Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in Computer Engineering

A Motion Planner for Complex Trajectories with Cable Suspended Parallel Robots

Master Candidate

Maliha khan

Student ID 2043885

Supervisor

Prof. Giovanni Boschetti

University of Padova

Academic Year
2023/2024

"Behind every successful robot is a team of dedicated engineers. Behind every dedicated engineer is a family who believed in them. Just as a robot's path is guided by algorithms, my path has been guided by the love and support of my family."

Abstract

Exact trajectory generation is still absolutely essential in the realm of advanced robotics if we are to increase the dynamic capability of robotic systems. This thesis explores the trajectory generation for a unique 3 Degrees of Freedom (DOF) robot meant to control a suspended ball by following challenging trajectories including spirals and cubic curves.

The key challenge is precisely and effectively generating motion paths in three-dimensional space using the DOF of the robot to negotiate the suspended ball across predefined trajectories.By means of a comprehensive kinematic and dynamic analysis, an ideal control approach is devised,therefore ensuring smooth and stable motion along spiral and cubic trajectories.

The combination of real-time feedback control with trajectory planning is provided as a novel approach to this. Using polynomial interpolation and spline-based approaches, the method builds continuous and differentiable paths.These paths then become control signals for the robot's actuators,therefore allowing precise motions and transitions.

Extensive simulations and experimental settings verify the efficiency of the proposed method by establishing its power to control complex motion scenarios while maintaining excellent accuracy and stability.Notable performance increases in the robot can be found in reduced trajectory deviation, improved path smoothness, and greater reaction to dynamic changes.

This work advances robotic trajectory planning by providing insights on the development of control methods for multi-DOF systems engaged in demanding motion tasks.Where precise trajectory control is essential,the results have applications in fields including dynamic manipulation, automated assembly, and entertainment robots.

Keywords: Trajectory Generation, 3 Degrees of Freedom, Spiral Trajectories, Cubic Trajectories, Suspended Ball Robot, Kinematics, Dynamics, Real-Time Control, Polynomial Interpolation, Spline Methods.

Sommario

Cable-driven robots offer significant benefits in applications needing large-workspaces and the ability to lift heavy weights. The small weight of these robots and their simplicity of reconfiguring define them.The evolution of methods for the generation of trajectories for a cable-driven robot with three degrees of freedom (DOF)intended to move a suspended ball along complex patterns,including spiral and cubic trajectories is discussed in this thesis.One of the most significant successes is the development of a new method combining polyn interpolation with spline techniques to create smooth and accurate paths, which are validated by simulations and experiments. Moreover, the thesis addresses safety precautions that can be used to control the probability of cable breakdowns,therefore ensuring consistent operation.The results throw light on the feasibility of precise and safe trajectory control, which will finally help to progress the use of cable-driven robots in many different fields outside research.

Contents

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

List of Acronyms

CDPR	Cable-Driven Parallel Robot
DOF	Degrees of Freedom
PID	Proportional-Integral-Derivative
MPC	Model Predictive Control
LQ	Linear Quadratic
CNC	Computer Numerical Control
PLC	Programmable Logic Controller
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
ROS	Robot Operating System
API	Application Programming Interface
GUI	Graphical User Interface
IoT	Internet of Things
AI	Artificial Intelligence
ML	Machine Learning
2D	Two-Dimensional
3D	Three-Dimensional

Glossary

A

Acceleration The rate of change of velocity of an object with respect to time.

Actuator A component of a machine responsible for moving or controlling a mechanism or system.

Algorithm A step-by-step procedure used for calculations, data processing, and automated reasoning tasks.

B

Ball Parallel Robot A type of parallel robot that uses a spherical joint to provide rotational and translational movements.

C

Configuration Space (C-space) The space of all possible positions and orientations of a robot.

Control Point A point used to define the shape and path of a trajectory.

D

Degree of Freedom (DOF) The number of independent movements a robot can perform.

Dynamics The study of forces and torques and their effect on motion.

E

End Effector The part of a robot that interacts with the environment, such as a gripper or tool.

F

Feedback Control A control system that uses feedback to regulate the motion and position of a robot.

Forward Kinematics The calculation of the position and orientation of the end effector given the joint parameters.

G

Gradient Descent An optimization algorithm used to minimize the error by iteratively moving towards the steepest descent.

H

Homogeneous Transformation Matrix A matrix used to describe the position and orientation of a robot in space.

I

Inverse Kinematics The calculation of the joint parameters needed to place the end effector at a desired position and orientation.

J

Jacobian Matrix A matrix that relates the rates of change of the robot's joint parameters to the rate of change of the end effector's position and orientation.

K

Kinematics The study of motion without considering the forces that cause it.

L

Linear Interpolation A method of constructing new data points within the range of a discrete set of known data points.

M

Manipulator A robot's arm used for positioning and orienting an end effector.

Motion planning The process of defining a sequence of movements that a robot must perform to achieve a specific task.

N

Numerical Optimization A mathematical method used to find the best possible solution to a problem by iteratively improving a candidate solution.

P

Path Planning The process of developing a feasible path for the robot to follow from the start point to the end point.

Pose The position and orientation of a robot or its end effector.

Q

Quaternion A mathematical representation used to describe orientation in three-dimensional space, avoiding gimbal lock.

R

Real-Time Processing The capability of a system to process data and provide results almost instantaneously.

Redundancy The presence of more degrees of freedom than necessary to complete a task, providing flexibility in motion planning.

S

Singularity A configuration of a robot where the end effector loses one or more degrees of freedom.

Spiral Trajectory A path that follows a helical or spiral pattern, often used in machining or robotic path planning.

T

Trajectory Generation The process of designing a path for a robot to follow, including positions, velocities, and accelerations over time.

Transformation The mathematical operation that moves points from one coordinate system to another.

V

Velocity The rate of change of position of an object with respect to time.

W

Waypoint A predefined point in space that a robot must pass through as part of its trajectory.

Y

Yaw The rotation around the vertical axis of a robot, one of the three components of orientation.

1 Introduction

These robots, supported by tight wires due to gravity, serve a variety of purposes, including capturing overhead footage during sporting events and aiding in the construction of large radio telescopes.An essential obstacle to ensuring the efficient functioning of these robots is the strategic design of their movement courses,also known as trajectories, in order to maintain constant tension in the cables during their operation.In the majority of studies on this subject, scientists have employed intricate dynamic equations to design trajectories that ensure that the cables remain taut and do not loose.

These equations serve as restrictions, directing the robot's movement to ensure that the tension in the wires remains constant at all times.Tight wires are critical for correctly controlling the robot's motions, as they can only exert pulling force and not pushing force.Recent research has used a distinct methodology for trajectory planning. Researchers have included the dynamic equations directly into the 3-DOF planar CSPRs rather than employing them just as constraints. Furthermore, the unsolved problem lies in the design of rotational paths, in which the robot must spin around a certain location while maintaining taut wires.To summarize, the approach to dynamic trajectory planning for CSPRs has progressed from including dynamic restrictions to directly integrating dynamic equations. This has revealed the significant influence of oscillation frequencies, as discussed in reference [20], on the generation of viable pathways. Although there has been notable progress in comprehending the generation of steady and melodious paths, further investigation is required to address the issues of changeable rigidity and rotating motions. This ongoing investigation

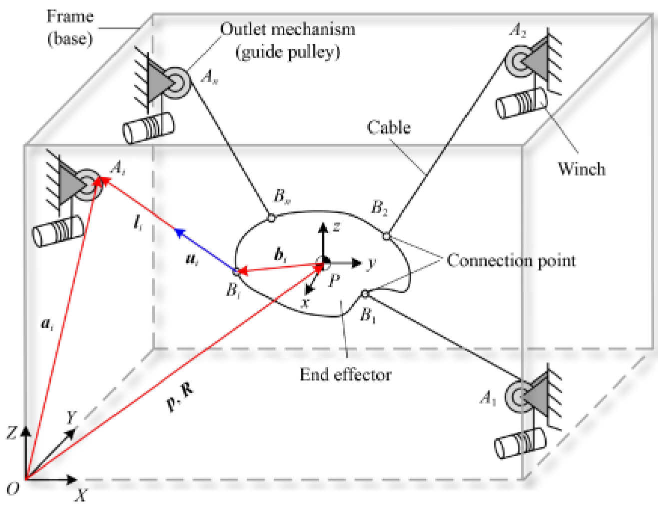

Figure 1.1: Schematic representation of a generic cable robot

holds the potential to improve the flexibility and functionality of CSPRs, opening up possibilities for novel and creative uses.The Skycam is an early and widely recognised instance of cable-suspended camera systems (CSPRs). It is probable that you have observed it gliding over stadiums during a game, recording exhilarating moments from an elevated position. The NIST RoboCrane [1] is a well-known example in the industry. The Skycam [2], shown in Figs. 1.2 and 1.3 respectively.

This robot, equipped with six adjustable degrees of freedom (DOF), assists in maintaining the stability of crane loads, therefore reducing hazardous swinging. CSPRs play a vital role in the assembly of large-scale constructions, such as the colossal 500-metre radio telescope in China, by aiding in the manipulation of the substantial and fragile components.

1.1 Problem Statement

A unique class of robotic systems that offer remarkable speed, accuracy, and adaptability in their movements are ball parallel robots. However, designing accurate and efficient routes for autonomous robots might be a significant challenge. Ball parallel robots have a complex kinematic structure, with several degrees of freedom and a highly nonlinear relationship between the joint angles and the end-effector location. This makes it challenging to create accurate mathematical models that represent the robot's movements, which is essential for effective trajectory planning.

Figure 1.2: NIST RoboCrane prototype

Figure 1.3: Skycam

Furthermore, the process of creating trajectories is made more difficult by the fact that there are several possible solutions to the inverse kinematics problem. This is due to the possibility that the robot can go in several directions to reach a certain end-effector location. Ball parallel robots can move quickly and powerfully, which places significant dynamic constraints on them that should be carefully taken into account while designing trajectories. When planning the movement of Cable-Suspended Parallel Robots (CSPRs), several crucial limitations need to be considered. These include the maximum speeds, accelerations, and forces the robot's joints can handle. Additionally, certain configurations, known as singularities, must be avoided because they can lead to uncontrollable or unpredictable behavior. Stability during the robot's motion is also essential to prevent excessive shaking or wobbling. Ignoring these constraints can result in impractical paths or poor performance, such as the robot oscillating too much, failing to follow its intended path accurately, or even causing damage to its components due to overstressed joints or excessive forces. Ensuring that these dynamic limitations are respected is key to maintaining the robot's reliability and precision.

1.2 Design

Within this particular segment. Over the years, the use of cables in robotics has led to the development of many design solutions.Essentially, cables can be used to either activate the kinematic chain in combination with a rigid-link structure or replace typical rigid-link architectures by directly affecting a rigid body,such as the moving platform or end-effector.Both scenarios include mounting the motors at the base of the manipulator, which effectively decreases the inertial load.Cable-driven serial connection architectures, as shown in Figure 1.4, are commonly utilised in planar devices, such as those referenced in citations [**27**] and [**29**]. While hybrid systems utilising a parallel chain of cables to activate a serial structure have been devised [27],[28], the most generally used pattern is one where each cable passes through the previous joints before reaching the matching rigid link).Therefore, it can be problematic to route cables around the joints in 3D applications.To address the issue, bowden cables can be employed Figure:1.4[30], [31].However, this approach introduces additional friction, necessitating the implementation of control mechanisms to counteract it. Recently, prototypes of this category of manipulators have been introduced in the field of rehabilitative robotics [32].The determination of the controllable workspace for generic cable-driven open chains is still unresolved, despite the recent publication of a paper addressing this issue [33]. Additionally, there have been designs of parallel rigid-link chains that are powered by cables, as seen in Figure 1.6.These designs offer an additional benefit, since the load can be evenly distributed among the actuators [34],[35].

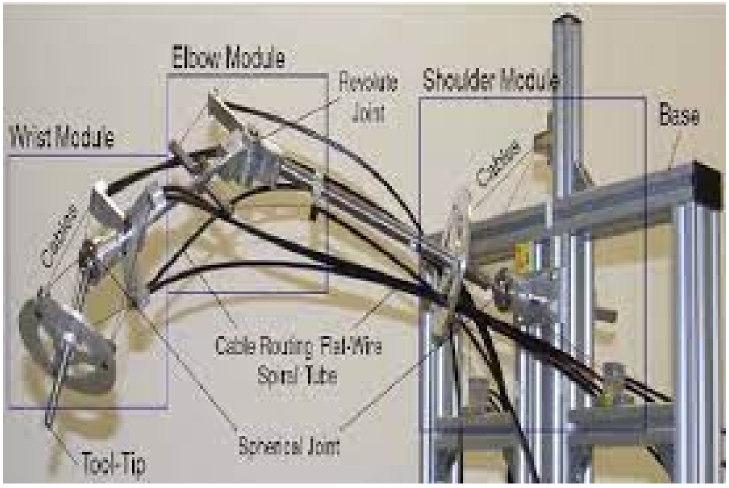

Figure 1.4: Cable-Driven Robotic Arm [30]

This study specifically examines cable-based parallel robots, which are systems that utilise cables to directly support a single rigid body known as the end-effector. Typically, each cable is operated by a separate motor that regulates the length and tension of the cable. In addition to the actuated cables, a collection of passive, unchangeable wires may be added to restrict the movements of the platform[3]. Seriani et al.figure:1.5[15] proposed a modular CDPR deployed by a rover shown in Figure 5. Due to the large work scale of CDPRs, the mentioned modular CDPR can be applied in inspection tasks in field and rugged environment. One key distinction between cable-based parallel systems and ordinary parallel robots is that cables are limited to transmitting tension forces only. This attribute is commonly known as unilateral actuation. The utilisation of elements with the ability to apply one-sided forces has numerous ramifications. Firstly, having a greater number of actuators than degrees of freedom (DOFs) imposes a constraint on the system. The designer should utilise algorithms that aim to achieve a practical distribution of tension for any particular wrench. Furthermore, in contrast to conventional inflexible-link mechanisms,

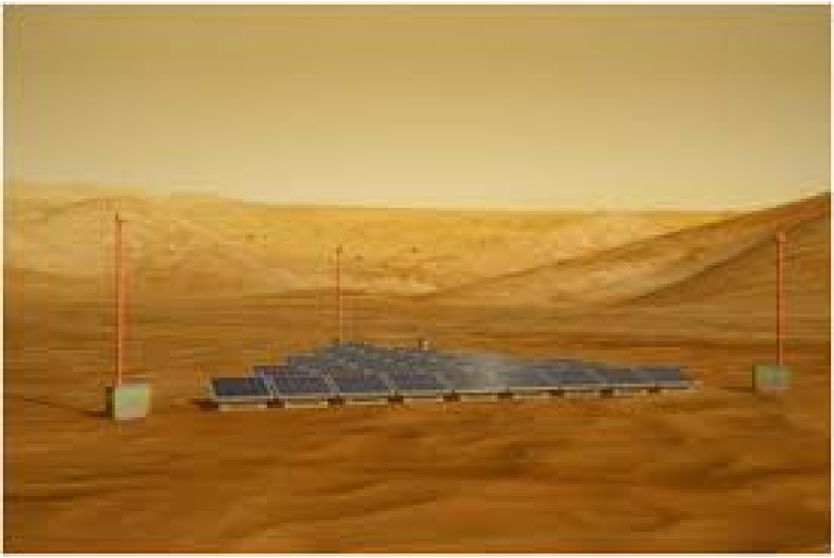

Figure 1.5: Modular CDPR for solar collection in field and rugged environment

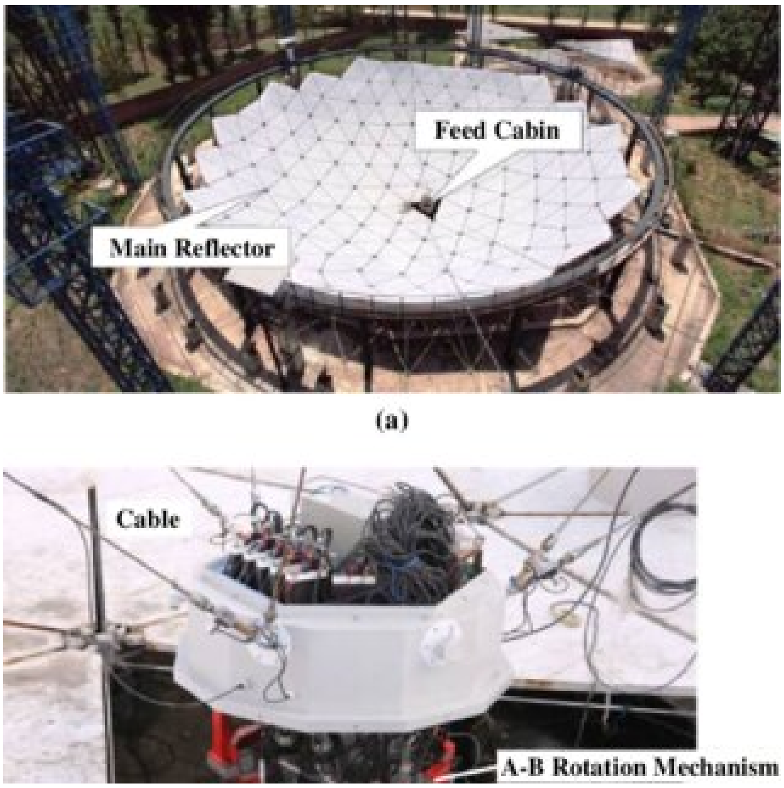

Figure 1.6: Similarity prototype of feed support system in FAST

the range of motion of the workspace is not simply determined by geometric limitations.The controllability of the mobile platform in the workspace depends on the arrangement of the cables and the permissible range of cable tensions. This sliding mode controller can work without prior knowledge as well as the linearization of dynamic models. Abdelaziz et al. [80] presented a position control method for CDPR. An internal cable tension control loop is introduced into the controller for compensating friction. Particular design constraints were considered including the length, size and materials of transmissions.Tang et al. [81] presented a hybrid position/force controller for CDPRs. The pseudo-drag problem of flexible cable is taken into account and prevented. The approach is validated in 1:15 similarity prototype of feed support system in FAST, as shown in Figure1.6.

1.3 Trajectory Optimization

For ball parallel robots, trajectory optimisation concentrates on reducing time, energy consumption, and obstacle avoidance. In fast pick-and-place procedures, when output is high, this is absolutely vital. Reducing task times allows the robot to complete more in a given timeframe, therefore increasing general output. Reduced energy consumption also results in better efficiency and cheaper running costs, therefore rendering the robot more ecologically friendly.

One of the major drawback of CDPRs is the cable sagging during the moving of EE. It is a challenging problem to solve the trajectory planning of CDPRs, due to the pseudo-drag problem of cables [25] .Trajectory planning also has to take into account roadblocks in the robot's path to create paths free of collisions and guarantees of safe motion. In real-time situations where the robot's course must be rapidly modified, this is particularly difficult. Practical and responsive trajectory planning solutions are developed using modern computational approaches and effective numerical methods. The performance of ball parallel robots is greatly improved by accurate trajectory planning, which results in faster job completion, greater movement accuracy, and more smooth operations. Plotting the robot's motions to minimise any threats improves safety as well, particularly in cooperative or industrial contexts. Furthermore, precise trajectory planning helps the robot to be dependable, therefore extending its lifetime and lowering the possibility of malfunctions. enhanced efficiency and output

- **Enhanced Efficiency and Performance**: Precise trajectory planning is pivotal for optimizing the performance of Cable-Driven Parallel Robots (CDPR). With accurately calculated paths, these robots can execute movements with heightened precision, leading to smoother operations and faster completion of tasks. This not only maximizes system throughput but also enhances the quality of output, thereby boosting productivity and efficiency in automated and industrial environments.
- **Safety in Sensitive Environments**: In scenarios where CDPRs operate in close proximity to humans or other machines, ensuring safety is paramount. Accurate trajectory planning is crucial to prevent collisions and unsafe interactions. By strategically navigating potential hazards, CDPRs can operate more securely and efficiently, reducing risks in complex environments such as manufacturing floors or research facilities.
- **Reliability and Longevity**: The reliability of CDPRs significantly depends on the quality of trajectory planning. Designing trajectories that consider the dynamic constraints of the robots minimizes wear and tear on mechanical components, reducing breakdowns and maintenance costs while extending the operational lifespan of the robots. This reliability is crucial for supporting continuous production processes and minimizing operational downtime, especially in critical industrial applications.
- **Optimization of Resources**: Precise trajectory planning in CDPR systems also optimizes the use of resources by ensuring that movements are executed in the most efficient manner possible. This includes minimizing the energy consumption and reducing the time taken for movements, which are essential for scaling operations and enhancing the overall sustainability of robotic systems.

1.4 Objectives

The major goal of this thesis is to design and apply a motion planner for a Cable-Driven Parallel Robot (CDPR) able to precisely track intricate trajectories, especially spiral and cubic paths. The motion planner guarantees exact and smooth trajectory tracking by addressing the unique challenges given by the suspended character of the CDPR.

1. **Design of a Quintic Polynomial Trajectory Planner:** Build smooth, continuous paths for the CDPR using a trajectory planner based on quintic polynomials. This method will ensure that the specified boundary criteria are satisfied by the initial and final positions as well as velocities, therefore providing a basis for smooth motion.

2. **Dynamic Trajectory Planning:** Find and regulate the intended pathways by using the dynamic equations of the CDPR.[23]. This approach will ensure that the cables sustain the right tension during the motion, thus managing the complex interactions among the cables and the moving platform.

3. **Building an Optimal Control Trajectory Planner:** Approach the task of trajectory planning as a control optimization challenge. While following the dynamic constraints and CDPR boundary conditions, the aim is to minimize a cost function considering elements such as accuracy, path smoothness, and energy economy.

4. **Modeling and Verification:** Conduct extensive simulations to confirm the effectiveness of the suggested motion planner. The simulations will evaluate the planner's accuracy in producing and following precisely both spiral and cubic trajectories. Moreover, actual testing with a suspended CDPR will show how to apply the developed motion planner practically.

5. **Overcoming Practical Challenges:** Address pragmatic issues including avoiding combinations that cause instability, regulating the orientation of the movable platform, and preventing collisions with the environment. The planner will be especially created to take these problems into account to guarantee safe and efficient operation in practical circumstances.

6. **Performance Metrics Evaluation:** Analyze significant criteria such as trajectory accuracy, computing efficiency, smoothness of motion, and resilience to disturbances to evaluate the motion planner. These tests will provide insightful analysis of the general effectiveness and suitability for various applications of the planner.

This thesis aims to satisfy these goals, contributing to the field of robotic motion planning, particularly for suspended CDPRs. The aim is to enable these robots to complete challenging tasks with great accuracy and efficiency.

1.5 Scope

This dissertation covers the design, implementation, and evaluation of a motion planner especially adapted for complex trajectories in cable suspended parallel robots (CDPRs). This work intends to solve the special difficulties and possibilities given by CDPRs by using their special features to accomplish exact and effective motion planning in Chapter 4 and Chapter 5.

2

Literature Review

Robotic motion planning is critically dependent on trajectory generation, which is the computation of a path or trajectory for the end effector or mobile platform of a robot to follow to complete a given task. Many techniques have evolved over time,each of which has special advantages and applications. This part studies basic approaches in trajectory planning and presents a chronological history.

2.1 Historical Development of Trajectory Planning

Over the past few years, much study on the evolution of trajectory planning for robots from simple linear pathways to sophisticated multi-dimensional trajectories has been conducted.Trajectory planning has evolved inside the framework of Cable-Suspended Parallel Robots (CSPR) or Cable-Driven Parallel Robots (CDPR) driven by the demand for accuracy, efficiency, and flexibility in numerous applications. This literature review together with important achievements and contributions to the area characterizes the evolution of trajectory planning historically.

First Development of Trajectory Plans

Early robotics research concentrated on producing simple trajectories such as straight lines and circular arcs.These were sufficient for rudimentary automation systems and simple pick-and-place activities[21].Early industrial robots guaranteed simple implementation and control by virtue of linear interpolation between points.

Introduction of Polynomial Trajectories

Particularly cubic position, polynomial trajectories were made to have more fluid motion.These paths guaranteed constant acceleration and speed, therefore lowering mechanical strain and enhancing the performance of the robotic systems.The flexibility of cubic splines and Bezier curves to offer flexible and smooth path generation helped them to gain popularity.

Advances in Parallel Robot Trajectory Planning

Closed-loop kinematic chains in parallel robots demanded more sophisticated trajectory planning techniques. These robots fit activities needing great accuracy since they provide higher stiffness and accuracy than serial robots. Renowned parallel robot Stewart platform proved the requirement of advanced trajectory planning to completely leverage its capabilities.

Cable-Driven Parallel Robots (CDPRs)

Emerging as a particular sort of parallel robot, CDPRs use cables rather than rigid linkages to control the end-effector. The lightweight and adaptability of cables allowed great payload-to-weight ratios and enormous workspaces as shown in figure 2.1. Aiming for fundamental path planning, early CDPR research addressed issues including workspace analysis, tension control, and cable sag.

Evolution of Complex Trajectory Planning

Time-Optimal and Energy-Efficient Trajectories :Researchers started looking at time-optimal pathways in pursuit of respect for dynamic restrictions and a shortened travel time between sites. This method especially is used in industrial automation and high-speed applications.Especially, energy-efficient trajectory planning became increasingly important to lower mechanical component wear and consequently energy consumption.

Figure 2.1: Prototype of CDPRs

Advanced Polynomial and Spline Methods : Two upgraded polynomials and spline algorithms were developed to guarantee smoother trajectories with continuous jerk (the derivative of acceleration), so enhancing motion quality.Quintic and higher-order polynomials were generated. More complicated and exact path generating was made feasible by B-splines and NURBS (Non-Uniform Rational B-Splines) since they gave more freedom and control over trajectory forms.

Spiral and Helical Trajectories : Applications depending on spiral and helical paths—such as CNC machining,robotic inspection, aerial drones—dependent on 3D navigation become extremely critical. These routes guaranteed perfect and effective operation by carefully covering three-dimensional areas.To satisfy the particular needs of many applications, mathematical and pragmatic development of spiral pathways was undertaken.

Integration of Control Strategies

PID and Adaptive Control : Adopted extensively for their simplicity and efficiency in controlling robotic system dynamic behaviour were proportional-integral-derivative (PID) controllers. PID control shifted control inputs depending on feedback to guarantee exact following of the intended path. Adaptive control techniques were designed to raise dependability and resilience by means of control of variations in the environment and uncertainty of the robot.

Model Predictive Control (MPC) : When one projects future states and optimises control inputs using a model of the robot's dynamics, Model Predictive Control (MPC) becomes a potent control tool. This method allowed real-time corrections, hence improving the robot's capacity to precisely follow difficult paths.More complex and sensitive motion control was made possible for CDPRs particularly by linking MPC with trajectory planning.

Recent Trends and Future Directions

Artificial Intelligence and Machine Learning : Since artificial intelligence and machine learning methods allow the learning from data and improve performance over time, they have been progressively implemented in trajectory planning.These techniques maximise paths depending on actual experience and change with the times.Especially fascinating in creating autonomous robots capable of controlling challenging environments with low human involvement is reinforcement learning.

Optimization-Based Trajectory Planning : Trajectory planning has benefited much from optimisation tactics since algorithms meant to identify the best potential trajectories depending on numerous variables including time, energy, and safety. Particle swarm optimisation, evolutionary algorithms, and other heuristic strategies have been investigated to effectively handle difficult trajectory planning.

Human-Robot Collaboration : As robots run alongside people more and more, trajectories planning must consider safety and cooperation.Safe interactions depend on constant, smooth routes. Research on fundamental trajectory planning techniques able to dynamically adapt to human motions and intents has been motivated by human-robot interaction.

2.2 Trajectory Generation for Parallel Robots

Traversal generation for parallel robotics is the planning and execution of particular motions of the end-effector of the robot usually in three-dimensional space. With their many connected kinematic chains from the base to the end-effector, parallel robots offer specific choices for trajectory design as well as chal-

Figure 2.2: Prototype of CAMCAT's

lenges.This part addresses contemporary approaches and techniques unique to parallel robotics together with their benefits, drawbacks, and uses.

Unprecedented freedom in capturing dynamic events in cinematography is made possible by cable-driven camera robots. Systems under cable control offer all-around and planar motion. While all-around motion systems offer shooting flexibility and a field of vision with endless rotation, planar motion systems balance and offset shooting angles.Beautiful and flexible are Skycam and Spidercam.While the spidercam uses standard components for smooth horizontal and vertical motion, August Design created the skycam, which travels fast and precisely. Both devices are manoeuvrable, hover, have hitherto unheard-of event access.

A novel cable-driven system called CoGiRo [16] has been recently created for transporting loads in industrial areas. This is the largest cable-driven parallel robot in Europe, designed with a crane-like configuration. The device can be operated manually using a joystick figure:2.2 displays the Cogiro. Kawamura et al.[22] introduced a novel robot design called FALCON-7, which is a cable-driven parallel system, capable of achieving extremely high speeds.Their findings demonstrated that employing wires with nonlinear spring characteristics enhanced the transient response of the system, albeit at the expense of complicating the analysis of stability.Their empirical findings demonstrated consistent performance of the manipulator. The study [14]introduced a novel approach that use the internal tension of wires to effectively mitigate vibration.

Yangwen et al. [26] developed a novel wire-driven parallel suspension system for an aeroplane model in a low-speed wind tunnel.They investigated techniques

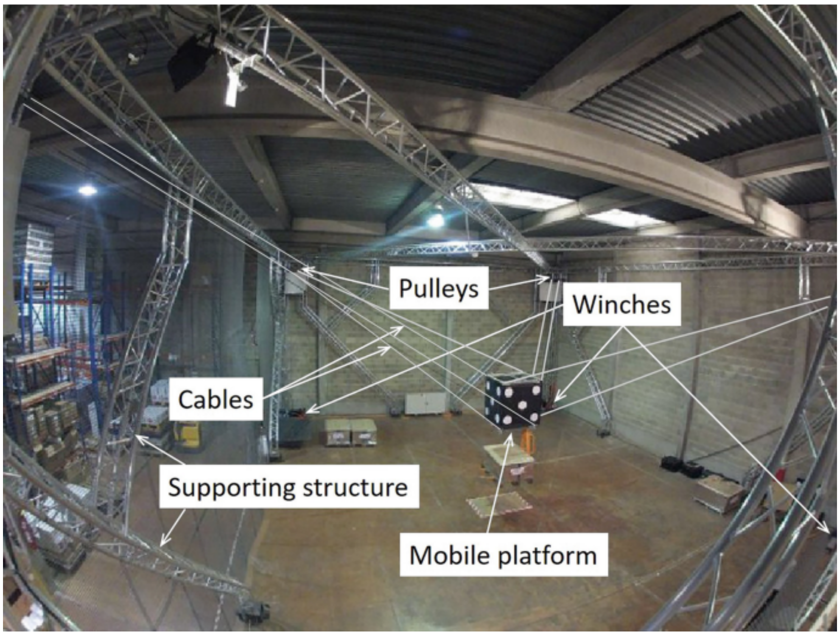

Figure 2.3: Cogiro (The Europe biggest cable-driven parallel robot)[5]

for measuring and calculating the aerodynamic properties of the aeroplane model.The research findings confirmed that a wire-driven parallel manipulator may be effectively utilised as the suspension system for conducting low-speed wind tunnel experiments.

Gosselin et al.[13] introduced a cable-driven planar parallel haptic interface with three degrees of freedom (3-DOF).The control of the prototype in this study relied on utilising the force/torque sensor to deduce the user's intents and compute the associated prescribed positions. In order to minimise the forces exerted, the platform of the haptic interface prototype was controlled manually using the hands. Gallina et al.[12] created a planar haptic device named Feriba-3, which is driven by 4 wires and has 3 degrees of freedom.The device exhibited excellent haptic capabilities, guaranteeing efficient manipulation throughout a wide range of workspace. A multitude of researchers have employed cable-driven systems in order to develop medical devices that are more efficient. Rosati et al.[20],[12] created a wire-driven robot called NeReBot figure: 2.4 for rehabilitating the upper limb after a stroke. The robot has three degrees of freedom (DoF). Essentially, the robot is comprised of three wires that are each controlled by their own electric motors. The wires are attached to the patient's upper limb using a splint and are held up by a movable frame positioned above the patient. Rehabilitation treatment can be administered throughout a wide range of mo-

Figure 2.4: NeReBot for rehabilitating

tion by changing the length of the wire used, whether the treatment is based on passive or active-assistive spatial motion of the limb. Brackbill et al.[13] introduced a cable-driven exoskeleton.A presentation was given on a wearable upper arm exoskeleton designed for human users.The exoskeleton has four degrees-of-freedom and is powered by six cables.The presentation included the dynamics, control, and initial experiments of the exoskeleton. Rehabilitation equipment utilise cable-driven systems, which offer certain advantagesIn [14], a proposal was made for a cable-driven active leg exoskeleton (C-ALEX)[26]. as shown in figure: [2.5]. designed specifically for human gait training.Cable-driven designs offer several benefits, including a streamlined structure, low impact on the movement of human limbs, and the absence of a need for perfect joint alignment.The experimental findings demonstrated that the suggested system has the ability to assist the subjects in accurately monitoring a specified ankle trajectory.

Cable-driven lifting robots are an important advancement in lifting equipment technology, offering versatility, efficiency, and cost-effectiveness for handling a variety of materials in various scenarios.

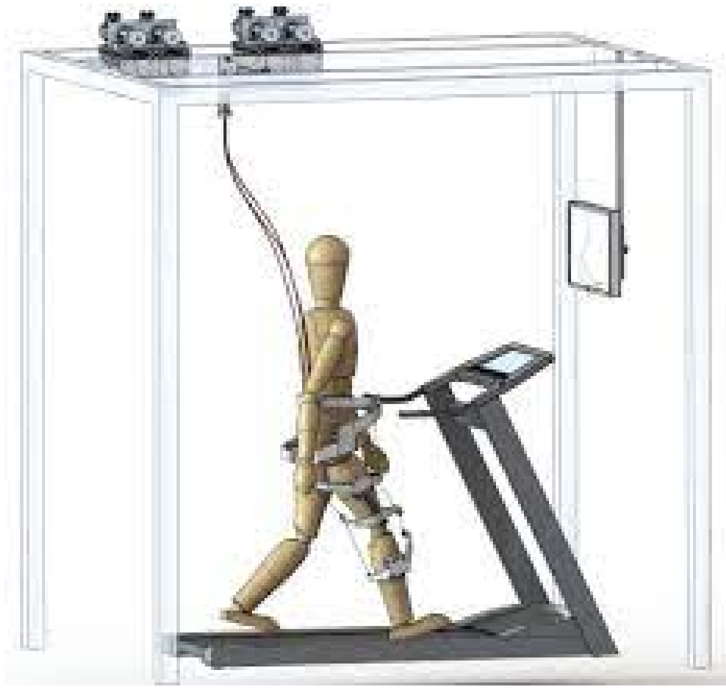

Figure 2.5: A design of a cable- driven active leg exoskeleton (C-ALEX))

2.2.1 Comparative Analysis of Trajectory Planning Techniques

This section provides a comparative examination of several trajectory planning methods employed for Cable-Driven Parallel Robots (CDPRs). The analysis explicitly evaluates the appropriateness, effectiveness, and computational efficiency of the subject. The contrasted methodologies encompass polynomial trajectories, spline-based methods, optimization-based planning, advanced control strategies including Model Predictive Control (MPC), and machine learning-based approaches. Polynomial trajectories refer to the paths or movements that can be described by polynomial functions.

2.2.2 Cubic and Quintic Polynomials

Polynomial trajectories are determined by utilising polynomial equations to calculate the path of the end-effector. Cubic polynomials ensure continuous position and velocity, whereas quintic polynomials further ensure continuous acceleration.

- **Advantages**:
    - Simplicity in execution

  - Flawless motion with continuous transitions
  - Suitable for basic tasks that require a modest level of complexity

- **Disadvantages**:
  - Limited flexibility for really complex routes
  - May struggle to effectively handle dynamic limitations

- **Applications**:
  - Used in circumstances that need smooth and direct paths, such as jobs involving the manipulation and positioning of objects, as well as basic automated movements

2.2.3 Spline-Based Methods

Spline-based methods, such as B-Splines and Non-Uniform Rational B-Splines (NURBS),[9] offer enhanced adaptability by allowing the trajectory to be defined using a series of control points.

- **Advantages**:
  - Substantial flexibility and precision in determining the appropriate course of action
  - Smooth transitions between segments
  - Suitable for complex and detailed routes

- **Disadvantages**:
  - Heightened computational intricacy
  - Requires precise calibration of control points and weights

- **Applications**:
  - Ideally suited for tasks that need precise and complex trajectories, such as robotic surgery, sophisticated assembly techniques, and CNC machining

2.2.4 Optimization-Based Planning

Optimization-based planning entails transforming the task of trajectory planning into a problem of optimisation. The goal is to minimise or maximise specific factors, such as time, energy, or path length.

- **Advantages**:
  - Offers the most efficient solutions according to defined criteria
  - Proficient in efficiently managing dynamic constraints and conducting multi-objective optimisation
- **Disadvantages**:
  - Significant computational expense
  - Necessitates the implementation of resilient optimisation techniques and solutions
- **Applications**:
  - This technology is especially well-suited for applications that require high speed, industrial automation, and situations where energy saving is extremely important, such as autonomous vehicles and drones

2.2.5 Advanced Control Strategies

Model Predictive Control (MPC)

The Model Predictive Control (MPC) technique uses a dynamic model of the robot to predict future states and optimise control inputs within a specified time period. It adjusts the trajectory in real-time by utilising feedback.

- **Advantages**:
  - Instantaneous optimisation and precise adjustments in real-time
  - Efficiently handles constraints associated with states and inputs
  - Enhances resilience against disturbances and inaccuracies in the model
- **Disadvantages**:

  - Requires accurate and dependable dynamic models
  - The real-time implementation necessitates a substantial allocation of processing resources

- **Applications**:
  - Ideally suited for use in dynamic and unexpected scenarios, such as autonomous navigation, complex assembly tasks, and real-time adaptive systems

Machine Learning Methods

The use of machine learning techniques, such as reinforcement learning and neural networks, allows for the determination of the most effective paths by analysing data. These techniques have the capacity to adapt and conform to new tasks and environments by accumulating information and abilities.

- **Advantages**:
  - Ability to gain knowledge and adapt depending on previous experiences
  - Skilled in overseeing complex and multifaceted environments
  - Suitable for positions that include variability and uncertainty

- **Disadvantages**:
  - Requires a significant amount of training data
  - Training may require a substantial allocation of computational resources
  - May exhibit low interpretability and may not ensure optimality

- **Applications**:
  - Ideally suited for tasks that need complex decision-making, such as autonomous exploration, human-robot interaction, and flexible control in many situations

2.2.6 Challenges and Future Directions

Despite advances, trajectory generation for parallel robots presents several challenges.

- **Complex Kinematics:** The intricate kinematics of parallel robots, particularly the forward kinematics, make real-time trajectory planning computationally intensive.
- **Dynamic Constraints:** Parallel robots often operate under stringent dynamic constraints, requiring careful consideration of forces, torques, and joint limits.
- **Environmental Uncertainty:** Real-world environments introduce uncertainty and variability, necessitating robust and adaptive trajectory planning algorithms.

Future research in trajectory generation for parallel robots is likely to focus on:

- **Advanced Optimization Techniques:** Develop more efficient optimization algorithms to handle complex kinematics and dynamic constraints.
- **Machine Learning Integration:** Leveraging machine learning techniques to predict and adapt to environmental changes, enhancing the robustness and adaptability of trajectory planning.
- **Real-Time Control:** Improving real-time control capabilities to enable rapid adjustments to trajectories in dynamic and uncertain environments.

2.3 Conclusion

Simple linear tracks to complex position and spiral trajectories, trajectory design for cable-suspended parallel robots has changed dramatically.Artificial intelligence, optimisation methods, and control strategies have significantly increased CDPR capability, therefore enabling their outstanding accuracy and efficiency to complete difficult jobs.Future studies on adding more sophisticated artificial intelligence algorithms, enhancing real-time performance, and guaranteeing safe and efficient human-robot interaction will most likely centre on technological advancement as well as their implications.

3

Theoretical Background of CDPRs

3.1 Kinematics of Ball Parallel Robots

Complex devices designed specifically to manipulate a spherical ball by using a parallel sequence of actuators are ball parallel robots. These very accurate robots find use for applications requiring precise placement and control of orientation. Ball parallel robotics solves inverse and forward kinematic problems using kinematic analysis.

Forward kinematics, from the actuator positions, is the study of the ball-position and orientation of the end-effector.Usually, the linked element of their mechanics makes this more difficult in parallel robots. Imagine first a ball parallel robot with three actuators positioned at A_1, A_2, and A_3. The end-effector (ball) is connected to these actuators at points B_1, B_2, and B_3. The lengths of the actuators are denoted as l_1, l_2, and l_3. Let the center of the ball be at position $P(x, y, z)$. Define the vectors from the center of the ball to the actuator connection points: $\vec{r}_1$, $\vec{r}_2$, $\vec{r}_3$. The positions of the actuator ends can be given by $\vec{A}_i = \vec{P} + \vec{r}_i$ for $i = 1, 2, 3$. Each actuator length gives a distance constraint: $|\vec{A}_i - \vec{B}_i| = l_i$. Substituting the positions into the distance constraints provides three nonlinear equations:

$$\sqrt{(x + r_{1x} - A_{1x})^2 + (y + r_{1y} - A_{1y})^2 + (z + r_{1z} - A_{1z})^2} = l_1$$

$$\sqrt{(x + r_{2x} - A_{2x})^2 + (y + r_{2y} - A_{2y})^2 + (z + r_{2z} - A_{2z})^2} = l_2$$

$$\sqrt{(x + r_{3x} - A_{3x})^2 + (y + r_{3y} - A_{3y})^2 + (z + r_{3z} - A_{3z})^2} = l_3$$

These equations are typically solved using numerical methods such as Newton-Raphson because of their non-linear nature.

The mechanics of ball parallel robots are revealed in the stresses and torques needed to traverse the spherical end effector of the ball through its workspace in Fig:3.1.

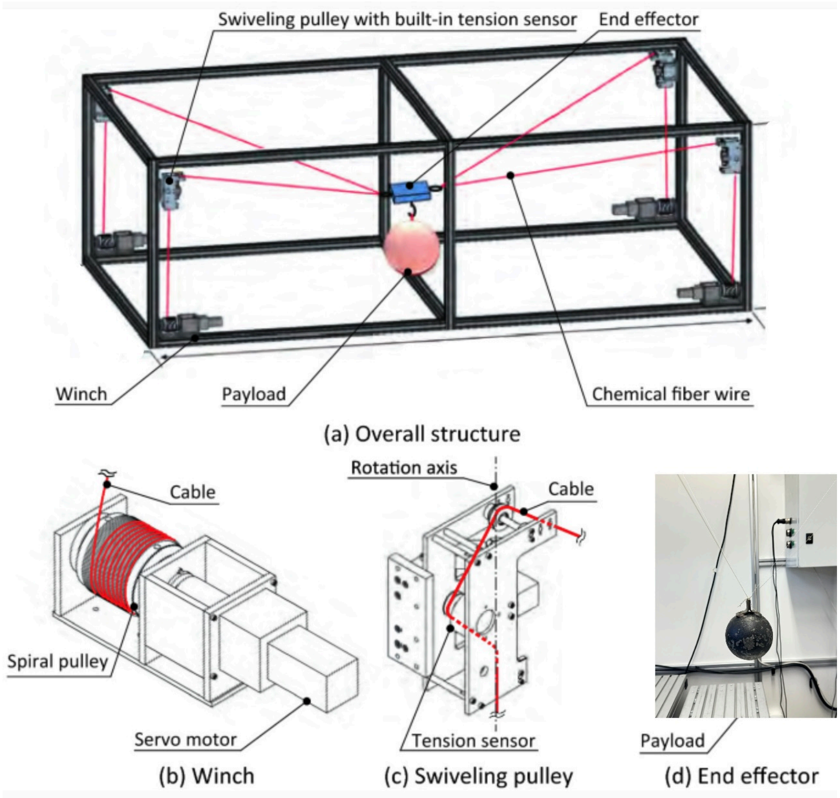

Figure 3.1: A Suspended Cable-Driven Parallel Robot

Good control and manipulation depend on awareness of these systems.This implies repeating the dynamics of the robot utilising control techniques, therefore guaranteeing the expected mobility and stability. One can replicate the dynamics of ball parallel robots by means of derived equations of motion defining the interaction between the applied forces/torques and the resultant motion of the ball. This calls for examining the kinematics as well as the physical characteristics of the robot, including mass and moments of inertia.

To start, define the robot configuration, assuming a ball parallel robot with three actuators positioned at A_1, A_2, and A_3, and connection points on the ball at B_1, B_2, and B_3. Let the ball mass be m and the inertia tensor moment be $\mathbf{I}$.

The kinematic equations relate the actuator positions and velocities to the ball's position and orientation.

Applying Newton's second law for translation yields:

$$m\ddot{\mathbf{P}} = \sum \mathbf{F}_i$$

For rotation, Euler's equation is applied:

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \sum \mathbf{M}_i$$

where $\mathbf{P}$ is the position of the ball's center, $\boldsymbol{\omega}$ is the angular velocity, $\mathbf{F}_i$ are the forces applied by the actuators, and $\mathbf{M}_i$ are the torques.

The torque and force of every actuator define the dynamics in whole.These are computed considering the actuator forces and their applications sites. Combining the translational and rotational equations generates a group of coupled differential equations reflecting the whole dynamics of the system.

Reaching exact motion control of the ball parallel robot requires the appropriate control methods to be applied. These techniques guarantee that, regardless of external disturbances or model errors, the desired position and orientation are kept.

First, defining control objectives and hence the trajectory or setpoints is choosing the intended ball location and orientation.While PID controllers and other feedback control systems help to reduce the error between the desired and actual positions, sensors track the ball's orientation and present location.

One can find the necessary actuator forces and torues by means of inverse dynamics.This is using the forces and torques required to get the target accelerations:

$$\mathbf{F}_{desired} = m\ddot{\mathbf{P}}_{desired}$$

$$\mathbf{M}_{desired} = \mathbf{I}\dot{\boldsymbol{\omega}}_{desired} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}_{desired})$$

Feedforward control inputs are produced from inverse dynamics using computed forces and torques,therefore enabling more precise and smoother motion.

Resilient performance in several operating environments is guaranteed by the use of adaptive control techniques to enable changes in the properties of the robot and uncertainty [3].Simulations validate the control strategies to verify their expected performance; the control settings are carefully changed depending on the simulation findings. Under real environmental conditions, empirical testing confirms the effectiveness of the control system.

Imagine a situation whereby the ball has to travel in a horizontal plane following a circular path. The desired trajectory is given by $P_{desired}(t) = (R\cos(\omega t), R\sin(\omega t), h)$, where R is the radius, ω is the angular velocity, and h is a constant height.

For the inverse dynamics calculation,the desired accelerations are:

$$\ddot{P}_{desired} = (-R\omega^2\cos(\omega t), -R\omega^2\sin(\omega t), 0)$$

The required forces are:

$$\mathbf{F}_{desired} = m\ddot{P}_{desired} = m(-R\omega^2\cos(\omega t), -R\omega^2\sin(\omega t), 0)$$

The actuators are arranged such that their configuration and attachment points distribute these forces.

Combining dynamics modeling with effective control strategies guarantees perfect positioning and orientation control by guiding the ball parallel robot to follow the desired trajectory. Many high-precision applications depend on the optimal performance of ball parallel robots depending on this integrated method of dynamics and control.

3.2 Unit Vector Calculation for Cable Driven System

The motion of the end-effector in a Cable-Driven Parallel Robot (CDPR) is controlled by the length and tension of several cables that are anchored at fixed points and connected to the end-effector.[7] [6].To effectively manipulate the end-effector along a desired trajectory, it is essential to compute the unit vectors of the cables that dictate the direction of the forces exerted by the cables.

Consider a CDPR with n cables. Each cable is attached at one end to a fixed base at point $\mathbf{P}_i$ and at the other end to the moving platform (end-effector) at point $\mathbf{Q}_i$.

- $\mathbf{P}_i = (P_{ix}, P_{iy}, P_{iz})$ - Coordinates of the i-th attachment point on the fixed base.

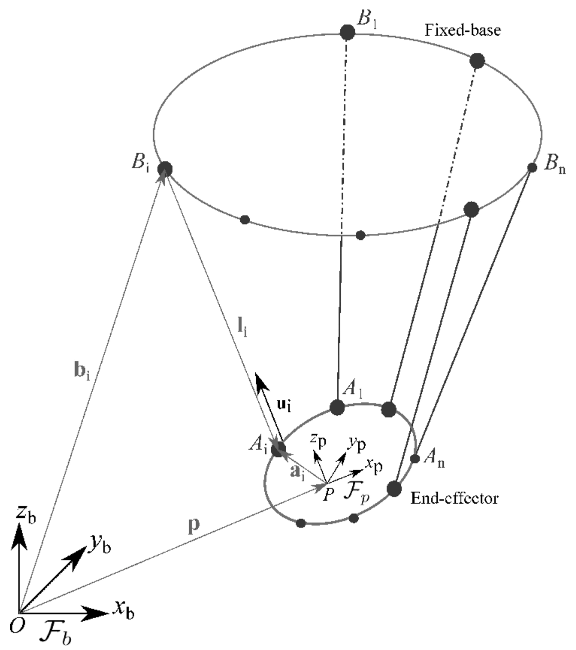

Figure 3.2: Closed Loop of CDPR

- $\mathbf{Q}_i = (Q_{ix}, Q_{iy}, Q_{iz})$ - Coordinates of the i-th attachment point on the moving platform.

3.2.1 Calculation of Unit Vectors

The vector $\mathbf{L}_i$ representing the direction and length of the i-th cable is given by the difference between the position vectors:

$$\mathbf{L}_i = \mathbf{Q}_i - \mathbf{P}_i = (Q_{ix} - P_{ix}, Q_{iy} - P_{iy}, Q_{iz} - P_{iz}) \tag{3.1}$$

The unit vector $\hat{\mathbf{L}}_i$ is the normalized form of $\mathbf{L}_i$ and is calculated by dividing $\mathbf{L}_i$ by its magnitude $||\mathbf{L}_i||$, where:

$$||\mathbf{L}_i|| = \sqrt{(Q_{ix} - P_{ix})^2 + (Q_{iy} - P_{iy})^2 + (Q_{iz} - P_{iz})^2} \tag{3.2}$$

Hence, the unit vector $\hat{\mathbf{L}}_i$ is:

$$\hat{\mathbf{L}}_i = \frac{\mathbf{L}_i}{||\mathbf{L}_i||} = \left(\frac{Q_{ix} - P_{ix}}{||\mathbf{L}_i||}, \frac{Q_{iy} - P_{iy}}{||\mathbf{L}_i||}, \frac{Q_{iz} - P_{iz}}{||\mathbf{L}_i||} \right) \tag{3.3}$$

A generic platform presents six degrees-of-freedom (6-DOF):

$$\mathbf{d} = [x, y, z, \alpha, \beta, \gamma]^T = [\mathbf{p}, \boldsymbol{\theta}]^T$$

This is where the position p and orientation of the platform are represented by the letter d.Due to the fact that a recovery approach often concentrates on the position of the end-effector,the end-effector was taken into consideration as a point mass in this undertaking.Only the vector p, and thus the first three degrees of freedom, have been taken into consideration. In addition, this research investigates redundant cable robots that are characterised by the values of $m \geq n + 1$.Therefore, the value of n is equal to three because this study takes into consideration a point mass end-effector.

Finally, it is only possible to carry out the proposed recovery strategy in the case that the after-failure architecture is at least partially operational and the after-failure practical workspace is not null. Making sure at least three cables have exit points not on the same vertical plane will help to guarantee that this last requirement is satisfied. This makes the produced work relevant to over-actuated spatial cable robots, which are robots featuring at least four cables.

For a Cable-Driven Parallel Robot (CDPR), understanding the direction of each cable relative to the end-effector is crucial for accurate control and movement. This section details the calculation of the unit vectors which represent the direction of the tension forces for each cable connecting the end-effector to fixed attachment points.

3.2.2 Definition and Calculation

For each cable i, the unit vector $\mathbf{u}_i$ defines the direction from the fixed attachment point to the end-effector figure: 3.3, and is calculated using the following formula:

$$\mathbf{u}_i = \frac{\mathbf{r}_i - \mathbf{p}}{||\mathbf{r}_i - \mathbf{p}||} \tag{3.4}$$

where:

- $\mathbf{u}_i$ is the unit vector for cable i,
- $\mathbf{r}_i$ is the position vector of the fixed attachment point of cable i,

- **p** is the position vector of the end-effector (or the ball in some configurations).

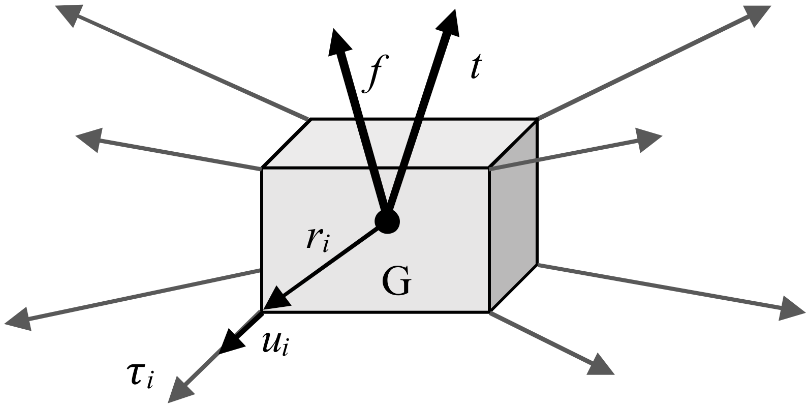

Figure 3.3: : Schematic representation of the vectors ui, ri, i

Vector from End-Effector to Fixed Point

The vector from the end-effector to the fixed point $\mathbf{r}_i$ is given by:

$$\mathbf{r}_i - \mathbf{p} \tag{3.5}$$

This vector points directly from the end-effector's current position to the fixed point where the cable is attached.

Magnitude of the Vector

The magnitude of the vector, representing the Euclidean distance between the end-effector and the fixed point, is computed as:

$$\|\mathbf{r}_i - \mathbf{p}\| \tag{3.6}$$

Normalization

The unit vector $\mathbf{u}_i$ is obtained by normalizing the vector $\mathbf{r}_i - \mathbf{p}$ as shown in Equation 3.4:

$$\mathbf{u}_i = \frac{\mathbf{r}_i - \mathbf{p}}{\|\mathbf{r}_i - \mathbf{p}\|} \tag{3.7}$$

This normalization ensures that $\mathbf{u}_i$ accurately represents only the direction of the cable without any magnitude, making it fundamental for controlling the tension and trajectory of the end-effector.

This formula efficiently shows how the direction of each cable is determined by the position of the ball in respect to the permanent attachment points. Understanding and changing these unit vectors will help you to regulate the forces applied on the ball by every cable, hence regulating its motion.

3.3 Dynamic Analysis for Cable-Driven Parallel Robots

Dynamic analysis forms a cornerstone for effective trajectory planning in Cable-Driven Parallel Robots (CDPRs). This analysis is crucial for both motion control and ensuring safety measures. The dynamics can be approached both directly, [11] to understand motion affected by external and inertial forces, and inversely, to determine necessary forces for achieving a desired motion trajectory.

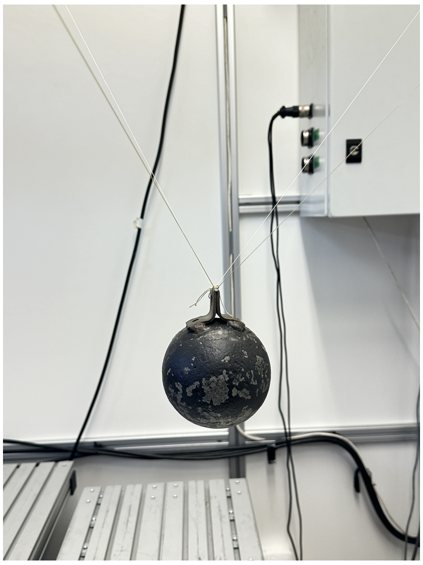

Figure 3.4: Prototype of End-effector

3.3.1 End-Effector Dynamics

We begin by considering the forces and moments acting on the end-effector as shown in figure:3.4, represented by the wrench vector $\mathbf{w}$:

$$\mathbf{w} = \begin{bmatrix} \mathbf{f}^T & \mathbf{m}^T \end{bmatrix}^T \tag{3.8}$$

where $\mathbf{f}$ and $\mathbf{m}$ are the vectors for forces and moments, respectively. The dynamic equilibrium, accounting for mass and inertia, is described by:

$$M\ddot{x} = \mathbf{w} \tag{3.9}$$

Here, M denotes the mass and inertia matrix:

$$M = \begin{bmatrix} M_e & 0 \\ 0 & I_e \end{bmatrix} \tag{3.10}$$

The external forces and the forces exerted by the cables combine into $\mathbf{w}$. The force vector from the cables, $\mathbf{w}_c$, is linked to the cable forces $\mathbf{f} = [f_1, f_2, \ldots, f_m]^T$ through the structure matrix S:

$$\mathbf{w}_c = S\mathbf{f} \tag{3.11}$$

S projects cable tensions into Cartesian coordinates:

$$S = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_m \\ \mathbf{r}_1 \times \mathbf{u}_1 & \mathbf{r}_2 \times \mathbf{u}_2 & \ldots & \mathbf{r}_m \times \mathbf{u}_m \end{bmatrix} \tag{3.12}$$

Here $\mathbf{u}_i$ is the unit vector from the base attachment point to the i-th pulley, and $\mathbf{r}_i$ is the vector from the center of mass to the i-th attachment point.

3.3.2 Pulley Dynamics

The dynamics of the pulleys involve the forces acting on each pulley, shown in figure 3.7, with the motor torque $\mathbf{T}$ related to the cable tension through the dynamic equation:

$$\mathbf{T} = I_m\ddot{\theta} + C_m\dot{\theta} + \tau \tag{3.13}$$

where I_m and C_m represent the rotational inertia and damping coefficients, respectively. The angular acceleration $\ddot{\theta}$ is derived as:

$$\ddot{\theta} = \frac{d}{dt}\left(J^{-1}\dot{x}\right) + \frac{d^2}{dt^2}x \tag{3.14}$$

Figure 3.5: Overview of Motor Attached with pulley

Figure 3.6: Configuration of Pulley

3.3.3 Dynamic Equilibrium of the Robot

Integrating the dynamics of both the end-effector and the pulleys, we derive the comprehensive dynamic equilibrium equation:

$$rM\ddot{x} + SIm\frac{d^2x}{dt^2} = S\mathbf{T} + rw_{\text{ext}} \tag{3.15}$$

This equation, consolidated in Equation 3.15, encompasses the forces and torques affecting the robot, foundational for precise control implementations.

3.4 Algorithm for Calculating Unit Vectors in CDPR

The algorithm inputs are the coordinates of attachment points on both the fixed frame and the moving platform (end-effector), and it outputs the unit vector for each cable.

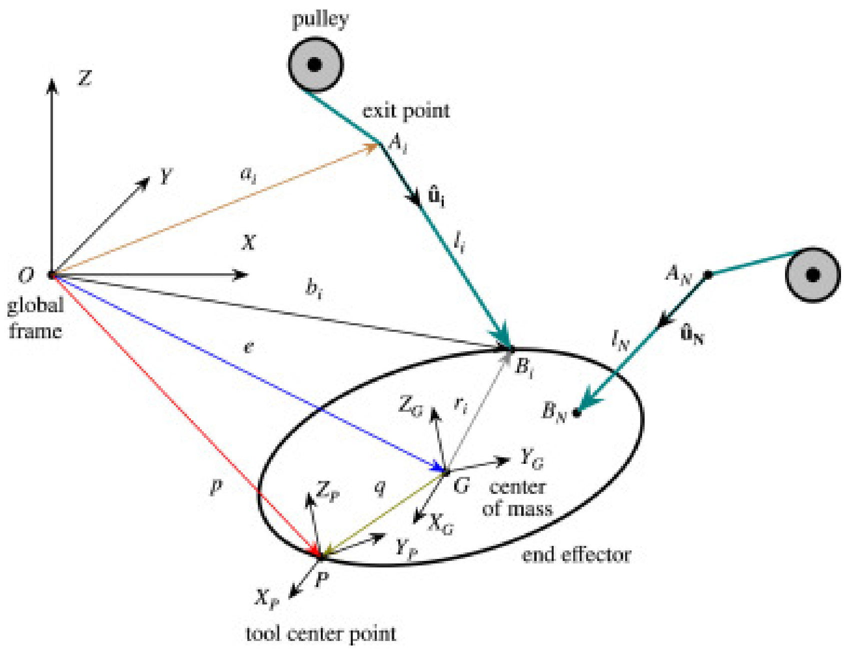

Figure 3.7: Generic suspended cable-driven parallel robot

Algorithm 1 Calculate Cable Unit Vectors

ComputeUnitVectors**P**, **Q** $n \leftarrow$ length of **P** Initialize vector **U** of length n $i \leftarrow 1$ to n $\mathbf{L}_i \leftarrow \mathbf{Q}_i - \mathbf{P}_i$ Calculate cable vector $\ell_i \leftarrow \sqrt{(\mathbf{L}_i \cdot \mathbf{L}_i)}$ Magnitude of $\mathbf{L}_i$ $\mathbf{U}_i \leftarrow \mathbf{L}_i/\ell_i$ Normalize the vector **return U**

3.4.1 Explanation

- The function ComputeUnitVectors takes two lists of vectors, **P** and **Q**, representing the coordinates of cable attachment points on the fixed frame and the moving platform, respectively.
- The vector $\mathbf{L}_i$, representing the i-th cable, is calculated by subtracting $\mathbf{P}_i$ from $\mathbf{Q}_i$.
- The magnitude ℓ_i of each cable vector $\mathbf{L}_i$ is then computed.
- Each vector $\mathbf{L}_i$ is normalized by dividing by its magnitude to compute the unit vector $\mathbf{U}_i$.
- The algorithm returns a list of unit vectors **U**, which are used in the control system of the CDPR to ensure precise movement.

The precise calculation and understanding of these unit vectors are essential for the effective operation of CDPR systems. They allow for the accurate control of the end-effector's path and orientation, crucial in applications requiring high precision and reliability.

3.5 Experimental Design for a Real-World Cable Suspended Parallel Robot (CDPR) Application

Using industrial-grade components to highlight its feasibility in real-world industrial applications, this section explores the experimental setup meant to evaluate a recovery technique using a Cable Suspended Parallel Robot (CSPR).

3.5.1 Control System Overview

The Programmable Logic Controller (PLC), a mainstay of automation for its dependability, economy, and continuous operating performance, drives our automated system at its core. Our concept moves from conventional hardware-based PLCs to more flexible soft-PLCs running on regular PCs utilising software to replicate classic PLC operations. This change has various benefits:

Soft-PLCs are quite flexible in construction, which helps them to be very suitable for complex industrial uses. They offer strong network connectivity features necessary in modern industrial environments. Enhanced processing capacity allows soft-PLCs to effectively manage more data quantities, hence improving the responsiveness of the system.

To create a strong real-time controller, we use TwinCAT3, set on a Windows 10 PC in kernel mode [5]. The combination of TwinCAT3 with Microsoft Visual Studio makes it possible to create automation modules leveraging both advanced C/C++ programming languages and standard IEC 61131-3.

3.5.2 Integration Using MATLAB Simulink

Integration of TwinCAT with MATLAB Simulink is fundamental in our system [9]. This configuration enables the direct integration of a Simulink created control model, therefore enabling the development of complex control systems anchored on strong computational models and control theory. Verified inside our testing setup, this integration greatly increases the accuracy and adaptability of the system. TwinCAT effectively fills in MATLAB's overall unsuitability for real-time applications resulting from its high-level character. It generates C++ code from Simulink's block diagrams,as shown in figure:3.8 therefore conserving the required computing capacity for real-time operations.

EtherCAT Message Customised especially for real-time industrial automation, our solution has an integrated EtherCAT interface for device communication. EtherCAT dynamically manages Ethernet data frames inside a master-slave network configuration, therefore simplifying data administration. This arrangement speeds up and improves network communication efficiency. Managed by

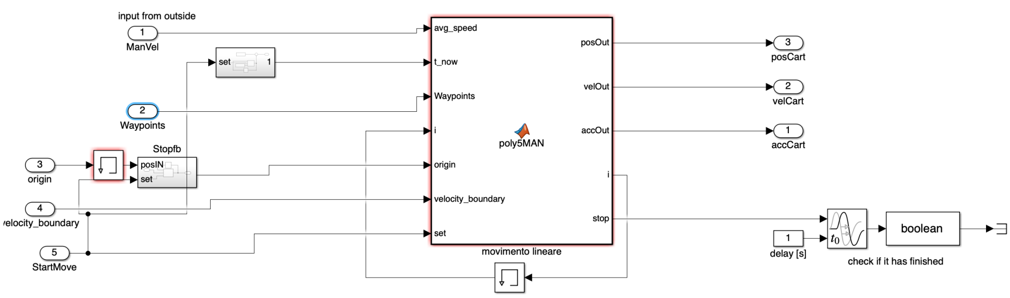

Figure 3.8: Integration Using MATLAB Simulink

TwinCAT, the EtherCAT master checks communications with servo drives to guarantee actuators are accurately guided depending on torque orders obtained from the Simulink trajectory planner. To guarantee strong and dependable connections, we build a very efficient EtherCAT master device using an industrial PC Ethernet controller—like the Intel PRO100. Our experimental design not only shows the feasible application of advanced control techniques inside a CSPR system but also emphasises the flawless interaction of contemporary software tools with conventional industrial components. By increasing the system's adaptability, operating efficiency, and precision, this synergy positions it as a model framework for innovative industrial automation initiatives.

3.6 Control Algorithms

A system will run better if appropriate high-level control algorithm is used. PID controllers were proposed as numerous control solutions for cable-driven parallel robots [2]. The PID control scheme is the most widely used control method in various applications especially in the industry because in significant part of its simplicity. Still, as reported in [8], changing the PID controller gains in the Cartesian cable robot proved difficult. Since they adjusted the PID especially for each of the three Cartesian axes both for translations and rotations, this method is more difficult than with a PID in joint space. Furthermore, even if the gains are weltuned, the PID controller exhibits poor resilience capacity in case of many problems.

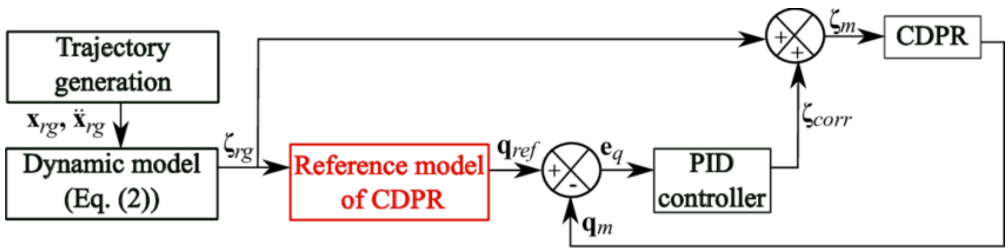

Figure 3.9: Feed-forward-model-based-PID-control

Advanced Control Methods for Cable-Driven Parallel Robots: Researchers in the field of robotics have made significant advancements in controlling cable-driven parallel robots, particularly in challenging environments. Among these, Robust PID control strategies stand out, especially for translational systems operating under zero-gravity conditions. These strategies are critical in space robotics and other applications where precision and reliability are of utmost importance.

Robust Point-to-Point Control:The Robust Point-to-Point (PTP) control strategy is designed to tackle the uncertainties that arise when robots operate in environments where actuator positions might be imprecise—such as during reassembly or in outdoor settings. These inaccuracies can lead to errors in the tension of the wires, adversely affecting the robot's positioning accuracy. By implementing adaptive compensation, this method adjusts the forces exerted by each wire, ensuring the robot's movements remain precise despite these potential inaccuracies. This approach effectively mitigates issues associated with variations in the Jacobian matrix caused by actuator placement errors, using external sensors to ensure exact positioning.

Adaptive Proportional-Derivative (PD) Control:

In an effort to further refine control strategies, researchers have developed an adaptive Proportional-Derivative (PD) controller that operates in task-space rather than joint-space, showing enhanced performance. This controller guarantees the accurate monitoring of the end-effector's position and orientation by employing quaternion algebra, making it robust against the model's complexities.

Linear Quadratic (LQ) Optimal Control:Linear Quadratic (LQ) optimal control represents another advanced method that has not been widely used in cable driven parallel robots but holds significant potential.This technique leverages feedback from all system states including velocity and position—to finely tune control actions. It allows for a straightforward comparison of cable robots with their rigidly linked counterparts, optimizing performance across various metrics. Habibnejad and colleagues have applied LQ optimal control in experimental settings on both 2-DOF limited planar and 6-DOF under-constrained cable robots.[10]. Their research demonstrates notable improvements in dynamic load-carrying capacity and optimized control gains, leading to excellent performance in motor torques and tracking errors. These developments underscore the progressive nature of control strategies in the realm of cable-driven parallel robots, paving the way for high precision and efficiency in complex robotic tasks. The specific information about the robotic workcell, such as the mass m_e and radius r_e of the end effector, can be found in table: 3.1. When constructing a recovery method for a cable robot, it is common to represent the end effector as a point mass. This modelling approach simplifies both the study of the motion and forces of the system, making it more efficient in achieving its main goal of preventing collisions with the end effector. In our configuration, the end effector is symbolised by a steel ball that is fitted with a ring for the purpose of attaching the cables as shown in figure:3.4. Despite the fact that this arrangement causes a minor displacement of the cable attachment points from the centre of mass of the end effector, it is considered an acceptable compromise considering the current technology limitations.

Table 3.1: Size of the robot workcell and of the end-effector

Parameter	**Value**	**Unit**
a	1775	mm
b	1690	mm
h	1950	mm
m_e	2.941	kg
r_e	90	mm

The characteristics of the adopted actuators are presented in Table 3.2 below. Moreover, the considered actuators are equipped with resolvers to monitor both the actuator position and velocity. The velocity data allow the implementation of the three terms of the PID control algorithm without deriving the position information; this is a great advantage since it is not necessary to define a filter required to define a filter for the position data, which is necessary for the derivate [6]. The presented system represents the setup of typical industrial applications fairly closely, since it has been designed with industrial-grade components. Moreover, the proposed control system does not require additional sensors and exploits those embedded in the actuators.

Table 3.2: Characteristics of the adopted actuators

Actuator Property	**Value**	c **Unit**
Nominal speed n_n	3000	rpm
Number of pole pairs	4	-
Nominal torque τ_n	1.3	Nm
Nominal current i_n	5.8	A
Stall torque τ_0	1.35	Nm
Stall current i_0	6	A
Maximum torque τ_{max}	4	Nm
Maximum current i_{max}	20.7	A
Maximum speed n_{max}	6600	rpm
Torque constant k_t	0.23	Nm/A
Voltage constant k_e	13.61	1000/rpm
Moment of inertia J	5.12×10^{-4}	kg·m^2
Braking torque τ_{br}	2.2	Nm
Gear ratio I	4	-
Drum radius ρ	36	mm
Static friction torque	0.2	Nm
Friction coefficient b	0.0015	-

3.7 Optimization Techniques in Trajectory Planning

In trajectory planning for a suspended ball robot, optimization techniques play a crucial role in ensuring that the trajectory is smooth, feasible, and satisfies various constraints such as cable tensions and physical limits. The goal of trajectory planning is to determine the optimal path $\mathbf{p}(t)$ for the ball such that the ball moves from an initial position $\mathbf{p}(0)$ to a target position $\mathbf{p}_{\text{target}}$ while minimizing an objective function and satisfying constraints.

1. Minimize Energy or Force: Ensure efficient use of cable tensions.

$$\min \int_0^T \sum_{i=1}^{m} f_i(t)^2 \, dt \tag{3.16}$$

2. Minimize Path Deviation: Ensure the ball follows a desired path closely.

$$\min \int_0^T \|\mathbf{p}(t) - \mathbf{p}_{\text{desired}}(t)\|^2 \, dt \tag{3.17}$$

3. Minimize Jerk: Ensure smooth motion by minimizing the rate of change of acceleration.

$$\min \int_0^T \|\dot{\mathbf{a}}(t)\|^2 \, dt$$

4. Dynamic Constraints: Ensure the trajectory adheres to the system dynamics.

$$\mathbf{F}(t) = m\mathbf{a}(t) + \mathbf{W} \tag{3.18}$$

5. Cable Tension Limits: Ensure cable tensions remain within allowable bounds.

$$f_{\text{min}} \le f_i(t) \le f_{\text{max}} \tag{3.19}$$

6. Kinematic Constraints: Ensure the ball stays within a safe workspace.

$$\mathbf{p}_{\text{min}} \le \mathbf{p}(t) \le \mathbf{p}_{\text{max}} \tag{3.20}$$

7. Initial and Final Conditions: Ensure the ball starts and ends at specified positions and velocities.

$$\mathbf{p}(0) = \mathbf{p}_{\text{start}}, \quad \mathbf{p}(T) = \mathbf{p}_{\text{target}} \tag{3.21}$$

Several optimization techniques can be used for trajectory planning. Each has its advantages depending on the complexity of the problem and the required solution quality.

3.8 Trajectory Planning Fundamentals

Beginning with a broad classification based on the ratio of degrees of freedom to cables, we explore the nuances of cable-driven robots in this chapter. Understanding the special qualities of many configurations depends on this classification [4]. As was already said, cable robots are basically parallel manipulators in which cables substitute stiff linkages. But the literature lacks a generally accepted taxonomy that would help to explain different terminology across sources.

Cable robots fall first into planar and spatial categories. Whereas spatial robots negotiate a three-dimensional workspace, planar robots operate within a specified plane. Furthermore, if a planar or spatial cable robot just translates without rotational motion at the end-effector, it can be translational as well.

Considering the link between degrees of freedom and cable count helps us to classify things more precisely. Cable robots are completely actuated when the number of active cables corresponds with the degrees of freedom, much as parallel manipulators are. On the other hand, under-actuated robots have less cables while redundant or duplicated actuated robots have more cables than degrees of freedom.

Moreover, depending on the robot's capacity to limit the end effector, we differentiate under-constrained from fully-constrained designs.To ensure equilibrium against outside influences, fully confined configurations usually ask for more wires than degrees of freedom.

Turning now to kinematic analysis, let us examine a cable robot with m cables controlling n degrees of freedom as shown in Fig. 3.6. Kinematics of the system depend critically on each pulley anchor point (Ai) and attachment point on the moving platform (Bi). Calculating the length of each cable (Li) given the known position of the centre of mass (xG) presents the inverse kinematic problem. Conversely, based on the actuation condition of the system, the direct kinematic problem finds the centre of mass position from known cable lengths. We investigate the relationship between the velocity of the end effector and the velocity of cables being reeled or retracted in terms of velocity analysis. Crucially important for knowledge of how cable velocities affect end effector motion is the kinematic velocity analysis establishes this link. Fundamental for motion control and safety techniques, this study provides understanding of the forces required to reach desirable motion profiles.

Parallel Configuration: Usually consisting of several cables attached to a centre ball-like end effector, a ball robot is configured parallelly. Every cable is fixed at a spot and under control to move the ball.

Positions Equations: By means of cable lengths (Li) and anchor point placements (Ai), one may determine the central ball (B) position. Taking a Cartesian coordinate system, the ball's position vector (B) can be expressed as:

$$\vec{B} = \sum_{i=1}^{m}(L_i \cdot \hat{u}_i) + \vec{A}_0 \tag{3.22}$$

where: - m is the number of cables. - L_i is the length of the i-th cable. - $\hat{u}_i$ is

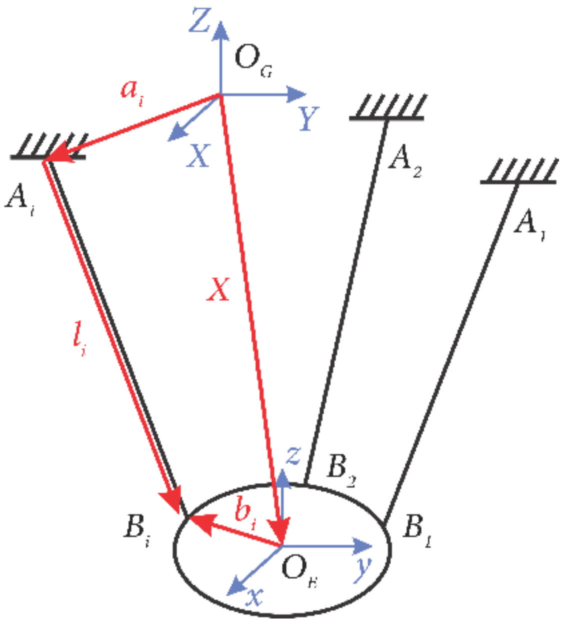

Figure 3.10: Kinematic model of a cable driven spatial robot

the unit vector along the direction of the i-th cable. - $\vec{A}_0$ is the position vector of the central anchor point.

Velocity Equations: The velocity of the ball (B) can be calculated based on the velocities of the cables being reeled or retracted.

$$\vec{V}_B = \sum_{i=1}^{m} (\dot{L}_i \cdot \hat{u}_i) \tag{3.23}$$

where: - $\dot{L}_i$ is the velocity of the i-th cable being reeled or retracted.

The wires in an above arrangement are set to suspend the central ball-like end effector from above. Applications where the robot must roam freely inside a workspace can have this arrangement.

Position Equations Using the cable lengths (Li) and the anchor point positions (Ai), one may determine the central ball (B) position similarly to in the parallel arrangement. In this situation, nevertheless, the anchor points lie above the ball as shown in figure: 3.7.

$$\vec{B} = \sum_{i=1}^{m} (L_i \cdot \hat{u}_i) + \vec{A}_0 \tag{3.24}$$

where: - m is the number of cables.

- L_i is the length of the i-th cable.

- $\hat{u}_i$ is the unit vector along the direction of the i-th cable.

- $\vec{A}_0$ is the position vector of the central anchor point located above the ball.

A key component of robotics, trajectory planning helps a robot to travel from a starting position to a desired objective under control of dynamics, kinematic restrictions, and constraints including obstructions. To provide smooth and exact motion, the robot's actuators must follow a sequence of locations, velocities, and accelerations computed here. Ensuring that these trajectories are practical, safe, and efficient depends on the mathematical underpinnings and algorithms for trajectory planning.

The foundation of trajectory planning is rooted in differential calculus and optimization.The goal is to determine a time-parametrized path $\mathbf{P}(t)$, where $\mathbf{P}(t)$ is the position vector of the robot at time t.The primary components of trajectory planning include:

1. Kinematic Equations: These equations describe the robot's motion without considering forces. For a point $\mathbf{P}(t)$ in a 3D space:

$$\mathbf{P}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} \tag{3.25}$$

The velocity $\mathbf{V}(t)$ and acceleration $\mathbf{A}(t)$ are given by:

$$\mathbf{V}(t) = \frac{d\mathbf{P}(t)}{dt}, \quad \mathbf{A}(t) = \frac{d\mathbf{V}(t)}{dt} \tag{3.26}$$

2. Dynamic constraints: These include limits on velocities, accelerations, and jerks (rate of change of acceleration) to ensure smooth motion. For example:

$$|\mathbf{V}(t)| \leq V_{max}, \quad |\mathbf{A}(t)| \leq A_{max}, \quad |\frac{d\mathbf{A}(t)}{dt}| \leq J_{max} \tag{3.27}$$

3. Path restrictions: Obstacles and environmental constraints are modeled to ensure that the path is collision-free. These are often represented by inequalities:

$$\mathbf{C}(\mathbf{P}(t)) \geq 0 \tag{3.28}$$

4. Objective Function: The objective function J to be minimized often represents the total travel time, energy consumption, or a combination of multiple criteria:

$$J = \int_{t_0}^{t_f} \mathcal{L}(\mathbf{P}(t), \mathbf{V}(t), \mathbf{A}(t))\, dt \tag{3.29}$$

where $\mathcal{L}$ is a cost function.

4

Cubic Trajectory Generation of CDPRs

4.1 Definition and Mathematical Model

The formation of cubic trajectories enables perfect and exact motion of the end effector, so controlling parallel ball robots is mainly based on this development of cubic trajectories.This chapter offers a thorough and precise guide for building trajectories for a Cable-Driven Parallel Robot (CDPR) as shown in figure:4.1. We will discuss building cubic trajectories considering restrictions on Cartesian acceleration and velocity. We will also look at path evolution with quintic-spline curves.The robot, equipped with four motors, achieves three degrees of freedom (3DOF) through a PID-controlled system [22].The approach ensures smooth and precise motion, using boundary conditions for velocity and acceleration at the waypoints.[17].

4.2 Algorithm Development

Determine a continuous and smooth path for the end effector of a Cable-Driven Parallel Robot (CDPR) to move from an initial position to a final position within a defined period, so preserving that the linear velocity and acceleration of the end effector remain within normal boundaries.

1. Cubic Generation: The cubic polynomial representing the trajectory is given by:

$$\mathbf{p}(t) = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3 \tag{4.1}$$

where $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$, and $\mathbf{a}_3$ are the coefficients that need to be determined.

Boundary Conditions:To determine the coefficients, we need four boundary conditions:

1. **Initial Position**:

$$\mathbf{p}(0) = \mathbf{a}_0 = \mathbf{p}_0 \tag{4.2}$$

2. **Final Position**:

$$\mathbf{p}(T) = \mathbf{a}_0 + \mathbf{a}_1 T + \mathbf{a}_2 T^2 + \mathbf{a}_3 T^3 = \mathbf{p}_T \tag{4.3}$$

3. **Initial Velocity**:

$$\mathbf{v}(0) = \mathbf{p}'(0) = \mathbf{a}_1 = \mathbf{v}_0 \tag{4.4}$$

4. **Final Velocity**:

$$\mathbf{v}(T) = \mathbf{p}'(T) = \mathbf{a}_1 + 2\mathbf{a}_2 T + 3\mathbf{a}_3 T^2 = \mathbf{v}_T \tag{4.5}$$

Given the boundary conditions, we can form a system of equations to solve for the coefficients of the cubic polynomial. By substituting the known values into these equations, we can solve for $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$. These coefficients define the cubic polynomial that describes the trajectory.

System of Equations: By solving the above equations, we find the coefficients as follows:

1. **Coefficient $\mathbf{a}_0$**:

$$\mathbf{a}_0 = \mathbf{p}_0 \tag{4.6}$$

2. **Coefficient $\mathbf{a}_1$**:

$$\mathbf{a}_1 = \mathbf{v}_0 \tag{4.7}$$

3. **Coefficient $\mathbf{a}_2$**:

$$\mathbf{a}_2 = \frac{3(\mathbf{p}_T - \mathbf{p}_0) - (2\mathbf{v}_0 + \mathbf{v}_T)T}{T^2} \tag{4.8}$$

4. **Coefficient $\mathbf{a}_3$**:

$$\mathbf{a}_3 = \frac{2(\mathbf{p}_0 - \mathbf{p}_T) + (\mathbf{v}_0 + \mathbf{v}_T)T}{T^3} \tag{4.9}$$

2: Quintic Spline Curve Generation

To generate a smooth path for the end-effector using quintic spline curves, we ensure smooth transitions with continuous acceleration and velocity. [15] The parameters are as follows:
- Initial position: $\mathbf{p}_0 = [p_{0x}, p_{0y}, p_{0z}]$
- Final position: $\mathbf{p}_T = [p_{Tx}, p_{Ty}, p_{Tz}]$

- Initial velocity: $\mathbf{v}_0 = [v_{0x}, v_{0y}, v_{0z}]$ (often 0)
- Final velocity: $\mathbf{v}_T = [v_{Tx}, v_{Ty}, v_{Tz}]$ (often 0)
- Initial acceleration: $\mathbf{a}_0 = [a_{0x}, a_{0y}, a_{0z}]$ (often 0)
- Final acceleration: $\mathbf{a}_T = [a_{Tx}, a_{Ty}, a_{Tz}]$ (often 0)
- Total time: T

For each coordinate (x, y, z), a quintic polynomial $p_i(t)$ is represented as:

$$p_i(t) = a_{0i} + a_{1i}t + a_{2i}t^2 + a_{3i}t^3 + a_{4i}t^4 + a_{5i}t^5 \tag{4.10}$$

To determine the coefficients $a_{0i}, a_{1i}, a_{2i}, a_{3i}, a_{4i}, a_{5i}$, we need six conditions for each coordinate:

1. $p_i(0) = p_{0i}$ (initial position) 2. $p_i(T) = p_{Ti}$ (final position) 3. $v_i(0) = p_i'(0) = v_{0i}$ (initial velocity) 4. $v_i(T) = p_i'(T) = v_{Ti}$ (final velocity) 5. $a_i(0) = p_i''(0) = a_{0i}$ (initial acceleration) 6. $a_i(T) = p_i''(T) = a_{Ti}$ (final acceleration)

Using these boundary conditions, we derive the following system of equations for each coordinate i:

$$\begin{cases} a_{0i} = p_{0i} \\ a_{1i} = v_{0i} \\ a_{2i} = \frac{3(\mathbf{p}_T - \mathbf{p}_0)}{T^2} - \frac{2\mathbf{v}_0 + \mathbf{v}_T}{T} \\ a_{3i} = \frac{2(\mathbf{p}_0 - \mathbf{p}_T)}{T^3} + \frac{\mathbf{v}_0 + \mathbf{v}_T}{T^2} \end{cases} \tag{4.11}$$

These equations provide the coefficients a_0, a_1, a_2, and a_3 necessary for our analysis based on the given boundary conditions.Using these boundary conditions, we derive the following system of equations for each coordinate i:

$$\begin{cases} a_{0i} = p_{0i} \\ a_{1i} = v_{0i} \\ a_{2i} = \frac{1}{2}a_{0i} \\ a_{3i} = \frac{20p_{Ti} - 20p_{0i} - (8v_{Ti} + 12v_{0i})T - (3a_{0i} - a_{Ti})T^2}{2T^3} \\ a_{4i} = \frac{30p_{0i} - 30p_{Ti} + (14v_{Ti} + 16v_{0i})T + (3a_{0i} - 2a_{Ti})T^2}{2T^4} \\ a_{5i} = \frac{12p_{Ti} - 12p_{0i} - (6v_{Ti} + 6v_{0i})T - (a_{0i} - a_{Ti})T^2}{2T^5} \end{cases} \tag{4.12}$$

By solving this system of equations, we obtain the coefficients for the quintic polynomials that describe the end-effector's trajectory.[19] This approach ensures smooth, continuous motion with proper acceleration and velocity transitions.

The coefficients of the quintic polynomial can be determined by setting up the following linear system derived from the boundary conditions at points t_0 and t_1:

$$\begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 0 & 0 & 0 & 6 & 24t_0 & 60t_0^2 \\ 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} p_0 \\ p_0' \\ p_0'' \\ p_0''' \\ p_1 \\ p_1' \end{pmatrix}$$

This matrix system can be solved to find the coefficients $a_0, a_1, a_2, a_3, a_4, a_5$, which define the cubic polynomial for the trajectory segment.

The precise integration of these quintic spline curves into the control system of the Cable-Driven Parallel Robot (CDPR) enables accurate trajectory generation and movement of the end-effector in 3D space.

4.3 Implementation Cable-Driven Parallel Robots

Cable-Driven Parallel Robots (CDPRs) are known for their large workspace and high payload capacity. For applications requiring precise movements, accurate trajectory generation and control are crucial. This thesis presents a cubic trajectory generation method considering boundary conditions for velocity and acceleration, implemented on a suspended CDPR.[1]. A PID control strategy is used to track the generated trajectory, validated through experimental results. Fixed to a fixed reference frame, assigned F_b, the CDPR system is attached to the robot's base structure with its origin at point O).

Motivated by a moving frame, $F_p$the cube-shaped end effector moves from point P.The change between the stationary frame F_b and the moving frame F_p can be precisely conveyed by means of a rotation matrix R.By guiding the coordinates and movements between the two frames to be translated, this matrix ensures proper positioning and orientation control.

Cable lengths and their unit vectors define fundamental parameters obtained from the loop closure equation in CDPR management.This equation takes into account the positions of the cable exit and anchor points, thereby directing the

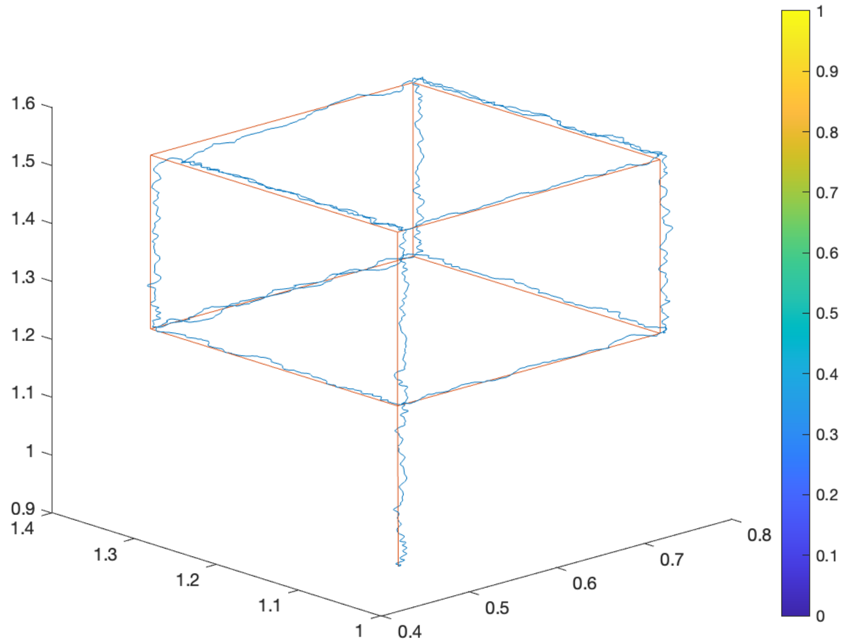

Figure 4.1: Cubic Trajectory of CDPR

precise length each cable must extend or retract to appropriately position the end effector. The unit vectors of these cables determine the forces each wire creates; hence, they also define their directions and ensure that the end-effector follows the desired trajectory and stability. Thorough knowledge of these cable lengths and unit vectors as well as the shift between frames will enable the CDPR to operate exactly and regulate intricate movements with remarkable accuracy.

Forward Kinematics Forward kinematics calculates the position of the end-effector based on the lengths of the cables. Given the current cable lengths, it determines the precise position of the end-effector in space.

MPU Data Fusion MPU Data Fusion combines data from multiple sensors to provide an accurate estimate of the end-effector's position and orientation. This involves using a Microprocessor Unit (MPU) to integrate sensor data for reliable position estimation.

Control Strategy for Trajectory Generation Accurate positioning of the end-effector in a Cable-Driven Parallel Robot (CDPR) is achieved through a PID control system [18].This system minimizes the error between the end-effector's actual position (x_e) and the intended position (x_d). The error is calculated as $e_x = x_d - x_e$.

To optimize the controller's efficiency, the proportional gain (K_p), integral gain (K_i), and derivative gain (K_d) are adjusted. Through experimentation,

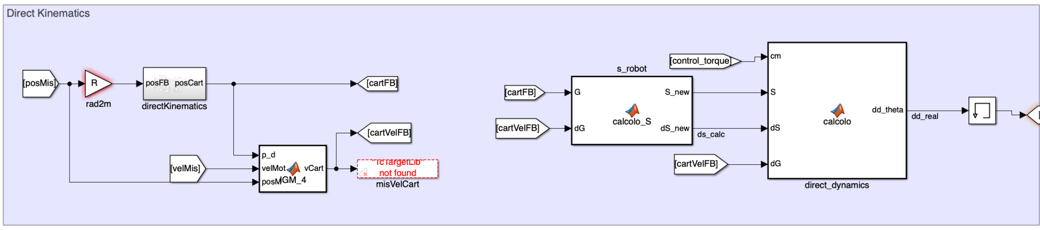

Figure 4.2: Control Sytem Block Diagram

CDPR was found to perform best with $K_p = 2.4$, $K_i = 0.3$, and $K_d = 0.1$. These gains were chosen to balance fast response and system stability, ensuring the end-effector can accurately and smoothly follow the desired trajectory with minimal overshooting or steady-state error. By integrating the PID control system as show in figure:4.2 with cubic polynomial trajectory generation, the CDPR achieves minimal positioning errors and precise control over the end-effetor movements in 3D space. The continuous adjustment of actuator commands, informed by accurate sensor data and the chosen PID gains, ensures the end-effector follows the desired path with high accuracy and stability.

4.4 Experimental Validation of CDPRs

The results confirmed the effectiveness of the proposed control algorithm and the robot's ability to perform precise movements.

To understand the behavior of a Cable-Driven Parallel Robot (CDPR) with respect to applied forces and moments, we need to analyze its dynamics. This system consists of four cables attached to a cube-shaped end-effector.[27] Each cable is connected to a spindle via pulleys, forming the mechanism that controls the movement of the end-effector. The analysis involves several key components:

4.4.1 Coordinate Frames

- **Fixed Frame** F_b:
  - **Origin** O: This is the reference frame from which all measurements are taken. It is fixed in space.
- **Moving Platform Frame** F_p:

  - **Origin** P: This frame moves with the end-effector (the cube), and its position changes as the robot operates.

- **Transformation Between Frames** To express the relationship between the fixed frame F_b and the moving frame F_p, we use a rotation matrix R.This matrix allows us to transform coordinates and orientations from one frame to another.Essentially, it helps in understanding how the cube's orientation changes relative to the fixed frame.

- **Cable Lengths and Directions**

  - **Cable Dynamics** l_i: The length of each cable l_i is computed using the loop closure equation. This equation considers the positions of the exit points (where cables leave the pulleys) and the anchor points (where cables attach to the cube) in both frames.

- **Unit Vector** $\hat{l}_i$: $\hat{l}_i$ represents the direction of the cable l_i as a unit vector. It indicates the cable's orientation in space, which is crucial for calculating forces.

4.4.2 Dynamic Equilibrium and Forces

To maintain dynamic equilibrium, we need to account for all forces and moments acting on the CDPR. This is represented using the wrench matrix W. The following components are involved:

- **External Wrench** w_e:
  This term represents the combined forces and moments due to external influences, primarily the weight of the platform (cube), which is affected by gravitational acceleration g.**Equation of Motion**:The forces in the cables produce a net force and moment that must balance the external forces. This relationship is given by:

$$W\tau = m\ddot{p} - mg \tag{4.13}$$

  where:

  - * W: Wrench matrix.
    * τ: Cable tensions.
    * m: Mass of the platform.
    * $\ddot{p}$: Acceleration of the platform.
    * g: Gravitational acceleration.

4.5 Simulation Results of Cubic Trajectory Generation

1. ***Trajectory Tracking***

- To assess how accurately the end-effector follows the predefined cubic trajectory.
- The desired trajectory is generated using cubic polynomials between specified waypoints.The PID controller adjusts cable lengths to follow this trajectory.
  Results: The end-effector precisely tracked the cubic trajectory with negligible variance. Minor deviations in position were noticed during quick bends or sudden changes in direction, however, these deviations remained within acceptable boundaries as a result of the PID tuning.

2. ***Error Analysis***

- To evaluate the positional error throughout the trajectory.
- Compute the error as the difference between the desired and actual positions of the end-effector.

Results: The positional error was consistently minimal throughout the trajectory, indicating accurate tracking of the desired path. The end-effector maintained close alignment with the intended positions, demonstrating the effectiveness of the curve generation and the PID control system in minimizing deviations.

3. Velocity and Acceleration Profiles

- To analyze the smoothness of the end-effector's motion.
- Track velocity and acceleration over time and compare with expected profiles

Results: The velocity profiles were smooth and followed the desired changes, indicating effective PID control. Acceleration peaks were within limits, suggesting no abrupt or jerky movements.

4.6 Comparative Analysis and Performance Evaluation of Cubic Trajectory

The simulation was performed using MATLAB R2021a,Simulink, custom software).The results showing how closely the CDPR followed the cubic trajectory. Using graphical plots to compare the desired and actual paths as below.

4.6.1 Positon of Cubic Anlaysis of CDPRs

Measuring the root mean square error (RMSE) between the intended position quantifies the spatial accuracy. Figure: [5.3] and real measured locations Figure:[4.4] underneath. The low RMSE value shows that the CDPR follows the cubic trajectory with minimum variation rather closely.System dynamics and small control system delays could help to explain observed modest variances.

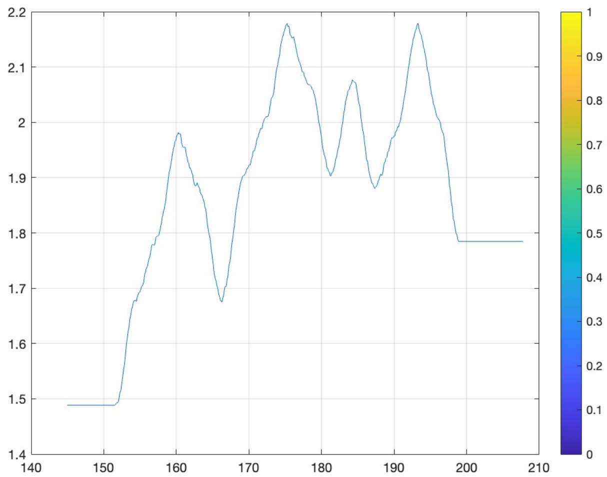

Figure 4.3: Measured Cartesian Position of Cubic CDPR

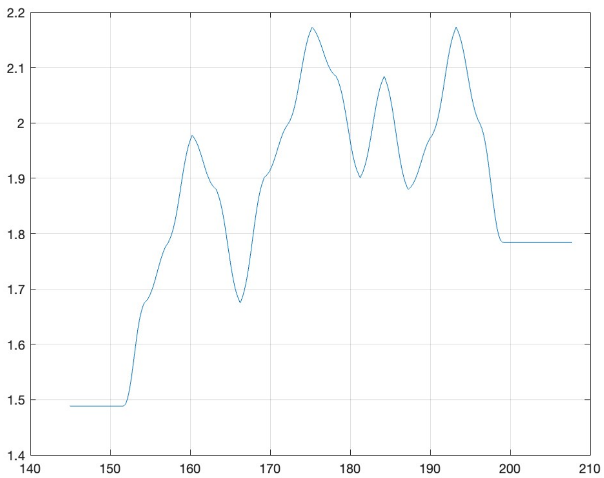

Figure 4.4: Referenced Cartesian Position of Cubic CDPR

4.6.2 Angular Velocity of Cubic Anlaysis of CDPRs

$$\mathrm{RMSE}_{\Delta\theta} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\theta}'_i - \theta'_i)^2} \tag{4.14}$$

The calculated RMSE for$\Delta\theta$ is found to be within acceptable limits, confirming the system's robustness in maintaining the desired angular velocity. This performance metric underscores the control system's effectiveness in minimizing deviations and ensuring smooth rotational motion.

4.6.3 Velocity of Cubic Anlaysis of CDPRs

The comparison of velocity profiles reveals that the actual velocities (Figure 4.6) closely match the desired velocities (Figure 4.5) throughout the trajectory. This indicates that the control system effectively manages the acceleration and deceleration phases, ensuring smooth transitions. Any discrepancies are minimal and are within acceptable bounds for operational performance.

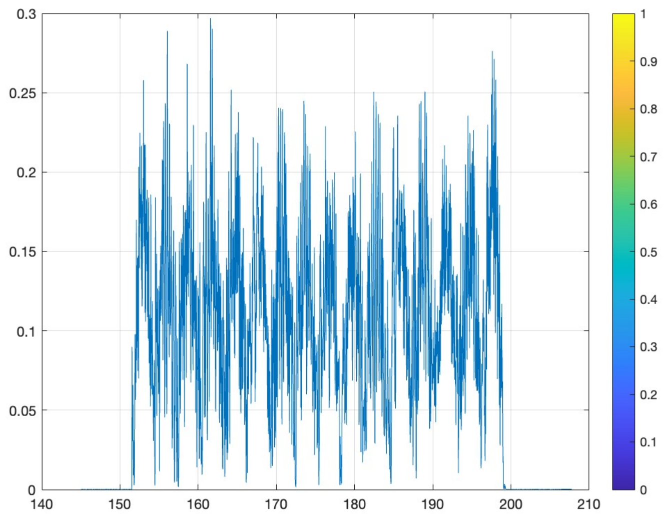

Figure 4.5: Measured Cartesian Velocity of Cubic CDPR

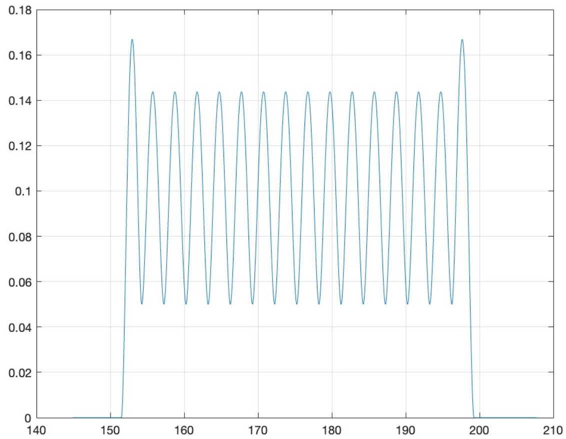

Figure 4.6: Referenced Cartesianof Cubic CDPR

4.6.4 Torque of Cubic Anlaysis of CDPRs

The torque analysis shows that the measured torque Figure[4.7] to follow the cubic trajectory remains within the system's capabilities. The actual torque Figure[4.8] profiles align well with the expected values, demonstrating that the control system can handle the dynamic loads efficiently without inducing excessive stress on the system components.

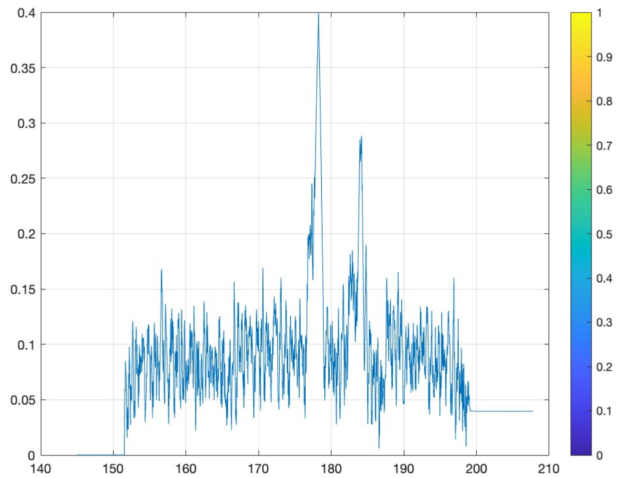

Figure 4.7: Measured Torque of Cubic CDPR

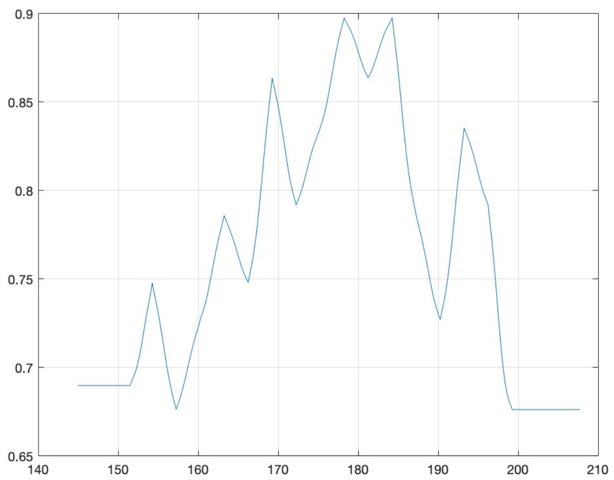

Figure 4.8: Referenced Torque of Cubic CDPR

5

Spiral Trajectory Generation of CDPRs

5.1 Introduction

Trajectory generation is crucial for the precise and smooth operation of Cable-Driven Parallel Robots (CDPRs). This chapter focuses on developing a spiral trajectory for a 3DOF CDPR figure:5.1. The spiral trajectory is chosen due to its complexity, which serves as a robust test for the robot's control system. The trajectory is defined using parametric equations, waypoints, unit vectors, quintic spline interpolation, and is tracked using a PID control system.

5.2 Spiral Trajectory Formulation

A spiral trajectory in a three-dimensional space can be described using parametric equations. The equations for the spiral trajectory incorporate radial distance, angular position, and height variation.

5.2.1 Parametric Equations

The parametric equations for the spiral trajectory are given by:

$$\begin{cases} x(t) = r(t)\cos(\theta(t)) + p_{0x} \\ y(t) = r(t)\sin(\theta(t)) + p_{0y} \\ z(t) = z(t) + p_{0z} \end{cases} \tag{5.1}$$

Where: - $r(t) = r_0 + kt$ (radial distance as a function of time, with r_0 as the initial radius and k as the growth rate of the radius)
- $\theta(t) = \omega t$ (angular position as a function of time, with ω as the angular velocity)
- $z(t) = z_0 + ht$ (height variation as a function of time, with z_0 as the initial height and h as the height growth rate)
- p_{0x}, p_{0y}, p_{0z} are the initial position coordinates of the end-effector.

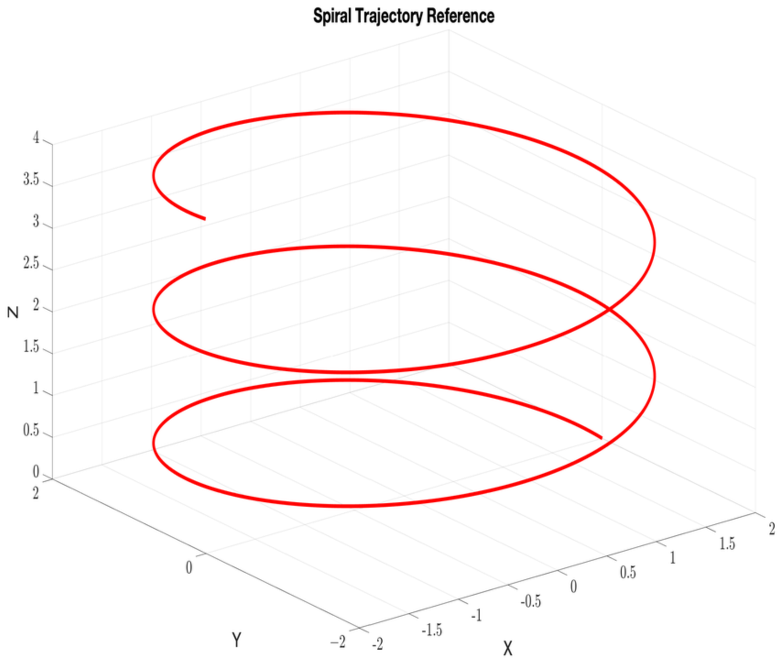

Figure 5.1: Spiral Trajectory for an CDPR

5.2.2 Waypoints Along the Spiral Path

To discretize the continuous trajectory for practical implementation, we define waypoints along the spiral path. The time interval $[0, T]$ is divided into N segments, and for each time step t_i, the corresponding waypoint position $\mathbf{p}_i = [p_{ix}, p_{iy}, p_{iz}]$ is computed as follows:

$$\mathbf{p}_i = \begin{cases} x(t_i) = (r_0 + kt_i)\cos(\omega t_i) + p_{0x} \\ y(t_i) = (r_0 + kt_i)\sin(\omega t_i) + p_{0y} \\ z(t_i) = z_0 + ht_i + p_{0z} \end{cases} \tag{5.2}$$

5.2.3 Unit Vectors Between Waypoints

For smooth trajectory generation, unit vectors between consecutive waypoints are calculated. The unit vector $\hat{\mathbf{u}}_i$ between waypoints $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ is:

$$\hat{\mathbf{u}}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|} \tag{5.3}$$

Where: - $\mathbf{p}_{i+1}$ and $\mathbf{p}_i$ are the positions of consecutive waypoints. - $\|\cdot\|$ denotes the Euclidean norm.

5.2.4 Trajectory Generation Using Quintic Spline Interpolation

To ensure smooth transitions and continuous acceleration and velocity profiles, quintic spline interpolation is used. For each coordinate i (x, y, z), the trajectory is described by a quintic polynomial:

$$p_i(t) = a_{0i} + a_{1i}t + a_{2i}t^2 + a_{3i}t^3 + a_{4i}t^4 + a_{5i}t^5 \tag{5.4}$$

The coefficients $a_{0i}, a_{1i}, a_{2i}, a_{3i}, a_{4i}, a_{5i}$ are determined by solving the following boundary conditions:

$$\begin{cases} p_i(0) = p_{0i} \\ p_i(T) = p_{Ti} \\ \dot{p}_i(0) = v_{0i} \\ \dot{p}_i(T) = v_{Ti} \\ \ddot{p}_i(0) = a_{0i} \\ \ddot{p}_i(T) = a_{Ti} \end{cases} \tag{5.5}$$

This leads to the system of equations:

lusion nclusion lusion nclusion

Figure 5.2: Trajectory of Spiral CDPR

$$\begin{cases} a_{0i} = p_{0i} \\ a_{1i} = v_{0i} \\ a_{2i} = \frac{1}{2}a_{0i} \\ a_{3i} = \frac{20p_{Ti}-20p_{0i}-(8v_{Ti}+12v_{0i})T-(3a_{0i}-a_{Ti})T^2}{2T^3} \\ a_{4i} = \frac{30p_{0i}-30p_{Ti}+(14v_{Ti}+16v_{0i})T+(3a_{0i}-2a_{Ti})T^2}{2T^4} \\ a_{5i} = \frac{12p_{Ti}-12p_{0i}-(6v_{Ti}+6v_{0i})T-(a_{0i}-a_{Ti})T^2}{2T^5} \end{cases} \tag{5.6}$$

5.3 PID Control System Implementation

To accurately follow the generated trajectory, a PID control system is implemented. The control system adjusts the cable lengths to minimize the positional error e_x:

$$e_x = x_d - x_e \tag{5.7}$$

where x_d is the desired position and x_e is the actual position. The control input $u(t)$ is given by:

$$u(t) = K_p e_x(t) + K_i \int e_x(t)dt + K_d \frac{de_x(t)}{dt} \tag{5.8}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

5.4 Simulation Results Spiral Trajectory Generation

The CDPR model is simulated with parameters corresponding to the physical dimensions and properties of the actual robot [24]. The control system utilizes the quintic spline curve generation and the PID control strategy described in previous sections.

5.4.1 Positional Error Analysis

- The positional error was consistently minimal throughout the trajectory, indicating accurate tracking of the desired path as shown in figure:[5.2]
- The end effector maintained close alignment with the intended positions, demonstrating the effectiveness of the quintic spline curve generation and the PID control system in minimizing deviations.

5.4.2 Velocity and Acceleration Profiles

- The velocity profiles were smooth and followed the desired changes, indicating effective PID control.
- Acceleration peaks were within limits, suggesting no abrupt or jerky movements.

5.5 Comparative Analysis and Performance Evaluation

Establishing a baseline for comparing referenced (theoretical or planned) and measured (actual) results of a spiral trajectory is crucial for any performance evaluation. This involves setting up a framework where both sets of data are analyzed against a set of predetermined criteria to understand discrepancies and improve system performance

5.5.1 Deriving Position of CDPRs

Creating a spiral trajectory for a robot, such as in a Cable-Driven Parallel Robot (CDPR) figure 5.3, 5.4 as seen below or any other robotic system that requires precise movement in three-dimensional space, involves calculating the position coordinates over time. A spiral trajectory often combines linear movement in one axis with circular motion in the plane perpendicular to that axis, which can be described using parametric equations:

$$r(t) = r_0 + kt \quad \text{where } r_0 \text{ is the initial radius, } k \text{ is the rate of increase in radius,its time.} \tag{5.9}$$

$$\theta(t) = \omega t \quad \text{where } \omega \text{ is the angular speed.} \tag{5.10}$$

$$z(t) = z_0 + ht \quad \text{where } z_0 \text{ is the initial height and } h \text{ is the rate of height increase.} \tag{5.11}$$

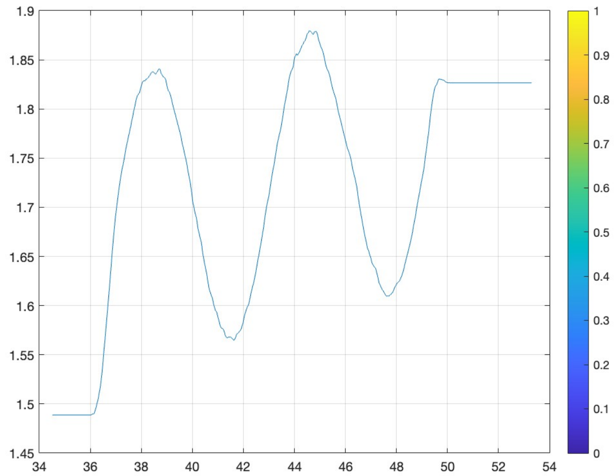

Figure 5.3: Measured Position of Spiral CDPR

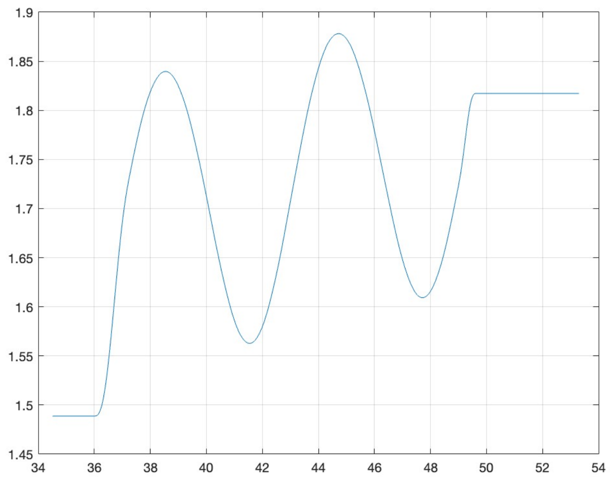

Figure 5.4: Referenced Position of Spiral CDPR

5.5.2 Deriving Velocity of CDPRs

The Cartesian coordinates (x, y, z) for the spiral trajectory, derived from the cylindrical coordinates of waypoints, we can derive the velocity components by differentiating these functions.

$$-x(t) = (r_0 + kt)\cos(\omega t) - y(t) = (r_0 + kt)\sin(\omega t) - z(t) = z_0 + ht \tag{5.12}$$

where:
$r(t) = r_0 + kt$ is the radial distance that increases linearly with time,
$\theta(t) = \omega t$ where ω is the constant angular speed,
$z(t) = z_0 + ht$ where h is the constant rate of change of height.

1.Velocity in the x-direction ($\dot{x}(t)$):

$$\dot{x}(t) = \frac{d}{dt}\left[(r_0 + kt)\cos(\omega t)\right] \tag{5.13}$$

Using the product and chain rules:

$$\dot{x}(t) = \frac{d}{dt}(r_0 + kt)\cos(\omega t) + (r_0 + kt)\frac{d}{dt}\cos(\omega t) \tag{5.14}$$

$$\dot{x}(t) = k\cos(\omega t) - \omega(r_0 + kt)\sin(\omega t) \tag{5.15}$$

2. Velocity in the y-direction ($\dot{y}(t)$):

$$\dot{y}(t) = \frac{d}{dt}\left[(r_0 + kt)\sin(\omega t)\right] \tag{5.16}$$

Similarly, applying the product and chain rules:

$$\dot{y}(t) = k\sin(\omega t) + \omega(r_0 + kt)\cos(\omega t) \tag{5.17}$$

3. Velocity in the z-direction ($\dot{z}(t)$):

$$\dot{z}(t) = \frac{d}{dt}(z_0 + ht) = h \tag{5.18}$$

Given the position functions for a spiral trajectory:

$$x(t) = (r_0 + kt)\cos(\omega t)$$
$$y(t) = (r_0 + kt)\sin(\omega t)$$
$$z(t) = z_0 + ht$$

The velocity components are derived as follows:

$$\dot{x}(t) = k\cos(\omega t) - \omega(r_0 + kt)\sin(\omega t)$$
$$\dot{y}(t) = k\sin(\omega t) + \omega(r_0 + kt)\cos(\omega t)$$
$$\dot{z}(t) = h$$

This representation highlights the changes in speed along the x, y, and z axes as functions of time, illustrating how the spiral's radial expansion and rotation contribute to the overall motion of the system. Such detailed derivations are crucial to understanding the dynamics of the trajectory, which can further aid in control system design and simulation as shown in figure: 5.5 5.6.

In this chapter, we closely monitor the angular position (θ) and the related torque values under controlled settings to assess the performance of Spiral Cable-Driven Parallel Robots (CDPRs). This part includes a detailed analysis of the torque applied by the system through a comparison between the measured and referenced θ values over a designated operational period.

5.5.3 Theta Measurements

The accuracy of the intended spiral trajectory in Spiral Cable-Driven Parallel Robots (CDPRs) is critically dependent on the angular position of the end-effector, which is denoted by θ. To assess the performance and precision of trajectory following, we compare the planned θ values—those outlined in the trajectory planning strategy with the observed θ values measured in real-time using precision sensors.

- **Planned Theta (θ_{planned})**: These values are derived from the trajectory planning algorithm and represent the ideal path that the end-effector should follow.
- **Observed Theta (θ_{observed})**: These are the actual values recorded during the operation of the robot, indicating the real-time behavior of the end-effector.

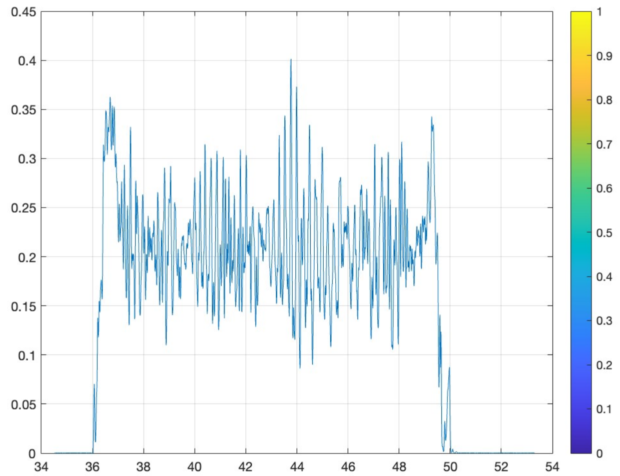

Figure 5.5: Measured Velocity of Spiral CDPR

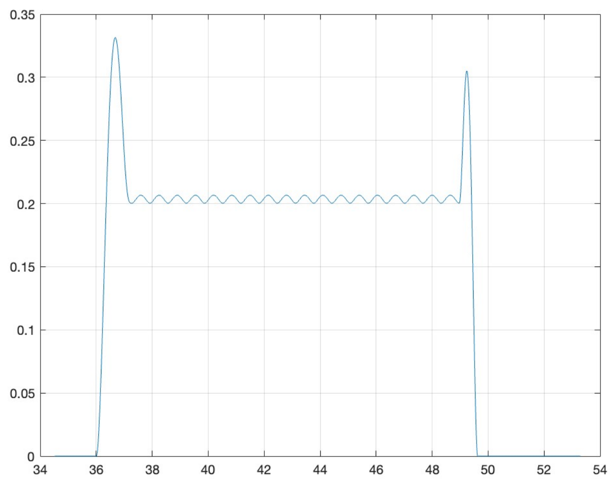

Figure 5.6: Referenced Velocity of Spiral CDPR

The comparison of these values is essential for understanding the discrepancies between the theoretical trajectory and actual path followed by the robot. This analysis helps in identifying potential areas of improvement in the control strategies or the mechanical setup of the robot to enhance the fidelity of trajectory following.

A graphical representation of θ_{ref} figure:5.7 versus θ_{meas} figure:5.8 over time is provided to illustrate the accuracy of trajectory following.

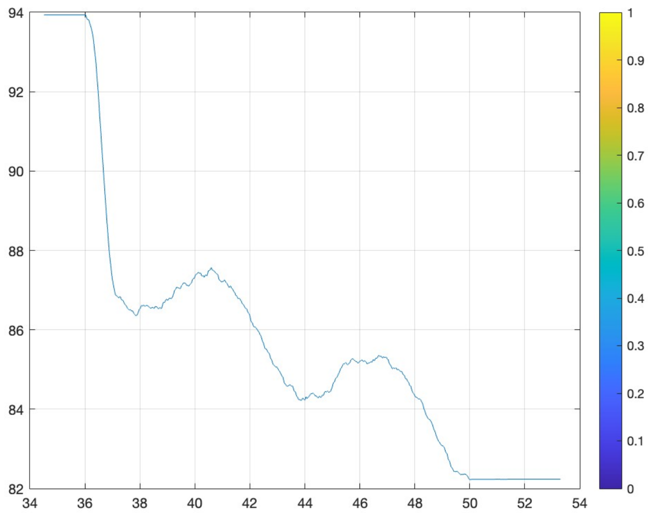

Figure 5.7: (θ_{meas} of Spiral CDPR)

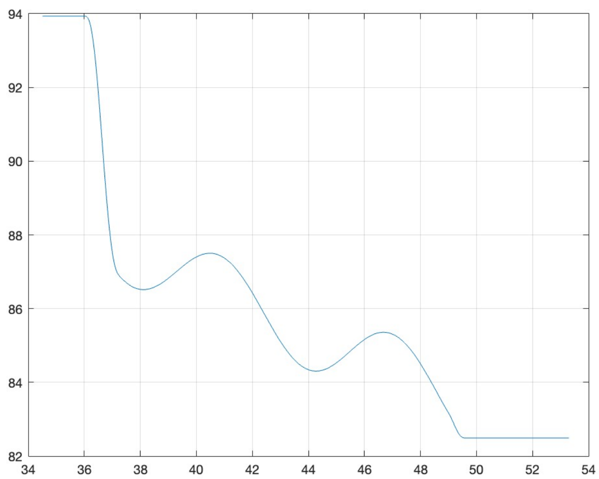

Figure 5.8: (θ_{ref} of Spiral CDPR)

5.5.4 Analysis of $\Delta\theta$ in Spiral Trajectories

Analysing the fluctuations in the angular position represented by $\Delta\theta$—is crucial for understanding the performance of Spiral Cable-Driven Parallel Robots (CDPRs). This analysis aids in evaluating the efficiency of the CDPR's spiral trajectory under various operational conditions.

- **Theoretical $\Delta\theta$ ($\Delta\theta_{\mathbf{ref}}$)**: These values are the expected variations in θ at each point along the trajectory, as calculated by the trajectory planning system.
- **Measured $\Delta\theta$ ($\Delta\theta_{\mathbf{meas}}$)**: These are the real-time angular positions recorded as the robot follows the spiral path.

5.5.5 Comparative Analysis of $\Delta\theta$ Values

By plotting the referenced $\Delta\theta$ figure:5.9 against the measured $\Delta\theta$ figure:5.10 over time, we can assess the fidelity of the CDPR's trajectory following. This comparison not only highlights the occasions where the trajectory deviates from the intended path but also emphasizes the robot's ability to navigate accurately. Analyzing these deviations allows for adjustments in the robot's control algorithms or mechanical design to enhance performance.

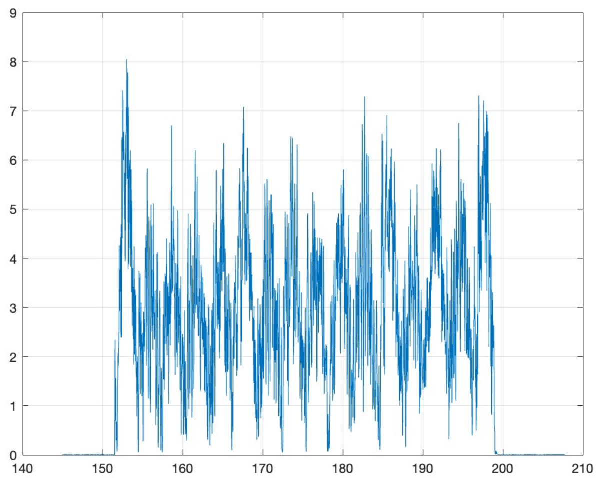

Figure 5.9: Measured $\Delta\theta$ of Spiral CDPR

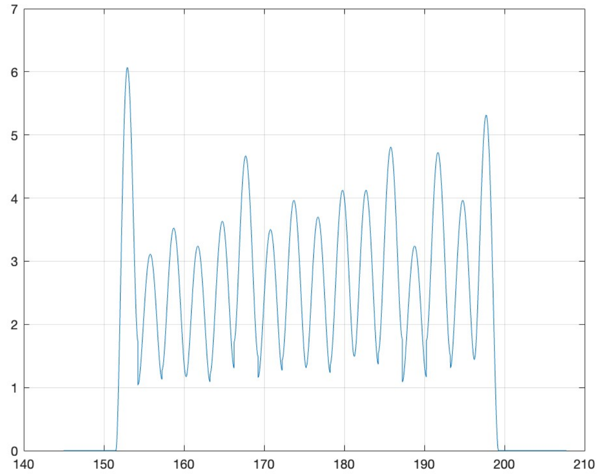

Figure 5.10: Referenced $\Delta\theta$ of Spiral CDPR

5.5.6 Torque Analysis

Torque plays a critical role in the operation of CDPRs,influencing the system's ability to maintain precision and efficiency in following the intended trajectory. We calculate the necessary torque to achieve the targeted θ as follows: figure:5.11 figure:5.12

$$\tau(t) = I \cdot \alpha(t) \tag{5.19}$$

where I denotes the moment of inertia of the end-effector, and $\alpha(t)$ is the angular acceleration, derived from θ through its time derivative.

The following figure compares the theoretical torque required (based on the reference θ) with the actual torque measured during the experiments:

5.6 Conclusion

The proposed trajectory generation algorithm for the 3DOF CDPR using quintic spline curves and PID control has been validated through simulations. The end-effector successfully follows a complex spiral path with high precision and smooth motion, demonstrating the robustness and accuracy of the control strategy.

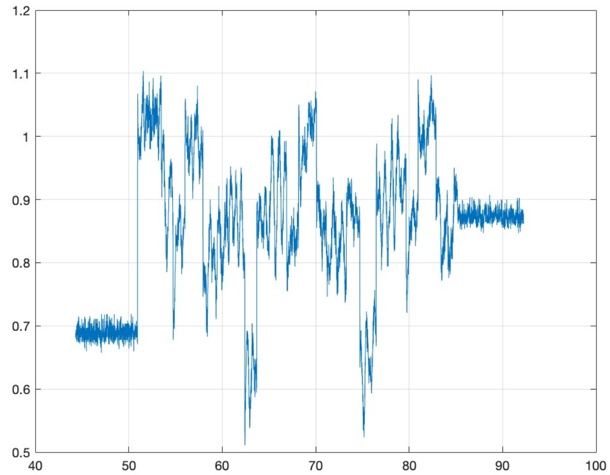

Figure 5.11: Measured Torque of Spiral CDPR

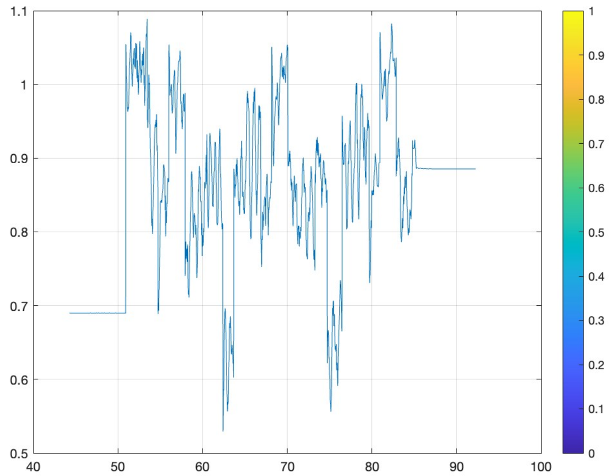

Figure 5.12: Referenced Torque of Spiral CDPR

6

Conclusions and Future Works

In this thesis, we have built and assessed a motion planner for intricate trajectories in cable suspended parallel robots (CSPr). Design, implementation, and analysis of several trajectory generating techniques including cubic and spiral trajectories was the main emphasis of this work to guarantee accurate, smooth, and efficient motion of the end-effector.

6.1 Results and Contributions

This showed how well cubic polynomials trajectories provide CSPRs continuous, smooth motion. Computationally efficient and guarantee that the end-effector moves smoothly between predefined control points, hence preserving continuous first and second derivatives. We investigated the creation of spiral trajectories, which are fundamental for uses involving navigation in three-dimensional space. To enable the useful use of these trajectories in robotic systems, cylindrical to Cartesian coordinates were adopted.

We created a strong control system able to precisely monitor the intended paths. Extensive simulations were used to validate the performance of the control system by making sure the end-effector follows the intended route with least variance. In particular, By means of optimal control techniques, the dynamics of the cable-suspended system was managed, therefore resolving issues including dynamic load handling and cable tension control.

Extensive simulations were carried out to assess the performance of the suggested motion planner with MATLAB and Simulink. With low Root Mean Square Error (RMSE) values implying great accuracy in trajectory tracking, the simulations revealed that the CSPR could precisely follow both cubic and spiral trajectories. Comparative investigation showed that spiral trajectories offer major benefits in tasks demanding complicated 3D navigation, even if cubic trajectories are well-suited for applications needing linear or planar motion. Applications for the created motion planner span industrial automation, robotic surgery, and aerial robotics among other real-world situations. Generating and following difficult paths guarantees that CSPRs can carry out exact and effective activities in various surroundings.

6.2 Future Action

Although this study has advanced motion planners for CSPRs significantly, numerous issues demand more investigation:

Future research could concentrate on building adaptive control algorithms that can dynamically react to changing load conditions and external disturbances, hence improving the resilience of the system. Using the suggested motion planner on a physical CSPR system would offer insightful analysis of its possible areas for development and practical performance. Researching optimisation strategies for trajectory planning might result in more effective pathways, hence lowering system component wear and energy consumption. Practical implementation depends on the trajectory generating and control algorithms operating in real-time being able. Future studies could look at methods to enhance real-time performance and simplify computing methods.

Development of a motion planner for complex trajectories in cable-suspended parallel robots is a significant advancement in the field of robotics. The capability of CSPRs not only increases but also their application range by means of their ability to generate and precisely follow challenging paths. This thesis provides a good platform for next research and development in this subject, therefore enabling more advanced and versatile robotic systems.

At last, since it provides a robust, efficient, and adaptable approach for trajectory generation in CSPRs, the proposed motion planner has great benefits for both academic research and industry purposes.

References

[1] S. Abdolshah and G. Rosati. “First experimental testing of a dynamic minimum tension control (DMTC) for cable driven parallel robots”. In: *Cable-Driven Parallel Robots*. Springer, 2015, pp. 239–248.

[2] Marco Alecci et al. “Development of an IR System for Argument Search.” In: *CLEF (Working Notes)*. 2021, pp. 2302–2318.

[3] G. Barrette and C. M. Gosselin. “Determination of the dynamic workspace of cable-driven planar parallel mechanisms”. In: *Journal of Mechanical Design* 127.2 (2005), pp. 242–248.

[4] P. Boscariol et al. “Spring design for motor torque reduction in articulated mechanisms”. In: *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, Cham, 2017, pp. 557–564.

[5] G. Boschetti, G. Carbone, and C. Passarini. “A fail-safe operation strategy for LAWEX (LARM Wire Driven EXercising Device)”. In: *IFToMM Symposium on Mechanism Design for Robotics*. Springer, Cham, 2018, pp. 424–431.

[6] G. Boschetti, C. Passarini, and A. Trevisani. “A strategy for moving cable driven robots safely in case of cable failure”. In: *Advances in Italian Mechanism Science*. Springer, Cham, 2017, pp. 203–211.

[7] G. Boschetti et al. “A fast algorithm for wrench exertion capability computation”. In: *Cable-Driven Parallel Robots*. Springer, Cham, 2018, pp. 292–303.

[8] P. Bosscher and I. Ebert-Uphoff. “Wrench-based analysis of cable-driven robots”. In: *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*. Vol. 5. IEEE. 2004, pp. 4950–4955.

[9] T. Bruckman et al. “Parallel manipulators, new developments, chap: Wire robot part I, kinematics, analysis and design”. In: 2008, pp. 109–132.

[10] X. Diao and O. Ma. "Workspace determination of general 6-dof cable manipulators". In: *Advanced Robotics* 22.2-3 (2008), pp. 261–278.

[11] A. Fattah and S. K. Agrawal. "On the design of cable-suspended planar parallel robots". In: *Journal of Mechanical Design* 127.5 (2005), pp. 1021–1028.

[12] P. Gallina and G. Rosati. "Manipulability of a planar wire driven haptic device". In: *Mechanism and Machine Theory* 37.2 (2002), pp. 215–228.

[13] M. Gouttefarde and C. Gosselin. "Time-optimal trajectory planning for cable-driven parallel robots". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 3555–3562.

[14] H. Li et al. "Optimal force distribution based on slack rope model in the incompletely constrained cable-driven parallel mechanism of fast telescope". In: *Cable-driven parallel robots*. Springer, 2013, pp. 87–102.

[15] A. Ming and T. Higuchi. "Study on multiple degree-of-freedom positioning mechanism using wires. I: Concept, design and control". In: *International Journal of the Japan Society for Precision Engineering* 28.2 (1994), pp. 131–138.

[16] NIST. *RoboCrane*. `https://www.nist.gov/el/intelligent-systems-division-73500/robo-crane`. Retrieved from. n.d.

[17] C. Passarini, D. Zanotto, and G. Boschetti. "Dynamic trajectory planning for failure recovery in cable-suspended camera systems". In: *International Journal of Mechanisms and Robotics* (2018). Advance online publication. DOI: `10.1115/1.4041942`.

[18] C. B. Pham, G. Yang, and S. H. Yeo. "Dynamic analysis of cable-driven parallel mechanisms". In: *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. IEEE. 2005, pp. 612–617.

[19] Andreas Pott and Horst Kröger. "Trajectory planning for cable-driven parallel robots". In: *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2009, pp. 3561–3566.

[20] A. Risoli et al. "A two degrees-of-freedom planar haptic interface with high kinematic isotropy". In: *Robot and Human Interaction, 1999. RO-MAN'99. 8th IEEE International Workshop on*. IEEE. 1999, pp. 297–302.

[21] B. Siciliano et al. *Robotics: Modelling, Planning and Control*. Springer Science & Business Media, 2009.

[22] K. Sreenath and A. Rodriguez. "Trajectory planning for cable-driven parallel robots: A survey". In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2013, pp. 3285–3292.

[23] J. S. Sulzer, M. Peshkin, and J. L. Patton. "Design of a mobile, inexpensive device for upper extremity rehabilitation at home". In: *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*. IEEE. 2007, pp. 933–937.

[24] A. Trevisani, P. Gallina, and R. L. Williams II. "Cable-direct-driven robot (cddr) with passive scara support: theory and simulation". In: *Journal of Intelligent and Robotic Systems* 46.1 (2006), pp. 73–94.

[25] W. Wang et al. "Design and control of a 500-m aperture spherical radio telescope". In: *Journal of Aerospace Information Systems* 13.4 (2016), pp. 265–275.

[26] X. Yin and A. P. Bowling. "Dynamic performance limitations due to yielding in cable-driven robotic manipulators". In: *Journal of Mechanical Design* 128.1 (2006), pp. 311–318.

[27] Y. Zhang, S. Ge, and Y. Zhang. "Trajectory planning for cable-driven parallel robots based on time-optimal control". In: *Journal of Intelligent & Robotic Systems* 57.1-2 (2010), pp. 159–171.

Acknowledgments

I would want to sincerely thank you to all individuals that helped this thesis on the creation of a motion planner for suspended Cable-Driven Parallel Robots to be successfully completed.

First of all, I would want to especially thank you to my thesis advisor, Mr.Giovanni Boschetti, whose knowledge, sensitivity, and patience greatly enhanced my graduation experience. I really appreciate ready availability of his time so liberally. Over this study and writing process, his direction and constant support have really helped me.

In addition, I would want to thank Mr.Riccardo Minto for his perceptive remarks and support; but, also for the difficult questions that motivated me to extend my research from several angles.

Additionally many thanks goes to the members of my thesis committee,for their time and thorough remarks, which were priceless during the preparation and execution of this research project.

I would want to especially thank my colleagues for giving me the tools I need and for their help in the lab allowing me to finish my research.

Particularly thanks especially to the constantly available and much valued help of the technical team of the Department DEI Unipd.

Finally, I would want to thank my family for all of their love and encouragement. For my parents who taught me the early rising and hard work ethic. For the great moral support and continual encouragement as well as for my partner for understanding and unbounded love.