i and the tension in a cable is given by the equation: $\tau_i = k(L_i - l_i)$

k : Is the linear stiffness constant of the cable

This model leads to considering the cable as a straight segment of length $L_i = d_i$, a function of the robot's position and statics. This model is useful in a fully constrained configuration where tensions can be significant and also when a long cable length is subjected to stress [24]. However, for long cable lengths with non-negligible mass, or when not all cables are subjected to significant tension, this model is not sufficiently realistic.

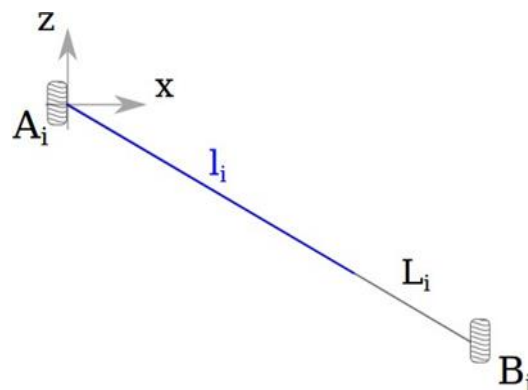


Figure 4.6: Zero-mass elastic cable

4.2.4.3 Elastic Cable with Non-Zero Mass

A well-known and one of the most realistic cable models comes from Irvine's work in civil engineering [25]. It is used in the modeling of cable structures (mainly for cable-stayed bridges), and very rarely for parallel cable robots.

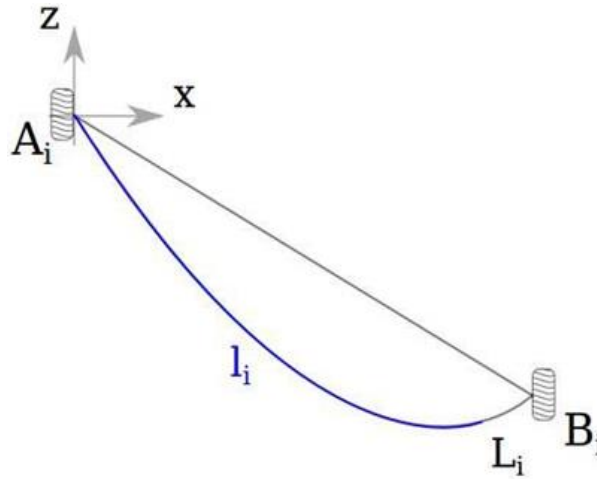


Figure 4.7: Elastic cable with non-zero mass

In our work, we used fishing line as a cable because it is closer to the first model of rigid, zero-mass cables, allowing us to simplify robot modeling.

4.2.5 Arduino Control Board

The Arduino Mega 2560 board is based on an ATmega2560 clocked at 16 MHz. It has 54 E/S pins, including 14 PWM, 16 analog, and 4 UARTs. It is ideal for [26]

applications such as robotics. Connectors located on the outer edges of the printed circuit board allow for the insertion of a range of add-on modules. It can be programmed using the Arduino software. The ATmega2560 controller includes a bootloader that allows the program to be modified without using a programmer. The software is available for free download. Main features:

- Power supply: via USB port or 7 to 12 V on the power connector.
- Microprocessor: ATmega2560.
- Flash memory: 256 kB.
- SRAM: 8 kB.
- EEPROM: 4 kB.
- 54 I/O pins, including 14 PWM pins.

- 16 10-bit analog inputs.
- Current per I/O: 40 mA.
- Clock speed: 16 MHz.
- 3 serial ports.
- I2C and SPI buses.
- Interrupt handling.

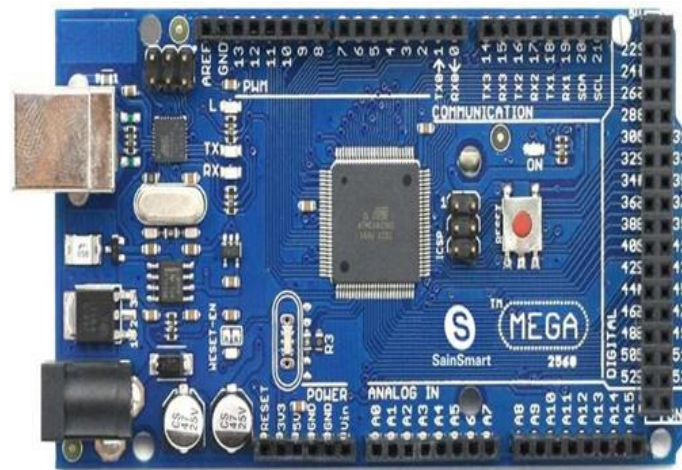


Figure 4.8: Arduino MEGA 2560 board.

4.2.6 Motor

The DC motor with 1:30 reduction gear and encoder is a compact geared motor unit consisting of a powerful DC motor, a 1:30 reduction gear enabling a rotational speed of 170 rpm (216 rpm no-load) with a torque of 1.5 kg·cm, and a high-precision encoder providing 360 pulses per revolution.

The encoder requires an input voltage between 3.5 and 20 V and consumes a maximum of 10 mA. The two outputs A and B return a square wave signal between 0 and the input voltage with a phase shift of approximately 90°. Each rising and falling edge of the two encoders indicates one revolution step. The frequency of the edges therefore indicates the rotational speed, and the order of the transitions indicates the direction[27].

Pinout diagram for 6-pin JST type PHR-6 connector:	
(1) Violet	Hall Sensor B Vout
(2) Bleu	Hall Sensor A Vout
(3) Vert	Hall Sensor GND
(4) Marron	Hall Sensor Vcc
(5) Rouge	+ Moteur
(6) Noir	- Moteur



Figure 4.9: DC motor

4.2.7 Driver Circuit

The L298 is a monolithic circuit integrated into a 15-lead Multiwatt and PowerSO20 package. It is a high-voltage, high-current, dual-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC motors, and stepper motors. Two enable inputs are provided to turn the device on or off independently of the input signals. The emitters of the lower transistors in each bridge are connected together, and the corresponding external terminal can be used to connect an external sensing resistor. An additional power supply input is provided so that the logic operates at a lower voltage.

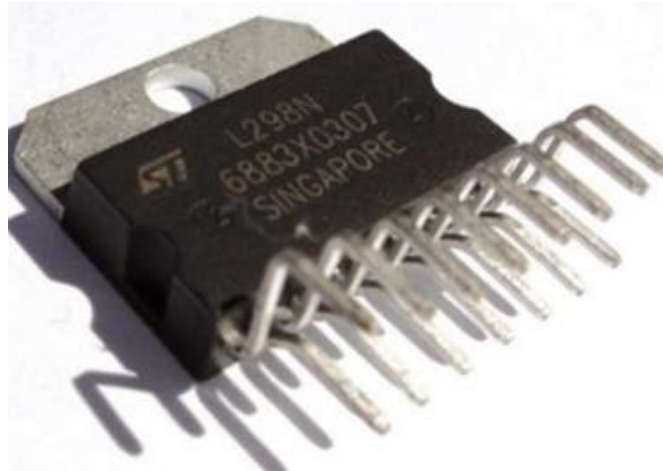


Figure 4.10 : Circuit L298N.

4.3 Software Section

In this section, we attempt to develop the control program for the 4-cable planar robot using the circuit defined in Appendix (1). After building the 4-cable planar robot and its control circuit, the implementation of the control techniques studied in Chapter 5 proved difficult. Therefore, we attempted to use PID control for motor servoing to simplify the process.

4.3.1 The Arduino IDE Interface

For programming, we used the Arduino IDE (Integrated Development Environment). This open-source environment uses the C/C++ language, which is traditionally used for programming microcontrollers and remains the most efficient solution[28].

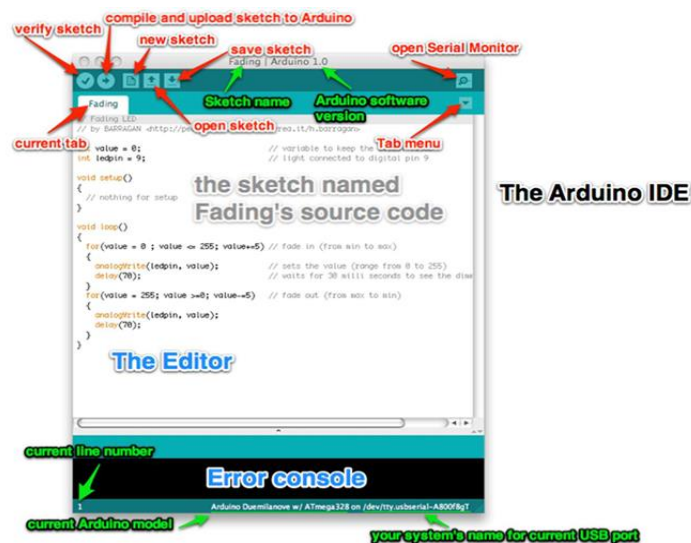


Figure 4.11: The Arduino IDE programming environment.

4.3.2 Position Control of Motors

The type of motors used (DC motor with gearbox and Hall effect encoder) which can be position controlled allows for the control of the lengths and angles of the wires to move the effector within the workspace.

Easy to implement on Arduino with a limited number of instructions, this PID controller with these characteristics is the best choice for controlling the motors[29].

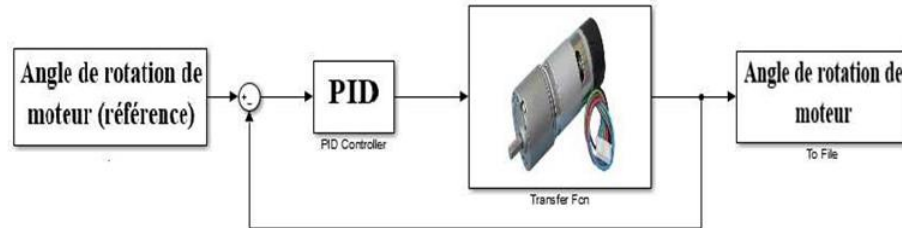


Figure 4.12: Motor position control.

4.3.3 Program Implementation

Using the Arduino interface, the control theory for a four-cable planar parallel robot was programmed to control the four motors' positions. The program takes as input the desired position or trajectory. The rotation angles of the four motors are calculated from the robot's geometric model to obtain the required cable lengths.

Initially, the four-cable planar robot was programmed to move from one point to another. The robot's general program is described by the following flowchart:

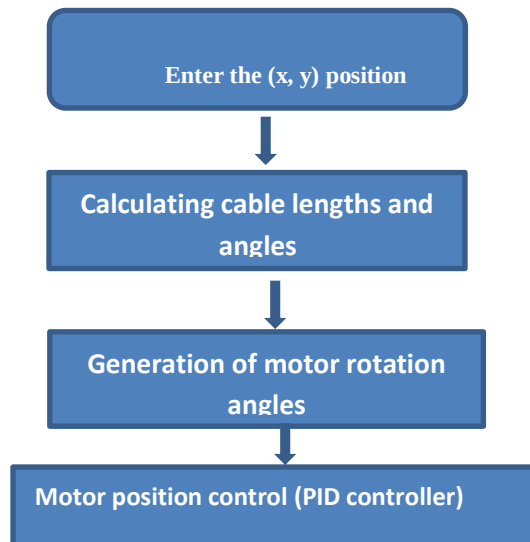


Figure 4.13: Flowchart of the robot control program.

4.4 Problems Encountered During the Practical Session

4.4.1 Motors

During the practical work, we observed that the motor resolution (1 degree) is low, given the robot's need for greater precision. Furthermore, the position control introduces motor vibration, which in turn causes system vibration. Therefore, to address this issue, we recommend using stepper motors for smoother movement and greater precision.

In addition, using the Simulink Arduino I/O package, we can utilize Arduino as an input/output interface and consolidate all programming and human-machine interface tasks within MATLAB.



Figure 4.14: Stepper motor.

Using Matlab Arduino communication is possible for controlling DC motors, but only for speed, not position, since the Simulink encoder block cannot translate the motor encoder pulses into the correct values.

4.4.2 Cables

The cables used in the robot are rigid and massless. The problem encountered in this project is that the cables do not maintain negative tension, which causes them to slip off the pulley. To ensure negative tension, a solution can be proposed: adding a spring to each cable.

4.4.3 Structure

To prevent cable failure, we propose the configuration developed by the research group in the publication "An experimental study on the vision-based control and identification of planar cable-driven parallel robots," which uses pulleys instead of guides.

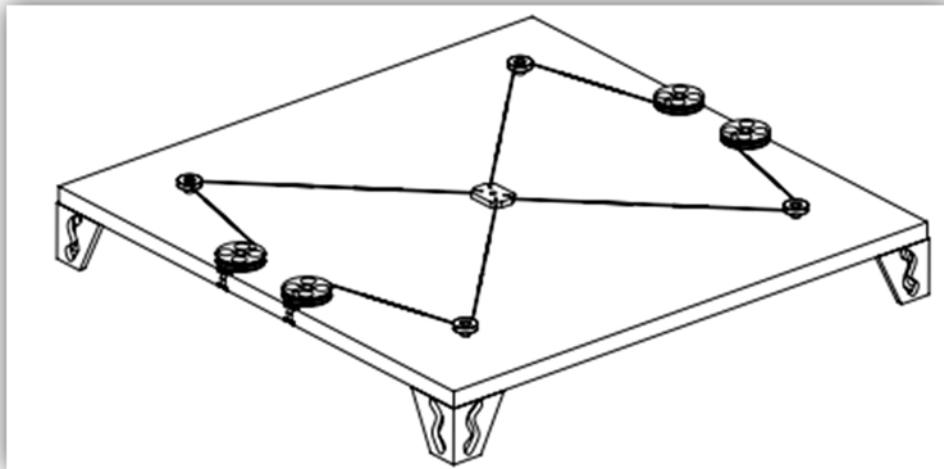


Figure 4.15: Proposed structure

4.4.4 The Pulley

Most research on this type of robot uses a specific type of pulley to prevent cables from slipping off and to reduce vibrations caused by motor control.



Figure 4.16: The pulley proposed for this type of robot.

4.5 Upper Limb Rehabilitation Using a Four-Cable Planar Robot

Abstract technology has become an important tool for rehabilitation, particularly for stroke patients. Robots can be more effective than human therapists because they can dedicate more time to therapy. The robot studied in this work has the following characteristics: a large workspace, reduced risk of injury, smooth movement, and ease of assembly and disassembly. These features make this robot an important choice for medical rehabilitation, especially for the upper limbs [30].



Figure 4.17: The application of the robot for upper limb rehabilitation

As a first prototype, we developed a simplified 4-cable planar robot. This means the robot is integrated into the desk where patients sit directly in front of it. Patients are expected to lean against the desk during therapy.

4.5.1 Upper Limb Movements

As shown in Figure (4.18), a human upper limb has seven degrees of freedom: three at the shoulder joint, two at the elbow joint, and two at the wrist joint.

1. Shoulder Internal and External Rotation 2. Shoulder Abduction and Adduction
3. Shoulder Flexion and Extension 4. Anterior Pronation and Supination
5. Elbow Flexion and Extension 6. Ulnar and Radial Deviation
7. Wrist Flexion and Extension

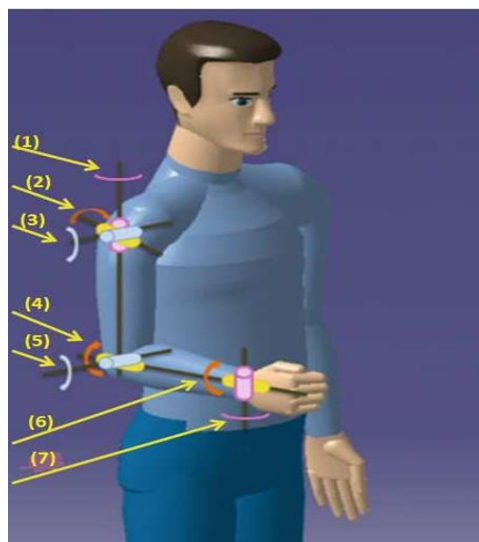


Figure 4.18: Movements of the upper limbs [30].

Activities of daily living (ADLs) for the upper limb are a combination of these basic movements. Although the fingers also have a wide range of motion, their movements are not considered in this study.

4.5.2 Pre-programmed Mode (Passive Assistant Mode)

In the first stage of rehabilitation, the system provides pre-programmed motion. Since at the beginning of rehabilitation, patients do not have enough muscle strength to move even their limbs and they experience joint pain during movement, the robot must assist their movement at a moderate speed. In this way, they can regain joint functionality in the upper limb [30]. We developed the trajectories of different shapes in planar motion. Figure (4.19) shows an example of a trajectory, the circle...the paths to follow.

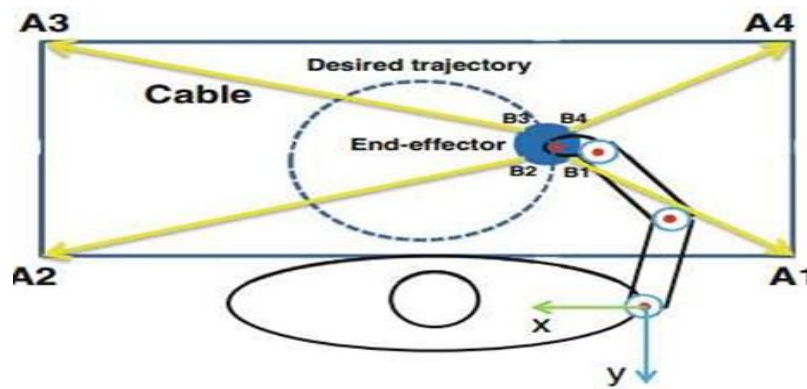


Figure 4.19: Passive assistant mode [30].

4.5.3 Active Assistant Mode

In active assistant mode, a desired path line and two guide path lines are offset from the original desired trajectory. When the patient moves the end effector along the desired path, they can move the robot freely in the admission mode control, where the force exerted by the patient is used to generate the desired robot speed. In this way, the patient moves their upper limb as the therapist. Active assistant mode can be considered as the combination of two virtual walls placed close together. This implements admission control and simultaneously provides the perception of the virtual wall. Active assistant mode can be considered as the combination of two virtual walls placed close together [30].

When the end effector makes contact with the virtual wall, the virtual contact force is considered.

In this mode, we used only one virtual wall instead of two. The virtual wall can be constructed in any shape. However, the basic geometry, for example... The rectangular line is considered because the circle can be obtained directly forward. The algorithm associated with this mode is essentially identical to that of the control strategy described in these walls, which connect the

desired work area and allow the final effector work inside the figure. In addition, the virtual wall can protect the patient from injury [30].

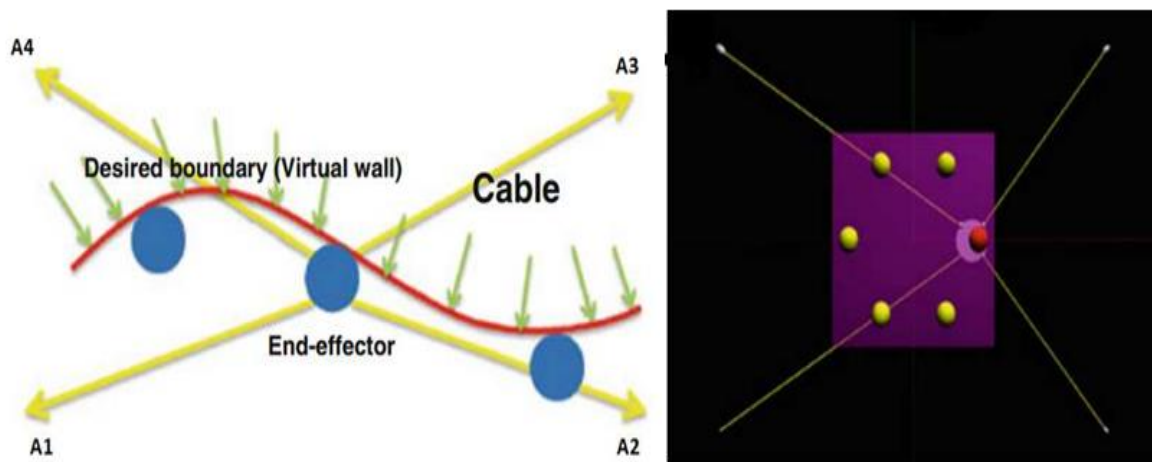


Figure 4.20: Active assistant mode [30].

4.5.4 Progressive Resistance Mode

As a final step, if patients recover well, they need to increase their muscle strength. For this reason, the progressive resistance mode is also provided, where the robotic system resists movement, so the patient must exert more force than others. As the figure shows, the patient applies force to move the final effector; the resistive reaction force is applied [30].

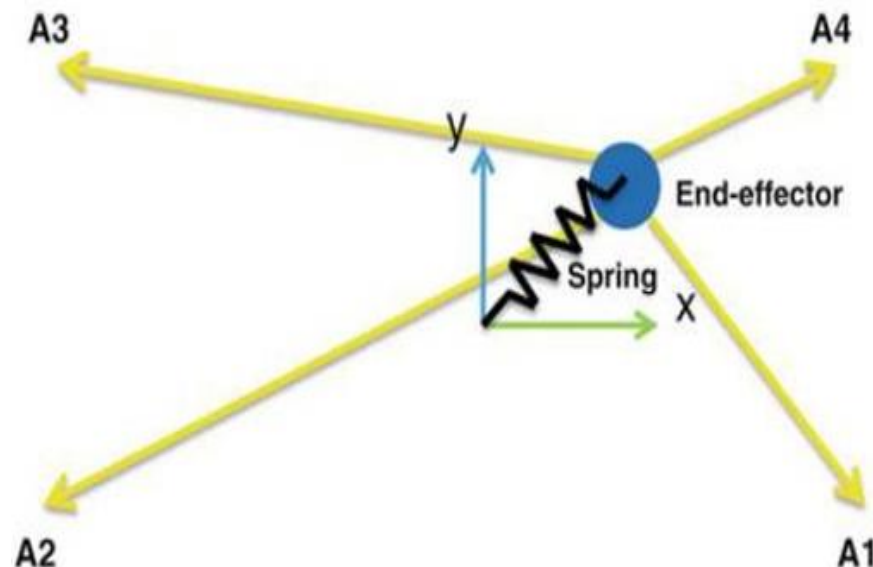


Figure 4.21: Mode of progressive resistance [30].

4.5.5 Virtual Rehabilitation Therapy Game

The effectiveness of the rehabilitation robot can be improved through virtual games. The graphical user interface (GUI) is developed to be intuitive and easy to understand, and the game is designed to provide force feedback to the player.



Figure 4.22: Billiards game for virtual rehabilitation through the robot [30].

While the patient participates in the simulation, it transparently measures and stores several parameters in the database: recording time until the end of the exercise, and errors between the ideal path and the path taken by the patient. Through various feedback modes, the patient will be encouraged and will strive to improve their results. A congratulatory message is displayed when the patient successfully completes an exercise.

4.6 Conclusion

The cable-driven planar parallel robot is among the best choices for upper limb rehabilitation. The specific design of this robot is determined using limb data from the database. To analyze the actual movement of the joints, we will need to measure the three-dimensional positions of the joints using an optical tracking system and analyze their angle variations.

In the future, more experiments will be conducted to quantify the role of upper limb joint angles more systematically. Clinical application is ultimately envisioned for intensive use of the machine with neurological patients (such as chronic post-stroke patients, hospitalized patients, and patients with brain injuries).

General Conclusion

As part of ongoing work on the 4-cable planar robot at the Process Control Laboratory (LCP), the objective of this study is the development and construction of a prototype 4-cable planar robot. This study involves the modeling, optimization, and control of the 4-cable planar robot. This prototype is intended to assist specialists in upper limb functional rehabilitation.

The study conducted led to the following conclusions:

- Although cable robots are a category of parallel robots, geometric modeling allowed us to determine the position of the end effector in the workspace by calculating the various angles of the cables with the x-axis and the different cable lengths. We developed programs in MATLAB to facilitate the calculation of these angles and lengths.
- Kinematic modeling allowed us to determine the velocity of the end effector.
- Dynamic modeling, which allows us to synthesize commands, provides the different differential equations of motion and the state representation of the planar 4-cable robot.
- To study the robot's behavior, we developed programs in MATLAB. This simulation provides the different responses of the robot (impulse, step, and sinusoidal). Based on these responses, we can conclude that our system is unstable.
- The cables can only pull the end effector but cannot push it. Given this limitation, the cables needed to remain under tension. A method for optimizing actuator torques was proposed to ensure positive torques, and a real-time torque calculation algorithm was developed for use in open-loop and closed-loop control.
- Validation of the state representation and the torque optimization algorithm using MATLAB programs revealed that the robot requires closed-loop control.
- The system's nonlinearity necessitated the use of nonlinear synthesis techniques. Comparison between state feedback control with pre-programmed gain and sliding mode control shows that both techniques are applicable to this type of robot, with the latter being superior.
- The experimental development of a prototype of the 4-cable planar robot for functional rehabilitation yielded very encouraging results. Several modifications to the prototype were proposed, along with an introduction to functional rehabilitation of the upper limbs.

However, due to time constraints and technological limitations, not all of the initial objectives were achieved. Therefore, we propose to continue this work, and for the development of this prototype, we recommend:

- The application of robust controllers.
- The use of an actuator torque optimization algorithm to avoid the problem of negative cable tension (to prevent the cables from slackening).
- The integration of patients' biomechanical data and the adaptation of exercises accordingly.
- The design of a human-machine interface to assist functional rehabilitation specialists in using this 4-cable planar robot.
- Several modifications to the prototype were proposed to improve its operation and reduce technological constraints (see section ()).

BIBLIOGRAPHY

- [1] J.P. Merlet. Kinematics of the wire-driven parallel robot MARIONET using linear actuators. In IEEE Int. Conf. on Robotics and Automation, Pasadena, 2008. (Cité en pages 9, 12, 15 et 63.)
- [2] Satoshi Tadokoro, Yoshio Murao, Manfred Hiller, Rie Murata, Hideaki Kohkawa, and Toshiyuki Matsushima. A motion base with 6-dof by parallel cable drive architecture. IEEE/ASME Transactions on Mechatronics, 7(2):115–123, 2002.
- [3] Simon Perreault and Clement M. Gosselin. Cable-driven parallel mechanisms-application to a locomotion interface. Journal of Mechanical Design, Transactions of the ASME, 130(10):1023011–8, 2008.
- [4] Tobias Bruckmann Editors. "Cable-Driven Parallel Robots book". Universitat Duisburg-Essen, Duisburg Germany, page 119.
- [5] Robert L. Williams II and Paolo Gallina. "Planar Cable-Direct-Driven Robots, Part I: Kinematics and Statics". 2001 ASME Design Technical Conferences 27th Design Automation Conference September 9-12, 2001, Pittsburgh, PA.
- [6] Robert L. Williams II and Paolo Gallina and Jigar Vadia. "Planar Cable-Direct-Driven Robots". Journal of Robotic Systems Vol. 20, No. 3, pp. 107-120, 2003.
- [7] ABDERREZEK Hadjer, « Contribution à la commande robuste des systèmes non linéaires », UNIVERSITÉ FERHAT ABBAS -SETIF-1, Algérie, 2014.
- [8] Merlet, J.-P. (2014). *Parallel Robots (3rd ed.)*. Springer International Publishing
- [9] Khadem, M., Inel, F., Carbone, G., & Slimane Tich Tich, A. (2023). A novel pyramidal cable-driven robot for exercising and rehabilitation of writing tasks
- [10] Carbone, G., Ceccarelli, M., & Otis, M. J. D. (2013). *Adiutor: A home rehabilitation robot for upper limb recovery*. In *Advances in Robot Design and Intelligent Control*. Springer.
- [11] Ceccarelli, M., Carbone, G., & Ottaviano, E. (2008). CaTraSys (Cassino Tracking System): A cable-based tracking system for motion analysis and biomechanical applications. Laboratory of Robotics and Mechatronics (LARM), University of Cassino, Italy.
- [12] Surdilovic, D., Bernhardt, R., & Zhang, L. (2007). STRING-MAN: A wire robot for gait rehabilitation, ICORR 2007
- [13] Rosati, G., Gallina, P., Masiero, S., & Rossi, A. (2007). NeReBot: A wire-based robot for neurorehabilitation of the upper limb. IEEE Transactions on Neural Systems and Rehabilitation Engineering.

- [14] Guilin Yang, Cong Bang Pham, and Song Huat Yeo. Workspace performance optimization offully restrained cable-driven parallel manipulators. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 85–90, Beijing, China, October2006
- [15] Xin Jun Liu, Xiaoqiang Tang, and Jinsong Wang. A novel 2-dof parallel mechanism based design of a new 5-axis hybrid machine tool. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 3990–5, Taipei, Taiwan, September 2003.
- [16]Wisama KHALIL, Etienne DOMBRE, Bases de la modélisation et des robots de la - manipulateurs de type série commande,2012.
- [17]Etienne Dombre, Wisama Khalil. "Modeling, Performance Analysis and Control of Robot Manipulators". Vol 413, 2007
- [18]J.P. Merlet. Kinematics of the wire-driven parallel robot MARIONET using linear actuators. In IEEE Int. Conf. On Robotics and Automation, Pasadena, 2008.
- [19] Robert L. Williams II and Paolo Gallina. "Planar Cable-Direct-Driven Robots". Journal of Robotic Systems Vol. 37, pp. 69-96, 2003.
- [20]Tobias Bruckmann Editors. "Cable-Driven Parallel Robots book". Universitat Duisburg-Essen, Duisburg Germany, page 119.
- [21] W.B.Gao and J.C.Hung. "Variable Structure Control of Nonlinear Systems : A New Approach", IEEE.
- [22] Richard C. Dorf. "OPTIMAL CONTROL SYSTEMS". Book edition, 2003.
- [23]Deghboudj Imen. "Commande des Syst`emes Non Lin´eaires par Mode Glissant d’Ordre Sup´erieur". Facult´e des Sciences de la Technologie, Universite de Constantine, Alg´erie.
- [24]Pott, A. (2018). Cable-Driven Parallel Robots: Theory and Application. Springer.
- [25]Merlet, J. P. (2006). Parallel Robots (2nd ed.). Springer.
- [26]Arduino, "Arduino Mega 2560 Rev3 Hardware Manual and Technical Specifications," Arduino Official Documentation, [Online]. Available:
- [27]J. Purdum, Microcontroller Projects for Amateur Radio: Operating and Modeling DC Geared Motors with Quadrature Encoders, 1st ed. New York: McGraw-Hill Education, 2020.
- [28]Arduino. Arduino IDE Documentation. Available at: arduino.cc.
- [29]MATLAB Simulink, PID Controller Design and Motor Control Documentation. MathWorks.

BIBLIOGRAPHY

[30]Xue Jun Jin, DaeIk Jun, Xuemei Jin, Jeongan Seon, Andreas Pott, Sukho Park, Jong-Oh Park and Seong Young Ko Upper ,Limb Rehabilitation Using a Planar Cable-Driven Parallel Robot with Various rehabilitation Strategies.

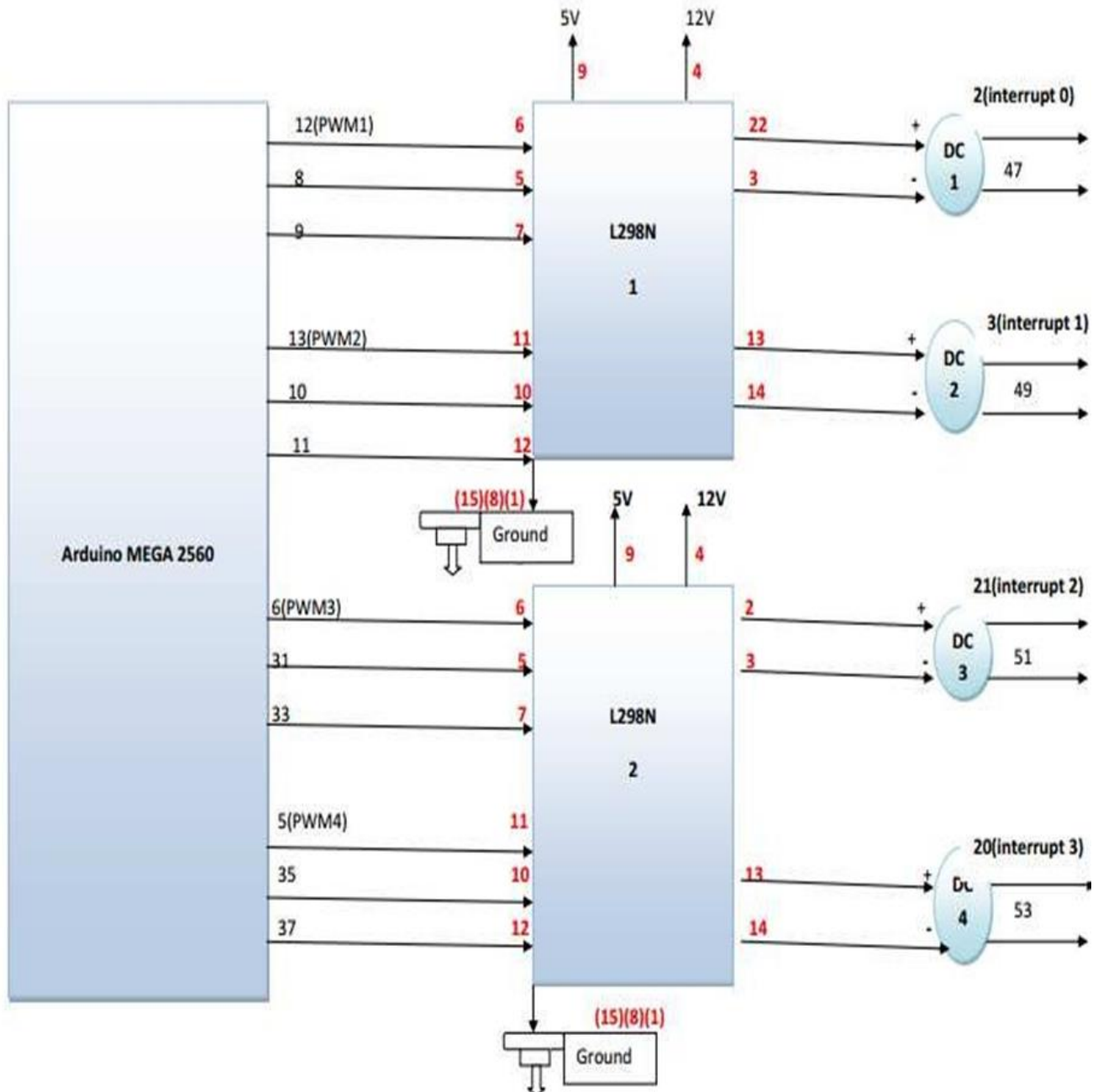
[31]www.pobot.org visiter 2017.

[32]www.lextronic.fr visiter 2017.

[33]arduino.blaisepascal.fr visiter 2017.

Appendix 1

The planar robot control circuit has four cables



Appendix 2

Graphical representation of the different control methods

State feedback control with pre-programmed gain

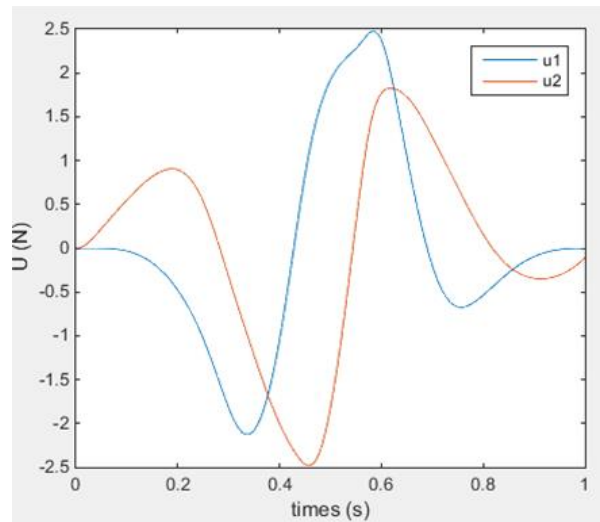


Figure 1: Circular trajectories.

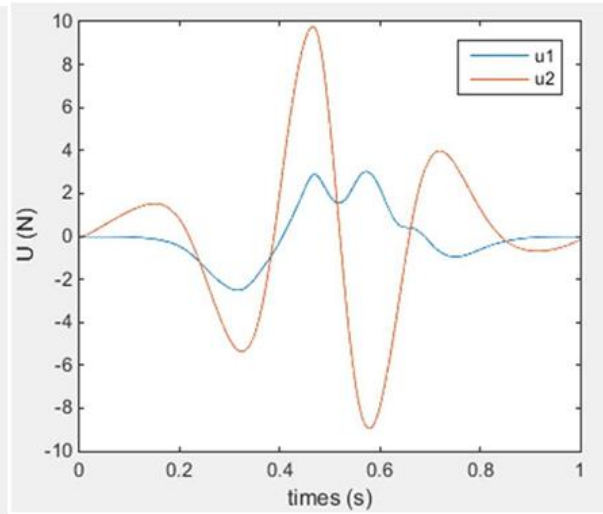


Figure 2: Lissajous trajectories.

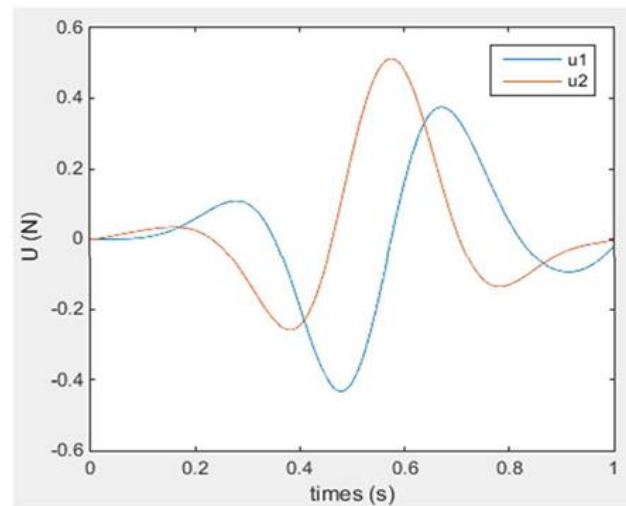


Figure 3: Spiral trajectory with period (2π).

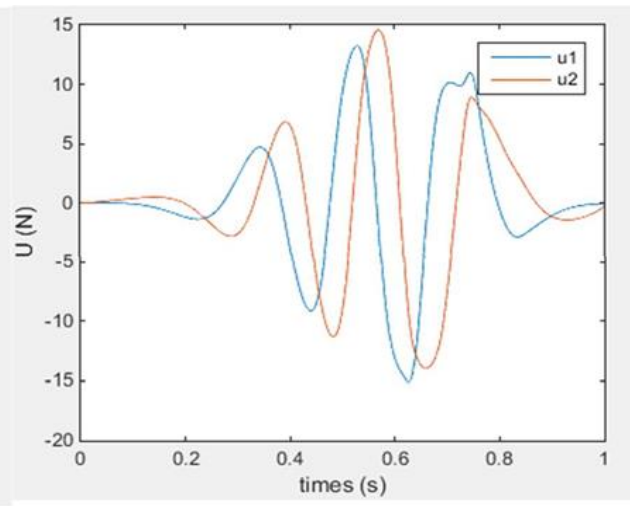


Figure 4: Spiral trajectory with period (6π).

Control via sliding mode without minimizing the chatter phenomenon

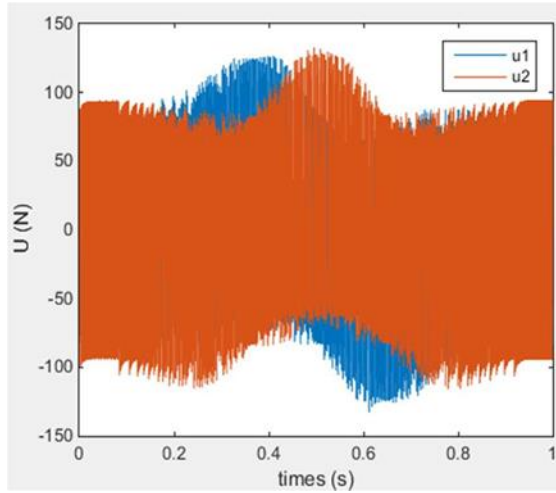


Figure 5: Circular trajectory.

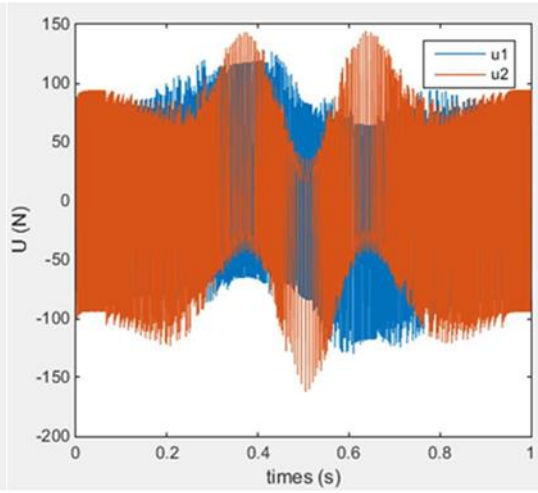


Figure 6: Lissajous trajectory.

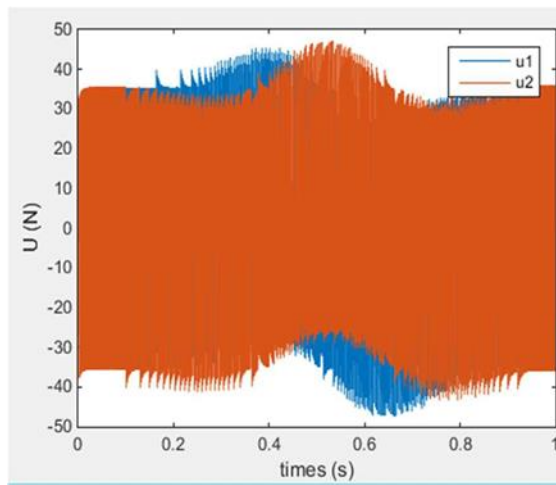


Figure 7: Spiral trajectory with period (2π).

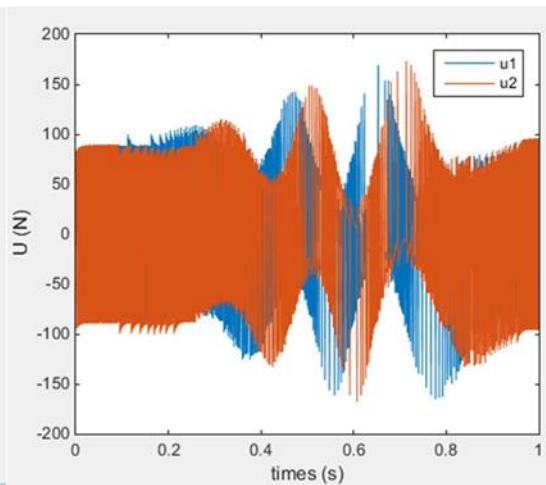


Figure 8: Spiral trajectory with period (6π).

Sliding mode control with minimization of the chatter phenomenon

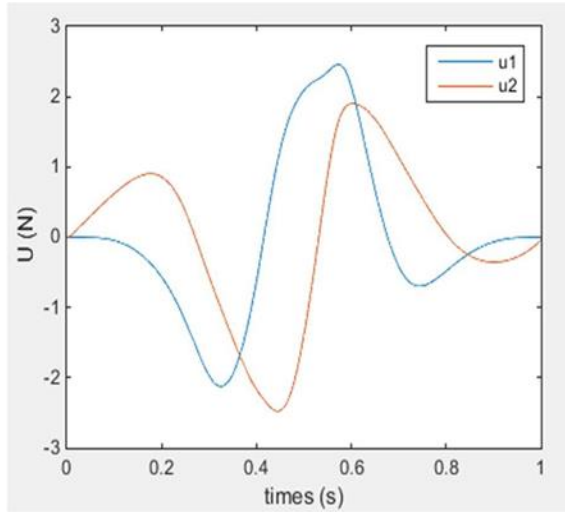


Figure 9: Circular trajectory.

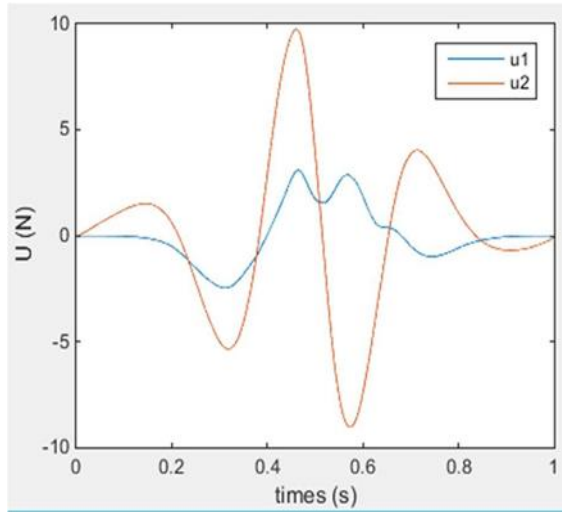


Figure 10: Lissajous trajectory.

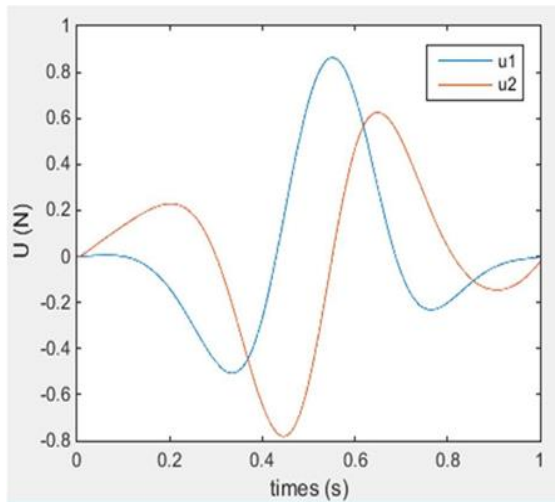


Figure 11: Spiral trajectory with period (2π).

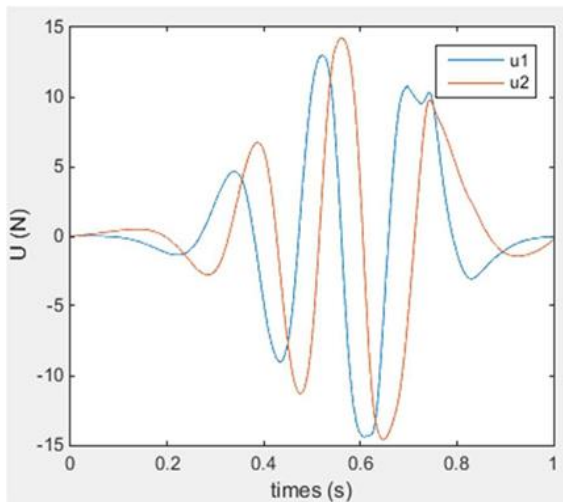


Figure 12: Spiral trajectory with period (6π).

Appendix 3

MATLAB programs

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%calcul des couples
optimales%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% données du probleme
clear all;
r=0.05;R=0.2165;lb=0.658; m=1; J1=0.0008; J2=0.0008; J3=0.0008;J4=0.0008; C1=0.01;
C2=0.01; C3=0.01;C4=0.01;
cpl11=[];cpl12=[];cpl13=[];cpl14=[];tt=[];xx=[];yy=[];tm1=[];t
m2=[];tm3=[];tm4=[];tn1=[];tn2=[];tn3=[];tn4=[];FF=[];
vxx=[];vyy=[];acxx=[];acyy=[];
t0=0;delta=0.01;tf=1;
for t=t0:delta:tf
    phi=2*pi*(10*(t/tf)^3-15*(t/tf)^4+6*(t/tf)^5); phipt=2*pi*(30*(t/tf)^2/tf-
    15*4*(t/tf)^3/tf+30*(t/tf)^4/tf); phiptt=2*pi*(60*t/(tf^3)-
    15*4*3*(t/tf)^2/(tf^2)+120*(t/tf)^3/(tf^2));
    %%la trajectoire
    x=R*cos(phi);
    y=R*sin(phi);
    vx=-R*phipt*sin(phi);
    vy=R*phipt*cos(phi);
    acx=-R*phiptt*sin(phi)-R*(phipt^2)*cos(phi);
    acy=R*phiptt*cos(phi)-R*(phipt^2)*sin(phi); vxx=[vxx
    vx];vyy=[vyy vy];
    acyy=[acyy acy];acxx=[acxx acx];
    %%l'angle theta
    A10x=-lb/2;A20x=lb/2;A30x=lb/2;A40x=-lb/2; A10y=-
    lb/2;A20y=-lb/2;A30y=lb/2;A40y=lb/2;
    thetal=(atand(abs(y-A10y)/abs(x-A10x)))*2*pi/360; theta2=(180-atand(abs(y-
    A20y)/abs(x-A20x)))*2*pi/360; theta3=(180+atand(abs(y-A30y)/abs(x-
    A30x)))*2*pi/360; theta4=(360-atand(abs(y-A40y)/abs(x-
    A40x)))*2*pi/360;
    %% les dirivée des angles theta
    deteta_1=(vy*(x-A10x)-vx*(y-A10y))/((x-A10x)^2+(y-A10x)^2);
    deteta_2=(vy*(x-A20x)-vx*(y-A20y))/((x-A20x)^2+(y-A20x)^2);
    deteta_3=(vy*(x-A30x)-vx*(y-A30y))/((x-A30x)^2+(y-A30x)^2);
    deteta_4=(vy*(x-A40x)-vx*(y-A40y))/((x-A40x)^2+(y-A40x)^2);
    %vitesse et acceleration angulaire des poulies dbeta=-
    1/r*[cos(thetal) sin(thetal);cos(theta2)
    sin(theta2);cos(theta3) sin(theta3);cos(theta4)
    sin(theta4)]*[vx;vy];
    ddbeta=1/r*([deteta_1*sin(thetal) - deteta_1*cos(thetal);deteta_2*sin(theta2) -
    deteta_2*cos(theta2);deteta_3*sin(theta3) - deteta_3*cos(theta3);deteta_4*sin(theta4) -
    deteta_4*cos(theta4)]*[vx;vy]-[cos(thetal) sin(thetal);cos(theta2)
    sin(theta2);cos(theta3) sin(theta3);cos(theta4) sin(theta4)]*[acx;acy]);
    %%mareice s
    S=[-cos(thetal) -cos(theta2) -cos(theta3) -cos(theta4);
    -sin(thetal) -sin(theta2) -sin(theta3) -sin(theta4)];
    %%matrice M M11=m*r+(J1*(cos(thetal))^2+J2*(cos(theta2))^2+J3*(cos(theta3)
    )^2+J4*(cos(theta4))^2)/r;
    M12=((J1*cos(thetal)*sin(thetal))+(J2*cos(theta2)*sin(theta2))
    +J3*cos(theta3)*sin(theta3)+(J4*cos(theta4)*sin(theta4)))/r; M21=M12;
    M22=m*r+(J1*(sin(thetal))^2+J2*(sin(theta2))^2+J3*(sin(theta3)
    )^2+J4*(sin(theta4))^2)/r;
    %%matrice N N11=(cos(thetal)*(C1*cos(thetal)-
    J1*sin(thetal))+cos(theta2)*(C2*cos(theta2)-
    J2*sin(theta2))+cos(theta3)*(C3*cos(theta3)-
    J3*sin(theta3))+cos(theta4)*(C4*cos(theta4)-
    J4*sin(theta4)))/r;
    N12=(C1*cos(thetal)*sin(thetal)+J1*deteta_1*(cos(thetal)^2+C2
    *cos(theta2)*sin(theta2)+J2*deteta_2*(cos(theta2)^2+C3*cos(th

```

```

eta3)*sin(theta3)+J3*deteta_3*(cos(theta3)^2)+C4*cos(theta4)*sin(theta4)+J1*deteta_4*(cos(theta4)^2))/r;
N21=N12;
N22=(sin(theta1)*(C1*sin(theta1)+J1*deteta_1*cos(theta1))+sin(theta2)*(C2*sin(theta2)+J2*deteta_2*cos(theta2))+sin(theta3)*(C3*sin(theta3)+J3*deteta_3*cos(theta3))+sin(theta4)*(C4*sin(theta4)+J4*deteta_4*cos(theta4)))/r;
%%% la solution par la pseudo invers
F=[M11 M12;M21 M22]*[acx;acy]+[N11 N12; N21 N22]*[vx;vy]; FF=[FF F];
taumin=max([J1 0 0 0;0 J2 0 0;0 0 J3 0;0 0 0
J4]*ddbeta+[C1 0 0 0;0 C2 0 0;0 0 C3 0;0 0 0 C4]*dbeta,0.05);
tm1=[tm1 taumin(1)];
tm2=[tm2 taumin(2)];
tm3=[tm3 taumin(3)];
tm4=[tm4 taumin(4)];
taup=pinv(S)*F;
N=null(S);
A(:,1)=[N(1,:);N(2,:)]\[taumin(1)-taup(1);taumin(2)-taup(2)];
A(:,2)=[N(1,:);N(3,:)]\[taumin(1)-taup(1);taumin(3)-taup(3)];
A(:,3)=[N(1,:);N(4,:)]\[taumin(1)-taup(1);taumin(4)-taup(4)];
A(:,4)=[N(2,:);N(3,:)]\[taumin(2)-taup(2);taumin(3)-taup(3)];
A(:,5)=[N(2,:);N(4,:)]\[taumin(2)-taup(2);taumin(4)-taup(4)];
A(:,6)=[N(3,:);N(4,:)]\[taumin(3)-taup(3);taumin(4)-taup(4)];
tn=transpose(N);
for k=1:6
TAU(:,k)=taup+transpose(transpose(A(:,k))*tn); end
optim(TAU);%%% fonction calcule les couples optimales%%
tension1=(ans(1)-C1*dbeta(1)-J1*ddbeta(1))/r;
tn1=[tn1 tension1];
tn2=[tn2 tension2];
tn3=[tn3 tension3];
tn4=[tn4 tension4];
cpl11=[cpl11 ans(1)];
cpl12=[cpl12 ans(2)];
cpl13=[cpl13 ans(3)];
cpl14=[cpl14 ans(4)];
tt=[tt t];t;xx=[xx x];yy=[yy y];
end
figure(1);
plot(xx,yy);
);
figure(2);
hold on
plot(tt,cpl11,'b');
plot(tt,cpl12,'g');
plot(tt,cpl13,'r');
plot(tt,cpl14,'y'); figure(3);
hold on
plot(tt,tm1,'b');
plot(tt,tm2,'g');
plot(tt,tm3,'r');
plot(tt,tm4,'y');
figure(4);
hold on
plot(tt,tn1,'b');
plot(tt,tn2,'g');
plot(tt,tn3,'r');
plot(tt,tn4,'y');
figure(5);
hold on
plot(tt,FF);
figure(6);
hold on
plot(tt,vxx,'-.b');
plot(tt,vyy,'--g'); figure(7);
hold on
plot(tt,acxx,'-.r');

```

```

plot(tt,acyy,'-.y');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% fonction calcule les couples optimales%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [tauopmin]=optim(TAU)
H=TAU;
TAUop=H;
for z=6:-1:1
for j=1:4
if H(j,z)<0.05-0.0001
TAUop(:,z)=[];
break
end
end
end
[m,n]=size(TAUop);
s=100;
for f=1:n
if sum(TAUop(:,f))<s
s=sum(TAUop(:,f));
tauopmin=TAUop(:,f)
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% commande on boucle ouverte%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% donne de probleme clear
; clc;

lb=0.4;R=0.1;
A10x=-lb/2;A20x=lb/2;A30x=lb/2;A40x=-lb/2; A10y=-lb/2; A20y=-
lb/2;A30y=lb/2;A40y=lb/2;
J1=0.0008;J2=0.0008;J3=0.0008;J4=0.0008;C1=0.01;C2=0.01;C3=0.01;C4=0.01;
m=1;r=0.05;x=[0;0;0;0];t0=0;tf=20;delta=0.001;tt=[];U=[];X=[];
Y=[];XX=[];YY=[];vx=0;vy=0;
for t=t0:delta:tf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% les angles theta
thetal1=(atand(abs(x(3)-A10y)/abs(x(1)-A10x)))*2*pi/360; thet2=(180-
atand(abs(x(3)-A20y)/abs(x(1)-
A20x)))*2*pi/360;
thet3=(180+atand(abs(x(3)-A30y)/abs(x(1)- A30x)))*2*pi/360;
thet4=(360-atand(abs(x(3)-A40y)/abs(x(1)- A40x)))*2*pi/360;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% les diriv   des angles
detet1=(x(4)*(x(1)-A10x)-
x(2)*(x(3)-A10y))/(x(1)-
A10x)^2+(x(3)-A10x)^2);
detet2=(x(4)*(x(1)-A20x)-x(2)*(x(3)-A20y))/(x(1)-
A20x)^2+(x(3)-A20x)^2);
detet3=(x(4)*(x(1)-A30x)-x(2)*(x(3)-A30y))/(x(1)-
A30x)^2+(x(3)-A30x)^2);
detet4=(x(4)*(x(1)-A40x)-x(2)*(x(3)-A40y))/(x(1)-
A40x)^2+(x(3)-A40x)^2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% la matrice M
M11=m*r+(J1*(cos(thet1))^2+J2*(cos(thet2))^2+J3*(cos(thet3))^2+J4*(cos(thet4))^2)/r;
M12=((J1*cos(thet1)*sin(thet1))+(J2*cos(thet2)*sin(thet2))+J3*cos(thet3)*sin(thet3)+(J4*cos(thet4)*sin(thet4)))/r; M21=M12;
M22=m*r+(J1*(sin(thet1))^2+J2*(sin(thet2))^2+J3*(sin(thet3))^2+J4*(sin(thet4))^2)/r;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% la matrice N
N11=(cos(thet1)*(C1*cos(thet1)-
J1*detet1*2*sin(thet1))+cos(thet2)*(C2*cos(thet2)-
J2*detet2*2*sin(thet2))+cos(thet3)*(C3*cos(thet3)-
J3*detet3*2*sin(thet3))+cos(thet4)*(C4*cos(thet4)-
J4*detet4*2*sin(thet4)))/r;
N12=(C1*cos(thet1)*sin(thet1)+J1*detet1*(cos(thet1)^2+C2*
cos(thet2)*sin(thet2)+J2*detet2*(cos(thet2)^2)+C3*cos(thet3)*sin(thet3)+J3*detet3*(cos(thet3)^2)+C4*cos(thet4)*sin(thet4)+J4*detet4*(cos(thet4)^2))/r;
N21=N12;

```

APPENDIX

```
N22=(sin(theta1)*(C1*sin(theta1)+J1*2*deteta1*cos(theta1))+sin
(theta2)*(C2*sin(theta2)+J2*2*deteta2*cos(theta2))+sin(theta3)
*(C3*sin(theta3)+J3*2*deteta3*cos(theta3))+sin(theta4)*(C4*sin
(theta4)+J4*2*deteta4*cos(theta4)))/r;

M2d=[1 0 0 0;0 M11 0 M12;0 0 1 0;0 M21 0 M22];
N2d=[0 1 0 0;0 -N11 0 -N12;0 0 0 1;0 -N21 0 -N22]; M=[M11
M12;M21 M22];
N=[N11 N12;N21 N22]; B=[0
0;1 0;0 0;0 1];
u1=sin(t);
u2=sin(t);
U=[u1;u2];%%%% la commande
F=inv(M2d)*N2d*x+inv(M2d)*B*U;
x=x+delta*F;
X=[X x(1)];Y=[Y x(3)];tt=[tt t];XX=[XX u1];YY=[YY u2];
end
figure(2)
hold on
plot(tt,X,'r');
plot(tt,XX,'g');grid
figure(3)
hold on
plot(tt,Y,'r');
plot(tt,YY,'g');grid
%%%%%%%%la commande par retour d'état avec gain préprogrammé%%%%%%%%
%%% données du probleme clear
;
r=0.05;R=0.2165;lb=0.658; m=1; J1=0.0008; J2=0.0008;
J3=0.0008;J4=0.0008;C1=0.01;C2=0.01;C3=0.01;C4=0.01;
tt=[];xx=[];yy=[];FF1=[];FF2=[];FF=[];
t0=0;delta=0.001;tf=1;exx=[];eyy=[];x=[R;0;0;0];
A10x=-lb/2;A20x=lb/2;A30x=lb/2;A40x=-lb/2;A10y=-lb/2;A20y=- lb/2;A30y=lb/2;A40y=lb/2;
for t=t0:delta:tf
    phi=2*pi*(10*(t/tf)^3-15*(t/tf)^4+6*(t/tf)^5); phipt=2*pi*(30*(t/tf)^2/tf-
    15*4*(t/tf)^3/tf+30*(t/tf)^4/tf); phiptt=2*pi*(60*t/(tf^3)-
    15*4*3*(t/tf)^2/(tf^2)+120*(t/tf)^3/(tf^2));
    %%%la trajectoire
    xr=R*cos(phi);
    yr=R*sin(phi);
    vx=-R*phipt*sin(phi);
    vy=R*phipt*cos(phi);
    acx=-R*phiptt*sin(phi)-R*(phipt^2)*cos(phi);
    acy=R*phiptt*cos(phi)-R*(phipt^2)*sin(phi);
    %%%l'angle theta
    theta11=(atand(abs(x(3)-A10y)/abs(x(1)-A10x)))*2*pi/360; theta12=(180-
    atand(abs(x(3)-A20y)/abs(x(1)-
    A20x)))*2*pi/360;
    theta13=(180+atand(abs(x(3)-A30y)/abs(x(1)- A30x)))*2*pi/360;
    theta14=(360-atand(abs(x(3)-A40y)/abs(x(1)- A40x)))*2*pi/360;
    %%%%%%%%%%%%% les dirivé des angles deteta11=(x(4)*(x(1)-A10x)-
    x(2)*(x(3)-A10y))/((x(1)-
    A10x)^2+(x(3)-A10x)^2);
    deteta12=(x(4)*(x(1)-A20x)-x(2)*(x(3)-A20y))/((x(1)- A20x)^2+(x(3)-
    A20x)^2);
    deteta13=(x(4)*(x(1)-A30x)-x(2)*(x(3)-A30y))/((x(1)- A30x)^2+(x(3)-
    A30x)^2);
    deteta14=(x(4)*(x(1)-A40x)-x(2)*(x(3)-A40y))/((x(1)- A40x)^2+(x(3)-
    A40x)^2);
    %%%%%%%%%%%%%la matrice M
    M11=m*r+(J1*(cos(theta11))^2+J2*(cos(theta12))^2+J3*(cos(theta
    13))^2+J4*(cos(theta14))^2)/r;
    M12=((J1*cos(theta11)*sin(theta11))+(J2*cos(theta12)*sin(theta
    12)))+(J3*cos(theta13)*sin(theta13)+(J4*cos(theta14)*sin(theta14
    )))/r;
    M21=M12;
```

APPENDIX

```

M22=m*r+(J1*(sin(theta1))^2+J2*(sin(theta2))^2+J3*(sin(theta
13))^2+J4*(sin(theta14))^2)/r;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%la matrice N
N11=(cos(theta1)*(C1*cos(theta1)-
J1*deteta11*2*sin(theta1))+cos(theta2)*(C2*cos(theta2)-
J2*deteta12*2*sin(theta2))+cos(theta3)*(C3*cos(theta3)-
J3*deteta13*2*sin(theta3))+cos(theta4)*(C4*cos(theta4)-
J4*deteta14*2*sin(theta4)))/r;
N12=(C1*cos(theta1)*sin(theta1)+J1*deteta11*(cos(theta1)^2)
+C2*cos(theta2)*sin(theta2)+J2*deteta12*(cos(theta2)^2)+C3*
cos(theta3)*sin(theta3)+J3*deteta13*(cos(theta3)^2)+C4*cos(
theta4)*sin(theta4)+J4*deteta14*(cos(theta4)^2))/r; N21=N12;
N22=(sin(theta1)*(C1*sin(theta1)+J1*2*deteta11*cos(theta1))
+sin(theta2)*(C2*sin(theta2)+J2*2*deteta12*cos(theta2))+sin
(theta3)*(C3*sin(theta3)+J3*2*deteta13*cos(theta3))+sin(the
ta14)*(C4*sin(theta4)+J4*2*deteta14*cos(theta4)))/r;
M2d=[1 0 0 0;0 M11 0 M12;0 0 1 0;0 M21 0 M22];
N2d=[0 1 0 0;0 -N11 0 -N12;0 0 1 0;0 -N21 0 -N22];
S=[-cos(theta1) -cos(theta2) -cos(theta3) -cos(theta4);
-sin(theta1) -sin(theta2) -sin(theta3) -sin(theta4) -
sin(theta4)];
B=[0 0;1 0;0 0;0 1];
a=inv(M2d)*N2d;
b=inv(M2d)*B;
c=[1 0 0 0;0 0 1 0];
k=place(a,b,[-100 -110 -110 -100]);
pre=inv(c*(-inv(a-b*k)*b));
xc=[xr;yr];
F=pre*xc-k*x;
FF1=[FF1 F(1)];
FF2=[FF2 F(2)];
FF=[FF F];
theta1= (atand(abs(yr-A10y)/abs(xr-A10x)))*2*pi/360; theta2=(180-
atand(abs(yr-A20y)/abs(xr-A20x)))*2*pi/360; theta3=(180+atand(abs(yr-
A30y)/abs(xr-A30x)))*2*pi/360; theta4=(360-atand(abs(yr-A40y)/abs(xr-
A40x)))*2*pi/360; deteta_1=(vy*(xr-A10x)-vx*(yr-A10y))/( (xr-
A10x)^2+(yr-
A10x)^2);
deteta_2=(vy*(xr-A20x)-vx*(yr-A20y))/( (xr-A20x)^2+(yr- A20x)^2);
deteta_3=(vy*(xr-A30x)-vx*(yr-A30y))/( (xr-A30x)^2+(yr- A30x)^2);
deteta_4=(vy*(xr-A40x)-vx*(yr-A40y))/( (xr-A40x)^2+(yr- A40x)^2);
%vitesse et acceleration angulaire des poulies dbeta=-
1/r*[cos(theta1) sin(theta1);cos(theta2)
sin(theta2);cos(theta3) sin(theta3);cos(theta4)
sin(theta4)]*[vx;vy];
ddbeta=1/r*([deteta_1*sin(theta1) -
deteta_1*cos(theta1);deteta_2*sin(theta2) -
deteta_2*cos(theta2);deteta_3*sin(theta3) -
deteta_3*cos(theta3);deteta_4*sin(theta4) -
deteta_4*cos(theta4)]*[vx;vy]-[cos(theta1)
sin(theta1);cos(theta2) sin(theta2);cos(theta3)
sin(theta3);cos(theta4) sin(theta4)]*[acx;acy]);
%%mareice s
S=[-cos(theta1) -cos(theta2) -cos(theta3) -cos(theta4);
-sin(theta1) -sin(theta2) -sin(theta3) -sin(theta4)]; taumin=max([J1 0
0 0;0 J2 0 0;0 0 J3 0;0 0 0 J4]*ddbeta+[C1
0 0 0;0 C2 0 0;0 0 C3 0;0 0 0 C4]*dbeta,0.05);
taup=pinv(S)*F;
N=null(S);
A(:,1)=[N(1,:);N(2,:)]\[taumin(1)-taup(1);taumin(2)-taup(2)];
A(:,2)=[N(1,:);N(3,:)]\[taumin(1)-taup(1);taumin(3)- taup(3)];
A(:,3)=[N(1,:);N(4,:)]\[taumin(1)-taup(1);taumin(4)- taup(4)];
A(:,4)=[N(2,:);N(3,:)]\[taumin(2)-taup(2);taumin(3)- taup(3)];
A(:,5)=[N(2,:);N(4,:)]\[taumin(2)-taup(2);taumin(4)- taup(4)];
A(:,6)=[N(3,:);N(4,:)]\[taumin(4)-taup(3);taumin(4)- taup(4)];
tn=transpose(N);
for k=1:6
TAU(:,k)=taup+transpose(transpose(A(:,k))*tn);

```

APPENDIX

```
end
    optim(TAU);

    S=[-cos(theta1) -cos(theta2) -cos(theta3) -cos(theta4);
        -sin(theta1) -sin(theta2) -sin(theta3) -sin(theta4)]; U=S*ans;
    F1=inv(M2d)*N2d*x+inv(M2d)*B*U; x=x+delta*F1;
    ex=x(1)-xr;
    ey=x(3)-yr;
    X=[X x(1)];Y=[Y x(3)];xx=[xx xr];yy=[yy yr];exx=[exx ex];eyy=[eyy ey];
    xcx=[xcx xc(1)];xcy=[xcy xc(2)]; tt=[tt t];
end
figure(1);
hold on
plot(tt,X,'r');
plot(tt,xcx,'g');
figure(2);
hold on
plot(tt,Y,'r');
plot(tt,xcy,'g');
figure(3);
hold on
plot(X,Y,'g');
plot(xx,yy,'--r'); figure (4) plot(tt,FF1); figure (5) plot(tt,FF2); figure

(6) plot(tt,FF);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% la commande par mode glissant%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% données du probleme clear
;
r=0.05;R=0.2165;lb=0.658; m=1; J1=0.0008; J2=0.0008; J3=0.0008;J4=0.0008; C1=0.01;
C2=0.01; C3=0.01;C4=0.01; tt=[];xx=[];yy=[];X=[];Y=[];FF1=[];FF2=[];FF=[];
t0=0;delta=0.001;tf=1;exx=[];eyy=[];x=[R;0;0;0];lenda1=150;len
da2=150;K1=1000;K2=1000;
for t=t0:delta:tf
    phi=2*pi*(10*(t/tf)^3-15*(t/tf)^4+6*(t/tf)^5); phipt=2*pi*(30*(t/tf)^2/tf-
    15*4*(t/tf)^3/tf+30*(t/tf)^4/tf); phiptt=2*pi*(60*t/(tf^3)-
    15*4*3*(t/tf)^2/(tf^2)+120*(t/tf)^3/(tf^2));
    %%%la trajectoire
    xr=R*cos(phi);
    yr=R*sin(phi);
    vx=-R*phipt*sin(phi);
    vy=R*phipt*cos(phi);
    acx=-R*phiptt*sin(phi)-R*(phipt^2)*cos(phi);
    acy=R*phiptt*cos(phi)-R*(phipt^2)*sin(phi); vxx=[vxx
    vx];vyy=[vyy vy];
    acyy=[acyy acy];acxx=[acxx acx];

a2=M12*N22-M22*N12;
a3=M21*N11-M11*N21;
a4=M21*N12-M11*N22;
b=M11*M22-M12*M21;
sx=x(2)+lenda1*(x(1)-xr);
sy=x(4)+lenda2*(x(3)-yr); if
abs([sx;sy])>1
    F=inv([M22/b -M12/b;-M21/b M11/b])*(-[0 lenda1+a1/b 0 a2/b;0 a3/b
    0 lenda2+a4/b]*x-[K1 0;0 K2]*sign([sx;sy])); else
    F=inv([M22/b -M12/b;-M21/b M11/b])*(-[0 lenda1+a1/b 0 a2/b;0 a3/b 0
    lenda2+a4/b]*x-[K1 0;0 K2]*[sx;sy]);
end
    FF1=[FF1 F(1)];
    FF2=[FF2 F(2)];
    FF=[FF F];
    theta1=(atand(abs(yr-A10y)/abs(xr-A10x)))*2*pi/360; theta2=(180-
    atand(abs(yr-A20y)/abs(xr-A20x)))*2*pi/360; theta3=(180+atand(abs(yr-
    A30y)/abs(xr-A30x)))*2*pi/360;
    theta4=(360-atand(abs(yr-A40y)/abs(xr-A40x)))*2*pi/360;
    deteta_1=(vy*(xr-A10x)-vx*(yr-A10y))/((xr-A10x)^2+(yr-
```

```

A10x)^2);
    deteta_2=(vy*(xr-A20x)-vx*(yr-A20y))/((xr-A20x)^2+(yr-A20x)^2);
    deteta_3=(vy*(xr-A30x)-vx*(yr-A30y))/((xr-A30x)^2+(yr-A30x)^2);
    deteta_4=(vy*(xr-A40x)-vx*(yr-A40y))/((xr-A40x)^2+(yr-A40x)^2);
%%vitesse et acceleration angulaire des poulies dbeta=-
1/r*[cos(theta1) sin(theta1);cos(theta2)
sin(theta2);cos(theta3) sin(theta3);cos(theta4)
sin(theta4)]*[vx;vy];
    ddbeta=1/r*([deteta_1*sin(theta1) -
deteta_1*cos(theta1);deteta_2*sin(theta2) -
deteta_2*cos(theta2);deteta_3*sin(theta3) -
deteta_3*cos(theta3);deteta_4*sin(theta4) -
deteta_4*cos(theta4)]*[vx;vy]-[cos(theta1)
sin(theta1);cos(theta2) sin(theta2);cos(theta3)
sin(theta3);cos(theta4) sin(theta4)]*[acx;acy]);
%%mareice s
    S=[-cos(theta1) -cos(theta2) -cos(theta3) -cos(theta4);
-sin(theta1) -sin(theta2) -sin(theta3) -sin(theta4)]; taumin=max([J1 0 0 0;0 J2
0 0;0 0 J3 0;0 0 0
J4]*ddbeta+[C1 0 0 0;0 C2 0 0;0 0 C3 0;0 0 0 C4]*dbeta,0.05);
    taup=pinv(S)*F;
    N=null(S);
    A(:,1)=[N(1,:);N(2,:)]\[taumin(1)-taup(1);taumin(2)-taup(2)];
    A(:,2)=[N(1,:);N(3,:)]\[taumin(1)-taup(1);taumin(3)-taup(3)];
    A(:,3)=[N(1,:);N(4,:)]\[taumin(1)-taup(1);taumin(4)-taup(4)];
    A(:,4)=[N(2,:);N(3,:)]\[taumin(2)-taup(2);taumin(3)-taup(3)];

    A(:,5)=[N(2,:);N(4,:)]\[taumin(2)-taup(2);taumin(4)-taup(4)];
    A(:,6)=[N(3,:);N(4,:)]\[taumin(4)-taup(3);taumin(4)-taup(4)];
    tn=transpose(N);
    for k=1:6
        TAU(:,k)=taup+transpose(transpose(A(:,k))*tn);
    end

    optim(TAU);
S=[-cos(theta11) -cos(theta12) -cos(theta13) -cos(theta14);
-sin(theta11) -sin(theta12) -sin(theta13) -sin(theta14)];
    U=S*ans;
    F1=inv(M2d)*N2d*x+inv(M2d)*B*U; x=x+delta*F1;
    ex=x(1)-xr;
    ey=x(3)-yr;
    X=[X x(1)];Y=[Y x(3)];xx=[xx xr];yy=[yy yr];exx=[exx ex];eyy=[eyy ey];
    xcx=[xcx xr];xcy=[xcy yr]; tt=[tt t];
    end figure(1);
    hold on
    plot(tt,X,'r');
    plot(tt,xcx,'b');
    figure(2);
    hold on
    plot(tt,Y,'r');
    plot(tt,xcy,'b');
    figure(3);
    hold on
    plot(X,Y,'b');
    plot(xx,yy,'--r');
    figure(4)
    plot(tt,FF1);
    figure(5)
    plot(tt,FF2);
    figure(6)
    plot(tt,FF);

```