République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Superieure et de la Recherche Scientifique

---

Université Echahid Hamma Lakhdar d'El-Oued
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE MECANIQUE

Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de

MASTER ACADEMIQUE

Domaine : Sciences et Technologies
Filière : Electromécanique
Spécialité : Electromécanique

Thème

Design Optimization of a Cartesian Parallel Manipulator

Devant le jury composé de :

Dr.Miloudi Khaled	Président
Dr.Gerfi Yousef	Examinateur
Dr.MANSOURI Khaled	Encadreur

Présenté par :

- Tedjani Mohammed Laid
- Khelaifa Ammar
- Kouider Abdallah
- Cherrahi Boubaker

2022-2023

Acknowledgment

We start our modest scientific research with the grace of God, the maximum amount of praise and thanks to him before and after that we have succeeded in reaching this degree of learning and development. Then, we thank our supervisor "Mr. MANSOURI KHALED" for the support and guidance he provided us on the topic of research and outside it, and all the professors who taught and supported us throughout our academic career and who were the reason for what we have reached today.

Dedication

We dedicate this work to whom do we prefer them over ourselves and who were and still a reason to facilitate the paths of life

To our parents

We dedicate it to our brothers and sisters, and we thank them for their encouragement and assistance, as well as to our family members. To all our friends from near and far.

Moreover, all those who stood beside us and helped us with everything they had.

We offer you this research

Table of contents

List of abbreviations

PKM	Parallel kinematic machine
DOF	Degree of freedom
DOE	Design of Experiments
PSO	Particle Swarm Optimization
EA	evolutionary algorithms
CPM	Cartesian parallel manipulators
PRRR	prismatic-revolute-revolute-revolute

Lists of figures

Introduction

The field of robotics has witnessed significant advances in recent years, leading to the development of highly capable and versatile robotic systems. Among these systems, parallel manipulators have emerged as a promising solution for a wide range of industrial applications due to their inherent advantages in terms of speed, accuracy, and payload capacity. One such parallel manipulator is the Cartesian parallel manipulator, known for its unique design and exceptional precision.

The design of a CPM plays a pivotal role in determining its overall performance and efficiency. Optimal design methodologies aim to enhance these key factors by carefully considering various parameters such as kinematics, dynamics, workspace, stiffness, and actuator selection. By employing systematic and advanced optimization techniques, it becomes possible to achieve an optimal design that maximizes the manipulator's capabilities while minimizing potential limitations.

The primary objective of this dissertation is to investigate and propose an optimal design for a Cartesian parallel manipulator. This research aims to address the existing challenges and limitations in the design of such manipulators and contribute to the development of more efficient and effective robotic systems. By exploring various design aspects and employing advanced optimization algorithms, this study seeks to improve the manipulator's performance, accuracy, and overall usability.

To accomplish the research goals, this dissertation will follow a structured approach involving several key steps. First, a comprehensive review of the literature will be conducted to establish a solid foundation and identify the existing design methodologies and their associated strengths and weaknesses. Next, the kinematic and dynamic analyses of the Cartesian parallel manipulator will be performed to gain insights into its behavior and identify potential areas for improvement.

Furthermore, this research will focus on optimizing the manipulator's workspace and its corresponding dimensions to maximize the available working area while ensuring the desired accuracy and precision. Additionally, the selection and optimization of the actuators will be explored to enhance the manipulator's performance and efficiency. Special attention will also be given to the manipulator's stiffness characteristics to improve its overall stability and rigidity.

Throughout this dissertation, advanced optimization algorithms, such as genetic algorithms, particle swarm optimization, or other metaheuristic approaches, will be applied to achieve an optimal design configuration. These algorithms will help explore the design space efficiently and find the best possible solutions, considering multiple design objectives and constraints.

The findings of this research are expected to make a significant contribution to the field of robotics, particularly in the domain of parallel manipulators. The proposed optimal design will serve as a valuable reference for researchers and engineers involved in the design and development of Cartesian parallel manipulators. Moreover, the insights gained from this study will pave the way for the application of similar optimization techniques in the design of other robotic systems, further advancing the field as a whole.

In conclusion, this dissertation aims to present an optimal design for a Cartesian parallel manipulator by addressing various design considerations and employing advanced optimization techniques. By enhancing the manipulator's performance, accuracy, and efficiency, this research endeavors to contribute to the advancement of robotics and pave the way for the development of more capable and versatile robotic systems.

Chapter I: Literature Review

I.1 Introduction

The literature review chapter presents a comprehensive overview of the existing research and knowledge related to the design and optimization of Cartesian parallel manipulators. It traces the historical development of parallel manipulators, classifies different types of parallel manipulators, and highlights the advantages of CPM [1]. The review also explores the kinematic analysis techniques employed in the study of these manipulators, along with optimization methodologies for performance enhancement. Additionally, real-world applications are examined to showcase the practical utility of Cartesian parallel manipulators. The chapter concludes by identifying research gaps and setting the stage for subsequent chapters, which focus on exploring optimal design parameters for improved manipulator performance[2].

I.2 Definition

Parallel robots, also known as parallel manipulators or (PKM), are a closed-loop mechanism that controls the motion of their end-effectors by connecting them to the base via at least two separate multiple linkages (kinematic chains). This loop is closed by an n DOF end-effector linked to the base by n separate chains with no more than two links, each operated by a single prismatic or rotary actuator [3].

I.3 Background and history of parallel manipulators

Parallel manipulators have a rich history and have emerged as a significant area of research in robotics and automation [4]. These manipulators differ from traditional serial manipulators, as they consist of a rigid platform connected to the base through multiple kinematic chains. This configuration offers unique advantages such as increased precision, stiffness, and higher payload capacities.

The concept of parallel manipulators can be traced back to the late 19th century when early designs were proposed by Stewart, Chebyshev, and other prominent engineers. However, it was not until the 1980s and 1990s that parallel manipulators gained widespread attention due to advancements in computation and control

Over the years, parallel manipulators have been extensively studied and classified based on their architecture and kinematic properties. Some of the well-known parallel manipulator configurations include the Delta robot, the SCARA robot, and the Cartesian parallel manipulator. Each configuration offers unique characteristics and is suited for specific applications[5].

The Delta robot, developed in the 1980s, is known for its high-speed and precision capabilities[6], making it suitable for applications such as pick-and-place operations in industries like food packaging and electronics assembly. The SCARA (Selective Compliance Assembly Robot Arm) robot is widely used in assembly tasks that require high repeatability and dexterity.

Among these configurations, the Cartesian parallel manipulator has gained significant prominence. It consists of three or more linear actuators connected in a parallel configuration, forming a Cartesian coordinate system. This design allows for simple and intuitive control and precise positioning of the end-effector. Cartesian parallel manipulators are used in various industrial applications, including CNC machining[7], material handling, and medical robotics

The field of parallel manipulators has witnessed significant advancements in recent years, driven by the increasing demand for high-precision and high-performance robotic systems. Researchers have focused on addressing challenges related to kinematics, dynamics, control, and optimization to further enhance the capabilities of parallel manipulators[8].

In conclusion, the background and history of parallel manipulators trace their origins to the late 19th century, with significant advancements and classifications occurring in the latter half of the 20th century[9]. Parallel manipulators, including the Cartesian parallel manipulator, have revolutionized the field of robotics and automation, offering superior performance characteristics and opening up new possibilities in various industrial applications.

I.4 Classification of parallel manipulator

Parallel manipulators can be classified based on various criteria, including their kinematic architecture, mobility, and actuation mechanisms. This classification provides a systematic framework for understanding the different types of parallel manipulators and their unique characteristics. Here are some common ways parallel manipulators are classified:

I.4.1 Based on Kinematic Architecture:

a. Planar Parallel Manipulators: These manipulators operate in a two-dimensional plane and typically consist of revolute or prismatic joints. They are commonly used for applications such as drawing and painting, 3D printing, and milling.

b. Spatial Parallel Manipulators: Spatial manipulators operate in three-dimensional space and involve more complex kinematic structures. They often have multiple degrees of freedom and are used in applications requiring higher dexterity and maneuverability, such as flight simulators, surgical robotics, and advanced assembly tasks.

I.4.2 Based on Mobility

a. Fully Parallel Manipulators: These manipulators have all the links connected directly to the base and the end-effector, resulting in a fixed number of degrees of freedom (DOFs). Fully parallel manipulators offer high stiffness, accuracy, and load-bearing capabilities. Examples include the Delta robot and Stewart platform.

b. b. Partially Parallel Manipulators: In partially parallel manipulators, some of the links are connected to both the base and the end-effector, while others are connected to either the base or the end-effector. This configuration allows for a variable number of DOFs and offers a balance between flexibility and stiffness. Examples include the Gough-Stewart platform and the 3-PRS parallel manipulator.

I.4.3 Based on Actuation Mechanism

a. Actuated Parallel Manipulators: In these manipulators, the joints or links are driven by actuators (such as motors) to control their motion. Actuated parallel manipulators provide precise control over the end-effector position and orientation and are commonly used in industrial automation and robotics[10].

b. Passive Parallel Manipulators: Passive parallel manipulators do not have active actuators and rely on external forces or gravity for motion. These manipulators often utilize compliant mechanisms or passive elements to achieve desired motion or force transmission[11].

I.4.4 Based on Specialized Configurations

Some parallel manipulators have specialized configurations tailored for specific applications, such as:

a. Delta Robot: A type of parallel manipulator with a unique triangular configuration, widely used in high-speed pick-and-place operations.

b. SCARA Robot: A planar parallel manipulator with a rigid arm and revolute joints, commonly used in assembly tasks requiring high repeatability[12].

These classifications provide a framework to understand the various types of parallel manipulators and their capabilities. Each configuration has its own advantages and limitations, making them suitable for specific applications across industries ranging from manufacturing and automation to healthcare and aerospace.

I.5 Kinematic analysis of parallel manipulators

The kinematic analysis of parallel manipulators involves studying the relationships between joint displacements, end-effector position, and orientation. It aims to understand the geometric constraints and motion characteristics of the manipulator's links and joints. Kinematic analysis plays a crucial role in determining the workspace, singularity conditions, and the ability of the manipulator to achieve desired end-effector poses[13].

The kinematic analysis of parallel manipulators typically includes two main aspects: forward kinematics and inverse kinematics.

I.5.1 Forward Kinematics

Forward kinematics refers to the determination of the end-effector position and orientation in the workspace based on known joint displacements. The goal is to establish a mathematical relationship between the joint variables and the resulting pose of the end-effector[14]. This analysis involves applying appropriate transformation matrices and coordinate transformations to track the end-effector's position and orientation relative to a fixed coordinate system. Various methods, such as the Denavit-Hartenberg convention or the product of exponentials, can be employed to derive the forward kinematic equations.

I.5.2 Inverse Kinematics

Inverse kinematics involves determining the required joint displacements to achieve a desired end-effector pose in the workspace. It focuses on finding the joint variable values that satisfy the specified position and orientation constraints. Inverse kinematics is particularly important for trajectory planning, motion control, and path generation. Solving the inverse kinematics problem can be challenging due to the nonlinear nature of the equations and the existence of multiple solutions or singularities[15]. Various numerical methods, such as iterative approaches or optimization techniques, can be used to solve the inverse kinematic equations.

During the kinematic analysis of parallel manipulators, several important aspects are considered, including the determination of the workspace, the identification of singular configurations, and the analysis of the Jacobian matrices.

I.5.3 Workspace Analysis

The workspace of a parallel manipulator refers to the region in which the end-effector can reach while respecting the constraints of the manipulator's structure and joint limits. It is essential to evaluate the workspace to determine the manipulator's capabilities, including its reach, dexterity, and potential limitations.

I.5.4 Singular Configurations

Singular configurations are specific joint configurations where the manipulator loses one or more degrees of freedom. These configurations can lead to motion restrictions, reduced accuracy, or increased sensitivity to disturbances. Identifying and analyzing singularities is crucial for avoiding them during operation and optimizing the manipulator's performance.

I.5.5 Jacobian Matrices

The Jacobian matrices provide a relationship between the joint velocities and the resulting end-effector velocities. They play a fundamental role in the analysis of manipulator dynamics, control, and trajectory planning[16]. Understanding the Jacobian matrices helps in assessing the manipulator's sensitivity to joint motion and optimizing its performance characteristics.

Overall, the kinematic analysis of parallel manipulators enables the understanding of the manipulator's motion capabilities, constraints, and the relationship between joint displacements and end-effector pose. It serves as a foundation for subsequent analyses, such as dynamics modeling, control strategies, and optimization techniques, leading to improved performance and efficiency of parallel manipulators in various applications.

I.6 Design optimization methods for parallel manipulators

Design optimization plays a crucial role in enhancing the performance, efficiency, and overall capabilities of parallel manipulators[17]. By employing optimization methods, researchers and engineers can systematically explore the design space to find optimal solutions that meet specific objectives and constraints. Here are some commonly used design optimization methods for parallel manipulators:

I.6.1 Mathematical Optimization Techniques

Mathematical optimization methods involve formulating the design problem as a mathematical optimization program and finding the optimal solution through mathematical analysis. These techniques include:

a. Genetic Algorithms (GA): GA is an evolutionary algorithm that mimics natural selection and genetic processes to search for optimal solutions. It utilizes population-based search and employs genetic operators such as selection, crossover, and mutation to evolve the solution towards optimality.

b. Particle Swarm Optimization (PSO): PSO is a swarm intelligence-based optimization technique inspired by the collective behavior of bird flocking or fish schooling. It employs a population of particles that explore the design space and communicate with each other to search for the global optimum.

c. Gradient-Based Methods: Gradient-based optimization methods use gradient information to iteratively update the design variables and converge towards the optimal solution. These methods include gradient descent, Newton's method, and quasi-Newton methods such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

I.6.2 Multi-Objective Optimization

Multi-objective optimization methods aim to optimize multiple conflicting objectives simultaneously. In the case of parallel manipulators, these objectives may include workspace maximization, dexterity enhancement, minimizing actuator forces, or energy consumption. Multi-objective optimization techniques, such as the weighted sum method, Pareto-based approaches, or evolutionary algorithms like the Non-Dominated Sorting Genetic Algorithm (NSGA), help in finding the trade-offs among different objectives and obtaining a set of optimal solutions known as the Pareto front[18].

I.6.3 Sensitivity Analysis

Sensitivity analysis assesses the impact of design parameters on the performance metrics of parallel manipulators. By evaluating the sensitivity of the system to changes in various design variables, engineers can identify critical parameters and optimize them accordingly. Sensitivity analysis techniques, such as finite difference methods or adjoint sensitivity analysis, help in understanding the influence of design variations and guide the optimization process.

I.6.4 Surrogate Modeling and Response Surface Methodology

Surrogate modeling techniques involve constructing approximate mathematical models, known as surrogate models, to represent the complex relationship between design variables and performance metrics. These surrogate models are computationally efficient and can be used for optimization purposes. Response Surface Methodology (RSM) is a common approach in surrogate modeling, which uses polynomial or Gaussian process regression to create a response surface and perform optimization on the surrogate model.

I.6.5 Design of Experiments (DOE)

DOE methods involve systematically planning and conducting experiments to collect data and analyze the influence of design variables on the performance of parallel manipulators. By carefully selecting the experimental settings, researchers can efficiently explore the design space and obtain insights into the relationships between design parameters and performance measures. Techniques such as factorial design, Taguchi methods, or Latin hypercube sampling are commonly used in DOE for parallel manipulator optimization.

These design optimization methods provide systematic approaches to improve the performance, efficiency, and robustness of parallel manipulators. By employing these techniques, researchers and engineers can explore the design space, identify optimal solutions, and enhance the capabilities of parallel manipulators in terms of workspace, precision, dexterity, energy efficiency, and other performance metrics.

I.7 Control systems for parallel manipulators

Control systems play a crucial role in the operation and performance of parallel manipulators. They enable precise and accurate control of the manipulator's motions, ensuring efficient and reliable operation. Here are some commonly used control systems for parallel manipulators:

I.7.1 Proportional-Integral-Derivative (PID) Control

PID control is a widely used control strategy that utilizes feedback loops to adjust the manipulator's joint positions or velocities based on the error between the desired and actual values. The control system applies proportional, integral, and derivative actions to calculate the control signals and provide stable and accurate tracking of the desired trajectories. PID control is relatively simple to implement and provides good performance for many parallel manipulator applications.

I.7.2 Adaptive Control

Adaptive control systems adjust the control parameters based on the manipulator's dynamic behavior and external disturbances. These control systems continuously adapt to changes in the manipulator's characteristics, such as varying loads, changes in inertia, or parameter uncertainties. Adaptive control algorithms, such as Model Reference Adaptive Control (MRAC) or Adaptive Robust Control (ARC), continuously estimate the system parameters and adjust the control signals to maintain performance and stability.

I.7.3 Model Predictive Control (MPC)

MPC is a control strategy that utilizes a predictive model of the system dynamics to optimize the control inputs over a future time horizon. By considering the future behavior of the manipulator and incorporating constraints on inputs and outputs, MPC can provide accurate tracking of desired trajectories and handle constraints such as joint limits or actuator saturation. MPC is particularly effective for parallel manipulators that require fast and precise control responses.

I.7.4 Force/Torque Control

Force/Torque control systems focus on regulating the interaction forces and torques between the manipulator and its environment. These control systems allow precise control of contact forces, enabling tasks such as compliant assembly, force-guided manipulation, or haptic feedback. Force/Torque control algorithms, such as impedance control or hybrid position/force control, modulate the control inputs to regulate the interaction forces based on sensor feedback.

I.7.5 Decentralized Control

Decentralized control systems distribute the control tasks among the individual actuators or limbs of the parallel manipulator. Each limb operates independently, controlling its own set of joints and sensors, while coordinating with other limbs through communication protocols. Decentralized control allows for fault tolerance, flexibility, and scalability, making it suitable for complex parallel manipulator architectures.

I.7.6 Trajectory Planning and Motion Control

Trajectory planning and motion control algorithms generate smooth and optimized trajectories for the parallel manipulator's end-effector motion. These algorithms consider kinematic and dynamic constraints, such as joint limits, workspace limitations, and acceleration/deceleration limits. They aim to minimize tracking errors, reduce vibration, and optimize the manipulator's performance during motion. Techniques such as spline interpolation, polynomial trajectory generation, or minimum-time control can be used for trajectory planning and motion control.

These control systems are tailored to the specific requirements of parallel manipulators and contribute to their precise, stable, and efficient operation. The choice of control system depends on factors such as the manipulator's architecture, the desired performance objectives, the level of complexity, and the application requirements. By employing suitable control systems, parallel manipulators can achieve accurate positioning, high-speed operation, load manipulation, and perform complex tasks in various fields such as manufacturing, robotics, aerospace, and healthcare.

I.8 Existing research on Cartesian parallel manipulators

Cartesian parallel manipulators, also known as Gantry robots, have been the subject of extensive research and development due to their unique kinematic characteristics and potential applications in various industries[19]. Here is an overview of some existing research on Cartesian parallel manipulators:

I.8.1 Kinematic Analysis and Workspace Optimization

Many studies have focused on the kinematic analysis of Cartesian parallel manipulators, including forward and inverse kinematics, singularity analysis, and workspace characterization. Researchers have investigated the geometric constraints, motion capabilities, and workspace optimization of different configurations of Cartesian parallel manipulators. These studies provide insights into the manipulator's kinematic properties, allowing for better design and control strategies.

I.8.2 Structural Design and Stiffness Analysis

Research has been conducted on the structural design and stiffness analysis of Cartesian parallel manipulators. This includes investigations into the selection of materials, linkages, and joints to enhance the manipulator's stiffness and rigidity. Analytical and numerical methods have been employed to evaluate the stiffness and deformation characteristics of the manipulator under different loading conditions, ensuring optimal design for improved performance.

I.8.3 Dynamic Modeling and Control

Dynamic modeling and control of CPM have been extensively studied. Researchers have developed dynamic models considering the manipulator's rigid body dynamics, joint flexibility, and actuator dynamics. Control strategies, such as PD control, adaptive control, and model-based control, have been investigated for accurate tracking of desired trajectories, disturbance rejection, and vibration suppression. These studies aim to improve the manipulator's dynamic performance and stability.

I.8.4 Singularity Analysis and Avoidance

Singularity analysis is a significant area of research in CPM Various methods have been proposed to identify singular configurations and analyze their impact on the manipulator's performance. Researchers have developed techniques to avoid or alleviate singularities during operation, ensuring smooth and continuous motion of the manipulator.

I.8.5 Fault Diagnosis and Tolerance

Research has been conducted on fault diagnosis and tolerance for CPM to enhance their reliability and robustness. Studies have explored the detection and identification of sensor failures, actuator faults, and structural damages. Fault-tolerant control strategies, such as redundancy-based control and reconfiguration techniques, have been investigated to maintain the manipulator's functionality and performance in the presence of faults.

I.8.6 Optimization and Performance Enhancement

Many researchers have focused on optimizing the performance of Cartesian parallel manipulators. This includes optimization of workspace, dexterity, accuracy, and energy efficiency. Optimization methods such as genetic algorithms, particle swarm optimization, and multi-objective optimization techniques have been employed to find optimal design parameters, control strategies, and actuator configurations.

I.8.7 Applications and Case Studies

Existing research on Cartesian parallel manipulators has explored a wide range of applications across industries. Studies have demonstrated the applicability of CPM in areas such as pick-and-place operations, manufacturing processes, automated assembly, material handling, medical robotics, and advanced motion control systems. Case studies and experimental validations have been conducted to showcase the capabilities and performance of CPM in real-world scenarios.

These research efforts contribute to the advancement of knowledge and understanding of Cartesian parallel manipulators. They provide valuable insights into the design, control, and optimization of these manipulators, leading to improved performance, expanded applications, and wider adoption in various industries.

I.9 Conclusion

the optimal design of a Cartesian parallel manipulator involves systematically exploring the design space to find the best configuration and parameters that meet specific performance criteria. By defining clear objectives, constraints, and requirements, engineers can formulate an objective function and identify design variables to be optimized. Through the use of appropriate optimization algorithms, the design space is explored, and different combinations of design variables are evaluated to find the optimal solution. The results are analyzed and evaluated, considering trade-offs and compromises, leading to informed decisions for design refinement. The optimal design of a Cartesian parallel manipulator enables improved performance, accuracy, efficiency, and productivity in various industries and applications. It plays a crucial role in pushing the boundaries of what is achievable and helps to achieve innovative, efficient, and sustainable design solutions in the field of robotics and automation.

Chapter II: Kinematic Analysis

II.1 Introduction

Design optimization is the process of finding the best design solution for a given problem by systematically exploring the design space and optimizing specific criteria or objectives[20]. It involves selecting the optimal values for design variables while considering various constraints and objectives.

The goal of design optimization is to improve the performance, efficiency, or cost-effectiveness of a product, system, or process. By optimizing the design, engineers and designers can achieve desired objectives such as maximizing performance metrics (e.g., strength, speed, or energy efficiency), minimizing costs (e.g., material usage or manufacturing expenses), or satisfying specific constraints (e.g., size limitations or regulatory requirements).

Design optimization typically involves defining the problem, identifying design variables and constraints, formulating an objective function to be optimized, and selecting an appropriate optimization algorithm. The optimization algorithm explores the design space by evaluating different combinations of design variables and iteratively refining the design solution until an optimal or near-optimal solution is found[21].

The benefits of design optimization include improved product performance, reduced costs, enhanced efficiency, and faster time-to-market. It allows engineers to explore a wide range of design possibilities, identify trade-offs, and make informed decisions based on quantitative analyses.

II.2 Geometry of the Manipulator

The kinematic structure of the Cartesian parallel manipulator is shown in Fig. 1 where a moving platform is connected to a fixed base by three PRRR (prismatic-revolute-revolute-revolute) limbs[22]. The moving platform is symbolically represented by a circle defined by B_1,B_2, and B_3 and the fixed base is defined by three guide rods passing through A_1,A_2, and A_3, respectively. The three revolute joint axes in each limb are located at points A_i, M_i, and B_i, respectively, and are parallel to the groundconnected prismatic joint axis. Furthermore, the three prismatic joint axes, passing through point A_i for i=1,2, and 3, are parallel to the X,Y, and Z axes, respectively. Specifically, the first prismatic joint axis lies on the X axis; the second prismatic joint axis is parallel to the Y axis with an offset e_z in the Z direction; and the third prismatic joint axis is parallel to the Z axis with an offset e_x in the X direction and e_y in the Y direction. Point P represents the center of the moving platform. The link lengths are denoted by

l_{i1} , l_{i2}, and l_3, respectively. The starting point of a prismatic joint is defined by d_{i0} and the sliding distance is defined by d_i. Note that each PRRR limb is equivalent to a CRR limb. In this regard, the 3-PRRR parallel manipulator is a kinematic inversion of the 3-RRC manipulator. Also note that the third prismatic joint axis is purposely located far away from the Z-axis to avoid mechanical interference among the three limbs.

For simplicity, three local coordinate systems are introduced with their origins located at A_i. The Z_1 , Z_2, and Z_3 axes are parallel to the X, Y, and Z axes, and the X_1, X_2, and X_3 axes are parallel to the Y, Z, and X axes, respectively, as shown in Fig. 1.

Due to the three parallel revolute joints located at points A_i, M_i, and B_i, any single limb constrains the moving platform from rotating about the local X_i and Y_i axes. Since each limb provides two rotational constraints to the moving platform, the combined effects result in three redundant constraints on the rotation of the moving platform and, therefore, completely constrain the moving platform from rotation. This leaves the moving platform with three translational degrees of freedom[23].

Figure 2 shows the joint angles of a limb with respect to the local coordinate system. We note that the joint angles of links $\overline{A_iM_i}$, $\overline{M_iB_i}$, and $\overline{B_iP}$ with respect to the local X axis are denoted by θ_{i1}, θ_{i2}, and θ_{i3}, respectively. The angle between $\overline{A_iB_i}$ and the X_i axis is denoted by θ_{ia}, the angle between lines $\overline{A_iM_i}$ and $\overline{A_iB_i}$ is denoted by θ_{ib}, and the angle between the extended line of $\overline{A_iM_i}$ and $\overline{M_iB_i}$ is denoted by θ^*_{i2}. Further, $\mathbf{r}_{ij}$ denotes a common perpendicular vector between the two revolute joint axes of a link, where the first subscript denotes the limb number, the second subscript represents the link number of a limb, and the leading superscript is used to denote the coordinate system with respect to which a vector is expressed.

Figure 3 shows two methods of actuation, namely, linear and rotary actuation methods. In the following, we discuss the kinematics of each actuation method in turn.

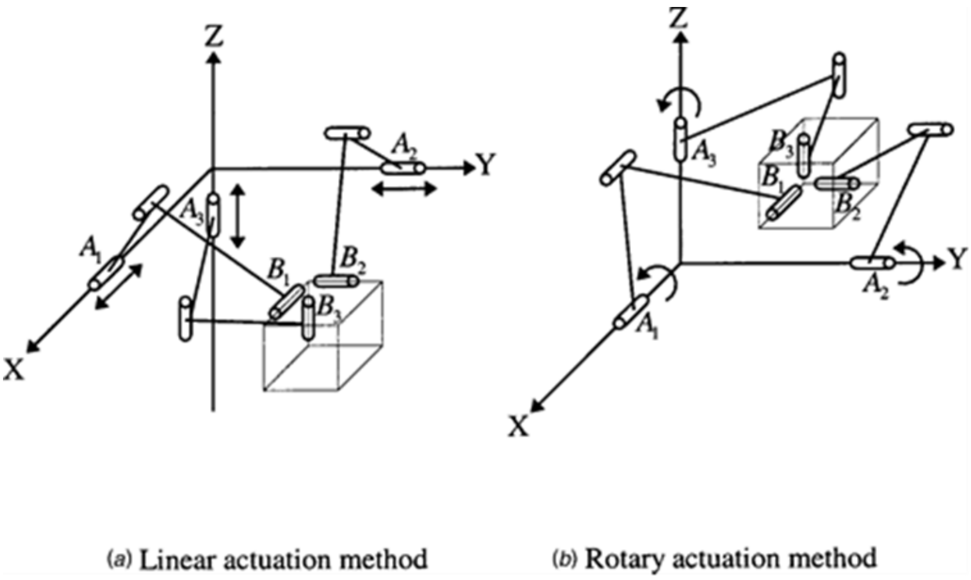


Figure II.1 Linear and rotary actuation methods

II.3 Rotary Actuation Method

For the rotary actuation method in a Cartesian parallel manipulator, a rotary actuator is responsible for driving the first revolute joint of each limb. In this configuration, the remaining joints in the manipulator are passive, meaning they do not have their own actuation and rely on the motion transmitted from the first joint[24].

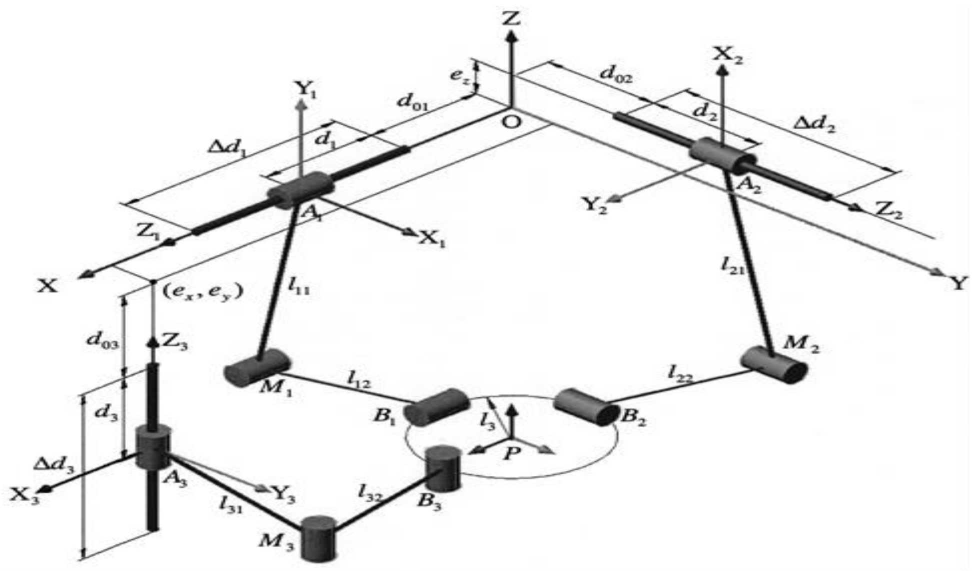


Figure II.2 Spatial 3-PRRR parallel manipulator

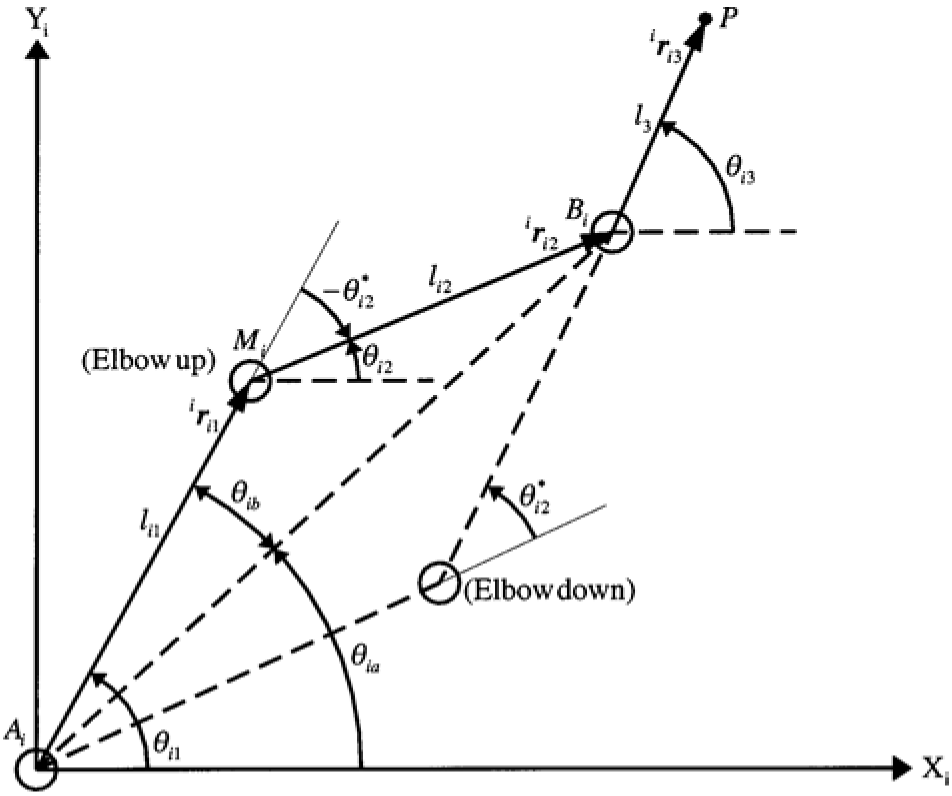


Figure II.3 Joint angles of the ith limb with respect to the ith limb coordinate system

II.3.1 Inverse Kinematic

The inverse kinematics is to find the input joint angles, θ_{i1}, for $i = 1{,}2$, and 3 , for a given endeffector position, $\mathbf{p} = \left[p_x, p_y, p_z\right]^T$. From the geometry given in Fig. 1, we obtain the position vectors of B_i with respect to each local coordinate system as follows:

$$^1\mathbf{B}_1 = \begin{bmatrix} p_y - l_3 \\ p_z \\ 0 \end{bmatrix}, \quad ^2\mathbf{B}_2 = \begin{bmatrix} p_z - e_z \\ p_x - l_3 \\ 0 \end{bmatrix}, \text{ and}$$

$$^3\mathbf{B}_3 = \begin{bmatrix} p_x + l_3 - e_x \\ p_y - e_y \\ 0 \end{bmatrix} \qquad \text{(II-1)}$$

Referring to Fig. 2 , the input joint angles, θ_{i1}, are found as follows:

$$\theta_{i1} = \theta_{ia} \pm \theta_{ift} = \tan^{-1}\frac{^iB_{iy}}{^iB_{ix}} \pm \cos^{-1}\frac{l_{i1}^2 + B_{ix}^2 + {}^iB_{iy}^2 - l_{i2}^2}{2l_{i1}\sqrt{B_{ix}^2 + {}^iB_{iy}^2}} \qquad \text{(II-2)}$$

for $i = 1{,}2$, and 3.

Due to the finite reach of each limb in the Cartesian parallel manipulator, a constraint is imposed to ensure that the minimum and maximum reaches of a serial chain are satisfied. This constraint ensures that the end effector remains within the valid workspace of the manipulator and prevents any unreachable positions.:

$$(l_{i1} - l_{i2})^2 \leqslant {}^{i}B_{ix}^2 + {}_{ix}^{i}B_{iy}^2 \approx (l_{i1} + l_{i2})^2. \tag{II-3}$$

II.3.2 Forward Kinematics

The forward kinematics is to obtain the end-effector position, $\left[p_x, p_y, p_z\right]^T$, when the input joint angles, θ_{i1} for $i = 1,2$, and 3 , are given.

Referring to Fig. 2, the second link length can be written in terms of the locations of the second and third revolute joints as follows:

$$\left({}^{i}B_{ix} - l_{i1}C\theta_{i1}\right)^2 + \left({}^{i}B_{iy} - l_{i1}S\theta_{i1}\right)^2 = l_{i2}^2, \tag{II-4}$$

where $C\theta_{i1} = \cos\theta_{i1}$ and $S\theta_{i1} = \sin\theta_{i1}$. Substituting Eq. (1) into (4) yields

$$\begin{gathered} \left[p_y - (l_{11}C\theta_{11} + l_3)\right]^2 + [p_z - l_{11}S\theta_{11}]^2 = l_{12}^2, \\ [p_z - (l_{21}C\theta_{21} + e_z)]^2 + [p_x - (l_{21}S\theta_{21} + l_3)]^2 = l_{22}^2, \\ [p_x - (l_{31}C\theta_{31} + e_x - l_3)]^2 + \left[p_y - \left(l_{31}S\theta_{31} + e_y\right)\right]^2 = l_{32}^2 \end{gathered} \tag{II-5}$$

Eliminating p_y and p_z from Eq. (5) gives an eighth-degree polynomial in p_x :

$$c_8p_x^8 + c_7p_x^7 + c_6p_x^6 + c_5p_x^5 + c_4p_x^4 + c_3p_x^3 + c_2p_x^2 + c_1p_x + c_0 = 0. \tag{II-6}$$

Hence, at most eight different manipulator configurations are possible for the forward kinematics.

II.3.3 Jacobian, Singularity, and Static Force Analyses

Differentiating Eq. (1) with respect to time yields the velocities of B_i, for $i = 1,2$, and 3 , relative to and expressed in their respective local coordinate systems as follows:

$${}^{1}\dot{\mathbf{B}}_1 = \begin{bmatrix}\dot{p}_y\\ \dot{p}_z\\ 0\end{bmatrix}, \quad {}^{2}\dot{\mathbf{B}}_2 = \begin{bmatrix}\dot{p}_z\\ \dot{p}_x\\ 0\end{bmatrix}, \quad \text{and} \quad {}^{3}\dot{\mathbf{B}}_3 = \begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ 0\end{bmatrix}. \tag{II-7}$$

Referring to Fig. 2 , the linear velocity of B_i relative to the local coordinate system can be written in terms of the joint rates as

$$\dot{B}_{ix} = -\dot{\theta}_{i1} l_{i1} S\theta_{i1} - \dot{\theta}_{i2} l_{i2} S\theta_{i2},$$
$$\dot{B}_{iy} = \dot{\theta}_{i1} l_{i1} C\theta_{i1} + \dot{\theta}_{i2} l_{i2} C\theta_{i2}. \qquad \text{(II-8)}$$

Eliminating $\dot{\theta}_{i2}$ in Eq. (8) yields

$$^{i}\dot{B}_{ix} C\theta_{i2} + {}^{i}\dot{B}_{i1} S\theta_{i2} = l_{i1} S(\theta_{i2} - \theta_{i1})\dot{\theta}_{i1}. \qquad \text{(II-9)}$$

Substituting Eq- (7) into (9), we obtain an input-output velocity relationship as follows:

$$J_x \dot{\mathbf{p}} = J_q \dot{\theta}_1 \qquad \text{(II-10)}$$

where

$$\dot{\mathrm{p}} = [\dot{p}_x, \dot{p}_y, \dot{p}_z]^T,$$
$$\dot{\theta}_1 = [\dot{\theta}_{11}, \dot{\theta}_{21}, \dot{\theta}_{31}]^T,$$
$$J_x = \begin{bmatrix} 0 & C\theta_{12} & S\theta_{12} \\ S\theta_{22} & 0 & C\theta_{22} \\ C\theta_{32} & S\theta_{32} & 0 \end{bmatrix},$$

$$J_q = \begin{bmatrix} l_{11} S(\theta_{12} - \theta_{11}) & 0 & 0 \\ 0 & l_{21} S(\theta_{22} - \theta_{21}) & 0 \\ 0 & 0 & l_{31} S(\theta_{32} - \theta_{31}) \end{bmatrix}$$

Hence, if $S(\theta_{i2} - \theta_{i1}) \neq 0$, the inverse velocity transformation can be written as

$$\theta_1 = J\dot{\mathbf{p}} \qquad \text{(II-11)}$$

Where

$$J = J_q^{-1} J_s \qquad \text{(II-12)}$$

is called the Jacobian matrix of the manipulator.

Next, singular configurations of the manipulator are examined. If $\det(J_g) = 0$, the joint rates cannot be determined. Hence, an inverse kinematic singularity [32] occurs when the first and second links of any limb are collinear; that is, when

$$S(\theta_{i2} - \theta_{i1}) = 0. \tag{II-13}$$

A direct kinematic singularity occurs when $\det(J_s) = 0$ or equivalently

$$C\theta_{12}C\theta_{22}C\theta_{32} + S\theta_{12}S\theta_{22}S\theta_{32} = 0. \tag{II-14}$$

Since Eq. (14) is satisfied for any combination of $C\theta_{i2} = 0$ and $S\theta_{j2} = 0$ for $i \neq j$, we conclude that there exist many direct kinematic singularities. For example, when the manipulator assumes a configuration of $\theta_{12} = 0°, \theta_{22} = 90°$, and $\theta_{32} =$ any, it becomes singular. Sinee such singularity may oecur within the workspace, the rotary actuation method is judged to be impractical.

For static force analysis in a Cartesian parallel manipulator, certain assumptions are made to simplify the analysis. Two common assumptions are that link weights and joint friction are negligible. By neglecting these factors, the static force analysis focuses solely on the external forces and moments acting on the manipulator, such as applied loads or external disturbances:

$$\mathbf{f} = J^T\boldsymbol{\tau} \tag{II-15}$$

where $\mathbf{f} = [f_x, f_y, f_z]^T$ denotes a vector of end-effector output forces and $\tau = [\tau_{11}, \tau_{21}, \tau_{31}]^T$ represents a vector of actuator input torques.

II.4 Linear Actuation Method

For the linear actuation method in a Cartesian parallel manipulator, a linear actuator is used to drive the prismatic joint in each limb, while all the other joints remain passive. This configuration allows for the actuation of the translational motion of the manipulator, while the remaining joints rely on the transmitted motion from the prismatic joint.

II.4.1 Forward and Inverse Kinematics

The forward and inverse kinematic analyses are trivial since there exists a one-to-one correspondence between the end-effector position and the input joint displacements. Referring to Fig. 1, each limb constrains point P to lie on a plane which passes through point A_i and is perpendicular to the axis of the linear actuator. Consequently, the location of P is determined by the intersection of three planes. A simple kinematic relation can be written as

$$\begin{bmatrix} p_s \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} d_{01} + d_1 \\ d_{00} + d_2 \\ d_{03} + d_3 \end{bmatrix} \tag{II-16}$$

II.4.2 Jacobian and Static Force Analyses

The velocity and output force of the end-effector also have a one-to-one relationship with that of the linear actuators. Taking the derivative of Eq. (16) with respect to time yields

$$\begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} = J \begin{bmatrix} p_x \\ p_z \\ p_z \end{bmatrix} \tag{II-17}$$

where J is the 3×3 identity matrix. Since J is an identity matrix, the manipulator is isotropic everywhere within its workspace.

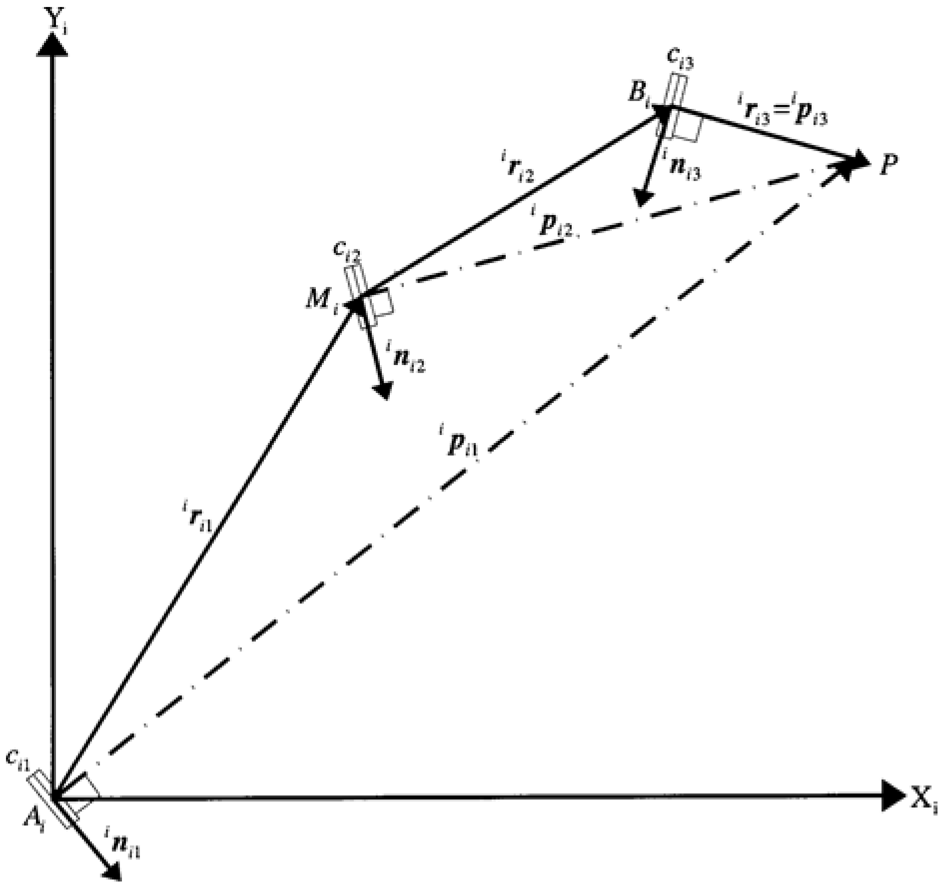


Figure II.4 A serial manipulator made up of three compliant joints

Neglecting the gravitational forces of the links and frictional forces in the joints, and applying the principle of virtual work, we can obtain a static force relation for the Cartesian parallel manipulator

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = J^7 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{II-18}$$

where $[f_x, f_y, f_z]^T$ denotes a vector of end-effector output forces and $[f_1, f_2, f_3]^T$ denotes a vector of linear actuator forces. We observe that due to the orthogonal arrangement of the linear actuators, there exists a one-to-one correspondence between the veetor of actuator input forces and that of the end-effector output forces.

Because of the de-coupled X, Y, and Z motions and the isotropic characteristics of the Jacobian matrix, the linear actuation method is chosen for further analysis and prototype development.

II.4.3 Stiffness Modeling

Figure 4 shows the ith limb sketched in its local coordinate system where the revolute joint axes at points A_i, M_i, and B_i (not shown) are all perpendicular to the $X_i - Y_i$ plane, ' $\mathbf{r}_{jj}$, denotes the common perpendicular vector between the two revolute joint axes of the j th link, and ' p_{ij} denotes a vector pointing from the j th revolute joint to point **P** in the moving platform. Following the above definitions, we have ${}^{i}\mathbf{p}_{i]} = {}^{i}\mathbf{r}_{i1} + {}^{i}\mathbf{r}_{i2} + {}^{i}\mathbf{r}_{i3}$, ${}^{i}\mathbf{p}_{i2} = {}^{i}\mathbf{r}_{22} + {}^{i}\mathbf{r}_{i3}$, and ${}^{i}\mathbf{p}_{i3} = {}^{i}\mathbf{r}_{i3}$.

We note that, excluding the prismatic joint, each limb form a planar 3-DOF open-loop chain. Since the revolute joints in each limb are free to rotate, linear actuator force ean only be transmitted in a direction parallel to the revolute joint axes; that is, along the local Z_i axis. When this force is resisted by the external force exerted on the moving platform, it generates a bending moment ${}^{4}\mathbf{n}_{jj}$ in each joint about an axis perpendicular to both vectors ${}^{2}\boldsymbol{P}_{ij}$ and Z_i as shown in Fig. 4. Assume that all links are nigid and the major source of compliance comes from the flexibility of the bearings in the joints. Then, the deflection between two members of a revolute joint can be modeled as an infinitesimal rotation about the axis of the bending moment. We call such an axis of deflection a "virtual axis" of compliance. The virtual axis of compliance coincides with the direction of the bending moment. Hence, we may cotnsider each limb together with the moving platform as a serial manipulator having three virtual axes of compliance.

Referring to Fig. 2, the vector ${}^{k}\mathbf{r}_{by}$ of each link can be expressed in the local coordinate system as follows:

$$ {}^{i}\mathbf{r}_{i1} = l_{i1}\begin{bmatrix} C\theta_{i1} \\ S\theta_{i1} \\ 0 \end{bmatrix},\ {}^{i}\mathbf{r}_{i2} = l_{t2}\begin{bmatrix} C\theta_{i2} \\ S\theta_{i2} \\ 0 \end{bmatrix},\ \text{and}\ {}^{i}\mathbf{r}_{i3} = l_{i3}\begin{bmatrix} C\theta_{i3} \\ S\theta_{t3} \\ 0 \end{bmatrix} \quad \text{(II-19)} $$

Following the definition of a local coordinate system, we have $\theta_{13} = 0, \theta_{23} = \pi/2$ and $\theta_{33} = \pi$ (sce Fig. 1).

As shown in Fig. 4 , let ' $\mathbf{n}_i$, be the reaction moment exerted on link j by link $f - 1$ of the i th limb, and ${}^{1}\mathbf{f}_i = [0, 0_2 f_{iz}]^T$ be the force exerted on the moving platform by the ith limb. A simple moment balance analysis yields

$$ {}^{i}\mathbf{n}_{ij} = {}^{t}\mathbf{p}_{ij} \times {}^{i}\mathbf{f}_1. \quad \text{(II-20)} $$

A small angular deflection of one link with respect to the other about the virtual axis of compliance, $\delta\theta_{ij}$, can be modeled as

$$^{i}\delta\theta_{ij} = c_{ij}n_{ij} \qquad \text{(II-21)}$$

where c_{ij} is the angular compliance constant associated with the bearings of the j th joint of the i th limb, $p_{ij} = \| {}^{i}\mathbf{p}_{ij}\|$, and $\vec{r}_{ij} = {}^{i}\mathbf{n}_{ij} \| = p_{ij}f_{iz}$ since ${}^{i}\mathbf{p}_{ij}$ and ${}^{i}\mathbf{f}_{i}$, are perpendicular.

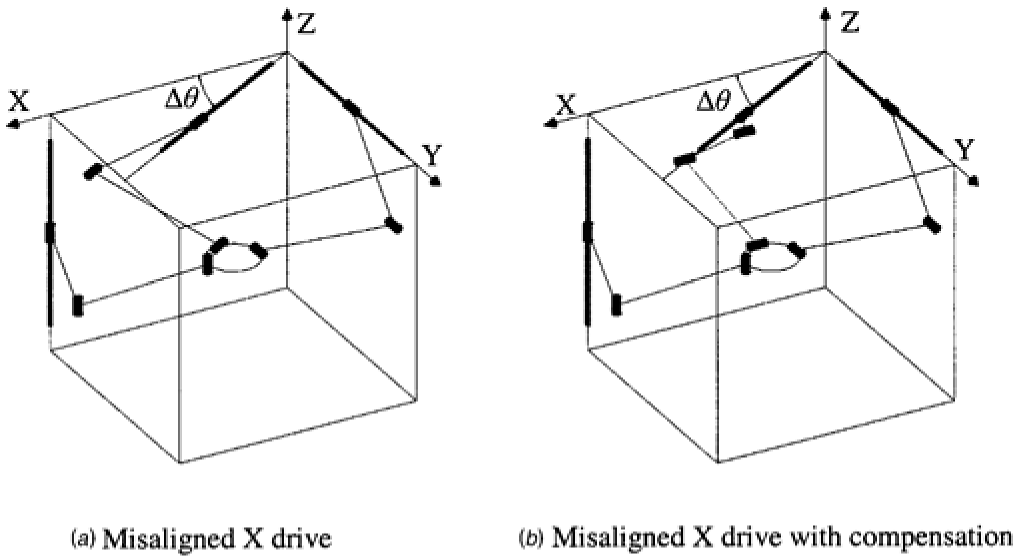


Figure II.5 Manipulator with a misaligned X actuator

Since the displacement of the moving platform caused by an infinitesimal rotation of $\delta\theta_{ij}$ about the j th joint axis of the $/$ th limb is equal to $\delta\theta_{ij}p_{ij}$ and it is parallel to the Z_i axis, the total displacement of the moving platform, eaused by infinitesimal rotations about the three revolute joints, is given by

$$(c_{i1}p_{i1}^2 + c_{i2}p_{i2}^2 + c_{i3}p_{i3}^2)f_{iz} \qquad \text{(II-22)}$$

From Eq. (18), it can be concluded that the X-component of the end-effector output force will only deflect the X-limb in the X-direction and so on. Hence, the overall deflection at the moving platform is simply a summation of the deflections of the three limbs. Equation (22) written three times, once for each limb, $i = 1{,}2$, and 3 , yields

$$\begin{aligned}
\delta p_x &= (c_{12}p_{11}^2 + c_{12}p_{12}^2 + c_{13}p_{13}^2)f_x \\
\delta p_y &= (c_{21}p_{21}^2 + c_{22}p_{22}^2 + c_{23}p_{23}^2)f_y \\
\delta p_z &= (c_{31}p_{31}^2 + c_{32}p_{32}^2 + c_{33}p_{33}^2)f_z
\end{aligned} \quad \text{(II-23)}$$

Writing Eq. (23) in matrix form, we obtain

$$\delta p = Cf \quad \text{(II-24)}$$

where $\delta\mathbf{p} = \left[\delta p_s, \delta p_y, \delta p_z\right]^T$ and $\mathbf{f} = \left[f_x, f_y, f_z\right]^T$. The matrix C, called the compliance matrix, is a 3×3 diagonal matrix whose diagonal elements are given by

$$C_{it} = c_{i1}p_{i1}^2 + c_{i2}p_{i2}^2 + c_{i3}p_{i3}^2 \quad \text{(II-25)}$$

Multiplying Eq. (25) by C^{-1}, we obtain

$$\mathbf{f} = K\delta\mathbf{x} \quad \text{(II-26)}$$

where $K = C^{-1}$ is called the stiffiness matrix of the manipulator.

II.4.4 Tolerance Consideration

In practice, it may be difficult to fabricate and assemble a perfectly orthogonal frame. In this section, we illustrate a method for compensating the manufacturing and assembling errors of the linear actuators by an example. We assume that the X actuator is skewed by a small angle $\Delta\theta$ about the global Z-axis, and the other actuators are mounted perfectly along the Y and Z axes, respectively, as shown in Fig. 5.

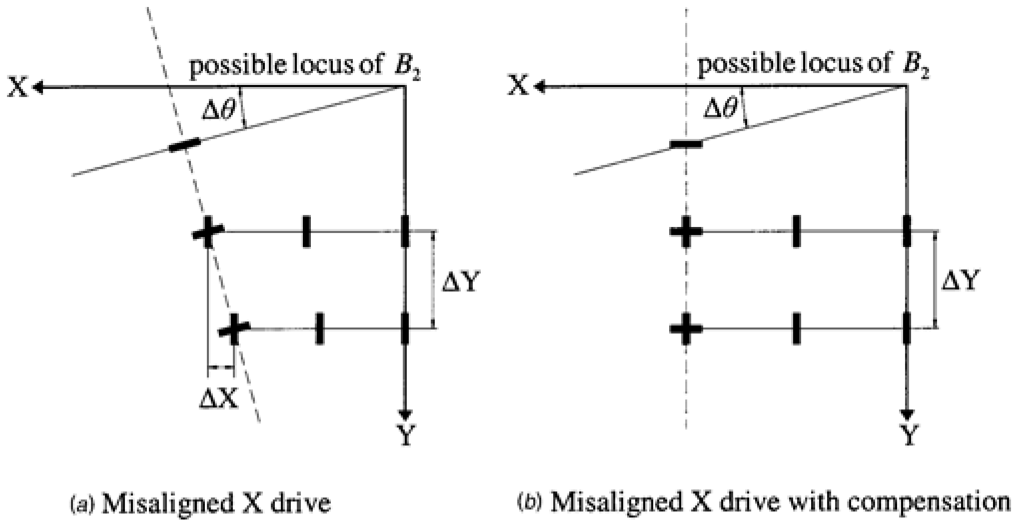


Figure II.6Top view of a misaligned actuator

First, we assume that the revolute joint axes in the X limb remain parallel to each other and to the prismatic joint axis as shown in Fig. $5(a)$. Figure $6(a)$ shows that with the X and Z actuators held stationary, the locus of B_2 follows the dotted line as

the Y-actuator moves along the Y-axis. Therefore, it can be concluded that if the revolute joint axes of the X limb remain parallel to the prismatic joint, the x, y, and z motions of the moving platform become slightly coupled.

The coupling problem can be solved by adjusting the orientation of the X limb such that the revolute joint axes are parallel to the global X-axis instead of the prismatic joint axis as shown in Fig. $5(b)$. In this case, the x, y, and z motions of the moving platform remain uncoupled as depicted in Fig. $6(b)$.

It should be noted that the three revolute joint axes in each limb must be always parallel to one another. Otherwise, the manipulator may not move or the motion may depend on the deflection of the links.

II.5 concluion

we have explored the kinematic analysis of a Cartesian parallel manipulator. By studying the forward and inverse kinematics, workspace analysis, singularity analysis, and motion characteristics, we gain a comprehensive understanding of the manipulator's behavior. This knowledge is essential for control, planning, and optimization of the manipulator's performance. The kinematic analysis serves as a foundation for further chapters, where we can delve into dynamic analysis, control strategies, and design optimization of the parallel manipulator.

Chapter III: Design Optimization

III.1 Introduction

After studying the general theoretical study of the parallel robot, we came to a practical studying. In this chapter, the optimization problem will be formulated in order to find the optimum dimensions of the studied robot by using MATLAB simulation software to extract the necessary results.

III.1.1 Workspace evaluation

Workspace evaluation is a common aspect of studying and optimizing the design of a parallel robot[25]. The workspace refers to the region in the 3D space that the robot's end effector can reach. Evaluating the workspace helps determine the range and limitations of the robot's motion, which is crucial for many applications.

To evaluate the workspace of a parallel robot, you can follow these steps:

1. Define the geometric and kinematic constraints: Specify the physical constraints of the robot, such as the length of the robot's limbs, joint limits, and any other geometric or kinematic limitations that affect its workspace.
2. Formulate the forward kinematics: Develop the mathematical equations that describe the relationship between the robot's joint angles and the position and orientation of its end effector. This allows you to compute the robot's pose for a given set of joint angles.
3. Generate a grid of joint angles: Define a grid or set of joint angle values within the allowed range for each joint of the parallel robot. This grid will be used to sample the workspace.
4. Compute the end effector poses: Use the forward kinematics equations to compute the position and orientation of the robot's end effector for each combination of joint angles in the grid.
5. Visualize the workspace: Utilize visualization tools or software, such as Matlab, to plot and visualize the reachable workspace of the parallel robot. This could be represented as a 3D volume or surface that depicts the region the end effector can reach.

You've highlighted some key points regarding the optimization of parallel robots. The geometry of parallel robots plays a crucial role in determining their performance. As a result, optimization problems have gained significant interest in the field of parallel robotics.

The performance of parallel robots is highly dependent on their topology and dimensions. The choice of topology, including the number and arrangement of kinematic chains and joints, greatly influences the robot's capabilities. Additionally, dimensioning parameters such as link lengths and joint angles affect the robot's workspace, dexterity, and other performance aspects.

To achieve desired performance characteristics, it is essential to introduce performance indices or optimization criteria to quantify and evaluate the robot's performance[26]. These metrics help in characterizing the robot's capabilities and enable comparisons between different designs. By formulating optimization problems, researchers and engineers can systematically search for optimal design parameters that maximize desired performance indices.

Among various performance metrics, the workspace holds primary importance in the design of parallel robots. Maximizing the workspace is often a key objective in the optimal design process. A larger workspace allows the robot to reach more positions and orientations, increasing its versatility and applicability in various tasks.

Overall, optimizing the geometry and dimensions of parallel robots is crucial to enhance their performance. Through the use of performance indices and optimization criteria, researchers and engineers aim to design robots with expanded workspaces and improved capabilities, making them more effective and efficient in fulfilling desired tasks.

III.1.2 Optimization Process

A design improvement process is indeed a valuable tool for achieving better designs or design improvements. It involves visualization and analysis techniques that help designers identify areas for enhancement and make informed decisions to optimize the design[27].It is a mathematical procedure for determining optimal solutions by representing all the complexities of the design in the form of design variable(s), objectives function(s) and/or constraint(s).

The basic elements of any constrained optimization problem are:

III.1.3 Objective function

An objective function or vector of objective functions is the mathematical expression that expresses the optimization goal in terms of design variables. Optimization process is required to either minimize or maximize the objective function. For instance, in robotics, maximization of the workspace or minimization of inertia/mass of a manipulator can be the objective functions.

III.1.4 Design variables

Design variables, also referred to as decision variables, are the unknowns in an optimization problem that the designer seeks to determine[28]. These variables represent the controllable numeric values that can be adjusted to achieve the desired objectives and affect the value of the objective function. Design variables can be continuous (such as a length/diameter/cross-section of the robot links) or discrete (such as the number of links/joints in a robot).

III.1.5 Constraints

Constraints play a crucial role in the design optimization process by imposing limits and conditions that the solution must satisfy. These constraints are mathematical expressions that combine the design variables and define the boundaries and limitations of the feasible solution space.

III.1.6 Variable bounds

Design variables are not usually permitted to take any value. Instead, these are usually have lower and upper limits, known as variable bounds. Variable bounds limits the design space and along with constraints, used to distinguish the solutions as feasible or unfeasible.
The optimization problem is then:

Find the values of the design variables that minimize or maximize the objective function while satisfying the constraints. Remembering that variables describe all situations and constraints describe all feasible situations Mathematically, an single-objective optimization problem can be expressed as:

$$\min_{x} f \text{ subject to: } \begin{cases} g_i(x) \leq 0 & i = 1, \dots . m \\ h_j(x) = 0 & j = 1, \dots . n \\ x_l \leq x \leq x_u \end{cases}$$

Where, x is the vector of design variables, f is the objective function to be minimized subject to m inequality and n equality constraints given by $g_i(x)$ and $h_j(x)$, respectively.

Optimization is an iterative process and involves at least some degree of trial and errors. As the problem complexity is increased, the search procedure becomes tedious and may not guarantee a solution in all cases.

The main steps involved in solving an optimization problem can be cited as:

- understand the problem, by drawing a diagram or flow chart which represents the problem;
- write a problem formulation in words, including decision variables, objective function, and constraints;
- write the algebraic formulation of the problem;
    - define the decision variables;
    - write the objective function(s);

- write the constraints;

- develop a spread sheet model;
- set up the Solver settings and solve the problem;
- examine the results and make corrections to the model;
- analyse and interpret the results.

III.1.7 Optimization method (A genetic algorithm (GA))

Genetic algorithms are a type of search technique used in computing to solve optimization and search problems. They belong to the class of (EA), which are inspired by the principles of natural evolution[29].

In a genetic algorithm, a population of potential solutions is evolved over generations using operators such as inheritance, mutation, selection, and crossover. The individuals in the population are represented by chromosomes or strings of genes, which encode the design variables or parameters of the problem. The algorithm evaluates the fitness of each individual based on an objective function and selects individuals for reproduction based on their fitness. Through the process of genetic operations such as crossover and mutation, new offspring are generated, inheriting characteristics from their parent solutions. Over time, the population evolves towards increasingly better solutions through iterations and selection mechanisms.

Genetic algorithms have been successfully applied to solve complex problems in various domains, including biology, engineering, computer science, and social science[30]. They are particularly useful in situations where the problem space is large, the solution space is complex or poorly understood, and traditional optimization methods may not be efficient or effective. Genetic algorithms offer a global search capability, exploring a wide range of possible solutions and often finding good approximate solutions to difficult optimization problems.

Genetic programming, on the other hand, is a variant of genetic algorithms that focuses on evolving computer programs or algorithms rather than individual solutions. Genetic programming aims to find a program or algorithm that can map input data to a desired output without relying on a predefined formula or model. By evolving the structure and parameters of programs, genetic programming can discover novel and effective solutions to problems that are not easily solved using traditional programming or mathematical approaches.

III.1.7.1 Genetic algorithm methodology

- **Initialization**

At first huge numbers individual results need aid haphazardly created to structure a starting number. Those populace span relies on the way of the problem, in any case commonly holds a few hundred or many workable results. Traditionally, the populace may be produced randomly, coating those whole extend for could be allowed results (the hunt space). Occasionally, those results might a chance to be “seeded” On regions the place ideal results are inclined to make discovered.

- **Selection**

Throughout every progressive generation, an extent of the existing populace may be chose will breed another era. Singular results are chosen through a fitness-based process, the place fitter results (as measured Eventually Tom's perusing a wellness function) need aid normally less averse will be chose. Certain Choice strategies rate the wellness for each result Also preferentially select those best results. Different techniques rate best An arbitrary example of the population, Likewise this methodology might be extremely drawn out[31].

- **Reproduction (mutation)**

The next step is to generate a second-generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation[32].

- **Termination**

This generational cycle is rehashed until an end condition has been reached. Basic ending conditions are:

- A solution is found that satisfies minimum criteria;
- Fixed number of generations reached;
- Allocated budget (computation time/money) reached;
- The highest-ranking solution’s fitness is reaching or has reached a plateau such that successive iterations no longer produce better results;
- Manual inspection;
- Combinations of the above.

III.2 Case study: Mono-Objective Design Optimization of CPM

The design optimization approach you described aims to maximize the stiffness of a manipulator while considering a given workspace volume. To quantify the stiffness, the average stiffness constant in the sense of Euclidean norm is used as the objective function.

$$k_{\text{ave}} = \sqrt{k_{11}^2 + k_{22}^2 + k_{33}^2} \tag{III-1}$$

where k_{ii} for $i = 1,2,$ and 3 denote the diagonal elements of the stiffness matrix K. We assume that $c_{ij} = c$ for all joints. It follows from Eq. (25) that the average stiffness constant can be factored into the product of a constant term and a geometric term,

$$k_{\text{ave}} = \frac{1}{c}\sqrt{\frac{1}{\left(\sum_{j=1}^{3} p_{1j}^2\right)^2} + \frac{1}{\left(\sum_{j=1}^{3} p_{2j}^2\right)^2} + \frac{1}{\left(\sum_{j=1}^{3} p_{3j}^2\right)^2}}. \tag{III-2}$$

Multiplying both side of Eq. (28) by c eliminates the constant term. Hence, a local design index (LDI) is defined as

$$\text{LDI} = ck_{\text{ave}} = \sqrt{\frac{1}{\left(\sum_{j=1}^{3} p_{1j}^2\right)^2} + \frac{1}{\left(\sum_{j=1}^{3} p_{2j}^2\right)^2} + \frac{1}{\left(\sum_{j=1}^{3} p_{3j}^2\right)^2}}. \tag{III-3}$$

The workspace of the manipulator may be limited by the link lengths and angular rotation limits of the revolute joints, or by the stroke lengths of the linear actuators. However, it is preferable to make the limb lengths sufficiently long such that the ranges of motion of all linear actuators can be fully utilized. To achieve this goal, we impose the constraint that the workspace volume is always equal to the product of the stroke lengths of the linear actuators[33]. Hence, the optimal design problem is formulated as

$$\begin{aligned} &\text{Maximize: } \eta = \frac{1}{W}\int_W \text{LDI}\, dW \\ &\text{Subject to: } W = \Delta d_1 \times \Delta d_2 \times \Delta d_3 \end{aligned} \tag{III-4}$$

where W and dW denote the total workspace volume and a differential workspace, respectively, and Δd_i represents stroke length of the i th linear actuator. We call η the global design index. Dividing the global design index, η, by the angular compliance constant, c, gives a linear stiffness. Hence, the global design index can be interpreted as an average stiffness in the workspace. Since the stroke lengths are determined by a desired workspace volume, they are treated as constant parameters during the optimization process. the design variables of the optimization problem are

$$\left[d_{0i}, l_{i1}, l_{i2}, l_3, e_x, e_y, e_z\right], \text{ for } i = 1{,}2\text{, and } 3. \quad \text{(III-5)}$$

In view of the difficulty in obtaining a closed-form solution to Eq. (30), the Monte Carlo method is employed for the optimization. Specifically, given a set of design parameters, the following procedure is used for the computation of the global design index[34].

- Given the stroke lengths of the linear actuators, define a rectangular box with a total volume of $W = \Delta d_1 \times \Delta d_2 \times \Delta d_3$.
- Divide the box into a set of orthogonal grids and calculate the total number of grid points as n_{total} . Initialize the number of points, $n_{\text{workspace}}$, that fall within the workspace of the manipulator to zero.
- For every grid point, two conditions are evaluated to check if it falls within the workspace of the manipulator:

$$\begin{gathered}(l_{i1} - l_{i2})^2 \leqslant B_{ix}^2 + {}^iB_{iy}^2 \leqslant (l_{i1} + l_{i2})^2, \\ \theta_L \leqslant |\theta_{i2}^*| \leqslant \theta_H.\end{gathered} \quad \text{(III-6)}$$

The first constraint comes from the minimum and maximum reaches of a 2-link serial chain. The second constraint is introduced to keep the serial chain away from the fully stretched-out, $\theta_{i2}^* = 0$, and folded-back, $\theta_{i2}^* = \pm\pi$, configurations. for the definition of θ_{i2}^*. The upper and lower joint limits of θ_{i2}^* are generally chosen such that $\pi - \theta_H = \theta_L \geqslant \theta_{\text{margin}}$ (say 15 15). Note that if margin angle, θ_{margin} , is too large, the workspace may be limited by the minimum and maximum reaches of the serial chains.

- If the point satisfies the above two conditions, increment $n_{\text{workspace}}$ by one and calculate the LDI by using Eq. (29).

- Repeat the above process for every grid point until all the points are accounted for and sum up the LDI for all points that fall within the workspace, $S = \Sigma \text{LDI}$.
- If $n_{\text{workspace}}$ is less than n_{total}, the set of design parameters is judged to be infeasible, since the equality constraint imposed by Eq. (30) is violated.
- If $n_{\text{workspace}} = n_{\text{total}}$, compute the global design index as:

$$\eta = \frac{S}{n_{\text{workspace}}} \tag{III-7}$$

Note that when link lengths are too long, $|\theta_{i2}^*|$ may exceed the upper limit θ_H in a region around the origin. On the other hand, when the link lengths are too short, $|\theta_{i2}^*|$ may drop below the lower limit θ_L in a region far away from the origin.

In this section, we optimize the design of a prototype manipulator. In this design, the stroke lengths of the linear actuators are chosen as follows:

$$\Delta d_1 = \Delta d_2 = 300 \text{ and } \Delta d_3 = 250(\text{ mm}).$$

To make the X and Y limbs symmetrical, we assume that

$$\begin{gathered} d_{0p} = d_{01} = d_{02},\ d_{0z} = d_{03},\ e_z = 0, \\ l_{p1} = l_{11} = l_{21},\ l_{z1} = l_{31},\ l_{p2} = l_{12} = l_{22},\ l_{z2} = l_{32}. \end{gathered} \tag{III-8}$$

Therefore, we are left with the following design variables:

$$\left[d_{0p}, d_{0z}, e_x, e_y, l_{p1}, l_{p2}, l_{z1}, l_{z2}, l_3\right] \tag{III-9}$$

The lower and upper joint angle limits are set at

$$\theta_L = 30^\circ \text{ and } \theta_H = 150^\circ.$$

To avoid mechanical interference, the following inequality constraints are imposed:

$$d_{0p} \geqslant 225,\ d_{0z} \geqslant 0 \text{ and } l_3 \geqslant 105(\text{ mm}).$$

In order to locate the third actuator away from the Z-axis, we add

$$e_x \geqslant d_{0p} + \Delta d_1 + l_3 \text{ and } e_y \geqslant 224(\text{ mm}).$$

Table 1 Optimized design parameters.

Design Variables	Optimum Values (cm)
d_{0p}	150.00
d_{0z}	150.00
e_x	0.00
e_y	0.00
l_{p1}	19.20
l_{m2}	19.80
l_{z1}	19.00
$l_{l_{22}}$	19.50
l_3	20.00

Finally, to prevent the elbows of the first and second limbs from hitting a workpiece in the prescribed workspace, two additional inequality constraints are added:

$$\theta_{12} \leqslant 30° \text{ and } \theta_{22} \geqslant 60°.$$

Sequential Quadratic Programming, which is known to be more efficient than the method of penalty function, is used as the solution technique for the constrained optimization problem. The total number of grid points used for the optimization is $41 \times 41 \times 31 = 52{,}111$. The optimized value of the global design index is $\eta = 3.2729\ \mathrm{m}^{-2}$. Table 1 shows the optimal design parameters.

General conclusion

In conclusion, the design and optimization of a Cartesian parallel manipulator is a complex and multidisciplinary process. It involves considering various factors such as kinematics, dynamics, structural analysis, and optimization techniques to achieve the desired performance criteria for a specific application.

By carefully formulating the design problem and setting clear objectives, engineers can analyze the kinematic and dynamic behavior of the manipulator. They can evaluate factors such as workspace, stiffness, accuracy, payload capacity, and energy efficiency to guide the design process.

The choice between linear and rotary actuation methods depends on specific requirements and constraints. The linear actuation method offers decoupled translational motion and allows for all actuators to be installed on the fixed base, simplifying the assembly. On the other hand, the rotary actuation method provides rotational motion and may require additional considerations for coupling effects.

Throughout the design process, optimization algorithms help search for the optimal design parameters, taking into account various constraints and objectives. Simulation and validation play a crucial role in assessing the performance and functionality of the design, while sensitivity analysis helps understand the robustness and reliability of the manipulator.

By iteratively refining the design based on simulation results and incorporating necessary modifications, engineers can achieve an optimal design that meets the desired performance criteria. This leads to a Cartesian parallel manipulator that exhibits superior motion capabilities, precision, stability, and efficiency.

The optimal design of a Cartesian parallel manipulator enables its successful deployment in a wide range of applications, including robotics, manufacturing, medical devices, and more. It provides enhanced capabilities for precise positioning, manipulation, and control in three-dimensional space.

Overall, the design and optimization of a Cartesian parallel manipulator require a comprehensive understanding of the system's kinematics, dynamics, and structural characteristics. By leveraging advanced analysis techniques and optimization algorithms, engineers can create highly efficient and effective manipulators that meet the specific needs of various industries and applications.

References

[1] D. V. Rosato, *Plastics processing data handbook*. Springer Science & Business Media, 2012.

[2] J. L. Myers, A. Well, and R. F. Lorch, *Research design and statistical analysis*. Routledge, 2010.

[3] M. S. Sisodiya, "OPTIMIZATION OF JOINTS AND LINKS IN PLANAR PARALLEL ROBOT MECHANISMS".

[4] J.-B. Izard *et al.*, "Large-scale 3D printing with cable-driven parallel robots," *Construction Robotics,* vol. 1, pp. 69-76, 2017.

[5] F. Pierrot, V. Nabat, O. Company, S. Krut, and P. Poignet, "Optimal design of a 4-DOF parallel manipulator: From academia to industry," *IEEE Transactions on Robotics,* vol. 25, no. 2, pp. 213-224, 2009.

[6] A. Gasparetto and L. Scalera, "From the Unimate to the Delta robot: the early decades of Industrial Robotics," in *Explorations in the History and Heritage of Machines and Mechanisms: Proceedings of the 2018 HMM IFToMM Symposium on History of Machines and Mechanisms*, 2019: Springer, pp. 284-295 .

[7] J. Yuan and J. Ni, "The real-time error compensation technique for CNC machining systems ",*Mechatronics,* vol. 8, no. 4, pp. 359-380, 1998.

[8] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research,* vol. 30, no. 11, pp. 1343-1377, 2011.

[9] K. Danziger, *Constructing the subject: Historical origins of psychological research*. Cambridge University Press, 1994.

[10] G. Vachtsevanos, K. Davey, and K.-M. Lee, "Development of a novel intelligent robotic manipulator," *IEEE Control Systems Magazine*, vol. 7, no. 3, pp. 9-15, 1987.

[11] A. D. Udai, A. A. Hayat, and S. K. Saha, "Parallel active/passive force control of industrial robots with joint compliance," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014: IEEE, pp. 4511-4516 .

[12] Z. Pandilov and V. Dukovski, "COMPARISON OF THE CHARACTERISTICS BETWEEN SERIAL AND PARALLEL ROBOTS," *Acta Technica Corviniensis-Bulletin of Engineering,* vol. 7, no. 1, 2014.

[13] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *Journal of Robotic systems,* vol. 5, no. 6, pp. 527-552, 1988.

[14] P. Jamwal, S. Xie, Y. Tsoi, and K. Aw, "Forward kinematics modelling of a parallel ankle rehabilitation robot using modified fuzzy inference," *Mechanism and Machine Theory,* vol. 45, no. 11, pp. 1537-1554, 2010.

[15] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer graphics forum*, 2 ,018vol. 37, no. 6: Wiley Online Library, pp. 35-58 .

[16] D. N. Nenchev, "Restricted Jacobian matrices of redundant manipulators in constrained motion tasks," *The International journal of robotics research,* vol. 11, no. 6, pp. 584-597, 1992.

[17] Z. Gao ,D. Zhang, and Y. Ge, "Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches," *Robotics and Computer-Integrated Manufacturing,* vol. 26, no. 2, pp. 180-189, 2010.

[18] F. A. Lara-Molina and D. Dumur, "Robust multi-objective optimization of parallel manipulators," *Meccanica,* vol. 56, no. 11, pp. 2843-2860, 2021.

[19] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, 1999.

[20] A. M. Mirzendehdel, M. Behandish, and S. Nelaturi, "Exploring feasible design spaces for heterogeneous constraints," *Computer-Aided Design,* vol. 115, pp. 323-347, 2019.

[21] V. Machairas, A. Tsangrassoulis, and K. Axarli, "Algorithms for optimization of building design: A review," *Renewable and sustainable energy reviews,* vol. 31, pp. 101-112, 2014.

[22] H. S. Kim and L.-W. Tsai, "Evaluation of a Cartesian parallel manipulator," *Advances in Robot Kinematics: Theory and Applications,* pp. 21-28, 2002.

[23] M.-J. Liu, C.-X .Li, and C.-N. Li, "Dynamics analysis of the Gough-Stewart platform manipulator," *IEEE Transactions on robotics and automation,* vol. 16, no. 1, pp. 94-98, 2000.

[24] J. K. Mohanta, S. Mohan, P. Deepasundar, and R. Kiruba-Shankar, "Development and control of a new sitting-type lower limb rehabilitation robot," *Computers & Electrical Engineering,* vol. 67, pp. 330-347, 2018.

[25] Z. Zhang, Z. Shao, and L. Wang, "Optimization and implementation of a high-speed 3-DOFs translational cable-driven parallel robot ",*Mechanism and Machine Theory,* vol. 145, p. 103693, 2020.

[26] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots,* vol. 38, pp. 65-88, 2015.

[27] H.-L. Chi, X. Wang, and Y. Jiao, "BIM-enabled structural design: impacts and future developments in structural modelling, analysis and optimisation processes," *Archives of computational methods in engineering,* vol. 22, pp. 135-151, 2015.

[28] H. Agarwal, J. E. Renaud, E. L. Preston, and D. Padmanabhan, "Uncertainty quantification using evidence theory in multidisciplinary design optimization," *Reliability Engineering & System Safety,* vol. 85, no. 1-3, pp. 281-294, 2004.

[29] T. Hanne, R. Dornberger, T. Hanne, and R. Dornberger, *Computational intelligence*. Springer, 2017.

[30] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[31] M. Kumar, M. Husain, N. Upreti, and D. Gupta, "Genetic algorithm: Review and application," *Available at SSRN 3529843,* 2010.

[32] G. K. Nune and P. V. Sena, "Novel artificial neural networks and logistic approach for detecting credit card deceit," *International Journal of Computer Science and Network Security (IJCSNS),* vol. 15, no. 9, p. 21, 2015.

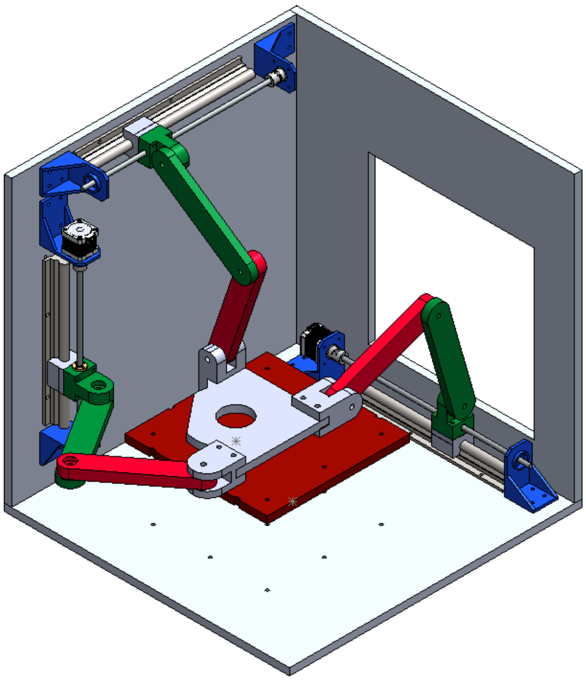
[33] A. J. Madhani, "Design of teleoperated surgical instruments for minimally invasive surgery," Massachusetts Institute of Technology, 1998 .

[34] A. Gupta and C. D. Maranas, "A two-stage modeling and solution framework for multisite midterm planning under demand uncertainty," *Industrial & Engineering Chemistry Research,* vol. 39, no ,10 .pp. 3799-3813, 2000.

Annexes:

Notation of links in layout graphs

embody this image Optimal design of a 3DoF planar parallel robot



Abstract

Report title: Design Optimization of a Cartesian Parallel Manipulator.

Master: Electromechanical.

Keywords: parallel robot, objective function, modelling, kinematic, workspace, forward kinematics and inverse kinematics.

Authors: Tedjani Mohammed Laid, Khelaifa Ammar, Kouider Abdallah, Cherrahi Boubaker.

Abstract:

This memory focuses on optimization the conception (design) of a parallel planar robot with theree degrees of freedom (3 DOF), and we have chosen the Biglid robot in this work. This work includes the geometric and kinematic study of the chosen robot, and this step is important given that the performance standards for a particular structure depend heavily on the dimensions of this structure, and then we were interested in studying the problem of optimizing the workspace of the selected structure with the help of the genetic algorithm (GA).

Titre du mémoire : Design Optimization of a Cartesian Parallel Manipulator.

Mots clés : robot parallèle, fonction objectif, modélisation, cinématique, espace de travail, cinématique directe et cinématique inverse.

Résumé :

Ce mémoire porte sur l'optimisation de la conception (design) d'un robot planaire parallèle à trois degrés de liberté (3 DOF), et nous avons choisi le robot Biglid dans ce travail. Ce travail comprend l'étude géométrique et cinématique du robot choisi, et cette étape est importante étant donné que les normes de performance pour une structure particulière dépendent fortement des dimensions de cette structure, et puis nous nous sommes intéressés à étudier le problème de l'optimisation de l'espace de travail de la structure sélectionnée à l'aide de l'algorithme génétique (GA).

عنوان المذكرة: التصميم الأمثل للروبوت المتوازي المستوي مع 3 درجات من الحرية.

الكلمات المفتاحية: الروبوت الموازي ، الوظيفة الموضوعية ، النمذجة ، الحركية ، مساحة العمل ، الحركية إلى الأمام ، الحركية العكسية.

الملخص:

تركز هذه المذكرة على تحسين تصميم روبوت مستوٍ متوازي بثلاث درجات من الحرية (DOF 3)، وقد اخترنا روبوت ثلاثي الانزلاق في هذا العمل، يشمل هذا العمل الدراسة الهندسية والحركية للروبوت المختار، وهذه الخطوة مهمة بالنظر إلى أن معايير الأداء لهيكل معين تعتمد بشكل كبير على أبعاد هذه البنية، ومن ثم اهتممنا بدراسة مشكلة تحسين مساحة عمل الهيكل المختار بمساعدة الخوارزمية الجينية (GA).