ON THE DESIGN OF MULTI-PLATFORM PARALLEL MECHANISMS

Andrew Johnson

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

May 2017

ABSTRACT

Parallel mechanisms have been examined in more and more detail over the past two decades. Parallel mechanisms are essentially the same design layout, a base, multiple legs/limbs, and a moving platform with a single end-effector to allow the mechanism to complete its desired function.

Recently, several research groups have begun looking into multiple-platform parallel mechanisms and/or multiple end-effectors for parallel mechanisms. The reason for the research in this new form of parallel mechanism stems from multiple sources, such as applications that would require multiple handling points being accessed simultaneously, a more controlled gripper motion by having the jaws of the gripper being attached at different platforms, or to increasing the workload of the mechanism.

The aim of the thesis is to modify the design process of parallel mechanisms so that it will support the development of a new parallel mechanism with multiple platforms capable of moving relative to each other in at least 1-DOF and to analyse the improvements made on the traditional single platform mechanism through a comparison of the power requirements for each mechanism.

Throughout the thesis, a modified approach to the type synthesis of a parallel mechanism with multiple moving platforms is proposed and used to create several case study mechanisms. Additionally, this thesis presents a new series of methods for determining the workspace, inverse kinematic and dynamic models, and the integration of these systems into the design of a control system. All methods are vetted through case studies where they are judged based on the results gained from existing published data.

Lastly, the concepts in this thesis are combined to produce a physical multi-platform parallel mechanism case study with the process being developed at each stage. Finally, a series of proposed topics of future research are listed along with the limitations and contributions of this work.

Acknowledgements

First, I would like to thank my primary supervisor, Dr. Xianwen Kong, who has given me the opportunity to work throughout this doctoral project both technically and financially. I will always be grateful for him supervising me throughout my study and research and for the encouragement and continuous support that he has given me throughout both the project and when times become difficult in my personal life. To Dr. Kong, I cannot thank you enough for your financial support, excellent supervision, time, patience, and tolerance.

I would also like to thank my second supervisor, Professor James Ritchie, for his invaluable guidance in during our meetings. Professor James Ritchie was invaluable during my study by helping me to clarify my research objectives, his support and reassurance that the thesis would eventually be finished, and always being available to talk to should I need him.

I am also very grateful to: Dr. Mathew Dunningan, who reviewed my first-year report and provided helpful comments; Alistair Blyth, as well as the entire Mechanical Engineering Department workshop technician team, whom ensured that all of the component parts for the prototype were made both on time and accurate according the drawings I provided and were always happy to modify parts if they required it; the entire Electrical Engineering Department workshop technician team for ensuring that the servomotor drivers were contained so that they could be used safely during operation; the support staff at both Maplesoft and National Instruments for their support when problems arose with the various software I used throughout the thesis; and all the staffs and students of Heriot-Watt University who helped either keep me to sane and progressing with my thesis.

I would especially like to thank all of my friends and family who have given me constant support, both financial and emotional throughout my PhD study. The invaluable support that they have given me has ensured that I was able to complete this thesis despite trying times in my personal life.

Lastly, I would like to say that I am extremely grateful to the support I received from the EPSRC DTA that allowed me to be able to finance my life throughout this project.

List of Publications

X. Kong, and A. Johnson, “Classification of Screw Systems Composed of Three Planar Pencils of Lines”, Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference, 2012, IDETC/CIE 2012, DETC2012-70636.

X. Kong, and A. Johnson, “Classification of Screw Systems Composed of Three Planar Pencils of Lines for Singularity Analysis of Parallel Mechanisms”, Journal of Mechanisms and Robotics, vol. 6 (2), 2014.

A. Johnson, and X. Kong, “Determination of the Workspace of Parallel Manipulators Using a CAD Software and the Concept of Virtual Chains”, Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference, 2013, IDETC/CIE 2013, DETC2013-12586.

A. Johnson, X. Kong, and J. Ritchie, “Determination of the Workspace of a Three-Degrees-of-Freedom Parallel Manipulator Using a Three-Dimensional Computer-Aided-Design Software Package and the Concept of Virtual Chains”, Journal of Mechanisms and Robotics, vol. 8 (2), 2016.

Declaration Statement

(Research Thesis Submission Form should be placed here)

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviation	Definition
R	Rotation/Revolute joint
U	Universal joint
P	Prismatic/Translational joint
S	Spherical joint
C	Cylindrical joint
$\overline{R}, \overline{P}$	Actuated Joint
IKM	Inverse Kinematic Model
FKM/DKM	Forward Kinematic Model or Direct Kinematic Model
GUI	Graphical User Interface
RKM	Redundant Kinematic Model
DOF	Degree(s) of Freedom
rDOF	Relative Degree(s) of Freedom
VC	Virtual Chain
$\mathcal{F}$	Mobility or relative DOF of a kinematic chain
n	Number of links including the base
g	Number of joints
f_j	Degree of freedom of the j^{th} joint
d	Number of independent constraint equations within a loop
v	Number of independent loops in the mechanism
$\mathcal{W}$	Wrench system of the mechanism or kinematic chain

Chapter 1 Introduction

When investigating rigid-body robotics, there are typically two basic categories of mechanism that are researched, these are serial and parallel mechanisms. Serial mechanisms are constructed by having a base for the mechanism connected to an end-effector by a single kinematic chain/limb; whereas parallel mechanisms are typically constructed with the base and end-effector/moving platform being connected by two or more kinematic chains; this chapter will detail several types of the latter in the form of a literature review. This is then followed by a brief review of the primary form of specialised mathematics used in the development of parallel mechanisms theory as well as several technologies to be utilised as part of the work covered in this thesis.

1.1 Parallel Mechanisms

The most common form of robotic mechanisms used in industry is that of a serial robot. A serial robot is constructed from a series of n links connected together by $(n-1)$ joints. All the joints are typically independent and therefore form an open-loop system where each joint must be actuated in order to control its motion. The most common type of serial mechanism is designed with the human arm as a blueprint, usually consisting of a shoulder, elbow, and wrist design with the shoulder being attached to the base of the robot and the wrist being connected to the end-effector.

The SCARA Robot (figure 1.1) is known for representing the form of a simplified human arm, consisting of joints and actuators allowing for the mechanism to move its end-effector along the x-, y- and z-axes in a translational motion as well as being able to rotate the end-effector about its vertical z-axis. These types of motion, known as Schöenflies motion, are typically the desired motion types of a mechanism used in the pick-and-place industry.

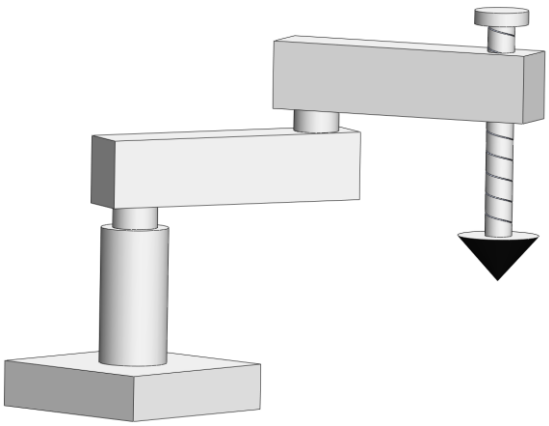

Figure 1.1: 4-degree of freedom (DOF) (3T-1R) SCARA Robot

Unlike a serial robot, a parallel robot is described as a robot made up of two or more kinematic chains or links beginning either from the same or different origins but leading to a single shared end-effector [1.1-1.3] producing a closed-loop system. In a parallel robot, the joints on or close to the base plate are actuated on each kinematic link; this reduces the overall moving mass of the robot and allows lower torque motors to be used to drive the robot during its operation. This concept was developed from a universal tyre testing machine invented in the 1950s [1.4] into several designs including well-known

robots such as the Delta, Hexa, H4, and Agile Eye mechanisms [1.5] and is still used today in architecture types; each of which have their own operating method and achieve their design goals in various ways.

Parallel and serial robots each have their own advantages over the other. Serial robots have a large workspace and are relatively cheap to produce; they also mimic nature rather well in the sense that a serial robot arm can easily be designed to represent a human arm or leg allowing for a wide variety of task and applications potentially dangerous for humans to attempt themselves. Parallel robots have a much smaller workspace but can operate at incredibly high speeds and can perform rapid accelerations during operation; they also have the ability to lift much larger masses due to the increased number of limbs sharing the load [1.2][1.6]. Parallel robots also have a much lower moving mass than their serial counterparts do as the actuators tend to be fixed on the mechanism base or cage. They are also much easier to model and simulate resulting in faster computations for use in trajectory planning during operation [1.2]. Parallel robots are also capable of very high precision when handling objects or moving around a load that requires accurate motion, i.e. the precise measurements of an object or structure [1.7] [1.8]. Due to these advantages over serial robots, parallel robots are becoming increasingly more likely to be found in factories performing pick-and-place tasks which would normally require a large work force to accomplish; examples of this would be the sorting of end products into shipping boxes and the movement of products from one conveyor belt to another in a production line.

The current procedure for developing a new parallel mechanism starts with the determination of what the mechanism will be doing, what physical and environmental limitations will it be expected to operate within and any marketing, storage and sales requirements if the mechanism is to be commercialised. These factors are used to produce a product design specification, which in turn is used to outline the desired mechanism's functionality. While there are many variants on the design process of parallel mechanisms, the procedure is usually constructed as shown in the flow chart detailed in figure 1.2.

As can be seen in figure 1.2, the procedure determines the desired limb configuration using type synthesis first before developing the inverse kinematic model (IKM), the equations of which dictate how the mechanism moves from one location to another. This is then used to produce the workspace of the mechanism using theorised link lengths and limiting the motion of the joints to a range that should avoid singularities from occurring. Once the workspace of the desired mechanism is determined, a dynamic and static analysis of the mechanism can be performed by calculating the mass of the joints, links and end-effector in order to determine the forces or torques required to actuate the mechanism. It is usual at this stage that a CAD model of the mechanism is developed to aid in the development of the static analysis of the system through finite element analysis (FEA) software that determines if the mechanism will remain ridged and unbending during the mechanism's operational life cycle. This process relies heavily on the designer having an in depth knowledge of high end mathematics such as screw theory, complex geometry and machine dynamics theory in order to fully develop the desired mechanism. Additionally, the design of the mechanism is typically checked at the fabrication of a

prototype stage that can potentially result in a long delay in the development process in the event that an issue is found.

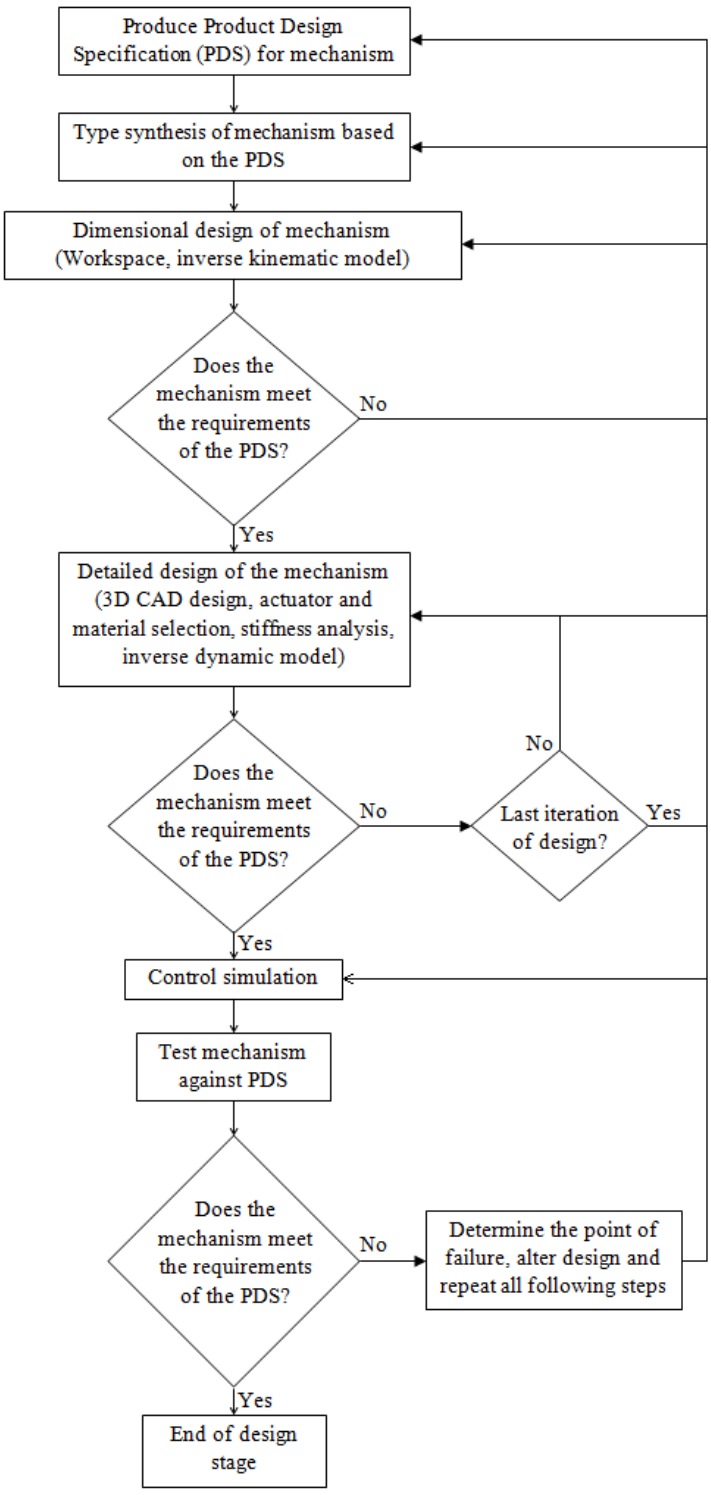

Figure 1.2: Parallel mechanism design process flow chart.

1.2 State of the Art Parallel Mechanisms

In 2001 Y. Li and G.M. Bone asked the question, "Are parallel manipulators more energy efficient than their serial counterparts?" [1.9]. In order to investigate this, the pair took a 3-DOF serial robot arm constructed with a rotation/revolute (R) joint aligned along each

of the world axes ($\bar{R}_z\bar{R}_x\bar{R}_x$) and compared its energy efficiency to that of a 3-DOF parallel robot similar to the Delta robot. The two were compared by the efficiency of the number of motors on each robot (three DC motors for each) and the amount of power consumed during the motion of each robot from its start point to a designated end location using the inverse kinematics of each robot. The average absolute power consumption for the robots were calculated using the equation [1.9]:

$$P_a = \sum_{n=1}^{3} \frac{1}{T} \int_0^{\Gamma} |P_{elect}| dt \quad (1.1)$$

where n is the motor number, T is the consumption period (how long the motor is running for), P_{elect} is the instantaneous motor power usage and Γ is the motor torque equation for each motor.

From their calculations they discovered that on average, parallel manipulators consumed only 26% [1.9] of the energy consumed by their serial counterparts. This revealed that when a task that can be performed effectively by either a parallel or a serial robot with an equal number of DOF, the parallel robot is the more economical and environmentally friendly option, providing that the parallel robot has been designed with efficiency in mind.

1.2.1 Conventional parallel mechanisms

A conventional parallel mechanism follows the design criteria of having a single moving platform attached to a single base platform by two or more kinematic chains or limbs. Several types of conventional parallel mechanisms are given below.

1.2.1.1 Gough-Stewart platform

During the late 1950s, V. E. Gough was researching methods of improving automotive stability mainly by looking at tyres which lead to him proposing a universal tyre testing machine which consisted of a platform (figure 1.3) capable of rotating about three axes to test tyre wear on different inclined surfaces, i.e. going uphill, downhill or round a corner [1.4].

Although the Gough platform was operational by the mid-1950s the platform is more recognisable as the Stewart platform due to a six DOF parallel robot platform that was proposed by D. Stewart in 1965 [1.10]. Stewart developed his platform to be used as an aircraft simulator due to its ability to manoeuvre like an aircraft in flight, being able to role, pitch and yaw about the x-, y- and z-axes and simulate turbulence by rapid oscillating motion along the vertical axis. This was later adapted to become a robot wrist and a tendon actuated parallel robot [1.7] [1.10].

The Stewart platform was made up of a moving platform at the top of the manipulator connected to its base, usually in the form of feet or a plate, secured to the ground, using six limbs each with a linear actuator, also known as a prismatic (P) joint, part way up each limb (figure 1.4) [1.10][1.11].

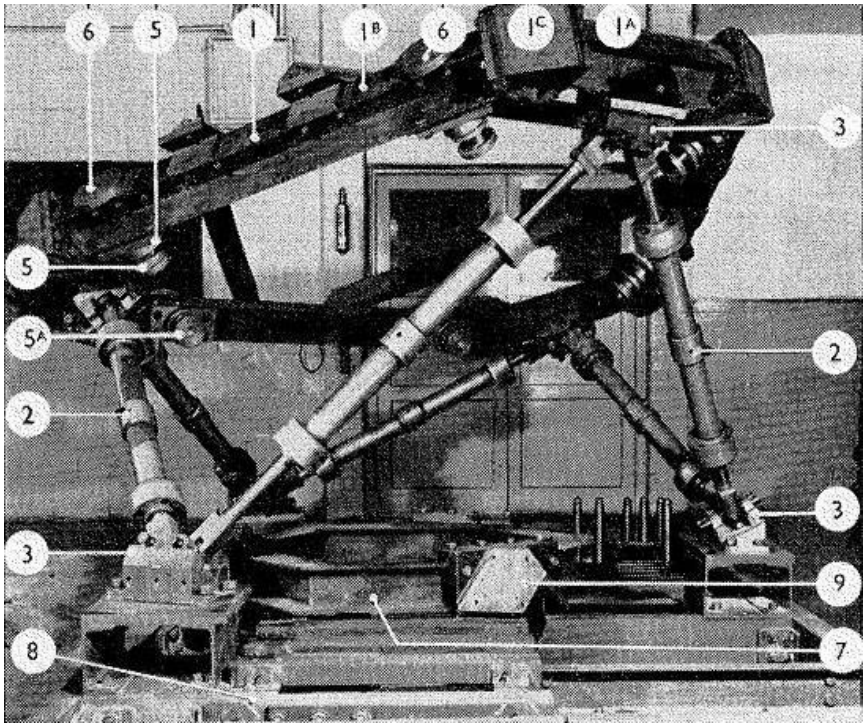

Figure 1.3: Photograph of the Gough platform [1.4].

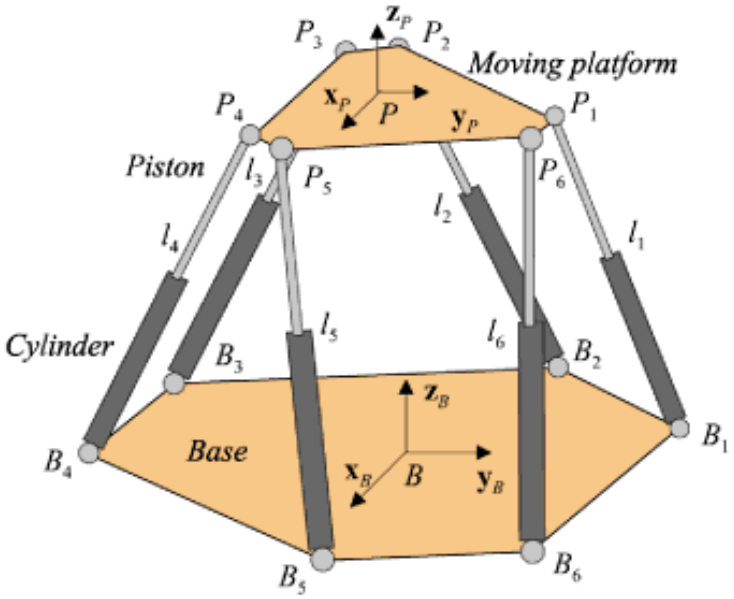

Figure 1.4: Stewart Platform [1.12]

The limbs are attached to the base and moving platform by either spherical (S) or universal (U) joints which are limited to a certain rotation on each axis to maintain rigidity [1.11]. The method of moving the plate revolves around extending one or more of the P joints while contracting others to allow the plate to rotate or translate about the x-, y- and z-axes. This design did mean that the length of an individual limb could not be modified by its P joint without affecting the others, which allowed for a smoother motion during operation [1.11].

1.2.1.2 Delta family architecture

In 1988, Clavel proposed a parallel manipulator, which he called the Delta robot, that would make use of three spatial translational DOFs [1.3] [1.6]. This robot became one of the most recognisable parallel manipulators, moving from theoretical studies to being used in industry for high speed handling applications such as pick-and-place operation and delicate procedures during surgery [1.6][1.13].

The Delta architecture designs are based around a set of kinematic chains/limbs extending from a base platform at either the top or bottom of a robot to an end-effector [1.1]. These chains are usually grouped in sets of three (Delta robot), four (H4 and I4) or six (Hexa robot) and allow the robot to manoeuvre the end-effector/payload combination along the x-, y- and z-axes making a fully parallel 3-DOF robot such as shown in figure 1.5 [1.14].

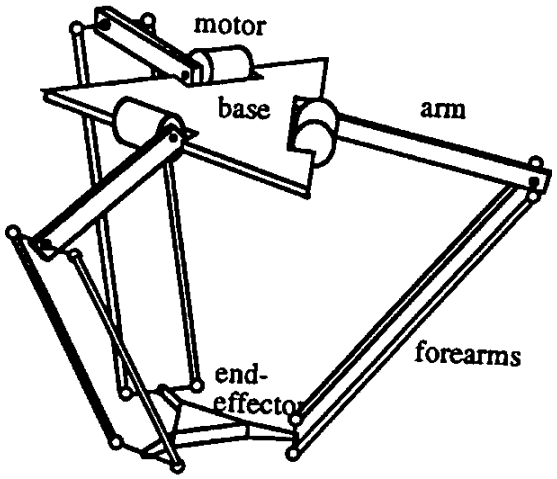

Figure 1.5: Delta robot [1.15]

Typically the individual links are made up of either an actuated R joint attached to a non-actuated U, U, R ($\bar{R}UUR$) configuration limb (figure 1.6a) where $\bar{R}$ is an actuated R joint; or an actuated R joint attached to a non-actuated double SS link finishing with an R joint ($\bar{R}(SS)^2R$) configuration as shown in figure 1.6b [1.14][1.16]. The latter configuration will be referred to as a "Delta limb" for the remainder of this thesis.

The middle rods of the Delta limb, the links between the S joints, keep a common plane meaning that the vectors joining the S joints together are equal for each individual limb on the robot. Assuming that the rods are solid objects, i.e. they cannot be twisted or bent, the dual link creates an impossible rotation of the rods about the vectors that joins the two upper/lower S joints together.

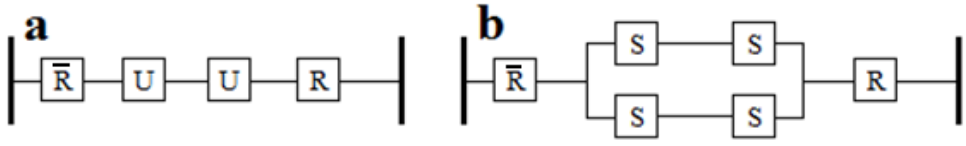

Figure 1.6: a) $\bar{R}UUR$ limb configuration; b) $\bar{R}(SS)^2R$ limb configuration (adapted from [1.16]).

The Delta robot success was limited by its limited DOF and therefore cannot be used for pick-and-place operations where the part being moved requires a certain orientation to be

picked up or put down that would require at least a single rotational DOF [1.6]. To solve the problem of the standard Delta robot not having a rotation on the payload, a fourth limb is sometimes added between the base plate and the end-effector, in a configuration similar to the initial configuration of limb ($RUPUR$) (figure 1.7). This enables the Delta robot to rotate the payload about the vertical axis allowing it to perform more complex tasks in pick-and-place operations [1.6].

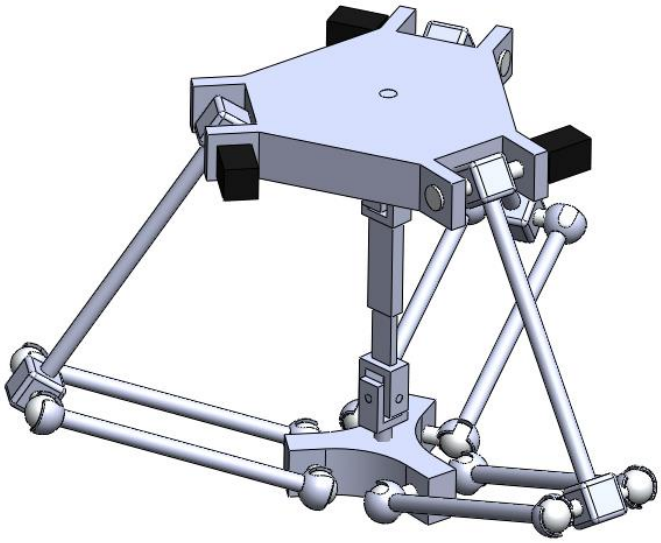

Figure 1.7: CAD model of a 4-DOF Delta robot

1.2.1.3 Hexa robot family

In the early 1990s a new robot was proposed by Pierrot under the name Hexa robot due to its 6-DOF (three translational and three rotational). Pierrot used the Delta robot architecture as a basis on which to expand and include the much-needed rotation required of the robot [1.6].

The focus of 6-DOF robots was to outperform humans and serial robots in industrial tasks such as laser cutting and insertion of objects into slots at odd angles. In order to produce the required results the proposed robots needed to be capable of fast motion and be able to utilise all 6-DOF. From these requirements, three main methods were produced [1.2].

The first was to attach a 3-DOF (all rotation) wrist to the base of a 3-DOF (all translational) Delta robot that would allow for the fast translation attributed to a Delta robot and to provide the needed DOF required for more complex tasks. However, this idea proved unsuitable, as the moving mass of the robot would have to be very large on the wrist and, due to the need of having power cables attached to the Delta robot's actuated joints, the range of allowable motion would also have been compromised [1.2].

The second method was to take the original Delta robot and add an extra-actuated R joint at the top of each chain with its axis of rotation being perpendicular to that of the other actuated R joint, i.e. $\bar{R}_x\bar{R}_z(SS)^2R$. A major drawback to this method was that the first actuated R joint would have to carry the load of the second, which in turn greatly increases the torque required for rapid motion [1.2].

The final method of creating a new 6-DOF parallel robot was the Hexa robot (figure 1.8) which took the layout of the Delta robot and replaced each of the $\bar{R}(SS)^2R$ limbs with two $\bar{R}US$ limbs. The R joint of the paired limbs are able to rotate in either the same direction as each other producing a translation of the moving platform or the pair can rotate in the opposite direction to each other resulting in a rotating motion. This meant that the Hexa robot was capable of operating in 6-DOF but in order to function properly it had two operating modes resulting in it being unable to perform both rotation and translation at the same time [1.2].

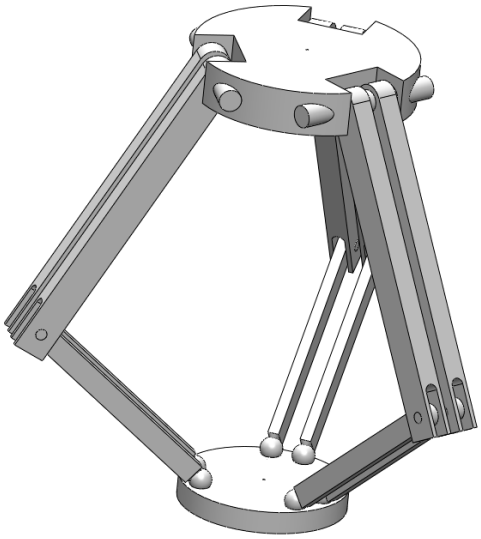

Figure 1.8: Hexa robot (adapted from [1.17])

Unfortunately, due to the general complexity of the robot caused by it requiring six motors, one for each limb, the inverse kinematic model (IKM) was overly complex causing large computational times for trajectory planning [1.6]. In addition to this fault, the cutting process for which the Hexa robot was designed only requires five controlled axes plus the rotation of the spindle [1.18] [1.6]. The Hexa robot also suffers from reduced mobility due to its small tilting angles and the incredibly high cost to manufacture each individual robot. All of these problems have resulted in the robot failing to reach the same levels of implementation and popularity in industry as the Delta robot [1.6].

1.2.1.4 Manta and Kanuk robots

The linear Manta mechanism (figure 1.9) is a 4-DOF parallel robot (3 translations and 1 rotation in the form of the moving platform tipping sideways) [1.19] and is made up of three main kinematic chains, the outer two limbs being made of two components: the actuator axis and the arm, which is a Delta limb of fixed length. For these kinematic chains, the Delta limbs are used to block two potential rotations about the vertical axis and the axis of the P joints [1.19] [1.20]. The central arm transmits a rotational torque through a single bar linked to two 2-DOF U joints and enables the tilting of the moving platform.

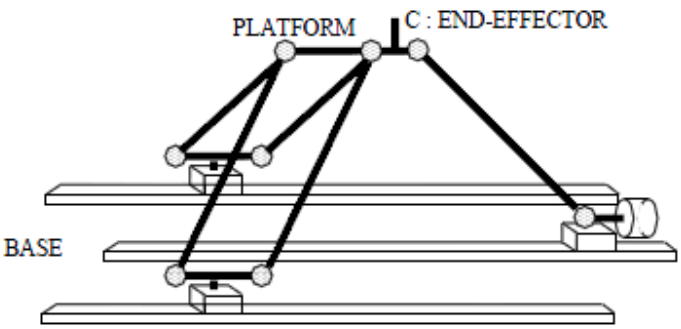

Figure 1.9: Linear Manta robot principle. [1.20]

The Kanuk concept is closely related to the Manta. It constitutes four main kinematics chains that are also made of the same two components: the actuator axis and the fixed length arm (figure 1.10). The Kanuk linear version is powered by linear actuators such as the Manta. To make the construction of the robot as simple as possible, two *P* joints are mounted to each rail, which can be made as long as required.

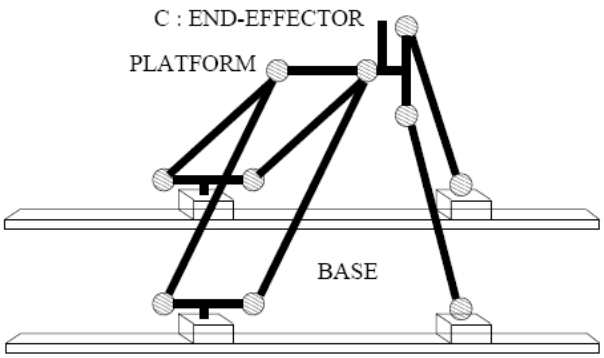

Figure 1.10: Linear Kanuk robot [1.20]

1.2.1.5 Orthoglide mechanism

In 2000, a new parallel mechanism architecture was developed by Philippe Wenger and Damien Chablat in order to produce a high speed machine tool parallel mechanism that included a regular workspace, homogenous performance and a stable dynamic performance while being capable of operating through an all translational 3-DOF motion pattern [1.21-1.23]. The parallel mechanism is made of three kinematic chains, each constructed in a $PRPaR$ format (figure 1.11), where the P joint is the actuated joint of the mechanism, and Pa represents a parallelogram joint that acts similarly to the parallelogram in the limb of a Delta robot, named the "Orthoglide" mechanism.

The Orthoglide mechanism is particularly impressive in design as it is both without singular configuration within its viable workspace [1.24] [1.25] and is completely self-collision free allowing the mechanism to be operated with no risk of destabilisation [1.24] [1.26].

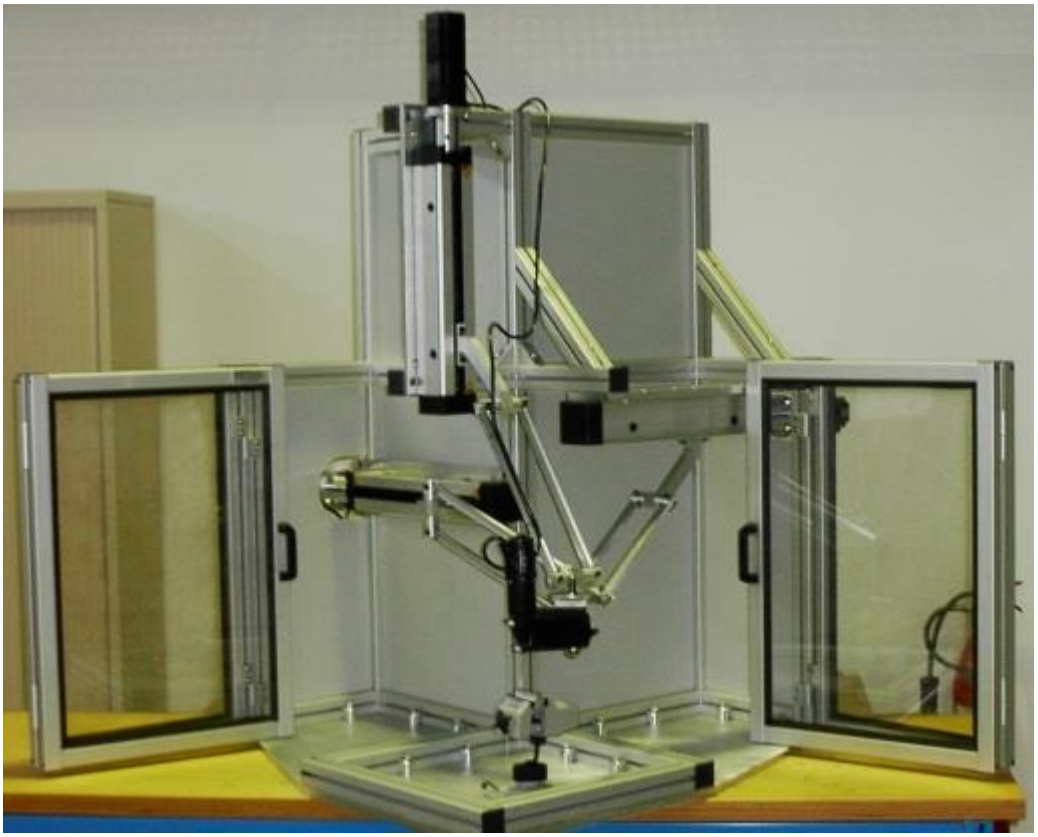

Figure 1.11: Orthoglide mechanism [1.24]

1.2.1.6 3-DOF 3-CRR translational parallel mechanism

In [1.27] a 3-CRR all translational 3-DOF parallel manipulator is investigated for its inverse kinematic model and singularities. This mechanism is constructed out of three kinematic chains in the form of a cylindrical (C) joint and two R joints set perpendicular to the axis of the C joint (figure 1.12). The unique characteristics of this mechanism is that it has a set of linear input-output equations.

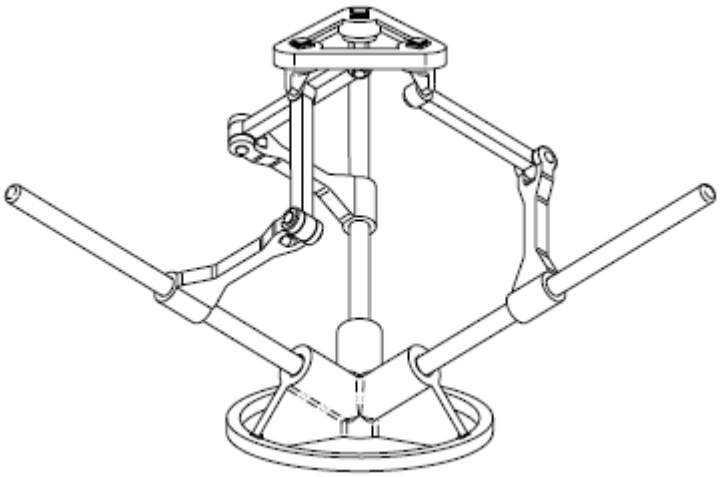

Figure 1.12: 3-CRR 3-DOF all translational parallel mechanism [1.27].

The moving platform is unable to move when all actuated joints are locked in place. This prevents any uncertainty in the position of the moving platform for the controller also known as "uncertainty singularities". The mechanism is also mostly singularity free with

the only possible singularities occurring when the axes of all the C and R joints with a leg are coplanar. This configuration only occurs when the mechanism is at the boundary of its workspace, and therefore easily avoided.

1.2.1.7 Quadrupteron mechanism

The Quadrupteron mechanism is a four-limbed mechanism connecting to a singular point. The kinematic chains of the mechanism form a partially decoupled 1-CRR-3-$CRRR$ 4-DOF parallel mechanism (figure 1.13) that produces three translational and one rotational DOF Schöenflies (SCARA) motions [1.28].

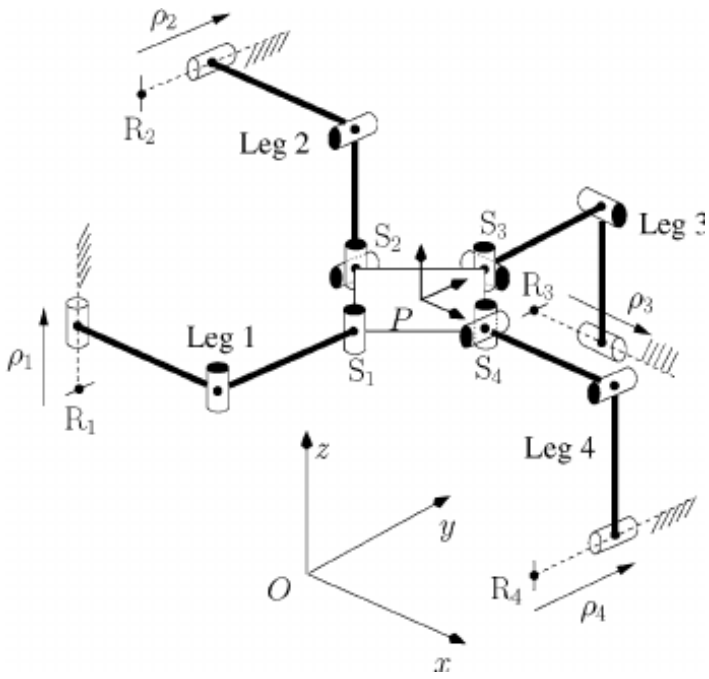

Figure 1.13: Quadrupteron 4-DOF parallel mechanism kinematic chain layout [1.29].

The partially decoupled state of the parallel mechanism simplified the inverse kinematics and workspace analysis resulting in a more efficient control system, enabling the mechanism to operate at higher speeds than a fully coupled parallel mechanism.

1.2.2 Generalised parallel mechanism

In the development of both parallel and serial mechanisms, the process is the same, beginning with a simple architecture and then developing to become more complex. In Xianwen Kong's MSc dissertation (in Chinese) published in 1990 [1.30], a series of mechanisms are shown to illustrate this process. With the initial mechanism being a simple serial mechanism that extends from a base to an end-effector, this mechanism is then developed to position the end-effector halfway along and attaching both ends to the base, thus turning it into a parallel mechanism. The mechanism is then developed to a more complicated mechanism by adding an additional kinematic chain. The next two stages of this development incorporate additional kinematic chains and moving platforms in order to produce a generalised approach to the mechanism.

An example of a generalised mechanism is displayed in [1.31] where a mechanism is given a tree branch structure replacing the rigid design structure by having a base and moving platform connected by one or more kinematic chains (figure 1.14).

Another example of a generalised mechanism is displayed in [1.32], in which the parallel mechanism is designed so as not be considered as a single rigid link between the base of the mechanism and its moving platform. This mechanism shown in figure 1.15 was inspired by the walking motion of a six legged insect in which the body of the insect is capable of rearranging itself slightly to allow for more fluidic motion when walking.

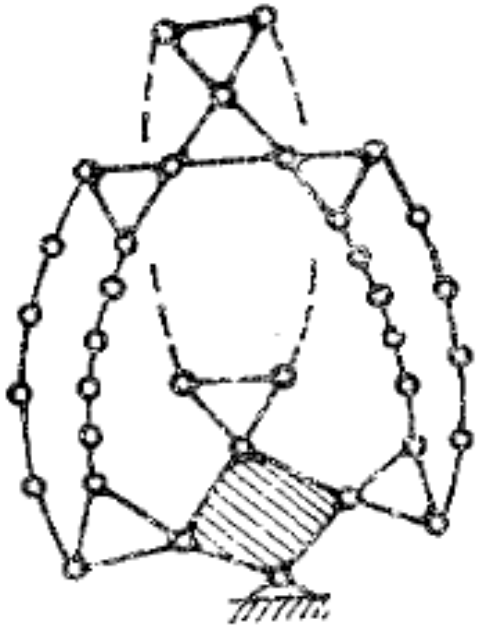

Figure 1.14: 13-DOF Parallel mechanism with tree like kinematic chains, a base, and moving platform [1.31].

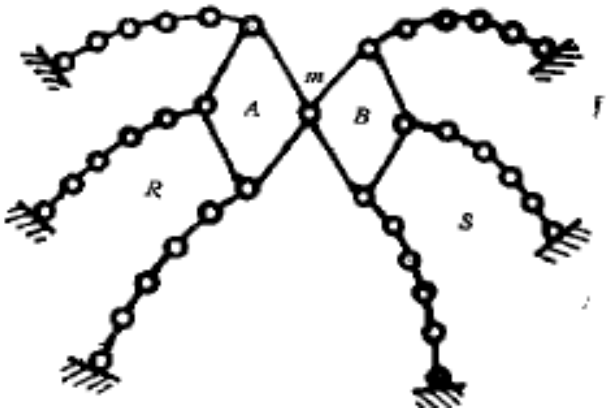

Figure 1.15: Parallel mechanism inspired by the walking gait of a 6-legged animal or insect [1.32].

The last generalised parallel mechanism to be detailed in this section looks at a parallel mechanism with a reconfigurable moving platform [1.33]. This mechanism is designed to allow the kinematic chains of the mechanism to reposition the two end-effectors in order to achieve a given task. This type of reconfigurable parallel mechanism can be applied in numerous industries including search and rescue, pick-and-place operations and the oil industry where the mechanism manoeuvres through pipes that vary in size.

1.2.2.1 H4 robot family

For most applications robots with 3-DOF, such as the Delta robot, have too few DOF and robots with 6, like the Hexa robot, have too many and are too complex for pick-and-place applications, whereas 4-DOF (3 translational, 1 rotational) is generally accepted as the minimum and preferred number of DOF to produce a high speed pick-and-place motion [1.18] [1.34]. Similar to the Delta robot design, the H4 design is based around a set of four links stemming from a either a shared base plate or individual runners located along the framework of the robot [1.18] [1.34]. The four chains are typically made in the form of an actuated R or P joint followed by a dual link of US joints that are then coupled to one of the other chains via the moving platform where another R joint connects the two chains at the end-effector to form the full robot [1.34] as shown in figure 1.16.

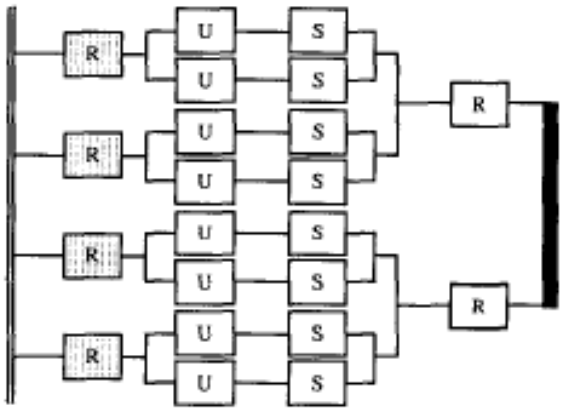

Figure 1.16: H4 joint map [1.34]

In 2003, Pierrot proposed a new form of the H4 robot also called "H4" as the term H4 was more the name of the architecture rather than the name of a single robot, which robot was to have an overhaul to its trajectory-tracking accuracy by using a dynamic control system originally proposed by Pierrot with the original H4 concept [1.6]

The basis of the new H4 design was to have the U joints replaced by S joints but still using the same concept for the rest of the limbs [1.35] [1.36] and the moving platform by having it made from two lateral bars and one centre bar connected by two R joints, one R joint operates a large gear which helps rotate the centre section of the moving platform allowing the end-effector to rotate about the vertical axis [1.36] [1.37].

1.2.2.2 I4 robot

The I4 robot was conceived to investigate the limitations of the H4 robots moving platform as the RRR joint could become problematic as it sometimes caused the two sections of the moving platform to collide with each other. The I4 moving platform was designed to see if the problems could be averted by modifying the plate to be made from two lateral parts connected to a central section by two P joints [1.38] as shown in figure 1.17.

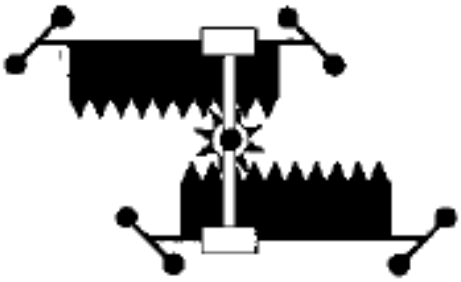

Figure 1.17: I4 moving platform [1.38]

The I4 architecture is based around the Delta/H4 architecture by having the actuated joints attached to the base of the robot in order to reduce the moving part mass of the robot. These joints are then attached to the same Delta limb model of two rods in parallel connected at each end by S joints (figure 1.6b). The I4 robot design proposed in [1.38] is constructed using actuated P joints which operate along two runners in parallel with each other; each joint is then connected to a Delta limb and then onto one of the end points at the moving platform as shown in figure 1.18a; the joint map for the I4 robot is shown in figure 1.18b.

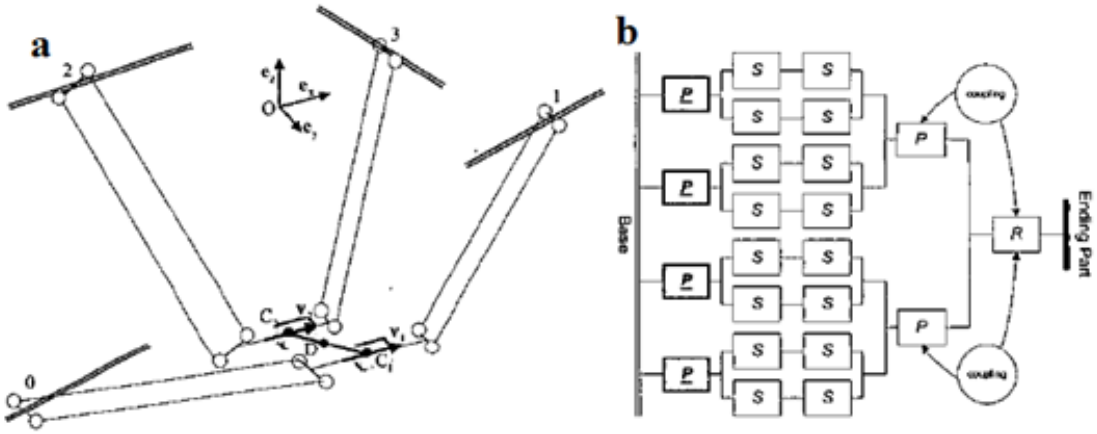

Figure 1.18: a) I4 robot build configuration [1.38]; b) I4 Joint map [1.38].

1.2.2.3 Par4 robot

The Par4 robot is the latest member of the H4 architecture family. The Par4 robot is based upon the original H4 robot but is similar to the I4 robot; it has a modified moving platform and it has a unique modified base on which the axes of the rotation joints are at $90°$ to the next R joint (figure 1.19).

Figure 1.19: Par4 robot [1.39]

The modified moving platform is made up of two main sections where the two of the robot's limbs attach to the moving platform via the S joints; these main sections are then connected by two connecting rods that are connected to the main part in between the individual limb's S joints (figure 1.20). This set up increases the stability of the moving platform's rotation about the y-axis however; it also limits the rotation of the end-effector to $\pm 45°$. The moving platform makes use of an amplification system in the form of either gears or a pulley/belt system to add an extra $\pm 180°$ to the y-axis rotation of the end-effector [1.39]. This design for the moving platform eliminates the problem, usually found in 3-DOF Delta robots, that requires a telescopic limb to add a 4th DOF to the robot (see section 1.2.1.2).

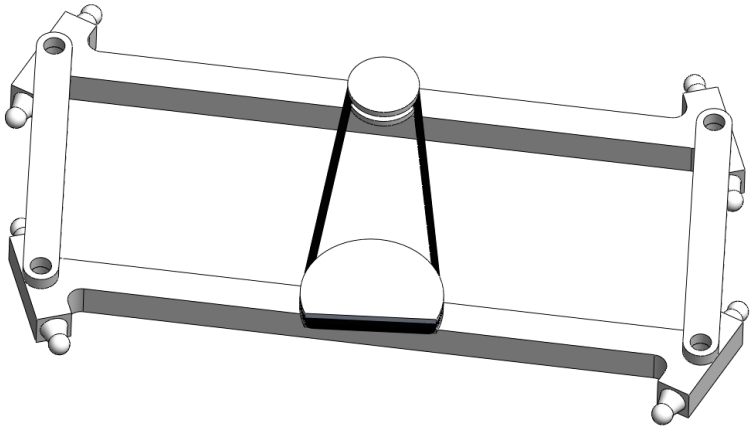

Figure 1.20: Moving platform design of the Par4 robot (adapted from [1.39])

The Par4 robot has been designed to respond to the shortcomings of both the H4 and I4 robots, namely the I4's P joint being worn out during high acceleration motion and the H4's actuated joint locations causing numerous internal singularity configurations.

1.2.3 Reconfigurable parallel mechanisms

A reconfigurable parallel mechanism is defined as a mechanism that can change the configuration or operation of one or more of its kinematic chains or moving platform in

order to switch between operation modes. This section will therefore give details on this type of mechanism.

1.2.3.1 DIRECTOR: disassembly-free reconfigurable parallel manipulator

The DIRECTOR mechanism developed at Heriot-Watt University in 2012 is a 3-DOF (two translational, one rotational) multi-mode parallel mechanism, consisting of two PRU kinematic chains and one PUU kinematic chain [1.40]. The limbs of the mechanism are arranged in a triangular format in which the two PRU chains are located to the sides of the front face of the mechanism and the PUU chain is located in the middle of the back face (figure 1.21).

The mechanism has lockable joints, which change the rotational DOF from parallel to the plane of motion to perpendicular. This kind of mechanism design allows the user to perform multiple tasks with a single mechanism without having to pass one or more kinematic chains through singular configurations in order to switch between operation modes [1.40].

1.2.3.2 3-5R Parallel mechanism

Another reconfigurable mechanism is the 3-$RRRRR$ (3-5R) parallel mechanism in which the links that connect the last R joint of each limb is connected to each other via another R joint (figure 1.22).

This moving platform design is often referred to as a Bricard-linkage [1.41] and, due to the arrangement of the six R joints across the moving platform, the mechanism is capable of being arranged into key orientations that allow for several operation modes to be realised (see Fig. 2 [1.41]). This is due to the arrangement of each of the limb's R joints in which the first R joint of each chain is aligned to meet at a single point; the fifth joint in each chain is coaxial with the axis of one of the other moving platform R joints. In order to switch between operation modes, one of the R joints on the Bricard-linkage is locked when the mechanism is in a certain configuration. This causes it to operate in one of its 3-DOF operational modes: a 3-DOF spatial translation motion, a spherical motion, zero torsion motion and a planar motion.

Figure 1.21: DIRECTOR mechanism (section view of mechanism to display kinematic limbs).

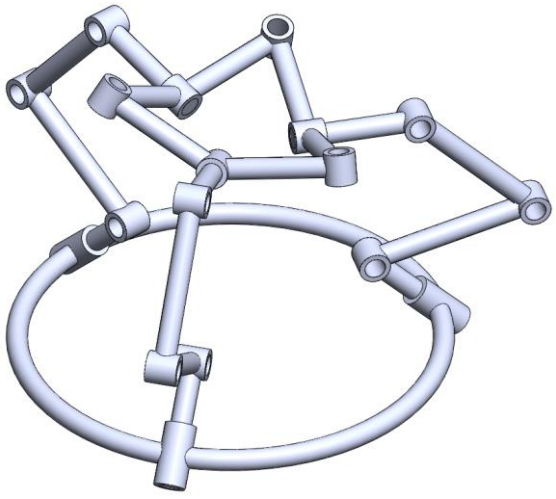

Figure 1.22: 3-5R parallel mechanism.

1.3 Screw Theory

Screw theory looks at joint and link motion and constraints in the form of two separate types: twists and wrenches. A spatial displacement of a rigid body by means of a rotation and translation about a line (along the link) or axis is also known as "Screw Displacement" [1.42].

The spatial displacement is made up of six parameters: three Euler angles, which define the rotation about the x-, y- and z-axes, and a three dimensional translational vector along the x, y and z axes (figure 1.23).

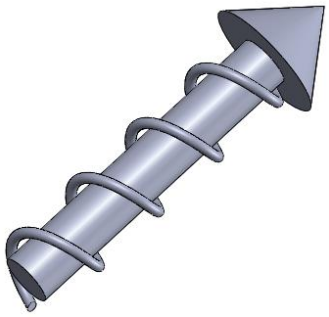

Figure 1.23: CAD model representation of a Screw.

A screw is typically represented by two 3-dimensional vectors [1.43]:

$$\$ = \begin{bmatrix} \hat{s} \\ \vec{s}_O \end{bmatrix} \tag{1.2}$$

where $\hat{s}$ is the unit vector about the axis and $\vec{s}_O$ is the moment about a fixed point O when $\hat{s}$ acts upon the rigid body.

The moment about the fixed point is defined by the equation [1.42] [1.44]:

$$\vec{s}_O = \hat{s} \times \vec{r}_O + h\hat{s} \tag{1.3}$$

where h is the pitch of the screw, $\vec{r}_O$ is the vector of the screw motion and $\times$ is the cross product.

If the pitch of the screw is infinite then the screw is reduced to:

$$\$ = \begin{bmatrix} \hat{s}/h \\ (\hat{s} \times \vec{r}_O)/h + \hat{s} \end{bmatrix} \tag{1.4}$$

$$\$ = \begin{bmatrix} \hat{s}/\infty \\ (\hat{s} \times \vec{r}_O)/\infty + \hat{s} \end{bmatrix} \tag{1.5}$$

And becomes a prismatic pair of vectors and mimics a P joint [1.43-1.46]:

$$\$ = \begin{bmatrix} 0 \\ \hat{s} \end{bmatrix} \tag{1.6}$$

Likewise if the pitch of the screw is equal to zero the screw is reduced to a rotation pair and can be defined as pure rotation [1.43-1.45]:

$$\$ = \begin{bmatrix} \hat{s} \\ \hat{s} \times \vec{r}_O + 0 \times \hat{s} \end{bmatrix} \tag{1.7}$$

$$\$ = \begin{bmatrix} \hat{s} \\ \hat{s} \times \vec{r}_O \end{bmatrix} \tag{1.8}$$

When investigating multiple screws, Lie algebra or motor algebra is used as it is sometimes known to calculate the arithmetic of the screws as follows [1.44] [1.47]:

- Addition

$$\$_1 + \$_2 = \begin{bmatrix} \hat{s}_1 + \hat{s}_2 \\ \vec{s}_{O_1} + \vec{s}_{O_2} \end{bmatrix} \tag{1.9}$$

- Multiplication by scalar

$$\varepsilon\$ = \begin{bmatrix} \varepsilon\hat{s} \\ \varepsilon\vec{s}_O \end{bmatrix} \tag{1.10}$$

- Dual motor product

$$[\$_1 \quad \$_2] = \begin{bmatrix} \hat{s}_1 \times \hat{s}_2 \\ \hat{s}_1 \times \vec{s}_{O_2} - \hat{s}_2 \times \vec{s}_{O_1} \end{bmatrix} \tag{1.11}$$

In instantaneous kinematics and statics of a parallel robot, the spatial motion of a rigid body is a screw motion about a line called the "screw axis". The rotation and translation of the screw motion is coupled by the scalar quantity p (the twist pitch) [1.48]. The screw in this form of kinematics is referred to as a "twist" and is represented by a pair of 3-dimensional vectors:

- The linear velocity (V)
- The angular velocity (Ω)

The twist can be written as:

$$\hat{T} = (\Omega|V) \tag{1.12}$$

where the linear and angular velocities are:

$$\Omega = \omega\vec{s} \tag{1.13}$$

$$\mathrm{V} = c \times \omega\vec{s} + v\vec{s} \tag{1.14}$$

where s is the direction of the twist, c is a point on the screw axis, ω is a scalar value of the magnitude of the twists angular velocity about the screw axis and v is the magnitude of the twists partial linear velocity along the screw axis.

The pitch of a twist is defined as the ratio of the linear and angular velocity and, as stated above, two special cases can be found in which a pure rotation or translation can be found when the twist pitch, ε, is either zero or infinite respectively [1.48].

In the static analysis of a parallel robot, a wrench consists of two vectors representing the moment M acting on a rigid body and the force F being applied to the body. The force applied to the rigid body is calculated using the equation [1.48]:

$$F = f\hat{u} \tag{1.15}$$

where f is a scalar magnitude of the force being applied along the screw axis and $\hat{u}$ is the direction of the wrench axis

The moment acting on the rigid body can likewise be calculated by:

$$M = r \times f\hat{u} + m\hat{u} \tag{1.16}$$

where r is a point on the wrench axis and m is the magnitude of the partial moment acting upon the screw axis.

The wrench can then be written as:

$$\widehat{W} = (F|M) = (f\hat{u} \mid r \times f\hat{u} + m\hat{u}) \tag{1.17}$$

Just as with the screw and twist, the wrench has the same two special cases where the wrench pitch, is represented as:

$$q = \frac{f}{m} \tag{1.18}$$

Therefore it can be either a pure force or a moment when q is equal to either zero or infinite respectively [1.48] [1.49].

Screw theory is used typically for the study of instantaneous motion in robotics. The principle of reciprocal screws in screw theory is the interaction between two screws on the same mechanism. Two screws are reciprocal to each other if they meet the following conditions [1.46] [1.49] [1.50]:

$$\begin{cases} \text{No conditions} & \text{if both screws pitches are infinite} \\ \cos\lambda = 0 & \text{if } \mathrm{h}_1 \text{ or } \mathrm{h}_2 \text{ is infinite} \\ (\mathrm{h}_1 + \mathrm{h}_2)\cos\lambda - \mathrm{r}_{12}\sin\lambda = 0 & \text{if } \mathrm{h}_1 \text{ and } \mathrm{h}_2 \text{ are finite} \end{cases}$$

where λ is the angle between the screw axes, r_{12} is the distance between the screw axes and h_i is the pitch of the i^{th} screw.

From this, we can see that:

- two screws of infinite pitch, i.e. pure translation, are always reciprocal to each other;
- if one screw has an infinite pitch and another has either a finite or zero pitch, i.e. pure rotation, they are only reciprocal if they are perpendicular to each other;
- if both screws have finite or zero pitch they are only reciprocal if they are coplanar, i.e. they are on the same plane.

For an n-system of screws where n is the total number of screws in the system $(0 \leq n \leq 6)$, all of the screws which are reciprocal to the original screw system form their own $(6-n)$-system of screws which can be obtained using the reciprocity conditions listed above [1.46].

The interaction of screws in a system has been shown to aid in the identification of singular configurations that cause static singularities [1.51]. This is done by investigating the planar pencil forces produced by the limbs of the mechanism, and how their reciprocal screw systems interact with each other concerning the overall order of the system; this is covered in detail in [1.50].

1.4 Improvements in CAD Design

Since the invention of 3D CAD modelling software in the 1970s designers have been increasing the rate and accuracy of their designs as the technology progressed from a program developed by Mathematics Application Group, Inc. for 3D nuclear radiation exposure analysis called “SynthaVision” [1.52] through the use of CAD programs which created 3D models from 2D drawings such as “I-DEAS” to modern packages such as “SolidWorks” and "Creo” (previously known as "Pro Engineer"). These modern programs use solid body extrusions to produce realistic, life like parts in a virtual environment so that designers can run various tests and simulations such as finite element analysis, motion studies and determine if new parts will fit into existing assemblies without the need to build a real world prototype which can be costly if the part does not behave in the manner expected or is not capable of being attached to a physical assembly. With the development of CAD software progressing in this manner, it has become increasingly easier to develop new architectures for parallel and serial mechanisms without the need of complex computation as the virtual mechanism should give the designer a realistic idea as to how the mechanism will move and interact within its environment and with itself.

1.5 Graphical Programming Software

Graphical programming software uses specially designed blocks on a graphical workspace to produce a computer code based on the type of blocks used and the method in which they are connected. Each block in this type of programming software adds to the program a background subroutine code that is compiled by the computer when the operation is run. The blocks are given inputs and output variables by the code, which can be wired up to other blocks in order to create the main program. Several examples of this type of programming software exist, such as the Matlab add-on software Simulink [1.53], the new Maplesoft software MapleSim [1.54] and National Instruments control software Labview [1.55].

The types of programming blocks vary between software but the main concept behind them remains the same i.e. connect blocks to form a complete code. In the Simulink software the type of connection between all blocks is determined by the software and is represented by using a simple grey line (figure 1.24).

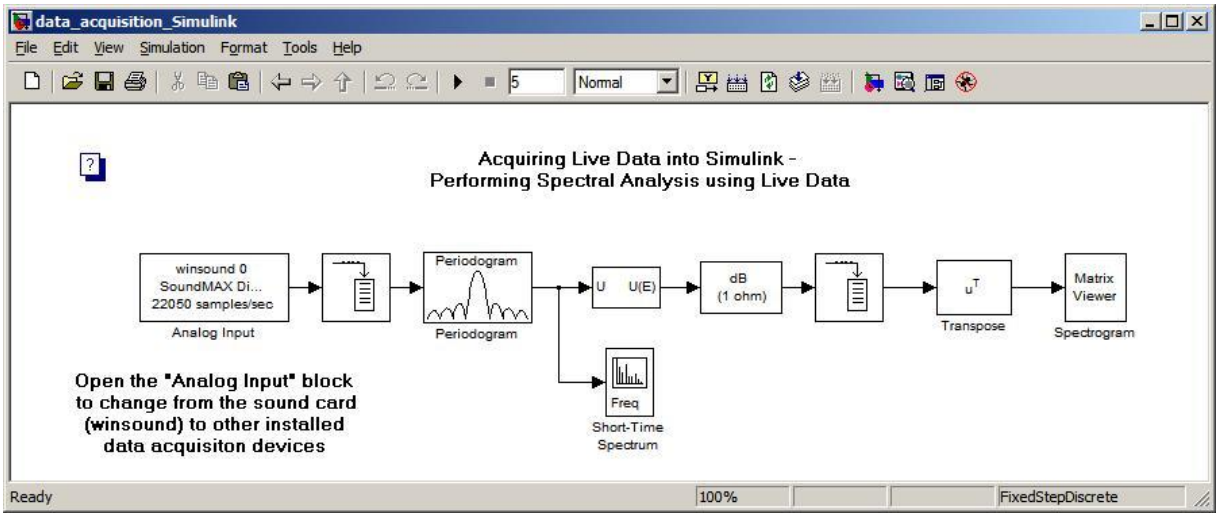

Figure 1.24: Example of Simulink code [1.56].

However, in MapleSim and Labview software, the types of connections between blocks are based on the type of output that is expected by the input/output ports of the different blocks. In the MapleSim software, the input/output port types are colour coded in order to inform the user as to what type of block and connection is expected in order to wire up the block to the rest of the software. The types of connections are listed below [1.57] and an example of these is shown in figure 1.25:

- grey connections for multibody and 1D rotational component connections;
- red connections for hydraulic and thermal component connections;
- blue connections for numerical data connections;
- green connections for 1D translational component connections;
- orange connections for magnetic component connections;
- purple connection for electronic component connections.

The Labview software in comparison uses the colour of its connections in order to display the type of data being transferred between blocks. The types of connections are detailed below with an example code displayed in figure 1.26:

- blue connections indicate integer data transfer;
- orange connections indicate float data transfer;
- pink connections indicate string data transfer;
- green connections indicate Boolean data transfer;
- yellow connections indicate error data transfer.

with thickset connecting wires representing array data of the type indicated by the connection's colour.

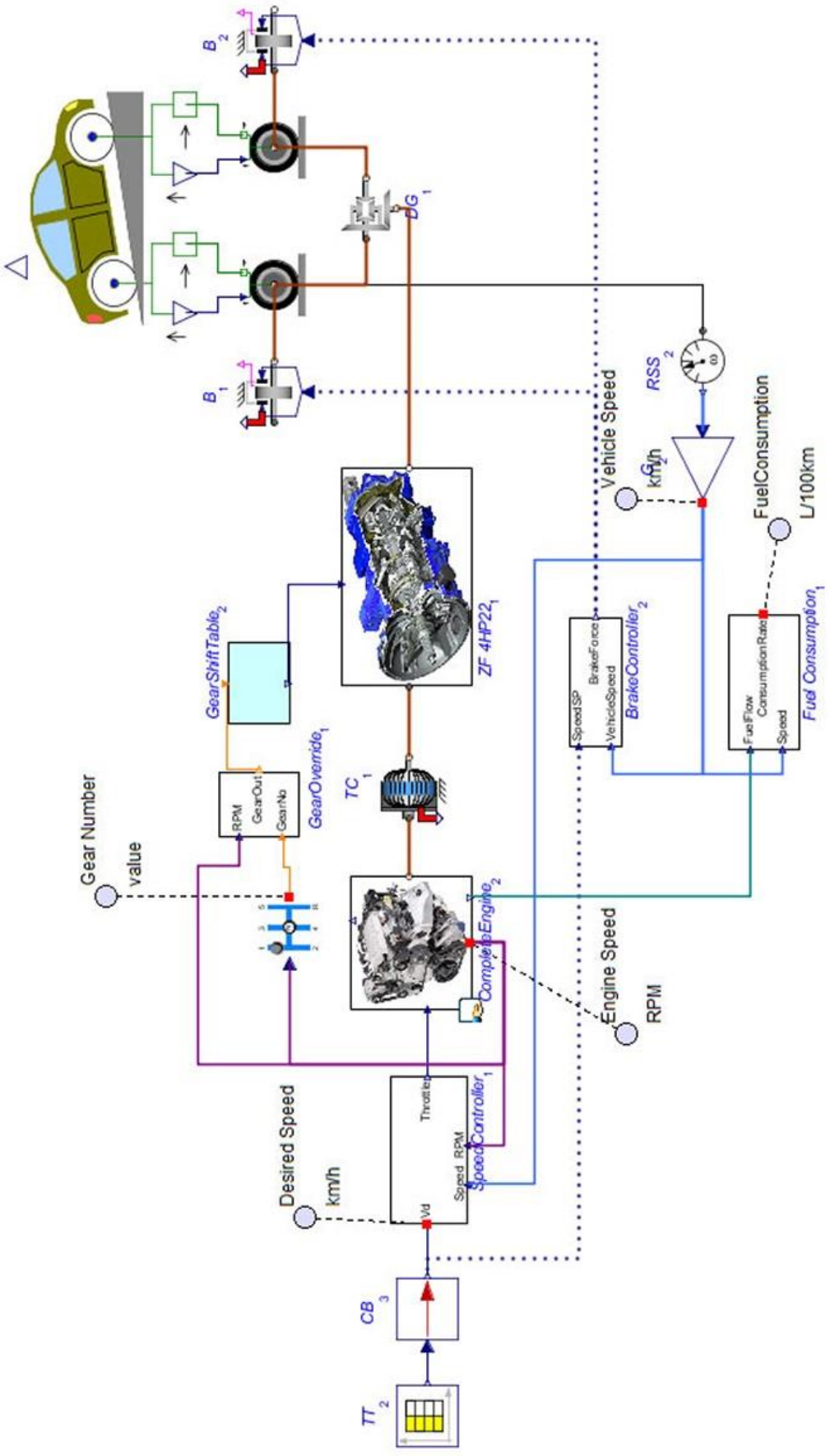

Figure 1.25: Example of MapleSim code [1.58].

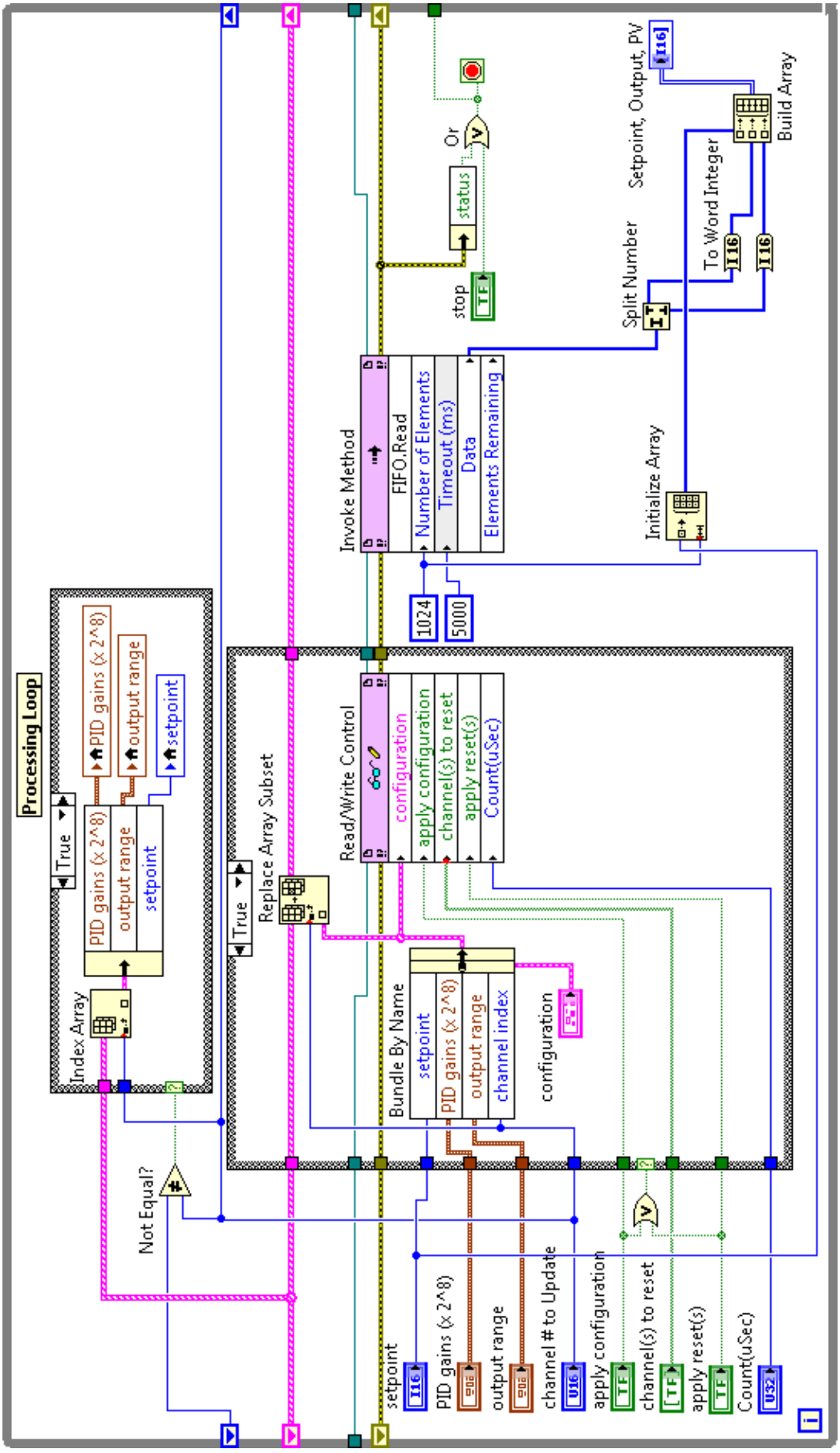

Figure 1.26: Example of Labview code [1.59].

1.6 Aim, Research Questions and Objectives of the Research

In the scope of research being carried out on parallel mechanisms, there has been a distinct lack of research being done on parallel mechanisms with multiple moving platforms. These mechanisms have the potential to be more energy efficient in terms of work done when compared to a parallel mechanism with only one moving platform. This statement can be envisioned by the example of a person moving bags of groceries from a car to a house. In this case, the person would carry multiple bags using both hands in order to ensure the work was done faster and more efficiently, as the increase in work done during the single trip would typically be less than the work done performing the procedure multiple times.

The aim of this study is to modify and enhance key stages of the current design process for such devices so that it will support the rapid development of energy efficient, cost effective parallel mechanisms. To achieve this goal, four research questions were defined revolving around the development of a parallel mechanism using the current process and how this can either be modified or improved upon. These were as follows:

1. Can current type synthesis methods be modified to incorporate the design of multi-platform parallel mechanisms using multiple virtual chains?
2. Can virtual chains support a more intuitive graphical-based approach to the conceptualisation, evaluation and definition of the workspace, inverse kinematic and dynamic analyses for parallel mechanisms?
3. Can generic cost and energy efficiency models be developed for parallel mechanisms?
4. How can a modified design process be evaluated and rigorously tested to address the proposed research aim?

In order to answer these questions, the research objectives of this thesis are as follows:

1. To investigate the design process of parallel mechanisms by systematically investigating a class of parallel mechanisms with multiple platforms in order to identify any potential improvements to the design process.
2. To define the functionality of a parallel mechanism by applying virtual chains to analyse the mechanism's workspace in CAD to produce a more visual representation of its limitations.
3. To determine whether the constraints of the workspace of the parallel mechanism can be used with virtual chains to produce an inverse kinematic model subsequently used to: (i) develop an inverse dynamic model; (ii) apply it within the control system.
4. To determine if it is possible to produce a method for evaluating the energy efficiency and cost of a parallel mechanism.
5. To formalise the modification of the design process for parallel mechanisms and to test each modification through a series of comparative case studies and design a more complex mechanism through the proposed processes.

1.7 Outline of the Thesis

In the second chapter of this thesis, the current method of type synthesis is reviewed with the current process involving the concept of virtual chains being used to represent the motion pattern of the mechanism being investigated. This method is then extended in Chapter 3 to incorporate the use of type synthesis with a class of multiple platform mechanisms before being used to produce several concept designs of multi-platform mechanisms. In Chapter 4 the concept of virtual chain will be extended upon to control mechanisms in a 3D CAD environment in order to produce the workspaces of several mechanisms. In Chapter 5 the virtual chains concept is extended further still into controlling a mechanism in the mathematical modelling software MapleSim, which will be used to produce an inverse kinematic model of several mechanisms as well as producing a dynamic analysis of a mechanism. Chapter 6 looks at the control element of robotics and how the inverse kinematic model produced in MapleSim can be utilised to control a parallel mechanism through a Labview VI. Chapter 7 investigates producing a working initial prototype of a mechanism designed and controlled using the concepts laid out in the previous chapters. Finally, in Chapter 8 the conclusions of the work will be presented along with the contributions of the work and a plan for further work to be done on the methodologies presented in this thesis.

References

[1.1] K. Miller, "Maximization of Workspace Volume of 3-DOF Spatial Parallel Manipulators", Journal of Mechanical Design, vol. 124, pp 347 – 357, 2002.

[1.2] F. Pierrot, P. Dauchez and A. Fournier, "HEXA: A Fast Six-DOF Fully-Parallel Robot", Advanced Robotics, vol. 2, pp 1158 – 1163, 1991.

[1.3] X.J. Liu, J. Wang, K.K. Oh and J. Kim, "A New Approach to the Design of a DELTA Robot with a Desired Workspace", Journal of Intelligent and Robotic Systems, vol. 39, pp 209-225, 2004.

[1.4] V.E. Gough, "Universal Tyre Test Machine", Proceedings of the Ninth International Technical Conference FISITA, pp 117 – 137, 1962.

[1.5] I.A. Bonev and C.M. Gosselin, "Analytical Determination of the Workspace of Symmetrical Spherical Parallel Mechanisms", IEEE Transactions on Robotics, vol. 22, no. 5, pp 1011 – 1017, 2006.

[1.6] H.B. Choi, O. Company, F. Pierrot, A. Konno, T. Shibukawa and M. Uchiyama, "Design and Control of a Novel 4-DOFs Parallel Robot H4", Robotics and Automation, vol. 1, pp 1185 – 1190, September 2003.

[1.7] K.-M. Lee and D.K. Shah, "Dynamic Analysis of a Three-Degrees-of-Freedom in-Parallel Actuated Manipulator", IEEE Journal of Robotics and Automation, vol. 4, no.3, pp 361 – 367, 1988.

[1.8] D. Chablat, P. Wenger, F. Majou and J.-P. Merlet, "An Interval Analysis Based Study for the Design and Comparison of 3-DOF Parallel Kinematic Machines", the International Journal of Robotics Research, vol. 23, no. 6, pp. 615 – 624, 2004.

[1.9] Y. Li and G.M. Bone, “Are Parallel Manipulators More Energy Efficient?”, Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2001.

[1.10] D. Stewart, “A Platform with Six Degrees of Freedom”, Proceedings of the IMechE, vol. 180, no. 15, pp 371 – 386, 1965-66.

[1.11] A. Korobeynikov and V. Turlapov, “Modelling and Evaluating of the Stewart Platform”, Journal of Sibirskii Zhurnal Vychislitel’noi Matematiki, vol. 9, no. 3, pp 279 – 286, 2006.

[1.12] A.M. Lopes and E.J. Solteiro Pires, “Complete Dynamic Modelling of a Stewart Platform Using the Generalized Momentum Approach”, *Nonlinear Science and Complexity*, J.A. Tenreiro Machado (Eds), Springer, pp 199 – 210, 2010.

[1.13] I.A. Bonev website, “Delta Parallel Robot – the Story of Success”, http://www.parallemic.org/Reviews/Review002p.html.

[1.14] F. Pierrot, C. Raynaud and A. Fournier, “DELTA: A Simple and Efficient Parallel Robot”, Robotica, vol. 8, pp 105 – 109, 1990.

[1.15] P. Vischer, and R.Clavel, "Kinematic Calibration of the Parallel Delta Robot", Robotica, vol. 16, pp. 207 - 218, 1998.

[1.16] 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.

[1.17] J.-P. Merlet, “Parallel Robots”, Kluwer Academic Publishers, chapter 2, p 42, 2000.

[1.18] A. Vivas and P. Poignet, “Predictive Functional Control of a Parallel Robot” Control Engineering Practice, vol. 13, pp 863 – 874, 2005.

[1.19] L. Rolland website, “Manta Parallel Robot: A 4 DOF Robot for Inexpensive Material Handling”, http://www.parallemic.org/Reviews/Review010.html.

[1.20] L.H. Rolland, “The Manta and the Kanuk Novel 4-DOF Parallel Mechanisms for Industrial Handling” International Mechanical Engineering Congress and Exposition, pp. 1 – 14, 1999.

[1.21] D. Chablat, and P. Wenger, "Architecture Optimization of a 3-DOF Translational Parallel Mechanism for Machining Applications, the Orthoglide", IEEE Transactions on Robotics and Automation, vol. 19, no. 3 pp. 403-410, 2003.

[1.22] A. Pashkevich, P. Wenger, and D. Chablat, "Design Strategies for the Geometric Synthesis of Orthoglide-type Mechanisms", Mechanism and Machine Theory, vol. 40, no. 8, pp. 907-930, 2005.

[1.23] P. Wenger, and D. Chablat, "Kinematic Analysis of a new Parallel Machine Tool: The Orthoglide", Advances in Robot Kinematics, pp. 305-314, 2000.

[1.24] D. Chablat, P. Wenger, and F. Majou website, "Orthoglide: A 3-Axis Parallel Machine Tool for High-Speed Machining", http://www.parallemic.org/Reviews/Review011.html.

[1.25] A. Pashkevich, D. Chablat, and P. Wenger, "Kinematics and Workspace Analysis of a Three-Axis Parallel Manipulator: The Orthoglide", Robotica, vol. 24, no. 1, pp. 39-49, 2005.

[1.26] F. Majou, C. Gosselin, P. Wenger, and D. Chablat, "Parametric Stiffness Analysis of the Orthoglide", Mechanism and Machine Theory, vol. 42, no. 3, pp. 296-311, 2007.

[1.27] X. Kong, and C. M. Gosselin, "Kinematics and Singularity Analysis of a Novel Type of 3-CRR 3-DOF Translational Parallel Manipulator", International Journal of Robotics Research, vol. 21, no. 9, pp. 791-798, 2002.

[1.28] P.-L. Richard, C. M. Gosselin, and X. Kong, "Kinematic Analysis and Prototyping of a Partially Decoupled 4-DOF 3T1R Parallel Manipulator", Journal of Mechanical Design, vol. 129, pp. 611-616, 2007.

[1.29] X. Kong, and C. M. Gosselin, "Forward displacement analysis of a quadratic 4-DOF 3T1R parallel manipulator: The Quadrupteron", Applied Mechanics and Materials, 2013.

[1.30] X. Kong, "Kinematic Influence Method for the Kinematic Analysis of Spatial Linkages" (in Chinese), MSc Dissertation, Yanshan University, 1990.

[1.31] X. Kong, and Z. Huang, "Derivation of Kinematic Influence Coefficients for General Parallel Robot Mechanisms" (in Chinese), Journal of Northeast Heavy Machinery Institute, vol. 17, no.4, pp. 293-297, 1993.

[1.32] Z. Huang, and X. Kong, "Kinematic Analysis on the Spatial Parallel Mechanisms with Redundant Degree of Freedom" (in Chinese), Chinese Journal of Mechanical Engineers, vol. 31, no.3, pp. 44-50, 1995.

[1.33] P. Lambert, "Parallel Robots with Configurable Platforms", PhD Thesis, Technische Universiteit Delft, 2013.

[1.34] F. Pierrot, O. Company, “H4: A New Family of 4-DOF Parallel Robots”, International conference on Advanced Intelligent Mechtronics, pp. 508 – 514, 1999.

[1.35] O. Company, F. Pierrot and J.-C. Fauroux, “A Method of Modelling Analytical Stiffness of a Lower Mobility Parallel Manipulator”, Proceedings of the IEEE International Conference on Robotics and Automation, pp 3232 – 3237, 2005.

[1.36] C. Corrandini and J.-C. Fauroux, “Evaluation of a 4-Degree of Freedom Parallel Manipulator Stiffness”, Proceedings of the 11th World Congress in Mechanism and Machine Science, 2003.

[1.37] F. Pierrot, F. Marquet, O. Company and T. Gil, “H4 Parallel Robot: Modelling, Design and Preliminary Experiments”, Proceedings of the IEEE International Conference on Robotics and Automation, pp 3256 – 3261, 2001.

[1.38] S. Krut, O. Company, M. Benoit, H. Ota and F. Pierrot, “I4: A New Parallel Mechanism for Scara Motions”, Proceedings of the 2003 IEEE International Conference on Robotics & Automation, pp 1875 – 1880, September 2003.

[1.39] V. Nabat, M. Rodriguez, O. Company, F. Pierrot and S. Krut, “Par4: Very High Speed Parallel Robot for Pick-and-Place” Intelligent robots and Systems, pp 553 – 558, 2005.

[1.40] X. Kong, and Y. Jin, "Type Synthesis of 3-DOF Multi-Mode Translational/Spherical Parallel Mechanisms with Lockable Joints", Mechanism and Machine Theory, vol. 96, pp. 323-333, 2016.

[1.41] Y. Chen, Z. You, and T. Tarnai, "Threefold-Symmetric Bricard Linkages for Deployable Structures", International Journal of Solids and Structures, vol. 42, pp. 2287-2301, 2005.

[1.42] J. Gallardo, J.M. Rico, A. Frisoli, D. Checcacci and M. Bergamasco, “Dynamics of Parallel Manipulators by Means of Screw Theory”, Mechanism and Machine Theory, vol.38, pp 1113 – 1131, 2003.

[1.43] Y. Fang and L.-W. Tsai, “Structure Synthesis of a Class of 4-DoF and 5-DoF Parallel Manipulators with Identical Limb Structures”, the International Journal of Robotics Research, vol. 21, no. 9 pp 799 – 810, 2002.

[1.44] J. Gallardo-Alvarado, J.M. Rico-Martínez and G. Alici, “Kinematics and Singularity Analysis of a 4-dof Parallel Manipulator using Screw Theory”, Mechanism and Machine Theory, vol. 41, pp 1048 – 1061, 2006.

[1.45] X. Kong and C.M. Gosselin, “Type Synthesis of 3T1R 4-DOF Parallel Manipulators Based on Screw Theory”, IEEE Transactions on Robotics and Automation, vol. 24, no. 2, pp 181 – 190, 2004.

[1.46] X. Kong and C. Gosselin, “Type Synthesis of Parallel Mechanisms”, Springer Tracts in Advanced Robotics, vol.33, chapter 2, pp 19 – 23, 2007.

[1.47] J.M. Rico, J. Gallardo and J. Duffy, “Screw Theory and Higher Order Kinematic Analysis of Open Serial and Closed Chains”, Mechanism and Machine Theory, vol. 34, pp 559 – 586, 1999.

[1.48] H.-J. Su, D.V. Dorozhkin and J.M. Vance “A Screw Theory Approach for the Conceptual Design of Flexible Joints for Compliant Mechanisms”, ASME Journal of Mechanisms and Robotics, vol. 1, no.4, pp 1 – 8, 2009.

[1.49] S. Amine, S. Caro, P. Wenger and D. Kanaan, "Singularity Analysis of the H4 Robot using Grassmann-Cayley Algebra", Robotica, pp 1 – 10, 2012.

[1.50] X. Kong, and A. Johnson, "Classification of Screw Systems Composed of Three Planar Pencils of Lines", Journal of Mechanisms and Robotics, vol. 6, no. 2, 2013

[1.51] S. Agrawal, and B. Roth, "Statics of In-Parallel Manipulator Systems", Journal of Mechanical Design, vol. 114, no. 4, pp. 281-290, 1990.

[1.52] Mathematics Application Group, Inc. information website, "Mathematics Application Group, Inc. (MAGI) Synthavision", https://design.osu.edu/carlson/history/tree/magi.html.

[1.53] MathWorks website, "Simulink - Simulation and Model Based Design", http://uk.mathworks.com/products/simulink/index.html?s_tid=gn_loc_drop.

[1.54] MapleSoft website, "MapleSim - High Performance Physical Modelling and Simulation - Technical Computing Software", http://www.maplesoft.com/products/MapleSim/.

[1.55] National Instruments website, "LabVIEW", http://www.ni.com/labview/.

[1.56] MathWorks website, "Data Acquisition into Simulink - Example Model", http://www.mathworks.com/matlabcentral/fileexchange/12524-data-acquisition-into-simulink-example-model

[1.57] MapleSoft website, "MapleSim Help", http://www.maplesoft.com/support/help/.

[1.58] MapleSoft website, "MapleSim Driveline Library", http://www.maplesoft.com/products/toolboxes/driveline/.

[1.59] National Instruments website, "Using PID on FPGA Targets (PID and Fuzzy Logic Toolkit)", http://zone.ni.com/reference/en-XX/help/370401J-01/lvpidmain/pid_on_fpga/.

Chapter 2 Virtual chain approach to the type synthesis of parallel mechanisms

This chapter details the design process used to produce a parallel mechanism utilising a process called "Type Synthesis" which is well documented in the literature. This process investigates the number of constraints and DOF required for a particular task and assigns potential limb configurations for the final design. The virtual chain approach is one of the key approaches within type synthesis where the desired motion pattern of the mechanism is used to produce single-loop kinematic chains that both describe the motion of the desired mechanism but also contain the kinematic chain that will be used to construct a limb of the mechanism.

2.1 Introduction to Type Synthesis

The moving platform of a parallel mechanism with $\mathcal{F}$-DOF can have numerous types of motion depending on several factors. An example of this is a 3-DOF mechanism, which can have the motion type of a Delta robot (all translational), a planar motion with a rotation, a spherical motion where all 3-DOF of rotation are about a central point or any other combination of 3-DOF motion. This means that knowing only the number of DOF is insufficient to describe the motion of the mechanism. Therefore, type synthesis is used to develop all the possible types of parallel mechanisms that have $\mathcal{F}$-DOF or a specific motion pattern, and then a concept mechanism design can be produced.

Numerous published works [2.1-2.3] use type synthesis to produce parallel mechanisms for a specified number of DOF. This method is based on the mobility criterion of the mechanism described by the general mobility criterion equation (2.1) or the independent constraint equation (2.2):

$$\mathcal{F} = d(n - g - 1) + \sum_{j=1}^{g} f_j \tag{2.1}$$

where $\mathcal{F}$ is the mobility or relative DOF of a kinematic chain, n is the number of links including the base, g is the number or joints, f_j is the freedom of the j-th joint, and d is the number of independent constraint equations within a loop.

$$f = \sum_{j=1}^{g} f_j - \min \sum_{i=1}^{v} d_i \tag{2.2}$$

where f is the number of degrees of freedom of the mechanism or kinematic chain, v is the number of independent loops in the mechanism, and d_i is the number of independent constraints within loop i. This form of type synthesis allows for the various forms of $\mathcal{F}$-DOF parallel mechanisms to be generated [2.1] [2.4]. This approach is well suited for the production of mechanisms that satisfy the general mobility criterion (2.1) but it is not robust enough to determine parallel mechanisms that do not meet this criterion. For these types of parallel mechanism, Kong and Gosselin [2.5] proposed a method of utilising virtual chains to act as the desired motion pattern for a mechanism and then performing type synthesis with both the virtual chain and a single kinematic chain of the mechanism forming a single-loop kinematic chain.

In this chapter, the method proposed in [2.5] for investigating the type synthesis for a specific classification of mechanisms by utilising virtual chains will be reviewed with the intention of expanding this method in future chapters.

2.2 Virtual Chain Approach [2.1]

In order to perform type synthesis on a parallel mechanism the desired motion of the mechanism should be established. This is done to clarify the type of mechanism is being designed since the number of DOF, does not accurately describe its motion. For example, the Delta Robot (figure 2.1a), the spherical parallel mechanism (figure 2.1b) and the 3-*RPS* robot (figure 2.1c) all have 3DOF but different motion patterns. The Delta robot has three translational DOF whereas the 3-*RPS* mechanism has one translational DOF and two rotational DOF about the horizontal axis. The spherical parallel mechanism has three rotational DOF about a single point.

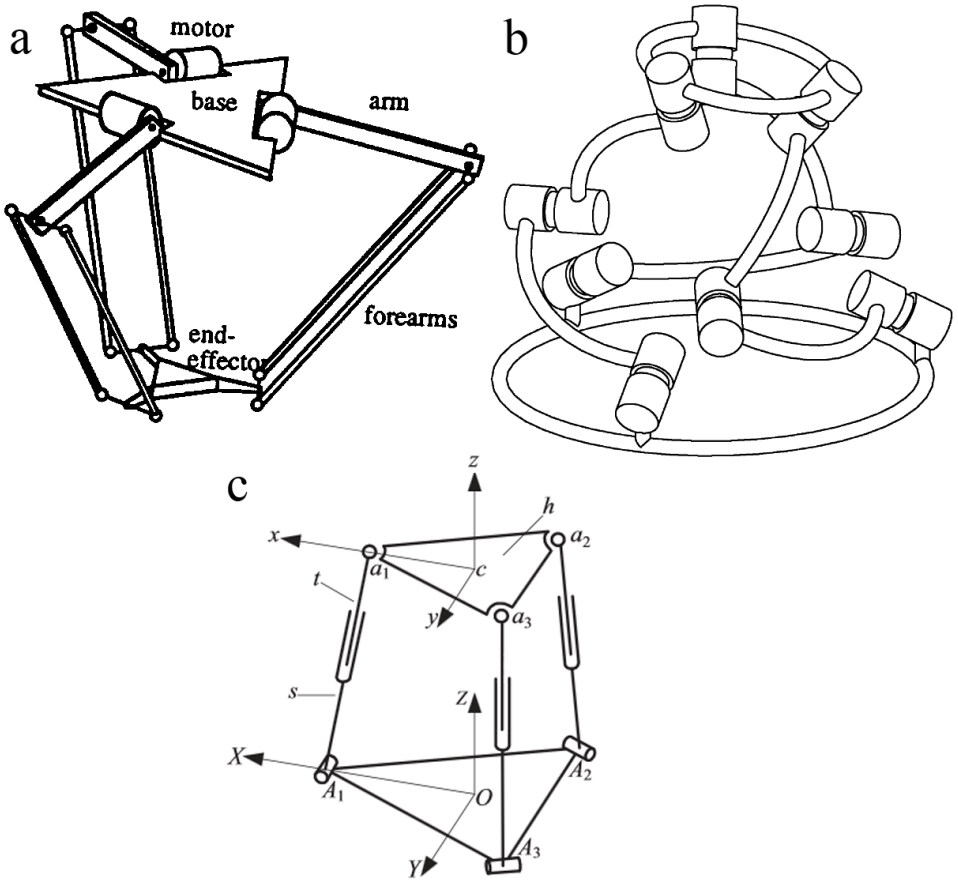

Figure 2.1: a) Delta Robot [2.6]; b) Spherical parallel Mechanism [2.7]; c) 3-*RPS* robot [2.8].

The motion pattern or desired motion of a mechanism represents the required motion of the moving platform or end-effector. One of the most common examples of a motion pattern in pick-and-place operations is the SCARA robot motion i.e. a three translational, one rotational 4-DOF motion pattern that allows the mechanism to manoeuvre around a production line and adjust the orientation of the end-effector in order to collect and place components.

2.2.1 Virtual chain

A virtual chain is a representation of the motion capabilities of the moving platform/end-effector of a mechanism. A virtual chain can be thought of as a series of kinematic joints and links that extend from the base of the mechanism or some arbitrary point to its moving platform or end-effector that accurately replicates the required motion pattern. It is generally taken as a rule that the most accurate and simplest representation of the motion pattern is selected as the virtual chain for the mechanism. This can be seen when using the wrench system represented by the PPP and $PRRR$ virtual chains in figure 2.2, both of which represent a 3-ζ_∞-system when the axes of the R joints are perpendicular to the axis of translation of the first P joint in the $PRRR$ chain. Since both of these systems are suitable as a motion pattern, the simplest motion pattern for the proposed system is chosen, which in this case is the PPP virtual chain.

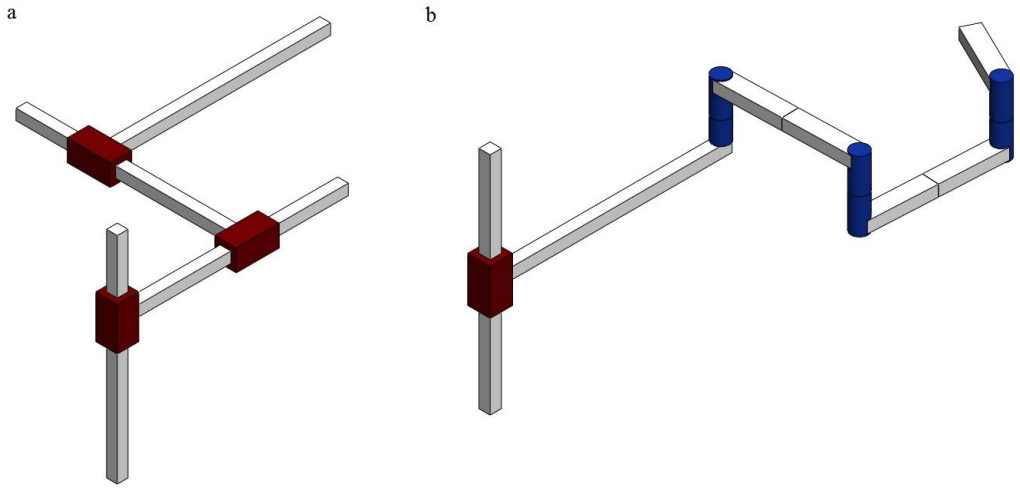

Figure 2.2: a) PPP virtual chain; b) $PRRR$ virtual chain.

Virtual chains are produced in either serial or parallel versions. A serial virtual chain is constructed as a single unbroken kinematic chain from the base of the mechanism to the moving platform or end-effector and represents the entire motion pattern of the mechanism. Parallel virtual chains are a minimum of two kinematic chains that also start at the base and extend to the moving platform of the mechanism.

2.2.2 Typical motion patterns of parallel mechanisms

In this section, several typical virtual chains of several motion patterns will be presented.

PPP virtual chain (figure 2.2a): The PPP virtual chain can adjust the translational position along any axis of the moving platform of the mechanism, while forcing it to maintain its orientation to the base. As the wrench system of this virtual chain is a 3-ζ_∞-system, the type of parallel mechanism most suited to this virtual chain would be a translational 3-DOF parallel mechanism.

E virtual chain (figure 2.3a): The E virtual chain can adjust the translational position along the horizontal axes of a mechanism and rotate the system about the vertical axis. As the wrench system of this virtual chain is a 2-ζ_∞-1-ζ_0-system where the ζ_∞ act along the horizontal plane and the ζ_0 acts perpendicular to the axis of all ε_∞. The type of parallel mechanism most suited is a planar 3-DOF mechanism.

S virtual chain (figure 2.3b): The S virtual chain is capable of rotating the moving platform about the centre of the S joint but prevents the translation of the centre of the S joint along any direction. The wrench system for this virtual chain is a 3-ζ_0-system where all ζ_0 pass through the centre of the S joint. The type of parallel mechanisms most suited for this virtual chain would be spherical.

The $PPPR$ virtual chain (figure 2.3c): The $PPPR$ virtual chain produces Schönflies motion, more commonly known as SCARA motion. Schönflies motion is made up of three translational components as well as a single rotational component about one of the axes. The wrench system of this virtual chain is a 2-ζ_∞-system in which all the ζ_∞ are perpendicular to the axis of the R joint. The type of mechanism recommended for this virtual chain is a 3T1R parallel mechanism.

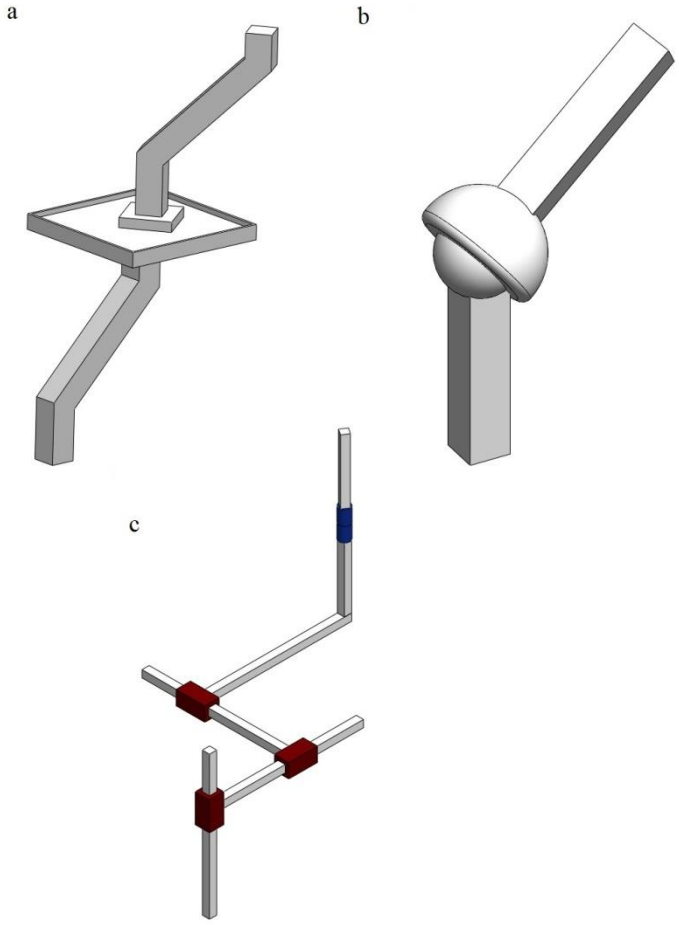

Figure 2.3: a) E virtual chain; b) S virtual chain; c) $PPPR$ virtual chain.

The PPP and $PPPR$ virtual chains are ideal for the types of mechanism found in pick-and-place and as such will be utilised in later chapters.

2.2.3 Type synthesis of parallel mechanisms

The virtual chain method to carry out type synthesis is determined by investigating the motion pattern of a mechanism and a specified number of over (or redundant) constraints (Δ) present. The reasoning behind this approach is that for many applications, a specific motion pattern is required for the parallel mechanism to complete its task and the number of over constraints aids the characterisation of the mechanisms mechanical properties.

For this approach, we need to know the connectivity between each link of the kinematic chain using the following equations (chapter 2 of [2.5]):

$$\sum_{i=1}^{m} c^i = 6 - C + \Delta = 6 - \mathcal{F} + \Delta \tag{2.3}$$

where c is the connectivity (number of DOF) between two links, usually the base and moving platform for parallel mechanisms; m is number of kinematic chains between the base and moving platform of the parallel mechanism not counting the virtual chain; C is the twist system of a single-loop kinematic chain; $\mathcal{F}$ is the mobility (total number of DOF) of the parallel mechanism; i is the ith leg of the parallel mechanism. We also need to determine the wrench of the system using the following equation:

$$\mathcal{W} = \sum_{i=1}^{m} \mathcal{W}^i \tag{2.4}$$

where $\mathcal{W}$ is the wrench system of the mechanism or kinematic chain.

$$f^i = 6 - c^i \tag{2.5}$$

where f^i is the number of DOF in the ith kinematic chain.

During the type synthesis of the parallel mechanism being designed in order to produce a more generalized method, Δ can range from zero to a maximum value determined by equation (2.3). The motion pattern corresponding to a virtual chain will be referred to as a V-motion and a parallel mechanism which generates a particular V-motion will be referred to as V= parallel mechanism which stands for a parallel mechanism whose motion is equivalent to that of the V joint. Considering a 3-DOF V= parallel mechanism, connecting the base and moving platform of the parallel mechanism with a parallel kinematic chain and the desired virtual chain produces the V-motion of the parallel mechanism; the function of the mechanism will not be affected. Therefore, any of the kinematic chains of the parallel mechanism and associated virtual chain constitutes a 3-DOF single-loop kinematic chain, which means that the wrench system must be the same as the virtual chain for any general configuration. Therefore, a parallel mechanism is a V= parallel mechanism if it meets the following two criteria:

1. Each kinematic chain of the parallel mechanism and a virtual chain make an $\mathcal{F}$-DOF single-loop kinematic chain.
2. The wrench system of the parallel mechanism is the same as the virtual chain in any general configuration.

This ensures that the moving platform can achieve the prescribed V-motion and that the virtual chain and the parallel mechanism have the same number of DOF. This setup results in the kinematic chains being over constrained due to the virtual chain constraining the mechanism to its prescribed motion pattern as it already constrains all the constraint wrenches.

The steps for developing the type synthesis of a parallel mechanism using the virtual chain approach will now be defined:

2.2.3.1 Step 1: Decomposition of the wrench system of a parallel manipulator

The first step of the type synthesis process is to determine all the leg-wrench systems and all the possible combinations of m-leg layouts for the chosen V= parallel mechanism and

a specified number of over constraints for the mechanism which is done using equations (2.3) and (2.4).

To determine the leg-wrench system (the wrench system of a single kinematic chain of the parallel mechanism) it is known that any leg-wrench system in a V= parallel mechanism must be a sub-system of the virtual chain's wrench system. The sub-system of the leg-wrench system is composed of a combination of the available wrenches in the V= parallel mechanism wrench system. For example, a translational parallel mechanism, which has a PPP virtual chain (figure 2.2a), has a 3-ζ_∞-system so therefore the sub-system for the legs can either be: a) a 0-system; b) a 1-ζ_∞-system; c) a 2-ζ_∞-system; or d) the full 3-ζ_∞-system. The sub-systems of the wrench system must combine to reform the full leg-wrench system. This can be seen using a 3-ζ_∞ leg-wrench system in which the mechanism has 3 kinematic chains, if the first kinematic chain is a 2-ζ_∞-system and the second and third are a 1-ζ_∞-system then the kinematic chains used are acceptable only if two of the ζ_∞ are parallel with the other two being perpendicular to each other and the two linear ζ_∞.

The motion patterns being considered in this thesis will primarily comprise wrenches of the same pitch i.e. ∞ pitch or 0 pitch. The connectivity of the leg-wrench systems of a mechanism are classified as $c^i\left(0 \leq c^i \leq c\right)$-systems of the same pitch and the combination of these leg-wrench systems can be represented by the combination of the order of c^i (see equation (2.3)). An example for the combinations of c^i for a 3-DOF parallel mechanism with two or three legs is shown in table 2.1 [2.5].

2.2.3.2 Step 2: Type synthesis legs for mechanisms

With the establishment of the available combinations of leg-wrench systems, the kinematic chains that instantiate these combinations can be formed. This is done in two stages:

a) The type synthesis of an $\mathcal{F}$-DOF single-loop kinematic chain that includes the virtual chain and one of the desired leg-wrench systems.

b) Generate the type of legs for the V= parallel mechanism by removing the virtual chain from the chain gained in step a).

In this section a 3-DOF single-loop kinematic chain will be studied. The first stage is to perform the type synthesis of the single-loop kinematic chain that includes a PPP connection that represents the virtual chain component of the kinematic chain.

To perform type synthesis, the following sequence of analysis must first be taken:

(1) Determine the number of joints. The number of joints is calculated with the equation below:

$$f = F + (6 - c) \tag{2.6}$$

where f is the number of 1-DOF joints and F is the mobility of the single-loop kinematic chain.

Table 2.1: Combinations of c^i for a 3-DOF parallel manipulator with two to four legs [2.5].

m	c	Δ	c^1	c^2	c^3
2	3	3	3	3	
		2	3	2	
		1	3 2	1 2	
		0	3 2	0 1	
3	3	6	3	3	3
		5	3	3	2
		4	3 3	3 2	1 2
		3	3 3 2	3 2 2	0 1 2
		2	3 2 2	2 1 2	0 1 1
		1	3 2 2	1 2 1	0 0 1
		0	3 2 1	0 1 1	0 0 1

(2) Find the geometric conditions on the joint axes based on the twist-wrench relationships based on the conditions given in section 1.3. This can be any of the following:

- The axis of an R joint is coplanar with the axis of any ζ_0 within the desired wrench system.
- The direction of a P joint is perpendicular to the axis of any ζ_0 within the desired wrench system.
- The axis of an R joint is perpendicular to the axis of any ζ_∞ within the desired wrench system.

(3) Identify the types of single-loop kinematic chains with the desired wrench system.

From (2) a set of single-loop kinematic chains with the desired wrench system that can be constructed using compositional units (see chapter 2, page 30 - 32 of [2.5]). By inserting p co-axial or co-directional compositional units into a c-ζ-system to make a $(c+p)$-ζ-system, single loop kinematic chains can be obtained. The DOF and number of joints of a single-loop kinematic chain with a $(c+p)$-ζ-system is:

$$\mathcal{F}' = \mathcal{F} - (f_p - p) \tag{2.7}$$

where f_p is the total number of 1-DOF joints within the p compositional units, $\mathcal{F}'$ is the mobility of the parallel mechanism when all its actuated joints are blocked and

$$f' = f - f_p \tag{2.8}$$

In the PPP= parallel mechanism (3-ζ_∞-system) case the first step is solved by identifying that the connectivity of the system (c) is 3, which means that from equation (2.6) we get the number of joints (f) as $F + 3$. (2) uses the fact that the 3-ζ_∞-system can be chosen to be three ζ_∞ that are not parallel to a plane (figure 2.4).

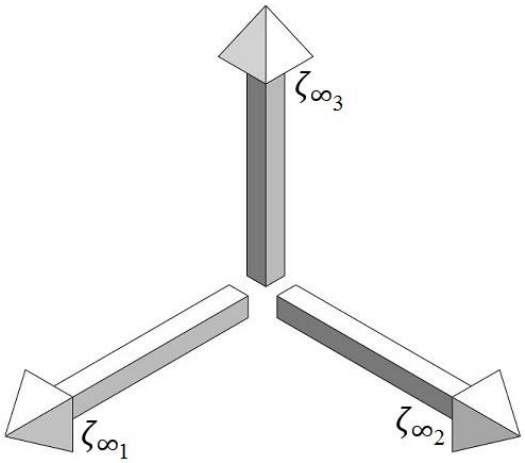

Figure 2.4: 3-ζ_∞ system

This means that the system geometric conditions for the joints comprise zero R joints that, if true, indicates that no R joint whose axis of rotation is perpendicular to the three wrenches can exist.

This assumes that these types of single-loop kinematic chains with the desired wrench system contain a special translational compositional unit (PPP) composed of $f = (F + 3)$ P joints arranged with their axes of translation being perpendicular to each other. As this compositional unit gives the loop all three of the ζ_∞ desired for the system, the rest of the loop only has to achieve a 3-ζ_∞-system or less depending on the desired system mobility.

Once the single-loop kinematic chains are categorised, any single-loop kinematic chains that have twists not included in the virtual chain can be discarded. Again using the 3-DOF single-loop kinematic chain, while considering the desired wrench system to be a 2-ζ_∞-system utilising the PPP virtual chain section, the number of joints is equal to $F + 4$. If the mobility of the mechanism is 3-DOF then the total number of joints will be seven, with three of those joints making up the PPP virtual chain. While the P joints of the virtual chain can be placed throughout a single-loop kinematic chain and still produce the same result, only the configurations in which three P joints are together and aligned along different axes are considered. The resulting single-loop kinematic chains are then inspected and the ones that do not satisfy the twists criteria of all the joints, excluding those in the virtual chain, are linearly dependent. Examples of this would be the ($RRRRV$) single-loop kinematic chain, where V stands for the prescribed virtual chain which in this case is a chain of three P joints (figure 2.5a) due to the twists of the R joints being linearly dependent and the nature of the ($PRRRV$) single-loop kinematic chain (figure 2.5b).

There are numerous cases in which the twists of the joints within one kinematic chain are linearly dependent; however, the following conditions allow us to gain kinematic chains where the twists of the joints are linearly independent [2.5]:

- There are no coaxial R joints.
- There are no P joints operating along the same direction.
- The operating direction of at most one P joint is parallel to the axis of an R joint.
- At most three R joints have parallel axes.
- The axes of at most three R joints are passing through a single point.
- The operating directions of at most two P joints are parallel to the same plane.
- The sum of the number of R joints with parallel axes and the number of P joints is not greater than four.
- The operating directions of x P joints are perpendicular to the axes of rotation of y R joints with parallel axes then $x + y \leq 3$.

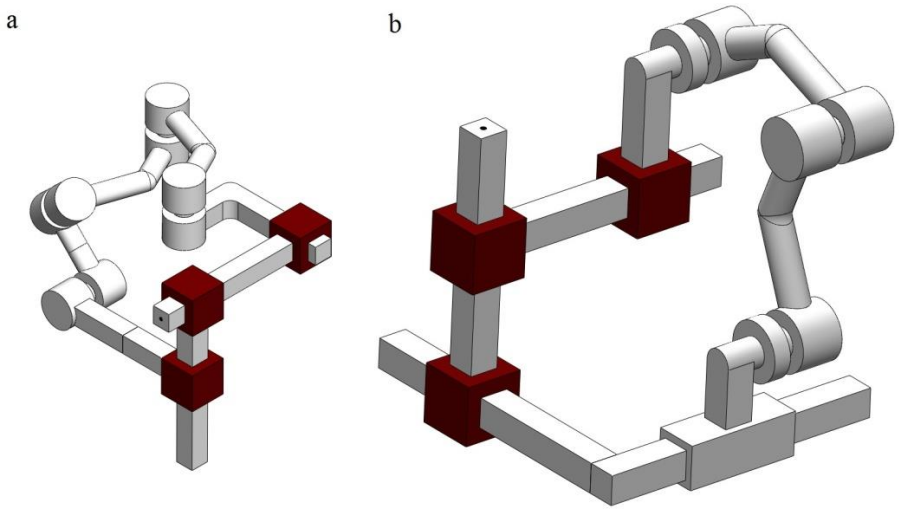

Figure 2.5: a) $RRRRV$ single-loop kinematic chain; b) $PRRRV$ single-loop kinematic chain.

To generate the type of legs in the parallel mechanism having already obtained the types of $\mathcal{F}$-DOF single-loop kinematic chains that include a viable virtual chain and consist of a desired leg-wrench system; the types of parallel mechanism legs required can be generated by removing the virtual chain from the single-loop kinematic chain. By doing this, the geometric conditions ensure that each leg, when put together with the same virtual chain, will constitute an $\mathcal{F}$-DOF single-loop kinematic chain.

2.2.3.3 Step 3: Assembly of legs

The type synthesis of parallel mechanisms consists of obtaining the different types of kinematic chains and assembling them in the form of the kinematic chains from Step 2 in accordance to the combinations of the m leg-wrench systems gained from Step 1.

When a parallel manipulator's kinematic chains have unchanging leg-wrench systems, i.e. all leg-wrench systems remain unchanged with regards to the base and/or the moving

platform, then the condition that ensures that the linear combination of all of the parallel mechanisms leg-wrench systems make up the leg-wrench system of the virtual chain will be revealed. If the parallel manipulator's kinematic chains are capable of change then all of the kinematic chains of a mechanism must produce the desired wrench system.

An example of this can be found when performing Step 3 on the 3-$PRRR$ parallel mechanism with a V-motion of PPP= parallel mechanism. The leg-wrench systems are of the former variety in which they do not change about the base or moving platform of the mechanism. By ensuring that the R joints connected to the moving platform are not parallel, it is guaranteed that the linear combination of the three 2-ζ_∞-systems form the 3-ζ_∞-system produced by the virtual chain.

In the event of the leg-wrench system varying with the configuration of the mechanism it is possible to enter a constraint singularity. A constraint singularity is a configuration of the parallel mechanisms when an extra DOF becomes available to the moving platform that cannot be controlled by actuators of the mechanism. This is caused by the constraint wrench system degenerating when the configuration of the mechanism causes the moving platform to have fewer DOF than the number of joints in any leg.

2.2.3.4 Step 4: Selection of the actuated joints

The process of selecting the actuated joints of the parallel mechanism requires identification of all of the possible parallel mechanisms that uses a particular V-motion for a given set of kinematic chains and removing those for which the actuated joints are invalid. For any selection, the following rules are followed:

1. The actuated joints should be distributed evenly between the kinematic chains
2. The actuated joints should be located towards (if not at) the base of the mechanism
3. P joints should always be actuated.

The reasoning behind rule 3 is that a P joint tends to perform poorly due to sliding friction and any offset angle or bending will lead to an increase in force being required to operate the joint. The first two rules are useful in the optimisation of the mechanism since a distributed set of actuated joints will ensure a more even power distribution among its kinematic chains as well as providing greater control. Also having the actuated joints closer to the base of the mechanism will aid the reduction of the moving mass of the mechanism and increase its efficiency.

As discovering all the possible combinations legs of V= parallel mechanisms is trivial, the selection of the actuated joints and their location will be used to determine the candidate for the V= parallel mechanism.

The number of actuated joints recommended for an $\mathcal{F}$-DOF parallel mechanism is $\mathcal{F}$. The selection of which joints should be actuated should be a set of joints that, when the mechanism is in a general configuration, the number of DOF blocked by the actuated joints should be zero.

The validity condition for the selection of actuated joints is performed by looking at the actuation wrenches of the proposed actuated joints. Let $\mathcal{W}_{\not\Rightarrow j}^{i}$ be the set of all wrenches that are not reciprocal to the twist of joint j and are reciprocal to all of the twists of the

other joints within the given kinematic chain, then $\mathcal{W}_{\not\partial j}^{i}$ is the set of wrenches that imposed upon the moving platform through actuating that kinematic chain's actuated joint.

Using ζ_j^i to represent the basis of the wrench system $\mathcal{W}^i$ and using $\zeta_{\not\partial j}^{i}$ to represent any one wrench that is part of $\mathcal{W}_{\not\partial j}^{i}$, then any wrench in $\mathcal{W}_{\not\partial j}^{i}$ can be expressed as:

$$\zeta_{\not\partial j}^{\prime i} = \alpha\zeta_{\not\partial j}^{i} + \sum_{k=1}^{c^i}\left(\beta_k^i\zeta_k^i\right),\ \alpha \neq 0 \tag{2.9}$$

where $\zeta_{\not\partial j}^{i}$ is referred to as the actuation wrench of joint j in chain i. For a non-redundant $\mathcal{F}$-DOF parallel mechanism, a set of actuated joints are considered valid if $\mathcal{F}' = 0$. From equation (2.3):

$$c' = 6 - \mathcal{F}' = 6 \tag{2.10}$$

where c' is the order of the wrench system of the parallel mechanism with all of its actuated joints blocked. From this it can be stated that for an $\mathcal{F}$-DOF parallel mechanism in which all twists within a single kinematic chain are linearly independent in a general configuration, then a set of $\mathcal{F}$ actuated joints are valid only if the basis wrenches of the wrench system of the mechanism is of an order of 6 when all actuated joints are blocked.

The procedure for the detection of a valid set of actuated joints for a parallel mechanism will now be outlined:

a) If one or more of the actuated joints of a candidate parallel manipulator are inactive then the proposed set of actuated joints is to be considered invalid. For the kinematic chains obtained in utilising the virtual chain approach to type synthesis the inactive joints are revealed during the type synthesis of the mechanism.

b) If the determinant of an $\mathcal{F} \times \mathcal{F}$ matrix of the t-components of all the actuation wrenches of actuated joints is always zero then the set of actuated joints for the kinematic chain satisfying these conditions is invalid, causing the candidate parallel mechanism to be discarded. Only the parallel manipulators in which all of the elements of the $\mathcal{F} \times \mathcal{F}$ matrix are constant and the determinant non-zero are to be considered proper and will have no constraint singularities.

2.3 Summary

In this chapter the process laid out in [2.5] for utilising virtual chains to perform type synthesis on a desired motion pattern was detailed for use in the future development of producing an extended theory, incorporating multiple moving platform mechanisms. The next chapter begins the investigation of the design process in which the first step is to develop the type synthesis method to allow for the development of parallel mechanisms that have multiple moving platforms with connecting kinematic chains between them.

References

[2.1] C.F. Earl, and J. Rooney, "Some kinematic structures for robot manipulator design", Journal of Robotic Systems, vol. 18, no. 5, pp. 213 - 219, 1983.

[2.2] J.M. Hervé, and F. Sparacino, "Structural synthesis of parallel robots generating spatial translation", Proceedings of the fifth International Conference on Advanced Robotics, pp. 808 - 813, 1991.

[2.3] K.H. Hunt, "Structural kinematics of in parallel-actuated robot-arms", ASME Journal of Mechanical Design, vol. 105, no.4, pp. 705 - 712, 1983.

[2.4] L.W. Tsai, "The enumeration of a class of three-DOF parallel manipulators", Proceedings of the 10th World Congress on the Theory of Machines and Mechanisms, pp. 1121 - 1126, 1999.

[2.5] X. Kong, “Type Synthesis of Parallel Mechanisms”, Springer Books, vol. 33, Chpt. 2 - 5, pp. 43 - 83, 2007.

[2.6] P. Vischer, and R.Clavel, "Kinematic Calibration of the parallel Delta robot", Robotica, vol. 16, pp. 207 - 218, 1998.

[2.7] I.A. Bonev, and C.M. Gosselin, "Analytical Determination of the Workspace of Symmetrical Spherical Parallel Mechanisms", IEEE Transactions on Robotics, vol. 22, no. 5, pp. 1011 - 1017, 2006.

[2.8] Y. Zhao, J.F. Liu, and Z. Huang, "A force analysis of a 3-RPS parallel mechanism by using screw theory", Robotica, vol. 29, pp. 959 - 965, 2011.

Chapter 3 Multi-platform parallel mechanism design

From the literature on the virtual-chain method for type synthesis in Chapter 2 and screw theory in Chapter 1, this chapter further develops the virtual chain type synthesis methodology to incorporate a class of parallel mechanisms with multiple platforms. This methodology will then be tested on two parallel mechanisms to a design specification with the intention of further developing the best candidate into a physical prototype in Chapter 7.

3.1 Introduction

This chapter investigates the initial design ideas for a novel parallel mechanism. The concept of type synthesis outlined in Chapter 2 will be expanded to incorporate the existence of multiple moving platforms and then applied to the design process approach. The mechanism to be designed will be detailed in section 3.2 with the industrial focus being a pick-and-place parallel mechanism similar to the operation of the Delta robot described in Chapter 1.

The primary reason for designing a mechanism with multiple moving platforms and end-effectors is to determine how such a mechanism can be made, how the mechanism could potentially operate in an industrial capacity and if the mechanism can be made more efficient than a single end-effector variant. As this process is followed, the design process will be studied to determine if a more intuitive operational focus approach can be developed.

Another reason for developing a multi-platform parallel mechanism comes from the theory that if a person were to move a stack of books from one area of a room to another, they would use both hands to carry twice the load a single hand could carry, as it would be more efficient than the latter. Therefore, taking this analogy through to the concept of parallel robotics, the concept of having a parallel mechanism that is capable of completing twice the number of tasks would theoretically be more efficient than a traditional mechanism.

3.2 Design Specifications

The design requirements of the proposed parallel mechanism are as follow:

- The mechanism must be capable of utilising multiple end effectors, which operate either in unison or independently of each other.
- The mechanism must have at least two or more moving platforms.
- The mechanism must have a minimum of three translational DOF for each moving platform and at least one relative DOF between each moving platform.

To fulfil these requirements, the mechanisms will have three translational DOF acting along the three world axes and a pair of moving platforms capable of 1- or 2- relative DOF (rDOF) along the horizontal axes, maintaining a common plane of operation for the end-effectors.

3.3 Type Synthesis of Multi-Platform Parallel Mechanisms

From the method detailed in Chapter 2, the virtual chain approach to type synthesis can be expanded to include parallel mechanisms that have multiple moving platforms that have x-rDOF of motion between the individual platforms (figure 3.1).

This means that the end-effectors of the mechanism can operate independently of each other within the confines of the rDOF of the mechanism. In order to determine the type synthesis of such mechanisms, the desired motion pattern needs to be determined and then the simplest virtual chain design applied to allow for the desired motion pattern of each platform while offering no additional DOF to the overall mechanism.

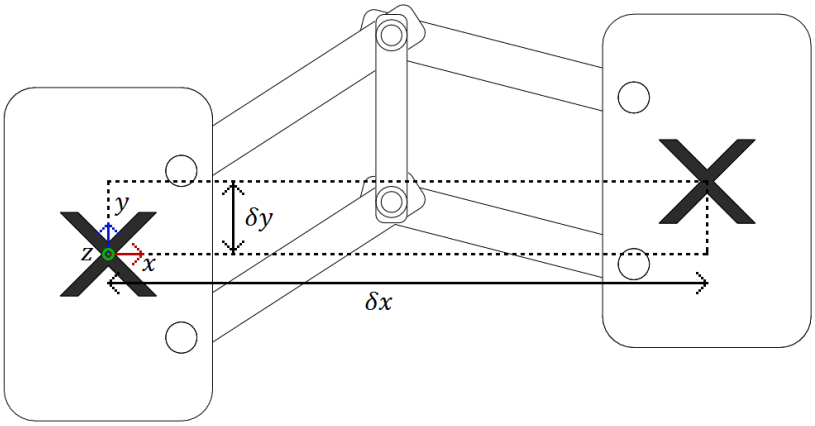

Figure 3.1: Multiple moving platforms with 2-rDOF.

Once the virtual chains for the platforms have been chosen, they and the moving platforms form a single-loop kinematic chain where the connecting link between the platforms forms the main constraint on the system's motion, limiting the rDOF of the system (figure 3.2).

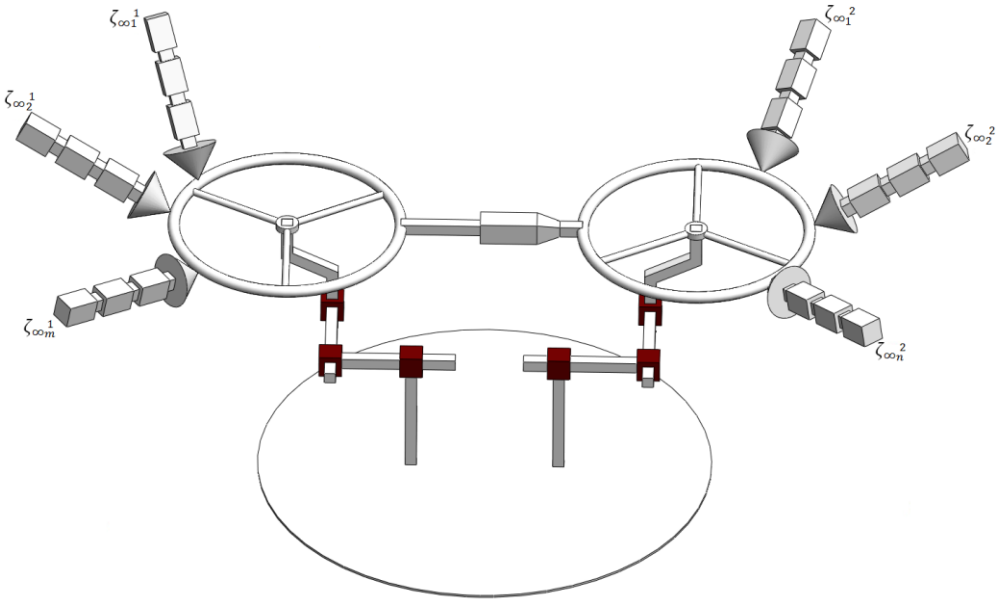

Figure 3.2: Single-loop kinematic chain with two virtual chains, moving platforms and connecting link.

The following section details the procedure to determine whether the proposed changes to the type synthesis method can achieve the desired result of producing various parallel mechanisms with a predetermined number of rDOF between two moving platforms.

This is done by graphically determining the total constraint couples on each moving platform within this single-loop mechanism by using the reciprocal screw theory detailed in Chapter 1. From this point, the methodology on type synthesis with virtual chains laid out in Chapter 2 is then followed.

3.4 Type Synthesis of Multiple Operation Mode Mechanisms

The virtual chain approach to type synthesis can also be applied to mechanisms with multiple operation modes by first investigating the mechanism from the perspective of one operational mode and then the next until all operational modes have been investigated. The resulting virtual chains from the numerous type syntheses can then be combined to form a serial or parallel virtual chain capable of performing each operational mode movement while not adding additional DOF to other operational modes.

An example of this is the DIRECTOR mechanism described in Chapter 1. The mechanism has two operational modes: a planar translation along the vertical and horizontal (y- and x-) axes when observed from the front and a rotation about the horizontal axis; and the same translational DOF however, the rotation of the moving platform is altered to be about the z-axis. The change between operational modes is achieved through the locking of certain joints that prevent one rotation or another.

The virtual chain for this mechanism therefore would be a pair of PPR virtual chains where the virtual P joints are aligned along the same axes but the virtual R joints are perpendicular. From this point the type synthesis of the mechanism can be performed much in the same manner as detailed in Chapter 2.

3.5 4-DOF Parallel Manipulators

In this section, the architecture being investigated will be of a 4-DOF configuration, where the robot will have a combination of three translational DOF and one rDOF with the single rDOF consisting of either a translational DOF between the two end-effectors or a constant radius DOF where one platform is manoeuvred around a particular point with a constant radius.

Using the extension to the type synthesis of a mechanism, the virtual chain pair that fits the desired motion pattern can either be a pair of PPP virtual chains or $PRRR$ virtual chains. Since the simplest solution is the one being used, both virtual chains will consist of a PPP chain with each P joint aligned along a different world coordinate. The constraint couples formed by the moving platform link will limit the motion of the moving platform to a single rDOF. This means that the proposed mechanisms will require either a combination of P joints forming a single, more robust P joint in order to connect the moving platforms or, using the rules of reciprocal screws stated in Chapter 1, one or more kinematic chains made of coplanar R joints.

The potential moving platform links being investigated in this Chapter are split into a 1-rDOF and 2-rDOF designs with the resulting mechanism having 4-DOF and 5-DOF respectively. The 1-rDOF designs proposed are as follows:

- A pair of RR links that form a parallelogram that produce a translational motion of one of the end-effectors along a curved path (figure 3.3).

- A Sarrus mechanism, which comprises of a pair of RRR links whose planes are offset by 90° from each other producing a single translational rDOF (figure 3.4b).
- A P joint producing a single translational rDOF (figures 3.4a and 3.4c)

With the moving platform connection links determined, the kinematic chains of the mechanism can be determined. Using the method laid out in Chapter 2, the type synthesis of the kinematic chains of the mechanism can be determined. The table for the connectivity is displayed in table 3.1.

From table 3.1, a series of potential kinematic chains can be constructed to produce the desired motion pattern. Some of these kinematic chains are detailed in the following sub-sections.

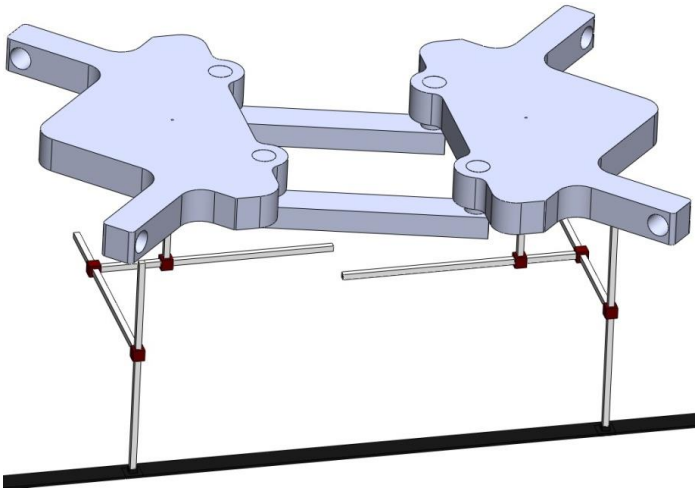

Figure 3.3: Single-loop 1-rDOF kinematic chain with moving platform having a parallelogram connecting kinematic chain.

It can be found, that the total constraints on each platform is a 3-couple system, therefore each leg can be a leg with an n-couple system (table 3.1) where n is a value between zero and the full couple system of the mechanism. This means that the kinematic chains of a given mechanism must be limited to producing at a maximum, the complete n-couple system of the mechanism.

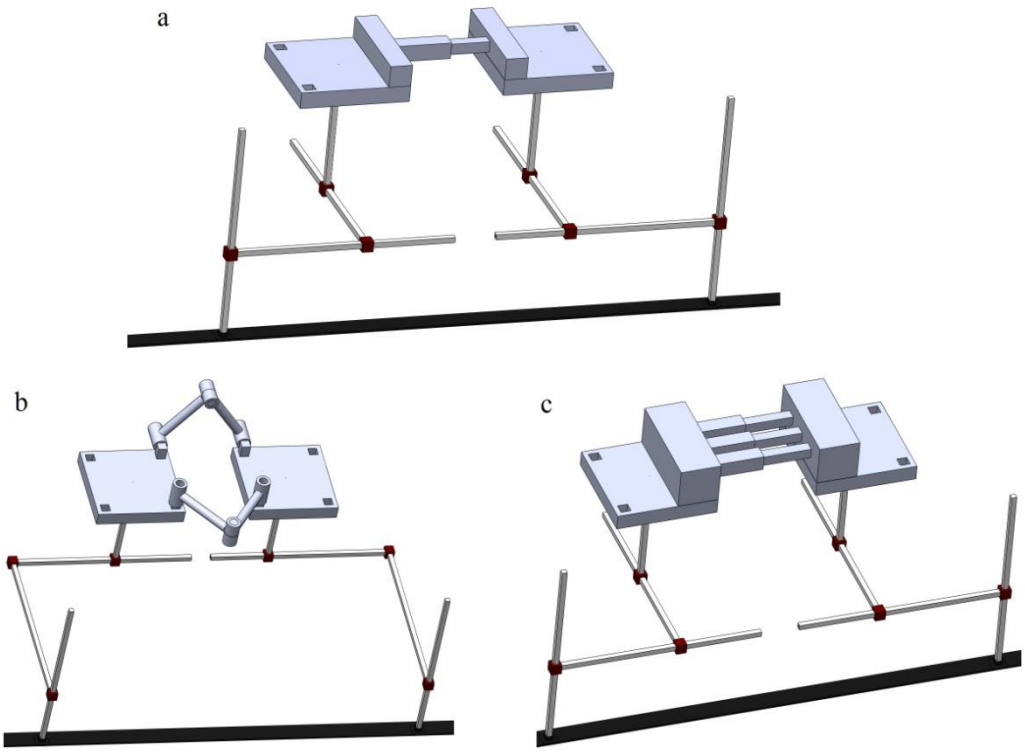

Figure 3.4: Single-loop 1-rDOF kinematic chain with a) Generalised case of *P* joint linkage; b) Dual RRR moving platform connecting links; c) Triple *P* joint moving platform connecting link.

For example, a four-legged parallel mechanism (figure 3.7) can be represented by a 2-$PRRR$-(MP1-P-MP2)-2-$PRRR$. In which, the 2-$PRRR$ legs connect the primary moving platform to the base of the mechanism and the 2-$PRRR$ legs connect the secondary moving platform also to the base of the mechanism, while the two platforms are connected by a *P* joint (represented in figure 3.7 by a trio of *P* joints to improve stability).

Table 3.1: Combinations of kinematic chains for two legged platforms.

m	c	Δ	c^1	c^2
2	2	4	3	3
		3	3	2
		2	3	1
		1	3 2	0 1
		0	2 1	0 1
2	3	3	3	3
		2	3	2
		1	3 2	1 2
		0	3 2	0 1

3.5.1 4-DOF Parallel Mechanism Design 1

The initial design again uses kinematic chains with a connectivity of 3 or 4 each. The layout for the kinematic chains for this mechanism is a *PRRR* format, in which each *R* joint is perpendicular to the *P* joint. The kinematic chains are spaced out evenly again with a 90° angle between each limb. The two limbs with a shared axis are to be connected to separate moving platforms in order to allow the rDOF to occur. The joint map for this design is shown below in figure 3.5.

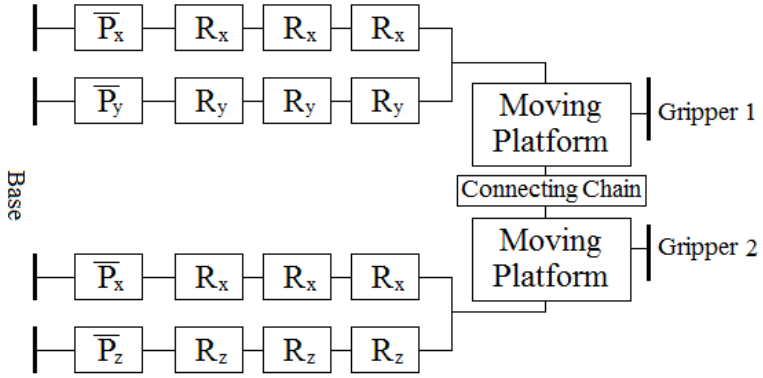

Figure 3.5: Joint map of 4-DOF Parallel Mechanism 1 Design 1.

For this mechanism, the last two 1-rDOF moving platform designs are utilised. The reason the first is ignored is due to the radial rDOF being impossible to fully control in this limb format. The designs of the mechanism with each moving platform design are shown below in figures 3.6 and 3.7.

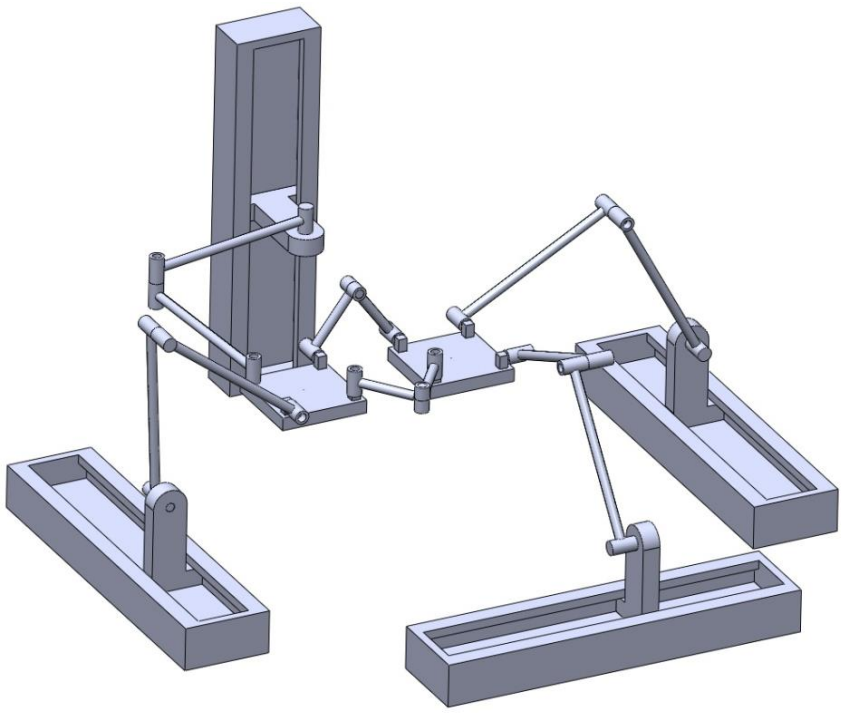

Figure 3.6: 4-DOF Parallel Mechanism Design 1 with dual *RRR* moving platform link.

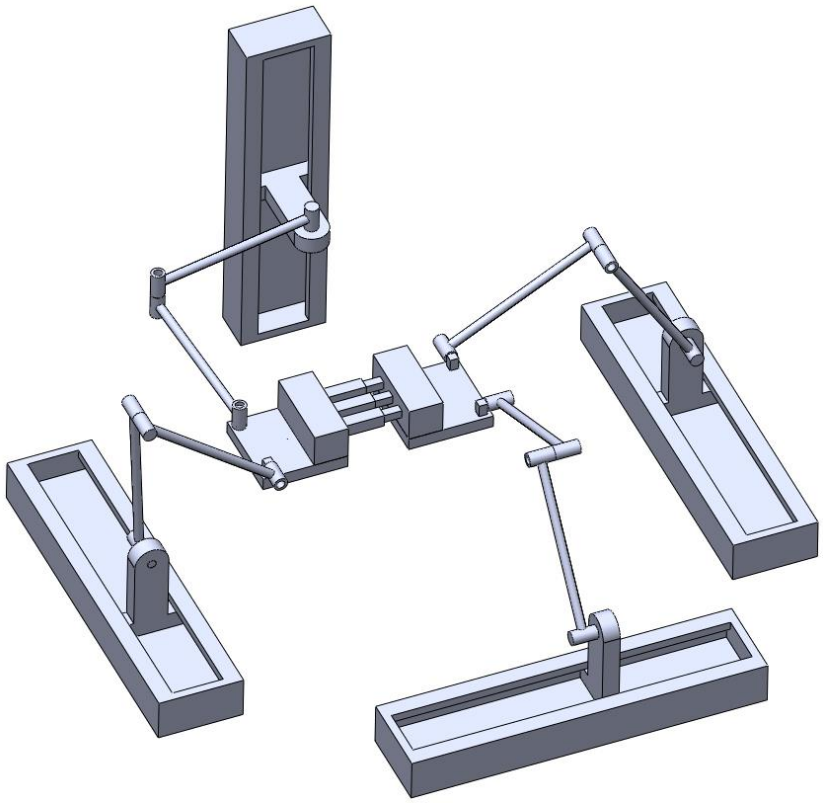

Figure 3.7: 4-DOF Parallel Mechanism Design 1 with *P* joint moving platform link.

The *P* joints of each limb is chosen as the actuated joints for the mechanism due to the fact that the third rule for selecting the actuated joints of a parallel mechanism detailed in Chapter 2 states that *P* joints should always be actuated. The *P* joints are also closest to the base of the mechanism and evenly distributed between the limbs.

3.5.2 4-DOF Parallel Mechanism Design 2

The second design also utilises kinematic chains with a connectivity of 3-DOF. The layout for the kinematic chains for this mechanism is a *PUU* format constructed from four *R* joints set perpendicular to their respective U joint companion, where the first and fourth *R* joint and second and third *R* joint pairs are parallel with each other. The kinematic chains are again spaced out evenly with a 90° angle between each limb. The joint map for this design is shown below in figure 3.8.

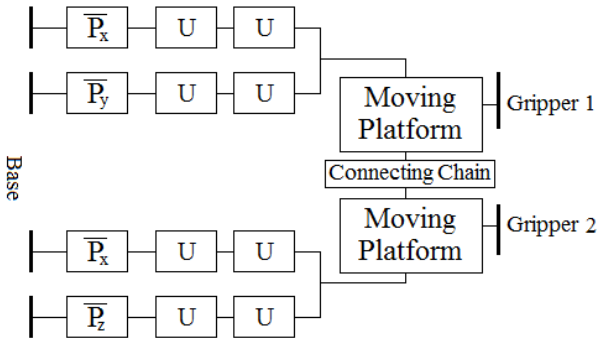

Figure 3.8: Joint map of 4-DOF Parallel Mechanism Design 2

For this mechanism, all three proposed moving platform links are investigated in order to determine how well each link design can be controlled by the mechanism design. The designs of the mechanism with each moving platform design are shown below in figures 3.9, 3.10 and 3.11.

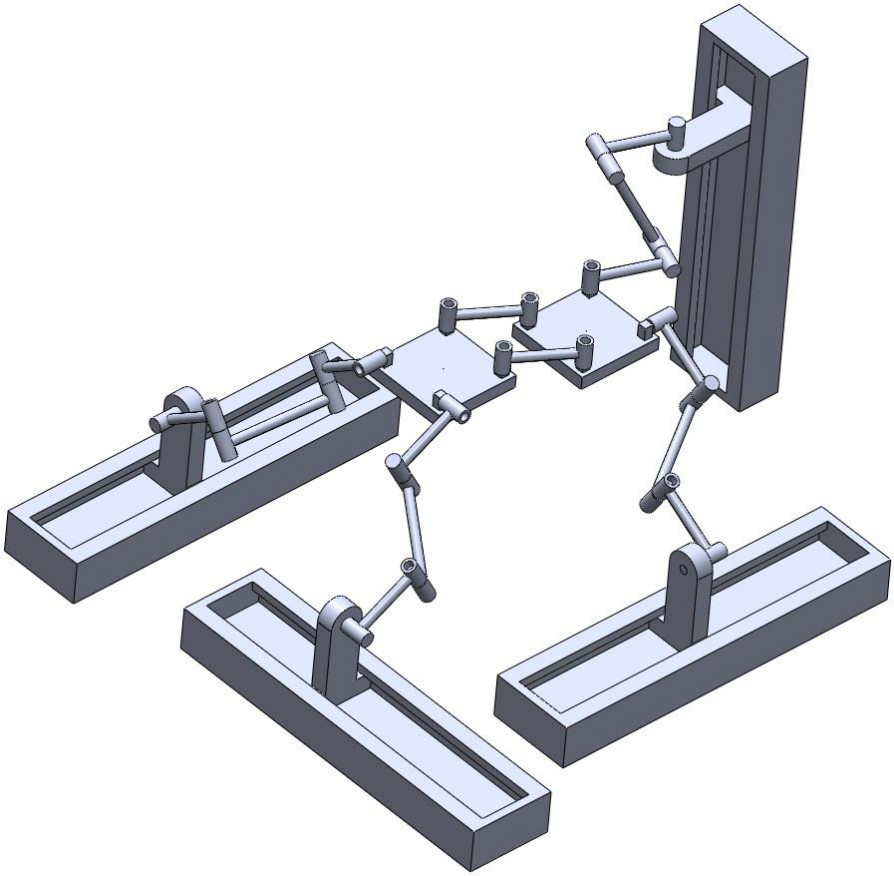

Figure 3.9: 4-DOF Parallel Mechanism Design 2 with parallelogram moving platform kinematic chain.

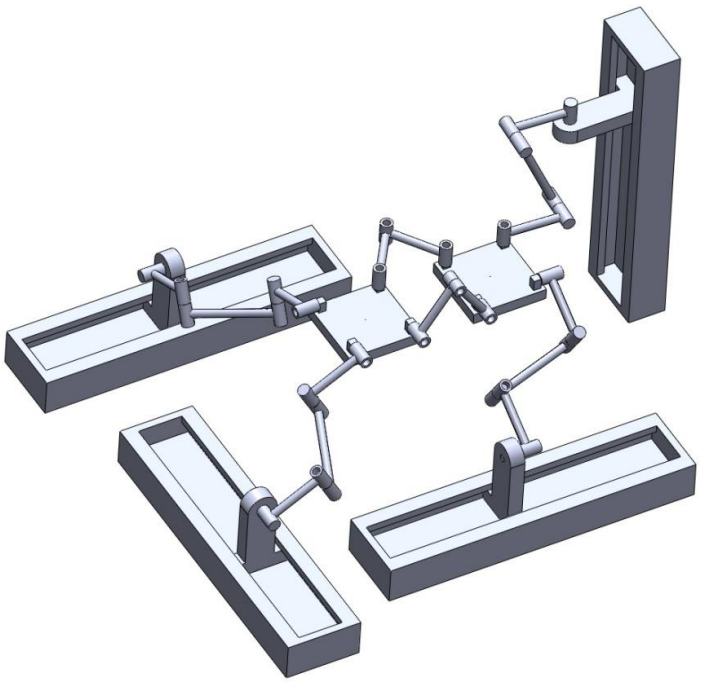

Figure 3.10: 4-DOF Parallel Mechanism Design 2 with dual *RRR* moving platform link.

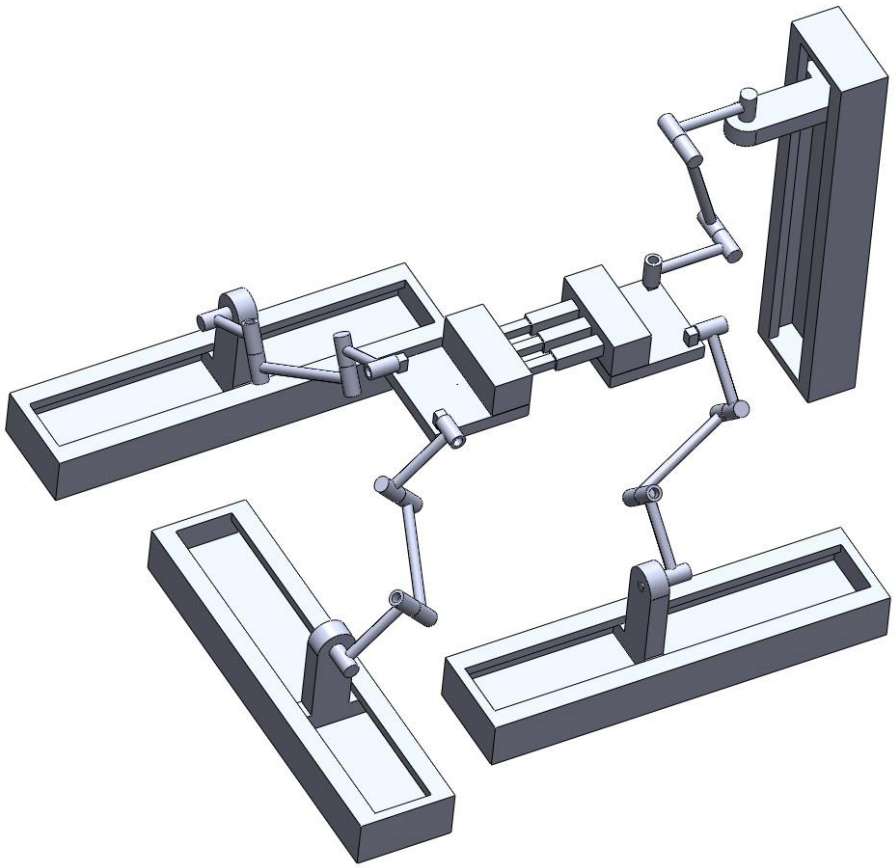

Figure 3.11: 4-DOF Parallel Mechanism 2 Design 2 with parallel 3P joint moving platform link.

The P joints of each limb are once again chosen as the actuated joints as they are the only joints that fulfil the criteria of set out in Chapter 2.

3.6 5-DOF Parallel Manipulators

This section investigates 5-DOF designs that have a combination of three translational DOF and two rDOF. Again using the extension to the type synthesis of a mechanism, the virtual chain pair that fits the desired motion pattern for the proposed mechanism is a pair of PPP virtual chains. The constraint couples formed by the moving platform link in this section will limit the motion of the moving platform to a pair of rDOF. This means that the proposed mechanisms will require only R joint links to produce the required motion.

As stated in section 3.4, the moving platform link designs are split into 1-rDOF and 2-rDOF in order to produce the desired motion of the moving platforms. In this section, the following 2-rDOF moving platform link designs will be investigated:

- A single RRR chain (figure 3.12).
- A pair of RRR chains with either an R or P joint between the centre R joints to allow for a greater range in motion of the two rDOF produced (figure 3.13).

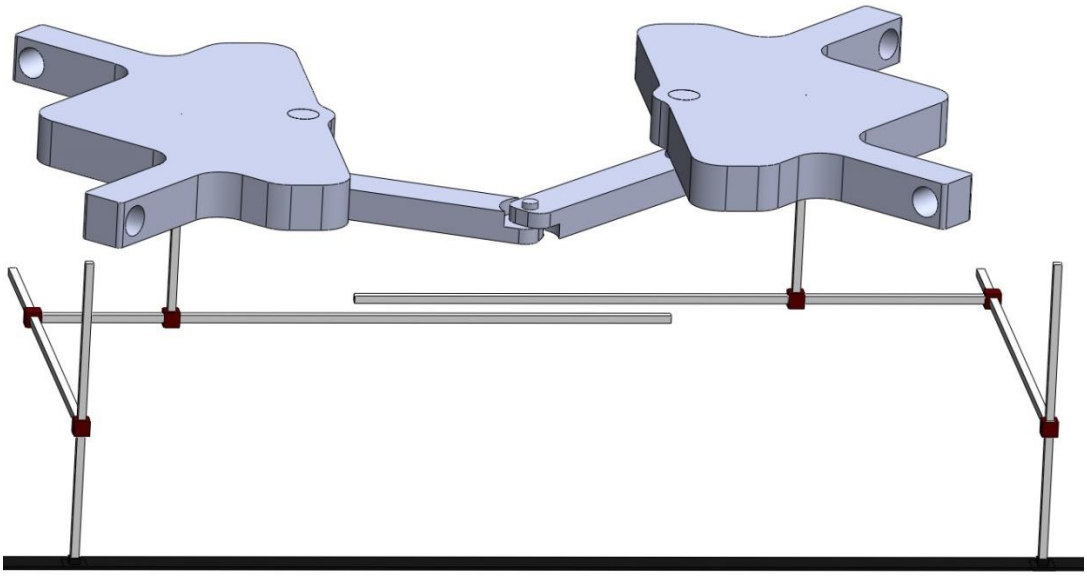

Figure 3.12: Single-loop 2-rDOF kinematic chain with an RRR moving platform link.

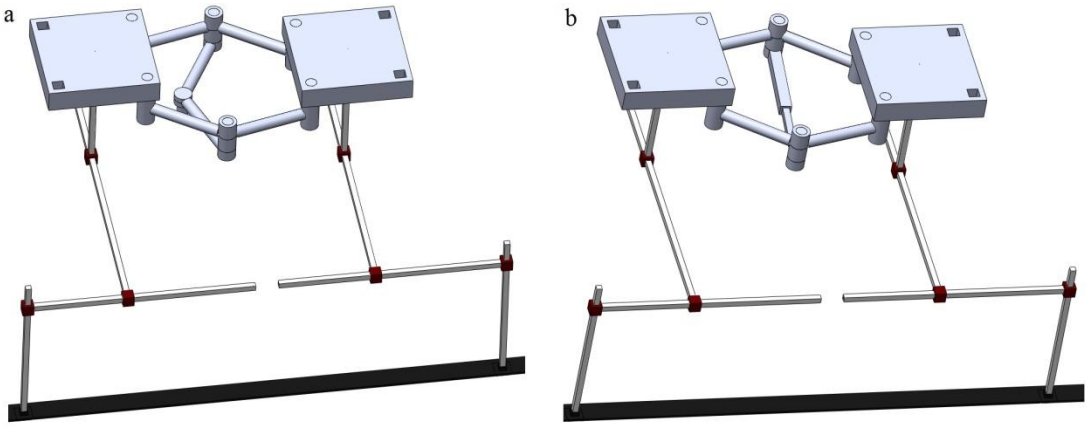

Figure 3.13: Single-loop 2-rDOF kinematic chain with a dual RRR moving platform and a) Central RRR moving platform link. b) Central RPR moving platform link.

Again using the method laid out in Chapter 2, the type synthesis of the kinematic chains of the mechanism can be determined. As the 5-DOF mechanism will include 2-rDOF there will be a need to investigate a mechanism in which one or more moving platforms have three limbs attached to it. The table for three limbed moving platforms is shown in table 3.2.

From tables 3.1 and 3.2, a series of potential kinematic chains can be constructed to produce the desired motion pattern. Some of these kinematic chains are detailed in the following sub-sections.

Table 3.2: Combinations of kinematic chains for three legged platforms.

m	c	Δ	c^1	c^2	c^3
3	2	6	3	3	2
		5	3	3	1
			3	2	2
		4	3	3	0
			3	2	1
		3	3	2	0
			3	1	1
		2	3	1	0
			2	2	0
			2	1	1
		1	3	0	0
			2	1	0
			1	1	1
		0	2	0	0
			1	1	0
3	3	6	3	3	3
		5	3	3	2
		4	3	3	1
			3	2	2
		3	3	3	0
			3	2	1
			2	2	2
		2	3	2	0
			3	1	1
			2	2	1
		1	3	1	0
			2	2	0
			2	1	1
		0	3	0	0
			2	1	0
			1	1	1

3.6.1 5-DOF Parallel Mechanism Design 1

The first 5-DOF design incorporates a moving platform with three limbs attaching it to the base and a moving platform with two. Using the information displayed in table 3.2, the type of kinematic chains available to allow the moving platform to operate in its desired motion pattern can be discerned. The kinematic chain layout for this design is a set of three kinematic chains with a connectivity of three DOF. The format for the kinematic chains is a $PRRR$ layout with the same joint alignment as the second 4-DOF design. The positioning of the kinematic chains is as follows: the first two kinematic chains connect directly to the moving platform and the third connecting to the centre of the moving platform connection link. The joint map for this mechanism is shown in figure 3.14.

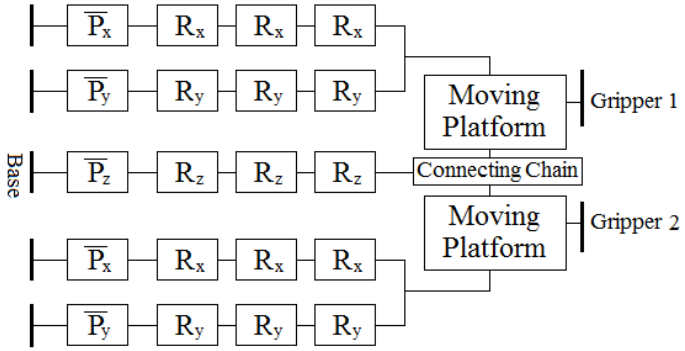

Figure 3.14: Joint map for 5-DOF Parallel Mechanism Design 1.

For this design, the moving platforms displayed in figure 3.13 are considered. The resulting mechanisms are displayed in figures 3.15 and 3.16.

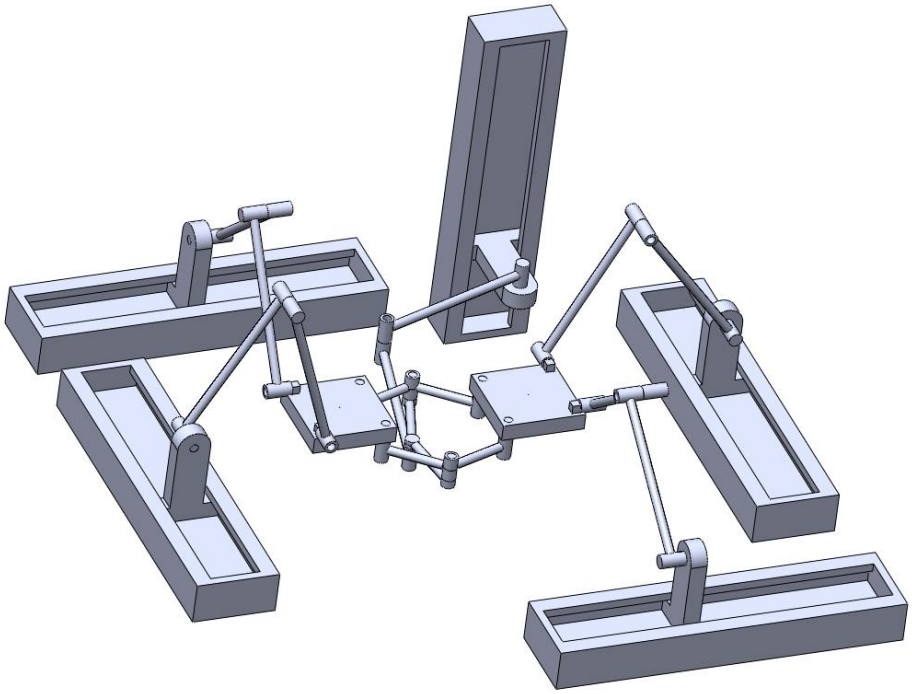

Figure 3.15: 5-DOF Parallel Mechanism Design 1 assembly with *RRR* moving platform link

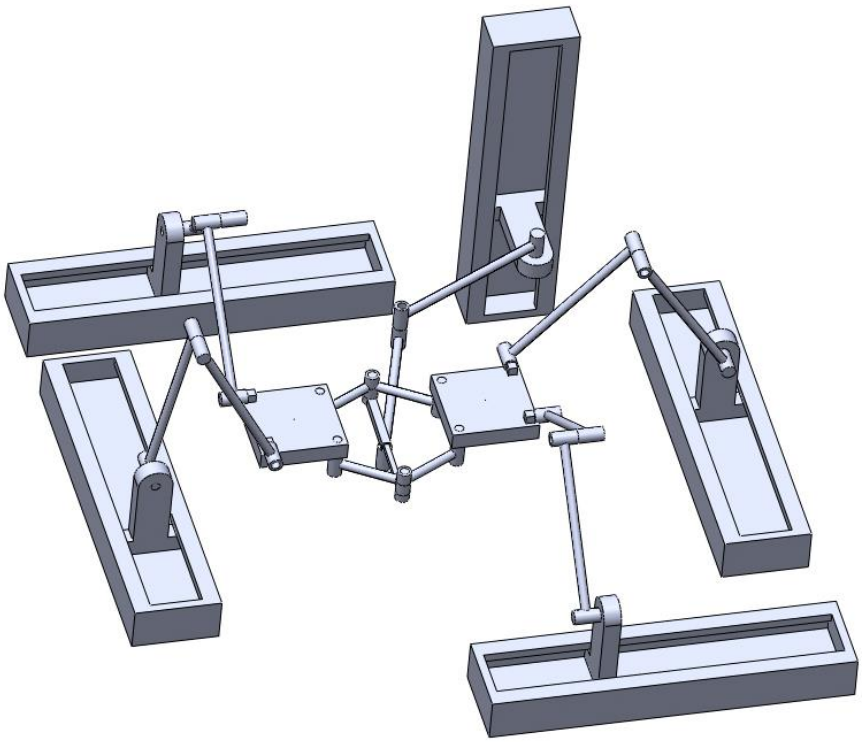

Figure 3.16: 5-DOF Parallel Mechanism Design 1 assembly with RPR moving platform link

The rules for actuating the joints of a parallel mechanism show that the mechanism shown in figure 3.15 is the most suitable moving platform design as the P joint in the connecting link would need to be actuated in order to improve the control of the mechanism.

3.6.2 5-DOF Parallel Mechanism Design 2

The second 5-DOF design investigates a mechanism that has three legs connecting each platform to the base of the mechanism. This time by only using the information displayed in table 3.2, the type of kinematic chains available for the desired motion pattern can be determined. Once again, the kinematic chain layout for this design is a set of three kinematic chains with a connectivity of three DOF. The format for the kinematic chains is a $PRRR$ layout with the same joint alignment as the second 5-DOF design. In this mechanism, the three kinematic chains will all be mounted to the moving platforms of the mechanism, with the kinematic chains operating along the horizontal axes mounted to the top face of the moving platform and the vertical kinematic chains attaching the bottom of the moving platforms. The joint map for this mechanism is shown in figure 3.17.

For the final design, all of the moving platforms detailed at the start of this section are considered. The resulting mechanisms are displayed in figures 3.18, 3.19, and 3.20.

Again the rules for actuating the joints of a parallel mechanism show that the either the mechanism shown in figure 3.18 or figure 3.19 is the most suitable moving platform designs as the P joint in the connecting link of figure 3.20 would need to be actuated in order to keep to the rules of actuation.

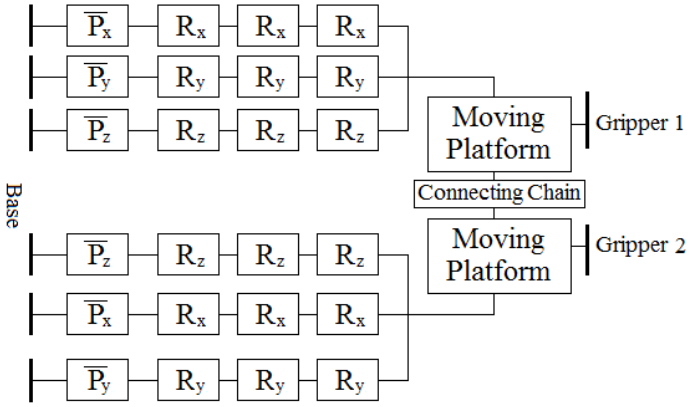

Figure 3.17: Joint map for 5-DOF Parallel Mechanism Design 2.

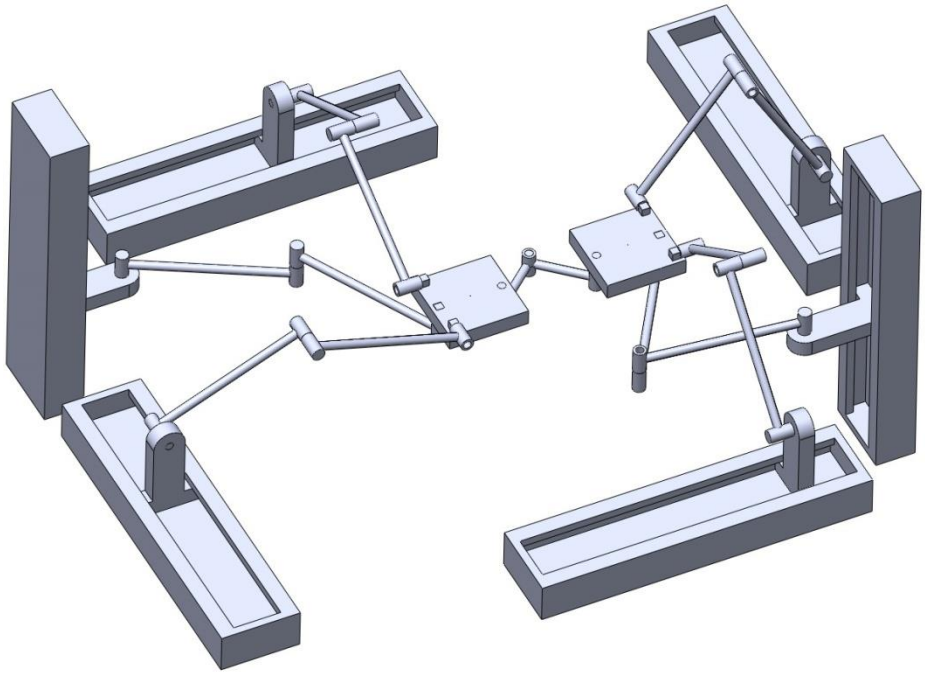

Figure 3.18: 5-DOF Parallel Mechanism Design 2 assembly with single *RRR* moving platform link.

3.7 Variations on Existing Parallel Mechanisms

The mechanisms laid out in section 3.6 are multi-platform mechanisms completely generated from the extended type synthesis method. While this is the preferred method of developing a mechanism, it is possible to take a pre-existing mechanism and apply the extended type synthesis method in order to alter the moving platform to become a multi-platform parallel mechanism. In this section the Delta robot will be shown with a 1-rDOF and 2-rDOF multi-platform variation.

3.7.1 4-DOF multi-platform Delta robot

The 4-DOF multi-platform Delta robot design detailed in this section incorporates the radial rDOF and Sarrus mechanism connecting links. The modified joint map of the Delta robot is displayed below in figure 3.21 with the CAD assemblies displayed in figures 3.22 and 3.23 for the radial and Sarrus connecting links respectively.

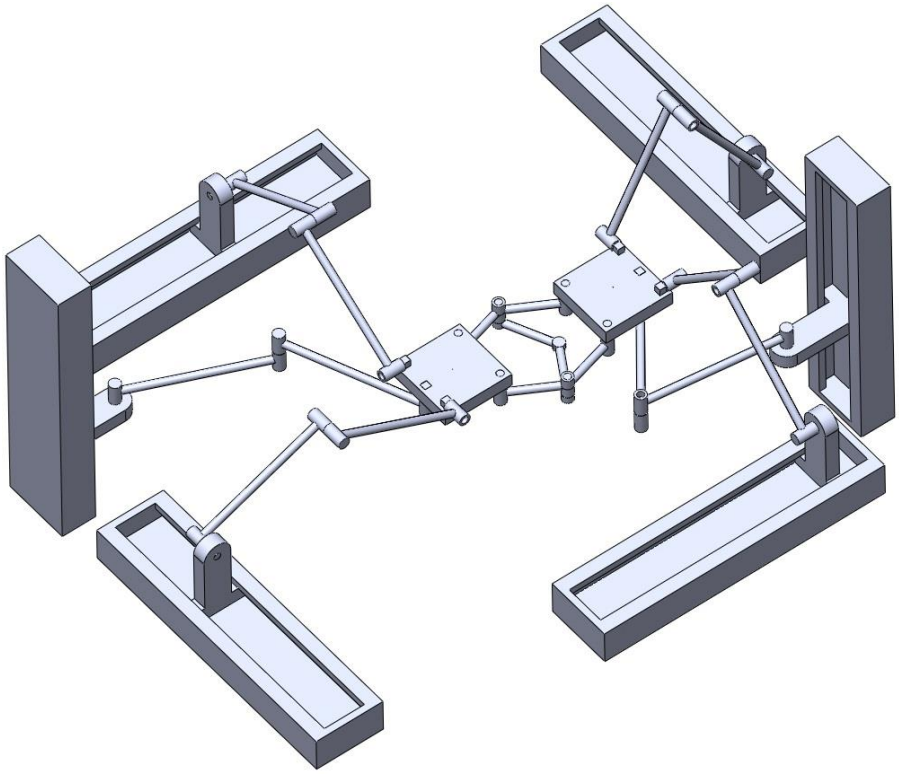

Figure 3.19: 5-DOF Parallel Mechanism Design assembly with dual RRR moving platform link and RRR central link.

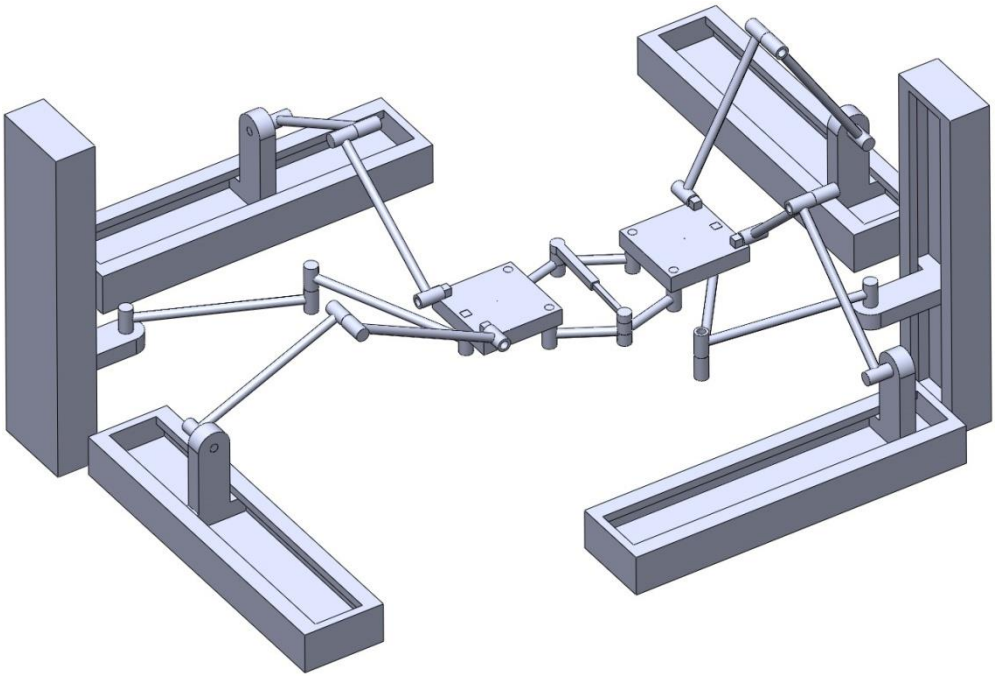

Figure 3.20: 5-DOF Parallel Mechanism Design 2 assembly with dual RRR moving platform link and RPR central link.

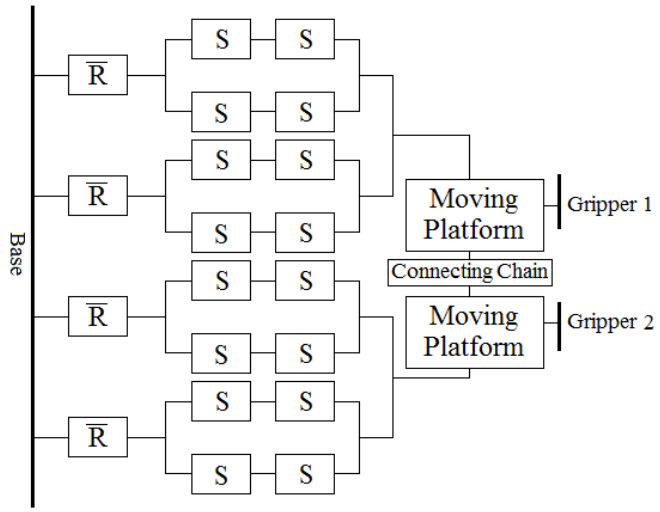

Figure 3.21: Joint map of Design 1.

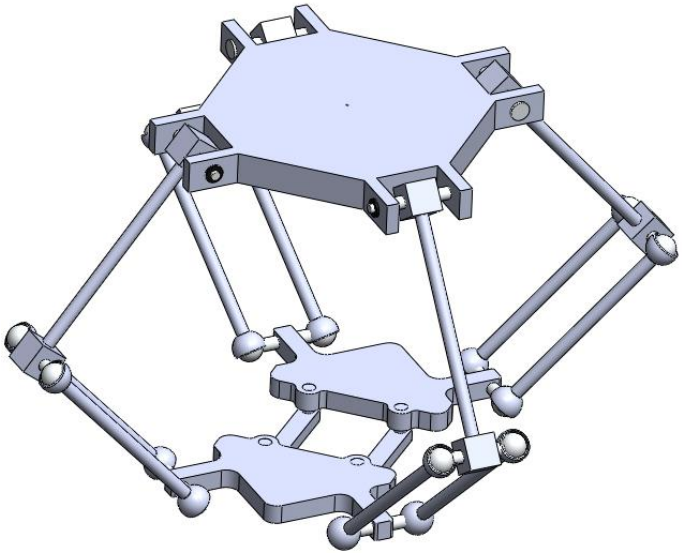

Figure 3.22: 4-DOF Parallel Mechanism Design 1 with radial rDOF moving platform link

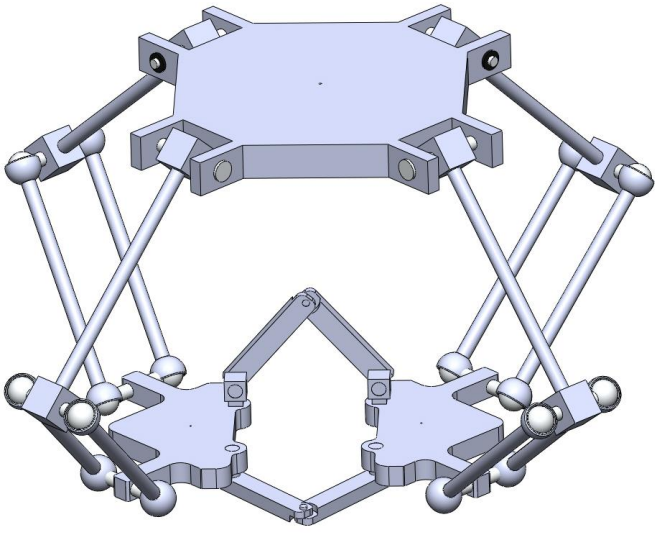

Figure 3.23: 4-DOF Parallel Mechanism Design 1 with dual *RRR* moving platform link.

The modified Delta robot designs are easily capable of the desired motion pattern specified in the design brief and, from publications on the standard Delta robot, the limitations of the architecture for the mechanism is well documented. The main drawback to this type of multi-platform mechanism is that each of the actuated joints on the mechanism needs to be engaged for any motion of either moving platform.

3.7.2 5-DOF multi-platform Delta robot

The moving platform selected for the 5-DOF multi-platform design of the Delta robot utilises the first 2-rDOF connecting link design displayed in figure 3.12; the joint map for the full mechanism is displayed in figure 3.24 and the assembled mechanism is displayed in figure 3.25.

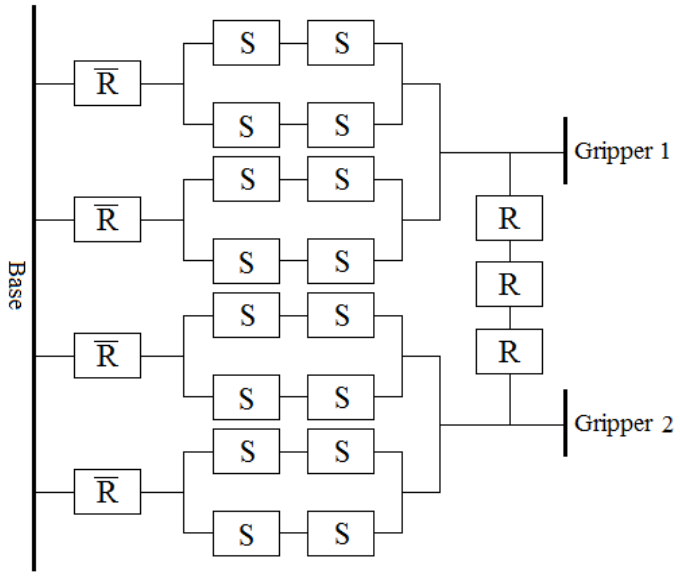

Figure 3.24: Mechanism's joint map.

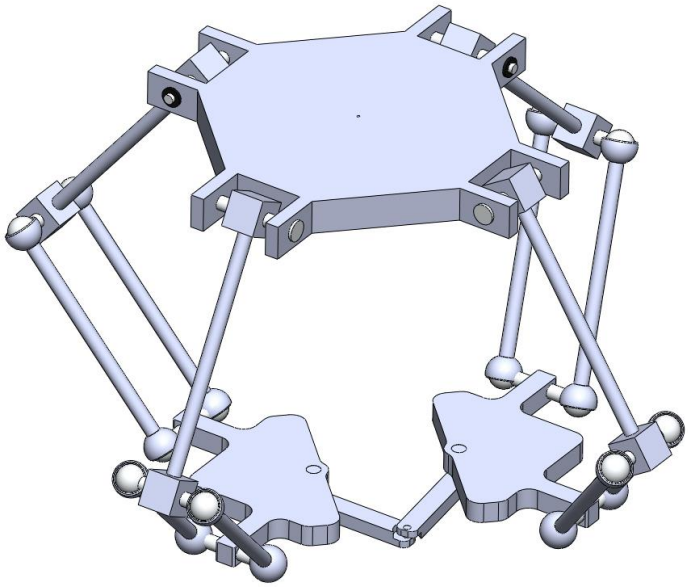

Figure 3.25: 5-DOF Parallel Mechanism Design 1 with single *RRR* moving platform link.

When testing the motion of the 5-DOF multi-platform design numerous stability and control problems were highlighted. The most obvious of these is attributed to the *RRR*

moving platform connecting link that would either move into a singular configuration (figure 3.26a) when the plates were moved to the maximum relative distance or allow the plates to rotate uncontrollably about the z axis (figure 3.26b).

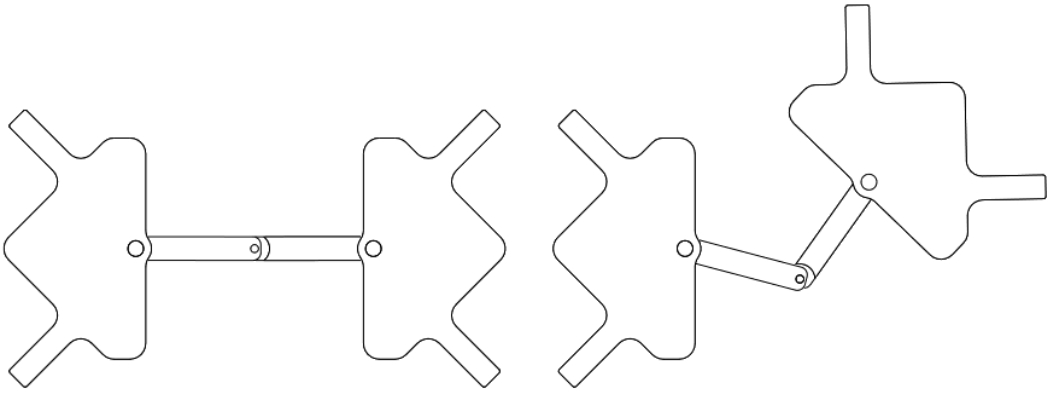

Figure 3.26: Concept moving platform a) in singular configuration; b) experiencing unwanted rotation.

In order to rectify these, the design was altered to include a second RRR link in parallel with the first between the moving platforms to produce a parallelogram; the intention being to improve the stability of the link as well as preventing the rotation of the platforms. Finally, a limit was placed on the central R joint in order to prevent the mechanism from entering the singular configuration (figure 3.27).

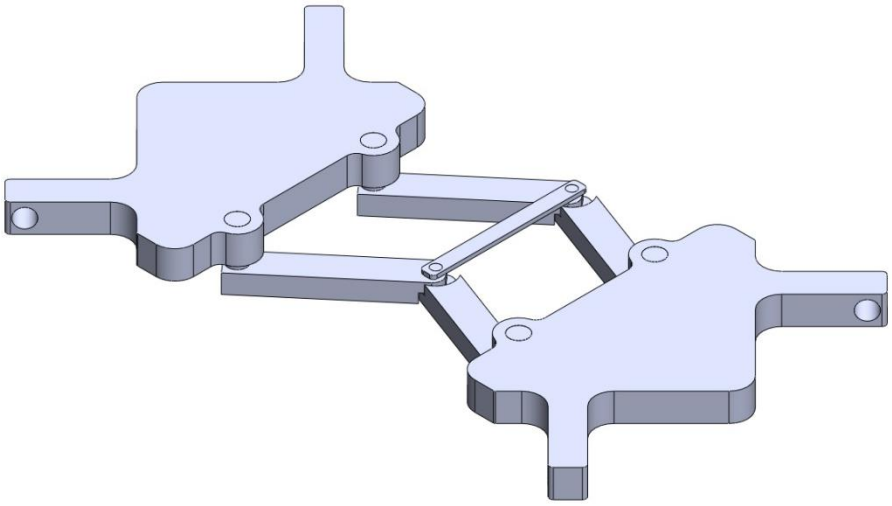

Figure 3.27: Moving platform with modified connecting link design.

Unlike the 4-DOF variation of the Delta robot, the 5-DOF multi-platform design only requires two of the four actuated joints to be engaged when making a single platform move as the other two can be locked to prevent the movement of the other platform. Since the main purpose of a multi-platform mechanism is to be more efficient than single-platform counterparts then this is vital, as it would effectively reduce the required power output by half.

3.8 Comparison of Parallel Mechanisms

The designs laid out in this chapter represent the basic potential design for a prototype mechanism. In order to determine the best mechanism for further development, each

mechanism must be compared against the other for mobility, ease of actuation and control stability. This section will initially look at the differences between the 4-DOF and 5-DOF designs to determine the best of each archetype, and subsequently the best of each will be compared to determine which mechanism will be taken forward for further research.

3.8.1 4-DOF Parallel Mechanism designs

While both of the 4-DOF designs in section 3.5 are capable of meeting the desired motion pattern set out by the dual *PPP* virtual chains, the initial design for the 4-DOF system can be considered a simpler option. This is because the kinematic chains of the first design are less likely to collide with the others or the moving platform during motion.

Comparing the first design with the Delta robot version shows that while again both mechanisms are capable of achieving the desired motion pattern, the first 4-DOF design is more desirable due to its simplicity in the actuation of the mechanism.

3.8.2 5-DOF Parallel Mechanism designs

The first 5-DOF design (figure 3.15) showed promise as it is capable of achieving the desired motion pattern for the mechanism; however due to the positioning of the vertical kinematic chain a high amount of stress could be produced on the moving platform connecting links during motion to the point that it could potentially limit the available motion of the mechanism.

The second 5-DOF design (figure 3.18) greatly improved the mechanism's stability, as the pair of vertical kinematic chains would share the load, allowing minimal deformation in the system. The only drawback is that the vertical kinematic chains would have to operate at exactly the same rate as any deviation in the motion speeds of the P joints would result in unwanted tilting of the moving platforms.

For the Delta robot variant of the 5-DOF mechanism (figure 3.25), the actuation causes several problems due to multiple actuators being required for any motion of the moving platform. Therefore, compared to the other 5-DOF mechanisms, the Delta robot variant can be considered unsuitable.

3.9 Summary

In this chapter, the method of using virtual chains to determine the motion pattern of a parallel mechanism for performing type synthesis to develop a series of parallel mechanisms is expanded upon to include the methodology for developing a class of parallel mechanisms with multiple moving platforms. The expanded methodology is then utilised to produce several designs of two parallel manipulator architecture types that are taken through the type synthesis process and developed into 3D CAD models using the SolidWorks CAD package. The designs are then given a multiple versions of multiple moving platforms in order to determine which mechanism best achieves the design criteria set out at the start of the chapter.

After comparing these mechanisms, the first design option was chosen as the best mechanism design for the 4-DOF case it will be easy to control with each actuated joint controlling a single DOF. For the 5-DOF case, the second mechanism proved the most stable and easily controlled as the vertical pair improved its structural stability.

The design being developed further will therefore be the second 4-DOF design, since the reduced cost of actuators and component parts will make the production of a physical prototype more feasible. Additionally, the issue of vertical actuation of the 5-DOF design would require either two actuators operating at precisely the same speeds or a gear chain between the actuator and both vertical actuated P joints in order to allow the mechanism to operate as intended.

In the next chapter, the concepts of multi-platform type synthesis and virtual chains are applied to a set of case studies in order to develop a viable visual representation of a parallel mechanism's workspace using a CAD software based approach.

Chapter 4 Workspace analysis

As described in Chapters 2 and 3, virtual chains are used to represent the desired motion pattern of a mechanism. This concept is adapted into a methodology that uses visual representation by means of 3D CAD software, in which a mechanism designed through type synthesis can be built with specified dimensions and simulated by utilising the virtual chain joints as the actuated joints for the mechanism itself. This approach differs from the traditional process, as rather than implementing an IKM and then determining the workspace through mathematical constraints in which link collisions are rarely incorporated, the simulation of the mechanism will produce a visual workspace of the total viable workspace in which no link collisions can occur.

4.1 Introduction

The main reason for investigating the workspace of a parallel mechanism is to determine the area or volume in which the mechanism can operate without over exerting itself, entering a singularity or colliding with itself or any other objects in the surrounding area. The workspace of a serial mechanism is also investigated to determine a safe distance from it at which people can operate without the risk of being hurt. The workspace of a parallel mechanism is usually enclosed within the external casing of the mechanism, thus already giving a visual area in which personnel are able to operate.

Traditionally the method for determining the workspace of a serial or parallel manipulator is to derive a set of equations that describes either the extremities or the permitted motion of the moving platform or allocated end-effector. These methods tend to use a set of equations that can be input into mathematical modelling and programming software packages, which have the ability to create 2D and 3D images of the manipulator workspace. This method of workspace analysis is well-documented [4.1-4.4] and, therefore, will not be investigated in this thesis. Recently, a new method of workspace analysis has been developed in which 3D CAD software is used to develop a virtual model of a proposed mechanism before producing a motion study in which the actuated joints of the mechanism are controlled to produce the workspace [4.5].

The work presented in this Chapter investigates the workspace of parallel mechanisms and suggests a new method of producing them with the assumption that all mechanisms designed in this chapter are statically stable and therefore affected by static singularities in which the actuators of the mechanism are locked but the moving platform is still capable of moving. A virtual chain is an imaginary kinematic limb extending either from the base of the mechanism or from an arbitrary point to the end effector of the mechanism. The VC is a virtual representation of the range of motion for the end-effector of the mechanism along and about the three world-coordinates axes (x, y and z). The virtual chain is only represented in a virtual environment and is not intended to be part of the mechanism, in this sense any collisions or interferences of the VC from the simulation restraints of the mechanism can be removed, allowing for the production of a full workspace.

The VC approach is then applied to a series of three case studies with the intent of proving the validity of the process on both spatial and planar mechanisms before being implemented on the chosen prototype. The workspaces formed by the virtual chain

approach will be compared to previously published results from experts in the field to ensure the method is sound. Following this chapter, the virtual chain approach will be applied to the finalised prototype designs in Chapter 7.

4.2 Virtual Chain Approach

This section investigates the method in which components of a virtual chain are produced in CAD software and how it can be constructed and then implemented into an existing Parallel Mechanism CAD model in order to produce the motion of the mechanism.

A VC is produced by modelling two types of joints in CAD and arranging them in various configurations suited to the motion of the mechanism being investigated. The representations of the two main types of joints, *R* and *P* joints, are now given:

- *P* joints are represented in the form of two cuboid extrusions. The outer casing of the joint being the larger of the two and being hollowed out to allow the insertion of the prismatic yoke.
- *R* joints are composed of two cylinders with the same radius connected together by a variable radius.

Modelling the required virtual joints is relatively straightforward as they are designed to be simple shapes and can be bulky as they will not be included when investigating the collisions of the manipulator's components during simulation. The procedure for developing the VC is detailed below, the steps being written using terminology used in the production of SolidWorks parts and assemblies:

1. Create a model of the parallel manipulator to investigate.
2. Model the required virtual joints and links in order to add to the mechanisms model an appropriate virtual chain. The virtual chain could be selected according to [4.6] for a parallel mechanism with a serial virtual chain or it can be selected as a 6-DOF virtual chain for a parallel mechanism with a parallel virtual chain.
3. Determine the location of motion to control, usually the moving platform of the manipulator.
4. Using a standard mate, attach the first joint of the virtual chain to either the point of motion or the origin/world coordinates of the robot and align with the desired axis.
5. Continue to attach the virtual joints to the previous virtual joint until the VC is complete.
6. Once the VC and mechanism are in their initial positions for simulation, right click the end of the furthermost *P* joint yoke and select the fix option, preventing it from moving during simulated motion.

The virtual chain can be constructed into numerous forms (figure 4.1) with multiple joints being either coupled with another, individually driven or fixed in place in order to investigate movement when an axis of motion is denied to the mechanism.

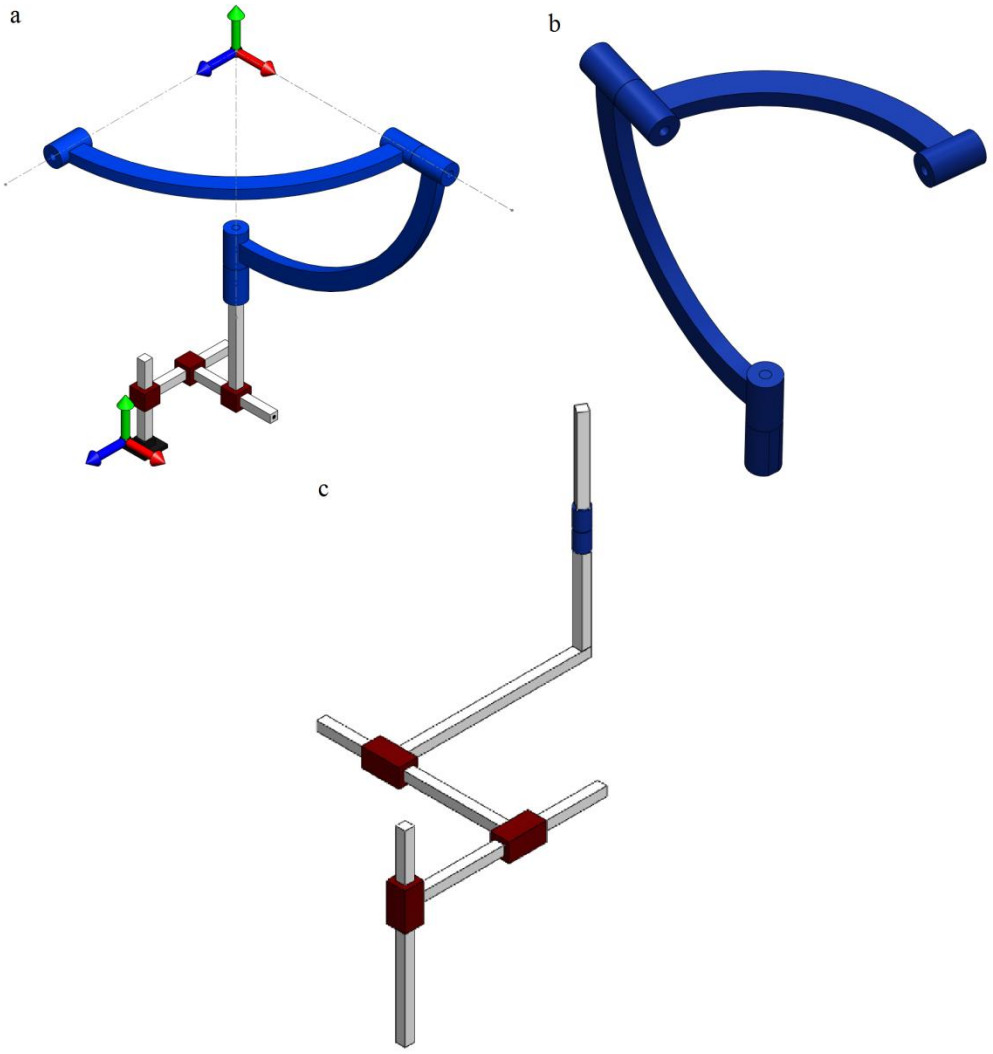

Figure 4.1: A selection of virtual chains constructed from virtual *P* and *R* joints: a) 6-DOF *PPPRRR* virtual chain; b) 3-DOF *RRR* virtual chain; c) 4-DOF *PPPR* virtual chain.

4.2.1 Virtual chain force method

With the virtual chain integrated into the mechanism CAD model, the workspace of the mechanism can then be investigated. The procedure for producing the 2D and 3D workspaces are essentially the same, with an additional few steps for the 3D workspace. For the initial investigation of a mechanism's workspace, a planar mechanism with a *PPR* virtual chain (figure 4.2) is used to illustrate the method.

After the component parts of the mechanism and virtual chain have been modelled, they are assembled into an assembly file using standard mates. Once the mechanism is built, the virtual chain joints and links are added to finish the mechanism. With the mechanism complete, if the assembly file has no premade motion study in the motion manager bar at the bottom of the screen (figure 4.3), select the "Motion Study" option in the toolbar ribbon (figure 4.4).

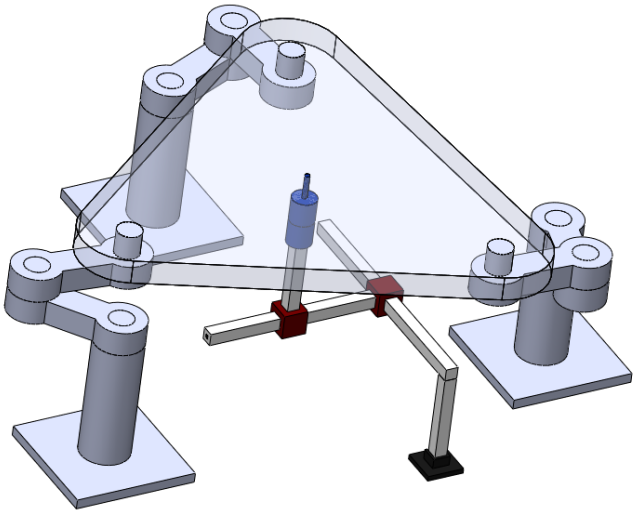

Figure 4.2: Planar parallel mechanism with a *PPR* virtual chain.

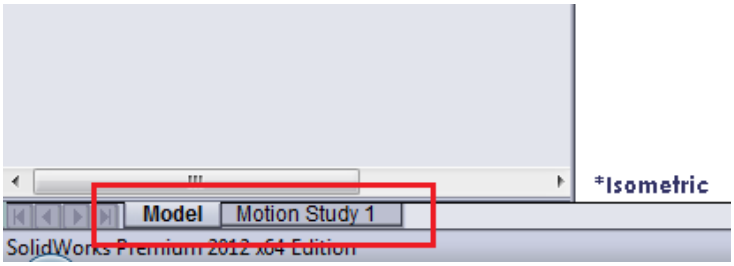

Figure 4.3: SolidWorks motion manager selection bar.

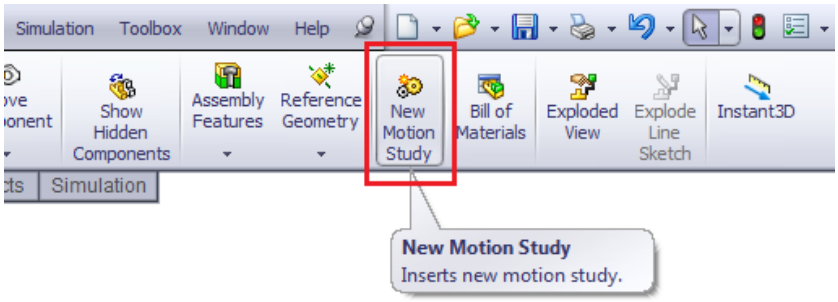

Figure 4.4: Motion Study button on SolidWorks toolbar.

Selecting the new motion study in the Motion Manager selection bar, the motion study type is modified from "Analysis" to "Motion Analysis" (figure 4.5). This option will only be available if the SolidWorks Motion add-on is installed and enabled for the document.

The next step of the process is to assign contact pairs to the links of the mechanism in order to prevent any unwanted motion. The contact pairs can either be assigned by first selecting the components that should not come into contact and then selecting the contact option in the Motion Manager toolbar (figure 4.6) or to select the Contact option on the toolbar and then select the components once the Contact Pairs option window opens (figure 4.7).

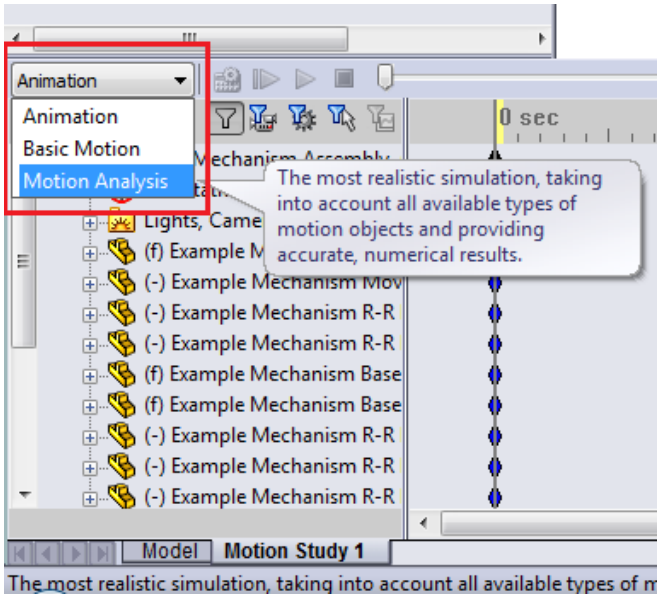

Figure 4.5: Motion study type dropdown window.

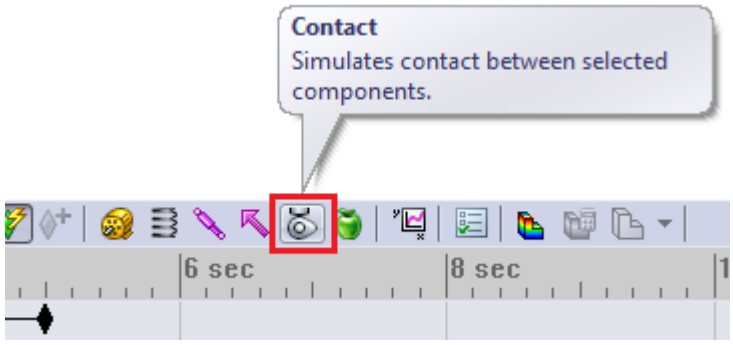

Figure 4.6: Contact option on the Motion Manager Toolbar.

Components that should not be included in contact pairs include the entire virtual chain, as they should not affect the motion of the mechanism other than to drive it and any components that are constantly in contact with each other, i.e. the pin of an R joint and its sockets. Multiple sets of contact pairs can be assigned to cover the full extent of potential impacts occurring during simulation as each pair will be looked at before motion rendering will take per frame.

In order to prevent the presence of singular configurations in the kinematic chains of the mechanism "Limited Angle Mates" can be used. This will prevent singularities such as a collinear configuration of a planar serial *RRR* sub-chain. To add limited angle mates, the "Mate" option is first selected on the SolidWorks toolbar (figure 4.8).

In the window that opens, "Advanced Mate" option (figure 4.9) is selected which opens a new set of mate options that allow for ranges of motion, specified motion paths, and various other methods of limiting the available motion between two components of the mechanism. Selecting the advanced mate angle option produces two additional field input boxes that allow a maximum and minimum angle to be specified while the initial input box indicates the current angle of the joint at the time the components were selected (figure 4.10).

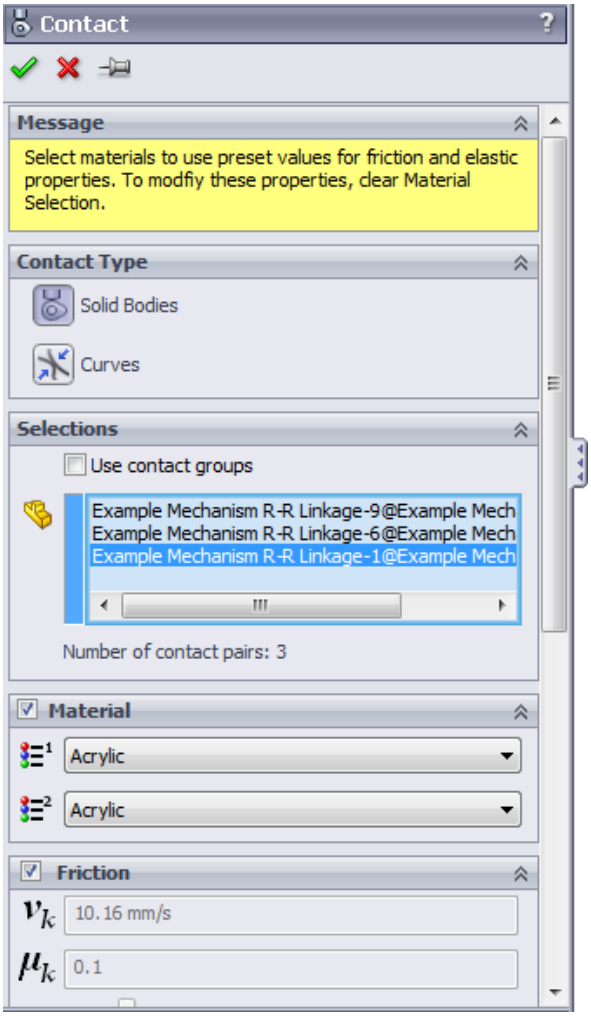

Figure 4.7: Contact option window with example selection

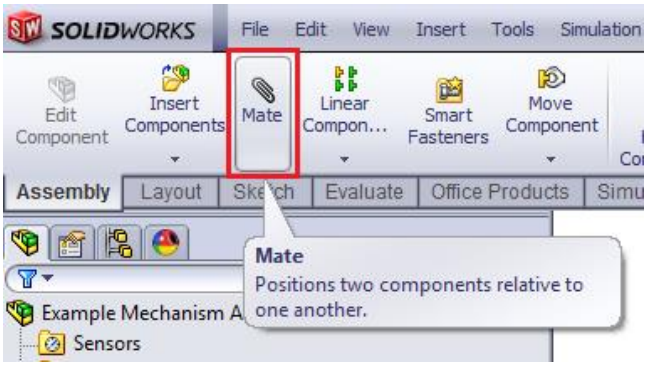

Figure 4.8: Mate button on SolidWorks toolbar.

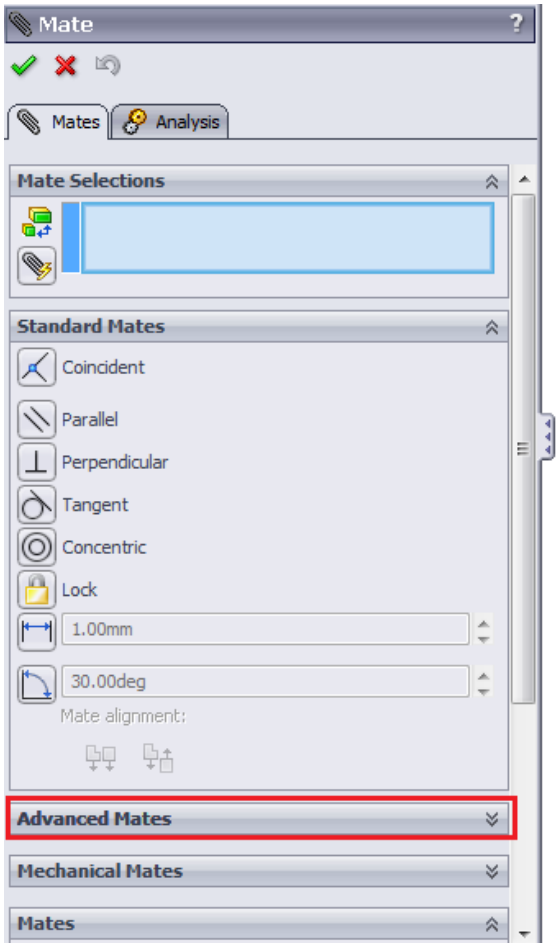

Figure 4.9: Mate builder window with Advanced Mates option highlighted.

With the kinematic links setup for collision detection and singular configuration avoidance, the next step is to add the motion drivers for the simulation. This is done by first determining what kind of workspace is desired. For 2D workspaces the virtual joint aligned with the axis not being investigated should be fixed in place by right clicking on the component and selecting the "Fix" option. This will prevent the component from moving during simulation regardless of any forces acting upon it.

For a 3D workspace, a motor component is applied to the virtual joint assigned to have the least variance in motion, typically the virtual joint aligned along the vertical axis. To add a motor to the simulation, the "Motor" button located on the Motion Manager Toolbar (figure 4.11) is selected. This opens a new window in which the type of actuator, linear or rotation, can be selected (figure 4.12). Next the location of the motor needs to be selected along with the position and direction of the motion in which the component is being driven.

In the motor options window, the type of motion can be determined by altering the option in the motion dropdown window as highlighted in figure 4.12. Using this dropdown box, several motion options become available. In order to produce a user defined controlled motion, the option of "Data Points" should be selected. This opens a new function builder window that allows for time stamps and motion components to be input to the degree of accuracy required for the simulation (figure 4.13).

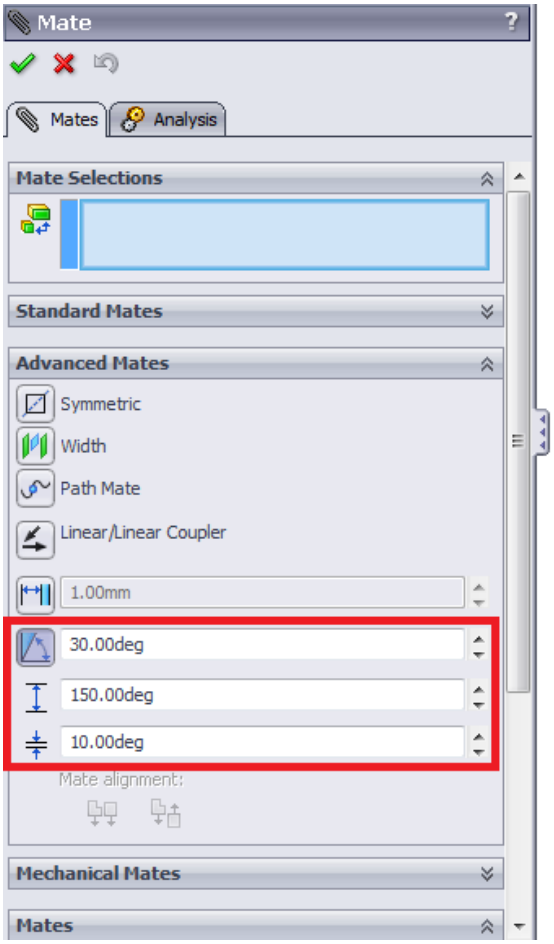

Figure 4.10: Advanced angle mate (A.K.A. Limited angle mate).

In order to produce the force method workspace of a mechanism in either 2D or 3D, the remaining virtual joints need to be actuated by force components, which apply a given force as either a point force or a torque on the chosen components in the desired direction. The "Force" button is likewise located on the Motion Manager Toolbar (figure 4.14) and produces a new window similar to that of the motor component (figure 4.14). As with the motor component, the method in which the force is applied can be specified by the user by selecting the Force Function dropdown box and selecting the "Data Points" option. This opens a similar function builder to figure 4.15 with the only exception being that it can only operate on Force values.

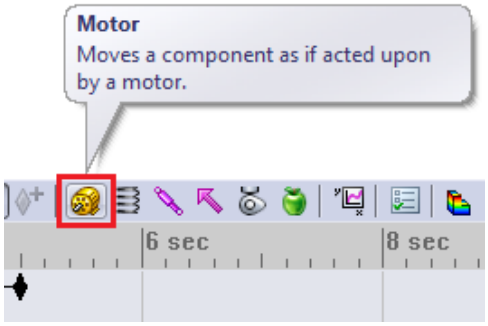

Figure 4.11: Motor option button on Motion Manager Toolbar.

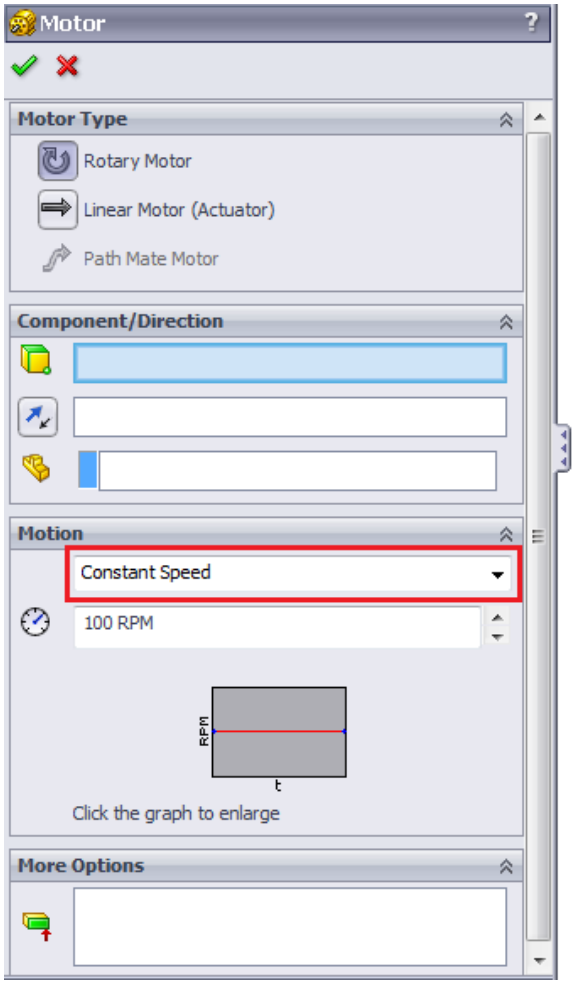

Figure 4.12: Motor option window.

The data points used for the motor component in a 3D force method workspace simulation need to begin at the lowest point that the mechanism can reach; this is typically set to zero by setting the mechanism to begin at the centre of its workspace at the lowest point before the simulation begins. This means that the motor component will only be required to move in a single direction in increments that suit the resolution of the 3D workspace desired.

The data points for the force/torque components for both a 2D and a 3D Force Method workspace simulation need to produce enough force in a positive and negative direction to complete a full rotation of the mechanism, which then needs to be repeated for each increment of the motor component in a 3D workspace.

With the motion of the mechanism finalised a plot tool needs to be assigned to the location being tracked. This is typically either the centre of the moving platform, the edges/corners of the moving platform or the location of the end-effector of the mechanism. The "Results and Plots" button is located on the Motion Manager Toolbar (figure 4.16) and when selected opens a new window in which numerous types of plots can be produced (figure 4.17).

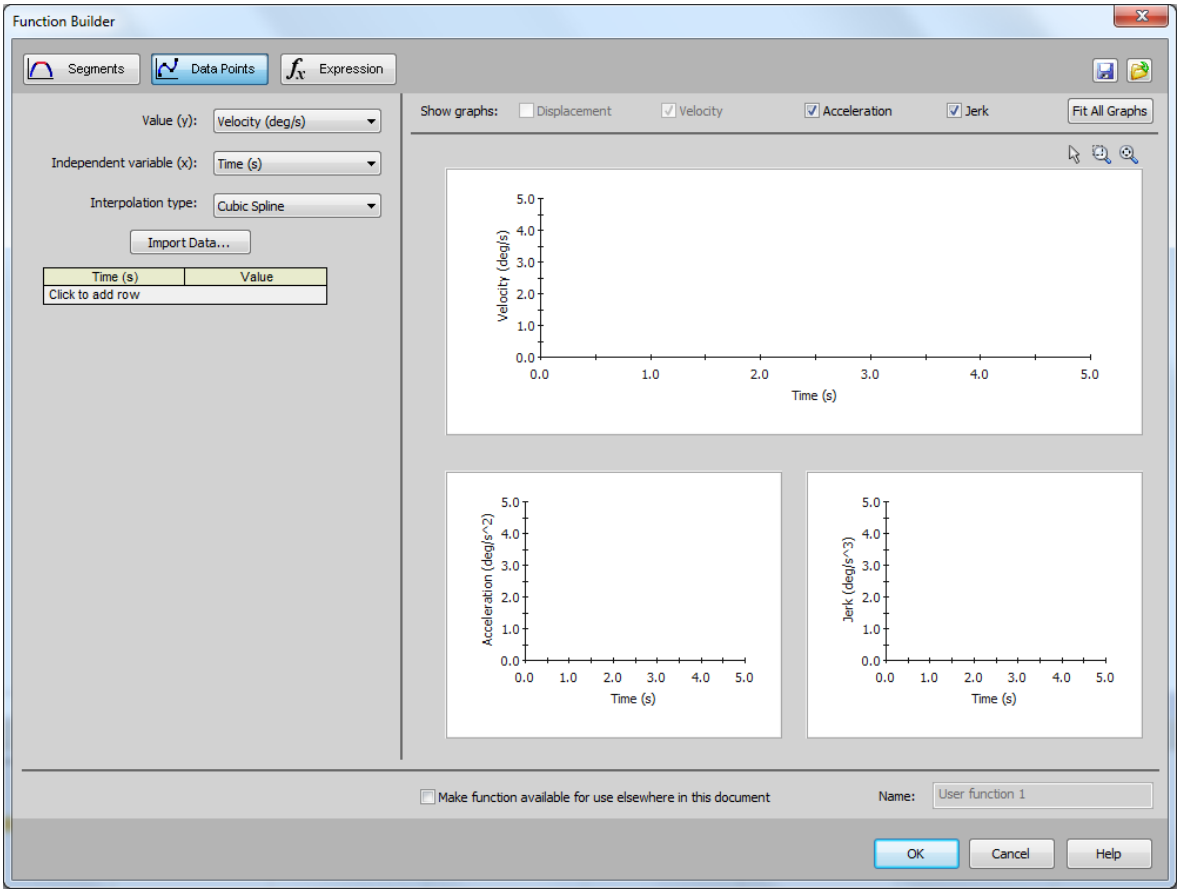

Figure 4.13: Motor Data Points Function Builder window.

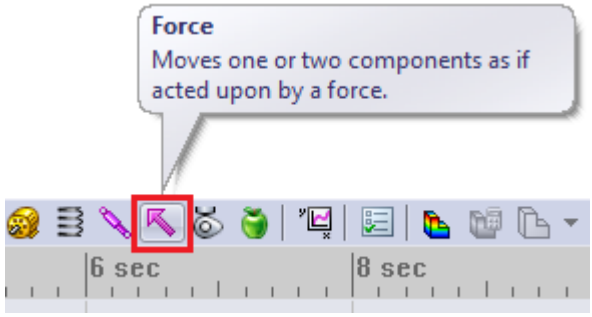

Figure 4.14: Force option button on the Motion Manager Toolbar.

The first dropdown box at the top of figure 4.17 determines the type of result desired at the end of the simulation. This dropdown box includes the values of the components displacement/velocity/acceleration, Force values acting on the mechanism in certain locations, Momentum/Energy/Power values at certain locations on the mechanism, as well as other forms of data. Utilising the Displacement/Velocity/Acceleration option, as shown in figure 4.17, the second dropdown box becomes active allowing displacement, velocity and acceleration values to be selected along individual axes. Included in this dropdown box is the option of "Trace Path" which takes one or more vertices of the simulated model and produces a black line tracing that point's path through the simulation. This option can also be used to give information on the positional data in order to refine the workspace.

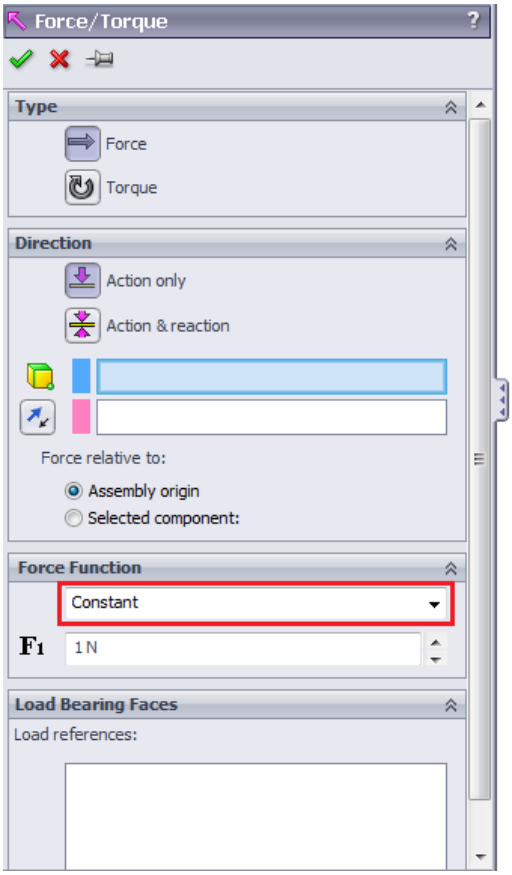

Figure 4.15: Force/Torque option window with Force Function dropdown box highlighted.

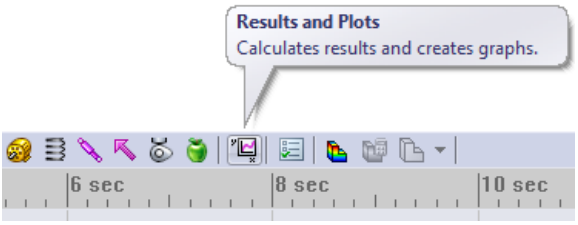

Figure 4.16: Results and Plots button on the Motion Manager Toolbar.

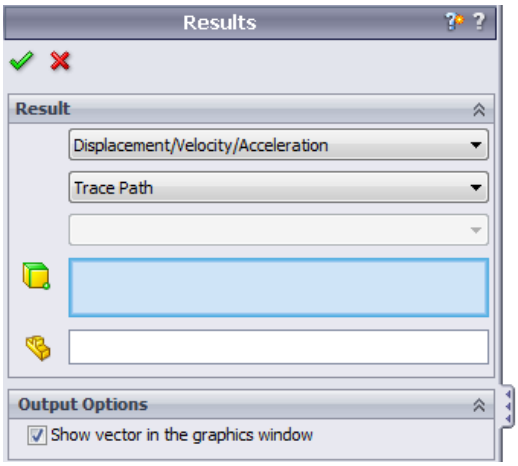

Figure 4.17: Results and Plots window.

Once the mechanism is complete, the final step is to press the simulation button. This is located at the right of the Motion Manager Toolbar next to the Motion Study Type dropdown box (figure 4.18). Once the simulation has completed, any sensor data, such as displacement values, forces, etc, will appear in individual windows that can be manipulated to gather the data required.

Figure 4.18: Simulation button on the Motion Manager Toolbar.

4.2.2 Virtual chain motor refinement method

For the initial workspace of the mechanism, the workspace is often crude and usually has several paths that have translated across the workspace due to joint configurations impeding certain motions during the simulation. To improve upon this method, the motor refinement method can be used. This method is setup in the same manner as the force method but with the motor components in place of the force/torque components and the data plots being redone based on information gained from the displacement sensors used in the force method simulation.

The motor components of the SolidWorks motion study tend to ignore limitations created by contact pairs that can compromise the simulation of the workspace. For this reason the workspace of the mechanism is reduced by a small amount, removing any outlying areas or sections of the workspace that will not be used during the life cycle of the mechanism.

The motor refinement method workspace requires the user to look at the results from the force method and produce a data point system that accurately replicates the discovered workspace but without the rough edges and intersecting lines. It should be noted that in the production of 3D workspaces this method requires that the mechanism be reset to the origin position before each increment of the original motor component from the force method.

4.3 Case Studies

To prove the validity of the method presented for the production of a 2D workspace in the previous section a series of case studies will be investigated and compared with pre-existing information. The mechanisms involved in these case studies are the 3-*RPS* mechanism, the spherical parallel mechanism [4.7], and the Delta robot.

4.3.1 3-*RPS* mechanism

The first case study looks at a 3-*RPS* parallel mechanism (see [4.6] [4.8-4.12] for examples). The mechanism has been designed (figure 4.19) with the lower base plate having an external diameter of 250mm, the upper moving platform having an external diameter of 130mm, the *RPS* links having a minimum length of 94.4mm extending to a maximum length of 150.5mm and raising the platform height being varied from 78mm to 147.75mm.

The VC of the mechanism is a 6DOF VC that consists of 3*P* joints representing the translational motion of the mechanism along the world x-, y- and z-axes (figure 4.1a) and

3*R* joints representing the rotational motion of the mechanism about the local x-, y- and z-axes.

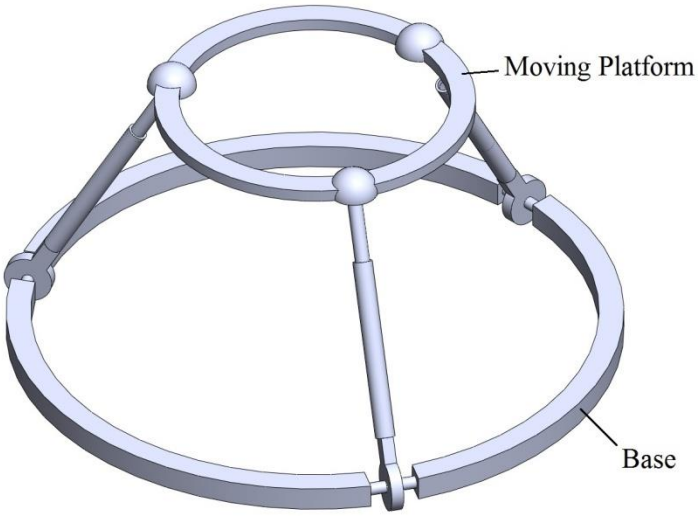

Figure 4.19: 3-*RPS* mechanism.

The central axis of the rotational VC is centred in the middle of the moving platform. While the translational VC is centred to the middle of the plate via a concentric mate with a hole on the moving platform, it is not coincident with the moving platform's plane allowing the VC to remain on the world axes while the moving platform rotates and moves (figure 4.20). The mechanism in figure 4.20 has 3-DOF; therefore three joints in the VC, including the one along the Z-axis, should be actuated when determining the workspace.

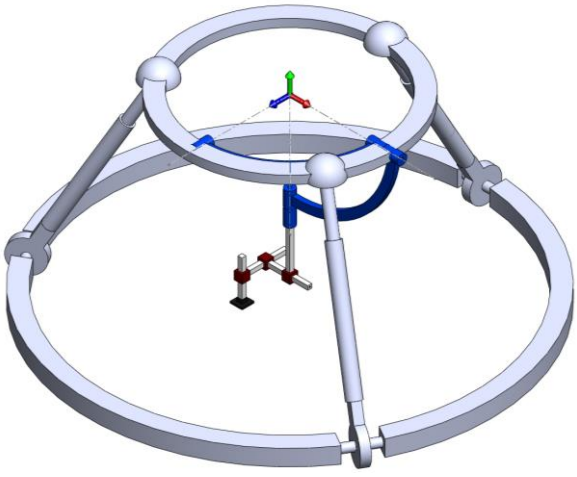

Figure 4.20: 3-*RPS* Mechanism complete with 6-DOF virtual chain.

For the simulation, one of the horizontal *P* joints on the all translational VCs are fixed in place allowing for the mechanism to be operated with torques applied to all but one of the horizontal *R* joints of the rotational VC. The point being traced in this case study is the highest point of the moving platform's vertical axis as shown in figure 4.20. The simulation was run multiple times while adjusting the vertical height of the moving

platform via the vertical *P* joint in the all-translational VC. The results of the simulations are shown in figure 4.21, where the black trace lines inside each section of the workspace are produced by the movement of platform when applying random torques to the *R* joints of the VC. This reduces simulation time while locating the external boundaries of the workspace of the mechanism.

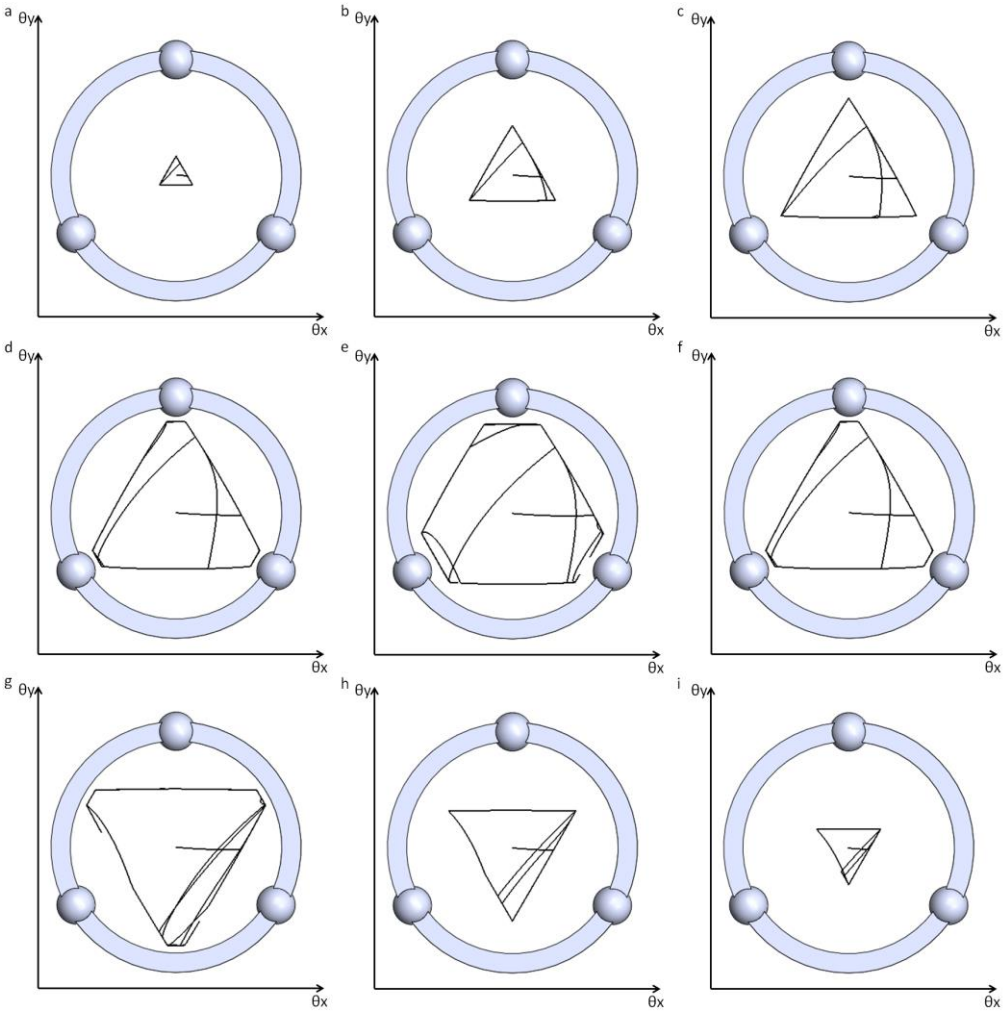

Figure 4.21: Horizontal workspace with a vertical height of: a) z = 142.5mm. b) z = 137.5mm. c) z = 132.5mm. d) z = 127.5mm. e) z = 122.5mm. f) z = 117.5mm. g) z = 112.5mm. h) z = 107.5mm. i) z = 102.5mm.

Comparing this to the workspace presented in figure 9 of [4.8], which utilised the traditional process of producing workspace equations from the inverse kinematic model and feeding these into Matlab, produced near identical results.

4.3.2 Spherical parallel mechanism

For the second case study, the mechanism studied is a spherical parallel mechanism [4.7] constructed from three kinematic chains of 3*R* joints each offset from the last by 120°, with the second *R* joint raised so that it will sit above the first *R* joint of the next kinematic chain. The last *R* joint is inverted so that, when the mechanism is in its starting position (figure 4.22a), the *R* joint is coaxial with the axis of the third kinematic chains first *R* joint and the second *R* joint of the second kinematic chain.

The spherical parallel mechanism is capable of producing an approximation of an actuated spherical joint about a single point where the axes of all three *R* joints of each chain meet. The virtual chain for this mechanism is in the form of three virtual *R* joints laid out, as shown in figure 4.1b and added to the mechanism in figure 4.22b.

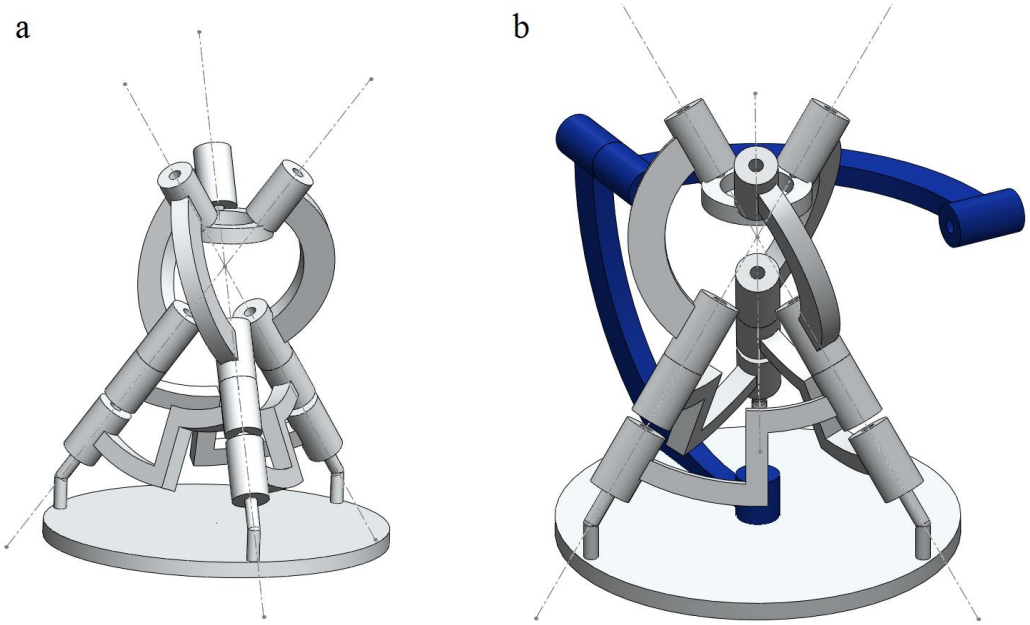

Figure 4.22: a) 3-*RRR* spherical robot with a common point of motion; b) spherical mechanism with virtual chain

To discover the available workspace, the force method is applied. A constant torque is applied to the virtual chain R joints and collision pairs assigned to the two link pairs separately to prevent the mechanism from seizing up since the middle R joint section of both links remain in contact throughout the simulation. The resulting workspace is displayed in figure 4.23 below.

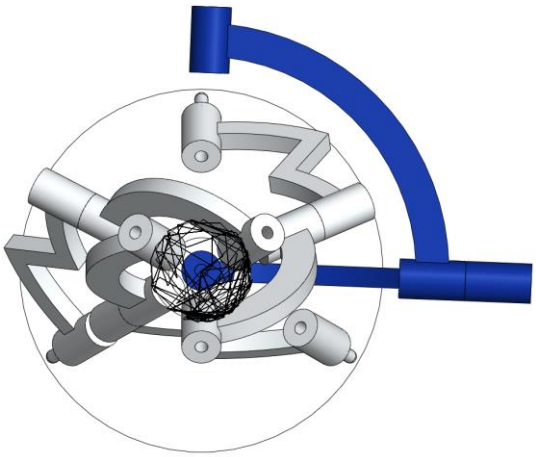

Figure 4.23: Workspace of the spherical parallel mechanism as produced via the force method.

In order to compare this mechanism to a workspace defined in an earlier publication, the motor method representation of the mechanism was developed to match that of the

workspace displayed in [4.7]. As this publication gives the workspace of the mechanism in terms of degrees of rotation, the motors attached to the virtual chain can be manipulated to match these values (figure 4.24).

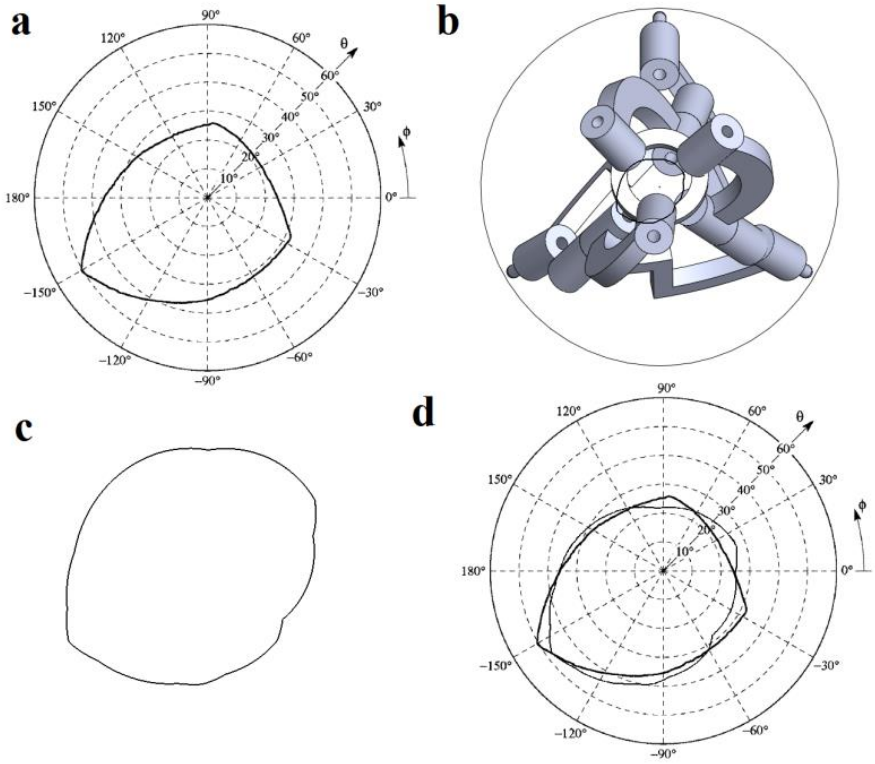

Figure 4.24: a) Predefined workspace of the spherical parallel [4.7]; b) Motor refinement method workspace of the spherical parallel with mechanism included; c) Motor refinement method workspace without mechanism; d) Comparison of the workspaces.

The resulting workspace from the motor refinement method shows that the methodology for discovering the workspace of the spherical parallel mechanism is almost exact, with only small deviations from the established workspace from [4.7] (figure 4.24d). These inconsistencies are most likely due to the inclusion of collision detection in the links which themselves have been designed in a different style to allow the link sizes to be easily modified.

4.3.3 Delta robot

For the final case study, the well-known Delta robot (section 1.2.2.) has the virtual chain 2D and 3D workspace methods applied and then compared against the theoretical data gained from [4.13] and [4.14]. The Delta robot in this case study has been designed with an equilateral triangle shaped moving platform where each side of the triangle measures at 37mm (figure 4.25a). The motion pattern prescribed for the Delta robot is a 3 translational, 0 rotational pattern resulting in the simplest available virtual chain to be made from three *P* joints where each *P* joint acts along a separate axis from the other two (figure 4.25b).

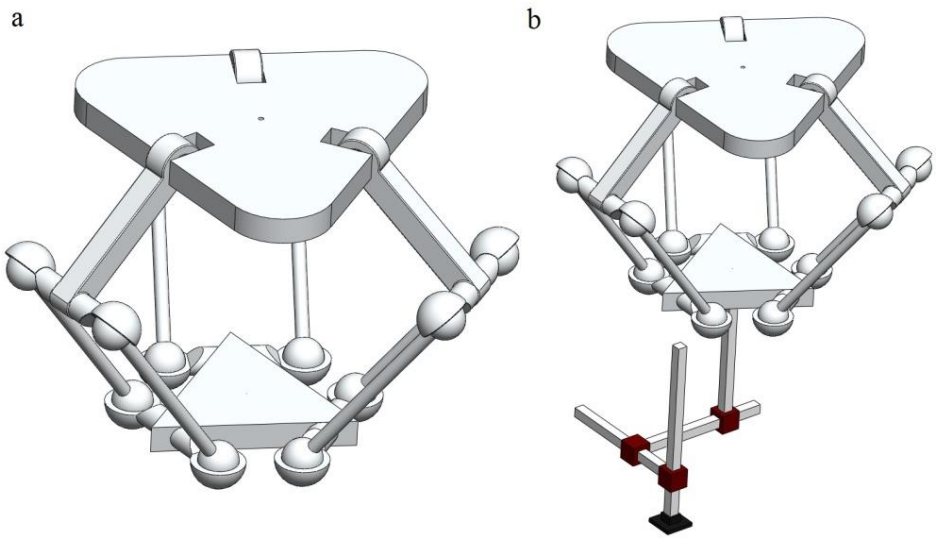

Figure 4.25: a) Delta robot model; b) with virtual chain.

To investigate the workspace of the Delta robot and compare it to that of the recorded workspace from [4.12], the vertical *P* joint of the virtual chain is limited to a height of 30mm. The remaining horizontal *P* joints were then actuated by an oscillating force applied to the external casings. After the motion study was simulated, the traced path component resulted in the rough workspace of the mechanism being produced (figure 4.26). In order to refine the workspace of the mechanism to bring it in line with previously found workspaces, displacement sensors were added to the location of the mechanism's end-effector. This produced the horizontal displacement values of the end-effector, which are shown in figure 4.27.

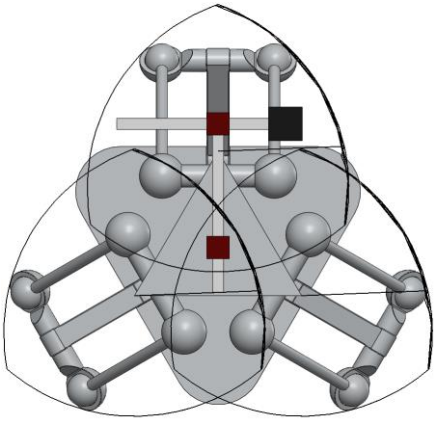

Figure 4.26: Horizontal workspace of the Delta robot from virtual chain force method.

From the displacement values in figure 4.27, a more refined workspace can be produced to better display the operational workspace for this mechanism. For comparison with the theoretical results shown in [4.12] the new workspace utilises the motor refinement method detailed in section 4.2.2 and is used to produce a circular workspace with a 25mm radius, from the centre of the mechanism (figure 4.28) which is within the limits of the x and z displacement values.

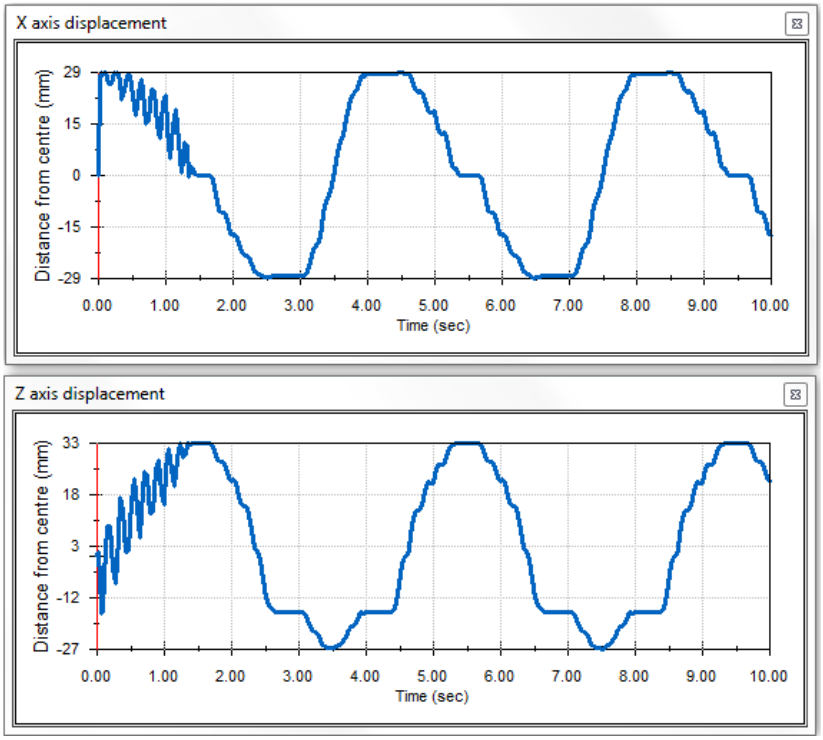

Figure 4.27: Displacement values for the horizontal workspace of the Delta robot from the virtual chain force method.

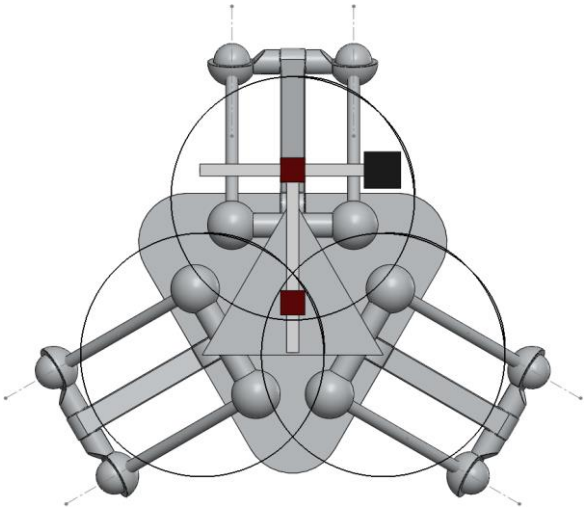

Figure 4.28: Refined horizontal workspace of the Delta robot using the virtual chain motor refinement method.

From the resulting workspace shown in figure 4.28, the method can be compared to the existing theoretical results gained in [4.12] (figure 4.29).

Figure 4.29c shows that the virtual chains method is capable of producing near identical workspace models for the outlying points of the Delta robot's moving platform to those produced via the traditional process. It should be noted that as the Delta robot from [4.12] does not give the dimensions of the mechanism or the final radius of the mechanism's

workspace. If this information had been provided in the paper, the workspaces would be identical.

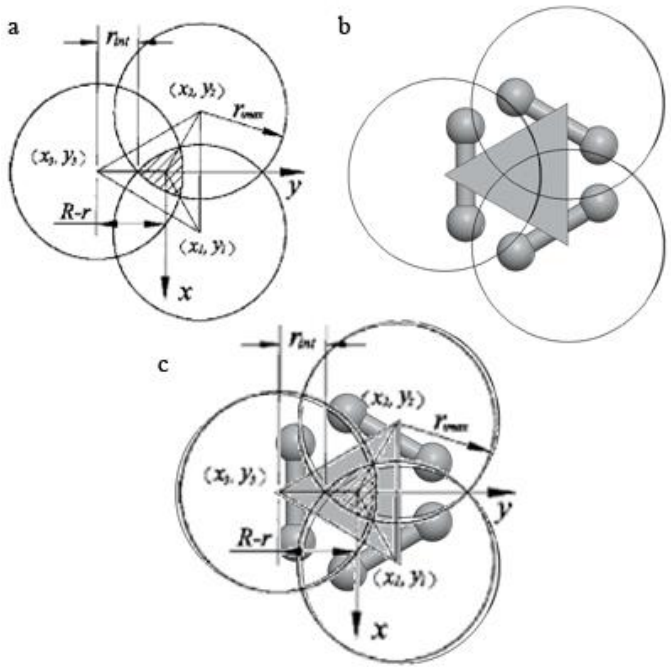

Figure 4.29: a) Theoretical workspace of the Delta robot [4.12]; b) Simulated workspace of the Delta robot (only moving platform shown for clarity); c) Comparison of theoretical and simulated workspaces.

With the successful reproduction of the 2D workspace, the 3D methodology will be applied to the mechanism to test its accuracy. The first method to be investigated is the virtual chain force method in which the vertical virtual *P* joint is actuated to manoeuvre through an incremental position changes of 2.5mm to allow a layered workspace to be formed. For this test, the traced location of the workspace is changed from the outer reaches of the moving platform to the centre of the platform where the mechanism's end-effector would be located. The displacement sensor for the vertical component of the mechanism's workspace is set as the distance between the underside of the base of the mechanism to the topside of the moving platform with a positive direction being in the upward direction. The result of the initial workspace is shown in figure 4.30 and the positional data from the displacement sensors of each axis are displayed in figure 4.31 where each graph shows the motion along a single axis resulting in an oscillating value as the moving platform moves back and forth along each axis past the starting value.

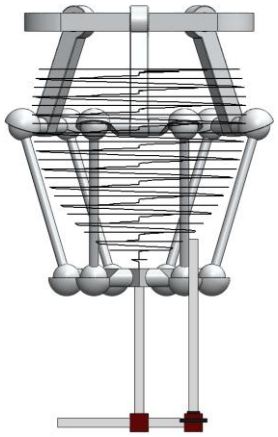

Figure 4.30: 3D workspace of the Delta robot from virtual chain force method.

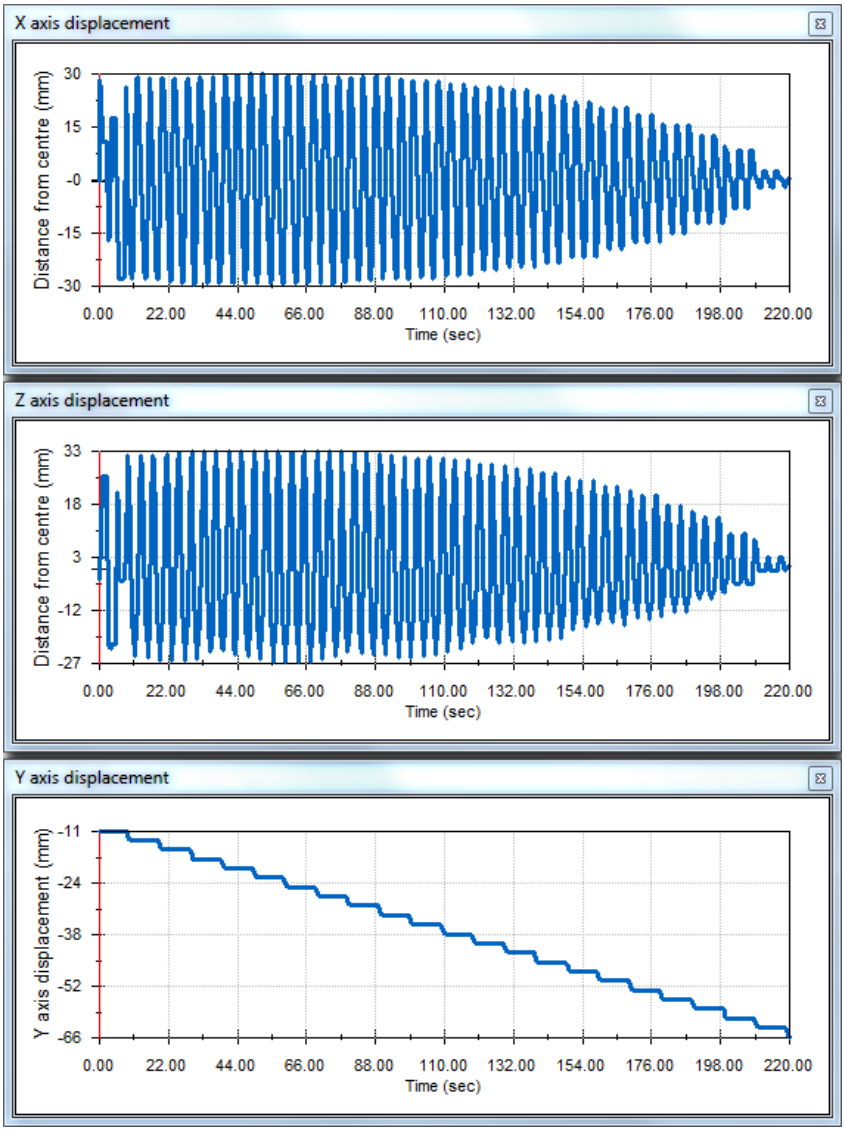

Figure 4.31: Displacement values for the 3D workspace of the Delta robot from the virtual chain force method for the x-, y-, and z-axes.

From the positional data shown in figure 4.31, the motor refinement method can be applied. The information from the displacement sensors allows the motor refinement

method to be optimised since the changes in the radius of the 2D slices will dictate the rate at which the slices are taken from the mechanism. The results of the motor refinement method are shown in figure 4.32 with a comparison of the resulting workspace made with the calculated workspace of a Delta robot using the traditional process [4.12] as displayed in figure 4.33.

From this, it can be seen that the workspace produced through the proposed method is capable of replicating the generalised shape of the Delta robot's workspace as calculated in [4.12] proving that the proposed methodology is valid.

The work completed in this section addresses the second objective of this thesis where virtual chains are utilised to develop a visual representation of parallel mechanisms' workspaces and the limitations there in by including the collision detection feature of the SolidWorks Motion Analysis simulation tool.

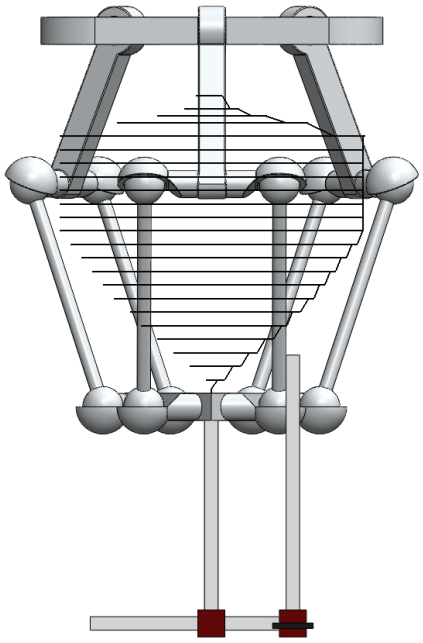

Figure 4.32: 3D workspace of the Delta robot from virtual chain motor refinement method.

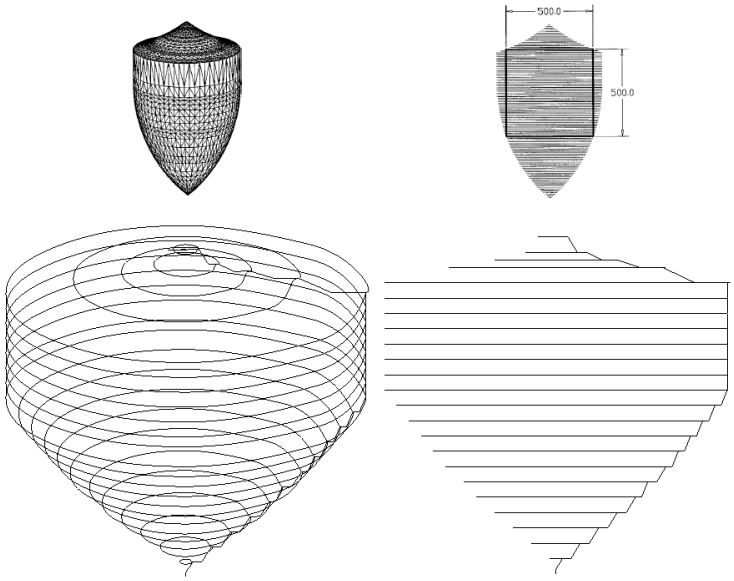

Figure 4.33: Theoretical workspace of the Delta robot [4.12]; Bottom: Simulated 3D workspace of the Delta robot using the motor refinement method.

4.4 Summary

In this chapter, a novel method of producing the workspace for a parallel mechanism with the application of the virtual chain method has been presented. The steps required for the production of initial 2D and 3D workspaces of parallel mechanisms are laid out in detail in the virtual chain force method and the concept refined by modifying the method to produce a motor refinement method workspace. These methods have then been investigated, successfully compared and proven with the use of three case study mechanisms. This novel approach enables the user to quickly build and simulate the model and workspace of any mechanism, with the external boundary of the workspace being the location of either a link or joint collision or the limitation of a joint in order to prevent a singularity from potentially occurring. This allows the user to modify the mechanism to suit a predetermined area of influence that the mechanism can operate in or allow the user to define the safe working area for personnel that will be working near the mechanism. This work therefore has addressed the second objective in section 1.6.

In the next chapter, the virtual chain approach used in this chapter and in previous chapters will be applied to the determination of the inverse kinematic model of the mechanism using the MapleSim software.

References

[4.1] M.A. Laribi, L. Romdhane and S. Zeghloul, "Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace", Machine and Machine Theory, vol. 42, pp 859-870, 2007.

[4.2] F. Pierrot, F. Marquet, O. Company and T. Gil, "H4 Parallel Robot: Modelling, Design and Preliminary Experiments", Proceedings of the IEEE International Conference on Robotics and Automation, pp 3256 – 3261, 2001.

[4.3] Z. Affi, L. Romdhane, and A. Maalej, "Dimensional synthesis of a 3-translational-DOF in-parallel manipulator for a desired workspace", European Journal of Mechanics A/Solids, vol. 23, pp. 311-324, 2004.

[4.4] M. Arsenult, and R. Boudreau, "Synthesis of Planar Parallel Mechanisms While Considering Workspace, Dexterity, Stiffness and Singularity Avoidance", Journal of Mechanical Design, vol. 128, pp. 69-78, 2006.

[4.5] D. M. Massala, "Analysis and Simulation of Parallel Robots for Sun Tracking Using a CAD - Based Approach", MSc Dissertation, Heriot Watt University, 2010.

[4.6] X. Kong and C. Gosselin, 2007, "Type Synthesis of Parallel Mechanisms", Springer Tracts in Advanced Robotics, vol. 33, chapter 7.8, page 123.

[4.7] I.A. Bonev and C.M. Gosselin, "Analytical Determination of the Workspace of Symmetrical Spherical Parallel Mechanisms", IEEE Transactions on Robotics, vol. 22, no. 5, pp 1011 – 1017, 2006.

[4.8] C.-R. Rad, S.-D. Stan, R.Balan and C. Lapusan, "Forward Kinematics and Workspace Analysis of a 3-RPS Medical Parallel Robot", Proceedings of the IEEE Conference on Automation, Quality and Testing, Robotics (AQTR), vol. 1, pp. 1 – 6, 2010.

[4.9] J. Gallardo, H. Orozco and J.M. Rico, "Kinematics of 3-RPS parallel manipulators by means of screw theory", International Journal of Advanced Manufacturing Technologies, vol. 36, pp. 598 – 605, 2008.

[4.10] K.A. Arrouk, B.C. Bouzgarrou and G. Gogu, "CAD based techniques for workspace analysis and representation of the 3CRS parallel manipulator", 19th International Workshop on Robotics. pp. 155 – 160, 2010.

[4.11] K.A. Arrouk, B.C. Bouzgarrou and G. Gogu, "CAD Based Geometric Procedures for Workspace and Singularity Determination of the 3-RPR Parallel Manipulator", Applied Mechanics and Materials, vol. 162, pp 131 – 140, 2012.

[4.12] X.-J. Liu, J. Wang, K.-K. Oh and J. Kim, "A New Approach to the Design of a DELTA Robot with a Desired Workspace", Journal of Intelligent and Robotic Systems, vol. 39, pp 209 – 225, 2004.

Chapter 5 Inverse kinematic and dynamic analysis of parallel mechanisms using MapleSim and the concept of virtual chain

Following the work in Chapters 2, 3, and 4, this chapter investigates a novel approach to analysis and developing inverse kinematic and dynamic models for parallel mechanisms by using the graphical programming software capability of MapleSim.

5.1 Introduction

When investigating the motion of a mechanism, there are two basic methods of determining its state, forward (direct) kinematic modelling or inverse kinematic modelling.

Forward kinematic modelling requires the joint angles of the mechanism to be known in order to determine the location of the end-effector of the mechanism. This method is preferred by the robot's operator as they can program the mechanism through a series of nodes that are not yet determined.

Inverse kinematics is the geometric study of a mechanism in which the position of the end-effector is known and the joint angles of the various joints along its limb(s) are the desired outcome of the calculations. The primary advantages of inverse kinematic modelling are that a location position can be specified for the mechanism's end-effector from which the control system can determine how the joints should be positioned to achieve the motion. This method provides a more intuitive design approach because it supports the analysis from a more practical application and allows for the development of a more functional product. For the purposes of mechanism control, the inverse kinematic modelling method will be used throughout the rest of this thesis.

In a similar fashion, to the inverse kinematic modelling of a mechanism, its inverse dynamic model is determined by first taking into consideration its desired motion and then calculating the amount of force required to produce the desired movement. This is particularly important when developing a new mechanism as the inverse dynamic model allows the designer to determine the maximum actuator forces required in order to enable the mechanism to perform to the desired speed and lifting power in relation to the design specification.

In this chapter, the theoretical methods of producing both inverse kinematic and dynamic models will be compared to a proposed novel method that uses the virtual chain approach to produce a simulated representation of these design elements. The chapter introduces the graphical programming software "MapleSim" and details how a mechanism can first be produced, by implementing the necessary virtual chain in order to quickly and intuitively create its inverse kinematic model. It then outlines the development of this approach by determining a dynamic model of a 3-RRR planar parallel robot. Lastly, a method to produce a mechanical efficiency model of a mechanism is developed in order to determine whether the system's design is efficient.

5.2 MapleSim

MapleSim is a graphical programming software that uses pre-programmed component blocks to write mathematical code for a number of different physical phenomenon from electrical circuits and magnetic physics components through to mechanisms. It also

produces a visual representation of the allocated components [5.1]. These mechanism blocks are referred to as multi-body components and are the focus of this chapter.

5.2.1 MapleSim interface

The interface of MapleSim is designed to allow the user to create numerous mathematical, electronic, and mechanical simulations with a basic understanding of how the components function in relation to each other. Figure 5.1 shows the general layout of the MapleSim Interface with several sections highlighted and numbered.

Section 1 in figure 5.1 shows the "Main Toolbar" (figure 5.2). This consists of the main buttons for running the simulation, copying and pasting components, creating/opening/saving a model and adding attachments to the model such as external data files and export file wizards for the model or its components.

Section 2 is the "Navigation Toolbar" (figure 5.3). This toolbar allows the user to switch between the base Maple code, which is represented by the model, its graphical representation, a list of the models parameters, and any model subgroups.

Sections 3 and 6 are collectively known as the "Pallets Pane" with Section 3 (figure 5.4) providing the display for the currently selected component and a search window to locate a particular component block. Section 6 (figure 5.5) includes the eight different component block options as well as any example models or projects installed within MapleSim as well as a section dedicated to "favourite" components where any block can be dropped as a convenient place from which to copy them later.

The eight-dropdown selections in the component library represent different types of blocks that can be used in MapleSim. A brief description of each option's component type is now detailed [5.1]:

- Signal Blocks: any components that are designed to manipulate or generate an input or output signal in the model.
- Electrical: any components intended to model electrical analogue or digital circuits, single-phase and multiphase systems and machines.
- 1-D Mechanical: any components designed to represent single dimensional mechanical motion (translational and rotational).
- Multibody: any components that will have a graphical representation and any components that control the force, motion or any other feature of such components. This option will be used primarily in this thesis as the components within it allow the production of mechanisms for use later in this and the following chapters.
- Hydraulic: any components that model the control system or represent a hydraulic system.
- Thermal: any components that model heat flow and/or heat transfer of the model.
- Magnetic: any components designed to simulate magnetic circuits.

Figure 5.1: MapleSim Interface.

Figure 5.2: MapleSim Main Toolbar.

Figure 5.3: MapleSim Navigation Toolbar.

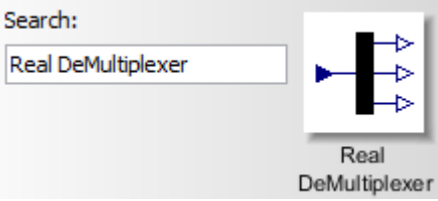

Figure 5.4: Pallet Pane search/current Component Box

The fourth highlighted section of figure 5.1 shows the "Model Workspace Toolbar" (figure 5.6). This allows the user to manipulate the model by use of the select tool (mouse pointer button), erase sections using the eraser button and add annotations with the text, line and shape buttons. The final three buttons on the toolbar allow the user to add a probe to the model in order to determine the value of a signal or joint angle/position or any other type of desired information. The middle of these three buttons allows the user to add a new parameter to the model and the final button allows the user to enable or disable any model selected components and connections.

The fifth section is the "Parameters Pane" (Figure 5.7). This allows the user to modify a selected component's parameters from the length, position and orientation of a link to the amplitude and frequency of an input signal's wavelength it also helps to define what information a probe placed into the model should record.

In the parameters pane, the manner in which the model's multibody components are arranged is set. This is done by manipulating either the position of the component, the length along the component's x-, y- and z-axes and/or the rotation of the component's end. The coordinate frames for the components are determined by the local coordinates of the model component that, by default, are set to the world coordinates; this means that any force, rotation or position data gathered by the probes will initially be set to these coordinates. By adjusting the rotation of the component to match that of the desired orientation of the model's component, the user can redefine how the probe data is orientated, resulting in a local value for any force, rotation or position.

The final two highlighted sections are the "Model Workspace" (7 on figure 5.1) and the "MapleSim Console" (8 on figure 5.1). The model workspace is the area in which all components are placed in order to produce the MapleSim model. In this window the components are connected by selecting the output port of a component and connecting it to the input port of another. The colour of the port determines the type of connection that can be made with grey representing a multibody connection, blue a signal connection and green a 1D mechanical translation component connection.

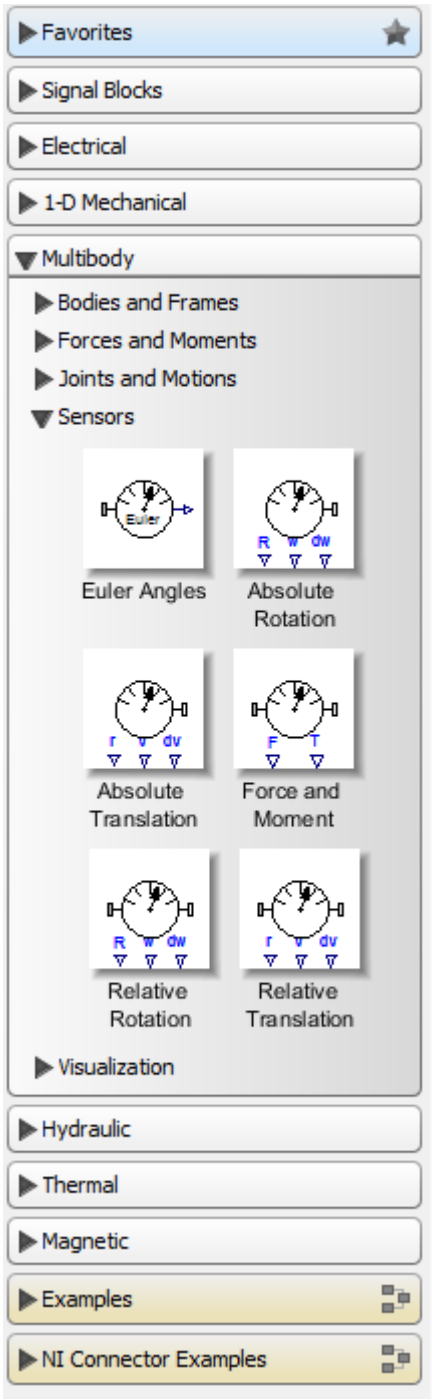

Figure 5.5: Pallet Pane Component Library.

Figure 5.6: MapleSim Model Workspace Toolbar.

Figure 5.7: MapleSim Parameter's Pane.

The console section (figure 5.8) is split into two sections: the message console at the top of the section and the console toolbar at the bottom. The message console gives detailed information on the progress of the simulation and, in the event of a failed simulation, it displays the error with a location in the model subgroups in order for the user to quickly locate the issue and fix it.

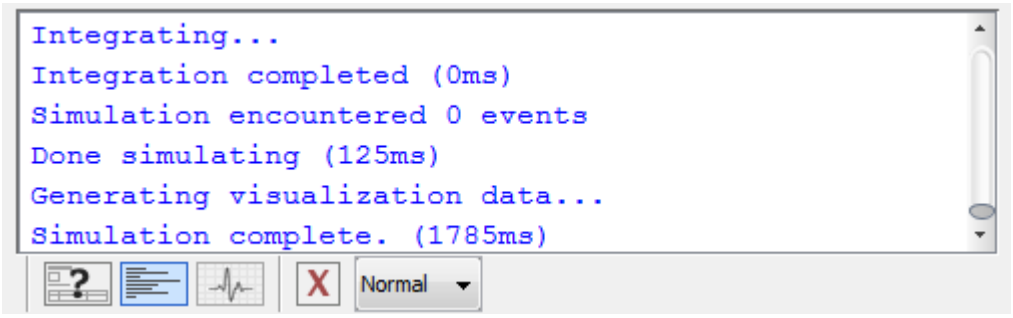

Figure 5.8: MapleSim Console.

The console toolbar allows the user to alter the amount of information displayed on the console, erase the information in the console and change the type of information being displayed.

5.2.2 Model creation

Creating a MapleSim model begins with determining the parts required for the mechanism to be correctly represented in the software's 3D environment. This is achieved by utilising the extensive list of multibody components available in the MapleSim library to represent components of a mechanism such as "Rigid Body Frame" components for a link and plate lengths and "Rigid Body" components for the mass of the mechanism's parts (figure 5.9).

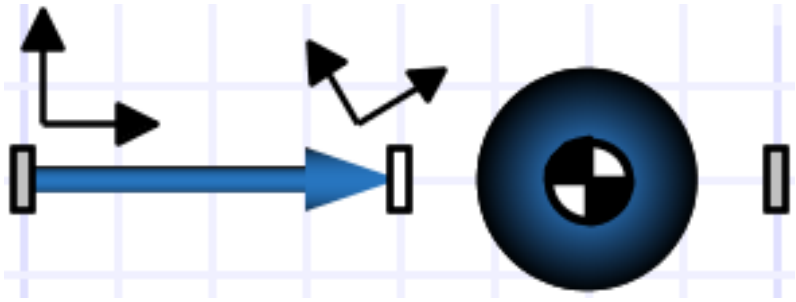

Figure 5.9: Rigid body frame and rigid body components from the MapleSim multibody library.

MapleSim models are fixed in place by "Fixed Frame" components that have a single multibody port and are defined in the virtual environment by a given value for the frame's x-, y- and z-coordinates as well as the initial rotation of the frame in relation to the world coordinates. As all frames used in multibody components take their orientation from the output port of the previous multibody component, it is especially important to re-orientate the initial frame if the kinematic chains of the mechanism are offset from the world coordinates, as is the case with the Delta Robot.

Due to the fixed frames being able to be located at the point where the kinematic chain meets the base plate, it is not usually necessary to include the mechanism's base plate in a MapleSim model, although for thoroughness the base plate can still be modelled utilising a single fixed frame for the location of the base plates centre. With this in mind, the kinematic chains of the mechanism are initially built up from the fixed frame component with the initial joint link usually an R joint or a P joint both of which can be represented in MapleSim using "Revolute" and "Prismatic" multibody components (figure 5.10) by connecting the output port of the fixed frame to the input port (port A) of the joint. The joints are then connected to a rigid body frame in the same manner as before. For a standard link of any length it is usual to represent the link with a pair of rigid body frames with a rigid body component in the centre as the mass moment of inertia can be controlled through the rigid body frame's properties window.

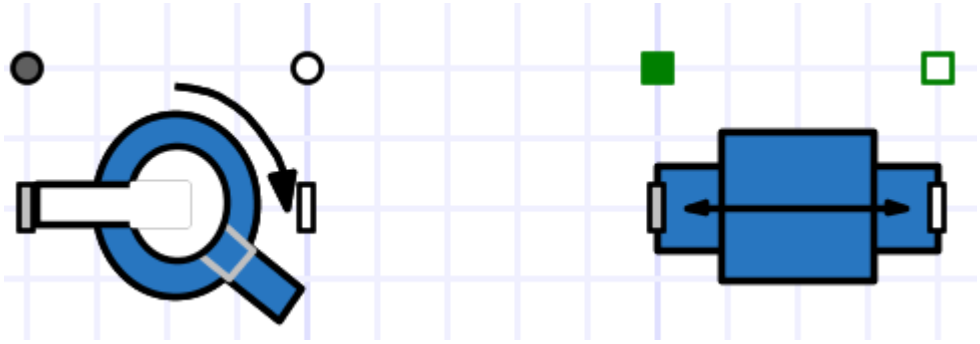

Figure 5.10: Revolute and prismatic components from the MapleSim Multibody library.

5.2.3 Case study

The first case study of this chapter looks at a planar parallel mechanism that is operated using a prismatic multibody component (figure 5.11). The mechanism has two limbs, each operating along a different axis on the horizontal plane. The kinematic chains are *PRR* chains and meet at a single location in the centre of the mechanism where the end-effector would be located (figure 5.12). As the mechanism has actuated P joints, the

method of controlling the actuated joints differs in the controlling block where the revolute multibody component requires rotational position components; the revolute joint component requires "Translational Position" 1D mechanical translational motion driver components. The method of feeding control signals into the translational position driver components is the same as the rotational position motion driver components and in this case, the mechanism will be driven from two time lookup table components that require data from a spreadsheet. The data from the spreadsheet is laid out so that the first column represents the time stamp of the simulation and the following columns are called up by the time lookup table component to produce a signal. For this case study, the data in table 5.1 gives the link parameters and masses of the links while table 5.2 is used to produce the motion the results of which are taken from the MapleSim probes outputs shown in figure 5.13.

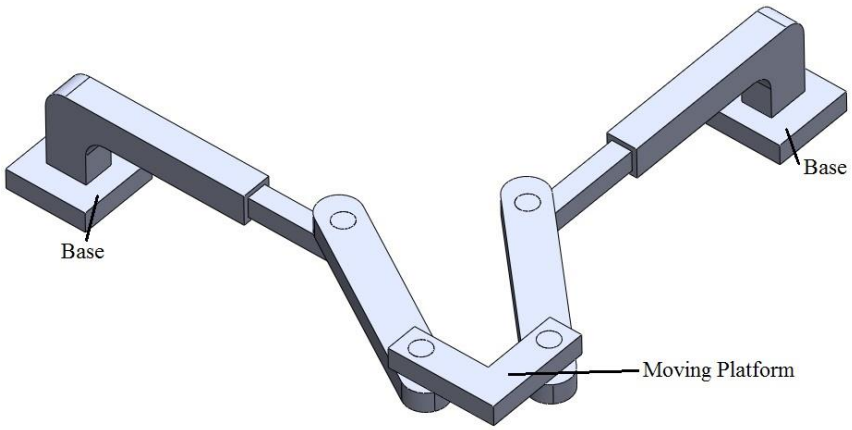

Figure 5.11: Planar parallel mechanism CAD model.

For the simulation to succeed the initial conditions for the simulated mechanisms joint positions are essential. For this case study the position of the prismatic joints are initially set to 0m from the fixed frame and the revolute joints are set at -30° and 30° for joints R_1 and R_2 of one arm respectively and -60° and 60° for joints R_3 and R_4 for arm 2 respectively. From this case study, it can be seen that while it is possible to produce the motion of a parallel mechanism using a time look up table component combined with a translational move motion driver component, the actual motion of the mechanism can be unpredictable as the equations are generated by the Maple mathematics program and are not directly controlled by the user.

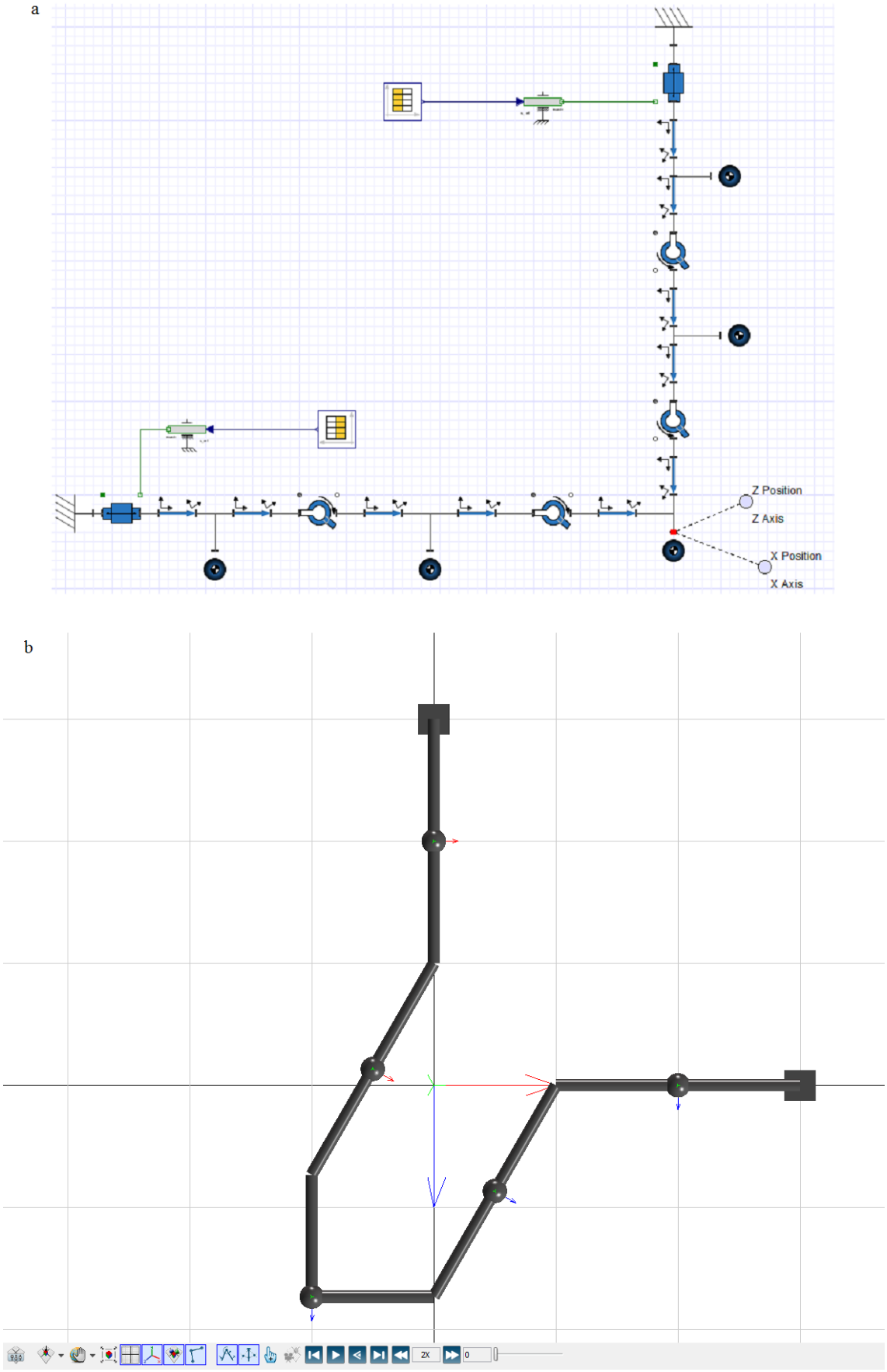

Figure 5.12: a) MapleSim block diagram of the planar parallel mechanism case study; b) 3D visual representation of the mechanism.

Table 5.1: Link parameters for the planar parallel mechanism.

Link	Length (m)	Mass (kg)
PR link	1	1
RR link	1	1
R to end-effector link (Moving Platform)	1	1

Table 5.2: Motion plot in meters from the centre of the workspace used for the time lookup tables in case study 3.

Time (s)	Position on x-axis (m)	Position on z-axis (m)
0	0	0
1	1	0
2	1	-1
3	-1	-1
4	-1	1
5	1	1
6	1	-1
7	-1	-1
8	-1	1
9	1	1
10	1	-1

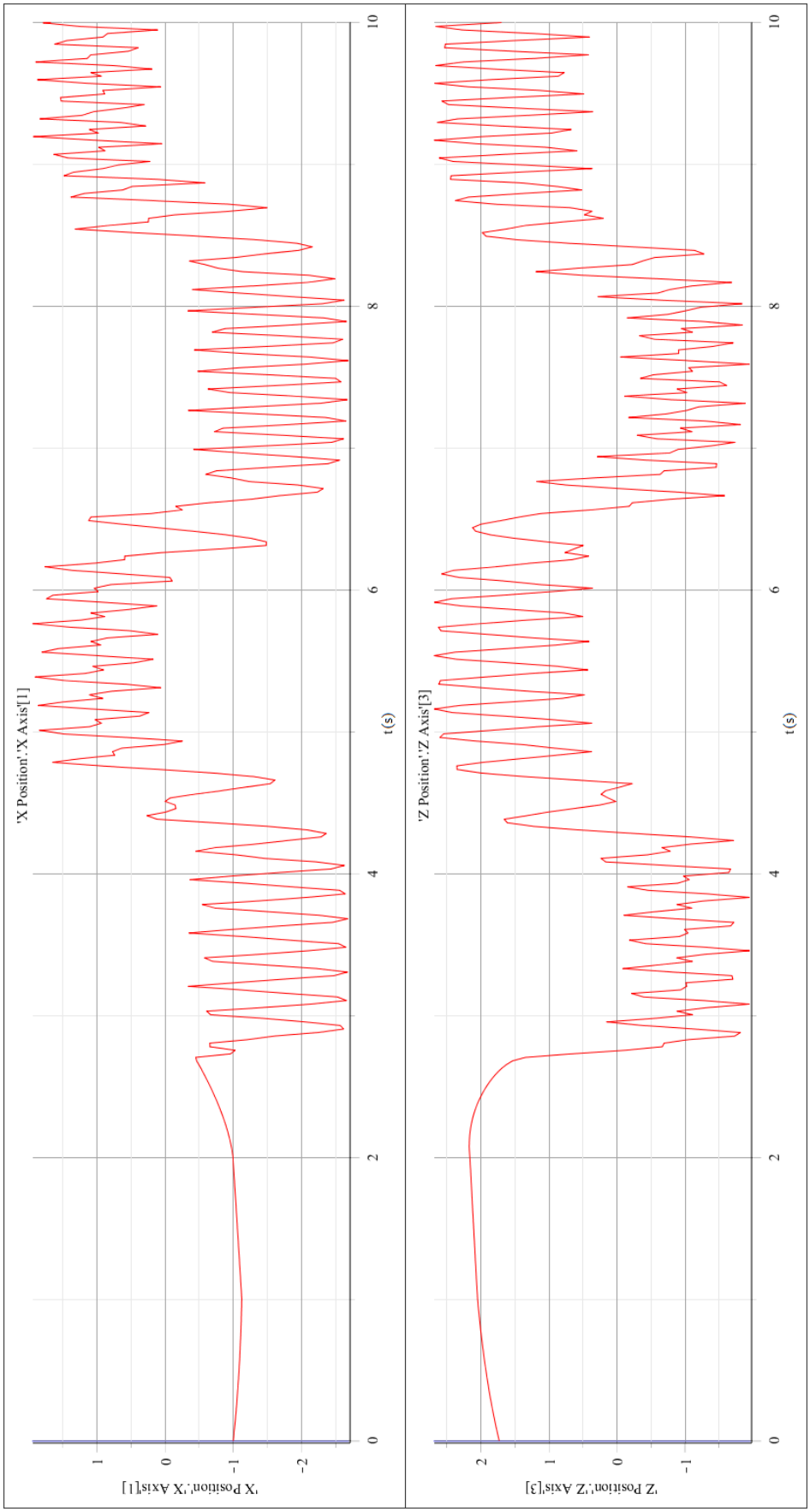

Figure 5.13: Probe data from the motion of the planar parallel mechanism case study MapleSim model during simulation.

5.3 Inverse Kinematic Analysis

As already stated, the basis of the virtual chain is that it is a representation of the output motion of a mechanism. As the virtual chain can be actuated to produce the workspace rather than its joints (chapter 4), the mechanism and virtual chain can be considered together as a visual representation of its inverse kinematic model. By applying a virtual chain to a MapleSim model of a parallel mechanism and controlling the virtual chain, the mechanism can be manipulated to produce the joint positions required for the mechanism during motion, giving results that would normally be produced through an inverse kinematic model.

To test this method, a case study involving a 3-RRR planar parallel mechanism is presented. The mechanism in question is shown in figure 5.14 as a CAD model with a PPR virtual chain attached to the moving platform. The mechanism is then converted into a MapleSim model with the virtual chain attached to the rigid body component that represents the middle of the moving platform as shown in figure 5.15.

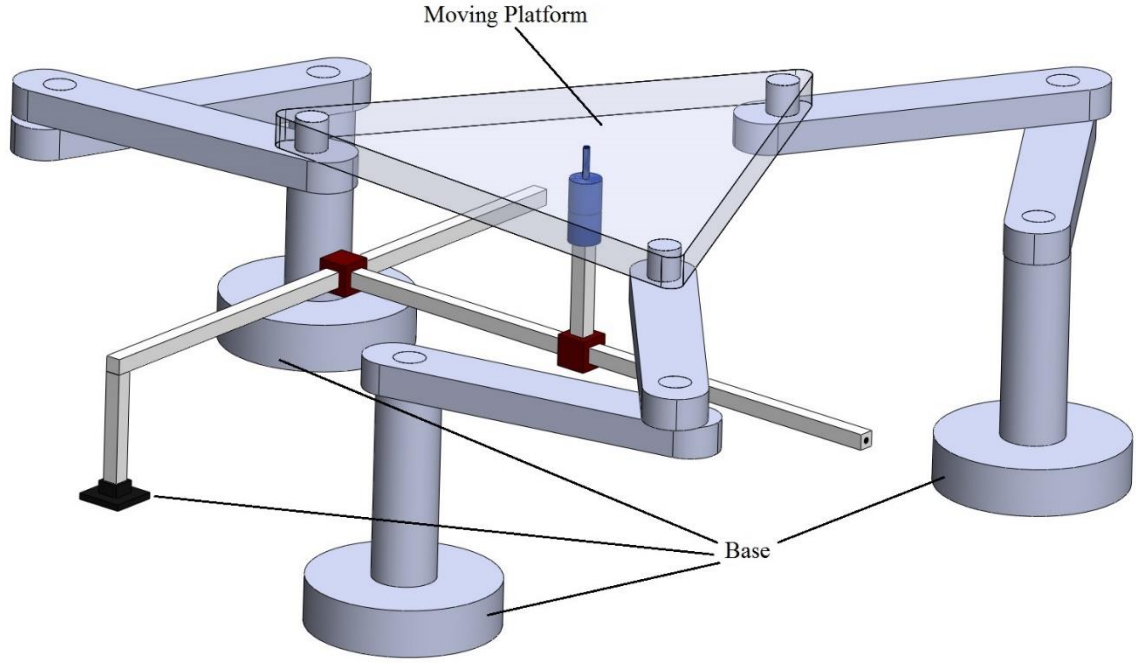

Figure 5.14: 3-RRR planar parallel mechanism with PPR virtual chain.

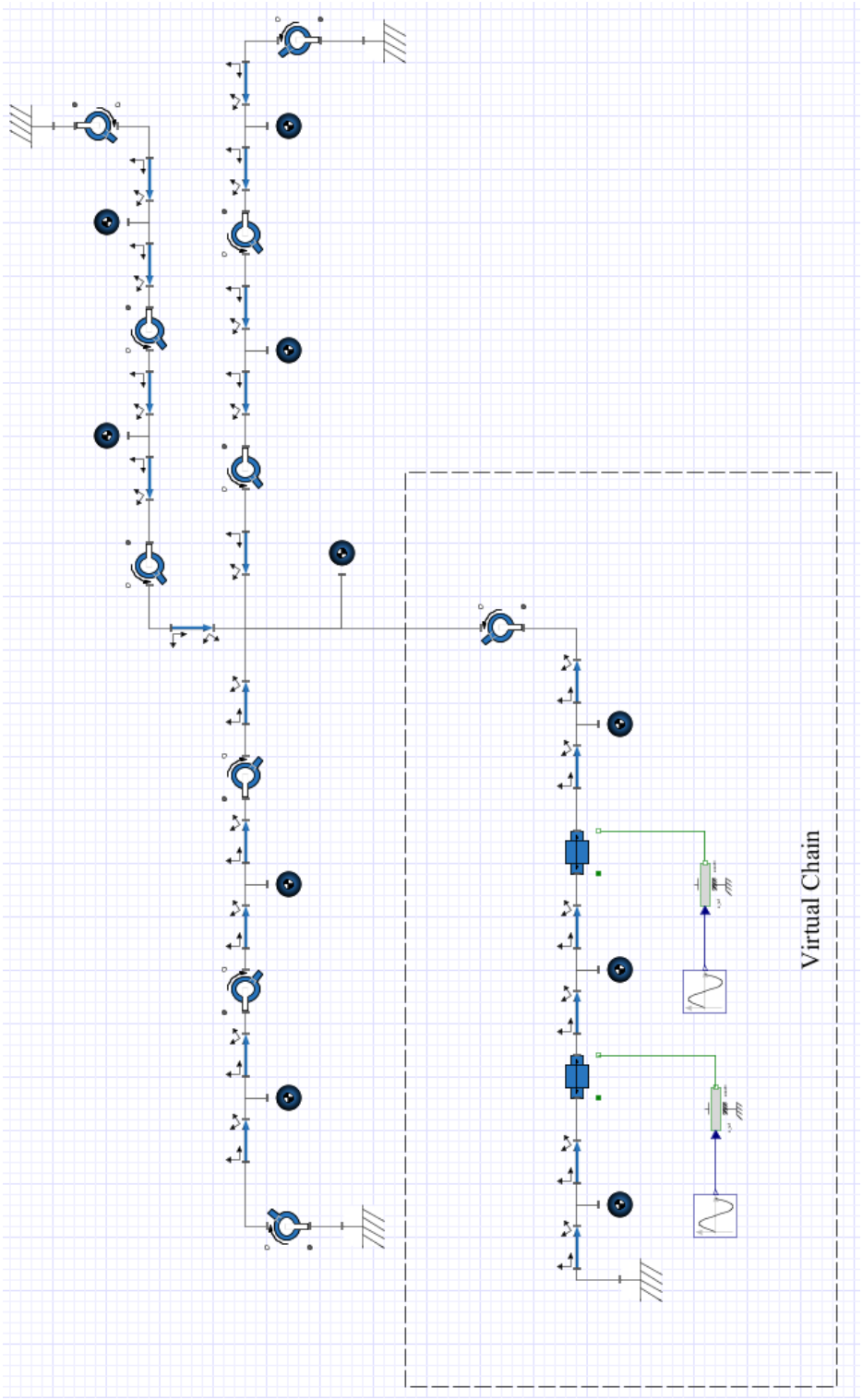

Figure 5.15: MapleSim representation of 3-*RRR* planar mechanism with virtual chain.

As the desired motion pattern of the mechanism is planar in nature, the *PPR* virtual chain is selected where the axes of translation for the *P* joints and the axis of rotation for the *R* joint are all perpendicular. Normally each joint of the virtual chain would include a motion controller to define the motion path of the mechanism; however in this case the *R* joint is left free so that the simulation is simplified.

The initial conditions of the mechanism's joints are inputted into the parameters pane of the individual joints with the virtual joint's starting positions set to "strictly enforce" so that the mechanism begins at the desired starting location. The sinusoidal signal blocks are then given a generic harmonic waveform subsequently translated into a positional movement for the virtual *P* joints by their attached transitional move components. These harmonic waveforms take on the desired position for the end-effector of the mechanism with the orientation of the moving platform being free to be determined by the required arrangement of kinematic limbs of the mechanism so that the desired motion can be realised. The simulation is then run to ensure that the motion path is viable with the lengths of the mechanism linkages and the rotation values of the revolute components at the base of each kinematic limb recorded by attaching probes to the real number output of the 1D rotational "Angle Sensor" components (figure 5.16).

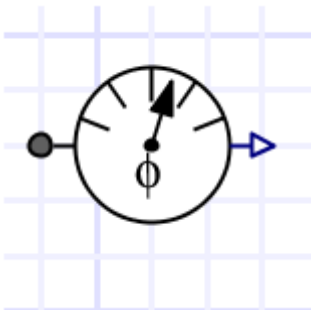

Figure 5.16: MapleSim 1D rotational angle sensor component block.

The results of the simulation are shown in figure 5.17.

In order to determine whether the inverse kinematic model is completely successful, the mechanism's model is copied into the model workspace and the inverse kinematic model component placed into a subgroup. The values of the angle sensors are wired to the external walls of the subgroup and then connected to "Angular Rotation Move" components ("Translational Position Move" component for *P* joints) (figure 5.18). These are then connected to flange_b of the first *R* joints in each kinematic chain in the secondary mechanism. Finally, the fixed frames of the second mechanism are translated across the x-axis by 1.5 meters so that the mechanisms can be distinguished in the visual representation after simulation. The MapleSim model is then simulated and the movement of both mechanisms inspected in the visual rendering of the model (figure 5.19).

From the simulation of the two mechanisms, it was determined that the virtual chain method for producing an inverse kinematic model was successful since the secondary mechanism mirrored the motion of the inverse kinematic model mechanism perfectly.

A comparison of this methodology is presented in [5.2] in which the inverse kinematic model of the 3-*RRR* planar parallel mechanism is developed. The resulting inverse kinematic equations for the actuated joint positions are given by equation (5.1).

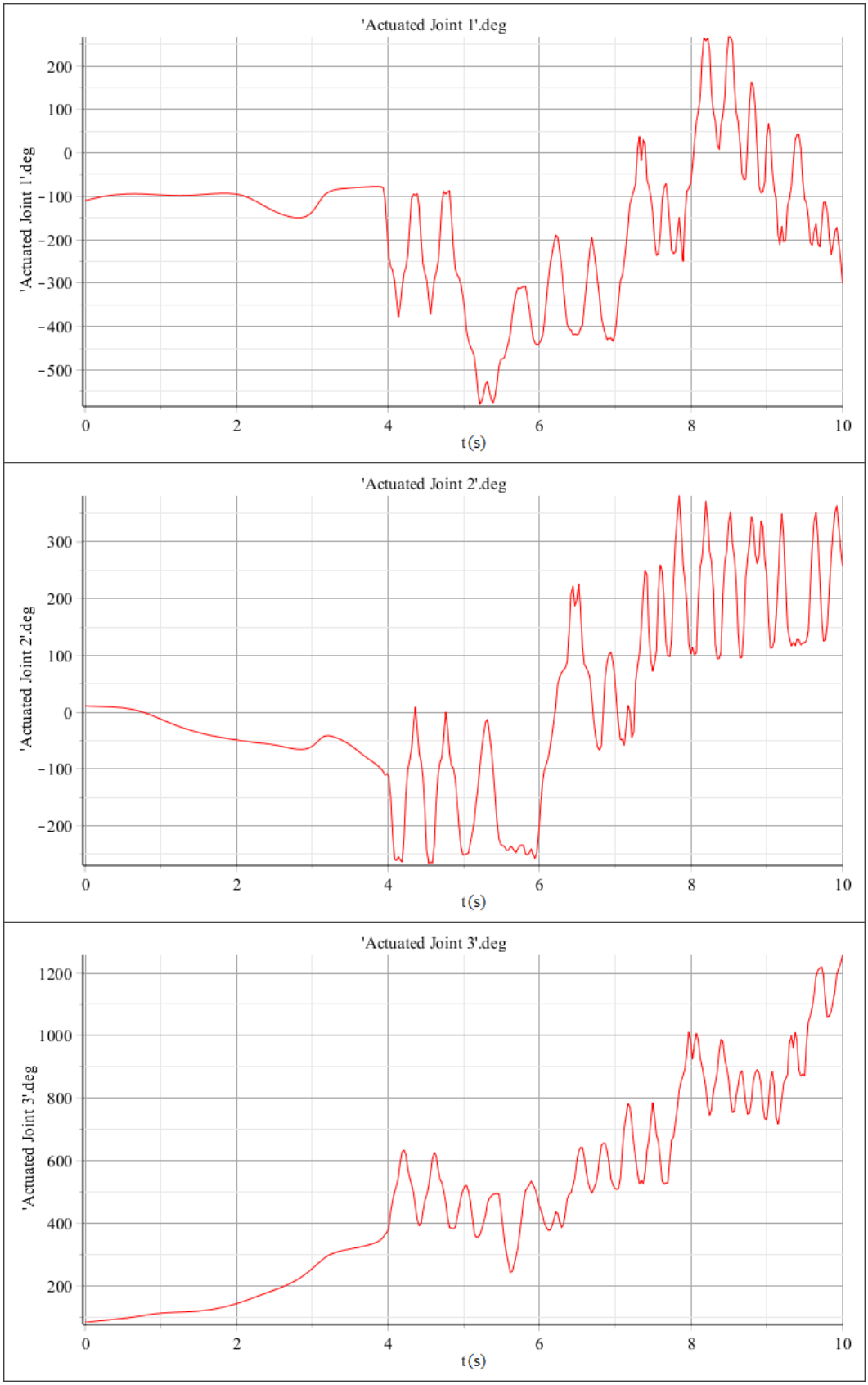

Figure 5.17: Probe data on the motion of the case study MapleSim model during simulation.

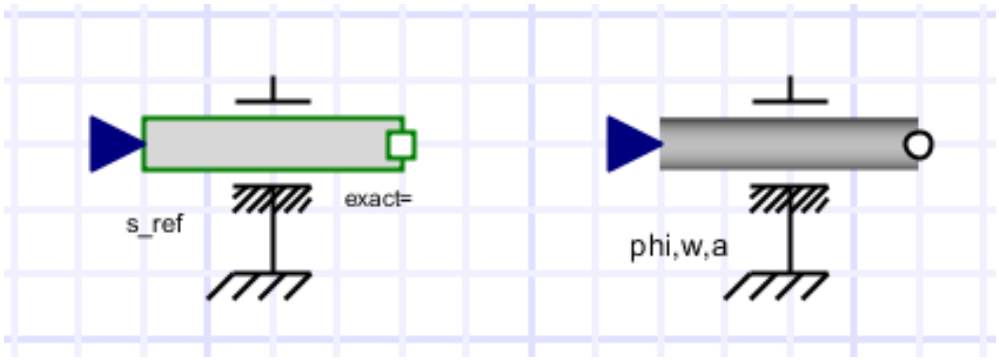

Figure 5.18: Translational position move and angular rotation move MapleSim blocks

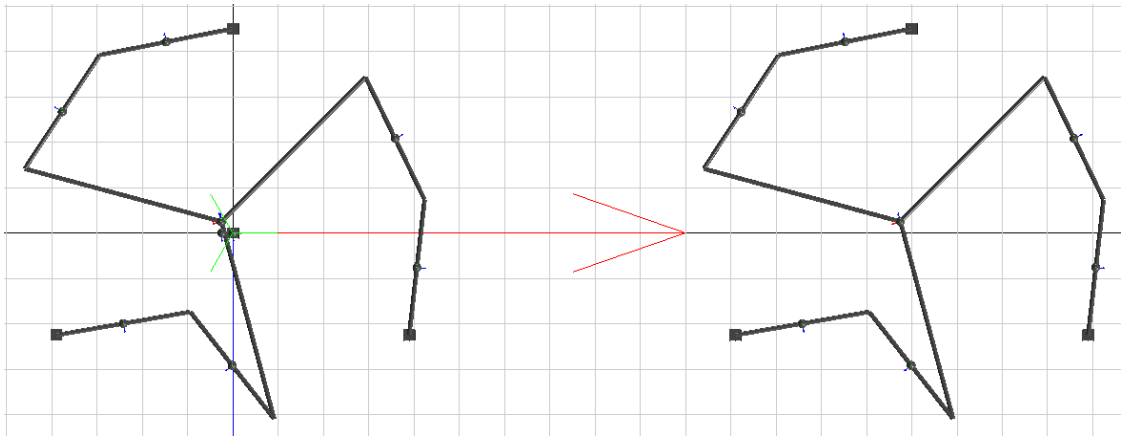

Figure 5.19: MapleSim visual representation of the two 3-*RRR* planar parallel mechanisms.

$$\theta_i = A\tan 2(K_i, F_i) \pm A\tan 2\left(\sqrt{(K_i^2 + F_i^2 - E_i^2), E_i}\right) \tag{5.1}$$

where:

$$\begin{aligned} E_i = {P_x}^2 + {P_y}^2 + {l_{Ai}}^2 + {l_{Bi}}^2 + {x_{Ai}}^2 + {y_{ai}}^2 - 2P_x l_{di}\cos(\sigma_i + \varphi) - 2P_x x_{Ai} \\ + 2l_{di}x_{Ai}\cos(\sigma_i + \varphi) - 2P_y l_{di}\sin(\sigma_i + \varphi) - 2P_y y_{Ai} \\ + 2l_{di}y_{Ai}\sin(\sigma_i + \varphi) \end{aligned}$$

$$F_i = -2l_{Ai} + 2l_{Ai}l_{di}\cos(\sigma_i + \varphi) + 2l_{Ai}x_{Ai} \tag{5.2}$$

$$K_i = -\left[-2P_y l_{Ai} + 2l_{Ai}l_{di}\sin(\sigma_i + \varphi) + 2l_{Ai}y_{Ai}\right]$$

where P_n is the central point of the moving platform when $n = x, y \text{ or } z$, l_{Ai} is the length of the link between the actuated R joint and the secondary R joint, l_{Bi} is the length of the link between the secondary R joint and the final R joint, l_{di} is the distance from the final R joint to the central point of the moving platform, x_{Ai} and y_{Ai} are the coordinates of the i^{th} actuated joint, φ is the orientation angle of the moving platform, and σ_i is the angle between the central point of the moving platform and the final R joint of the i^{th} kinematic chain.

The work completed in this section addresses the first portion of the third objective laid out in Chapter 1, where virtual chains are utilised to develop a the inverse kinematic model of a parallel mechanism. Additionally this section continues to address the second

objective by producing a visual representation of a parallel mechanism's inverse kinematic model with the ability to change portions of the mechanism in solve any issues that may arise during the simulation of the model.

5.4 Inverse Dynamic Analysis

In the previous section, the virtual chain approach was applied to a MapleSim model in order to replicate an inverse kinematic model. The next stage in developing this method is to determine the dynamics of the system by measuring both the dynamic forces being input to the system and the forces acting at the location of the moving platform.

Traditionally this is done by utilising one of three methods for calculating a parallel mechanism's dynamic model. These are known as: the Newton-Euler approach ([5.3-5.6]), the Lagrange approach ([5.7-5.9]) and the virtual work/Kane's method ([5.10]) [5.11]. All three methods will produce a dynamic model of a mechanism [5.13-5.16] so only one needs to be investigated to prove the validity of the proposed method.

As the dynamic model of a mechanism investigates the forces acting upon the moving platform of a mechanism including the generalised weight component, the case study for this section will not be compared to a pre-existing mechanism from published work. The mechanism that will be investigated is the same 3-RRR planar mechanism used in the inverse kinematic model case study in the previous section (figure 5.14).

For the development of a dynamic model in MapleSim, the method of attaching a virtual chain to the end-effector or moving platform and controlling it through it will not produce an accurate dynamic model, as there will be a greater amount of force acting on the moving platform than the total amount of force present at the actuated joints. This is due to gravity being applied to the model in the simulation settings as well as any friction components that are being applied to the model. These friction components can be either: a "Bearing Friction" component for rotational friction of R and S joints; a "Translational Friction" component for the friction produced by P joints or when two objects slide against each other (figure 5.20); or a custom MapleSim block which simulates the friction of a bearing and then applies the resulting reduction in motion to the movement of the joint. These friction components cause the model to lose force as it passes through the mechanism from the moving platform to the actuated joints of each kinematic chain.

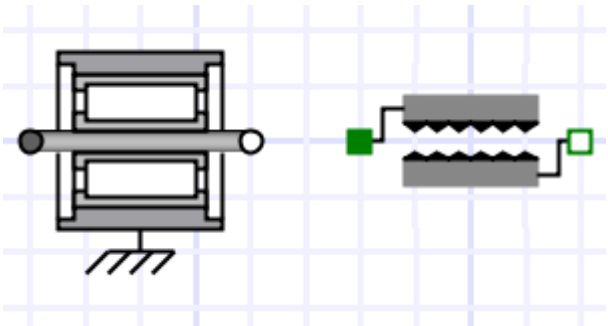

Figure 5.20: Bearing Friction and Translational Friction MapleSim blocks

To avoid this problem, the additional mechanism used in the inverse kinematic model case study, used to identify whether the values of joint motion gained in the simulation were accurate, is modified slightly in order to record the forces required to move the

actuated joints through the desired motion and to determine the forces present at the moving platform of the mechanism.

The reason for using this method is that the virtual chain controlled mechanism will be manoeuvred through a given path with the virtual chain joints controlling the mechanism itself. The selected actuated joints then have their position (for P joints) or their rotation (R joints) recorded using either a "Translational Position Sensor" or an angular rotation sensor component. These components read the position or rotation of the joint and then output a real value signal that can be taken and placed directly into a translational position move or an angular rotation move component, which when attached to a prismatic or revolute multibody component moves the joint in exactly the same manner as the joint being read.

This allows the virtual chain driven mechanism to drive the second mechanism from the actuated joints, effectively making the first mechanism an inverse kinematic model control system for the second mechanism. The second mechanism can then be investigated to determine the forces acting upon the kinematic chains and the moving platform, which can then be recorded via probes, thus giving the user the ability to find the dynamic forces acting within the mechanism. The force lost through the system can be determined by calculating the difference between the total force produced at the actuated joints and the force acting upon the moving platform (equation 5.3).

$$F_L = \sum F_i - F_e \tag{5.3}$$

where F_L is the force lost through the motion of the mechanism, F_e is the force acting upon the moving platform, F_i is the force produced at actuated joint i.

To record the forces produced at each R joint and at the moving platform, a "Force and Moment" sensor component is placed into the mechanism between the joints and the rigid body frames of each kinematic chain as well as between the rigid body frames and the rigid body component at the location of the end-effector (figure 5.21).

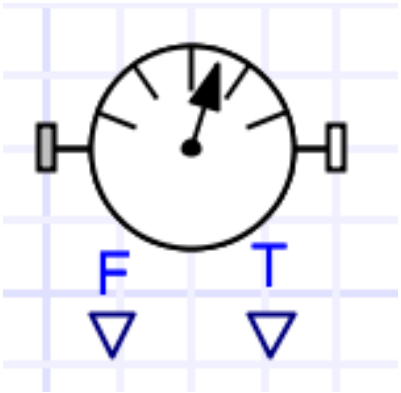

Figure 5.21: Force and Moment sensor block

The Force and Moment sensor component produces two (3×1) force vectors of real numerical data, the force and torque of being applied to the inboard node. These arrays consist of three numerical values, one for each local axis. This means that the arrays are orientated by the rotation of the node they are connected to via the inboard port and

therefore must be rotated back to a standardised global coordinate frame if they are to be accurately applied to equation (5.3) (figure 5.22).

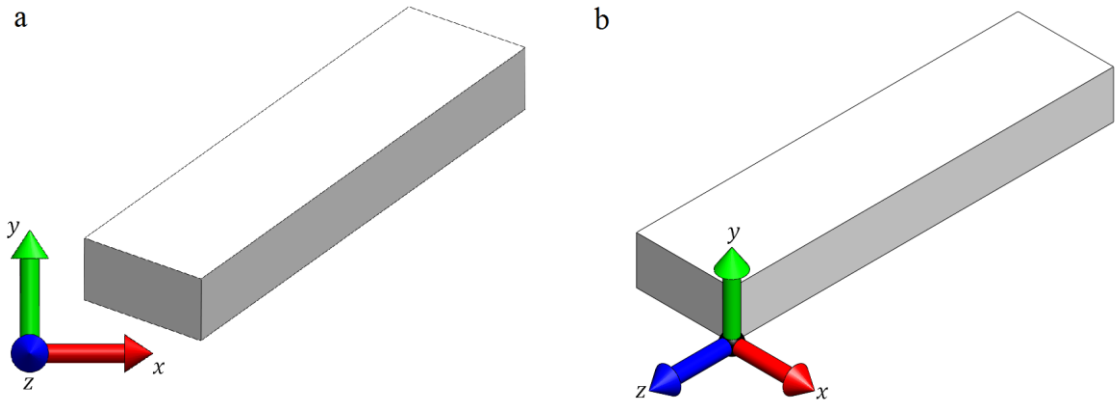

Figure 5.22: Force and moment frames in the a) global coordinate frame. b) local coordinate frame.

The MapleSim block diagram for the inverse dynamic model of the 3-RRR planar parallel mechanism is shown in figure 5.23 and the results for the torque values of the inverse dynamic model are detailed in figures 5.24 and 5.25 for the kinematic chains and moving platform respectively.

The force lost through the system is then calculated by adding the actuator torques together and subtracting the total force found at the end-effector. The results of which are shown in figure 5.26.

The work completed here addresses the third objective's requirement of using the inverse kinematic model of a parallel mechanism to develop an inverse dynamic model of the mechanism.

5.5 Efficiency Calculation

As stated in chapter 1, the efficiency of parallel mechanisms has been extremely under researched with the only noticeable case being written by Y. Li and G.M. Bone in their paper on whether parallel mechanisms are more efficient than serial mechanisms [5.12]. In this section, a method of determining the total input and output power of the 3-RRR planar mechanism to determine the mechanical efficiency of the system is laid out. The equation for mechanical efficiency is shown below in equation (5.4)

$$Efficiency = \frac{actual\ power}{ideal\ power} \times 100\% \tag{5.4}$$

where the actual power of the system is the recorded power at the end-effector of the mechanism and the ideal power is the total input power of the actuated joints of the system, as a 100% efficient system would have no power loss throughout.

To determine the mechanical power of the actuated joints and moving platform equation (5.5) can be used.

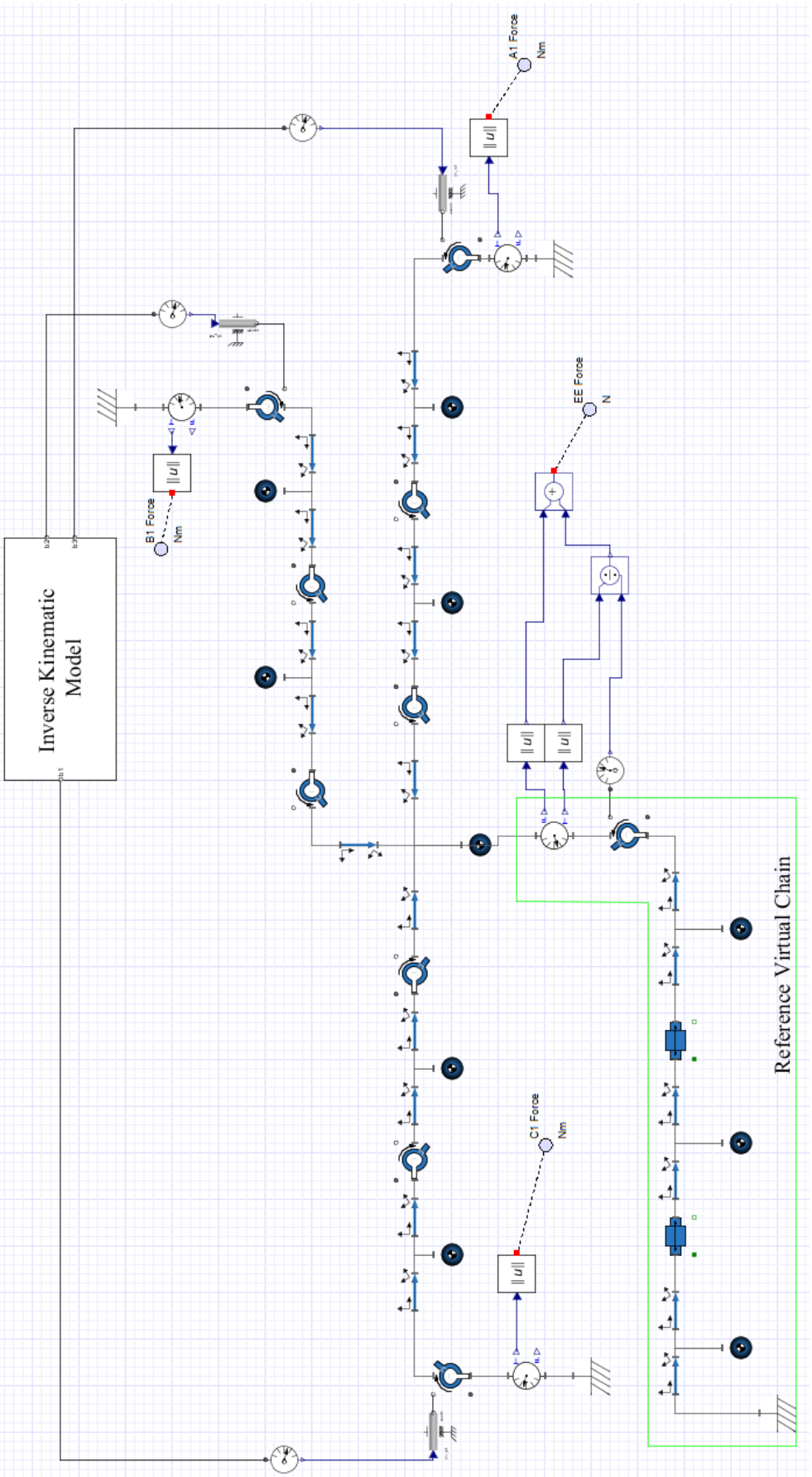

Figure 5.23: MapleSim Inverse Dynamic Model of the 3-*RRR* planar parallel mechanism.

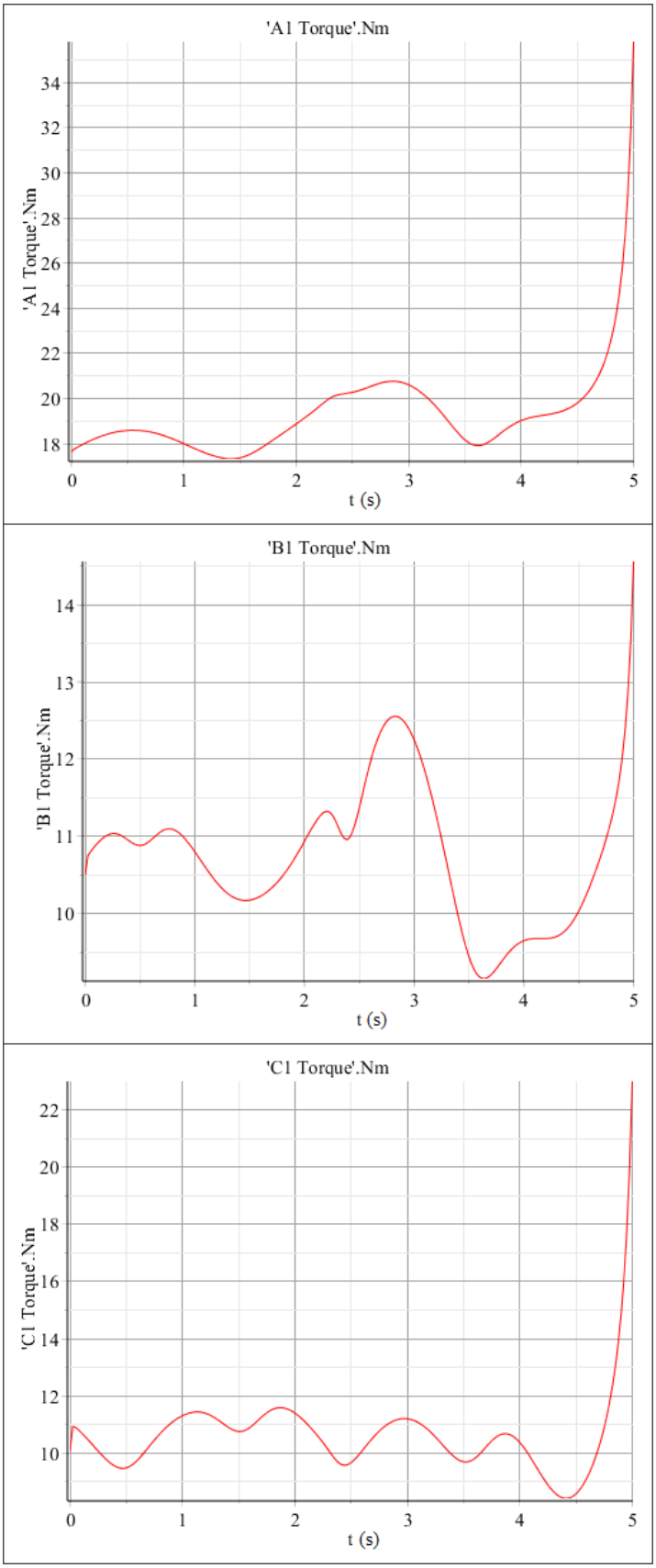

Figure 5.24: Force values for the actuators of 3-RRR planar parallel mechanism.

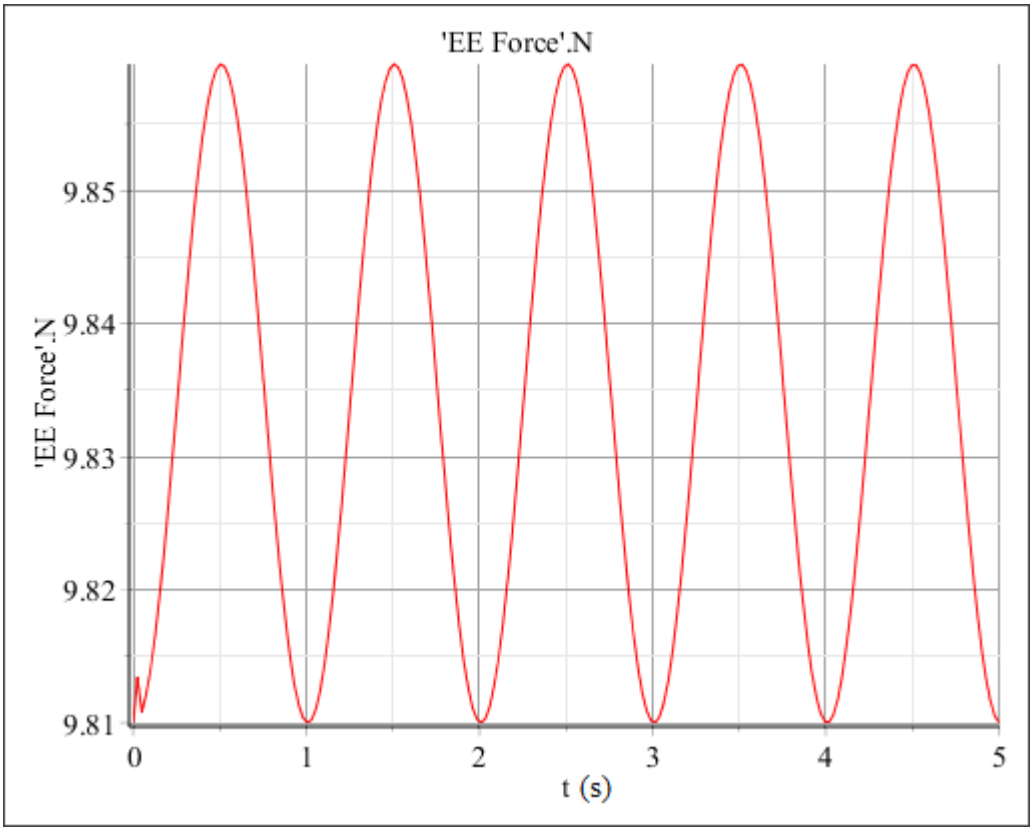

Figure 5.25: Force values for the moving platforms of the 3-*RRR* planar parallel mechanism.

$$P(t) = \frac{W}{t} \tag{5.5}$$

where W is the work done, and t is the time taken to complete the work. As work is calculated as force applied over a distance, the equation becomes

$$P(t) = \frac{F \cdot d}{t} = \vec{F} \cdot \vec{v} \tag{5.6}$$

where F is the force applied, d is the distance moved, and v is the velocity of the object during motion. For torques this equation becomes:

$$P(t) = \tau \cdot \omega \tag{5.7}$$

where τ is the torque applied and ω is the angular velocity of the object during motion.

To calculate the efficiency of the MapleSim model, the inverse dynamic model is modified to identify the angular velocity values for the actuated joints of the mechanism. This is done by adding 1D rotational "Angular Velocity Sensor" blocks (figure 5.27) to the revolute joints of the secondary mechanism.

The torque values recorded from the actuated joints of the secondary mechanism are passed through an "Absolute Vector" signal block (figure 5.28a) in order to convert the 3x1-force vector to a real signal. The resulting real numerical signal is then passed to the u_1 port of a "Product" component block (figure 5.28b) while the real numerical signal from the angular velocity sensor is passed to the u_2 port, resulting in the power required at each actuator. The power values for each actuated joint are passed through absolute

value signal blocks (figure 5.25c) to convert any negative power values, caused by rotations in an anti-clockwise direction, to positive to help determine the full input power.

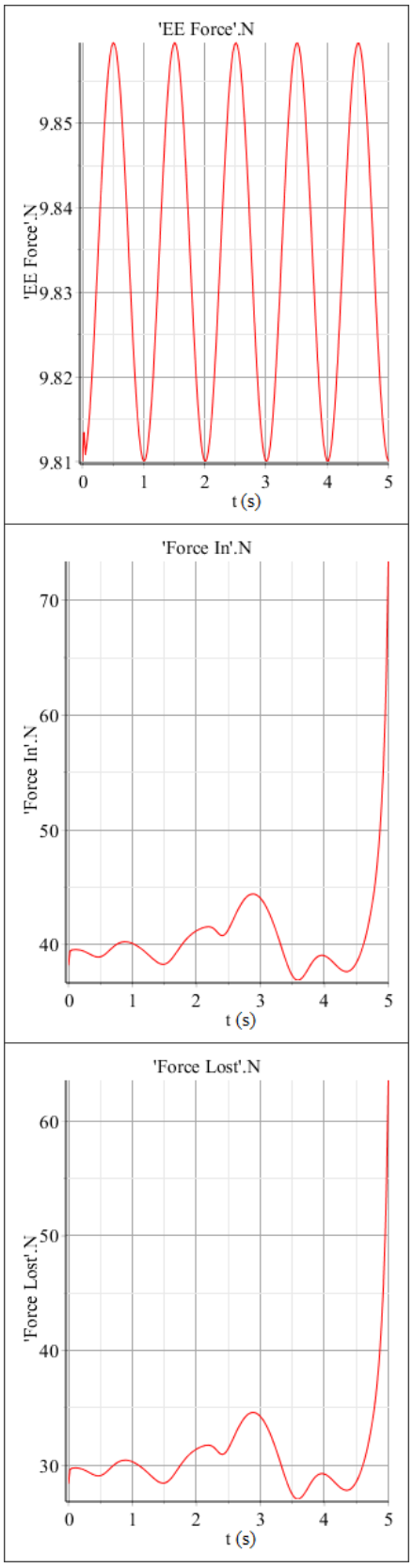

Figure 5.26: Total force lost during the motion of the mechanism.

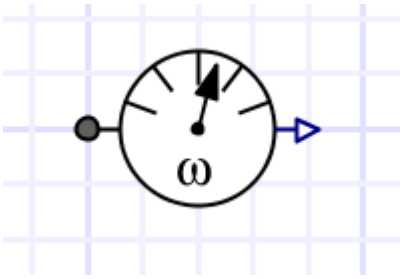

Figure 5.27: 1D rotational "Angular Velocity Sensor".

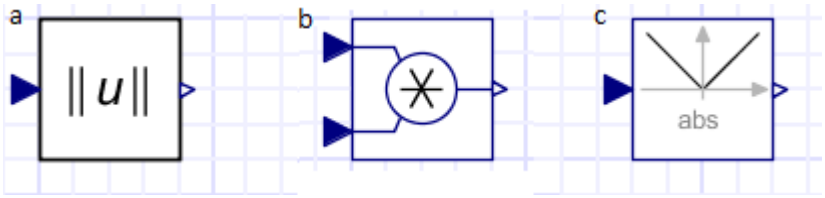

Figure 5.28: a) MapleSim Absolute Vector component block. b) MapleSim Product component block. c) MapleSim Absolute Value component block.

In order to determine the output power of the system a virtual chain, identical to the inverse kinematic model virtual chain, is added to the second mechanism by connecting the R joint to the rigid body located at the intended location of the mechanism's end-effector. The joint velocities of this virtual chain represent the velocity of the end-effector as it is manoeuvred throughout the simulation and therefore are combined in a Pythagoras theorem calculation to determine the total end-effector velocity. The force value determined at the end-effector is then combined with the total velocity from the virtual chain in order to calculate the output power.

Finally, the power value from the end-effector is passed to the u_1 port of a "Division" signal component block (figure 5.29) and the total input power of the actuated joints is passed to the u_2 port. The resulting value is then combined with a constant signal block with its k value set to 100 to determine the efficiency of the system.

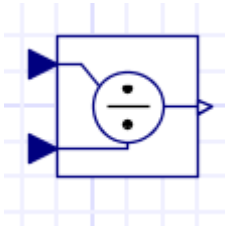

Figure 5.29: MapleSim "Division" signal component block.

The block diagram of for the efficiency study of the 3-RRR planar parallel mechanism is displayed in figure 5.30 and the resulting power in, power out and efficiency are displayed in figure 5.31.

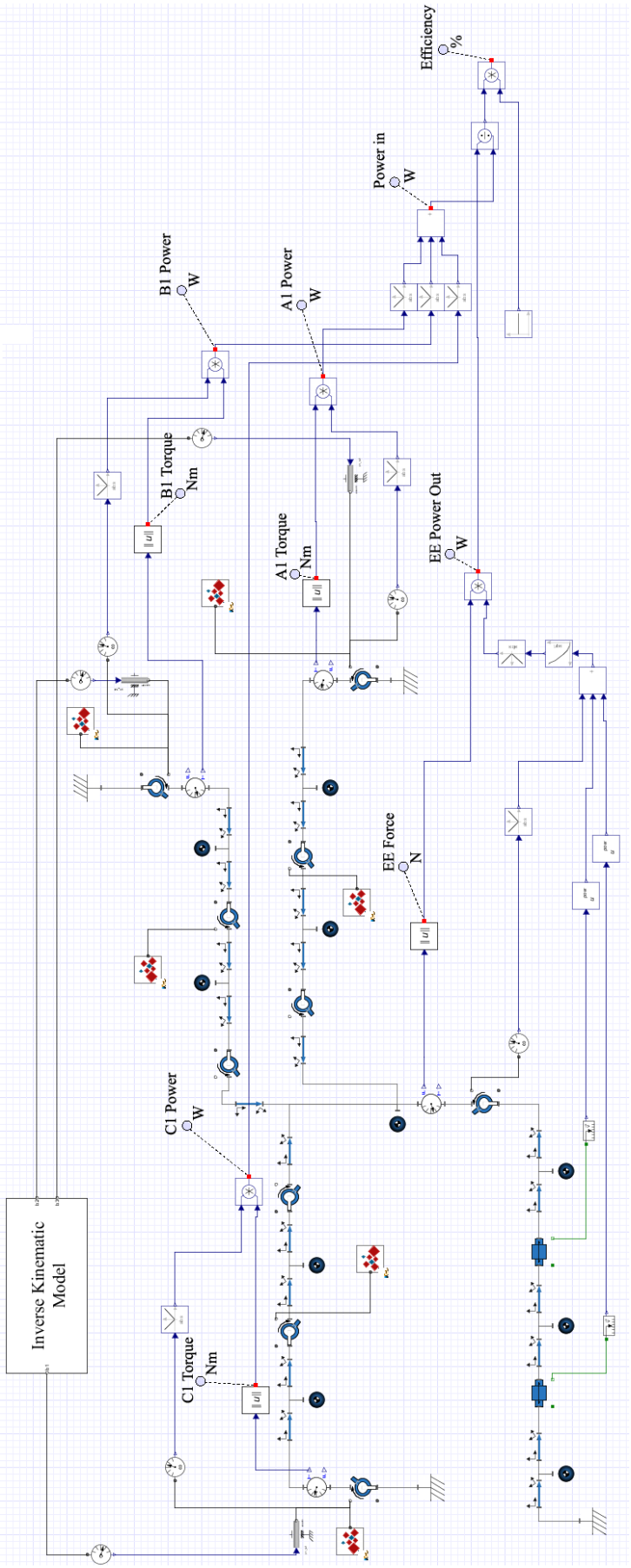

Figure 5.30: Block diagram of efficiency calculations.

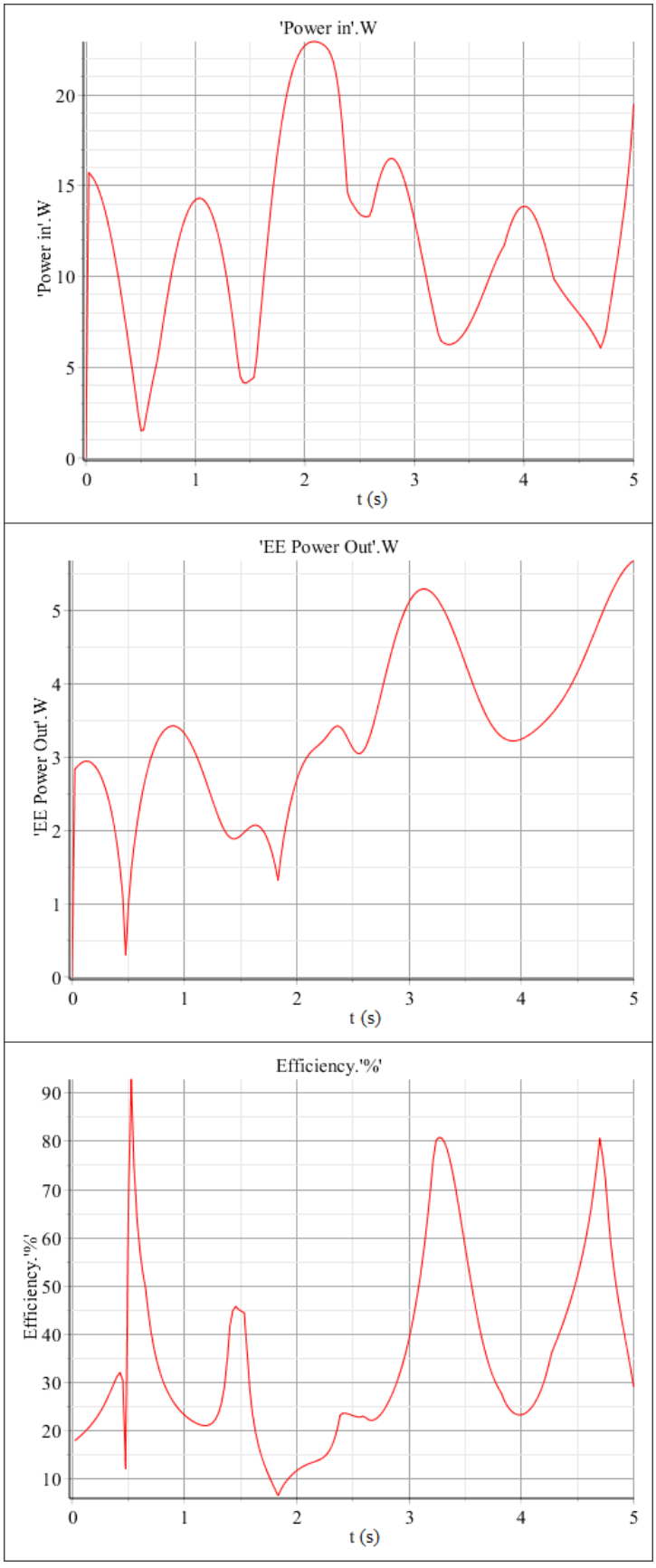

Figure 5.31: Probe data from the efficiency calculation.

The sudden spike at around 0.5 seconds into the motion of the mechanism suggests the presence of a singular configuration where one or more of the actuated joints are applying next to no power in order to manoeuvre the moving platform from its current location. The motion pattern of mechanism was then loaded into the CAD model of the mechanism and run in a motion study by setting up the mechanism with the motor refinement method described in section 4.2.2. The results of this are shown in figure 5.32, which shows the left most kinematic limb entering a singular configuration where the upper portion of the limb crosses above the lower portion of the limb.

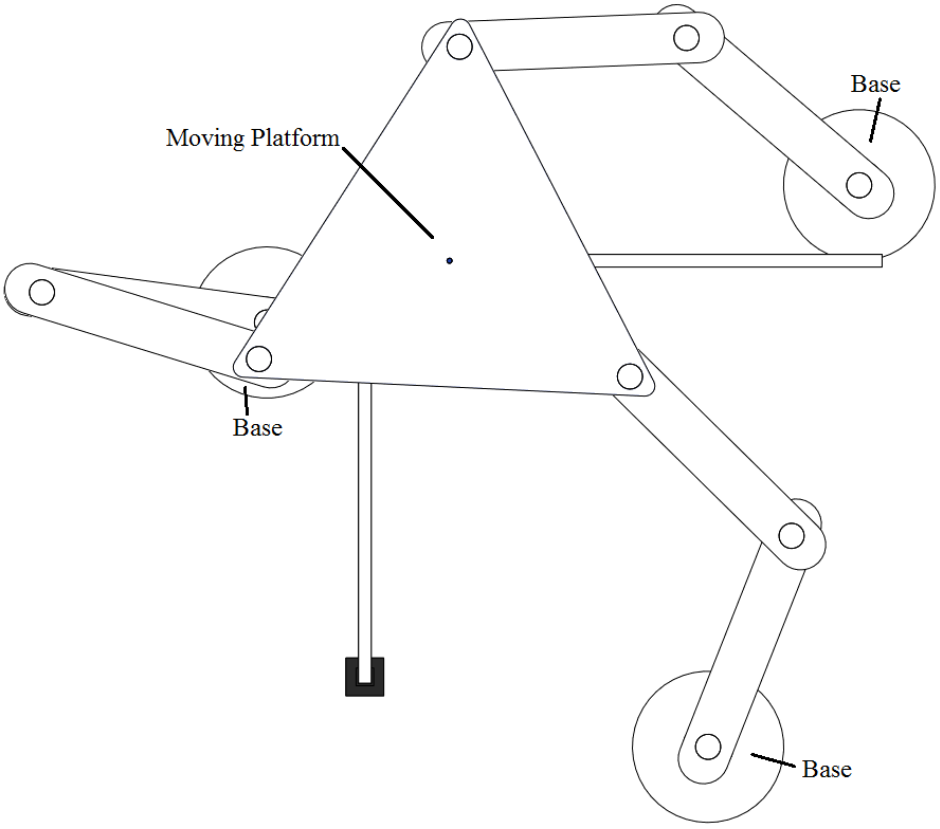

Figure 5.32: CAD rendering of the motion of the 3-*RRR* Planar Parallel mechanism in singular configuration.

The work completed in this section again addresses the second objective of this thesis where virtual chains are utilised to determine a potential singularity in the mechanism's viable workspace in which one of the kinematic chains overlaps itself during the programmed motion. Additionally this section further addresses the fourth objective as the inverse dynamic model is modified to development a method of determining the mechanical energy efficiency of the system.

5.6 Summary

In this chapter, the MapleSim mathematical programming language was fully introduced with its interface detailed. The software was utilised to produce mechanisms and manipulate them using virtual chains to produce the inverse kinematics of a 3-*RRR* planar parallel mechanism.

The method was then further developed in the form of a case study in which the inverse dynamic analysis of the mechanism was produced. This case study detailed how the

forces of the mechanism can be determined by producing an identical version of the mechanism in the same model workspace and controlling its actuated joints by recording the motion of the inverse kinematic model mechanism model and converting the motion into force to drive the second mechanism.

Finally, the inverse dynamic model was further developed to allow the efficiency of the mechanism to be determined by converting the force/torque values into their power equivalents and comparing the simulated actual power at the end-effector of the mechanism to that of the ideal power of the system applied across the mechanism's actuated joints. From these results it can be stated that the proposed method for determining the inverse kinematic and inverse dynamic models of a system can be done without needing to have the advanced engineering knowledge base required in the design process. Additionally the proposed method of producing the mechanical energy efficiency of a parallel mechanism has been validated as the work by Liu [5.12] shows that the average efficiency of a parallel mechanism would be similar to the values gained from the energy efficiency study.

These methods are further utilised in chapter 7 in the production of a physical mechanism prototype and the inverse kinematic model generation method further utilised in chapter 6 in order to produce a control system in Labview.

This chapter addressed objectives 3 and 4 of this thesis as the inverse kinematic model, inverse dynamic model and energy efficiency of a 3-RRR planar parallel robot were developed utilising virtual chains as a forward kinematic model of the systems motion pattern.

References

[5.1] MapleSoft Inc. website, "MapleSim Component Library Overview", http://www.maplesoft.com/support/help/MapleSim/view.aspx?path=componentLibrary/libraryOverview.

[5.2] O. Hamdoun. "Inverse kinematic Modelling of 3-RRR Parallel Robot", 22ème congrès français de Mécanique, August 2015.

[5.3] O. Company, F. Marquet and F. Pierrot, "A New High-Speed 4-DOF Parallel Robot Synthesis and Modelling Issues" Robotics and Automation, vol. 19, No. 3, June 2003.

[5.4] B. Dasgupta and P. Choudhury, "A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators", Mechanism and Machine Theory, vol. 34, pp 801-824, 1999.

[5.5] W. Khalil and O. Ibrahim, "General Solution for the Dynamic Modelling of Parallel Robots", Proceedings of the IEEE International Conference on Robotics and Automation, 2004.

[5.6] W.Q.D. Do and D.C.H. Yang, "Inverse dynamic analysis and simulation of a platform type of robot", Journal of Robotic Systems, vol. 5, no. 3, pp 209-227, 1988.

[5.7] F. Pierrot, P. Dauchez and A. Fournier, "HEXA: a fast six-DOF fully-parallel robot", Advanced Robotics, vol. 2, pp 1158-1163, 1991.

[5.8] M.A. Laribi, L. Romdhane and S. Zeghloul, “Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace”, Machine and Machine Theory, vol. 42, pp 859-870, 2007.

[5.9] K. Miller, “Optimal Design and Modelling of Spatial Parallel Manipulators”, The International Journal of Robotics Research, vol. 23, no. 2, pp 127-140, 2004.

[5.10] M.-J. Liu, C.-X. Li and C.-N. Li, “Dynamic Analysis of the Gough-Stewart Platform Manipulator”, Robotics and Automation, vol. 16, no. 1, pp 94-98, 2000.

[5.11] A. Sokolov and P. Xirouchakis “Dynamics analysis of a 3-DOF parallel manipulator with R P S Joint Structure”, Mechanism and Machine Theory, vol. 42, pp 541 – 557, 2007.

[5.12] Y. Li and G.M. Bone, “Are Parallel Manipulators More Energy Efficient?”, Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2001.

[5.13] St. Staicu, and D.C. Carp-Ciocardia, "Dynamic Analysis of Clavel's Delta Parallel Robot", Proceedings of the IEEE International Conference on Robotics & Automation, vol. 3, pp. 4116-4121, 2003.

[5.14] Y. Li, and Q. Xu, "Dynamic Analysis of a Modified DELTA Parallel Robot for Cardiopulmonary Resuscitation", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 223-238, 2005.

[5.15] St. Staicu, D. Zhang, and R. Rugescu, "Dynamic modelling of a 3-DOF parallel manipulator using recursive matrix relations", Robotica, vol. 24, pp. 125-130, 2005.

[5.16] S. Kucuk, "Inverse Dynamics of RRR Fully Planar Parallel Mechanism Using DH Method", Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization, pp. 3-18, 2012.

Chapter 6 Inverse kinematics based parallel robot control

In this chapter, the methodology of developing an inverse kinematic model of a mechanism is further developed to inform and produce a viable control system. This concept focuses on the MapleSim to Labview connector add-on in order to create a custom simulation block in Labview to control a mechanism.

6.1 Introduction

To achieve the high speeds, acceleration and accuracy expected of parallel robots, advanced model based controllers are designed and implemented to ensure that such robots perform to the required levels for each task [6.1].

In general, typical robot control systems consist of a trajectory generator, an inverse kinematic model (IKM), the controller and the robot itself usually with some form of feedback to the controller reporting the current joint position error to allow for adjustment; this concept is detailed in figure 6.1 below.

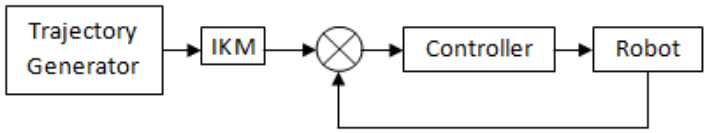

Figure 6.1: Typical Robot Control System.

However, there is also a need to see the position of the robotic mechanism. Therefore, in addition to the control system in figure 6.1, a Direct or Forward Kinematic Model (DKM or FKM) and Graphical User Interface (GUI) are added at the node where the robot relays the position error to the controller so that the robot movement can be observed. The addition of these components allows the user to input locations of target positions to which the robot must move via the GUI, thus giving the user direct control over the robot as shown in figure 6.2 [6.2].

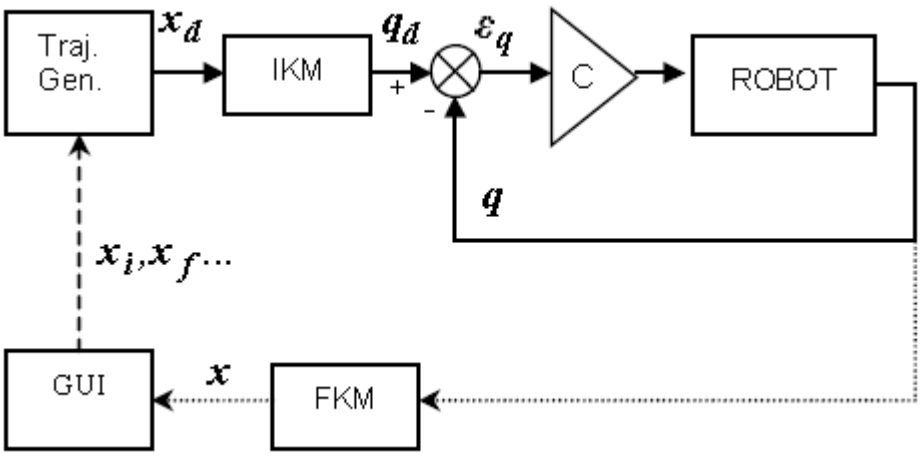

Figure 6.2: Control system of a typical robot [6.2].

When working with parallel mechanisms, errors in the non-actuated joints can occur due to a number of issues such as control errors where the position of the robot and the

requested position match, kinematic model errors usually caused by inaccurate calibration due to the design not perfectly matching the physical robot, or higher order errors caused by deformation in the robot due to thermal and/or mechanical faults [6.2] which can cause it not to reach the desired location. These errors do not appear to affect the standard control models for a parallel manipulator as the non-actuated joint angles go unrecorded.

In order to prevent these errors, redundant sensors can be placed at choice locations along each limb on the non-actuated joints [6.2] providing that they do not interfere with the robot's range of motion or its ability to operate.

As shown in figure 6.3, to solve kinematic model and higher order errors an additional kinematic model based on the redundant sensor readings can produce a model output showing the desired locations.

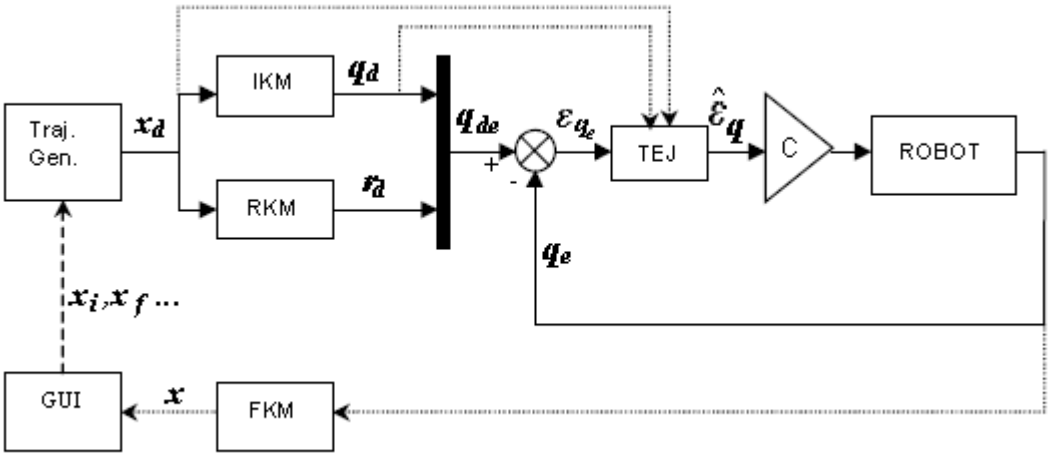

Figure 6.3: Control system of a typical robot with a redundant kinematic model [6.2].

From this, the trajectory generator produces a single co-ordinate location for the end effector that is then fed into the IKM and the RKM, which each subsequently produce a model of where the desired joint angles should be located. These model outputs are then added together to produce a single signal which generates a final trajectory that is fed into the controller and moves the robot's actuators.

For the purposes of this chapter, the IKM of the control system is implemented using the MapleSim methodology detailed in Chapter 5 and the controller block and feedback loop implemented using a Labview-based approach.

6.2 Labview Virtual Instruments

Labview virtual instruments (VIs) [6.3] are designed using the software's two panel system windows: the front panel for producing a GUI and the block diagram panel for building up the VI's code. When building a VI the user can opt to either: (i) set up a GUI prototype on the front panel by placing intractable virtual devices such as numerical inputs, switches, and probes (figure 6.4). Allowing the user to manipulate the program and observe the data passing through the system in one form or another (figure 6.5); or (ii) start by building up the background code of the VI through the use of pre-programmed blocks in the block diagram panel.

The GUI interface screen starts as a blank grey grid window, which allows for a particular family of programming blocks, known as control blocks to be placed upon it. Once placed, these can be manipulated to work within the functional criteria required by the user.

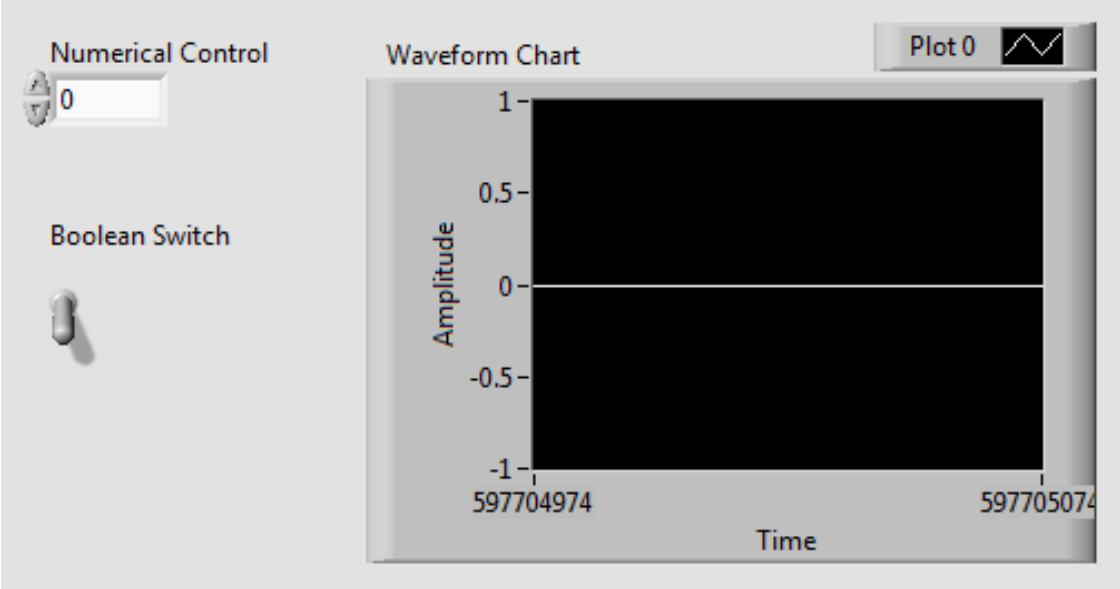

Figure 6.4: Labview front panel GUI example.

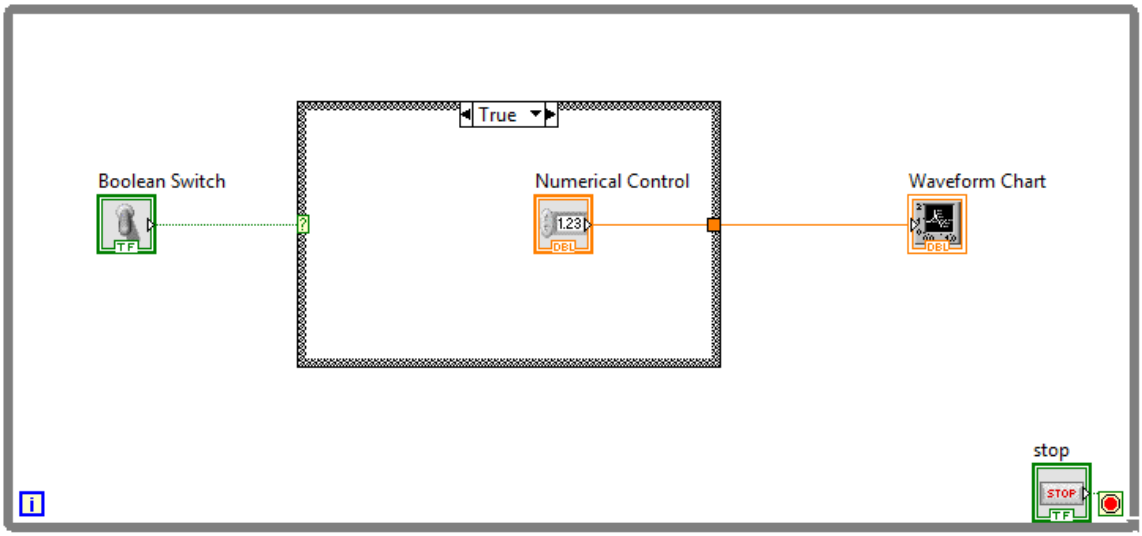

Figure 6.5: Example Labview block diagram program

6.2.1 Labview data types

In Labview, three types of data can be passed or manipulated in the block diagram using control blocks, these are numerical, string and Boolean. The controls and indicators associated with each type are explained below:

- Numerical Controls and Indicators [6.4]: Numerical data is the overreaching data type whether it is real numbers, integers, floats or double numbers, etc. These data types are represented through blue (integer) or orange (float/double) connecting wires, blocks and ports in the block diagram and are typically manipulated through control blocks in numerous forms such as numerical sliders,

increment/decrement buttons and analogue dials (figure 6.6). This type of data can be typically indicated as a numerical value or as a signal wave using numerical indicators such as a scope block or a numeric indicator window (figure 6.7).

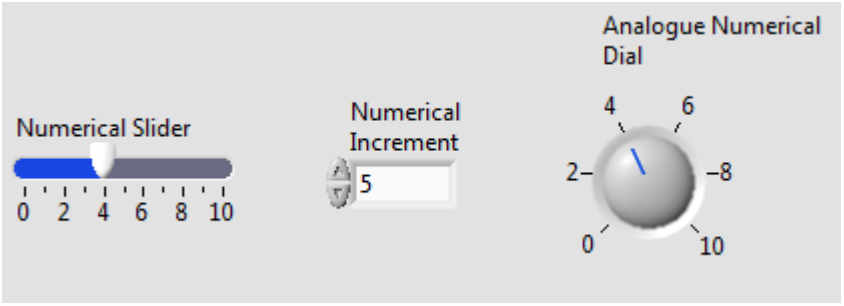

Figure 6.6: Labview front panel numerical control examples.

Figure 6.7: Labview front panel numerical indicator examples.

- String Controls and Indicators [6.5]: String data is the data type that uses characters and numbers that have no other meaning other than the order in which they are written. This type of data is typically used for user names and passwords for a more secure program but can also be used for other tasks. The string data type is represented in the interface through pink connecting wires, blocks and ports and is controlled using text boxes or tables that allow the user to input string data by selecting the box and typing in commands or information (figure 6.8). Output string data is also usually indicated once again by using a text box or file path that has been set-up to have a read only function (figure 6.9).

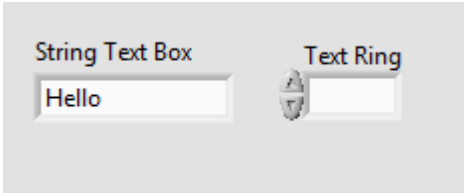

Figure 6.8: Labview front panel string control examples.

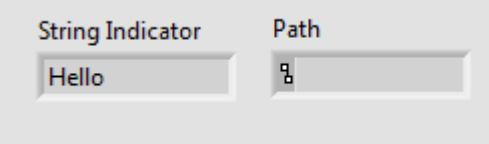

Figure 6.9: Labview front panel string indicator examples.

- Boolean Controls and Indicators [6.6]: The final type is Boolean data, also known as the high/low data in which the value passed is either a 1 or a 0. Boolean data is indicated in the block diagram using green connecting wires, blocks and ports, is controlled using two-position toggle switch control blocks and typically indicated by two state indicators like lights (figure 6.10). Boolean data is typically used to activate a process upon selection, such as selecting a particular subroutine to run or changing between two system states in conditional coding blocks.

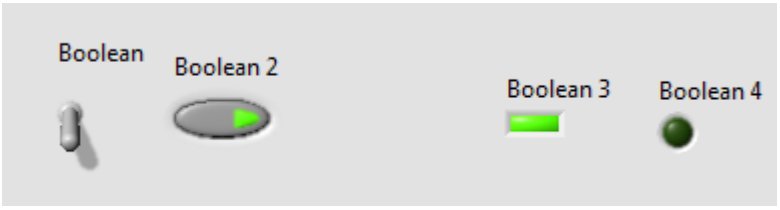

Figure 6.10: Labview front panel Boolean control and indicator examples.

6.2.2 Labview block diagram functions

The block diagram functions are the primary operating elements of the Labview code and can appear as simple mathematical functions like add or subtract or as complex system tools such as pointers to other VIs which operate as subroutines within the code. These function blocks differ from the control blocks on the front panel, as they are not displayed on the front panel GUI.

One of the main function blocks used in Labview programming is the SubVI node that attaches an additional VI to the current VI in a similar manner to a subroutine in text-based code. The node acts as an equivalent to call statements in text based codes, which point to a section of additional code to be run outside the main program any number of times and are typically used to keep the main program, or in this case the VI, tidier and easier to read.

Another aspect of coding which can be utilised in Labview is the ability to produce complex execution structures that allow for the continuous manipulation of data so long as certain requirements are met. These execution structures are utilised in several ways including loops and case structures that allow processes to be repeated several times until a condition is met or to run only in the event of a particular set of criteria being met. The case structure function creates an area in which additional code can be placed

All GUI panel blocks placed on the front panel of the VI automatically place their corresponding block diagram view counterparts onto the block diagram layout. The block diagram view is the section of the VI that resembles the MATLAB Simulink software as well as traditional coding languages, where variables, mathematical functions, I/O devices and signal modifiers are combined together using the wiring system detailed in Chapter 1 to produce a full-bodied code.

6.3 MapleSim to Labview Connector

To test these methods, two case studies are presented focusing on determining the inverse kinematic model of an RRR serial robot arm. The first case study investigates the inverse kinematics of the RRR serial robotic arm by utilising the traditional process for determining the IKM of a mechanism, the results of which are produced using Matlab. The second case study uses the methodology developed in chapter 5 to produce a MapleSim IKM of the mechanism and then, using a virtual chain, determine the required joint angles for the arm's end position.

6.3.1 Traditional process case study

This method looks at the geometry of the mechanism with the user inputting the world coordinates of the mechanism's end-effector (x, y) and its orientation (ϕ); the proof is given in [6.7]. The component parts of the mechanism are as shown in figure 6.11, with the mechanism's end-effector being denoted by "E", the R joints being denoted as having origin locations of O_1, O_2 and O_3.

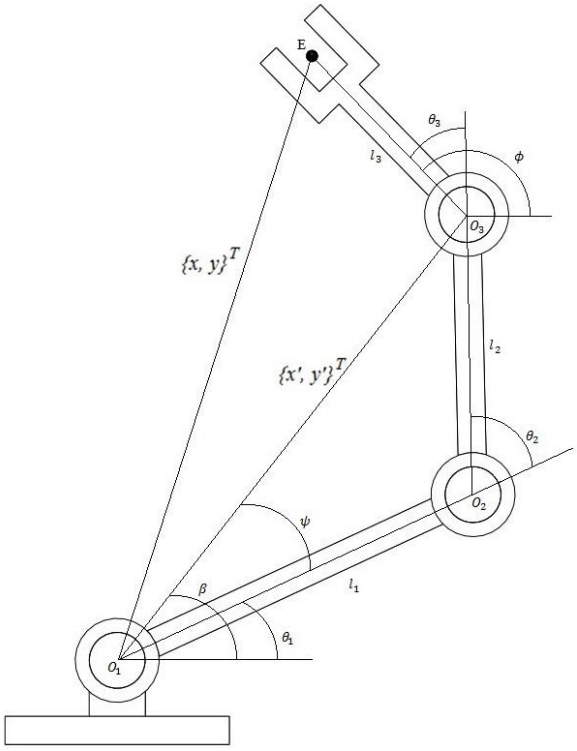

Figure 6.11: RRR serial robot arm

Taking a vector between the origin of the robot at the initial R joint (O_1) to the third R joint (O_3) for the vector elements x' and y' the following equations are obtained:

$$x' = x - l_3 \cos\phi$$
$$y' = y - l_3 \sin\phi \tag{6.1}$$

where l_3 is the length of the third link, ϕ is the angle between the horizontal plane and the rotated plane of the final link and x and y are the vector components between the origin of the mechanism (O_1) and the end-effector.

The next equation utilises the required joint angles of the mechanism to calculate the rotation of the second R joint (O_2):

$$x'^2 + y'^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(180 - \theta_2) \tag{6.2}$$

which can be re arranged to make:

$$\cos\theta_2 = \frac{x'^2 + y'^2 - l_1^2 - l_2^2}{2l_1 l_2} \tag{6.3}$$

This presents only one component of the second R joint's position for future calculations $\sin\theta_2$ is also required, therefore, by utilising Pythagoras theorem:

$$\sin\theta_2 = \pm(1 - \cos^2\theta_2)^{1/2} \tag{6.4}$$

This produces two possible orientations for the second R joint, i.e. the next joint is either above or below the horizontal component of the joint angle (figure 6.12).

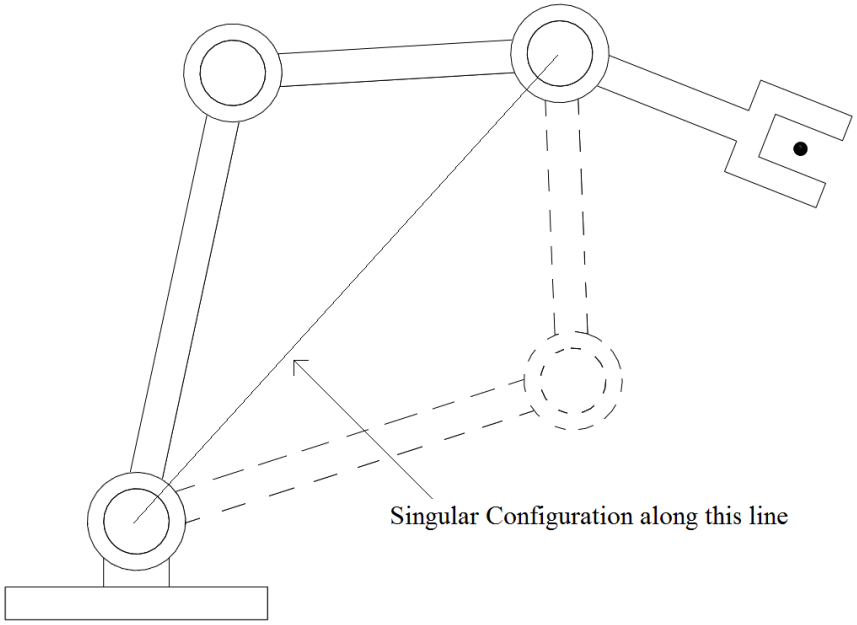

Figure 6.12: An *RRR* serial robot with two orientations of the second joint (adapted from [6.8]).

From figure 6.12, the two linkages and the vector of $\{x' + y'\}$ form a triangle. Using the cosine rule the value for ψ can be found:

$$\cos\psi = \frac{x'^2 + y'^2 + l_1^2 - l_2^2}{2l_1(x'^2 + y'^2)^{1/2}} \tag{6.5}$$

and from the sine rule:

$$\sin\psi = \left(\frac{l_2}{(x'^2 + y'^2)^{1/2}}\right)\sin\theta_2 \tag{6.6}$$

To find the angle of the first R joint, θ_1, the total angle between the horizontal and the vector of $\{x' + y'\}$, β, needs to be calculated:

$$\sin\beta = \frac{y'}{(x'^2 + y'^2)^{1/2}}$$

$$\cos\beta = \frac{x'}{(x'^2 + y'^2)^{1/2}} \tag{6.7}$$

From equations (6.4), (6.5), (6.6) and (6.7) the value of θ_1 can be calculated:

$$\theta_1 = \beta - \psi \tag{6.8}$$

With θ_1 and θ_2 now known, θ_3 can be calculated:

$$\theta_1 + \theta_2 + \theta_3 = \phi$$

$$\therefore \theta_3 = \phi - \theta_1 - \theta_2 \tag{6.9}$$

To test these equations, the values of the mechanism's link lengths and the end position and orientation of the end-effector are used. These are as follows:

- Link 1 (l_1) = 10m
- Link 2 (l_2) = 8m
- Link 3 (l_3) = 5m
- End-effector position vector (x, y) = (16, 5)
- End-effector orientation (ϕ) = $\frac{\pi}{2}$

The equations have been written out in full in Matlab with the above values inputted. The results are shown in Table 6.1 and the resulting mechanism displayed as a Matlab plot in figure 6.13.

Table 6.1: Joint angle values

R Joint	**Angle (°)**
θ_1	-23.44
θ_2	54.901
θ_3	59.244

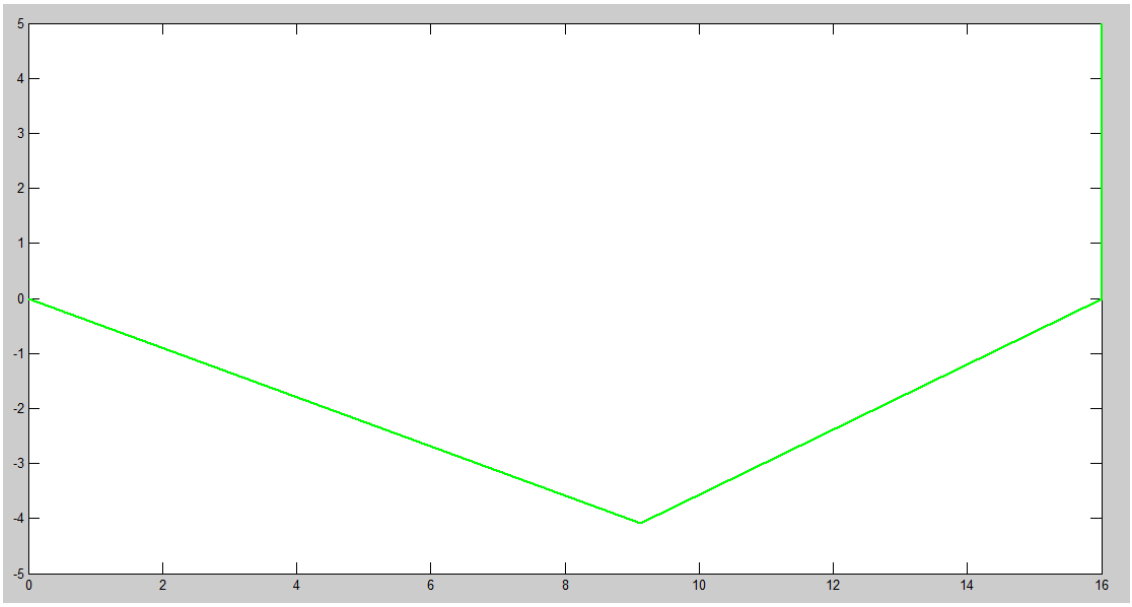

Figure 6.13: Matlab plot representation of mechanism in end location and orientation.

6.3.2 MapleSim inverse kinematic model method case study

The corresponding MapleSim model representation is displayed in figure 6.14 and shows a virtual chain of two prismatic components and one revolute component. The virtual chain's prismatic components act along the vertical axis (y-axis in MapleSim) and along one of the horizontal axes (x-axis); the revolute joint rotates about the remaining horizontal axis (z-axis). The lengths of the mechanisms links are the same as in the previous case and the end-effector position vector values and orientation, set in radians, are fed into their respective virtual joints.

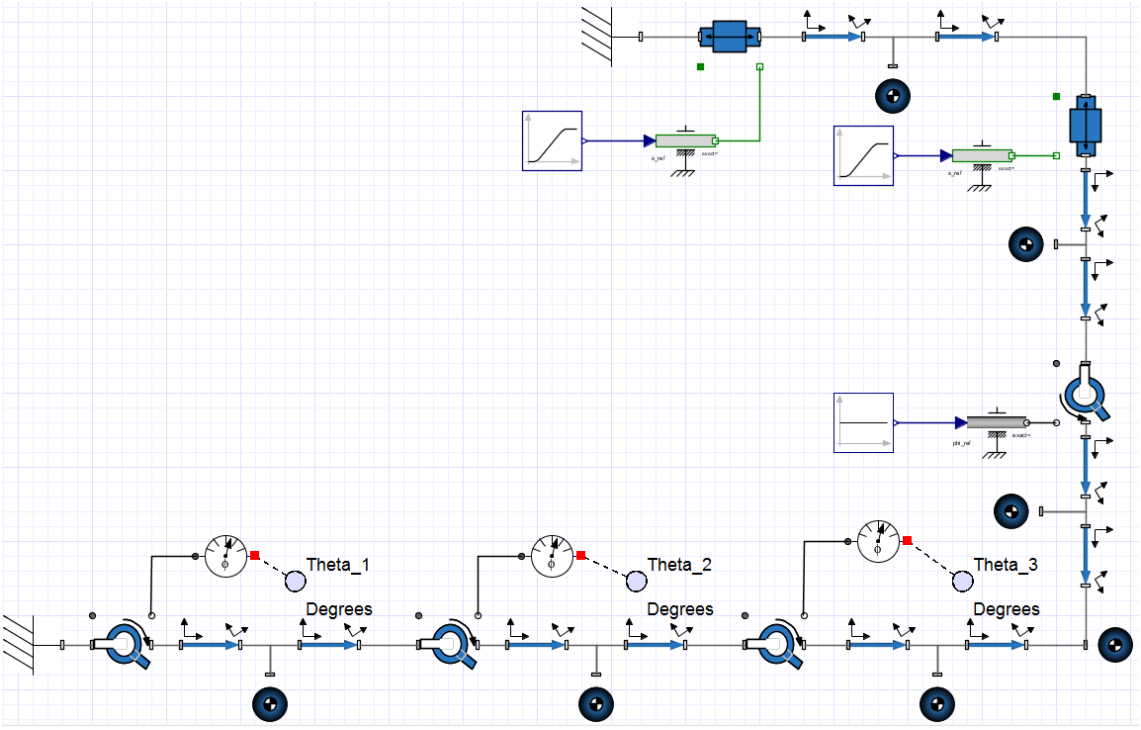

Figure 6.14: MapleSim block diagram of *RRR* serial robot with *PPR* virtual chain.

The rotation values of the mechanism's revolute components are recorded using angle sensor components. The results of the simulation are shown in figure 6.15 and the visual representation of the mechanism shown in figure 6.16.

When compared, the results shown in figures 6.13 and 6.16 show that the values gained through the MapleSim inverse kinematic model and the traditional process are the same, further proving the validity of this process.

6.3.3 MapleSim to Labview connector

The MapleSim connector for LabVIEW and NI VeriStand Software is an add-on component that can be purchased from Maplesoft and is used to export any MapleSim model to Labview, the process is detailed in a video in [6.9]. The MapleSim model is exported as a dynamic link library (.dll) file type to be uploaded to Labview VI using the Exported Model function block found in the Utilities section of the Simulation function block list for the block diagram window. In order to produce a viable MapleSim model for conversion to Labview, the following steps must be followed:

- Step1: Set up the model
- Step 2: Define inputs and outputs
- Step 3: Export the model
- Step 4 Import the model into Labview

Each step is now described in detail below.

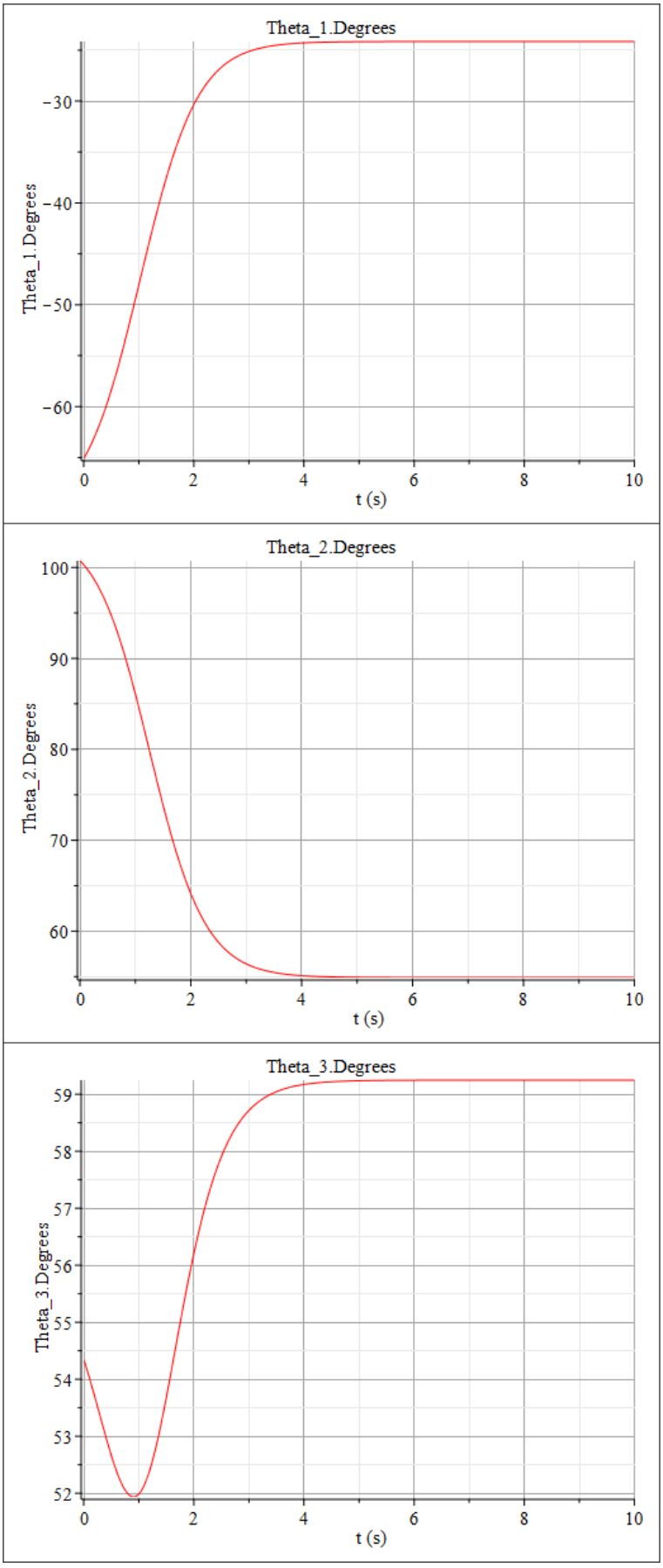

Figure 6.15: Probe data of the mechanisms rotation joints.

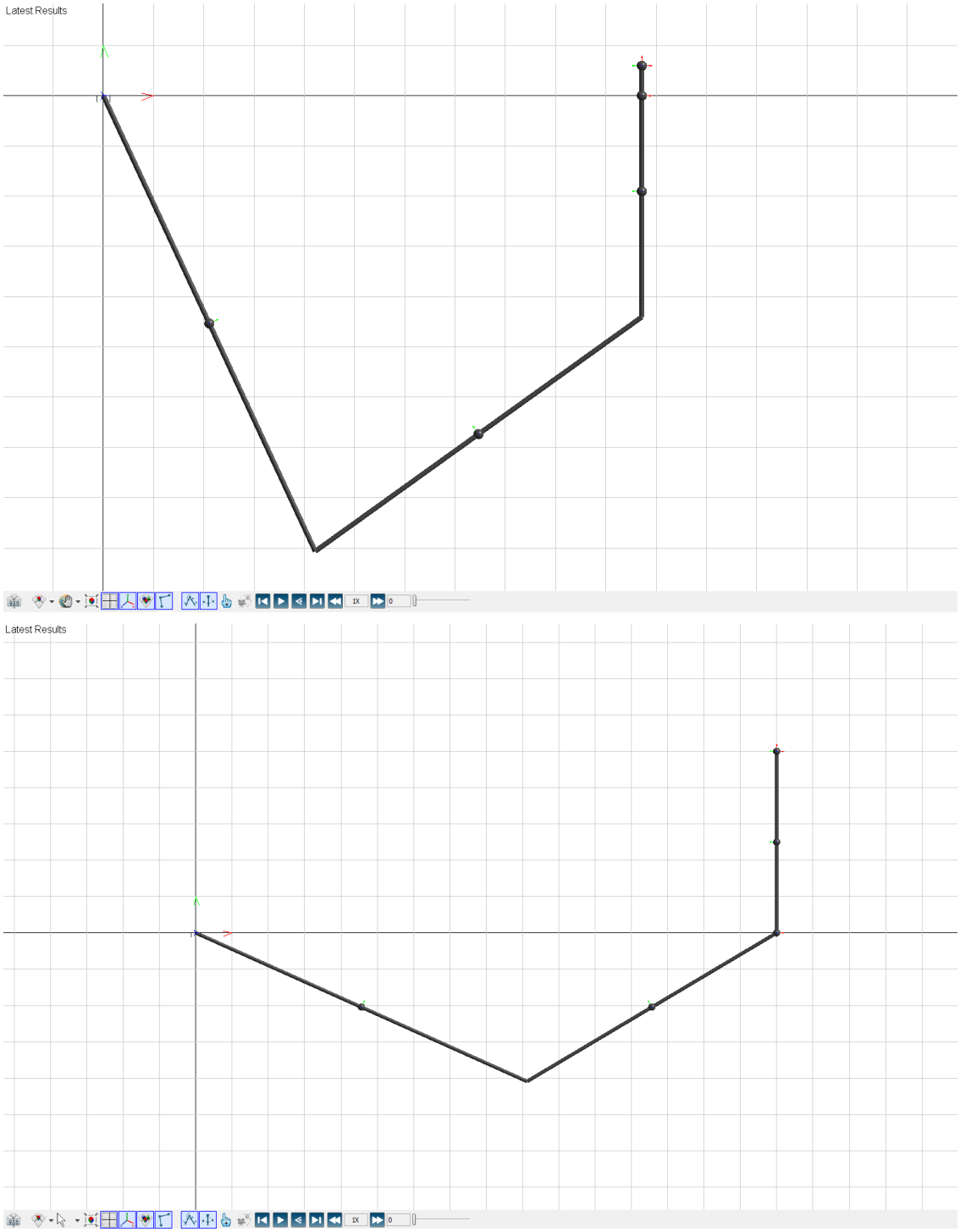

Figure 6.16: 3D visualisation of the serial mechanisms inverse kinematic MapleSim model at the start and end of simulation (1 square = 1m along axis).

6.3.3.1 Step 1: Setup the model.

The first step in preparing the MapleSim model for being exported to Labview is to convert the Labview model to a subgroup of the MapleSim model file. This is done by selecting all of the model's components that represent the physical form of the mechanism and any control aspects included in the model (figure 6.17). Using the hotkey shortcut Shift + G, a small window will appear asking for a name to identify the subgroup to be formed (figure 6.18). Once entered, the selected components are enclosed into a subgroup

component block. By double clicking on the component block, the subgroup can be accessed, allowing modifications to be made to the model of the mechanism (Figure 6.19).

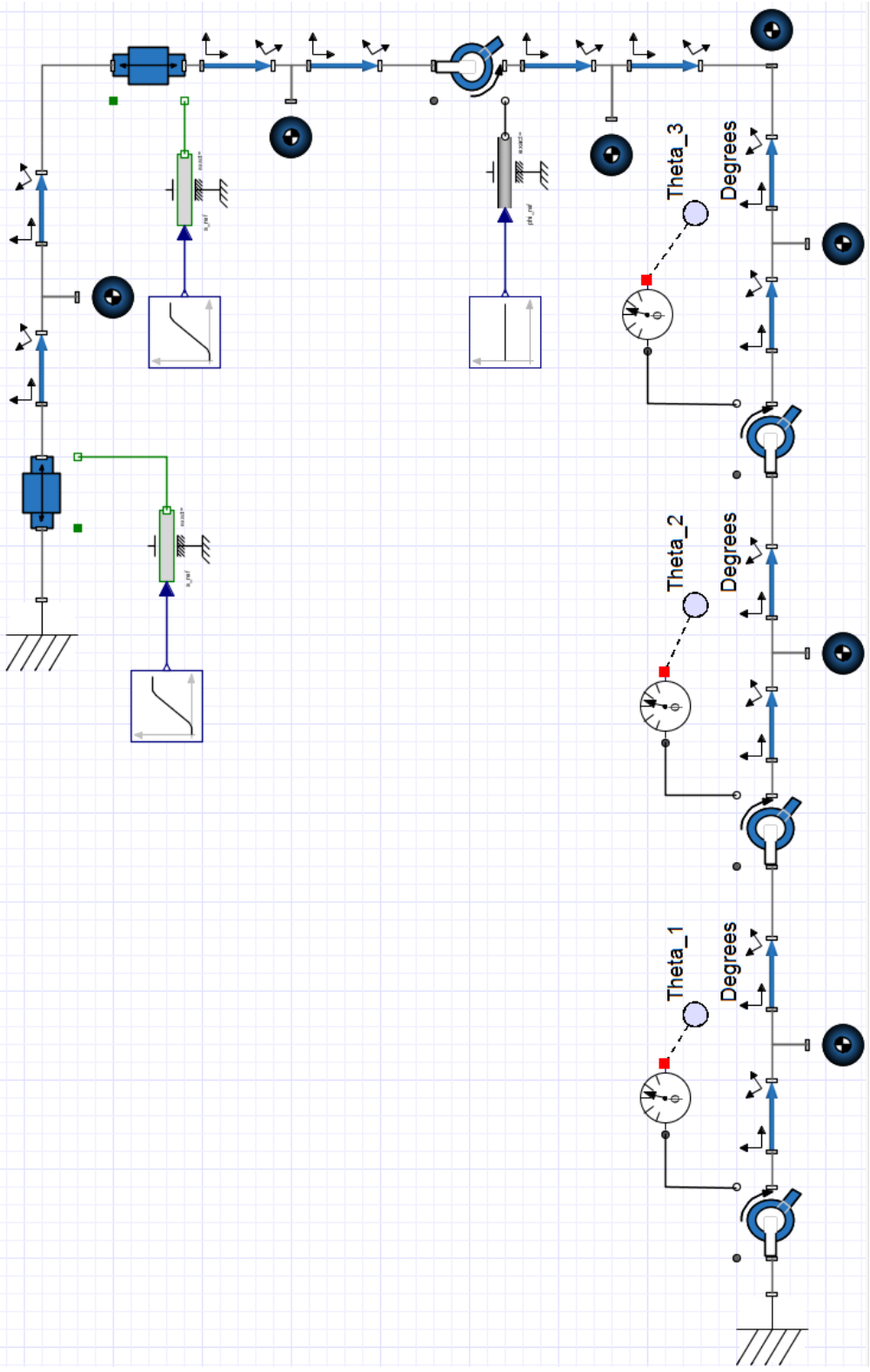

Figure 6.17: MapleSim representation of the mechanism.

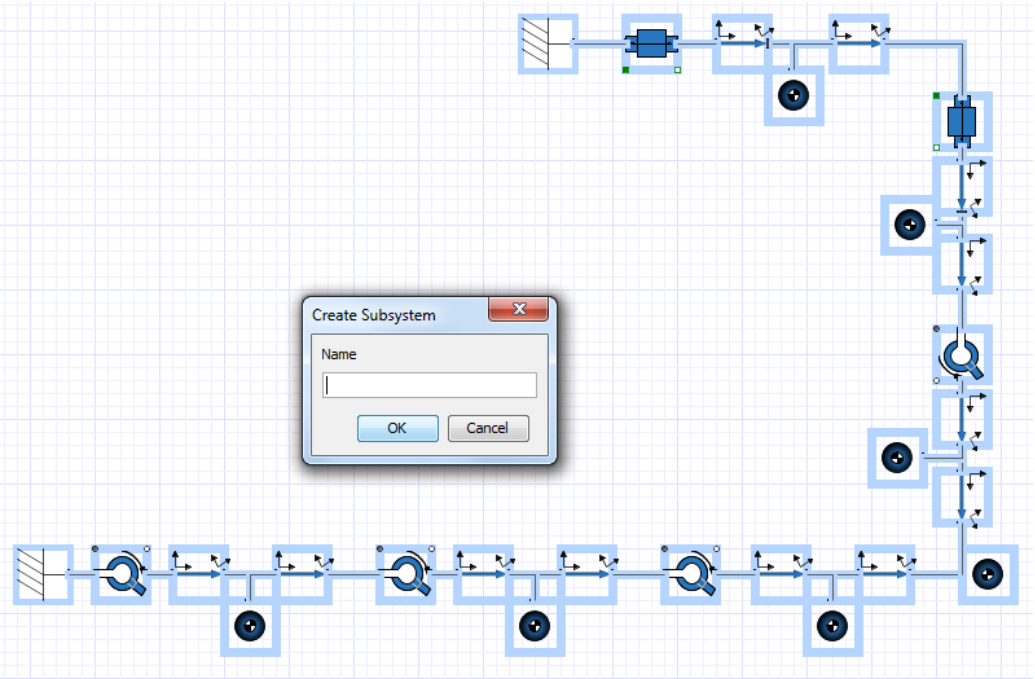

Figure 6.18: MapleSim window for creating a Subgroup.

6.3.3.2 Step 2: Defining inputs and outputs

With the mechanism enclosed in a subgroup the inputs and outputs of the block are now required; these will be replaced later by Labview components. Output components are traditionally in the form of real numerical values that represent various aspects of the MapleSim model; typical examples of outputs are position or angle values for the P and R joints, and required torque or force values of actuators. Input components tend to be the main control values of the system, usually in the form of signals for motion drivers or power curves for motors. To place an input or output node into the subgroup the main control and sensor components need to be placed into the subgroup model at the appropriate places (figure 6.20).

From these components, the control component's inboard nodes and outboard nodes can be connected to the dotted exterior of the subgroup (figure 6.21). The subgroup now has the desired inboard and outboard nodes visible on the subgroup component block and can now be given meaningful names, which aids in the construction of the model when connected to the inputs components and output probes (figure 6.22).

It should be noted that this step could be avoided if, in creating the model, the inputs, control components, sensors and output systems have all been put in place before the model is converted into a subgroup. In this scenario, the model should be selected along with all other components excluding any existing input and output components.

6.3.3.3 Step 3: Exporting the model

Using the templates menu in the main toolbar of the MapleSim window, the NI LabVIEW EMI Block Generation template option is selected to begin the process of exporting the constructed model. Selecting this option produces a "MapleSim custom component block creation" setup window to complete the process of producing the Labview block (figure 6.23).

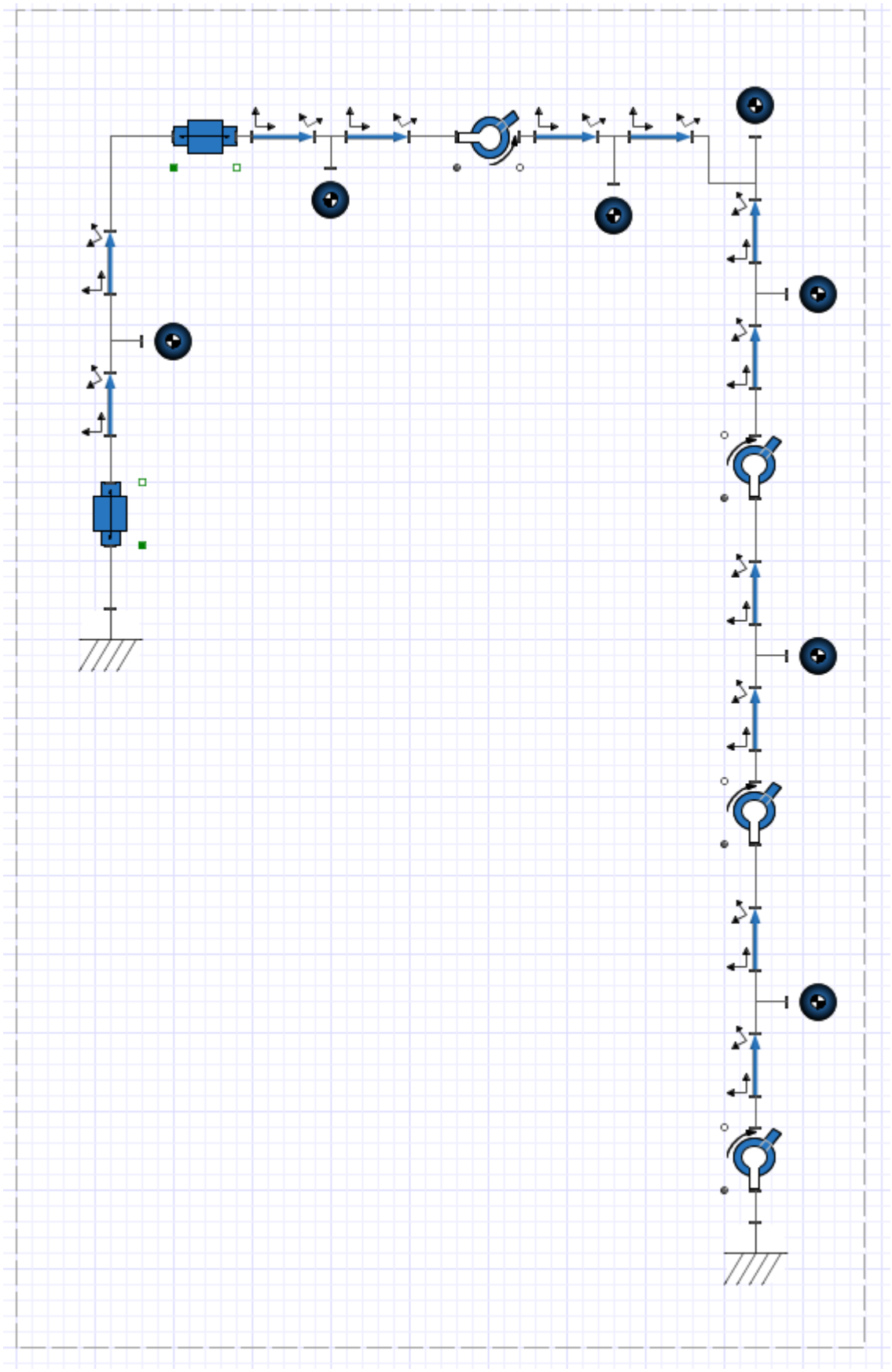

Figure 6.19: MapleSim model in subgroup level.

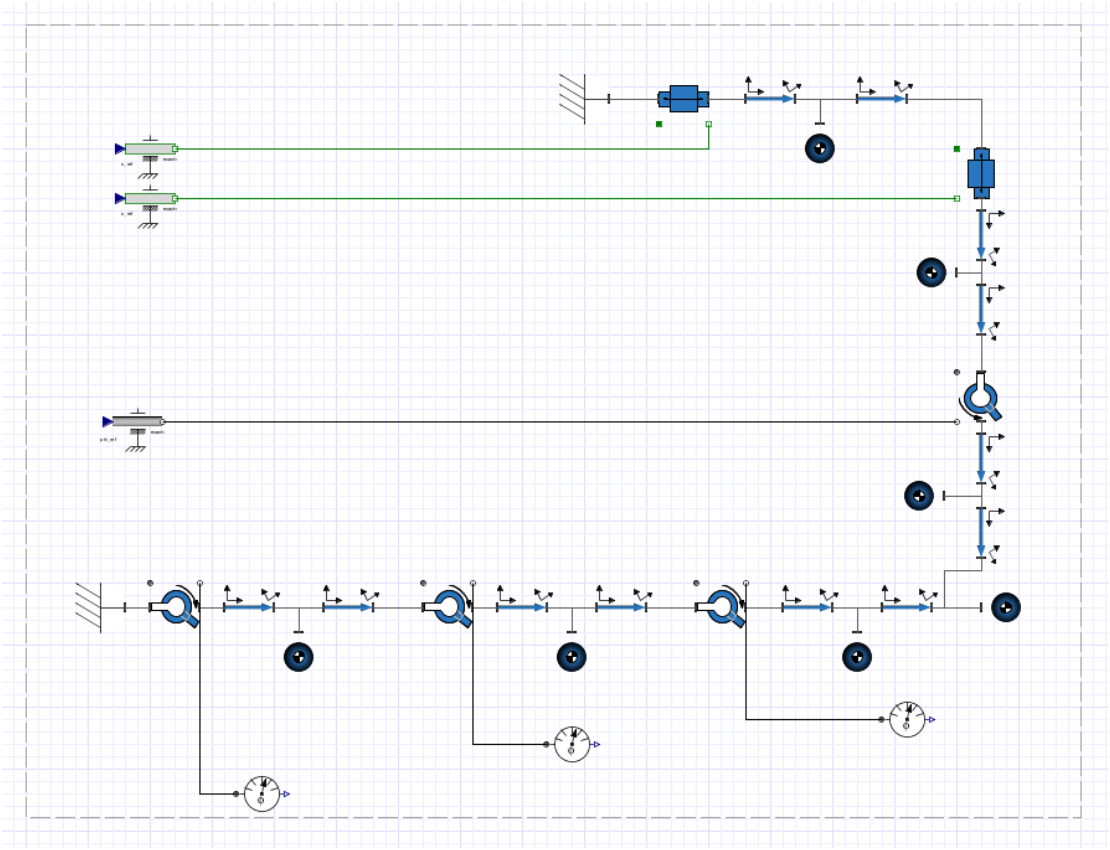

Figure 6.20: Subgroup mechanism with input motion drivers and output sensors

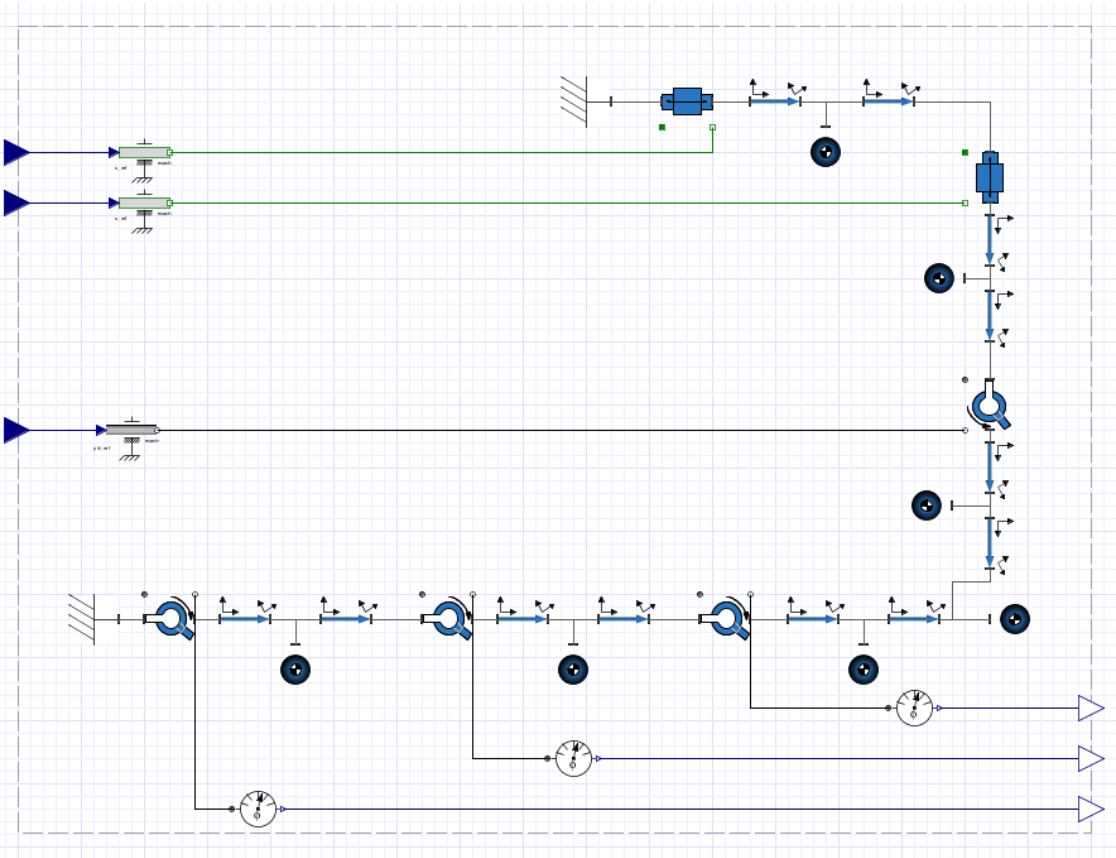

Figure 6.21: Subgroup inputs and outputs added to block diagram

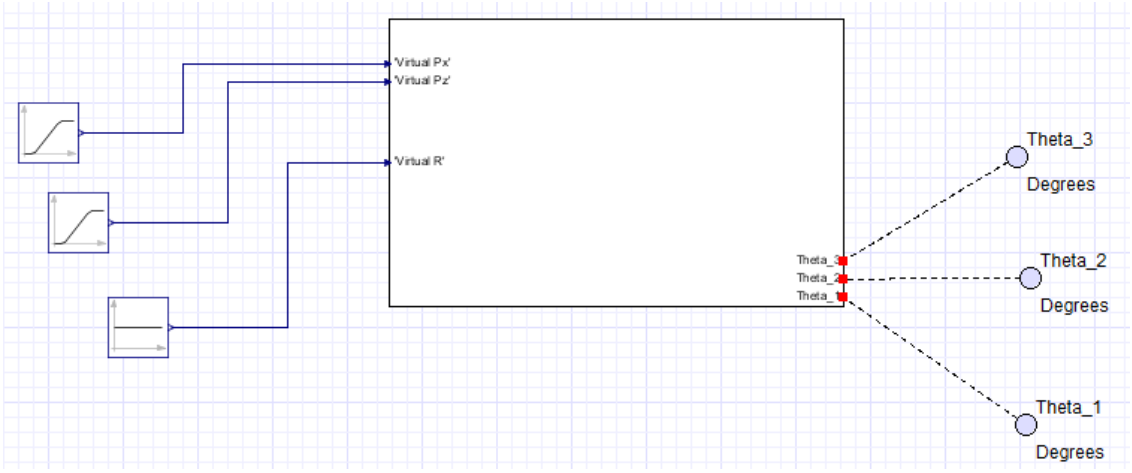

Figure 6.22: Subgroup with meaningful names and control/probe elements for testing

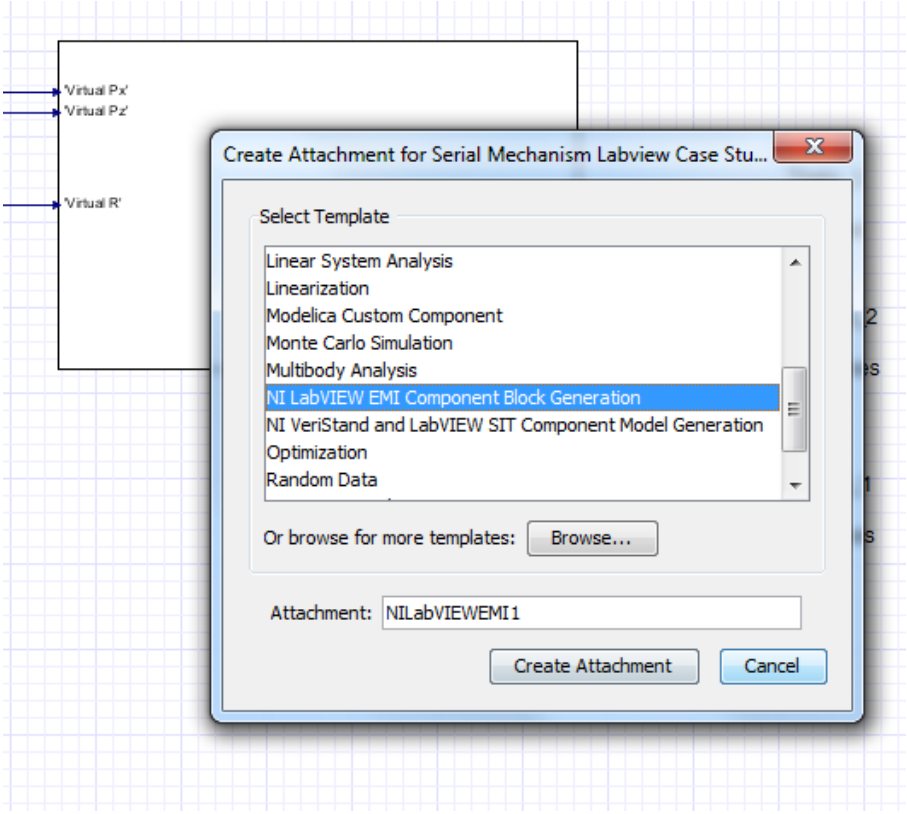

Figure 6.23: Select the NI LabVIEW EMI Component Block Generation option

Inside of the model panel, make sure that the subgroup block is selected and press the Retrieve System button in order to populate the input and output fields of the template. Next, the directories for the block generation, LabVIEW and the Visual C++ directories need to be assigned so that the block is created in the correct location, and all of the coding libraries and components required to generate the block can be found while the LabVIEW component is generated (figure 6.24). The final step in this section is to click the Generate to LabVIEW button at the bottom of the window, which will produce the Visual Studio project and dynamic link library file (.dll) for the LabVIEW EMI block.

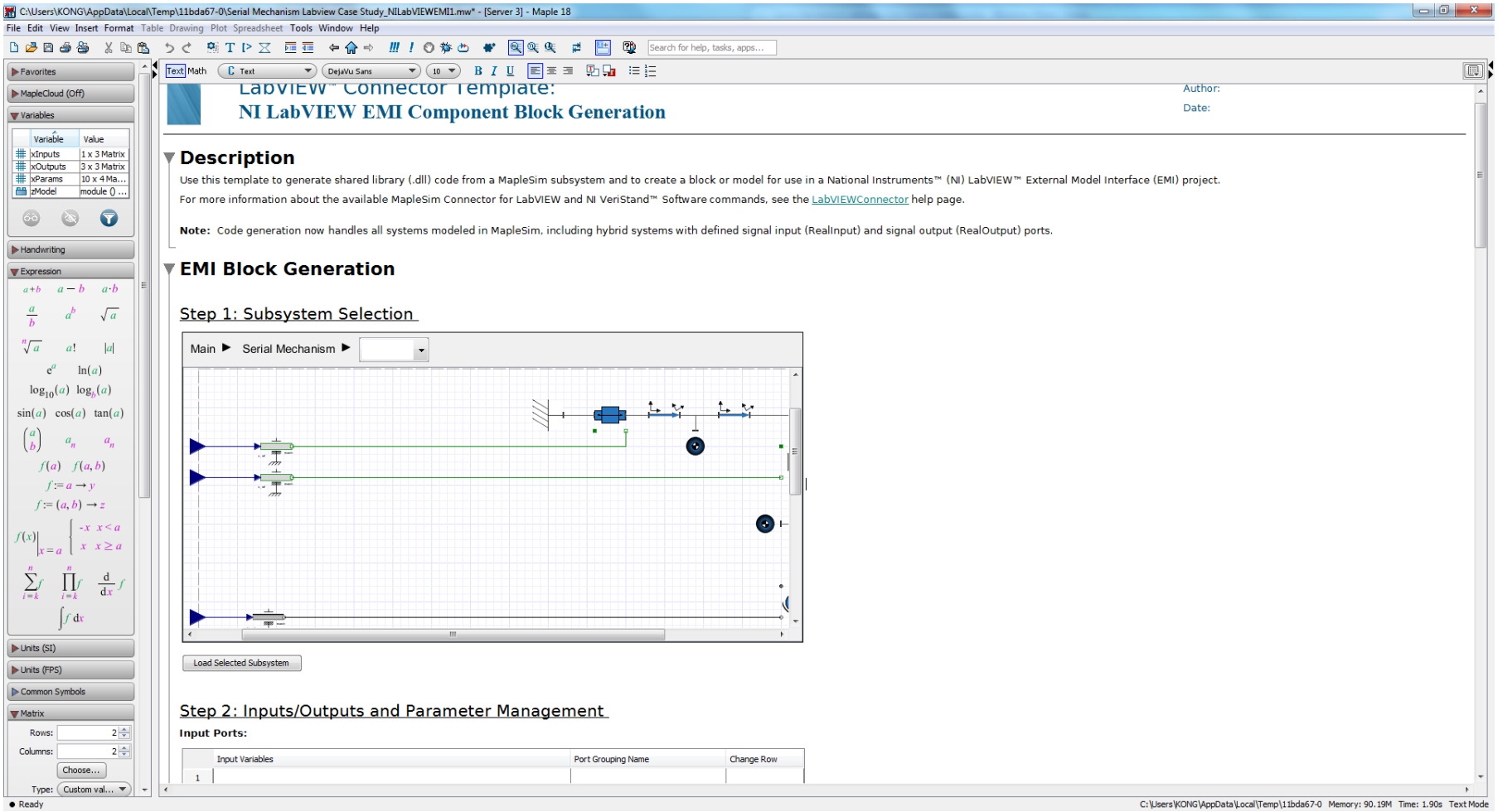

Figure 6.24: EMI Block Generation window

6.3.3.4 Step 4: Importing the EMI block into LabVIEW

In LabVIEW, open a new VI and access the block window view of the project. In the functions menu, select the Control and Simulation Loop component from the Simulations section and draw a simulation loop box that will eventually house the MapleSim model and allow it to run repeatedly while the VI is active (figure 6.25).

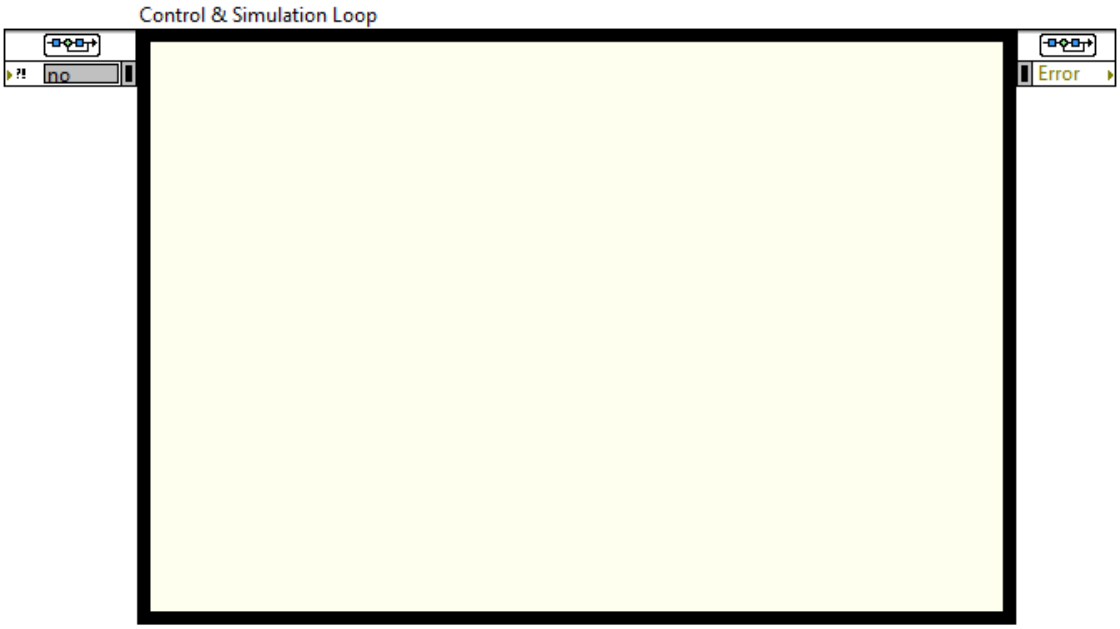

Figure 6.25: LabVIEW Control and Simulation Loop block

With the simulation loop block added, drag the External Models block into the simulation link, which will result in a file path window to open. Using the browse option in the window, select the .dll file produced in step 3 (section 6.3.3). This will produce a custom LabVIEW block with the inputs and outputs of the MapleSim model to appear with the names for each port being given the name entered in the original MapleSim file. Wire up the inputs to designated control blocks in the VI that will give the desired type of input data and the outputs of the model block to the intended output such as scopes, numerical indicators, motor drivers, etc and run the program.

6.4 MapleSim to LabVIEW Virtual Chain Inverse Kinematic Model Design

In this section the process of using a virtual chain, driven IKM produced in MapleSim and then converted to a LabVIEW external model component for control purposes is tested in a case study utilising the serial RRR mechanism from the previous chapter. Following the steps laid out in section 6.3, a new LabVIEW VI is built up and the MapleSim model added to the system. From here, key numerical inputs, such as the numerical control that represent the virtual chain joint positions and rotation, are added to the front panel of the VI to allow for the desired end position of the workspace of the RRR serial mechanism to be inputted. On the block diagram panel, these inputs are then wired into the simulation loop and then into the custom MapleSim external model block. Next the output ports form the custom MapleSim block are wired to the external wall of the simulation loop before being loaded into numerical indicators which show the desired joint angle of each of the robot's joints. These were then wired into subtraction blocks along with a local variable that reference the current joint angle of the RRR robot at the

start of the iteration. The subtraction block's output is then wired up in to a numerical indicator that shows the amount of rotation required for each joint from its current position.

The entire VI is then enclosed inside of a frame sequence block to ensure that this process takes place before anything else in the sequence. In the second frame, local variables of the desired joint angles of the system are fed into numerical indicators that represent the current values of each joint angle. This allows the system to simulate the movement of the mechanism to its new location and then allow the system's operator to move the mechanism again. Finally the entire VI is enclosed in a while loop block which allows the simulation to run until a stop button is pressed on the front panel, ensuring that the system runs until the user determines that the system is finished. The front panel view is shown in figure 6.26 and the block diagram panel is shown in figure 6.27.

Finally, the system is tested by inputting various end-effector positions into the system inputs with the values placed from the previous chapter's case study being displayed in figure 6.28. By investigating the full range of the mobility of the mechanism, it was discovered that when a desired position for the end-effector exceeded the range of motion of the mechanism, the system crashed.

This shows that after the program was run, the Labview system successfully recreated the motion of the MapleSim model as the desired values that were calculated through the exported model component generated the same values as gained in the traditional process and the MapleSim model.

This issue can be dealt with in multiple ways, the first being to modify the maximum and minimum limit values for inputted information into the numerical input components on the front panel thus preventing the mechanism from exceeding those values. This is best suited for parallel mechanisms as the actuated joints only have a set range of motion that will enable them to operate inside their workspace. The second way to deal with this is to include a type of redundant IKM which has the maximum and minimum values set up for the potential inputs, this can then be wired into an Or logic gate with the inputs looking into which simulation loop produces an error code. This then activates whichever simulation loop allows for a viable motion path with the user-controlled loop having priority in the event that both supply a viable motion.

Observing the results from the theoretical case study in section 6.3.1 against the values gained in the Labview control program, it can be seen that the methodology produced the same values for the joint angles. This shows that in the event of a physical robot arm being connected to the control system the inverse kinematic model produced in MapleSim would work as well as an inverse kinematic model directly coded into the software using the equations from the traditional process.

This section addresses the second point in the third objective of this thesis where inverse kinematic model of a mechanism is utilised in the development of a Labview control system through the MapleSim to Labview connector add-on.

Figure 6.26: Front panel for case study mechanism.

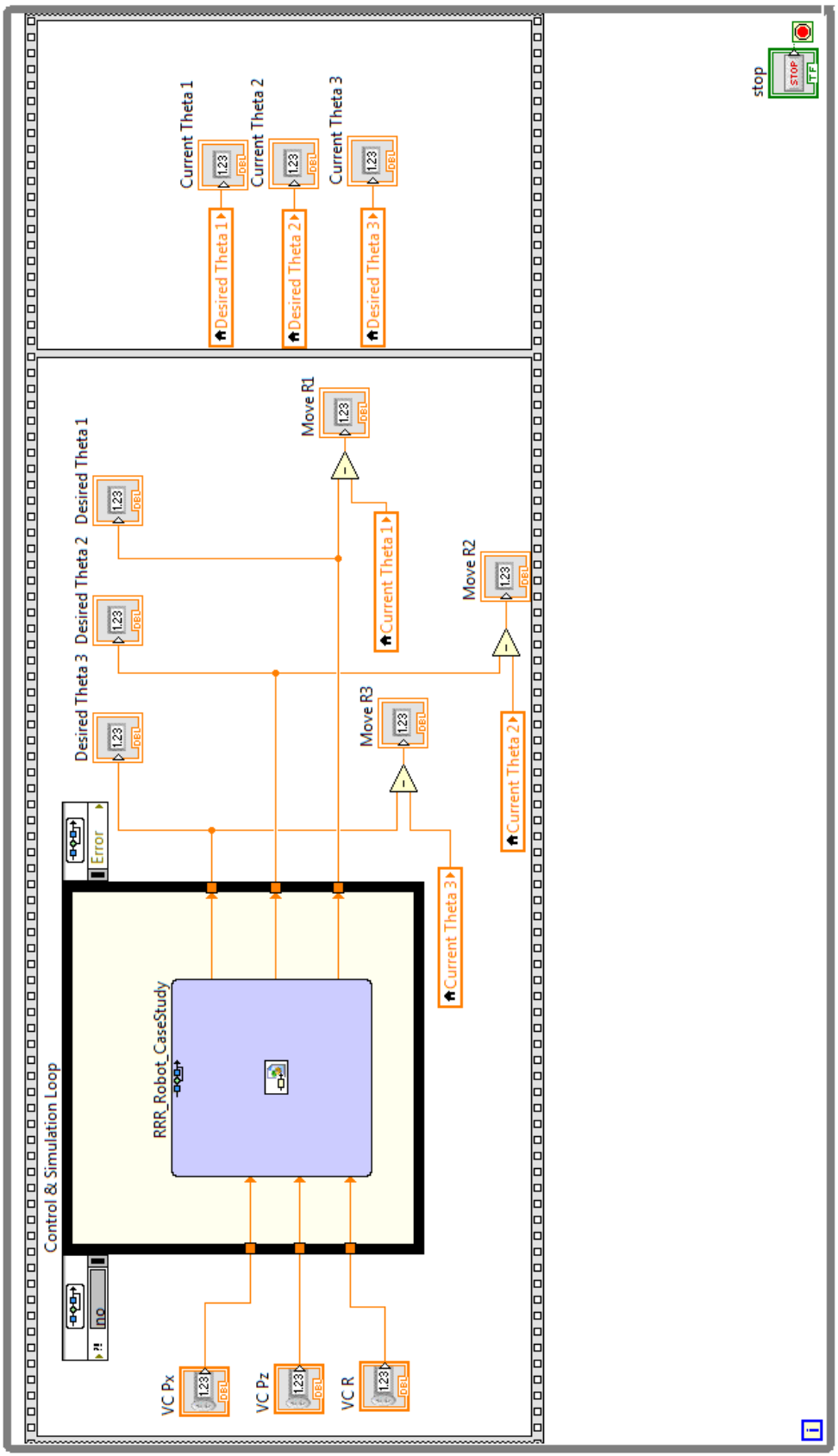

Figure 6.27: Block diagram panel of case study mechanism.

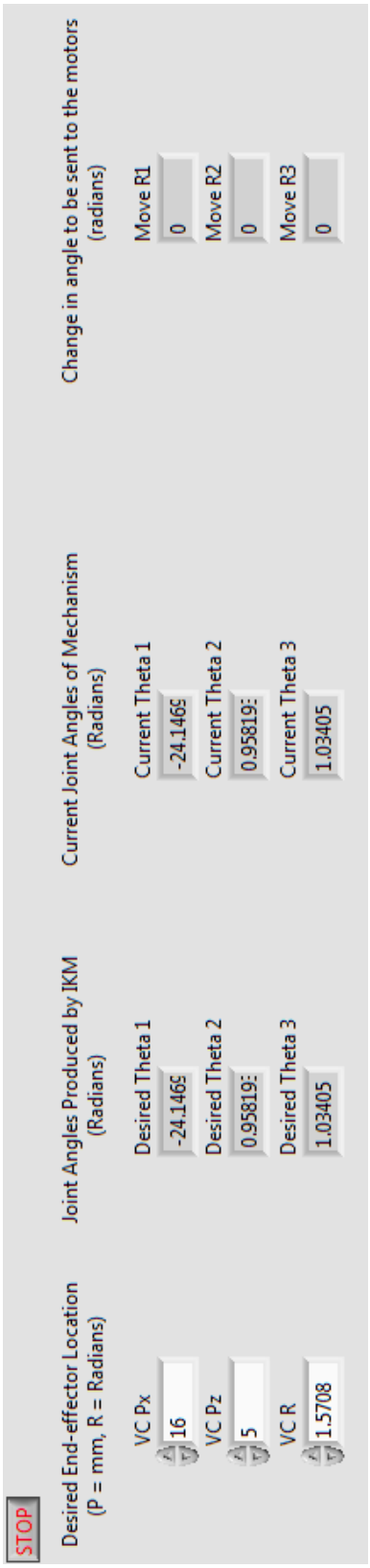

Figure 6.28: Front panel of case study mechanism after desired position is inputted.

6.5 Summary

In this chapter, the development of a control system utilising the inverse kinematic model developed in Chapter 5 has been achieved. The control system methodology detailed in

this chapter shows how to modify a MapleSim model to allow it to be exported and run in a Labview VI via the MapleSim to Labview Connector add-on to MapleSim. A series of case studies were investigated, detailing how the control system can be built up to allow a virtual or actual mechanism to be controlled in this manner while allowing for a comparison between the different methodologies. From the work detailed in this chapter, it can be seen that the proposed method of developing an inverse kinematic model in MapleSim can be modified in order to produce part of the control system of a physical or virtual mechanism as validated by comparing the traditional process to that of the proposed process.

This methodology will be implemented in the next chapter in the production of the control system for a physical multi-platform parallel mechanism prototype.

From the work detailed in Chapters 3-6, the modified design process stages can be summarised in flow diagrams similar to figure 1.2 in Chapter 1. These modifications include novel design processes that allow a designer to produce a new mechanism using basic knowledge of CAD design and MapleSim modelling throughout which a visual representation of the various stages is given. The flow charts for the proposed changes are shown in figure 6.29.

These proposed design process modifications address the fifth objective of the thesis by potentially formalising the modifications to several of the design process stages for the development of a new multi-platform mechanism in a more visual manner.

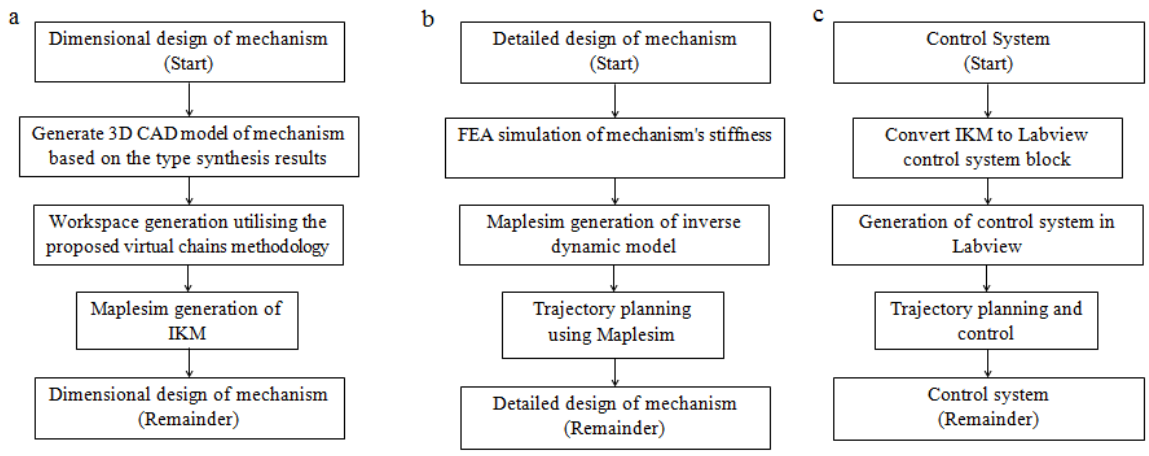

Figure 6.29: Modification of the design process' a) dimensional design stage; b) detailed design stage; c) control simulation stage.

References

[6.1] A. Vivas and P. Poignet, “Predictive functional control of a parallel robot” Control Engineering Practice, vol. 13, pp 863 – 874, 2005.

[6.2] F. Marquet, O. Company, S. Krut and F. Pierrot, “Enhancing Parallel Robots Accuracy with Redundant Sensors”, Robotics and Automation, vol. 4, pp 4114 – 4119, 2002.

[6.3] National Instruments, "LabVIEW", http://www.ni.com/labview/.

[6.4] National Instruments, "Numerical Indicators and Controls - LabVIEW 2014 Help - National Instruments", http://zone.ni.com/reference/en-XX/help/371361L-01/lvconcepts/numeric_controls_and_indicators/.

[6.5] National Instruments, "String and Path Indicators and Controls - LabVIEW 2014 Help - National Instruments", http://zone.ni.com/reference/en-XX/help/371361L-01/lvconcepts/string_and_path_controls_and_indicators/.

[6.6] National Instruments, "Boolean Indicators and Controls - LabVIEW 2014 Help - National Instruments", http://zone.ni.com/reference/en-XX/help/371361L-01/lvconcepts/boolean_controls_and_indicators/.

[6.7] X. Kong, "Inverse Kinematics of an RRR mechanism", Lecture note on Robotics, Heriot-Watt University, 2010.

[6.8] Y. Koren, “Robotics for Engineers”, McGraw-Hill Book Company, chapter 1, page 12, 1985.

[6.9] Maplesoft Website, "MapleSim Connector for LabVIEW and NI VeriStand Software - MapleSoft", http://www.maplesoft.com/products/toolboxes/labview_connector/.

Chapter 7 Prototype development

In this chapter, the novel processes and methodologies described in chapters 3-6 are integrated into novel changes to the design process and validated via the development of a new multi-platform parallel mechanism. The initial design step will produce the type synthesis of the intended mechanism, which will utilise the extended virtual chain approach to develop a 4-DOF parallel mechanism with two moving platforms connected by a P joint. The next step is the determination of the workspace of the mechanism using the virtual chain approach laid out in Chapter 4. This workspace calculation allows the entire workspace to be visually created, ensuring that the finished mechanism will be capable of performing the required task. Following this, the development of the inverse kinematic model and dynamic model of the mechanism is implemented in MapleSim to allow for the minimum specifications for motors to be determined. The last step in the design stage is the production of the mechanism's control system using the MapleSim to Labview connector. The final section of this chapter will be the production of a physical prototype to demonstrate that the new methodologies for the production of the inverse kinematic model and control system proposed in this thesis are valid.

7.1 Type Synthesis

Using the expanded virtual chain approach for type synthesis detailed in Chapter 3, the multi-platform parallel mechanism is designed.

The first step is to identify the wrench system of the mechanism. The proposed mechanism utilises a 3-ζ_∞-system on the primary platform and a 3-ζ_∞-system on the secondary platform with a single ε_∞ connecting twist between the two platforms. This means that the motion pattern of the mechanism is 3-tranlational for both platforms but with a relative translational DOF between the two platforms. Therefore, the potential wrench systems of a leg of the parallel mechanisms can either be: a 0-system; a 1-ζ_∞-system; a 2-ζ_∞-system; or a 3-ζ_∞-system.

For the prototype, the specification states that the mechanism should have a total of no more than three kinematic chains on the primary platform and no more than two kinematic chains on the secondary platform. The table for the viable kinematic chains for the proposed prototype mechanism is displayed in table 3.1 in chapter 3.

The virtual chain for this mechanism was determined to be a pair of PPP kinematic chains as both platforms are required to be able to translate along all three axes simultaneously. The connecting link between the platforms, producing the 1-rDOF of the mechanism can therefore be one of the following:

- A Sarrus mechanism, which is comparable to a combination of P joints in parallel.
- A pair of RRR kinematic chains.

As the design concept chosen at the end of chapter 3 was the 4-DOF system (3-DOF, 1rDOF), the layout of the kinematic chains can be determined in the same manner as two pairs of $PRRR$ kinematic chains. The axes of rotation for the R joints and axis of translation of the P joint all act along a single axis for each kinematic chain, where two chains act along the same axis but are connected to different platforms producing the

single rDOF. Following the rules of actuator selection the *P* joints of each of the kinematic chains have been selected as the actuated joints in order to improve the mechanism's operational quality.

7.2 CAD Concept Designs and Analysis

As the chains of the mechanism have been selected to be two pairs of $PRRR$ kinematic chains attached to each platform and the platforms will be joined by a 1-DOF kinematic chain or joint, the mechanism can be developed into a 3D CAD model for analysis. As discussed in Chapter 1 concerning the H4, I4 and Par4 mechanisms, it was found that an even rotation of the kinematic chains greatly improves the stability of the mechanism being designed. With this in mind, each kinematic chain is orientated to be perpendicular from its partner with the two kinematic chains that share an axis being distributed between the two platforms. The various 1-DOF kinematic chains or joints can be arranged in several forms as shown in figures 7.1 and 7.2 for the Sarrus mechanism link and *P* joint link respectively.

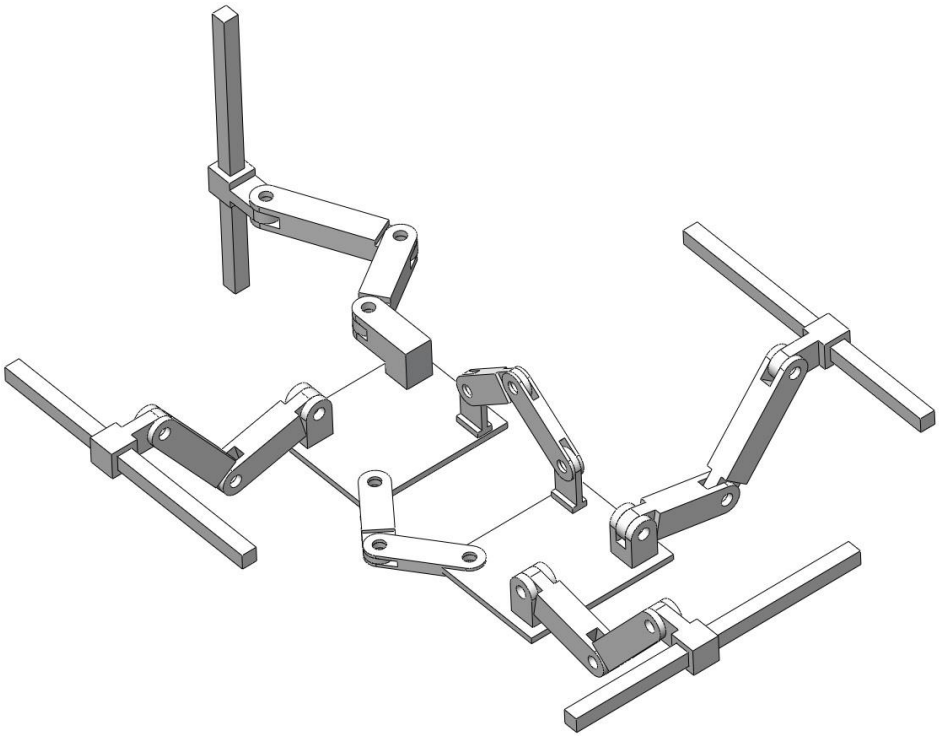

Figure 7.1: Kinematic chain layout with a Sarrus mechanism.

Connecting the two platforms with the either connecting link produces a mechanism capable of the desired DOF and that can be analysed for static displacement and workspace capabilities.

7.2.1 Non-actuated prototype development

In order to determine the practicality of the chosen mechanism, it was decided to produce a non-actuated scaled down prototype. To reduce the cost of production and to save on manufacturing time and complexity, the model mechanism has been design to be produced using a 3D printer to produce the complex components in PLA and a laser cutter was used to produce the links and moving platforms from Perspex. The design in section 7.2 required modification to suit the chosen materials. The *P* joint casing was reworked

to include two cylindrical aluminium rods as runners which would pass through a 3D printed block in place of the original square hollow design.

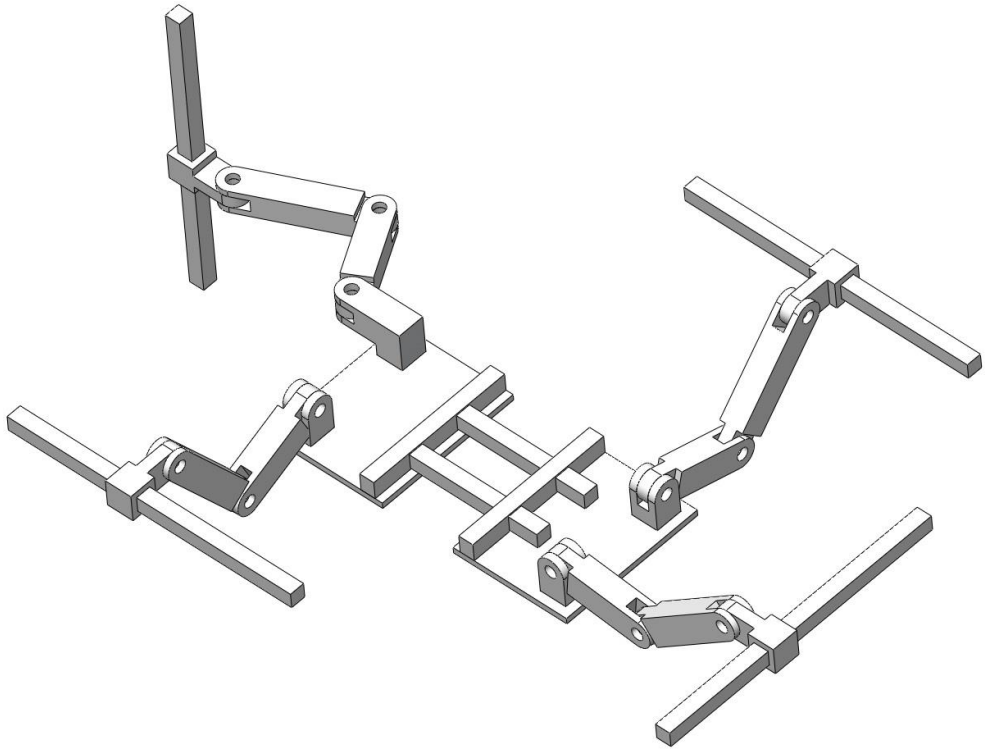

Figure 7.2: Moving platform concept with 1-DOF kinematic chains.

In order to laser cut the upper and lower arm links the design was altered to speed up production time. In order to increase the rigidity of the design the first RR kinematic link was increased from a single running strut to three, while the second RR kinematic link was increased from a single strut to being a pair, which allows the first link to hold the second firmly.

The moving platform remains mostly unchanged in shape but in order to reduce the complexity of the mechanism as a whole the kinematic chain connectors have been redesigned to allow the new kinematic links to be slotted around the connector. Likewise the moving platform RRR connecting links were redesigned to improve the viability of the 3D printing as the 3D printer has difficulty printing at angles greater than 45°.

The mechanism is then constructed in the same manner as its counterpart in section 7.2 with the only difference being the two shared axis P joints running along the same aluminium rods to save on costs.

Constructing the non-actuated prototype scaled model requires a cage to be built to secure the aluminium rods in place. To this end, a cage was designed out of equal angle aluminium extrusions. Holes were added along each length to allow for the adjustment of the positions of the aluminium rods as well as any modification required.

Finally, the mechanism was placed into the cage and secured by the 8mm aluminium rods. The aluminium rods were altered to be threaded rods to allow nuts to be added on the external faces of the cage to prevent unwanted movement (figure 7.3).

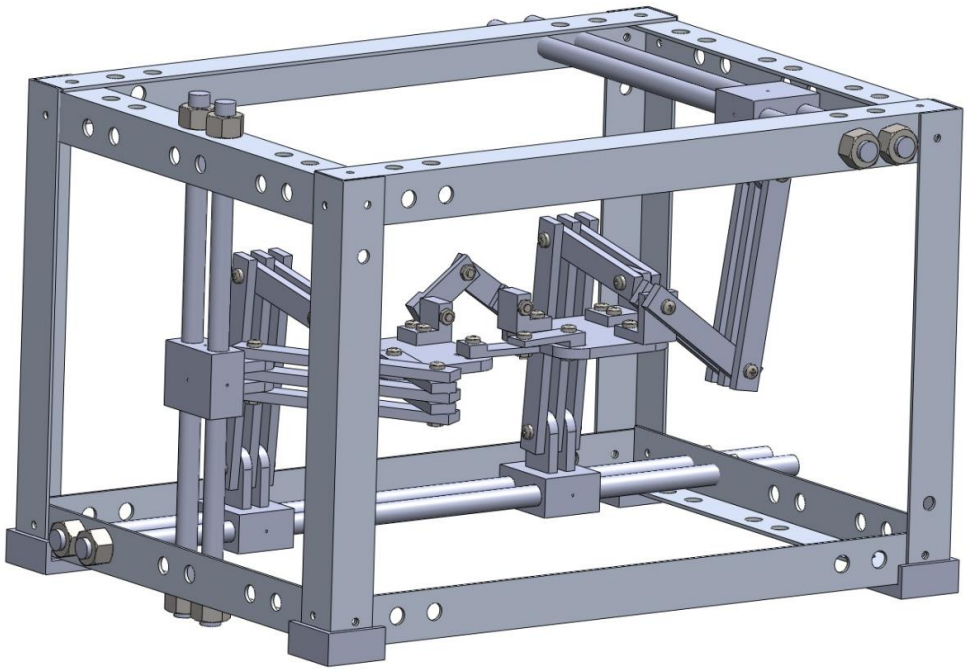

Figure 7.3: Final non-actuated prototype design with cage.

The non-actuated prototype was built to specification and is shown in figure 7.4.

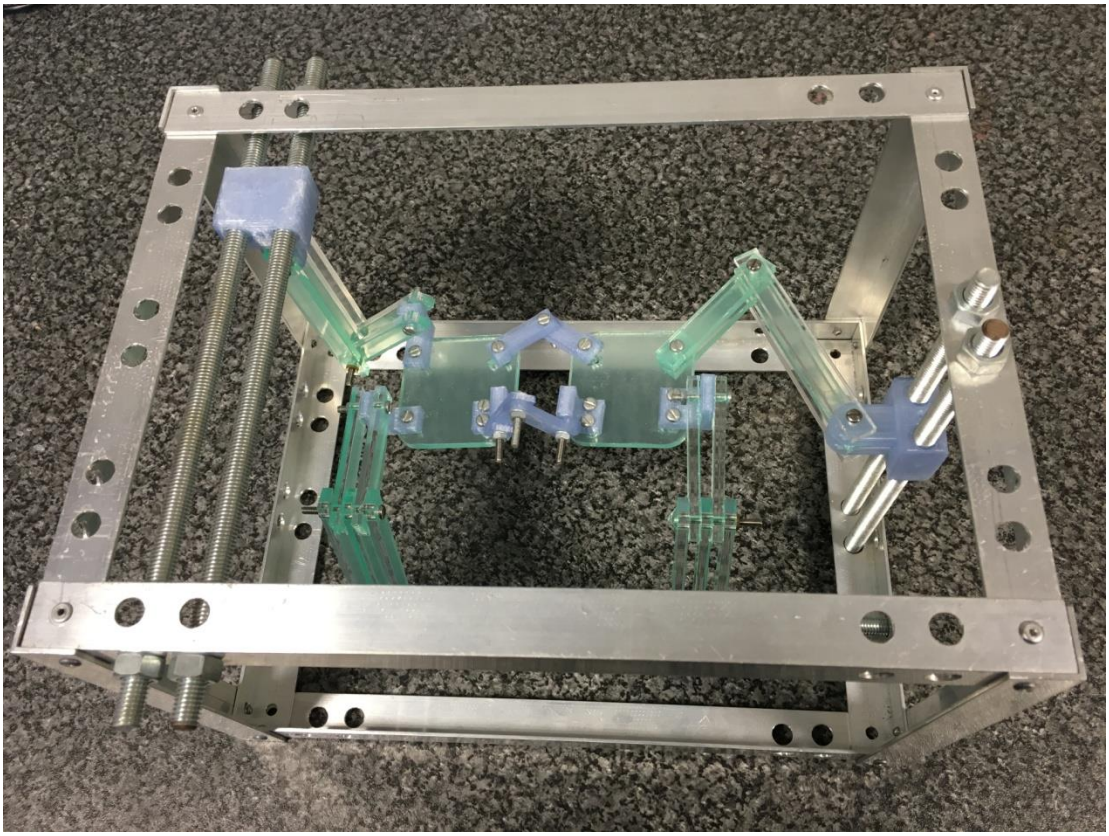

Figure 7.4: Completed physical non-actuated prototype model

Once the non-actuated prototype had been constructed, there were several design flaws apparent regarding to the integrity of the RRR links of the moving platform. It was found that the links were not rigid enough in their current state to maintain the horizontal plane

of the two moving platforms. The single vertical kinematic chain of the mechanism was found to be incapable of supporting the entire weight of the mechanism. This issue can be solved with one of the following solutions:

- Adding a pulley-counter weight system that will be suspended above the mechanisms centre.
- Adding another vertical kinematic chain that would be either actuated through a gear chain, attached to the primary vertical kinematic chain's servomotor coupling or left un-actuated. This would allow the system to distribute the load between both vertical kinematic chains equally.
- Adding springs which would be attached across the two RR links in each arm, which would restrict the overall motion of the joint allowing for the platform to retain its rigidity
- Replacing the moving platform RRR link with the paired P joint link design along with the modification of the moving platform to allow for a more rigid design.

Despite the design flaws in the system the non-actuated prototype displayed that, with modification, the mechanism would be capable of the desired motion pattern when manipulated by hand. While the development of a scaled-down model of the proposed mechanism is not a necessary part of the design process, it can be advantageous as it can be used to aid in the determination of physical design flaws and limitations early in the development process. It will therefore be added to the proposed changes to the design process as an optional block.

7.2.2 Dimensional design of the actuated prototype

Following the results from the non-actuated prototype, the design of the mechanism was altered to increase the structural integrity of the kinematic chains and the moving platforms. The mechanism has been designed to be constructed from flat bar aluminium lengths in order to reduce potential bending in the links and moving platform.

The P joints of the non-actuated prototype worked well in testing resulting in the design for the actuated prototype being based upon this. The P joints for the actuated prototype are driven by motors attached via a coupling to a power transmission lead. In order to prevent the P joint from rotating, a pair of 5mm aluminium guiding rods has been added to the mechanism in parallel with the lead screw. As the components for the kinematic chain are designed from aluminium flat bar sections, the PR link, RR link, and R to moving platform link in each chain had to be altered to allow the new material to work. The chosen solution involves a slot design in which the R joint of the previous link slots around the coupled R joint of the current link allowing a pin to be placed along the axis to hold the two components together. All of the concept designs in this section utilise the same arm sizing to maintain a comparable result. The lengths of the PR, primary RR, and secondary RR links are 100mm, 200mm, and 180mm respectively with each having a width of 44.5mm and a depth of 25.4mm. The R joint pins are designed to be 6mm in diameter and 44.5mm in length to allow for a flush fit with the rest of the arm. Finally the P joint socket is designed to be 18mm in diameter to allow for the corresponding nut for a 10x3mm power transmission lead screw to be attached to the block during the final build of the mechanism.

The final section of each kinematic chain comprises the moving platform connectors that are designed as inverted T shaped blocks for the horizontal (x and y) kinematic chains and a dual slotted connecting block. The dual slotted connecting block fits into the end of the secondary RR link of the vertical (z) kinematic chain that is then connected to the moving platform by slotting the platform into the final slot of the connector. Both types of connecting blocks are pinned in place by a pair of bolts, removing any unwanted additional rotations that the motion of kinematic chains could impose on the moving platform.

As stated, the connecting links designed to connect the two moving platforms for the prototype will either be a pair of RRR links set perpendicular to each other in order to reduce the available rDOF between the two platforms to one or a combination of parallel P joints to improve rigidity of the connection. The dual RRR link is designed to use the inverted T-shaped R to moving platform connectors used to attach the kinematic chains of the mechanism to the moving platforms for the raised RRR link. The horizontal RRR links are instead designed to have the moving platforms slotted inside of the first and last R joint with the remaining R joints of the two links slotting into each other. The parallel P joint link has two separate designs, the first is similar to the one proposed at the start of this section where, in this case, three cylindrical P joints are aligned with the direction of the shared axis to allow for an increased stability and a reduction in the friction caused when P joints are used. The second design has one moving platform in the shape of a U and the other in the shape of a T with the two gripper's positions placed at the cross section of each platform. The second platform is then located inside of the slot created by the U-shaped hole and a pair of P joints passed across the gap, starting at one side of the first platform and ending at the other with the second platform connected via a raised block.

The last components sized are the bearings that are used in all R joints of the kinematic chains. As each kinematic chain is designed to apply force only along the axis of motion along which its P joint translates, the axial load for each bearing is reduced to near 0N. Therefore the bearings to be selected need only be reliable and of a practical size so that the 6mm pin used to hold together each R joint can be placed through the bearing's centre. With this in mind, NSK Deep Grove Ball Bearings were selected. The bearings were modelled into SolidWorks and appropriate slots were incorporated into the kinematic chain's links. The final mechanism designs are shown in figures 7.5, 7.6 and 7.5.

7.2.3 Static analysis of designs

Using the three designs from the previous section, a static finite element analysis (FEA) was performed on each fully assembled model. The faces of the PR link block closest to the pair of guiding rods are set to be the grounded locations of the FEA model and all components are given their particular material types. All 6mm pins have been replaced by Pin components within the FEA system to make the system more robust and, in the case of concept designs 2 and 3, the P joints setup as appropriate in order to allow any deflection to move the joint rather than deflect the whole system. Gravity is then applied to the system and the mechanisms are placed into several configurations in order to determine a range of stresses and displacements of each mechanism. The results for static stresses and displacement are displayed in figures 7.8, 7.9 and 7.10 for concept Design 1, figures 7.11, 7.12 and 7.13 for concept Design 2, and figures 7.14, 7.15 and 7.16 for concept Design 3.

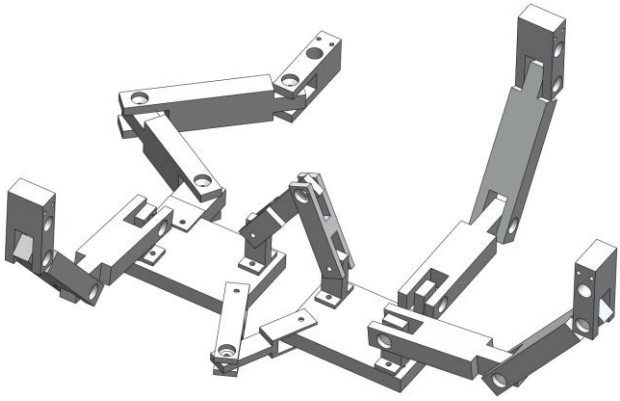

Figure 7.5: Fully assembled prototype concept Design 1.

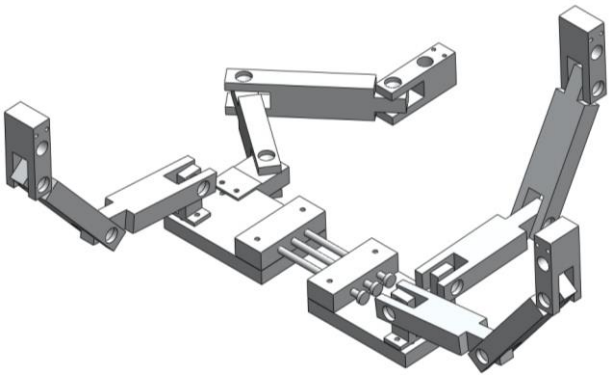

Figure 7.6: Fully assembled prototype concept Design 2.

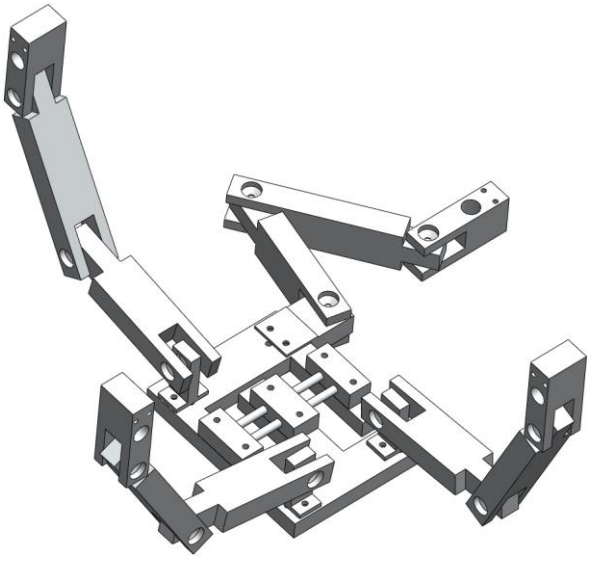

Figure 7.7: Fully assembled prototype concept Design 3.

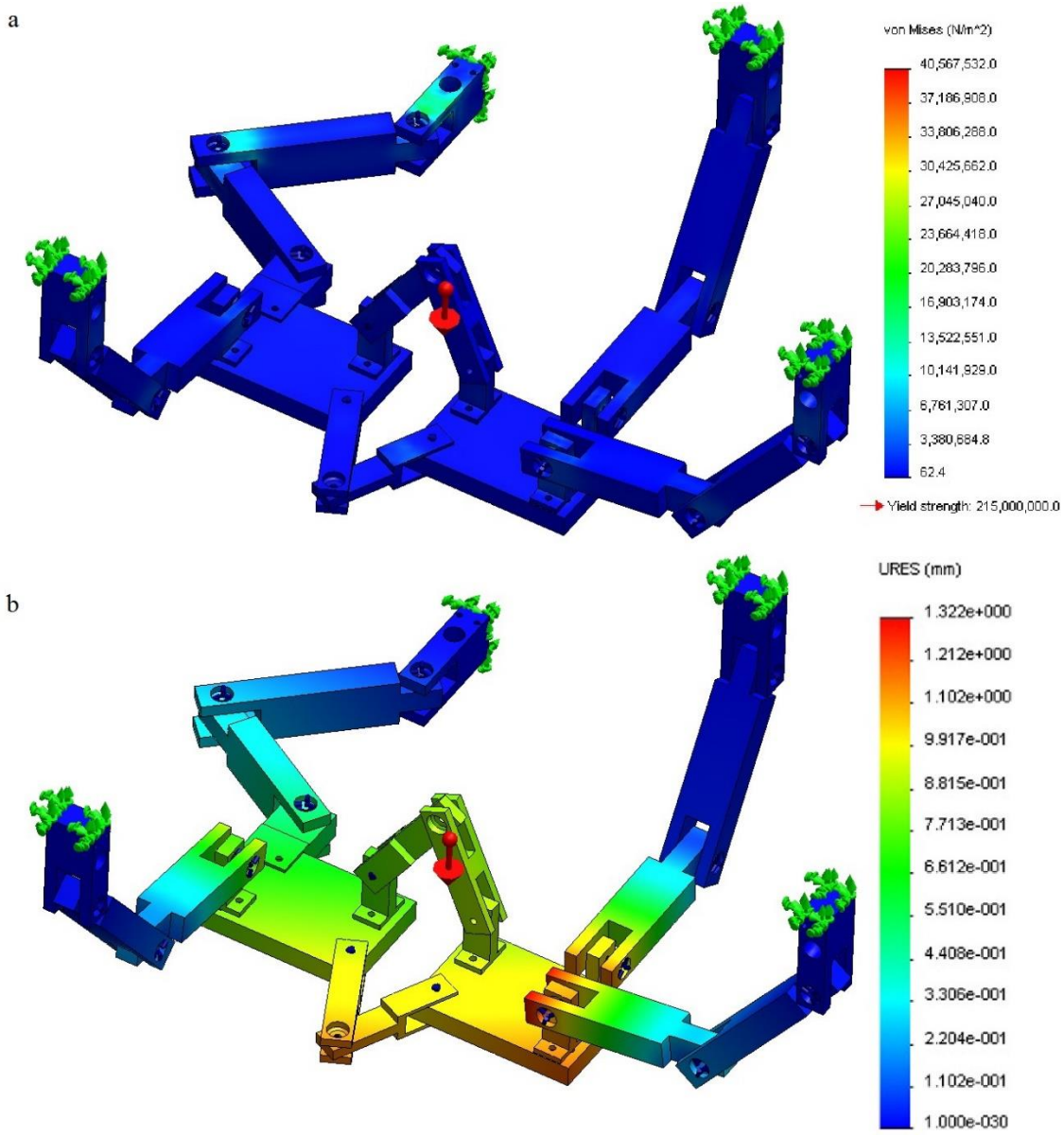

Figure 7.8: Prototype concept Design 1 position 1 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

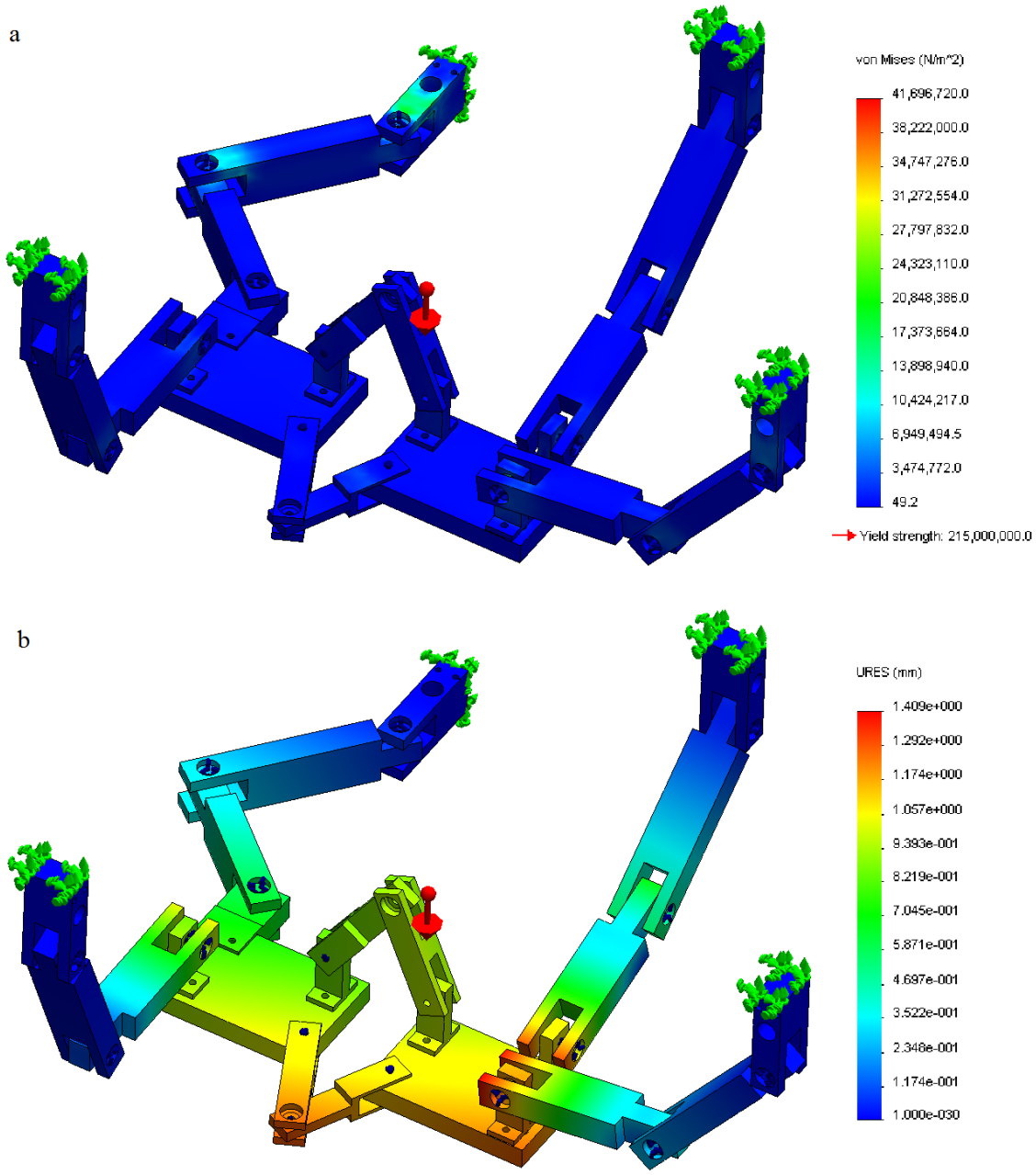

Figure 7.9: Prototype concept Design 1 position 2 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

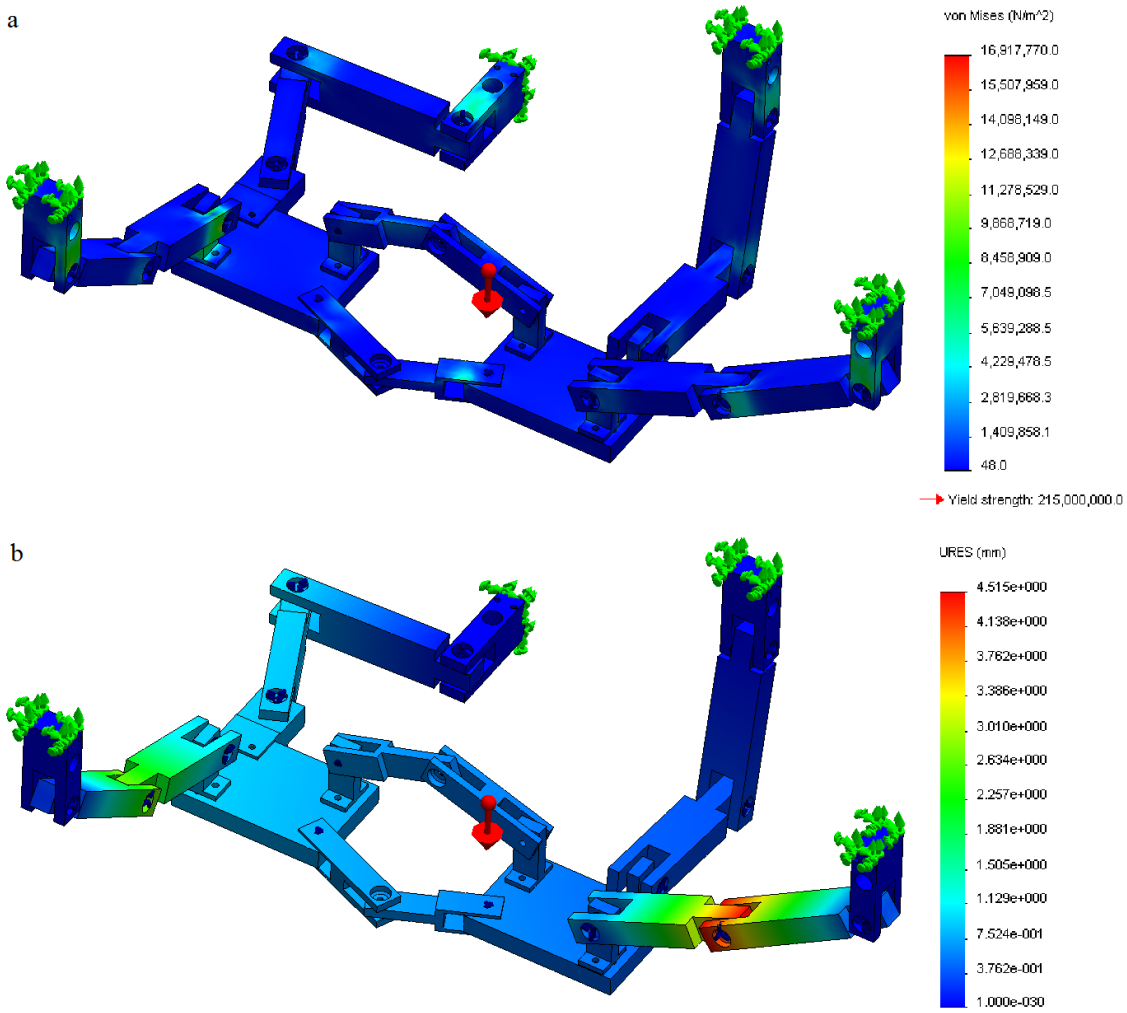

Figure 7.10: Prototype concept Design 1 position 3 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

Comparing the results of the finite element analysis of concept designs 1 and 2, it can be seen that the inclusion of the P joints caused the mechanism to undergo lower levels of displacement with marginally higher levels of stress in the horizontal axis. Additionally the modification of the moving platform in concept Design 3 when compared to that of concept design, showed that the third design was more statically robust as lower stress levels were detected in the mechanism and the amount of deformation in the system was half that of the other mechanisms at its highest point. Due to the results of the non-actuated prototype, the FEA step will be added to the design process to improve upon the success rate of the method as shown in figure 7.17.

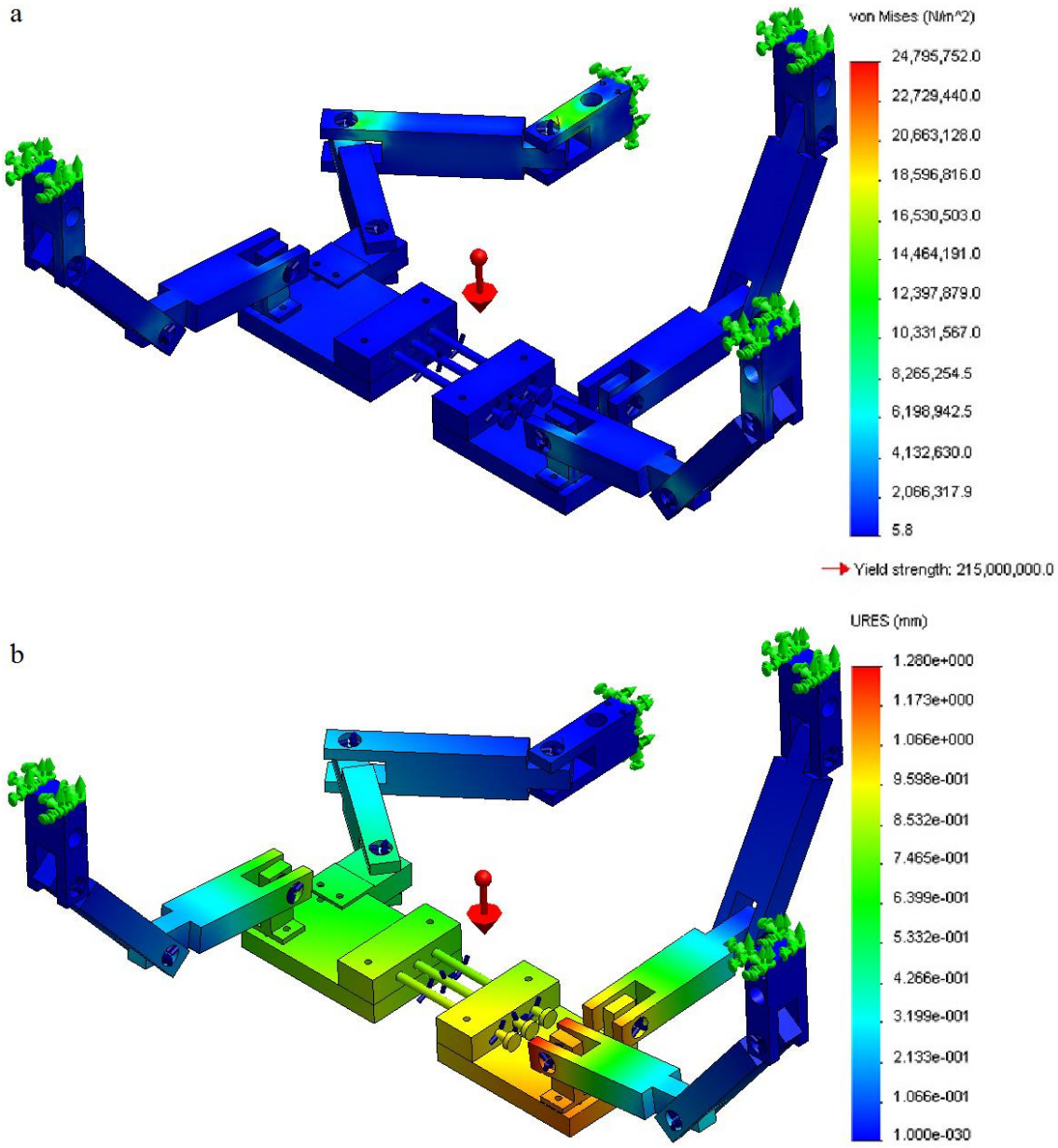

Figure 7.11: Prototype concept Design 2 position 1 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

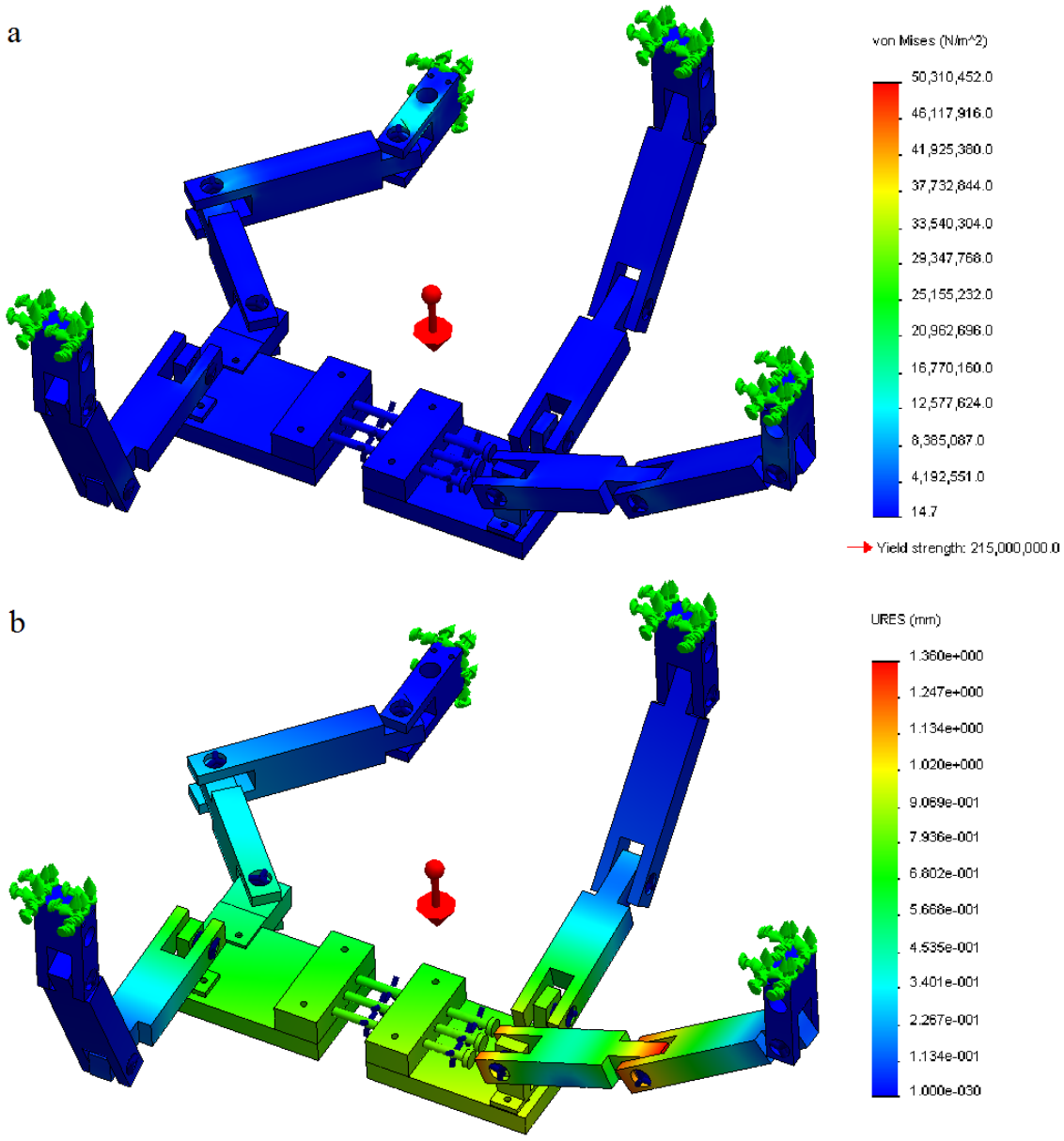

Figure 7.12: Prototype concept Design 2 position 2 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

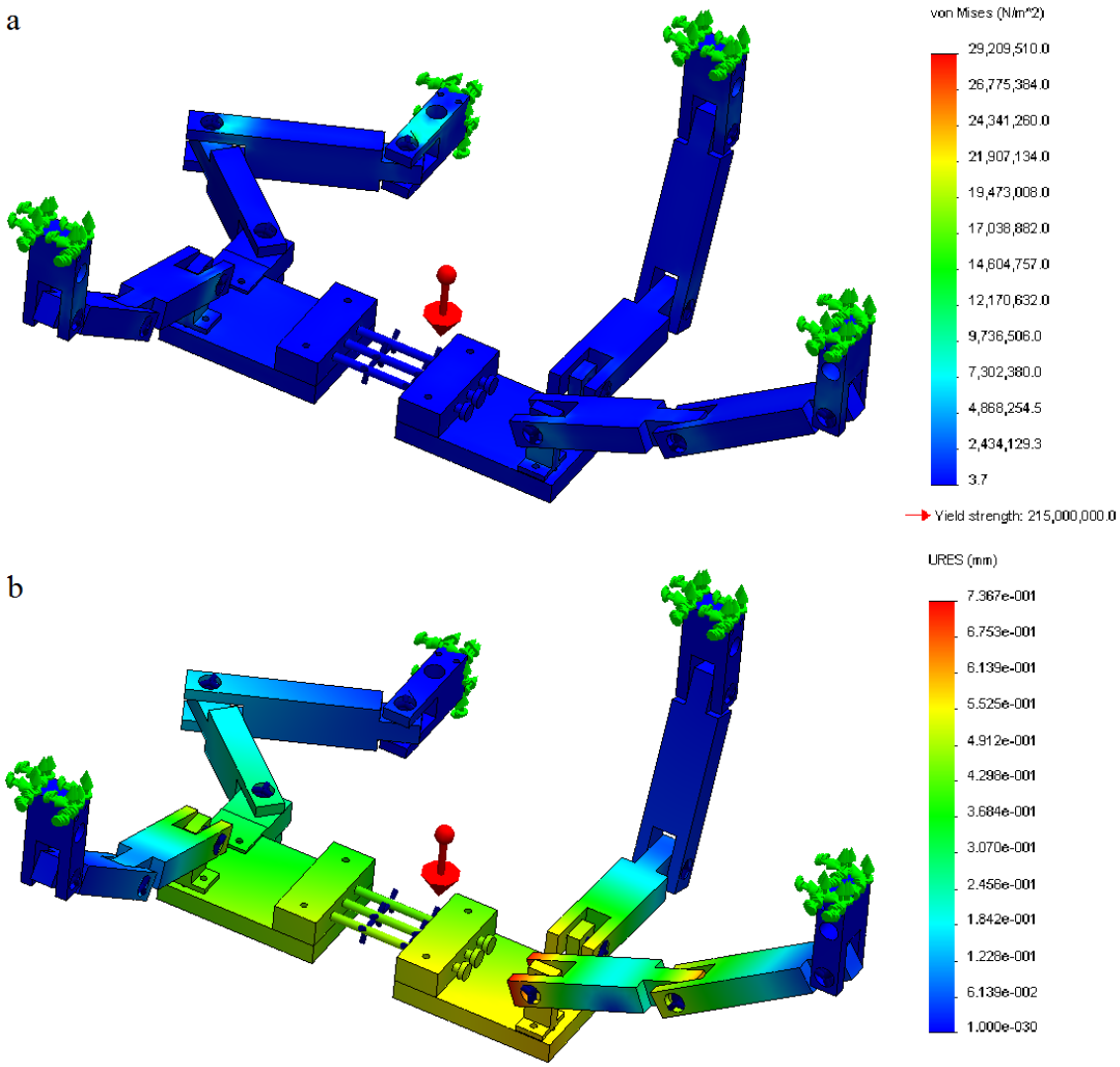

Figure 7.13: Prototype concept Design 2 position 3 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

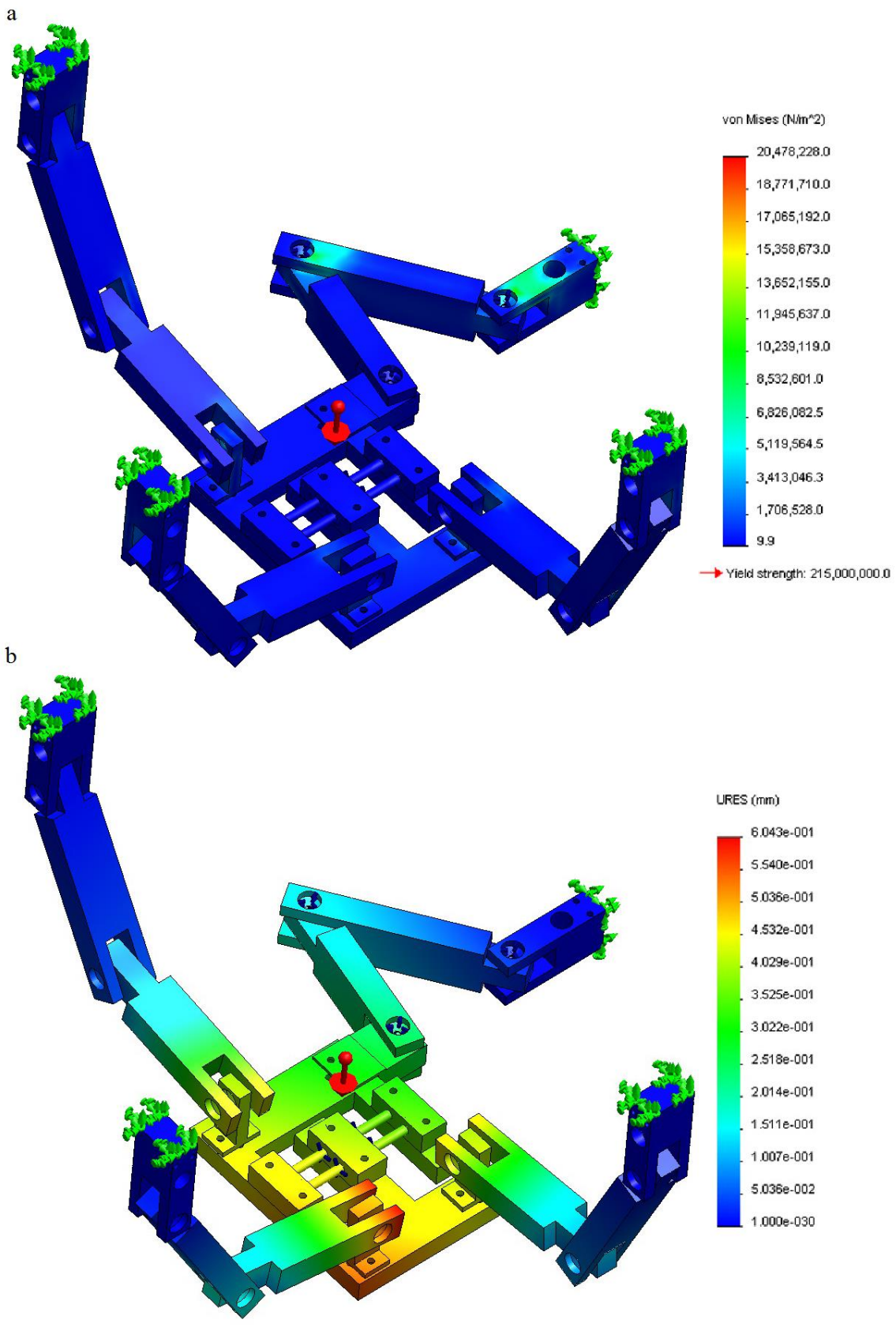

Figure 7.14: Prototype concept Design 3 position 1 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

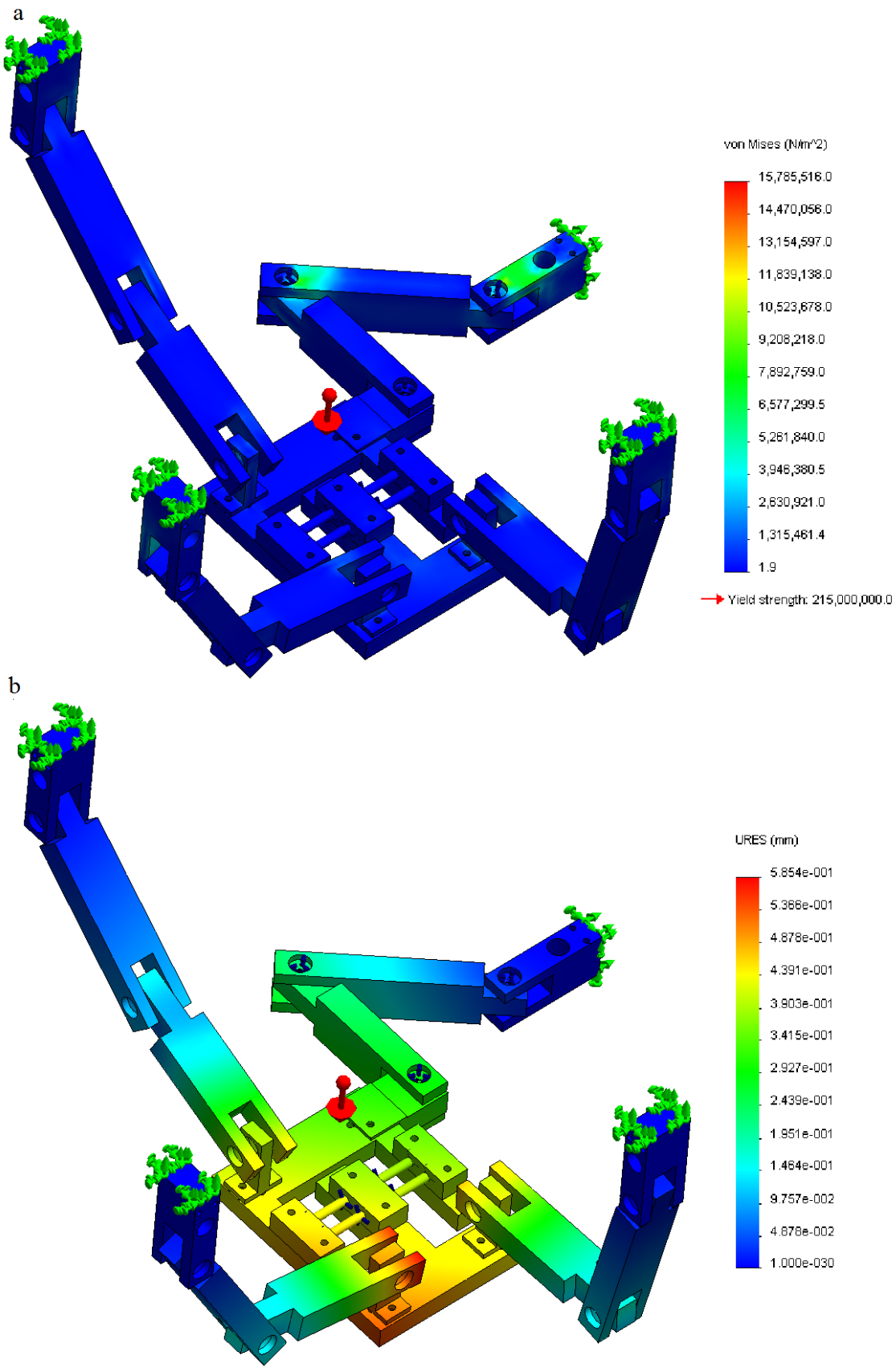

Figure 7.15: Prototype concept Design 3 position 2 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

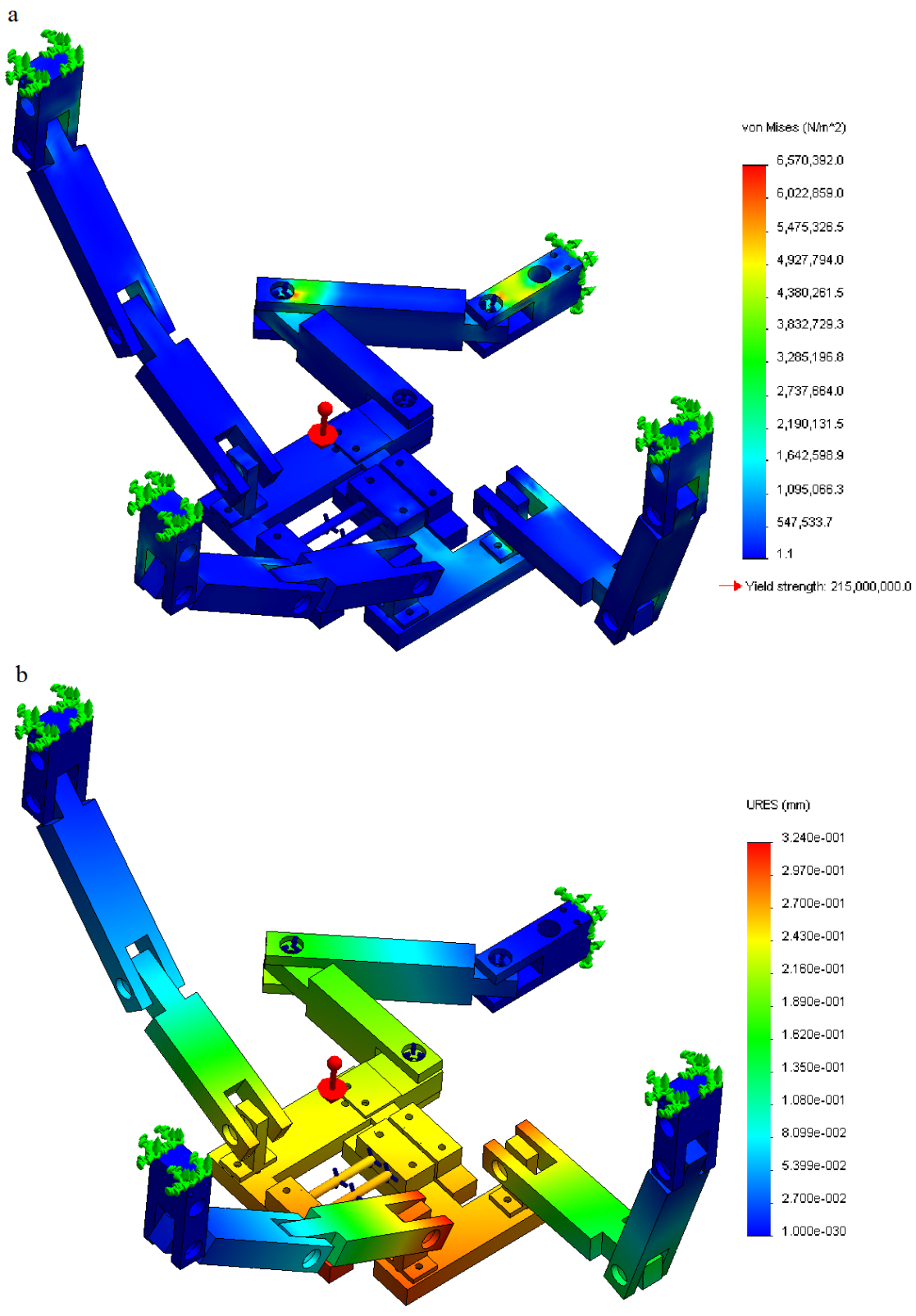

Figure 7.16: Prototype concept Design 3 position 3 FEA of: a) static stresses in the system under load; b) static displacements in the system under load.

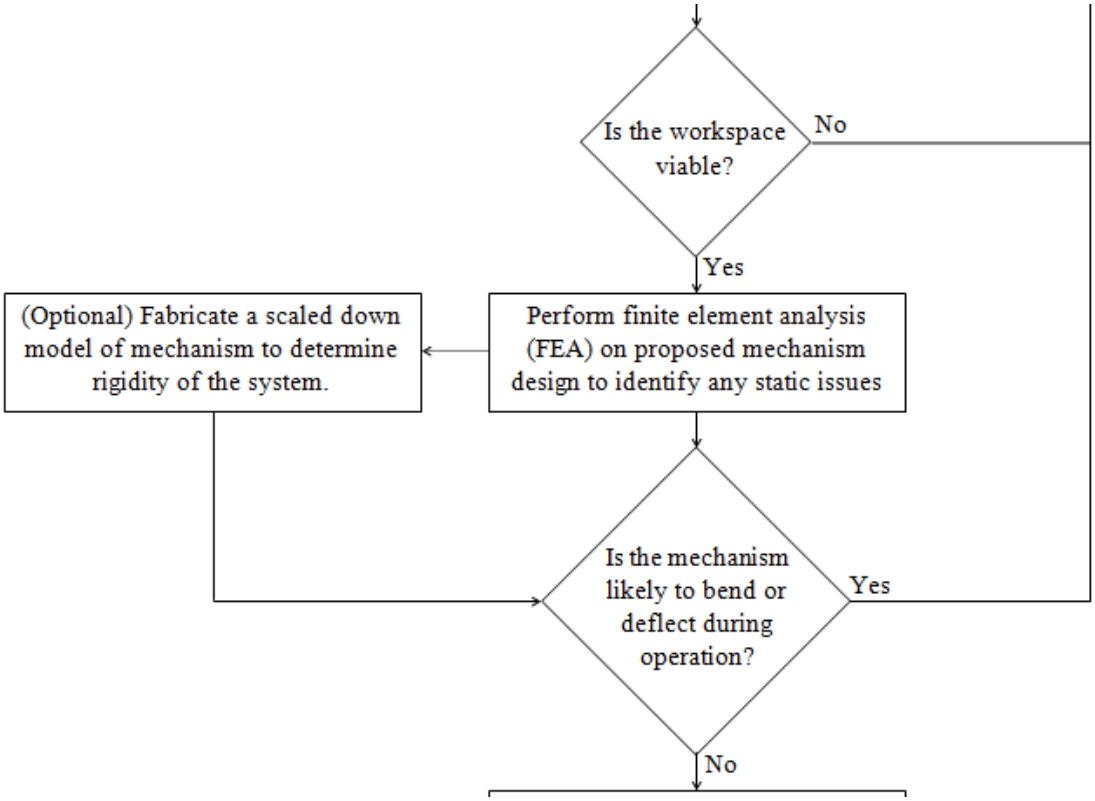

Figure 7.17: FEA step inputted into place in the proposed modification to dimensional design stage.

7.2.4 Workspace analysis of designs

Using the process of developing a visual representation of a mechanism's workspace as detailed in Chapter 4, the workspaces of each mechanism is produced. As each of the mechanism designs allow only for a single rDOF, the two virtual chains can be combined to simplify the model. This is done by combining the two axes of motion incapable of differing in position from each other; in this case, the vertical P joints and the horizontal P joint aligned perpendicular to the rDOF axis. The remaining P joints that share the rDOF are placed on the same P joint yoke to allow the simulation to be as straight forward as possible (figure 7.18).

In order to produce the workspaces of the three concept designs a base for each mechanism needs to be implemented. This is done through the addition of a basic cage in which, four bars are suspended between vertical, and in the case of the z-axis, horizontal struts. The layout of these bars, which represent the lead screws in the system, requires a slightly different layout for concept 3 when compared to the others due to the way in which the kinematic chains are laid out. The cage design for concept designs 1 and 2 is displayed in figure 7.19 and for concept 3 in figure 7.20. In these figures, the bars are coloured blue to help distinguish them from the cage's framework.

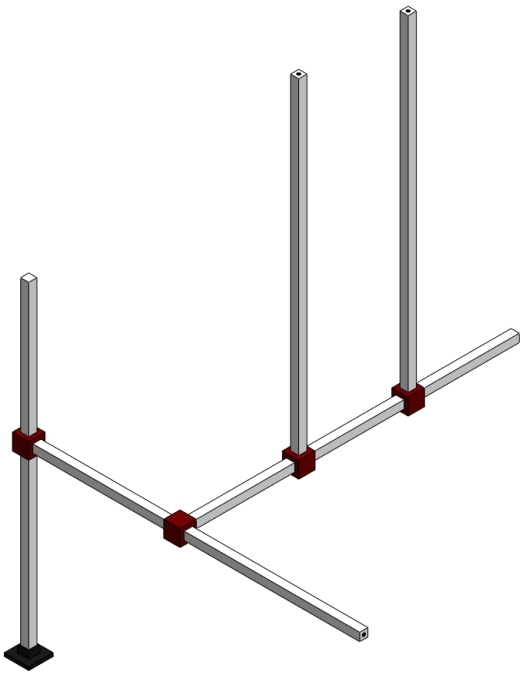

Figure 7.18: Simplified virtual chain design for concept designs.

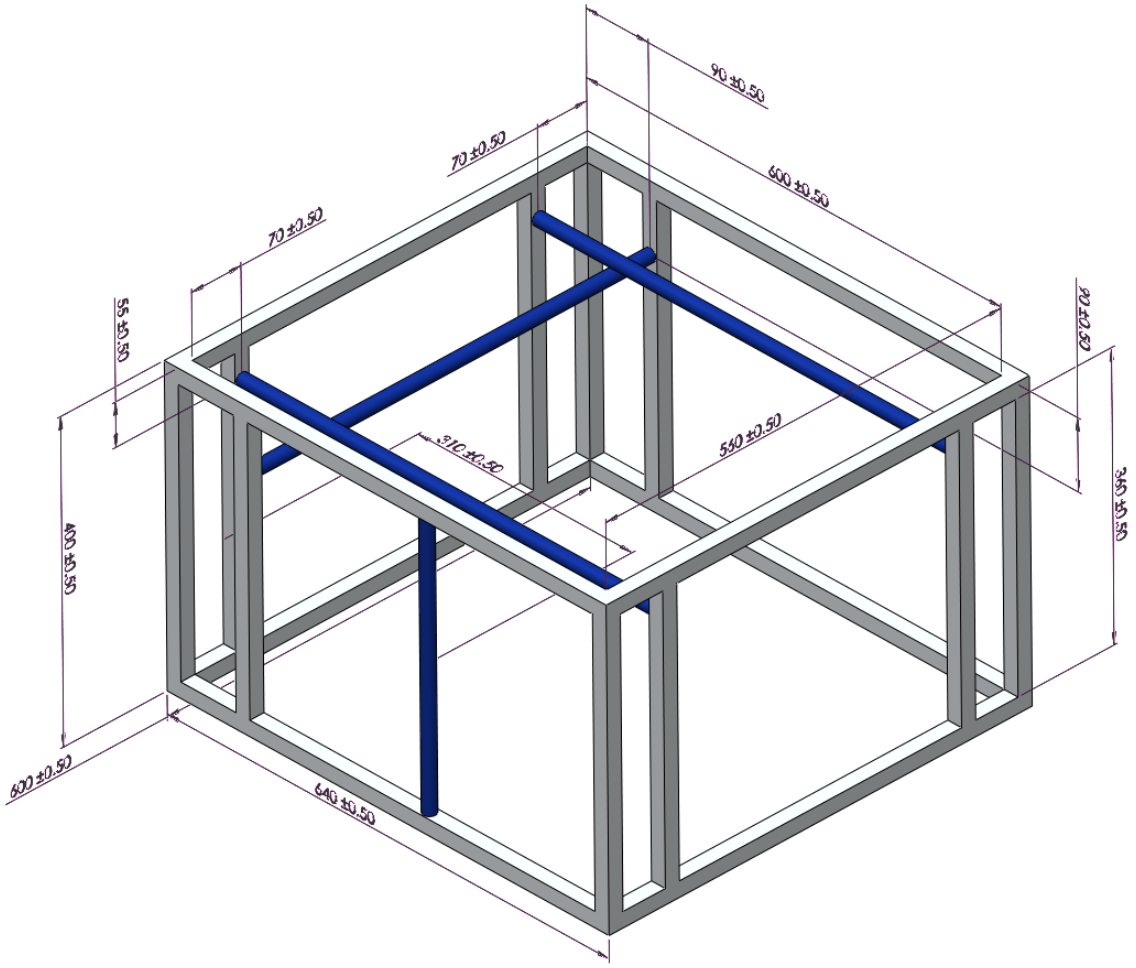

Figure 7.19: Cage design for mechanism concept designs 1 and 2.

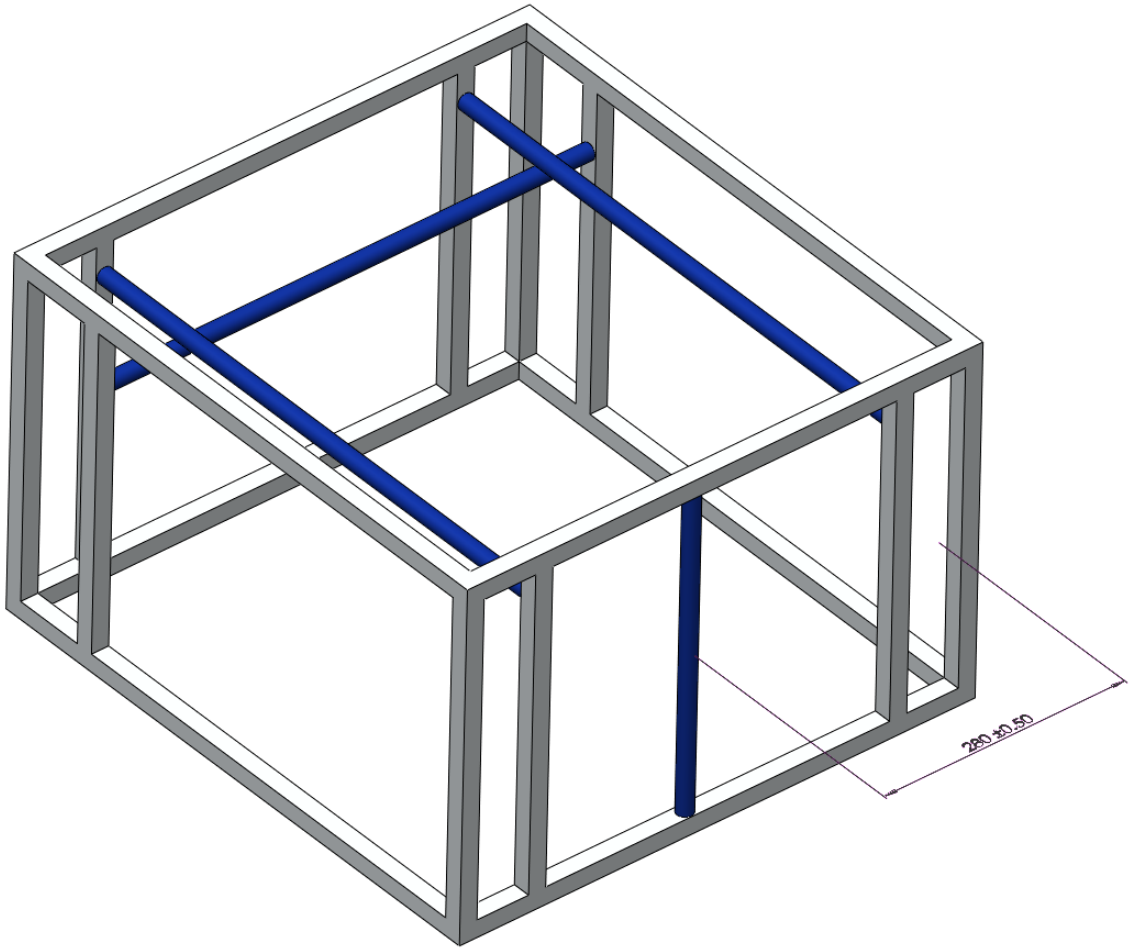

Figure 7.20: Cage design for mechanism concept Design 3.

Applying these cages to the mechanism allows the force method of generating workspaces to be applied. For the simulation of a multi-platform mechanism's workspace, the force method needs to be applied multiple times in order to produce the active workspace of each individual platform. The results for the primary and secondary moving platform of concept Design 1 are shown in figures 7.21 and 7.22 respectively. Additionally the results for the primary and secondary moving platforms of concept Design 2 are displayed likewise in figures 7.23 and 7.24 respectively.

The final set of force method workspace results are displayed below in figures 7.25 and 7.26; these represent the results of the force method generated workspaces for the primary and secondary platforms of concept Design 3 respectively.

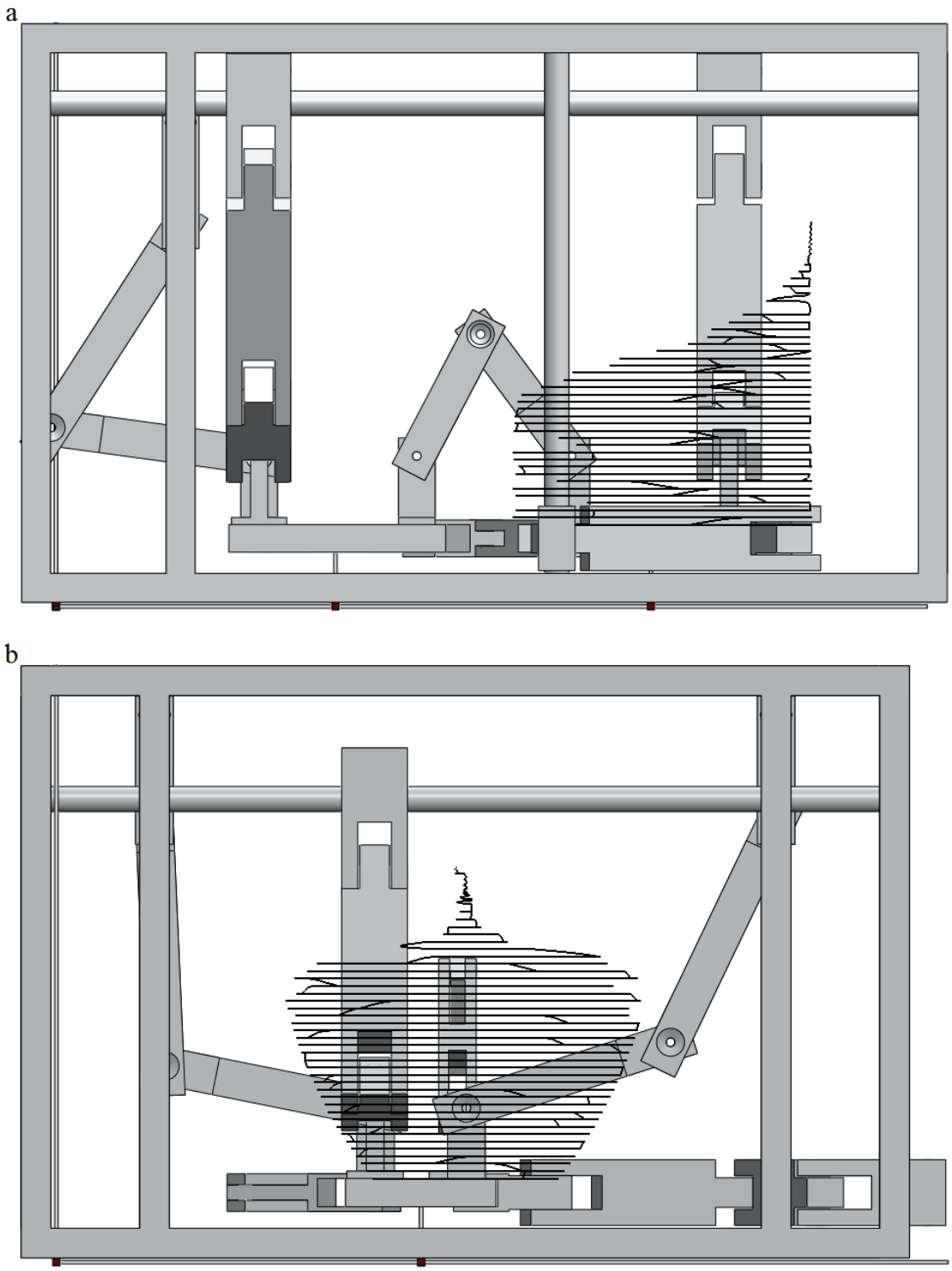

Figure 7.21: Workspaces for the primary platform of concept Design 1 on the a) x-z plane, and b) y-z plane.

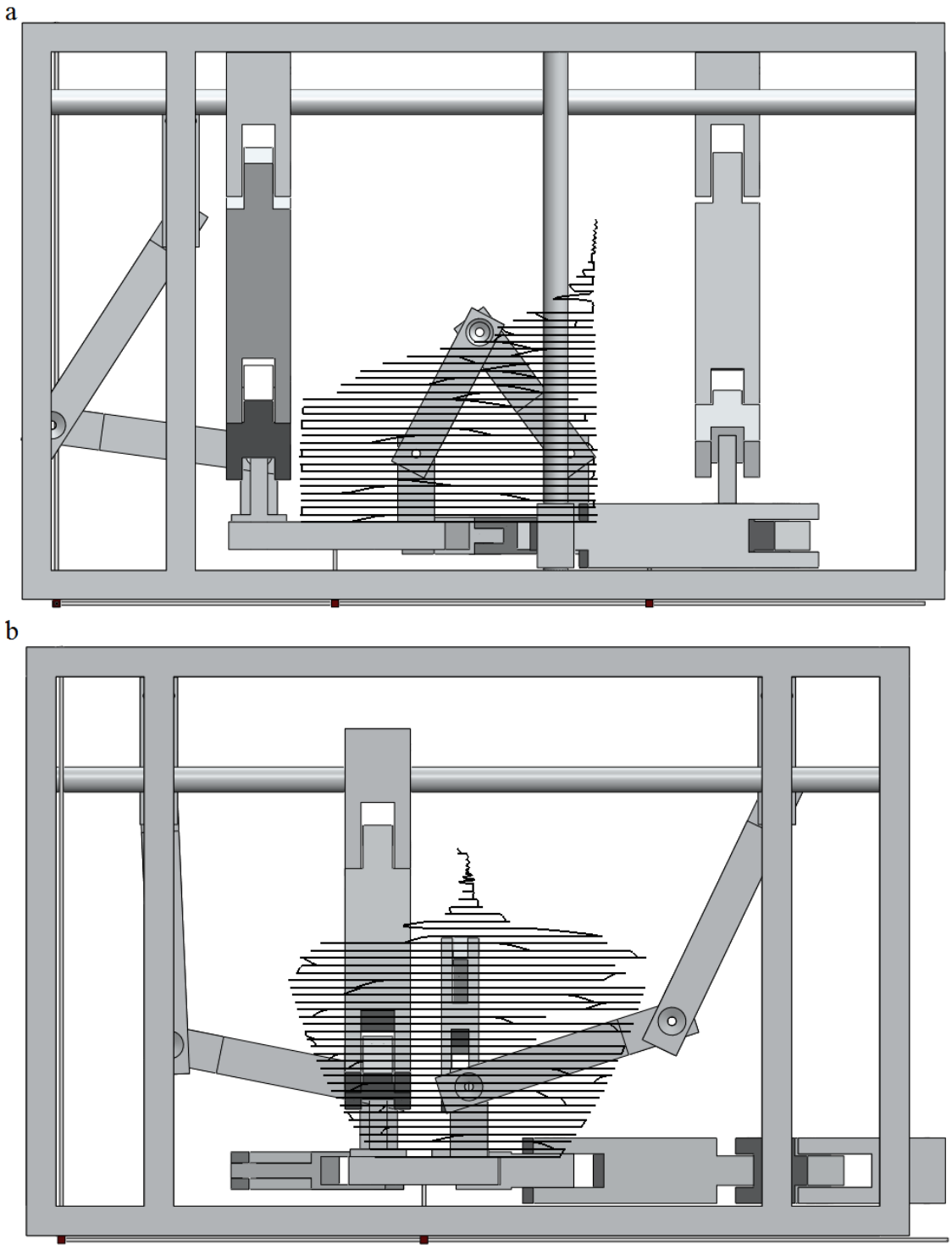

Figure 7.22: Workspaces for the secondary platform of concept Design 1 on the a) x-z plane, and b) y-z plane.

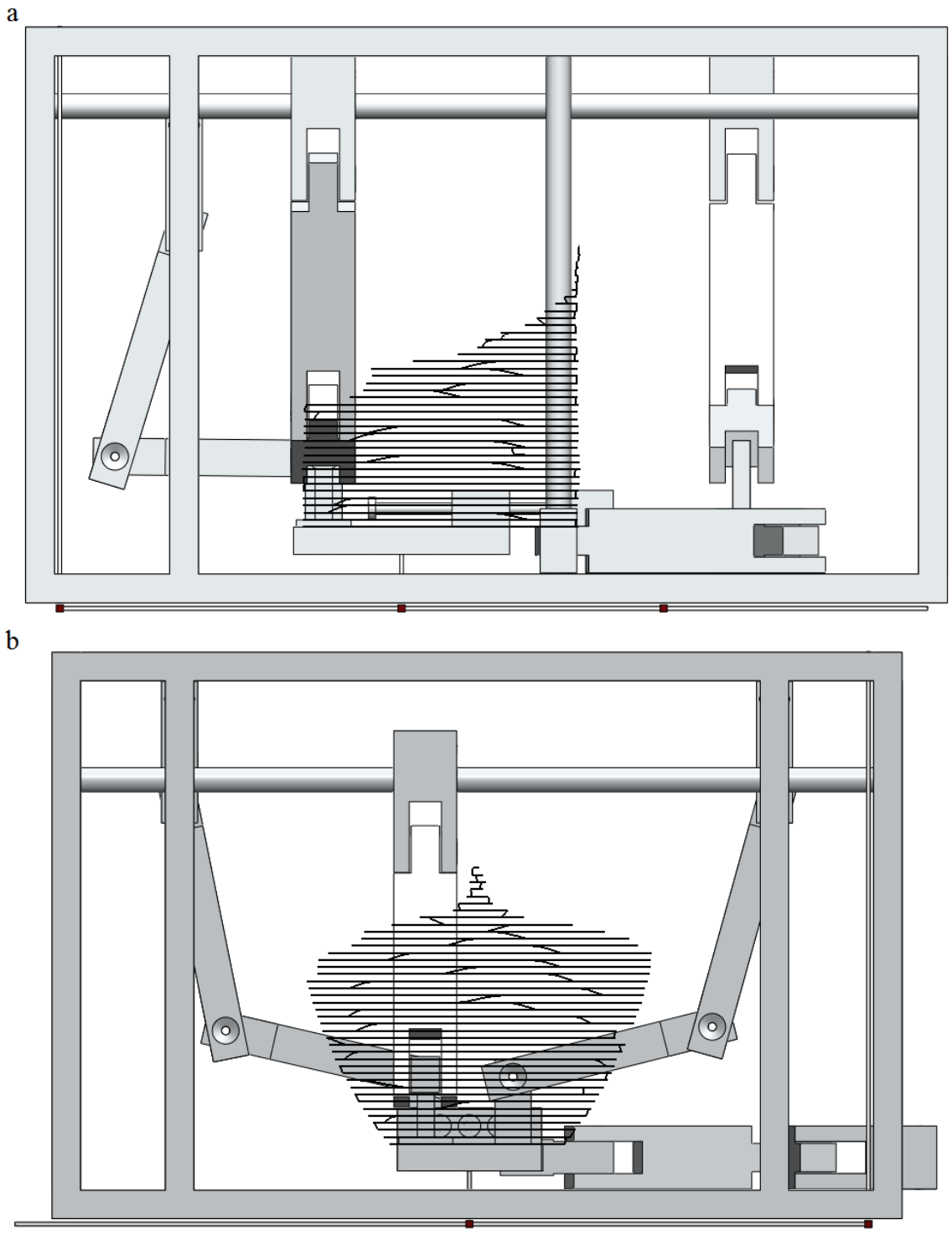

Figure 7.23: Workspaces for the primary platform of concept Design 2 on the a) x-z plane, and b) y-z plane.

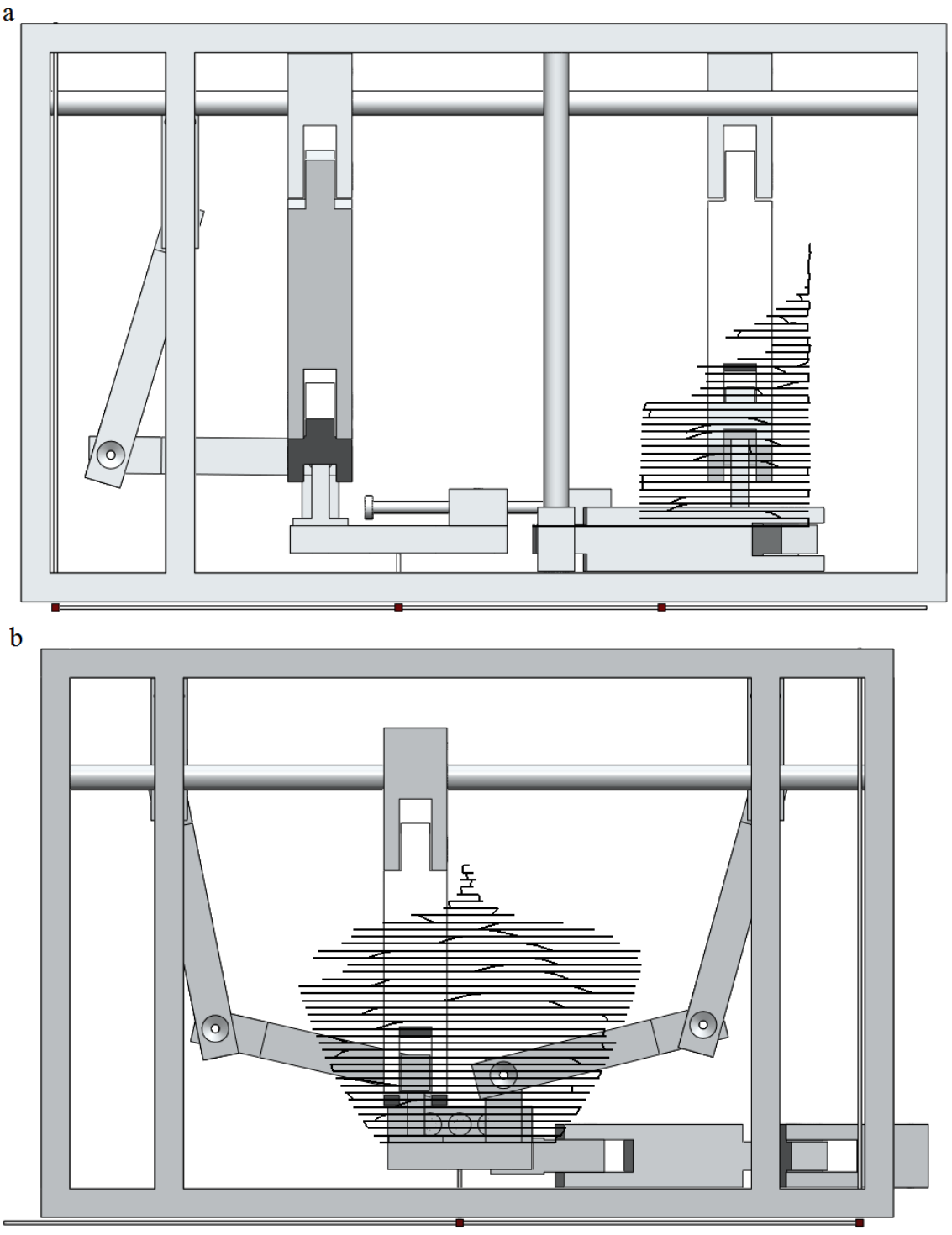

Figure 7.24: Workspaces for the secondary platform of concept Design 2 on the a) x-z plane, and b) y-z plane.

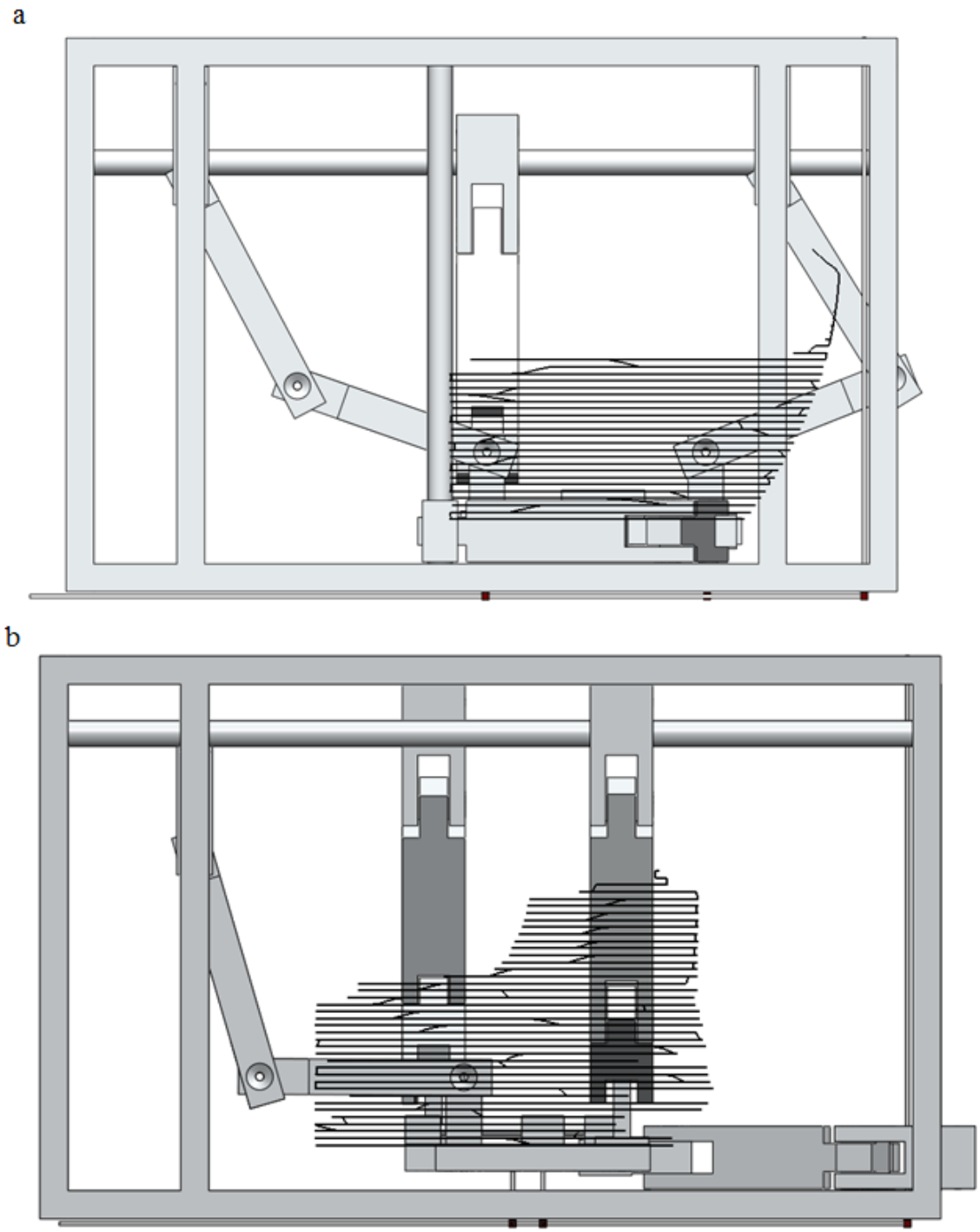

Figure 7.25: Workspaces for the primary platform of concept Design 3 on the a) x-z plane, and b) y-z plane.

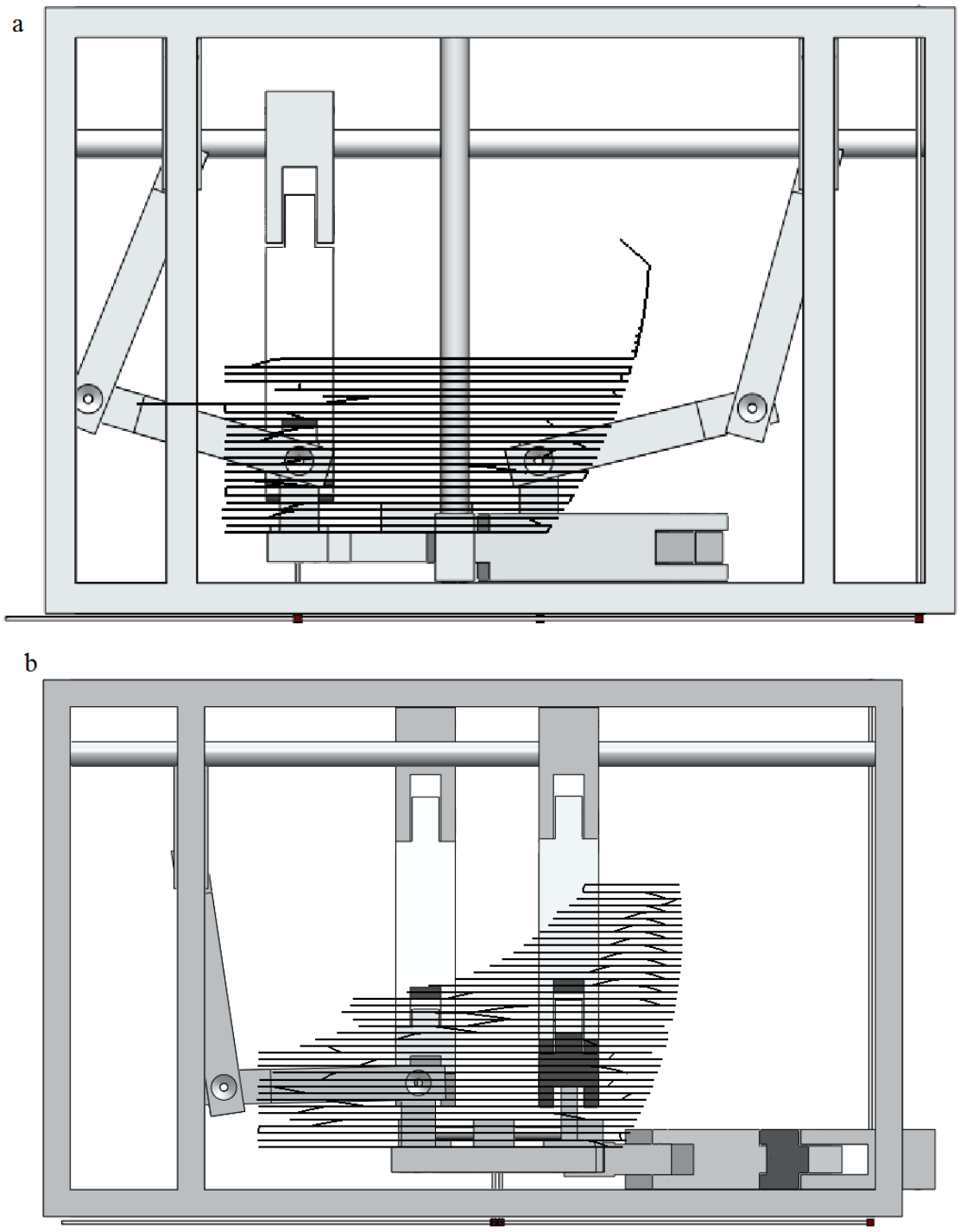

Figure 7.26: Workspaces for the secondary platform of concept Design 3 on the a) x-z plane, and b) y-z plane.

From the data gained from the workspace and static analysis, the decision was made to proceed with the development of the third concept design into a full scale, fully actuated prototype as it has the largest are of cross over in the workspaces of the two moving platforms. With concept Design 3 chosen as the finalised design, the motor refinement method can now be applied to improve the visual of the viable workspace for the two platforms. As stated in chapter 4, this is done by first determining the actual size of the workspace and then producing a motor driven version of the workspace of the

mechanism. The actual workspace displacement for this concept design is taken with reference from a corner section of the cage, which will mean that the displacement graphs will need to be offset in order to produce the final workspace. The workspace displacements for the primary and secondary moving platforms of concept 3 are shown in figures 7.27 and 7.28 respectively.

It should be noted that the displacement required to offset the primary platform results to the origin point of the moving platform is 310.15mm in the x-axis, 52.12mm in the y-axis and 270.5mm in the z-axis. The offset required to set the displacement of the secondary moving platform to its original location at the start of the simulation is 304.34mm in the x-axis, 52.12mm in the y-axis, and 409.49mm in the z-axis.

A profile of the viable workspace can be developed in which the primary moving platform is kept within the range of displacement values gained by the force method. From figures 7.23 and 7.24, the adjusted range of motion for the motor method workspaces for both platforms at the starting position for the mechanism across the x-axis is from -110mm from the origin to 50mm from the origin, the adjusted range for the z-axis is -50 to 150mm from the origin. The values for the rest of the workspace are summarised in table 7.1 for the primary and secondary moving platforms.

Table 7.1: Motor refinement method ranges.

x-axis range (mm)	**z-axis range (mm)**	**y-axis range (mm)**
-100 to 50	-50 to 150	0 to 95
-100 to 50	0 to 150	95 to 110
-50 to 50	0 to 150	110 to 125
-50 to 50	15 to 150	125 to 155

The resulting motor refinement workspaces of the primary and secondary moving platforms are shown in figures 7.29 and 7.30 respectively.

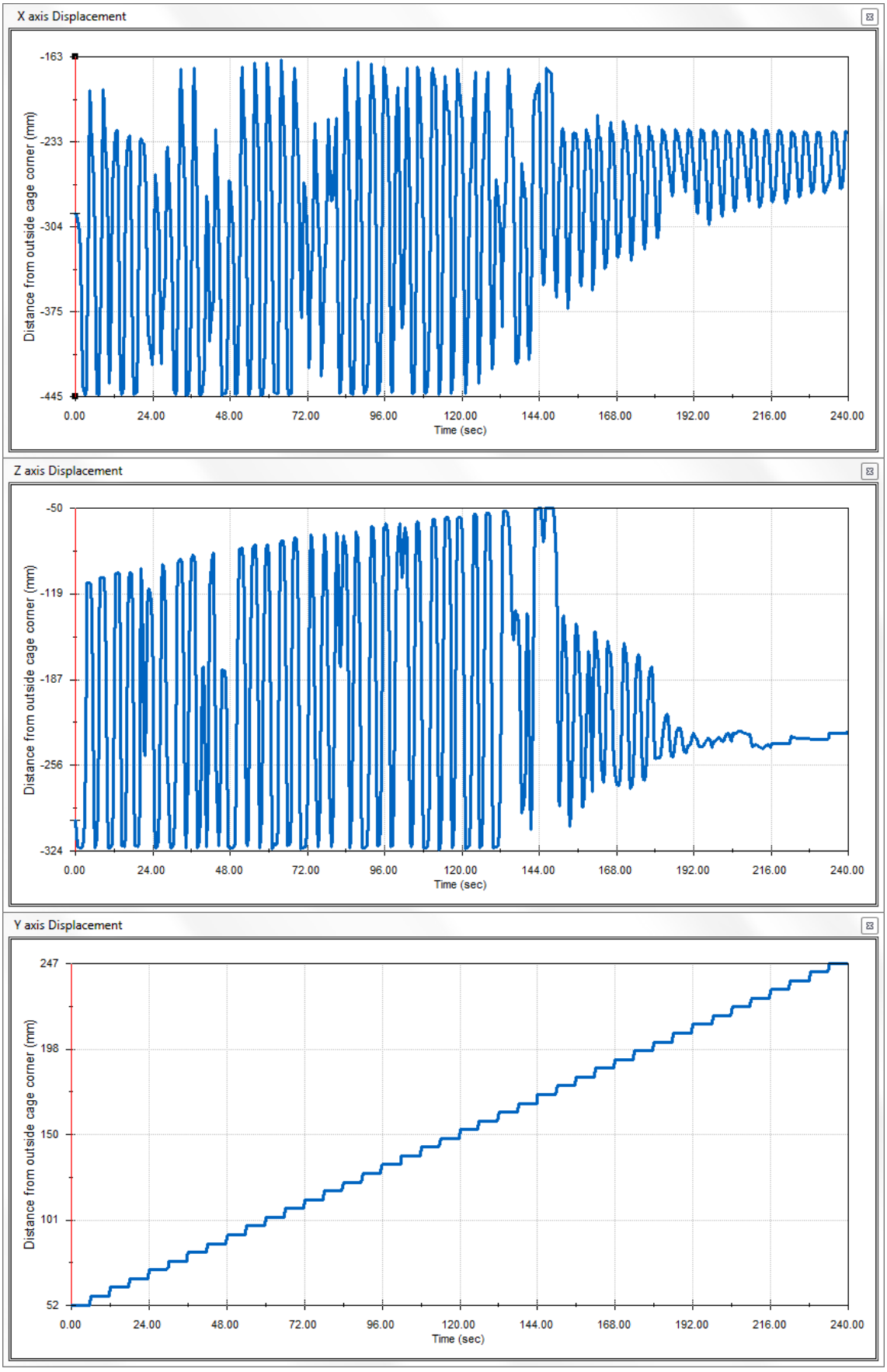

Figure 7.27: Displacement graphs for the primary moving platform of concept Design 3 using the force method.

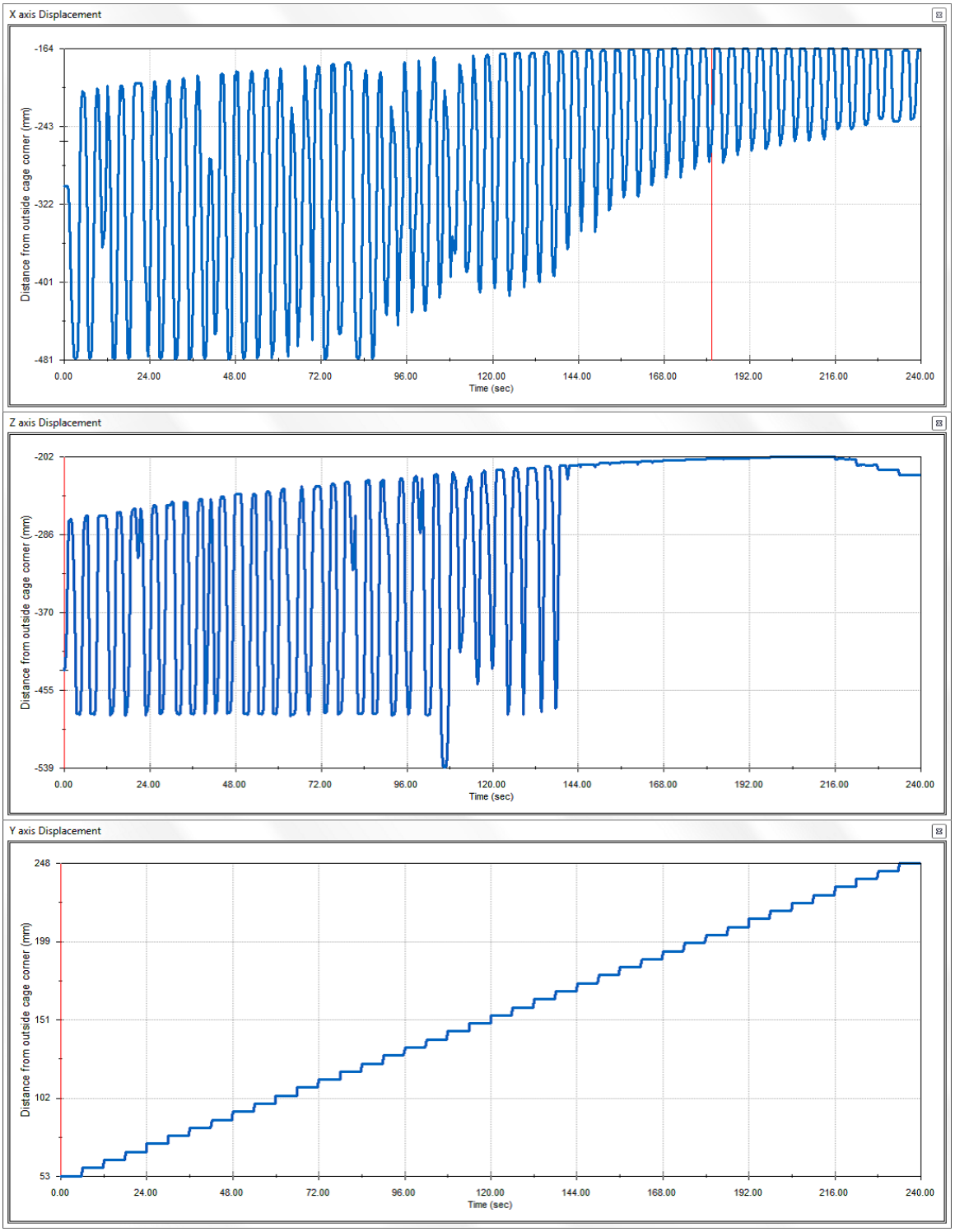

Figure 7.28: Displacement graphs for the secondary moving platform of concept Design 3 using the force method.

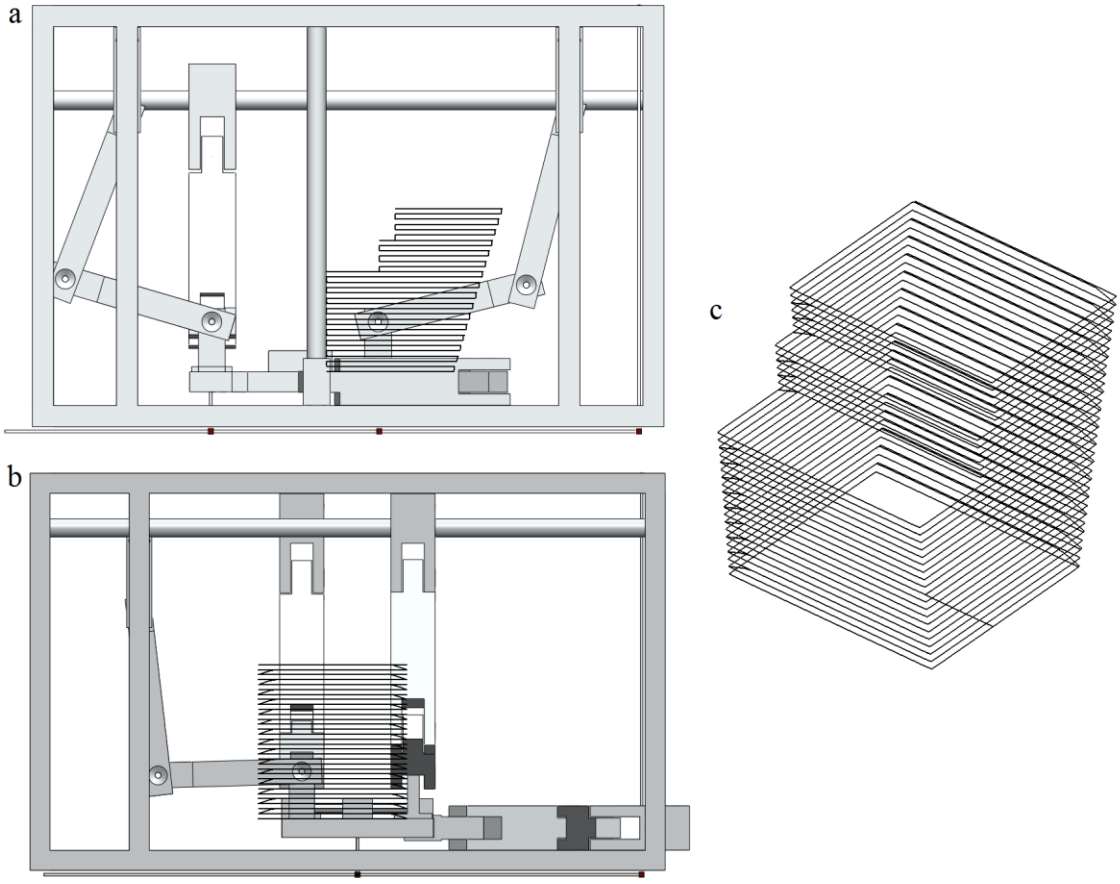

Figure 7.29: Primary moving platform's workspace using the motor refinement method.

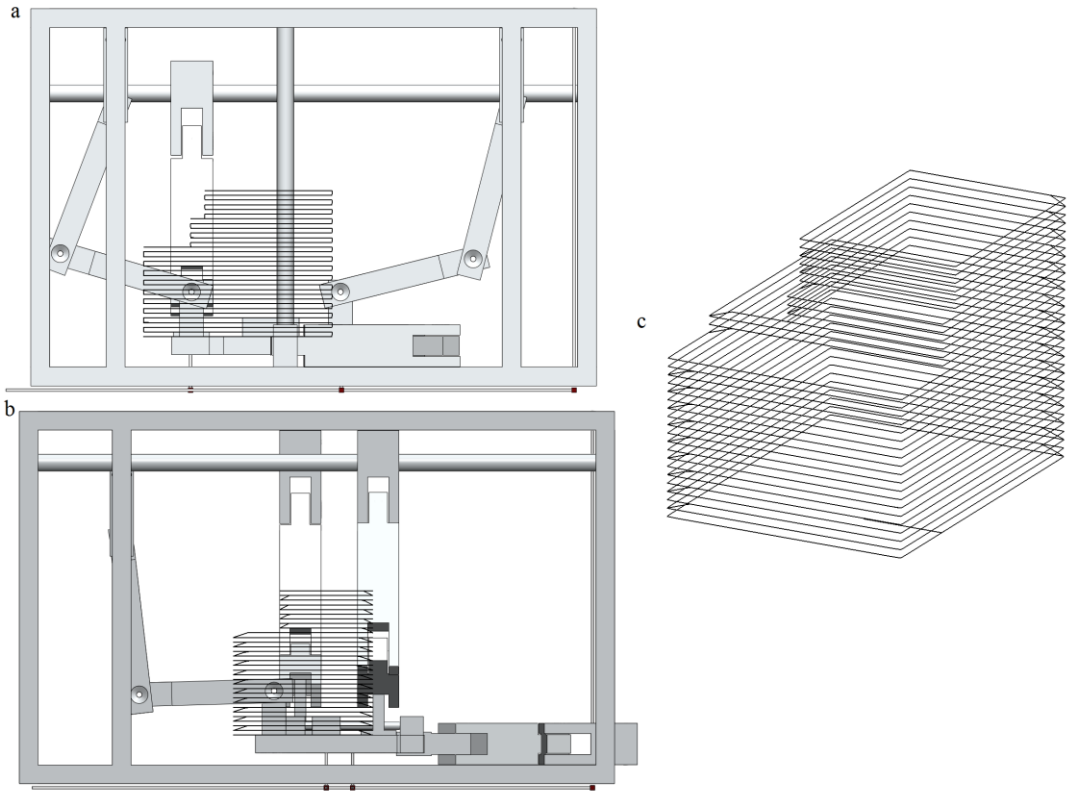

Figure 7.30: Primary moving platform's workspace using the motor refinement method.

7.3 Inverse Kinematic Model

The inverse kinematic model is developed using the virtual chain approach laid out in chapter 5. This method first required the development of the MapleSim multibody model of the proposed mechanism design.

With each leg having a layout of $PRRR$ (figure 7.31), the MapleSim model for an individual kinematic chain is built using a fixed frame which represents the centre point of motion for each arm connected to a prismatic multibody component with the motion of actuation set to the particular axis of motion designated for that particular kinematic chain. The outboard frame of the prismatic component is attached to a subgroup constructed out of two rigid body frame multibody components set to the length of the P-R link and a rigid body multibody component set to the mass of the component. This subgroup is then connected to a revolute multibody component set to rotate about the same axis as the prismatic joint's axis of motion. The remainder of each arm is built using the same subgroup setup only with the lengths and masses modified to allow for the differences in the mechanisms link lengths and masses with revolute components representing the remaining two R joints (figure 7.32). The component's dimensions and masses are listed in table 7.2 below.

Table 7.2: Mechanism link parameters.

Component Name	Length (mm)	Width (mm)	Depth (mm)	Mass (kg)
P joint block	100	44.45	25.4	0.18
Upper limb link	200	44.45	25.4	0.45
Lower limb link	180	44.45	25.4	0.36
Horizontal limb connector	60	12/38.1	25.4	0.05
Vertical limb connector	80	44.45	20/25.4	0.15

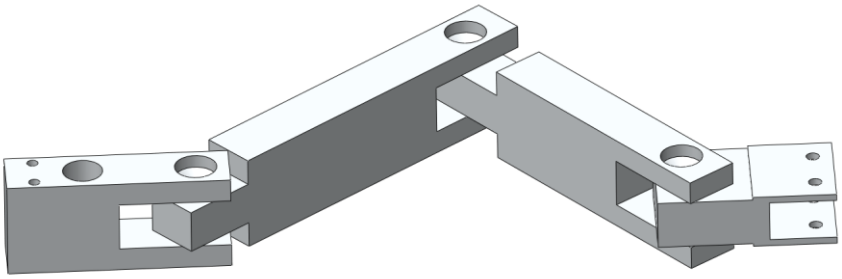

Figure 7.31: $PRRR$ leg.

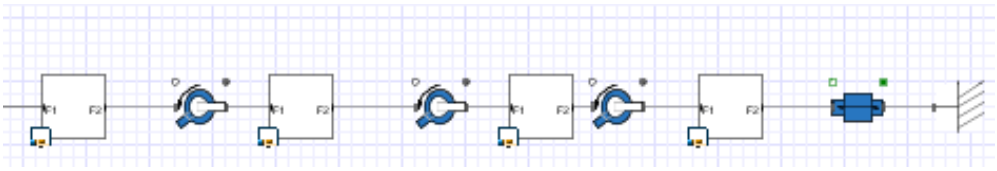

Figure 7.32: MapleSim model representation of kinematic chain.

The moving platforms of the mechanism are produced using a rigid body frame and rigid body components laid out in the same manner as the moving platforms of the mechanism. The two P joints connecting the two platforms are replaced with a single prismatic multibody component attached at the midpoint between the two P joints in order to simplify the model (figure 7.33).

The virtual chain of the mechanism as stated in section 7.1 is produced using two groups of virtual P joints along the shared axes of motion of the mechanism and a pair of P joints with a shared axis of translation. This is produced in MapleSim by placing a fixed frame component at the centre location of motion for the workspace of the mechanism and then attaching the virtual P joint that translates along the vertical axis.

This section addresses the final part of the third objective's first condition as the limitations imposed by the generated workspace are incorporated into the MapleSim inverse kinematic model to ensure that the mechanism does not attempt to leave its own workspace.

7.4 Dynamic Model and Power Comparison

In this section, the method for producing an inverse dynamic model from Chapter 5 is used on the IKM gained from the previous section in order to determine the actuator torques required in order to manoeuvre the mechanism. The second part of this section investigates how the dual platform mechanism compares to a single platform mechanism using the same kinematic limb structure but will only include three kinematic chains and a single T-shaped moving platform (figure 7.34). The power of the two mechanisms will be recorded by probes in MapleSim using the method detailed in section 5.5.

7.4.1 Inverse dynamic model

From the inverse kinematic model, force sensors were added to the secondary mechanism of the MapleSim model and simulated in order to determine the required amount of linear force that would be needed for the actuation of the mechanism.

The methodology laid out in section 5.4.2 for producing the dynamic model of a mechanism is applied to the inverse kinematic model in order to calculate the minimum force requirement for moving the mechanism through a prepared path (figure 7.35).

In order to make the dynamic model as accurate as possible, a pre-planned scenario for the movement of the mechanism was prepared. The mechanism will move its two end-effectors from a starting location to a pair of locations within the workspace of the mechanism and then return to the starting location. The simulation is of a pick-and-place mechanism collecting two offset objects from a conveyor system and then placing them in-line in a depositing area. The layout for this simulation is detailed in figure 7.36 and the detailed motion of the mechanism is given in table 7.3.

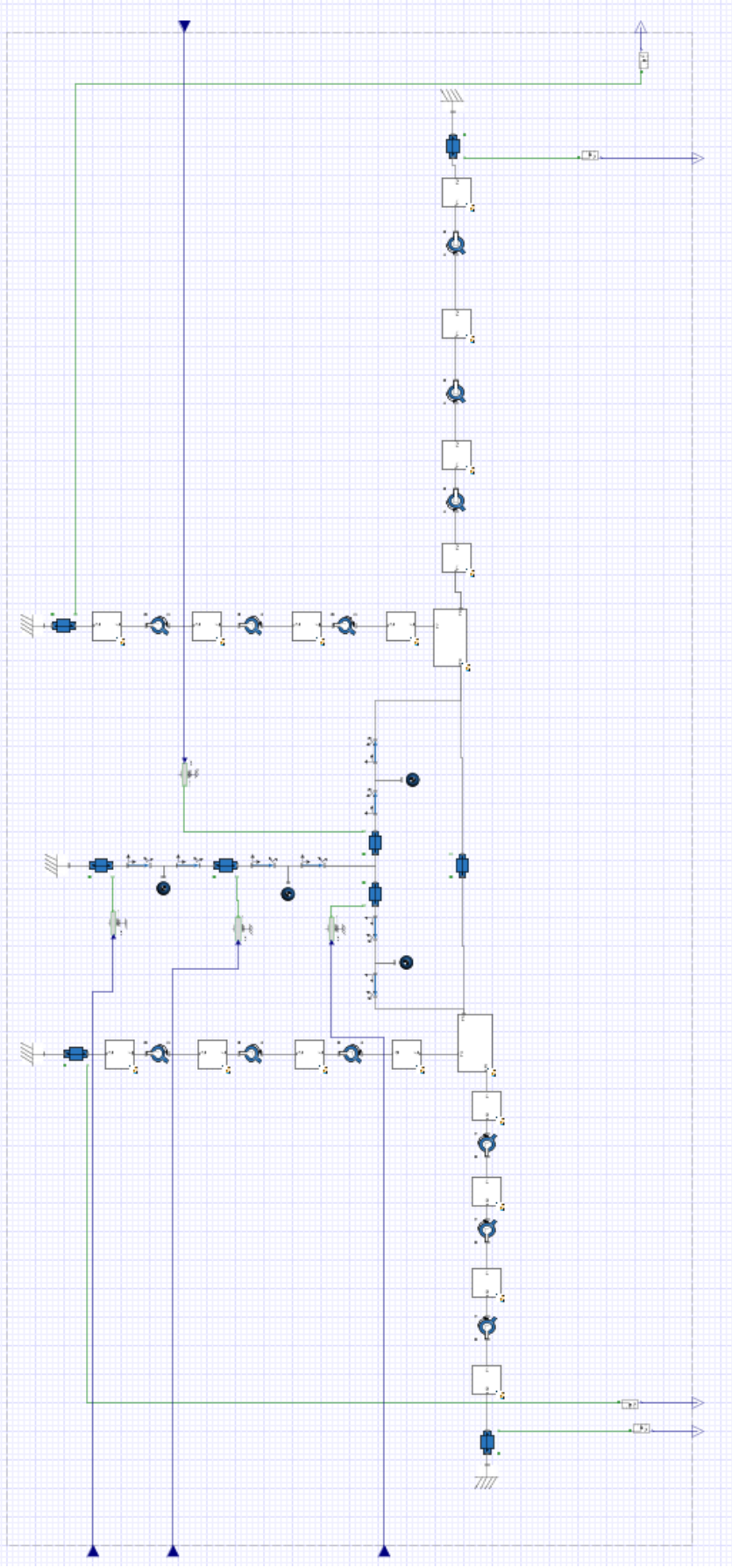

Figure 7.33: MapleSim model representation of full mechanism.

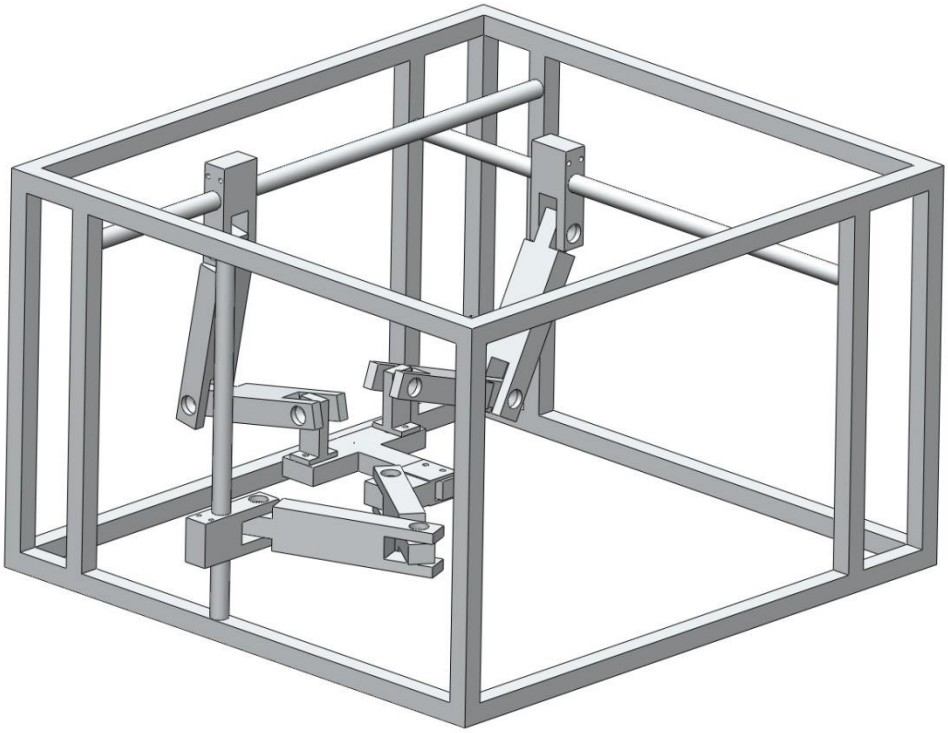

Figure 7.34: Single moving platform version of the prototype mechanism

Table 7.3: Motion path values for the multi-platform mechanism.

Time (s)	X1-axis (m)	X2-axis (m)	Y-axis (m)	Z-axis (m)
0	0.2	0.02	-0.03	0
1	0.015	0.01	-0.02	0
2	0	0.005	-0.01	-0.01
3	-0.03	-0.04	0	-0.02
4	0	-0.005	-0.01	-0.01
5	0.01	0.005	-0.02	0
6	0.02	0.02	-0.03	-0.01
7	0.02	0.02	-0.03	-0.02
8	0.02	0.02	-0.03	-0.01
9	0.02	0.02	-0.03	0

The force values required for actuating the *P* joints for the kinematic chains on the primary platform are displayed below in figure 7.37 and the secondary platform results are displayed in figure 7.38.

From this simulation it can be seen that the required driving force is at a maximum value of 0.75Nm which when converted into a driving torque about the 10mm diameter lead screw gives a maximum required torque value of 3.75×10^{-3}Nm. Using this as a minimum requirement, the servomotors selected for the physical prototype had the capability to produce a maximum torque of 0.84Nm; the robust encoders and the compatibility of the servo drivers with the Labview cRio device and the National Instruments 9415 modules were selected for this project.

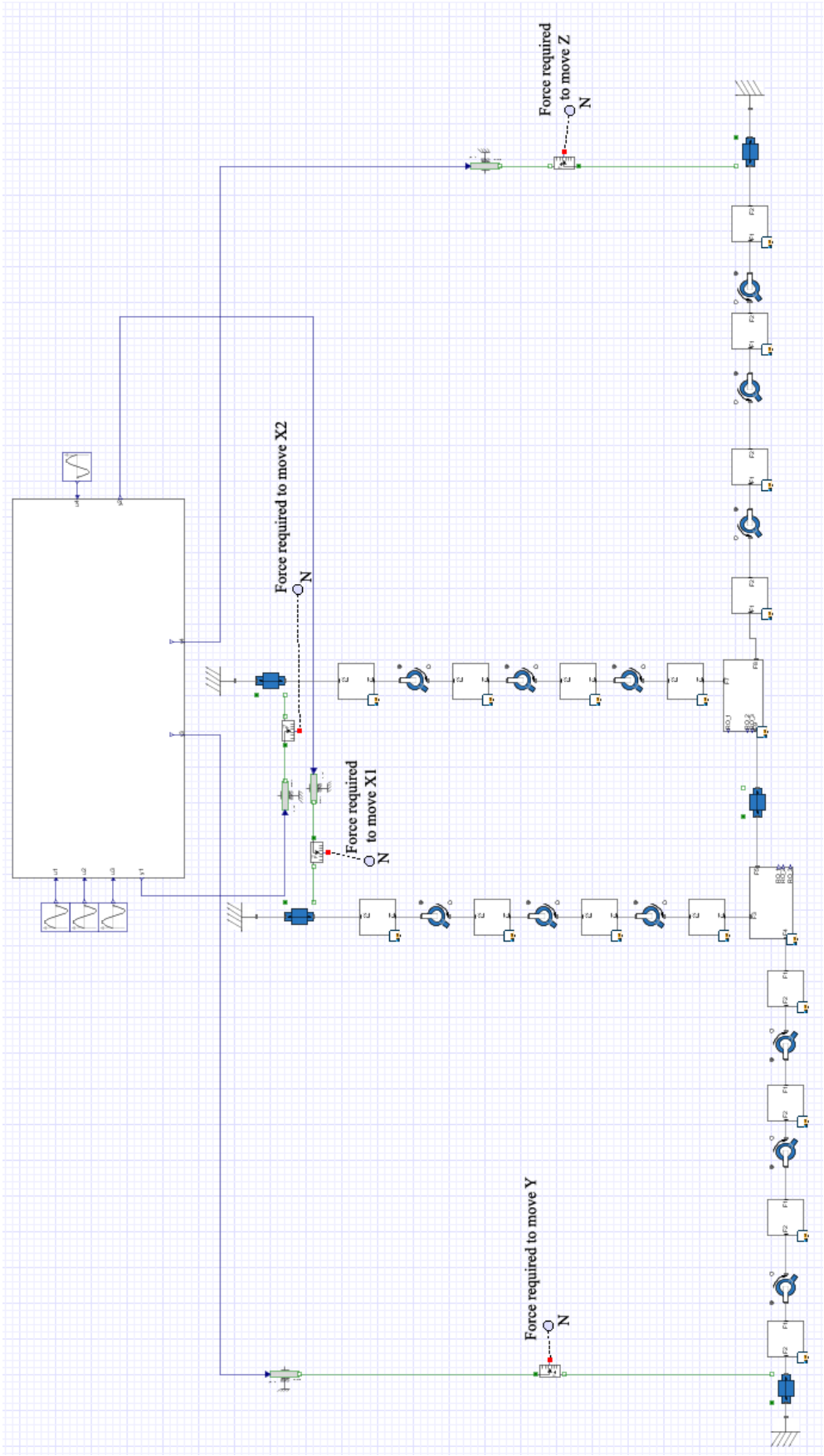

Figure 7.35: MapleSim prototype dynamic model.

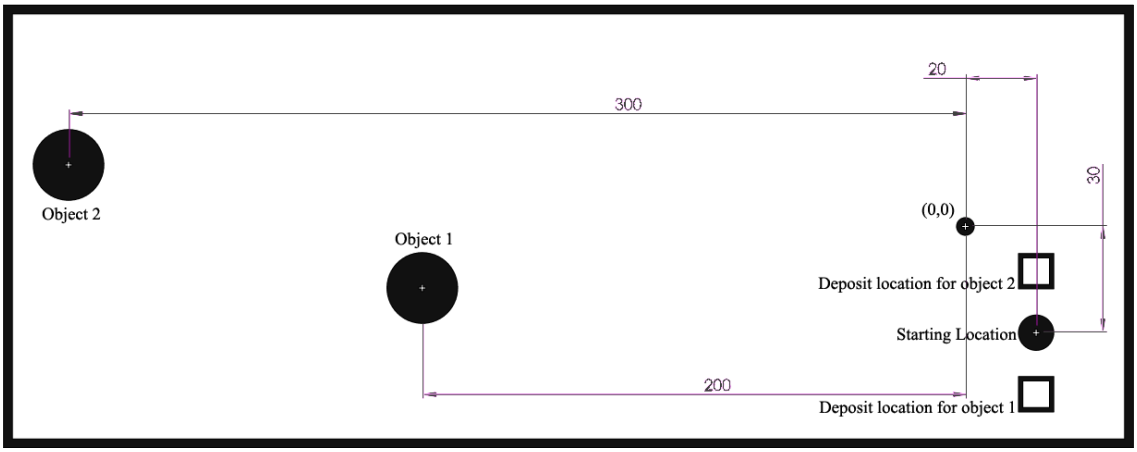

Figure 7.36: Diagram of intended task key locations (all dimensions in mm).

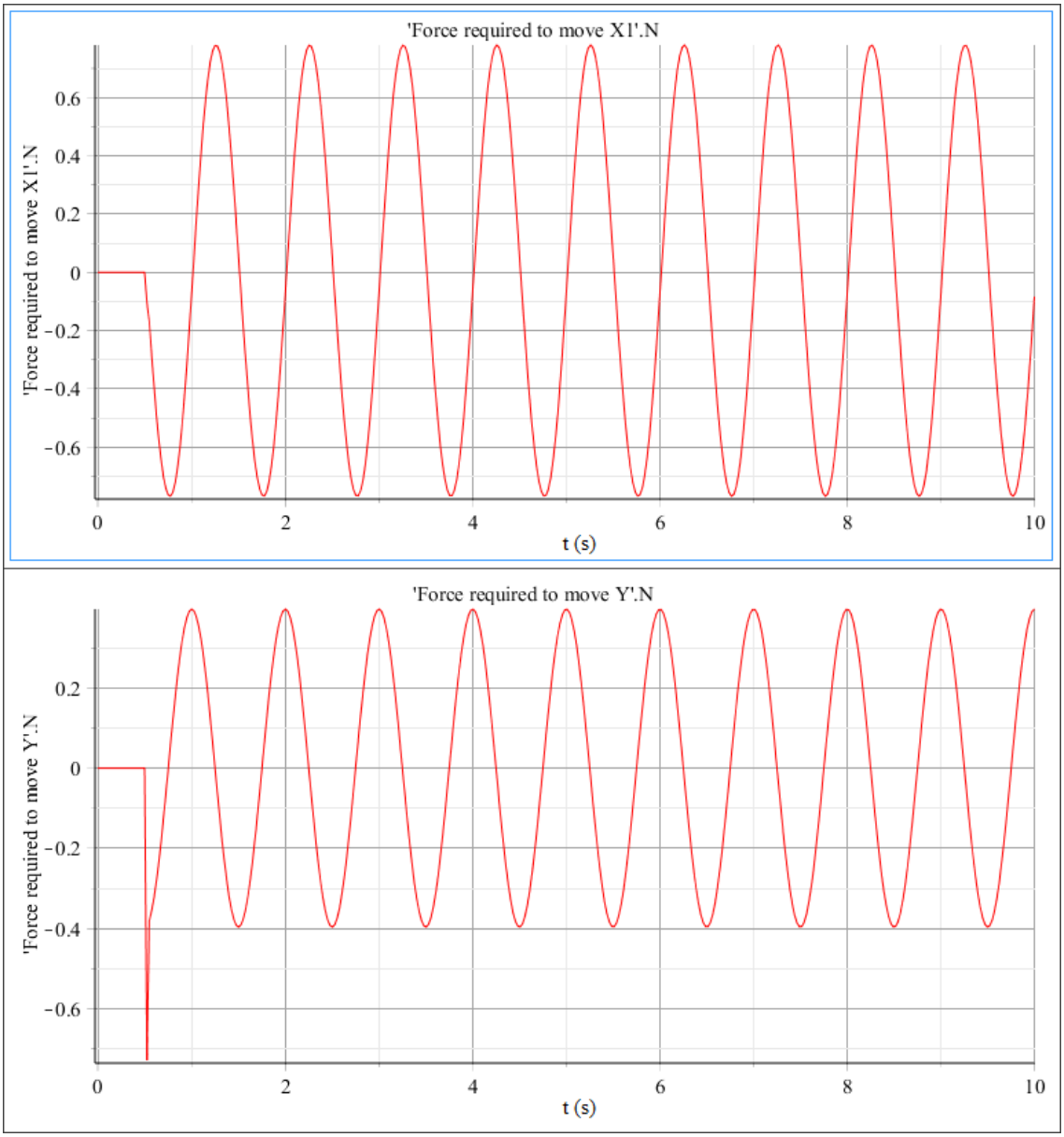

Figure 7.37: Force requirements for the primary platform

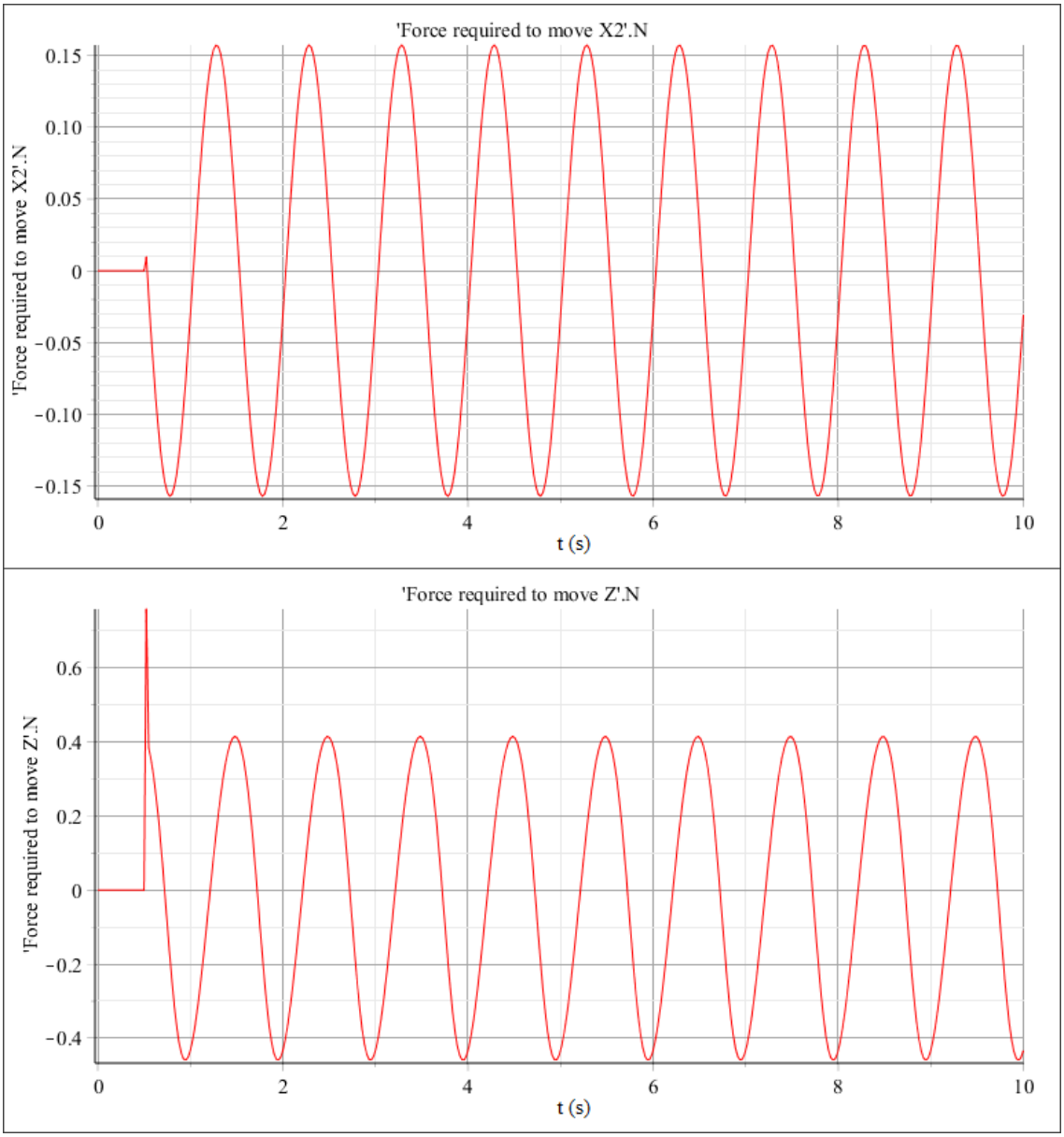

Figure 7.38: Force requirements for the secondary platform

7.4.2 Power comparison with a single platform mechanism

In order to determine the power requirements for the multi-platform and single platform mechanism, the forces of the mechanisms must first be converted to power values. This is done in the same manner as detailed in Chapter 5, where the force/torque recorded at the actuated joint on the secondary model is multiplied to the joint velocity/angular velocity to produce the power of the system. As the joints will be moving in both possible directions the velocity and force profiles will have negative components. This is rectified by applying a vector norm signal block to the output of the product component block, ensuring that all power values remain positive. Next the power signals are added together to produce the total power in signal that in turn is then passed through a Sum signal block to give the accumulated total input power for the mechanism. The MapleSim block diagram for this is shown in figure 7.39.

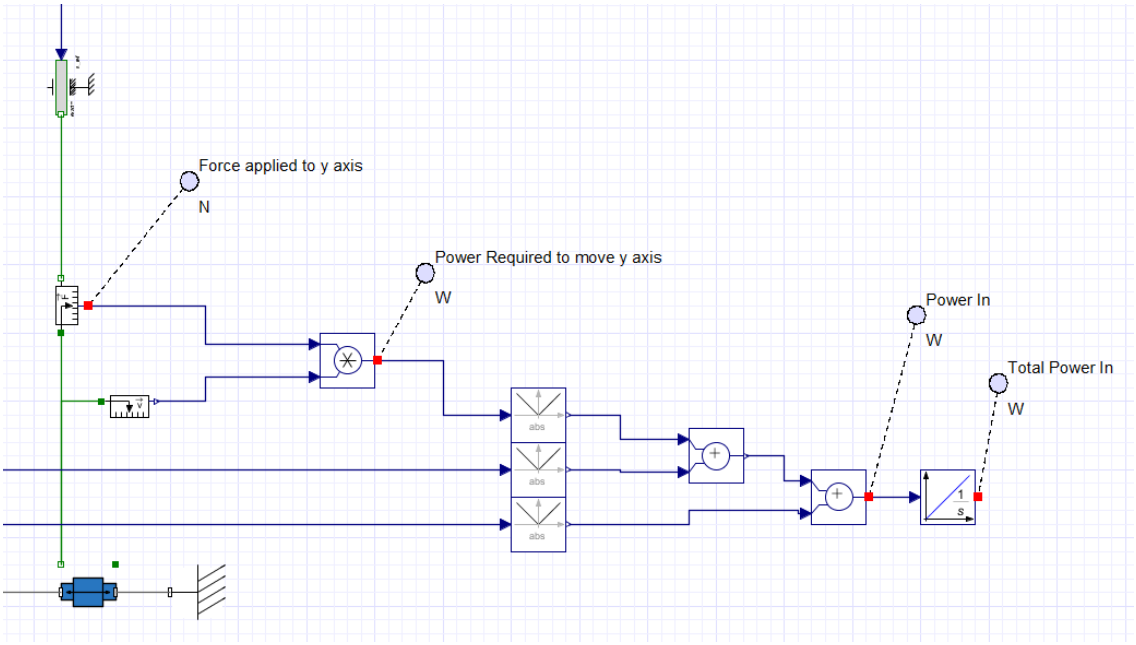

Figure 7.39: MapleSim block diagram of power calculation.

In order to compare two mechanisms, the pre-planned path used in the dynamic simulation of the multi-platform mechanism is used to simulate both mechanisms collecting the objects and returning them to set locations. The path planned for this test will have both mechanisms starting at the same location and then going through the necessary movements in order to complete the task in the same amount of time. The detailed motion of the single gripper version of the mechanism is detailed in table 7.4.

Table 7.4: Motion path values for the single-platform mechanism.

Time (s)	X1-axis (m)	Y-axis (m)	Z-axis (m)
0	0.02	-0.03	0
1	-0.03	0	-0.02
2	0	-0.015	0
3	0.02	-0.03	-0.02
4	0	-0.015	0
5	-0.04	0	-0.02
6	0	-0.015	0
7	0.02	-0.03	-0.02
8	0.02	-0.03	0
9	0.02	-0.03	0

By applying the equation to the MapleSim model of the prototype, the resulting power values are shown in figure 7.40.

As can be seen from figure 7.40, the total power required for the entire manoeuvre is just under 6W. When analysing the data it can be seen that there is a sudden drop in power at the 3-second mark, using the motor refinement method for determining the workspace of a mechanism, the problem can be identified (figure 7.41).

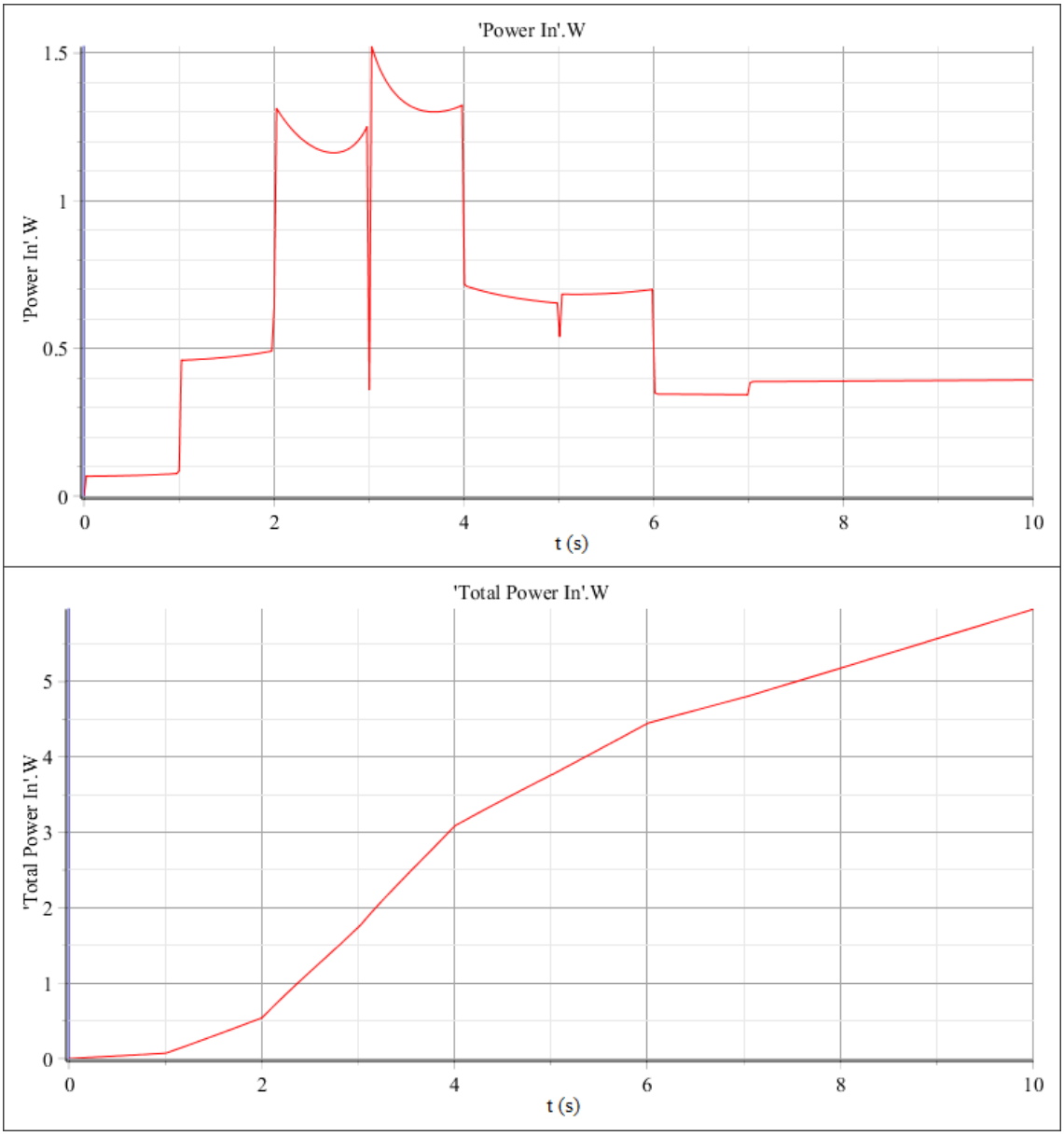

Figure 7.40: Results of power test on the multi-platform mechanism.

From figure 7.41, it can be seen that while one of the upper arm links becomes parallel with the P joint block, another upper arm link has collided with the lower arm link of the same leg; these factors would therefore explain this sudden drop in power. With the value now known for the multi-platform mechanism case, the single platform mechanism shown in figure 7.34 is converted into a MapleSim inverse dynamic model (figure 7.42) and then is simulated using the path detailed in table 7.4. The results of the test are shown in figure 7.43.

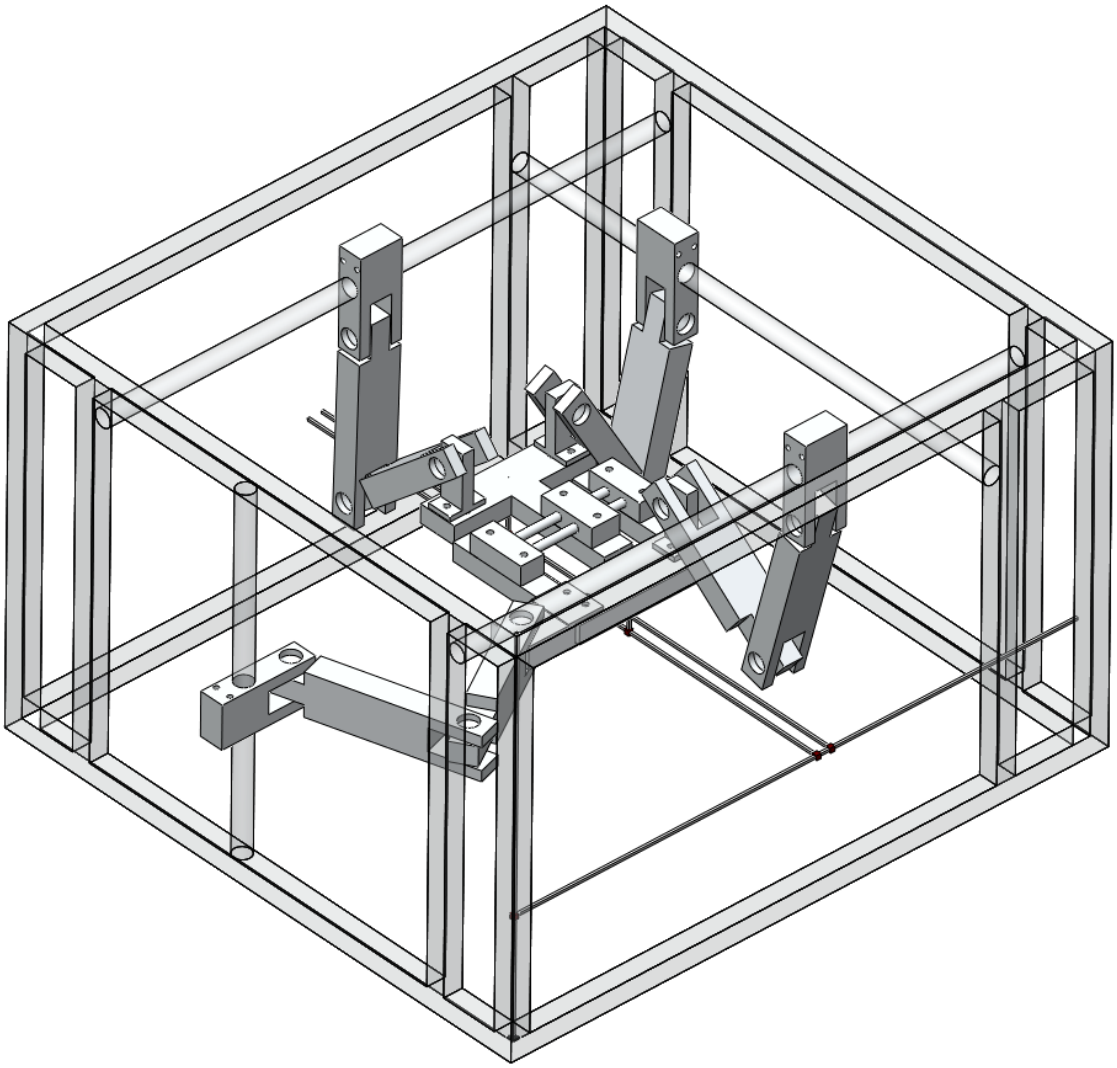

Figure 7.41: Power drop configuration.

As can be seen in figure 7.43, the total power required to run the single platform mechanism through the same task as the multi-platform mechanism and in the same amount of time required is 8.5W. The simulation was then run to determine the power consumed when the multi-platform mechanism operated at the same speeds as the single platform mechanism. The motion path taken for the high-speed test is detailed in table 7.5 and the resulting probe readings displayed in figure 7.44.

Table 7.5: Motion path values for the high-speed multi-platform test.

Time (s)	X1-axis (m)	X2-axis (m)	Y-axis (m)	Z-axis (m)
0	0.02	0.02	-0.03	0
1	-0.03	-0.04	0	-0.02
2	0	0.005	-0.015	0
3	0.02	0.02	-0.03	-0.02
4	0.02	0.02	-0.03	0

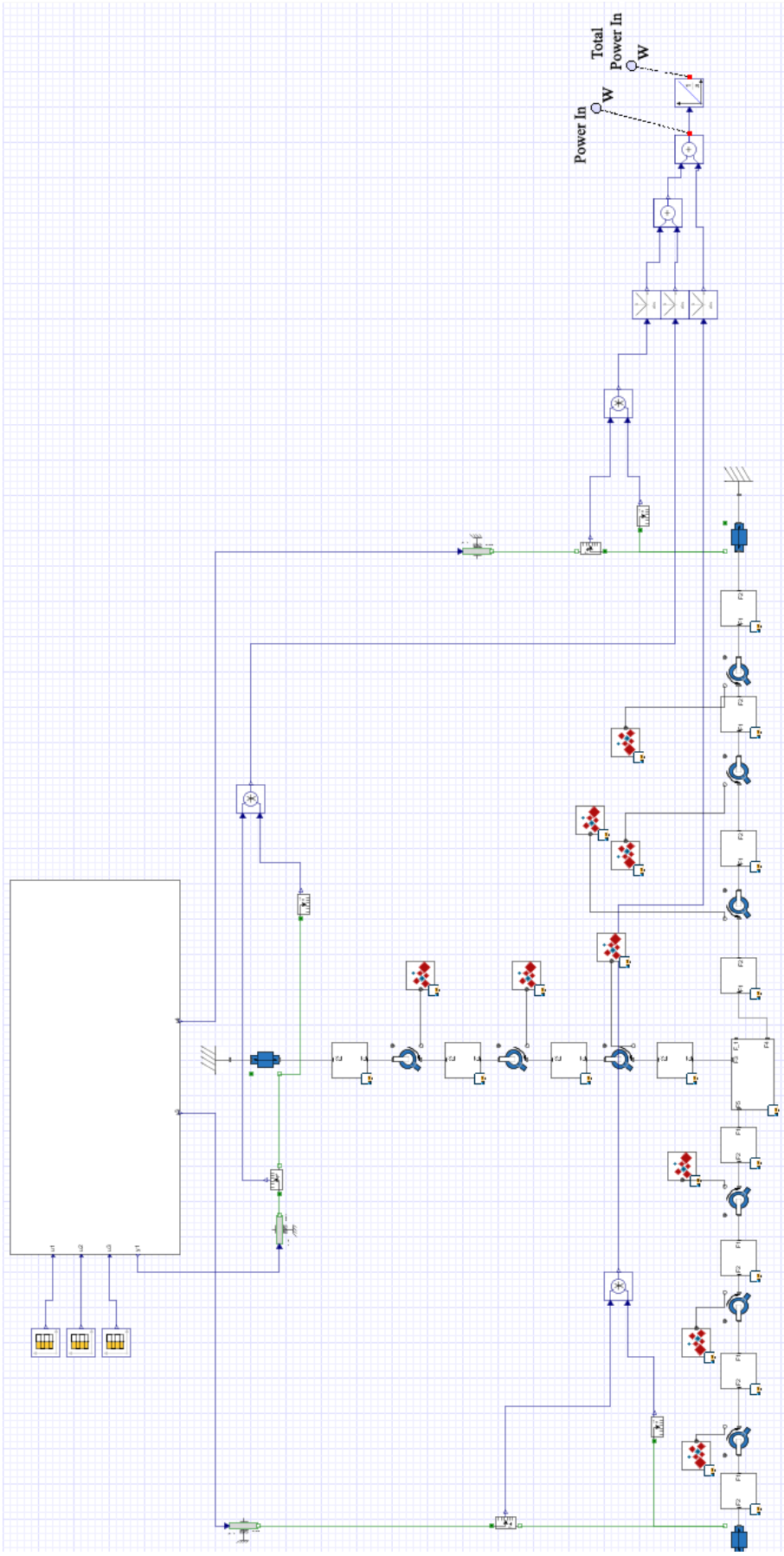

Figure 7.42: MapleSim inverse dynamic model of the single platform mechanism

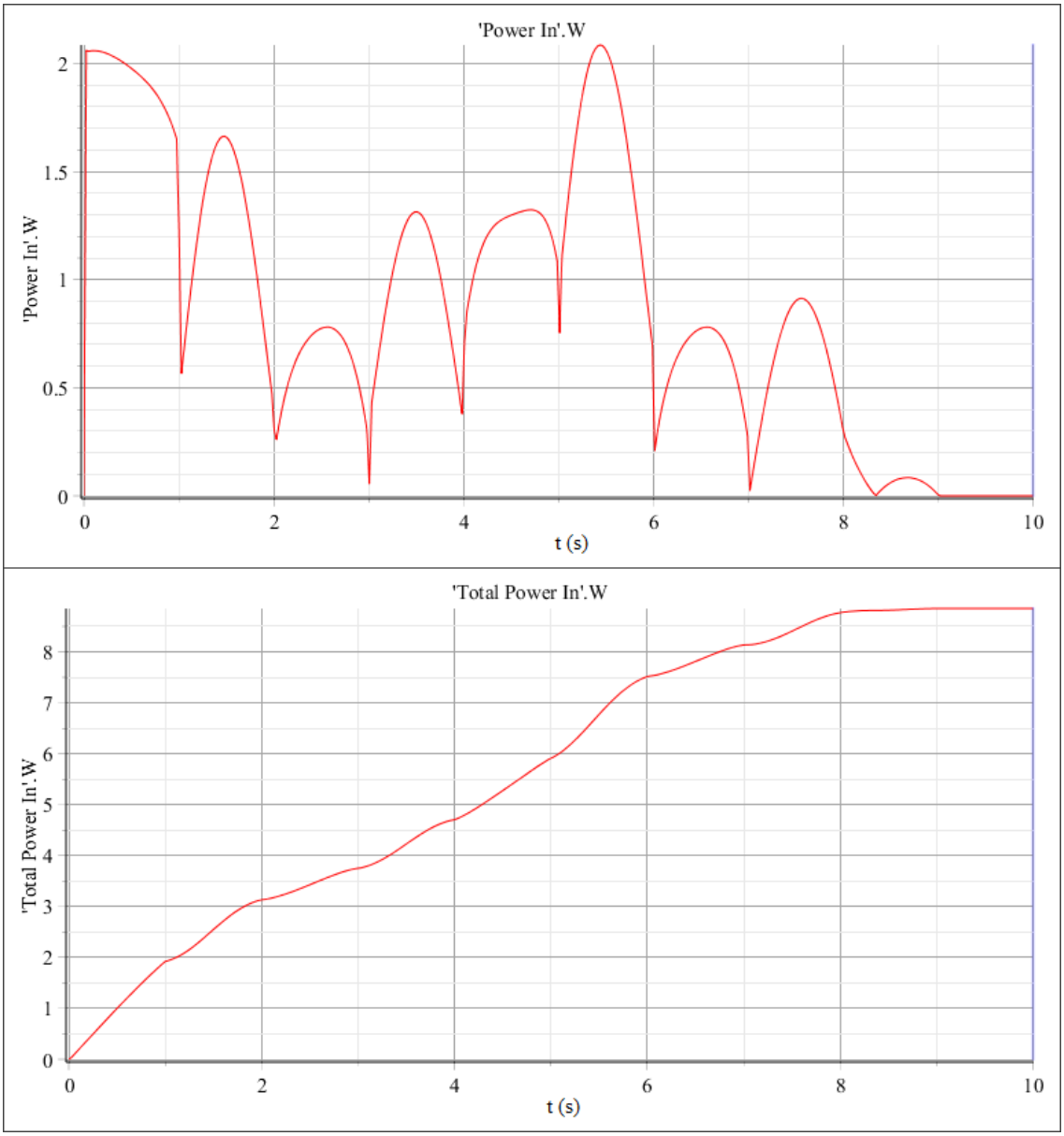

Figure 7.43: Results of the power test on the single-platform mechanism

Figure 7.44 shows that the total power requirements for the multi-platform mechanism to perform at a high enough speed to match the operational time of a single collection of the single gripper mechanisms is about 8.1W of power. From these results, it can therefore be determined that the multi-platform mechanism is a more efficient mechanism when compared to its single platform variant as both simulations of the multi-platform mechanism resulted in a lower total power requirement when compared to the single platform.

This section fully addresses the fourth objective of this thesis as the displayed method not only provides an additional method for determining the mechanical efficiency of the mechanism but also allows for the comparison of two similar mechanisms to determine which mechanism would have the lowest operational costing. This therefore answers the third research question as this method, when combined with the method given in Chapter 5, allows for a generic methodology for determining the cost and energy efficiency of a mechanism to be determined.

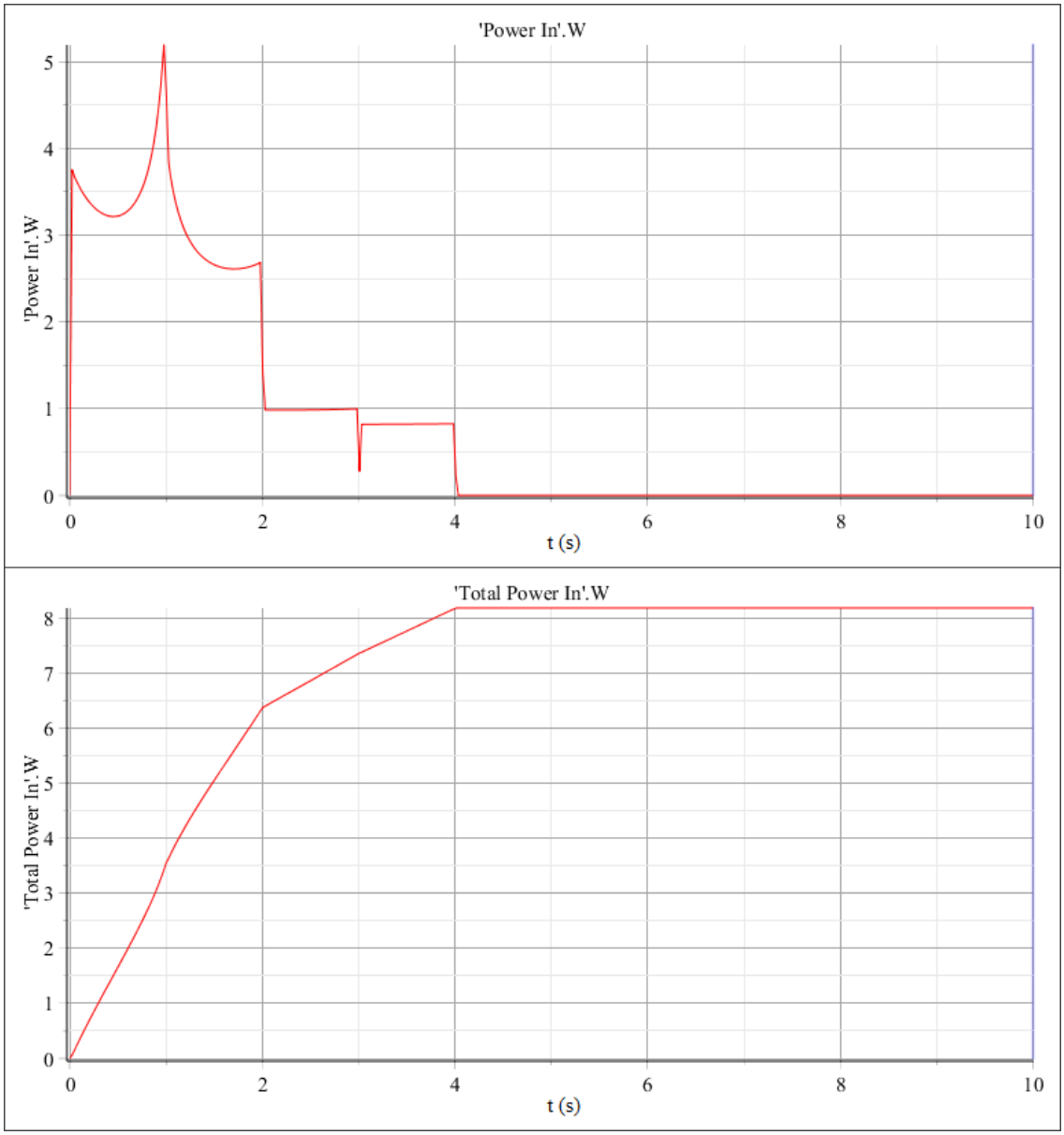

Figure 7.44: Results of the high-speed multi-platform power test

7.5 MapleSim to Labview Control System

Following the technique laid out in chapter 6, the subgroup of the inverse dynamic model for the prototype displayed in figure 7.35 was converted to a .dll file for use as an external model in the Labview control system. The initial setup for the VI simulation loop was performed on the .dll file, with the external model being placed inside of the simulation loop and the input/output nodes being wired to the external walls of the loop. Lastly, the entire program was enclosed within a While loop function with a stop button wired to the exit clause. The input values were initially wired into numerical increments and the output nodes were wired up to numerical indicators so that the .dll file could be tested to ensure there were no issues with the conversion.

The next step in the development of the control system was to determine how the VI would be constructed to work with the servomotors and the .dll file. Normally, a simulation file can be uploaded to an external controller like the cRio device that we are using however, as the cRio device uses a Linux based operating system and the .dll file

is a Windows based file type, the file could not be uploaded. Therefore, rather than have a single VI that would allow the entire program to be run from the cRio, the control system was split into two, the simulation VI and the servomotor VI.

Having confirmed that the MapleSim model had indeed been converted correctly, the inputs to the simulation were wired into a division function block along with a constant value of 1000. This modification allows the input of movements in metres and for the control system to convert them to millimetres as the simulation is designed to expect millimetre input for the virtual chain joints. The outputs from the simulation were likewise wired into multiplication blocks with a constant value of 1000, to convert the values back to metres, and then passed into another division function block along with a constant value block set to 3 (determined by the pitch of the screw thread) in order to determine the number of rotations that are required to complete the move. The resulting values are then placed into global variables labelled by the axis of motion the signal represented along with the suffix "Desired Position".

Once this was accomplished, control elements were added to the user interface so that the user could set an end position without the mechanism moving while the position was being set. This was done by enclosing the entire Labview program in a case structure and adding a Boolean toggle switch that when the switch is set to true will run the simulation to determine the motion of the mechanism. The resulting front panel GUI for the control system is displayed in figure 7.45 below.

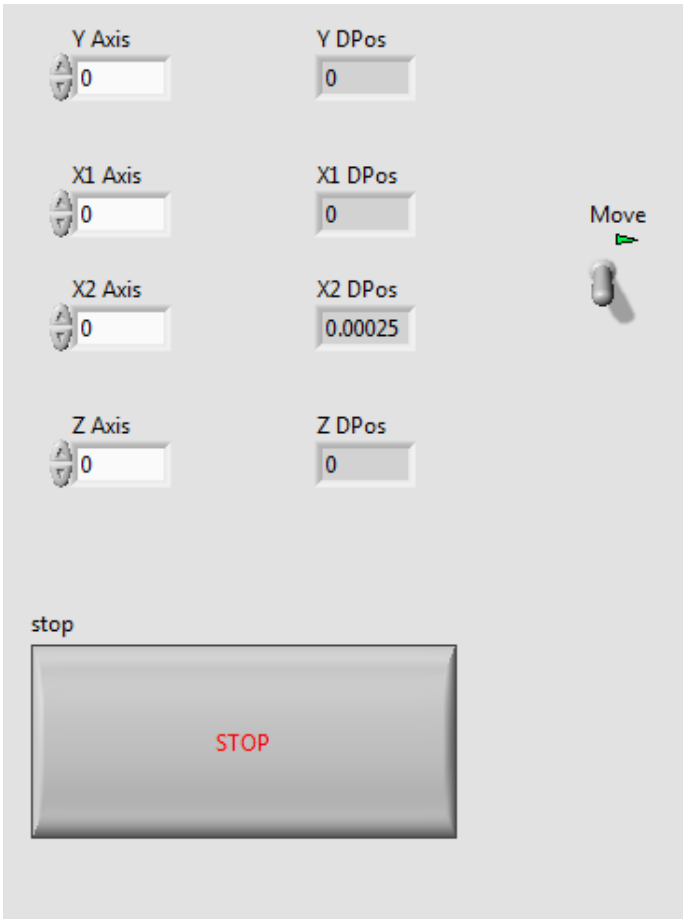

Figure 7.45: GUI of the prototype control system.

In the "false case" window of the case structure, a series of global variables also labelled with the axis of motion the signal represents along with the suffix CPos which represented the current position of the P joint along a given axis. These variables were then wired into the numerical indicators labelled in the same manner with the suffix being DPos (desired position). The last step for setting up the VI was to add a timing function of 100ms to ensure that false signals were not sent during run time and a sequence structure added outside of the while loop so that on initialisation of the control system, the preset starting position for the various P joints were loaded into the system. The finished VI is displayed in figures 7.46 and 7.47 for the true and false cases respectively.

With the simulation of the mechanisms working in a control system format, a servomotor VI that communicates the number of rotations to the servomotors is available. The servomotor drivers come with a custom Labview block that allows the user to incorporate the motors into a VI. Applying a While loop to the new VI, the first components to be added to the VI were the Straight-Line Move controller blocks for the motors. These component blocks allow for a motor position (number of turns), motor velocity and motor driver to be assigned to the block in order to control the motor when ran. For this chapter the motor velocity for each motor was set to 2rps so that the mechanism could move at speed but would not be at risk of damaging itself in the event that something went wrong.

The position value connector for the straight-line move block was connected to the exit node of a subtraction function block that takes the desired position of the mechanism via a global variable and subtracts from it the current position of the motor in order to determine the number of rotations required to reach the desired position and fix the motor direction.

This section of the VI was then placed into a sequence structure that passes Boolean values from the first frame to the second which return a high value if the desired position of the mechanism and the current position of the mechanism are the same. In side of the second structure frame, a series of special Read blocks, designed to take information out of the motors allows the VI to read the current number of rotations the motor made from its zero position. These position nodes are wired into the global variables for the axes CPos variables. Finally a shift register is placed inside of the While loop and around all the other components to allow the CPos values to be fed back around to the start of the program.

Lastly, the initialisation of the CPos variable values was completed in the same sequence structure setup as in the simulation VI. The completed VI block diagram schematic is displayed in figure 7.48.

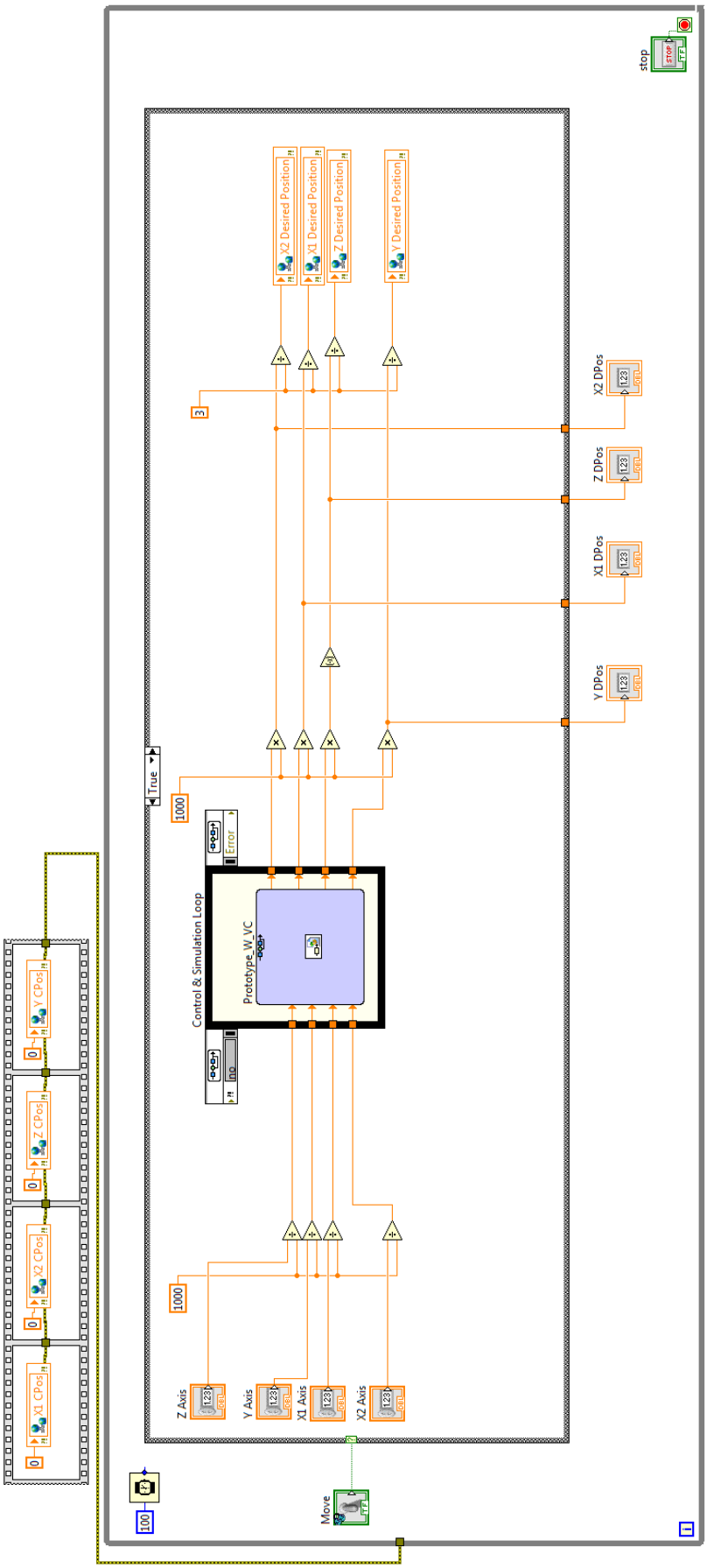

Figure 7.46: Simulation Labview VI true case.

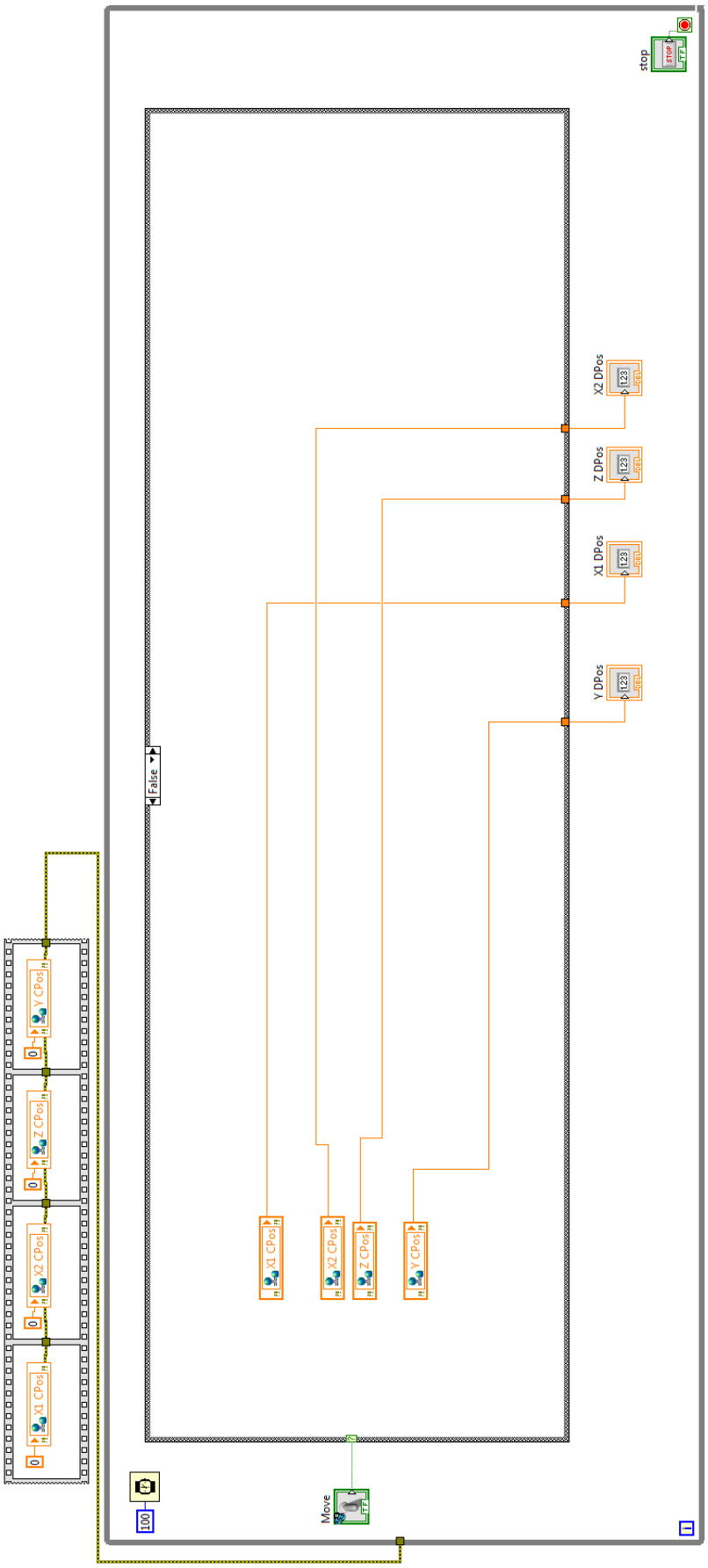

Figure 7.47: Simulation Labview VI false case.

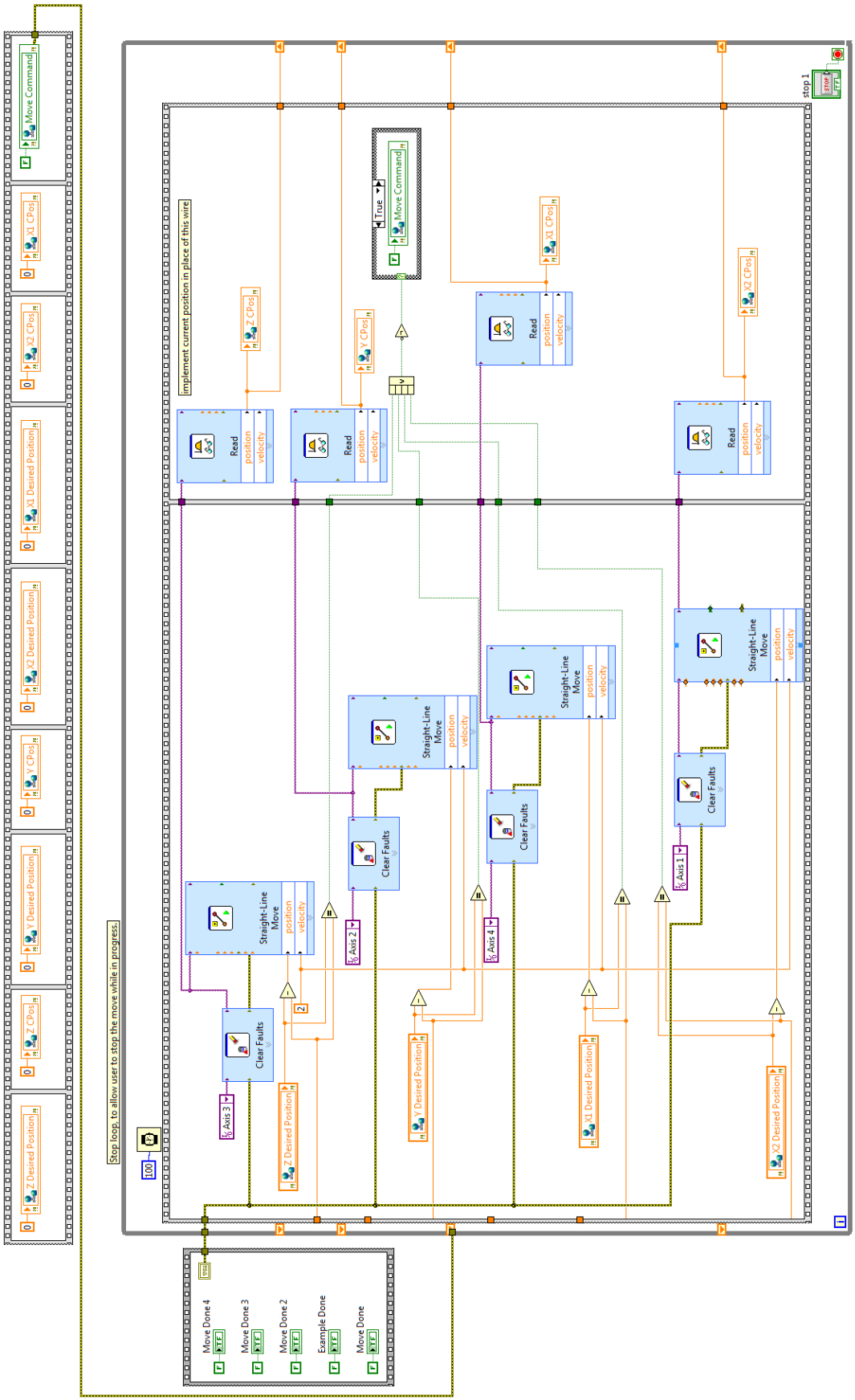

Figure 7.48: Servomotor Labview VI.

This answers the second research question as the concept of virtual chains have been utilised to produce a fully graphical-based approach to the development of a parallel mechanism and the remainder of this chapter will investigate the fabrication of a physical prototype to confirm this assumption.

7.6 Physical Prototype

With the concept dimensional design of the mechanism finalised, the components of design were modified to include cover plates that would prevent the R joint pins from slipping out of the mechanism during motion. From the results of the refined motor method for chosen design, the decision was made to include the presence of chamfers on the corners of the kinematic chains links to improve the workspace reliability. Lastly, the moving platforms of the mechanism were modified to include a cut out section where the grippers had to be inserted and the locations in which components are attached to each other with screws reworked to represent the screw thread sizes intended in the physical prototype.

Lastly the bearings and R joint pins were drawn up and added to the CAD model of the mechanism to ensure that all the components fit correctly (figure 7.49a).

Taking the design concept of the non-actuated cage of the prototype as the base of the mechanism, a more refined cage was designed to house the mechanism and its servomotors. As the motors are large and require at least 100mm of cable length before the cables become flexible, the motor cages were placed outside of the main cage. This prevents the mechanism from losing any of its viable workspace due to a kinematic chain potentially colliding with another chain's motor cage.

The cage was designed using 20 x 20mm aluminium profile extrusions, with 20 x 20mm three way cubic connectors being used to secure the corner sections and inner brackets to connect the vertical and horizontal sections that extend part way along an extrusion to the opposite side of the cage. The reasons extrusions were selected was due to their low cost, availability and robust design. The resulting design cage is detailed in figure 7.49b.

In order to finalise the design of the mechanism, the method for connecting the kinematic chains to the cage was chosen; this was done through the application of aluminium guiding blocks designed to allow the kinematic chain's lead screw to pass through and be connected to the servomotors while securing the lead screw to the cage. The guiding blocks also serve as a stopper to the aluminium guiding rods that have been designed to ensure that the kinematic chain does not rotate during motion.

Finally, the supports for the servomotors of the mechanism had to be designed so that the motors could be mounted onto the cage. First however, the servomotors themselves needed to be measured and built into a CAD model so that the mounting plates could be made accurately.

With the servo motors modelled, the plates were designed to be produced from 3mm thick sheet aluminium that would first be cut out by a laser cutter and then be bent into shape using a bending press.

With the final components modelled, the various CAD models of the kinematic chain components, moving platform components, servomotors, servo motor mounting brackets

and cage were compiled into a single CAD assembly file to produce the finalised design shown in figure 7.49c.

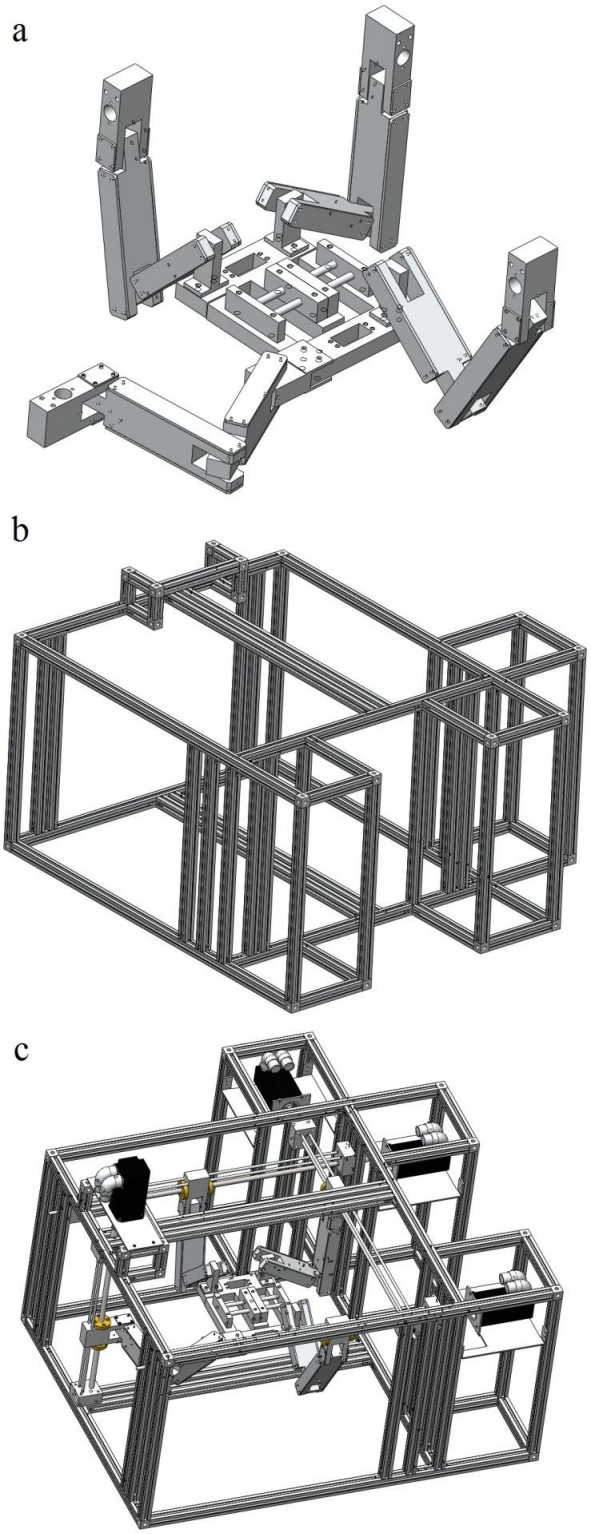

Figure 7.49: a) Fully assembled prototype kinematic chains and moving platform with pin covers. b) Fully assembled prototype cage CAD design. c) Fully assembled prototype CAD design.

7.6.1 Prototype build

With the intention of reducing the cost of production the entire kinematic chain and moving platform assembly were designed to be produced from a single block of 6 inch by 1 inch aluminium flatbar. This flat bar section was cut into the components of the kinematic chains, moving platform and guiding blocks. Finally the various rods and lead screws were turned down to the required dimensions.

With the components of the mechanism fabricated, the KJN cage was built to the layout designed in Figure 7.49b. With the cage finished the kinematic chains were constructed as in the layout detailed in figure 7.49a. Once the kinematic chains had been constructed it was discovered that the pin joints were acting as a tolerance fit with the joint's bearings due to the stock being slightly larger than the ordered size. This meant that the face plates for the kinematic chain's links were no longer necessary as the contact with the inner walls of each of the joint's bearings would meant that the pin would be incapible of sliding during initial testing, allowing for the kinematic chains to be taken apart and modified with ease if necessary.

Once the kinematic chains were constructed, the *P* joint linear bearings were mounted into the *P* block guide which was then mounted onto the T shaped moving platorm. The 8mm aluminium bars were then slid through the guiding block and then secured into the two *P* joint holding blocks before they were mounted onto the two protruding legs of the U shaped moving platform.

Finally, the components for connecting the moving platforms to the kinematic chains were attached to the ends of their respective kinematic chains and then mounted onto their designated moving platforms as shown in figure 7.49a.

With the mechanism constructed the lead screw nuts were placed into the slots cut out into the actuated *P* joints and the 10x3mm lead screws were wound through. The 5mm aluminium guiding rods were then passed through each of their *P* joints before mounting the ends of each lead screw were fitted through the aluminium guiding blocks before one end was slotted into the servomotor coupling. Lastly the mechanism was mounted to the cage and the servomotor couplings were aligned with the servomotors of the mechanism.

Once the mechanism was mounted, it became clear that the mechanism had a small amount of deflection along the three horizontal lead screws that increased the required torque of the motors to beyond their specifications.

This was solved by adding to each kinematic chain a steel plate that runs above the lead screws and attaching them to the two guiding blocks on either side and to the *P* joint block of the kinematic chain via a slotted hole. This allows for the *P* joint to maneouver through its maximum desired range of motion while preventing the mass of the mechanism from bending the lead screw. This modification is shown in figure 7.50.

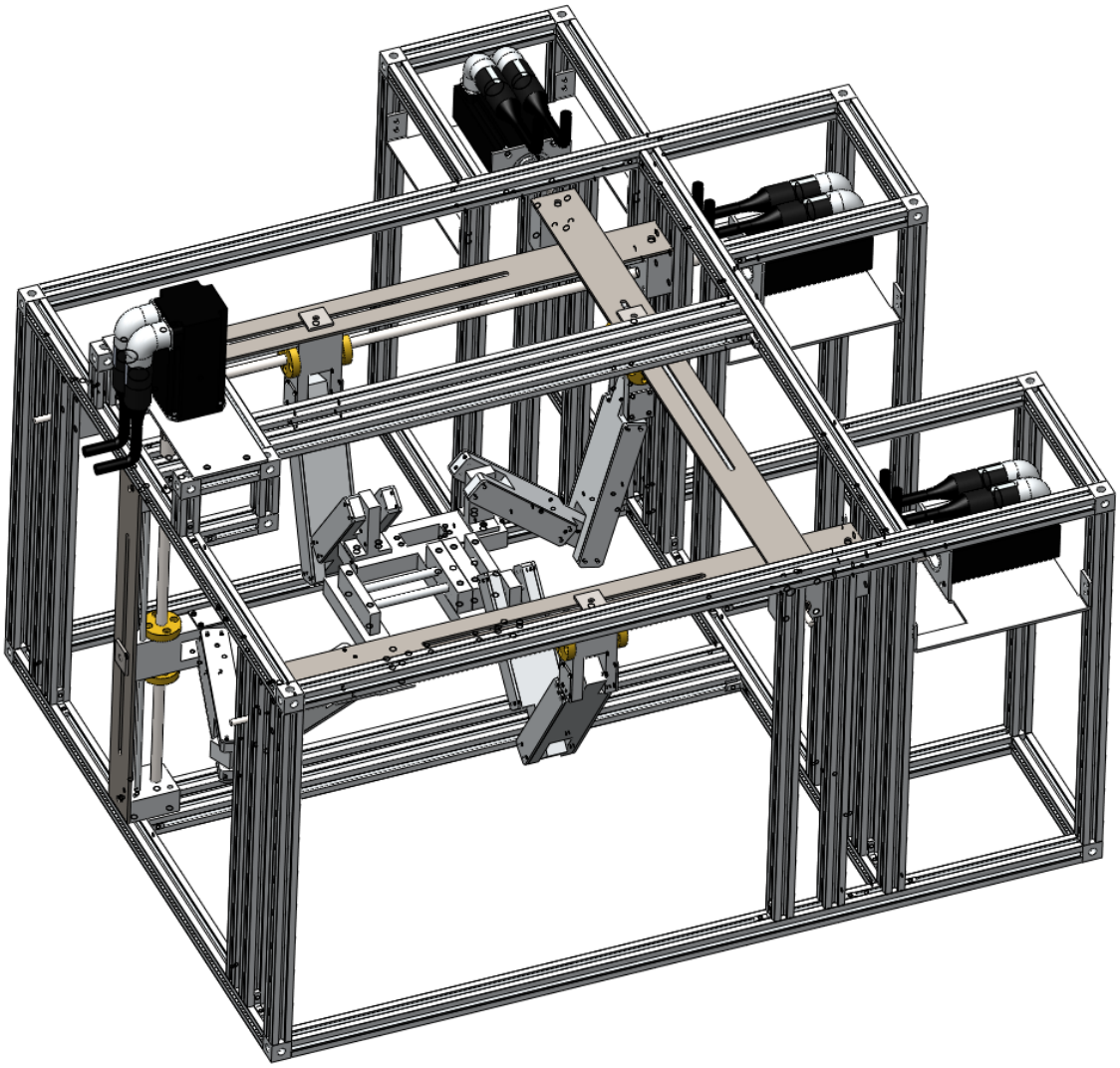

Figure 7.50: Final prototype design with steel plates.

The material supplies for the components of the prototype mechanism were then ordered and subsequently fabricated in the University's Mechanical Engineering Workshop. The mechanism was then constructed to produce a physical prototype as shown below in figure 7.51 with a close up of mechanism shown in figure 7.52.

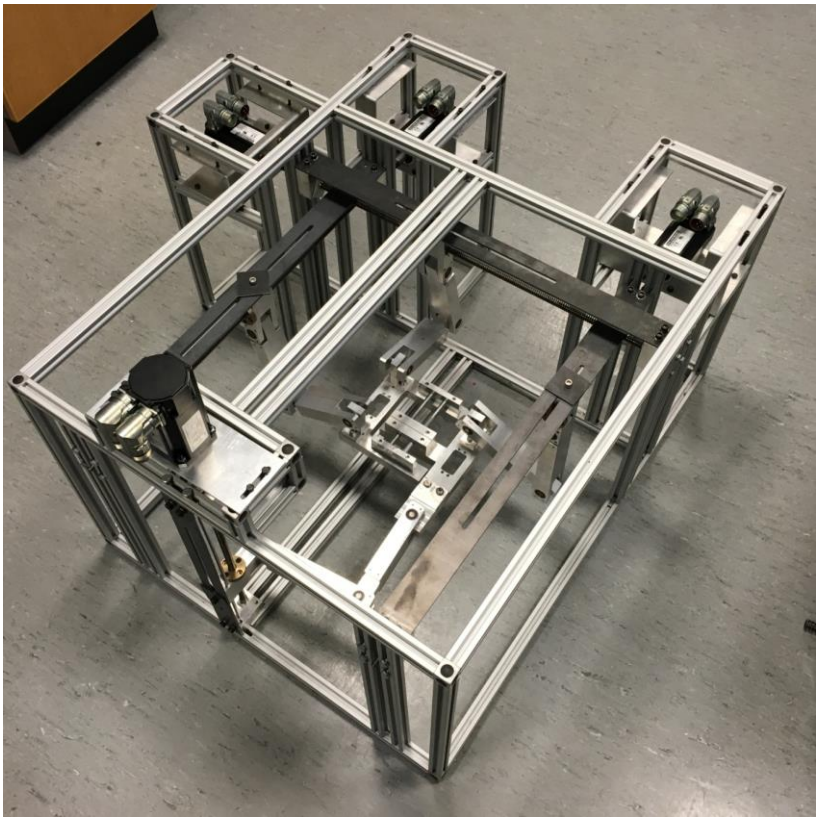

Figure 7.51: Physical prototype mechanism and cage.

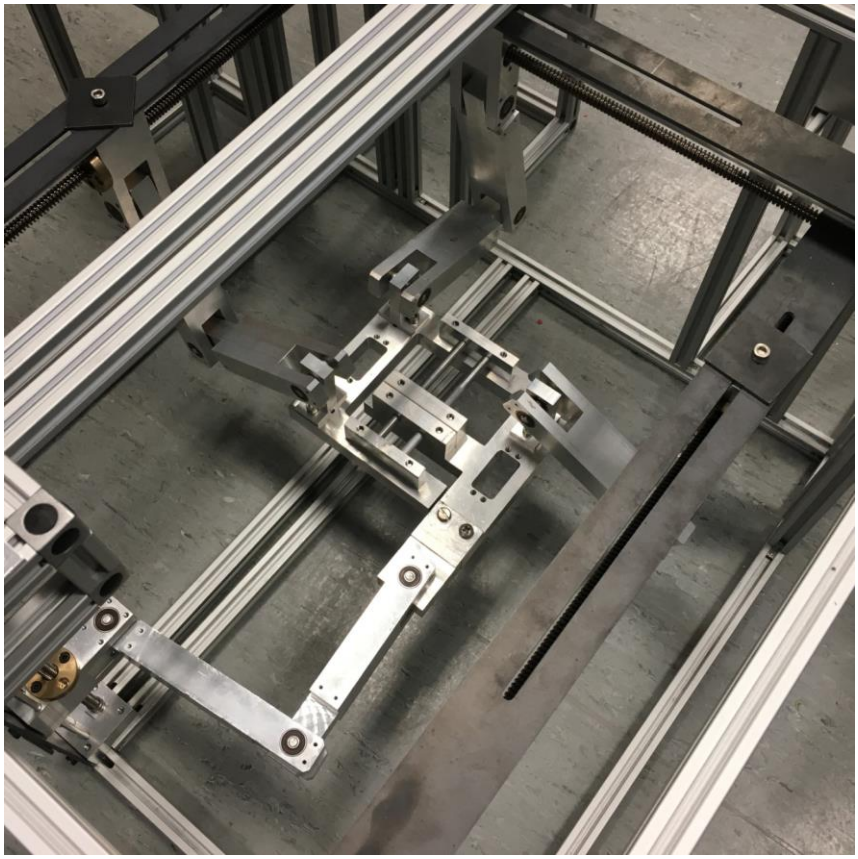

Figure 7.52: Close up of physical prototype mechanism.

7.6.2 Testing of prototype

Using the feedback information from the servomotors as a digital check and the motion of the mechanism via a visual check, the parallel mechanism was tested using the control system designed in section 7.5. From the motion of the mechanism, it was seen that a small amount of deflection occurred at the moving platform that, while this did not prevent the mechanism from moving, did hamper its accuracy as the kinematic chain for the vertical P joint extended outwards. This was rectified by adding a pulley-counterweight system above the central aluminium extrusions and attaching the central arm of the T-shaped moving platform to the non-weighted end. This improved the motion considerably and allowed the mechanism to complete a path programmed into the Labview control system, which proved that the MapleSim inverse kinematic model was working accurately.

7.7 Summary

In this chapter, the novel changes to the design process laid out at the end of Chapter 6, the various steps of which were developed in detail in Chapters 3-6, was implemented into the development of a new parallel mechanism that incorporated the multiple moving platform extension of type synthesis. The dynamic modelling techniques produced in Chapter 5 were successfully implemented to determine the forces required to operate a physical prototype of the chosen design. The inverse dynamic model was then modified to determine the total power requirements necessary to operate the prototype mechanism through a series of pre-planned points to simulate the mechanism in motion.

The mechanism design was then compared to a single platform in order to determine which mechanism required the least amount of power to manoeuvre through a predetermined exercise. This was done with the assumption that if both mechanisms utilised the same motor setup then the efficiency of the mechanism could be determined simply through which mechanism required the most power to move it. The test showed that while the multi-platform design required less power than that of its single platform counterpart, it did enter a singular configuration during the motion resulting in a power drop. This discovery will alter the current flow diagram of the proposed changes to the design process, as this stage will potentially identify any singular configurations not identified previously in the development process.

The inverse kinematic modelling technique was applied to the design to allow for the motion of the mechanism to be simulated and then implemented into a Labview control system that allowed the finished mechanism to be controlled as intended.

The workspace design methodology gave insight into the most suitable moving platform link arrangement and allowed for the addition of steel plates in the final mechanism to ensure that there was no interference with the desired workspace of the mechanism without any additional calculations being required.

Lastly, the development of a non-actuated prototype gave evidence to issues with the current design model, allowing changes to be made to the design of a fully actuated physical prototype. The prototype was then built and tested with the Labview control system and resulted in the desired motion path being achieved.

Chapter 8 Conclusions and Future Work

The main goal of this thesis was to propose a method for the rapid development of multi-platform parallel mechanisms. This aim was then broken down into three main research goals: to investigate whether virtual chains could be used to produce a more graphically based approach to the conceptualisation of a class of generalised parallel mechanisms; whether a generic cost and energy efficiency model could be developed for parallel mechanisms; and whether an alternative design approach could be evaluated and tested in order to address the original aim.

Additionally, another motivation for this work was to determine if it was possible to not only speed up the development time for designing a parallel mechanism by using only visual methods but also to determine whether the efficiency of a mechanism could be determined via a simulation.

This thesis begins with a detailed look at parallel mechanisms, the mathematics behind their development, conception in the way of type synthesis, and the various software packages used to devise them.

Then, the concept of type synthesis using virtual chains has been extended to Parallel Mechanisms with multiple platforms allowing the method to be applied to more mechanism architectures. The extension incorporates the virtual chain type synthesis methodology into the design of multi-platform parallel mechanisms.

This thesis also provides a new method of determining the workspace of parallel mechanism using a CAD based approach as well as the inverse kinematic, and inverse dynamic models of parallel mechanisms by utilising the MapleSim software, and a method of combining the inverse kinematic model developed in MapleSim with a Labview VI in order to produce a workable control system for parallel mechanisms.

Lastly, the methodologies laid out in this thesis have been tested using various case studies before being applied to the development of a new multi-platform mechanism concept design. The design was then converted into a full-scale physical prototype that was actuated using servomotors selected based on the results of the inverse dynamic model produced using the MapleSim method. The mechanism was tested by manoeuvring the platforms through a prepared path and was found to function suitably when a counter weight was applied to aid in levelling out the moving platform.

8.1 Main Contributions

The work presented in this thesis has shown that the design of a mechanism can be produced with a visual approach that, rather than relying upon traditional design processes for workspace and dynamic analysis, and the production of an inverse kinematic model, the production of these elements can be done with CAD and graphic mathematical modelling software. The main contributions of this thesis are as follows.

Type synthesis is a method of using screw theory to develop a parallel mechanism from the intended motion pattern produced by the desired product. The new method involves initially investigating the number of DOF in the system, number of kinematic chains, and

over constraints in order to produce a table of legs with a list of how many DOF would be available on each leg.

The novel expansion of this method allows a designer to include multiple moving platforms or end-effectors with relative DOF in the design process and answers the first research question. This allows for an already robust system of developing parallel mechanisms to allow the method to incorporate all future iterations of this system.

The traditional process for producing a workspace has always fallen short of being able to replicate the motion of the mechanisms kinematic chains and the individual links. This has led to the production of a viable workspace for any given mechanism to be reduced from a larger workspace as the designer must take into consideration any potential link interferences while the mechanism is operating.

The new and novel workspace analysis method presented in Chapter 4 of this thesis allows the user to set limitations to the desired manoeuvrability of certain joint angles and to include link interferences into the calculation of the mechanisms workspace. The designer, using this method, can get a more refined idea of what the mechanism they have designed would be capable of and where potential collisions can occur without having to produce a prototype beforehand. This means that a mechanisms workspace can be maximised to the point that the user can be sure that no link interference or singularities can occur or modify the mechanism to avoid such events happening without a waste in physical resources.

This work allowed the second objective of the thesis to be fully addressed as the novel method allowed for the workspace of several parallel mechanisms to be developed using virtual chains in a more visual-based approach. Additionally this work addressed aspects of the fourth research question directly by testing the method through a series of comparisons with previously published work ranging from simple 2D workspace slices, to a full 3D workspace image.

The theory used in producing an inverse kinematic model for a parallel mechanism relies heavily on the use of specialised mathematics in the form of screw theory. This means that for a developer to produce the inverse kinematic model for a mechanism, they must first be capable of fully understanding the mathematics.

The work presented in Chapters 5, 6 and 7 give a series of new process stage layouts in which a designer working on a mechanism can produce the concept design of a mechanism first using the type synthesis method detailed in Chapters 2 and 3, and then produce a CAD model of the mechanism. The design of the mechanism can then be converted into an inverse kinematic model using the object orientated mathematical modelling software MapleSim.

This work showed that the range of viable motion in a parallel mechanism's generated workspace could be used to develop the inverse kinematic model of the mechanism when combined with virtual chains, which addresses the first condition in the third objective. This work also goes towards addressing the fourth research question as the method is compared with the traditional process of a 3-RRR planar parallel mechanism as well as an RRR serial robot arm.

In addition to the inverse kinematic model, the inverse dynamic analysis of a parallel mechanism, similarly to the inverse kinematic model, requires the designer to have an understanding of complex mathematical formulae that look at the propagation of forces through a mechanism from the moving platform to the actuated joint in order to determine the force or torque required to manoeuvre the mechanism in a desired manner.

The inverse dynamic analysis extension to the inverse kinematic model detailed in Chapter 5 allows the designer to simply add a copy of the inverse kinematic model mechanism, remove the virtual chain and then control the mechanism using the positional information taken from the actuated joints of the inverse kinematic model. The MapleSim software approach also allows for a modification of components masses, centre of mass and joint lengths to allow the user to determine the most optimal arrangement of components in the mechanism.

Similarly, this work addresses the requirements of the third objective and fourth research question by showing that the inverse kinematic model produced in MapleSim can be used to develop a working inverse dynamic model and then comparing it to previously published work on a 3-RRR planar parallel mechanism.

The topic on the efficiency of parallel and serial mechanisms is heavily neglected, with only a few examples having been published. The new and novel work detailed in Chapter 5 gives a new method for quickly determining the efficiency of a mechanism based on the forces obtained from the MapleSim inverse dynamic model and then using a secondary virtual chain to determine the forces at the end-effector or moving platform of the mechanism.

Additionally, the comparison in Chapter 7 shows another method in which the various requirements of two similar mechanisms can be determined by again modifying the MapleSim inverse dynamic model of each mechanism. Lastly, the comparison in Chapter 7 showed that multi-platform mechanisms are more efficient when compared to a single platform variant when performing the same task in the same amount of time.

The novel work in this thesis on energy efficiency allows a designer to determine the mechanical energy efficiency of their mechanism without the need to fabricate a physical prototype. This work directly addresses the fourth objective and in doing so answers the third research question. This work also addresses the fourth research question as the testing of the energy efficiency of the proposed prototype detailed multiple work scenarios to determine whether the mechanism was more or less energy efficient than its single-platform variant.

The proposed MapleSim to Labview control system method detailed in Chapter 6, shows how a virtual chain generated inverse kinematic model can be applied to a control system. This methodology removes the need for a mathematical variant of the inverse kinematic model to be generated by the user prior to or during the build of a control system. The special servomotor Labview blocks used in the control system allowed the actuators of the mechanism to return current values on the number of rotations the motor had made from its zero position in order to act as an error value to the inverse kinematic model's expected position.

The MapleSim to Labview Control System is also an example of utilising the communication protocols between the cRio and hub computer in order to pass vital information regarding the position of the mechanism to ensure that the mechanism is where it is meant to be at, at all times.

This novel work addresses the last remaining component of the third objective as it details the procedure for modifying the inverse kinematic model produced in MapleSim to be compatible with the graphical programming software Labview where it is utilised in the production of a control system.

The resulting work detailed in Chapters 3-6, allowed the modification to the current design process to be formalised. This new process allows a designer to produce a new parallel mechanism without the need of advanced programming and mathematical skills.

The novel changes to the process also gives a graphical representation of the mechanism and its components throughout the development process while also allowing the user to modify components of the mechanism quickly and efficiently at each stage of development.

The work given in Chapter 7 further develops the proposed changes by adding a static analysis stage to the development process using the SolidWorks FEA add-on software as well as the optional stage of developing a non-actuated scaled model of the mechanism that allows the designer to get a more tactile impression of how the mechanism will operate once completed. This also allows the user to modify the design of the mechanism before resources are committed to the production of a fully actuated prototype.

From the work detailed in Chapter 7, the current proposed flow diagram for the modified design process can be updated to include the optional scaled down non-actuated model and FEA steps as shown in figure 8.1 where the red highlights identify contribution sections in figure 8.1. In figures 8.2, 8.3 and 8.4, the individual design processes are expanded with red highlights to identify novel design processes being implemented; yellow highlights to identify optional extra steps to stages; and green highlights to identify steps that have been moved to new stages to suit the new process.

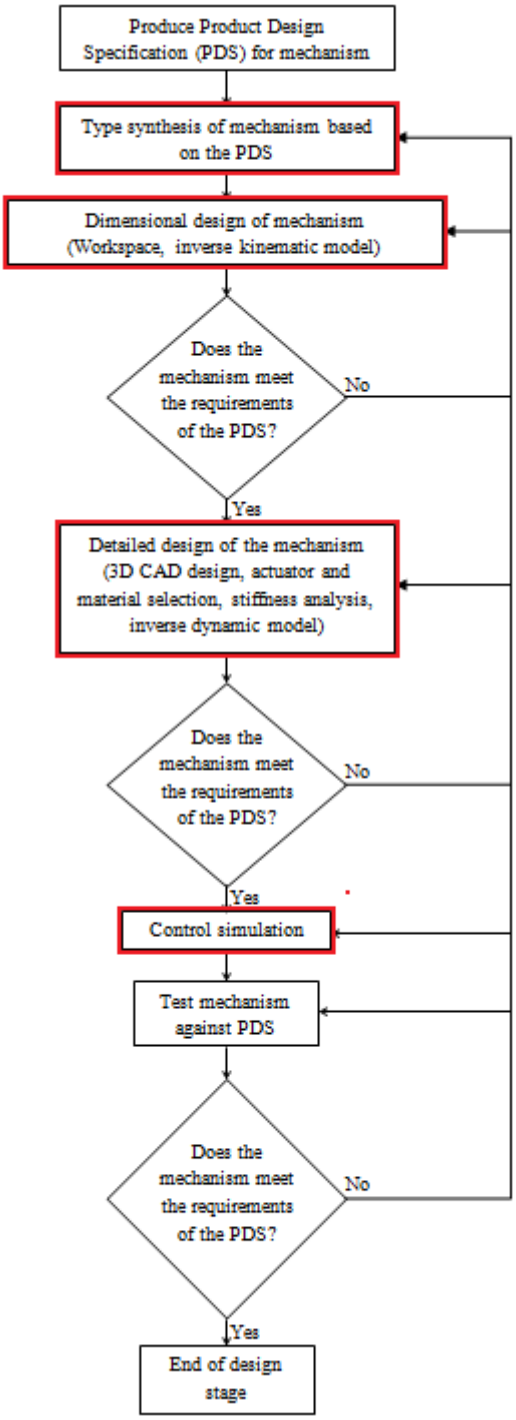

Figure 8.1: Highlighted flow diagram of the design process for multi-platform parallel mechanisms with highlighted stages where contributions have been made.

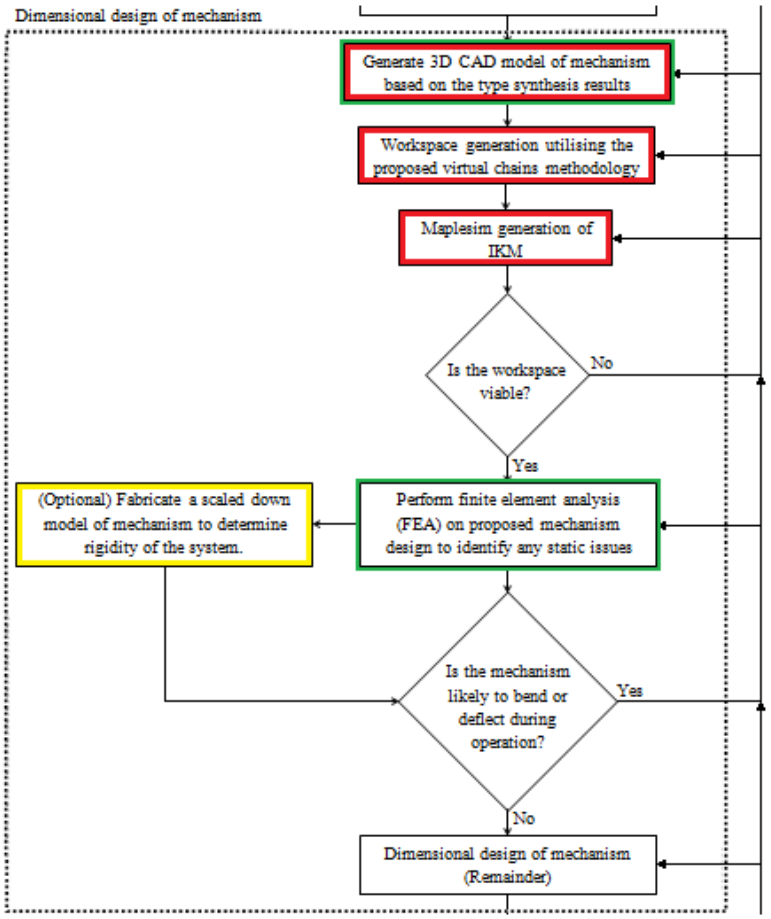

Figure 8.2: Highlighted flow diagram of the dimensional design stage

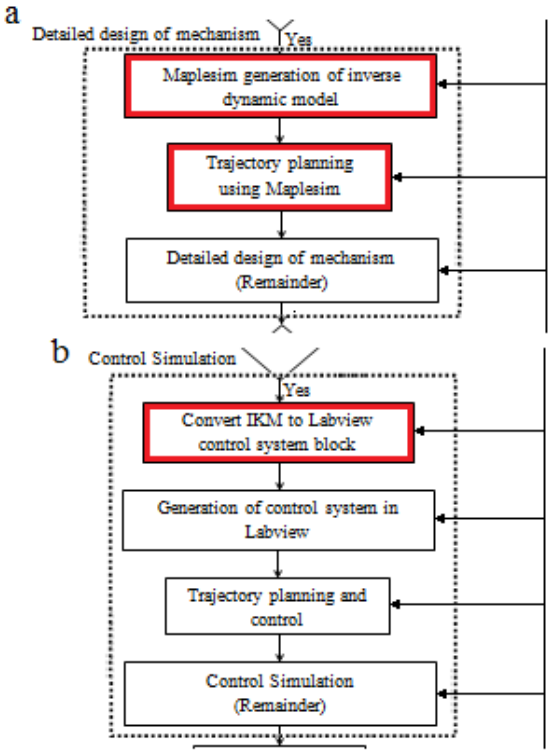

Figure 8.3: Highlighted flow diagram of the: a) Detailed design stage, and b) Control system stage.

From the development of the multi-platform parallel mechanism in Chapter 7, it can be seen that the proposed changes to the design process are validated since a full scale, fully actuated multi-platform parallel mechanism was developed from an initial concept design, in addition to a non-actuated scaled model which has been added to the design process, through to being fabricated in a short time frame. This therefore achieves the final objective of the thesis

The proposed changes to the process allows a designer to quickly and intuitively design and improve a new parallel mechanism with little to no programming knowledge and without the need of a background in advanced mathematical and engineering practices while having fully visualised representations of the mechanism and its component parts throughout the process. This therefore completes the fourth research question as this formalised novel changes to the design process have been tested at each stage through various case studies on numerous different parallel mechanisms.

From the completion of the objectives it can be summarised that: the second research question was addressed by the creation and formalisation of the modifications to the design process that allows a user to develop a new parallel mechanism using a graphical-based approach by utilising the concept of virtual chains in a CAD and graphical programming software environments, the third research question was addressed by the development of a method of determining the mechanical power required to manoeuvre a mechanism through a given path and then comparing the total power with that of the power received at the end-effector of the mechanism and by comparing the power between two separate mechanisms that are performing the same task, and the final research question was addressed throughout this thesis as each stage of the new design process was rigorously tested against published works in order to validate the methods as well as identify any issues the proposed method may have.

From the work detailed in this study, the aim of creating novel methods within the design process of a parallel mechanism that supports the rapid development of an energy efficient, cost effective parallel mechanism has been achieved due to the four research questions being answered by the work developed throughout the chapters of this study. The finalised design process proposed in figure 8.1 is the accumulation of the work presented in this study in which, a novel method for determining the energy efficiency of a mechanism's mechanical structure is proposed.

An additional contribution relates to the novel method of using virtual chains to determine the workspace of a parallel mechanism using CAD software and the development of inverse kinematic and inverse dynamic models using MapleSim as well as the application of an inverse kinematic model to produce a novel method for creating a control system using the MapleSim to Labview connector add-on and Labview allow for a more intuitive graphical-based design process.

8.2 Limitations of Work

The primary limitation for the impact of this work is that the system has only been run through completely for a single multi-platform parallel mechanism with 1-rDOF, while the individual stages have been put through numerous comparisons of individual mechanisms, more applications for testing should be done in order to continue to refine the proposed changes to the design process.

The approach detailed in chapter 4 for the development of a mechanisms workspace is not able to account for static singularities in its current form as the current method assumes that the mechanism is statically stable.

Additionally the work completed in Chapter 7 showed that the non-actuated model failed to remain rigid, while this led to the inclusion of new steps in the design process, further development is required to ensure that the mechanism will remain rigid once constructed.

Another limitation of the work is that the energy efficiency study of the mechanism is only a theoretical study and requires further development to allow the study to be refined to a greater level of detail.

Lastly, the energy efficiency of multi-platform mechanisms needs to be improved upon in order to allow for the total system energy of the mechanism to be determined which incorporates when only one platform is in motion, both platforms are moving together and both platforms are moving apart.

These limitations are known at this point, which allows the required work to be investigated in future projects and studies.

8.3 Future Research

With the physical mechanism accurately being controlled by a virtual chain included within the inverse kinematic MapleSim model, the following future work are proposed in order to improve the work presented in this thesis.

8.3.1 Improvements to the design of the mechanism

From the results of the physical prototype, an additional non-actuated kinematic chain aligned along the vertical axis should be included on the moving platform with only horizontally aligned kinematic chains in order to improve the stability of the mechanism and to reduce the bending of the moving platforms.

8.3.2 A more robust MapleSim model

The current MapleSim models being used utilise the standard graphical designs of its components. This can be improved upon by including attachments of CAD designed versions of the kinematic chains components and the moving platforms. This would allow the user to see how the intended movements of the mechanism would affect the mechanism similarly to the CAD based workspace approach detailed in Chapter 4. Additionally, the MapleSim model can include further frictional forces to determine how the joints motion affects the systems force requirements. This would also allow for the energy efficiency of the mechanism to be calculated.

8.3.3 Control system positional errors

In order to make control systems more robust, they traditionally have a direct or forward kinematic model taking inputs from positional sensors on the mechanisms joints in order to identify errors that have arisen from motor and joint tolerances. To produce this kind of model, a MapleSim model could be produced that would be produced in a similar method to the secondary model in the IKM and IDM models with the only exception being that the non-actuated joints are driven by inputs from sensors mounted to the physical model. A virtual chain can then be attached to the end-effector and be used to identify the exact position of the mechanism in its current form. The control system

would then be able to use this data, compare it to that of the desired position, and make adjustments based on the resulting information.

8.3.4 Energy efficiency study

The current efficiency study method determines the efficiency of the system based on the current motion of the mechanism only. To improve upon this, the method should be expanded upon to allow for the determination of the total system efficiency based on the various motions that the mechanism is capable of achieving.

Additionally the electronic efficiency of the electronic components of the mechanism can be simulated in a similar way to the mechanical efficiency, allowing for a full system efficiency tests to be carried out on a mechanism before the mechanism reaches the prototype production stage. This could potentially save the designer time and money in the event that the system is not able to perform to the degree expected of it due to electronic power restraints.

8.3.5 Incorporate a vision system to take over the user interface

With the completion of the user controlled control system, the next step would be to develop and incorporate a vision system into the control system that would replace the user-inputted locations.

This can be done by placing the mechanism above a conveyor belt to simulate a pick-and-place environment and then adding a camera ahead of the mechanism. The camera would then feed into a LabVIEW VI that would identify components needing to be picked up and then assigning one of the end-effectors to collect the specified object while assigning the other end-effector to collect the next object on the conveyor belt.

The vision system would allow the mechanism to be operated at the speed required for a pick-and-place mechanism and would allow the theory that multiple platform mechanisms are more efficient than their single platform counterparts are.

8.3.6 Inverse kinematic and dynamic models for mechanisms with multiple actuation modes

The concept detailed in Chapter 5 can be expanded upon to allow for the production of an IKM and IDM for parallel mechanisms with multiple actuation modes by duplicating the first model several times, or one for each actuation method, while each new model would have a different actuation mode's virtual chain. Then through the process of either a series of Boolean values or a single integer value applied by the user that interrupts the signals from all but one IKM, the secondary mechanism model can be controlled by the IKM with the required actuation mode virtual chain producing results for the IKM and IDM of each actuation modes.

This method could then be expanded upon to produce a full control system for the mechanism where the various IKM MapleSim models are loaded into the control system and then the desired actuation mode can be once again selected by the user.

8.3.7. Singular configurations identified in the workspace

In Chapter 4, the workspace of a parallel mechanism is determined by the limited joint angle mates applied by the user to prevent obvious configuration singularities. The simulation is then run in order to produce the external area of the workspace. This method

therefore does not necessarily identify additional singular configurations within the defined are of the mechanisms workspace, meaning that over-mobility or under-mobility singularities could exist within the specified workspace. Therefore, more work is required into the identification of any singular configurations present and how they can be eliminated from the mechanisms viable workspace within the CAD software. Finally, the IKM and IDM can be modified to remove any identified singular configurations by limiting certain joint angles on the first mechanism in each model.