Modelling and Control System of a Six Degree-of-freedom Robotic Manipulator for Shipbuilding Welding

Hongkai Hu
20319007
Hongkai.Hu@nottingham.edu.cn
Supervisor: Dr Adam Rushworth
Adam.Rushworth@nottingham.edu.cn

Abstract

This thesis described the control system design of a six-degree-of-freedom robotic manipulator with welding tools on a hexapod walking platform. To begin with, past studies on control systems, hexapod walking robots, welding techniques, and the current type of mobile welding robots are reviewed. Then, the paper explained several key elements and features of a control system design, including DH Parameter, Jacobian, Euler-Lagrange equation, and adaptive control. An elementary control system is designed for a specific design of robotic manipulator. The designed system is then taken into simulation software (ADAMS and MATLAB) for feasibility testing. A PD control system is added when simulation is carried out in MATLAB to find out reliability of the system under unstable environment. Finally, a conclusion is drawn from the result that this system is working but the stability of end effector can still be improved through application of higher level of control system like PID controller.

Contents

1 Introduction

The shipbuilding industry is an up-and-coming field of robotics application. Since the shipbuilding industry remains one of the labor-intensive industries, the shipbuilding companies naturally tend to invest heavily into robotic automation techniques for the improvement of efficiency. [1] One of the many problems in this industry is the welding procedure for large double hull ship blocks. [2] The unique and complicated working environment inside the block poses some special requirements for the welding procedure. For example, because the welding procedure needs to be carried out inside the block, the heat produced during the welding process cannot be released quickly. This causes the temperature of the working environment to rise rapidly. Also, there usually is little light in the blocks, which will make it extremely difficult to find welding tracks and identify the welding quality. These are just a few among the list of complicated conditions for shipbuilding welding. In order to deal with the problems, multiple methods or designs of the robots were raised. To begin with, several types of welding systems consisted of a welding robotic arm and a fixed orbit inbuilt inside the blocks. These types still require human operation, and the application of robotic manipulators is just for improving the welding efficiency and welding quality. Later, P. Gonzalez de Santos et al. designed the ROWER, which is an automatic controlled programmable walking robot with a SCARA type arm mounted on the platform for welding [3]. Also, the Hitachi-Zosen shipyard in Japan developed a type called NC painting robot. With modification on the end-effector, this robot can also be applied to the welding process.[4] Another type is the rail runner mechanism, RRX3, developed by a group of researchers in Seoul National University.[5] Among all the designs mentioned above, ROWER bears further potential since it requires almost zero human interfaces during the welding procedure. The problem with this type is that it is too big in size and the robotic arm was SCARA, which means that the movement of the end effector (In this case, the welding torch) has a bigger range in the horizontal direction than in vertical range. It can be concluded that ROWER is on the right track of design, but there remained improvement. The control could be further modified specific for welding procedure, the robotic arm could be replaced by other types for more degrees of freedom, also, there was an increasing trend to add computer vision on

the welding tool to monitor the seam and welding quality. This thesis will be focus on the control of the robotic manipulator. So far there is already some successful designs on the mobile welding robots, but it still has space for improvement. The size of those robots mentioned above is rather big and it still requires human assistance, some of the types still needs a railway for direction. There is yet a fully automatic mobile welding robot for shipbuilding.

There will be eight sections in total. Section 1 is the introduction. Section 2 is the literature review. Section 3 states the aim and objective of the project. Section 4 describes the methodology applied. Section 5 presents fundamental theories, including the DH method, calculation of Jacobian, and dynamics of the robotic manipulators for building the controller. The mathematical model of the applied robotic arm will be established, and kinematic problems will be raised and solved. Section 6 will be the simulation and results. The established mathematic model will be put into the MATLAB and ADAMS for simulation and motion study. The aim of the simulation is to verify the designed control system for the specific design and configuration of the robotic manipulator. Section 7 is the discussion and future work, in which the unsolved problem will be drawn, and further study will be discussed for improvement. The last section is the conclusion.

In this thesis, the focus will be put on the control of the robot. To be more precise, the coordinate movement of the platform and manipulator and the elimination of error of the end effector movement.

1.1 Project Aims and Objectives

The aim of the project is to establish the model of the manipulator and design a control system for it to be functional under a certain degree of interference like the vibration from the joint motors. The following objectives are set for achieving the aim of the project: First, establish the physical and mathematical model of the robotic arm. Second, run simulation in ADMAS to determine the kinematic parameters of the arm motion. Third, establish a control system for the arm with determined kinematic parameters, compare the stability of end effector before and after the control system is applied. Consider the required stability for the end effector, the joint angle velocity for end effector needs to be within 0.5 rad/s.

2 Literature review

This project involves the specific application of robotic control theory and robotic modelling. For a mobile welding robot, there are multiple related areas that require study. Thus, this review is going to be divided into the following parts: the first part is about the industrial robots for the welding equipment. The next part is the control theory of the robot system. All the literature in this part will be about control theories, including D-H parameter, calculation of Jacobian and adaptive control. The third part will be about mobile platforms, like hexagon platforms, hybrid leg-wheel robots and robot walking gait. The fourth part is the welding technique. Papers about GTAW and SMAW will be reviewed. Also, there will be articles regarding the welding quality control by application of computer vision. The final section is about the current type of welding robots.

2.1 Industrial robots

The field of the industrial robot has been developed for decades, during which various types of robots were applied in manufacture and construction sites. According to the definition established by International Organization for Standard, the industrial robot is an automatically controlled, reprogrammable multipurpose manipulator programmable in three or more axes.[6] Based on different mechanical structures, the industrial robot can be divided into five types: Cartesian robot; SCARA robot; Articulated robot; Parallel robot and Cylindrical robot. Each type has its own feature.

2.1.1 Cartesian robot

The Cartesian robot is a robot that has prismatic joints on each axial that is built along the Cartesian coordinate system. The advantage of this type is that usually, it has higher accuracy and, depending on the mechanical structure, higher load-bearing[7]. The short come is also rather obvious, the occupied space and workspace are positively related. To get a bigger workspace, a higher occupied space is inevitable. This is why most Cartesian

robots can be applied when heavy load or precise operation is involved.

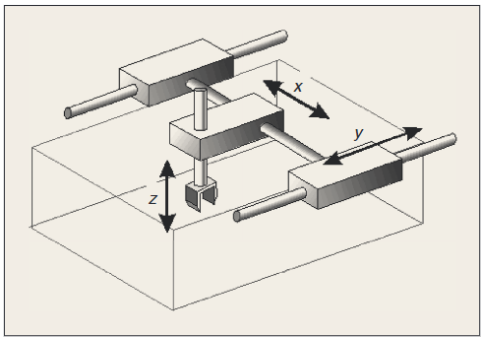

Figure 2 1 Cartesian Robot [8]

2.1.2 SCARA robot

A SCARA robot is a type that has multiple parallel rotational joints. Due to the multiple parallel rotational joints, the SCARA robot provides extremely high speed for movements from one point to another[7]. But due to the parallel joints, the workspace for this type is limited to one plane. SCARA robots are mostly applied on the assembly line where little vertical movement is required but rapid and repeatedly horizontal movement is needed.

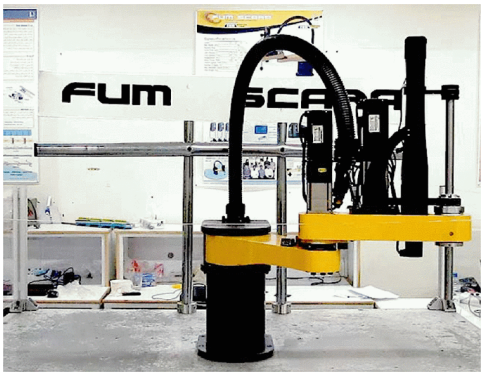

Figure 2 2 ACARA robot[9]

2.1.3 Articulated robot

This kind of robot has at least three rotary joints, extra joints lead to additional DOF (Degree of Freedom). The articulated robot is the most wildly used type in factories and laboratories. It is famous for its flexibility and a rather big workspace. The working range is a sphere with the radius of the length of the arm. Multiple joints enable the end effector to reach every point within its working range[10].

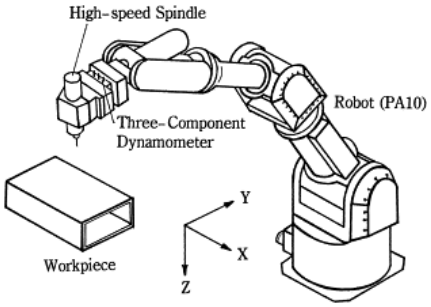

Figure 2 3 Articulated robot

2.1.4 Parallel robot

The Parallel robot has a unique structure which consists of multiple concurrent prismatic or rotary joints. It is also known as the Stewart Platform. With the kinematic chains, it can provide with high stiffness of the end effector in the workspace[11]. The weak spot is that usually, one parallel robot consists of six or more mixed joints, which makes the dynamic model of the robot rather complicated and thus adds difficulty to the control.

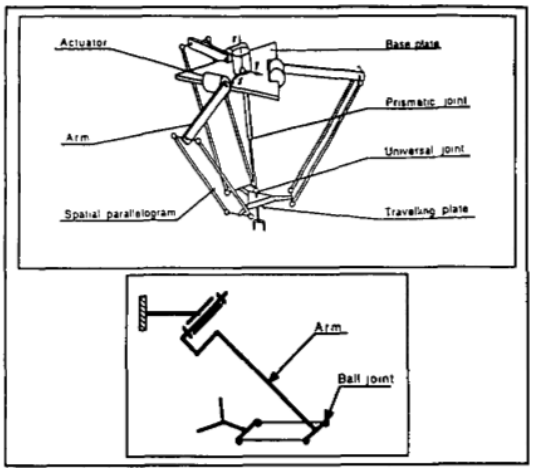

Figure 2 4 Delta Robot[11]

2.1.5 Cylindrical robot

The cylindrical robot is a unique type of industrial robot. The axis of the cylindrical robot forms a cylindrical coordinate system instead of a normal cartesian coordinates system.[12] So, the workspace of a cylindrical robot is limited to a hollow cylindrical column. This type is usually applied on assembly systems or combined with the SCARA type to increase its flexibility in the z-axis.

2.2 Control theory

2.2.1 DH Parameter

The control of the robotic system is a huge area. In this project, the focus is on industry robotic arm control. The first key feature for robot arm controlling is the DH Parameter.[13] DH Parameter is the most wildly used method for robotic modelling due to its high precision and clear physical meaning.[14] But the DH Parameter has its own limitation. It was pointed out by Gao's research that 95% of the movement inaccuracy comes from the inaccurate diameter of the arm. That is to say, the error was produced even before the movement actually started. However, this also means that most errors can be eliminated by simply change the diameters of the arm's model. The team thus developed a structure parameter identification method to be applied on the controller of the manipulator. It works under the calibration method which detects the uncertain parameters including manufacture errors, environment temperature changes and material deformations and then modify the model of the manipulator in the controller to achieve a higher accuracy and less error in actual movement. After the testing, the new method can reduce the uncertainty for over 95%, which can be concluded as a big success.

2.2.2 Forward kinematics and Inverse kinematics

Identifying the DH Parameter of the arm is the first step. The next is to apply inverse kinematics to solve for the workspace and gesture of the arm. A research conducted provides an example of the application of the inverse kinematics.[15] They first established the DH Parameter of one particular arm. Then the homogenous transform matrix of the end effector is obtained. The third step is to calculate the closed-loop solution of inverse kinematics, that is to say, instead of using angular variables to present the coordinate of the end effector, each joint angel will be presented by the coordinate variables of the end effector. The final step is to verify the solution by applying an actual set of joint angles. This article provides a standard procedure for solving inverse kinematic problems.

2.2.2.1 Jacobian

Another key feature of robotic modelling is the Jacobian. It describes the velocity factor of the arm. The calculation of Jacobian can be rather complicated, and this is why David E. Orin and William W. Schrader discussed and compared six different methods to calculate the Jacobian.[16] These six methods were developed by different researchers during different times. After these methods are introduced, they are taken into a robotic controller for comparing the calculation time for the controller. The result shows the method developed by Renaud, M. (1981) has the highest efficiency. In this method, the manipulator is divided into two halves, and the Jacobian for half of the manipulator is calculated twice and combined to form the final result.[17]
While the Jacobian describes the velocity factors of the system, it is also necessary to determine the forces and torques for joints because they decide the output of the motors, and the output of the motors are the actual element human can control through controller panels. In Roy Featherstone and David Orin's research, they concluded the major achievement in the field of robotic dynamics.[18] The paper provides a comprehensive description of this area, starting from the fundamental work. According to the team, the classic way is to use the Euler-Lagrange equation for the equation of motion. It is straight forward but the computation efficiency is low, thus in more cases, the Newton-Euler method is applied. Instead of calculating the system energy as a whole in the Euler-Lagrange equation, the Newton-Euler method calculates the dynamic status of each joint, and with forward-backwards recursion, the general description is achieved. Several other algorithms are also introduced for improving efficiency. The dynamics of the robotic system are used to establish the controller of the system, and in this case, an adaptive controller.

2.2.2.2 Adaptive control

Adaptive control is one of the most commonly used methods for the robotic system when there is unpredictable interference involved. Dr K. J. Åström provided a detailed introduction to adaptive control in his article.[19] In it, he gave the definition of adaptive control as "a special type of nonlinear control system

which can alter its parameter to adapt to a changing environment.” In the article, he also introduced several types of adaptive control, including gain scheduling, Model Reference Adaptive System (MRAS), self-tuning regulators, and stochastic adaptive control. There is also an attempt to combine the adaptive control with the Jacobian. C. C. Cheah et al. [20] introduced an approximate Jacobian adaptive control for robotic manipulators. The theory is by applying parameter update law and update the uncertain kinematic and dynamic parameters online to the robot controller and thus ensure the end effector converge to the desired trajectory. The team also introduced the Jacobian adaptive control with a passive mechanism. In addition to operate under uncertain kinematics and dynamics, this type of control can also operate when the end effector is under a counter force in situations like opening doors or screwing a bolt. The controller designed is then taken into a 2-link direct drive robot for testing. With the testing figures shown, the controller is effective, the position error of the end effector is varying around 0 in a small range (within ±0.02).
The general procedure of designing the control system for a robotic manipulator is introduced in Jamshed Iqbal et al. research.[21] The first step shown in their paper is to establish the forward kinematic model of the manipulator, including the DH Parameter and the Jacobian. The next step is to solve the inverse kinematic problem and validate the close-loop solution. This step lays the foundation of what’s coming next, which is a workspace analysis. The analytical result determines the working range of the manipulator, which is important for the actual application of the controller. Eventually, the controller will be taken into a real manipulator for testing and the result is judged from the speed, accuracy and working range.

2.3 Welding technique.

As this project is about a welding robot, the welding technique in use is certainly a vital aspect that requires research. There are various types of welding for different situations today, two of them are GTAW and SMAW. In the following part, these two will be reviewed for their features.

2.3.1 SMAW

Shield Metal Arc Welding (SMAW) is one of the most wildly

applied welding techniques. It is also pointed out in Ibrahim Alkahla and Salman Pervaiz's paper that SMAW is the most commonly used welding technique in shipbuilding.[22] The paper provides a full introduction to this welding technique and the sustainability study as well. It started with the description of SMAW, illustrated with a flow chart to show the procedure. A power supply is connected to both the welding surface and the welding head, the theory is to melt the metal on the welding head and add it between the parts thus they would come together after the melted metal is cooled again. The chart also includes the input and output of the procedure. Then, the sustainability research is conducted in several different aspects. The triple bottom line method is used to help verify the sustainability of SMAW. The analysis is conducted from the economic, social and environmental aspects to determine whether SMAW is a sustainable technique. It is concluded that SMAW at the current stage still has a lot of space for sustainability. The cost can be further reduced with optimizing methods or employ automatic systems, the hazards fume can be eliminated by optimizing the working procedure and the energy consumption can also be reduced by using a modern power supply.

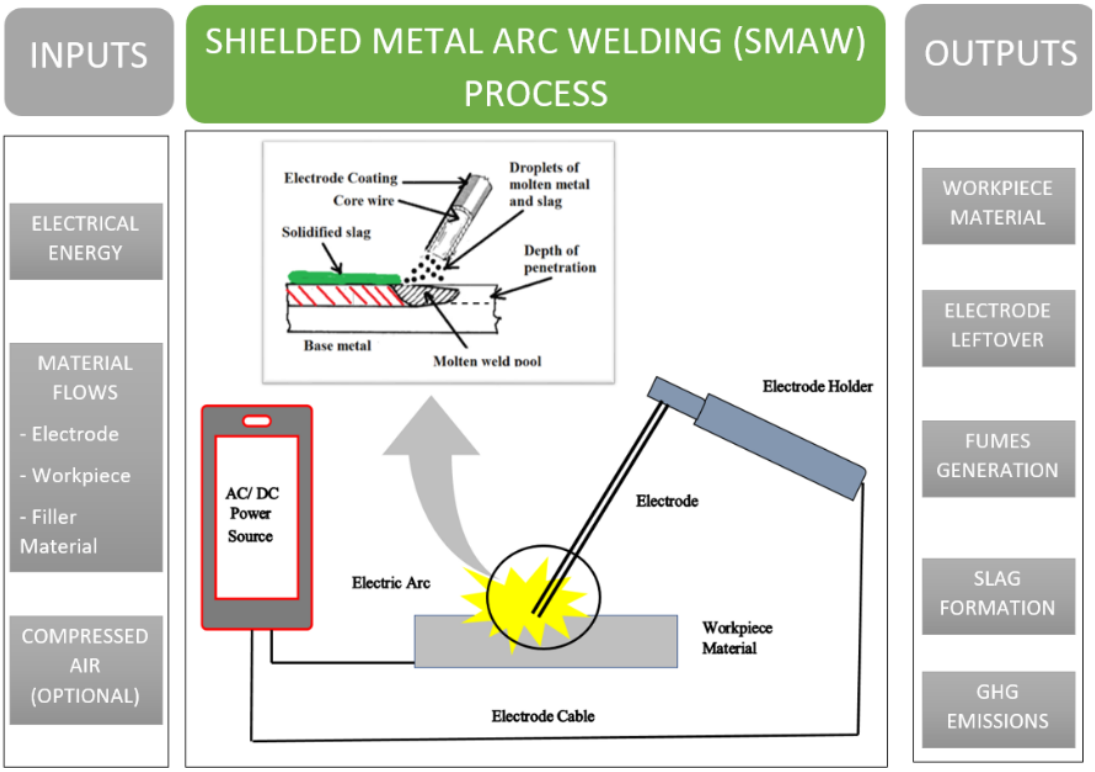

Figure 2 5 SMAW process[22]

2.3.2 GTAW

Gas Tungsten Arc Welding is another option for the welding robot. It also involves melting metal with an electric arc. The GTAW uses a non-consumable tungsten electrode to produce weld. Inert gas is applied to prevent contamination or moist. A constant-current welding power supply is used to produce the welding arc. There are lots of research done on different aspects of this welding technique. There was research conducted by R. W. Niles and C. E. Jackson on the thermal efficiency of welding with different protect gas.[23] Also, when it comes to the combination of modern digitalized technology and welding, GTAW became more popular, especially when computer vision was involved. GTAW usually comes up with a cleaner and more uniformed weld, which makes the identification of seam and welding pool much easier for computers. A welding robot system embedded with seam tracking and weld pool control was designed by Hong-yuan Shen et al. [24]. Fenggui Lu . et al. conducted research on modelling and finite element analysis on GTAW and weld pool.[25] In the paper, the integral mathematical model of fluid flow and heat transfer of welding arc and weld pool is established. First, the mathematical model o the GTAW process is calculated. The equation involved includes the electromagnetic equation, the continuity equation, the momentum equation and the energy consumption equation. A range of boundaries is also set up for the coming steps. The next step is the finite element analysis, where the model of welding procedure and weld pool is taken into ANSYS for simulation and finally the experiment is carried out and according to the result, the established mathematic model can prevent assumption which surface temperature of weld pool is constant, and the boundary conditions set are reliable.

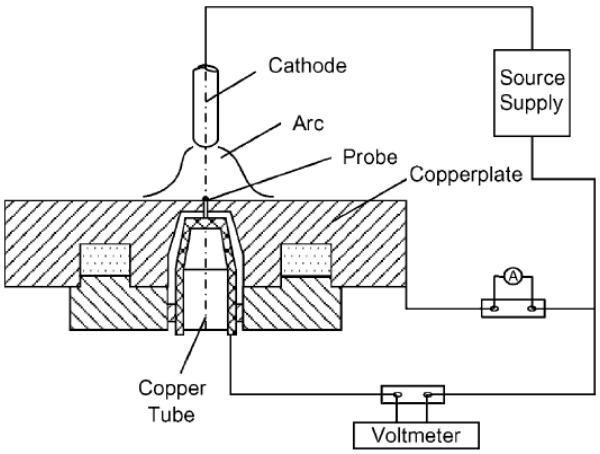

Figure 2 6 GTAW with current density measurement used in the research[25]

More research in the welding area focuses on improving the welding quality and accuracy. Those projects are usually conducted on welding robots. For example, Yanling Xu . et al. developed a real-time seam tracking control technology based on a passive vision sensor.[24] In this article, Yanling Xu and the team designed a system that does not require the robot to be taught in advance before it carries out the welding process. Unlike most of the types in practice at that time, the welding robot loaded with this system can self-rectify deviations in the movement during welding. The paper presented the vision sensor system applied and how the image from the sensor is processed. It also explained how the improvement of precision of image processing could raise the welding quality. A flow chart is shown to illustrate the process procedure of the whole system. In the end, the testing and results are shown, and it is concluded that the designed system does improve the welding quality. Another team lead by Hong-yuan Shen also carried out another research in a similar area. They came up with a different system equipped with passive visuals to track the seam and monitor the welding quality. It is hard to compare the two systems above since both of them demonstrated an increase in welding quality after the system was applied.

2.4 Mobile platform

The welding hand in this project is required to be placed on a mobile platform. There are several options for the robotic welding system to be placed on: wheel-driven platform; legged walking platform; hybrid platform, or drones. The drone was

ruled out first for two reasons. One is that the narrow and complicated inner structure of the ship block will make it extremely difficult for the drone to move and maintain balance. For those types which are capable of shuffling through the blocks, it is even harder to maintain stability to carry out the welding process. Another reason is that the welding equipment required to weld the ship blocks mostly comes in two parts: welding torch and power supply. This leads to multiple hard-wired connections between the welding tool and power supply. It will be too heavy for the drone to carry the whole system on the platform, and the wire connection makes it difficult to control the balance of the platform and limits the working range. So, the drone is making the design of the robotic arm unnecessary complicated, and thusly ruled out in the first place.

2.4.1 Wheeled platform

There are several types of wheeled platforms judging by the number of wheels and type of wheels. First, there is the traditional four-wheel vehicle used by most robot systems. It is easy to control and high in speed. It can be applied to most situations where the terrain is not changing rapidly, like indoor areas, grass, or gravel area. The suspension system of the vehicle can handle most of the vibration when traveling.[26] The problem with wheeled robots is that when it comes to complicated terrains, they cannot be applied, such as stairs, continuous slopes, or obstacles higher than the chassis. Other than four-wheel platforms, there are also three-wheel omnidirectional platforms, two-wheel balance platforms, or six-wheel mars rowers. These platforms are mutations of the traditional platform and can be applied to specific cases. However, consider that the inside of the ship blocks is usually filled with doorsills every several meters, a wheeled platform is not the best choice for shipbuilding work.

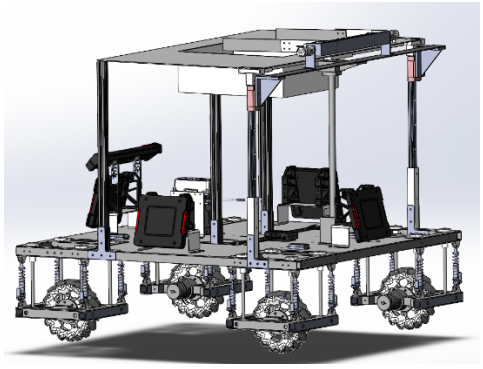

Figure 2 7 Wheeled platform

2.4.2 Legged platform

This kind of platform is known as a "walking robot". Depending on the shape of the platform and degree of freedom required, the number of legs can vary from four to six, in some special cases, even more. There used to be a minimum number of legs, and it was four. This was because it takes a minimum of 3 legs to maintain the platform to be stable while the fourth leg moves to make the platform move forward. However, with the development of biological and robotics, a walking mechanism with 2 legs named bipedal walking robots was developed. This type is inspired by the walking gesture of human beings. The biggest advantage of this type is that it fits all kinds of terrains. Other than usual situations like grass and gravel, places like stairs and soft soils can also be covered by walking mechanisms. There have been various research done on the walking robot, especially the hexapod-shaped ones, many companies have developed their own model.[27] In the 90s, there was Ambler[28], ASV[29] and TUM[30], and in the first decade of the Millennium, there was Biobot[31], Hamlet[32] and Lauron series[33]. For the recent decade, there has been Aqua[34], Comet-IV[35], Mantis[36], and the famous Big Dog, or recent model, Spot, developed by Boston Dynamics[37]. The control theory and the gait of the robot were developed rapidly at the same time. Before any control system is designed for the robot, a mathematic model is to be established first, and kinematic problems are to be solved. One of the frequently used control system designs is called hierarchy control system. This system divides the control tasks into different hierarchies or layers. When lower layers control focus on the actuators and sensors, higher layers decide more general concepts like speed or gesture of the robot.

2.4.3 Hybrid platform

While the walking robot was developing rapidly, a new concept of hybrid mechanism was raised. It is a joined structure of legs and wheels. While the upper structure is a joint-link structure, a wheel is attached to the end. This type of structure provides the platform with great flexibility in all kinds of terrains and increases the efficiency of movement. When the platform is moving on rather flat ground like indoor areas or tarmac roads, the wheels are used for faster speed and easier control. When

the terrain becomes more complicated or fragile, the walking mechanism will be activated, and the wheels will be locked and work as the foot. The combination of wheel and leg provides more possibilities in the design of the legs. For example, the legs do not have to be uniform in configuration. There is one type designed by H. Adachi and N. Koyachi et al. [38] that has four wheel-legs. The legs were divided into the "front pair" and "rear pair", the front pair and rear pair have different configurations in coupe with different scenarios. When it comes to obstacles like steps, the front pair with three degree-of-freedom comes into effect, and when on a plain surface, the rear pair comes into effect for higher speed. There are also researches for the hybrid platform to move on uneven terrains. Yanjie Li conducted a study to simulate a leg-wheel robot to move on uneven terrain.[39] In his research, a specific configuration of hybrid robot is used for ADAMS simulation to determine the position and acceleration of the mass center and each leg of the platform. The results from ADAMS can be used as the foundation to develop the control system of the hybrid platform.

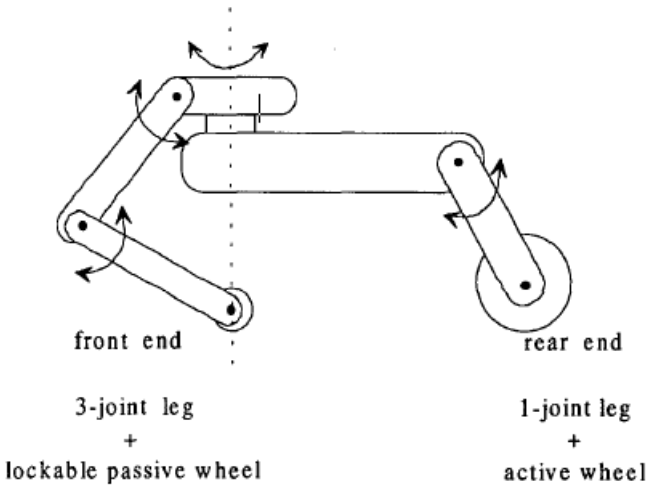

Figure 2 8 Leg-wheel robot[38]

2.4.4 Hexapod walking robot

Generally, consider the need for a welding robot and the working environment inside the ship blocks, the hexapod walking robot was chosen to be the platform of the welding manipulator. It can overcome the stiffener inside the ship easily and since in the welding procedure, accuracy and persistency outweighs the speed, wheels are not necessary.

The hexapod walking platform already has a lot of variations that have been discussed, which has been concluded in Franco Tedeschi and Giuseppe Carbone's paper[27]. In this thesis, various types of hexapod walking robots are presented and discussed. The paper started with an overview, including the earlier designs of the robot and the recent development, illustrated with a number of figures and a table demonstrating several characteristics and performance of the types mentioned. Next, the group listed the performance indices of the robot, which are concluded from the table in the previous section. These indices quantify the performance of the robot. Then, the paper raises the questions on the design considerations. Since the hexapod robot can come out in carious designs and each design has its own configuration, a list of criteria is built on key features of the robot to determine whether the design fits the requirements or is feasible. A flow chart is illustrated to show the design procedure of the hexapod walking robot. Each step in the flow chart is then discussed separately, like leg configurations, control schemes, etc. It can be seen from the paper that hexapod walking robot come in all kinds of shapes, in this research, however, only the control and the gait of the platform is focused.

To understand the control and gait of a hexapod platform, the modeling of a platform can be a start point. J.P. Barreto et al. provide a detailed method of establishing the kinematic and dynamic model of a six-leg robot by applying the free body diagram method[40]. The core theory is to apply dynamic equations of the isolated rigid body for overcoming the difficulties in dynamic modeling of the legged robots. First, a mathematical model of the robot is established. The system consists of six legs and a central body with 24 degree-of-freedom in total. The next step is to calculate the kinematic equations. The transfer matrix of a single leg is calculated first, then expanded into the matrix of a central symmetric structure with two legs and the central body. This set is then considered as a 2D body and in the following section, its dynamic equation is obtained by free body diagram method. The traditional method for pursuing dynamic equations is the Lagrange approach or Newton-Euler method. In the situation of the legged robot, the independent variables are too much thus neither of the methods can be applied without massive complex calculation. The free body diagram approach can calculate the equations for a 2D structure first and expand it to a 3D structure, thus reduces the amount of calculation. The calculated

equations are then taken into a simulation to verify the reliability and the result shows that the equation obtained through the FBD method is reliable.

After the dynamic equation of the walking platform is settled, the control is the next question to be concerned about. Enric Celaya and Josep M. Porta conducted research on the control of a six-legged robot walking on abrupt terrain.[41] In the paper, the research team presented a control hierarchy structure. There are six levels in total: Hold, Balance, Adaptation, Force Compliance, Walk and Drive level, with the input as sensors and output as motors. The Hold level is the lowest level in this system, controlling the motors directly. The function of this level is to keep the leg at the last command position. The second level, Balance level, is to keep a correct body attitude and stance while moving. Five different behaviors are devised to control different aspects of the body. The third level is the Adaption level. It changes the targets set by the balance level in more conflictive situations for the body to reach a more reasonable stance. Three different behaviors are developed in this level: Advance adjustment, Height adjustment, and Attitude adjustment, each adjusting a different aspect of the body. The fourth level is the Force compliance level. This level ensures that all the feet of the robot are stepping on the ground to support the body. The loading can then be evenly distributed to each leg. The fifth level is the Walk level. As shown in the name, this level controls the general movement of the robot, trying to make it advance and avoid obstacles. The last level, the Drive level, controls the stroke of legs to avoid obstacles. The testing is then carried out. In general, the behavior is good, but an unintended effect emerged is observed due to small tolerance in the balance level.

For hexapod walking robots, it is universally accepted that a hierarchy-structure control system needs to be applied. The following presents a complete design of a hexapod walking robot conducted by H.-J. Weidemann, F. Pfeiffer and J. Eltze.[30] The type is named TUM walking robot. It is a walking stick insect-inspired robot, thus the kinematic of the robot is studied through the leg of insects, so does the leg control. The controller of the legs simulates the neuron activity in the insect that activates and deactivates the muscles to drive the motors of the legs. A layered controller is designed for this robot, which consists of three layers: Leg coordination, Single leg control, and Swing/stance control. Six individual leg controllers are applied, and during the movement, they communicate with

each other to reach the best control scheme. The onboard electronics and joint design are demonstrated in the rest of the part.

2.5 Current studies on mobile welding robots.

There already exist several studies on the mobile welding robot, some of them already have a prototype and have been commercialized for quite some time till now. One of the examples is the ROWER developed by P. Gonzalez et al. [3]. In their paper, a walking welding platform designed specifically for shipbuilding is introduced. The whole system consists of a commercial welding system mounted to a commercial manipulator, and the manipulator is attached to a walking platform. The platform is specifically designed to move in a complex environment, which is between the double hull of a ship. As shown from the blueprint and the pictures of the prototype, the whole platform completely fulfills the space in the hull between the upper and lower stiffeners. The legs used on the platform have three degree-of-freedom, consist of two rotational joints and a translational joint, so it's a SCARA type of leg. The vertical link attached to the end of the leg has two grasping devices on two ends so during movement, it can grasp the two layers of stiffeners to maintain maximum stabilization. In the field test, the ROWER provides a third of less work time to finish the same amount of welding operation than human workers, and the welding quality still proves to be good. There is one problem for this type to be solved, which is the volume. The large size of the platform limits the access for this robot to some narrow places or sharp turns.

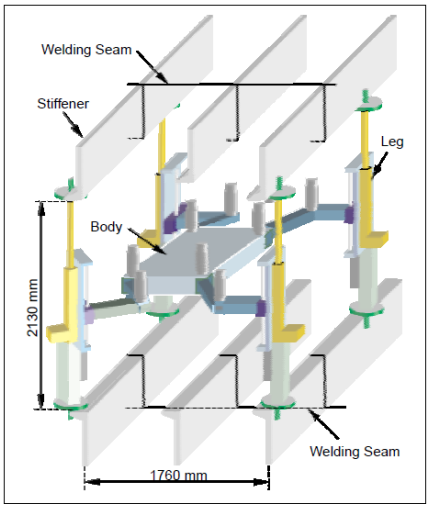

Figure 2 9 ROWER[3]

Another type was developed by Donghun Lee et al. [42]. The team developed a semi-automatic welding system. It was developed from the RRX mobile platform, and the team named it RRXC. The paper first introduced current types of mobile welding platforms and pointed out the main problem with the current mobile welding robot is the size. The team argues that it is better to have a smaller mobile welding platform that operates under human assistance than a large autonomous robot. Then the design of the team is presented. This system consists of a six-degree-of-freedom welding manipulator and a six-axis controller. A multi-slider system and a fold-up racks are applied to increase the platform's ability to move through narrow spaces. The controller uses a four-layer structure, which consists of a task manager, task planner, actions for task, and a task executer. The kinematic and dynamic simulation for the welding manipulator is carried out in ROBCAD. According to the simulation and field testing, which is carried out in a double-hull ship, the system satisfies the requirements for shipbuilding welding.

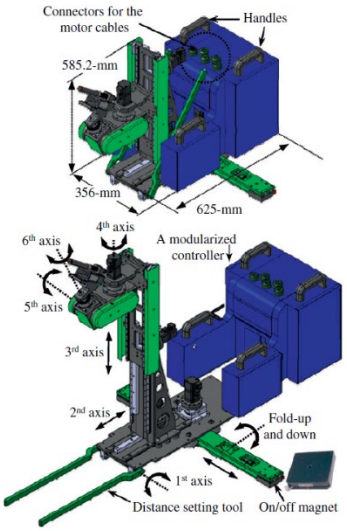

Figure 2 10 RRXC Robot[2]

2.6 Conclusion

In the literature review, dozens of papers regarding several aspects related to the project were reviewed and summarized. For the control system of the robot, a number of researches have already been done from different angles. Generally, the commonly used design of control system for robotic manipulators is a layered structure, with the lower layer

controlling the drive of motors and the higher layer control the direction and speed of the platform. The benefit of this layered structure is that it allows the application of multiple control boards for different layers and thus improves the efficiency and accuracy of the calculation. In application situations where an unstable working environment is involved, the adaptive control is usually used for end effector stability. For the platform that carries the arm, there are several types that are generally used, but as is stated above, the hexapod walking platform is a better solution, thus most of the papers reviewed are about the design of this type of robot, aspects including the specific design of some prototypes, the control system and the gait of the robot. These aspects are reviewed to help understand the movement equation of the platform, which will eventually be the movement equation of the arm base. Innovated methods of establishing the model of the platform and calculating the equation of motion are reviewed to optimize the similar process for this project. For the welding technique, two of the techniques are reviewed and compared. The working method of both techniques are similar, but it is concluded that SWAM is easier to be mobilized and placed on a small platform. There are several prototypes of mobile welding robots that are currently under testing and further study. After a comprehensive study, it can be concluded that the aim of this project is trying to design a mobile welding walking robot of small size and high stability when working in a complicated environment. The advanced control method can be applied for higher accuracy and stability.

3 Research Methodology

1. Mathematic modelling

The D-H Parameter and the Jacobian of a particular robotic manipulator are calculated to form the model in a mathematical sense. This step is to define the configuration of the robotic arm, including joint type, link length, materials, and other properties that are required during the simulation process coming next. The movement of the platform and end effector were also determined in this step.

2. Simulation

With a defined mathematic model of the manipulator and movement function of the platform and end effector, the robotic system was taken into software for simulation. The first software is SolidWorks. It can visualize the model of the manipulator with accurate diameters and material properties. The second software was ADAMS, which is a dynamic simulation software. The SolidWorks model can be recognized and imported into ADAMS for movement simulation. Here the movement functions of the platform and end effector were added. The simulation ran in this software was to determine the output of the joint motors when both ends were moving. The third software was MATLAB. By using MATLAB Simulink, a controller was added to reduce the effect of interference.

3. Data analyze

The simulation results from ADAMS and MATLAB were analyzed. For results from ADAMS, they can determine the maximum torque and speed needed for each of the joints so the type of motors of each joint can be selected. The result from MATLAB presents the efficiency of the controller and helps decide whether other control theories are required.

4 Mathematical modelling

In this section, the basic modelling of robotic manipulators will be established. The key features for system modelling were introduced, including D-H Parameter, Jacobian, and Euler-Lagrange Equation. Examples were interpreted to demonstrate the steps for establishing these features. In addition, the movement of the platform and the end effector were also discussed in this section.

4.1 Basic dynamic modelling

4.1.1 Transform matrix and D-H Parameter of the robotic arm

To present the robotic arm in a mathematical model, the first step is to put the arm into a coordinate system. In the coordinate system, the arm is simplified into a combination of joints, links, and mass centres. To start with, a single degree of freedom arm is presented as follow:
The arm consists of a base, a revolute joint and a link. The joint enables the link to rotate from 0 to 180 degrees, or to put in radian form, 0 to π. Since the joint only rotates in one vertical plane, a two-dimensional coordinate system is established. The origin is placed at the centre of rotation, the x-axis is put horizontally along the rotating plane, and the y-axis is put vertically along the direction of gravitational force. For a single degree of freedom arm, one coordinate system is sufficient. However, consider that more links are expected to be installed on the arm, a second coordinate system is placed at one end of the first link for the position of later links. To separate the different coordinates, the base link will be marked as ox_0y_0, and the second coordinate will be marked as ox_1y_1. In practical cases, there is another coordinate located on the base to present the position of the first joint, or frequently named as shoulder joint, since this case is purely for demonstration of the method, so the base coordinate and the first joint are considered as one. The arm can now be presented in vector form to establish the mathematical model. As is the way every vector is established, the vector form of the link is calculated by

two ends of the link. So, the coordinate of two ends of the link will be (x_0, y_0) and (x_1, y_1), and thus the vector form of the link will be (x_1^0, y_1^0). Assume that the corner between the link and the positive direction of the x-axis is θ, then the rotation of the link can be expressed in a matrix form:
The x-axis coordinate will be:

$$x_1^0 = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \qquad (4.1)$$

And for the y-axis coordinate:

$$y_1^0 = \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} \quad (4.2)$$

And to combine (4.1) and (4.2) to form the matrix form of the arm:

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad (4.3)$$

This matrix is also called the rotation matrix, for it also represents the rotation movement of the links. This matrix can also present the relative position and gesture of two coordinates at two ends of the link. So, this matrix can also be in the form of a dot product of two coordinates. In this case, it will be:

$$R = \begin{pmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 \end{pmatrix} \quad (4.4)$$

For that, the dot product presents the projectile of one coordinate system on another coordinate system.

4.1.1.1 Two degree-of-freedom arm

As the above shown the modelling of one degree of freedom robotic arm, the second degree of freedom is added to demonstrate the movement of the arm in three dimensions.
For a two-degree of freedom robotic arm, the z-axis is added for the third dimension. The configuration can be shown in the following figure (4.1):

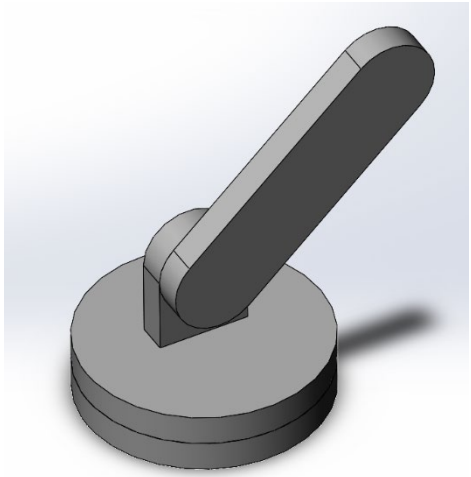

Figure 4. 1Two-degree-of-freedom configuratuin

The method applied to establish the mathematical model for this robotic system is the same as the one introduced above. Instead of a two-element vector, a three-element vector is used. And thus, the rotation matrix is shown as follows:

$$R = \begin{pmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{pmatrix} \quad (4.5)$$

The rotation movement in a three-dimensional system can be disassembled into several individual rotating steps around different axis in a sequence. In this case, when the shoulder joint is rotating around the z-axis, it makes the dot product $z_1 \cdot z_0 = 1$ and all the other dot product related to z-axis zero. Thus, the rotation matrix for this movement will be:

$$R_{z,\theta} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.6)$$

And when the rotation is around the x-axis or y-axis, the matrix can be calculated as follow:

$$R_{x,\theta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad (4.7)$$

$$R_{y,\theta} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \quad (4.8)$$

Now combine the rotation together simply by multiplying them in the order of disassembly. For example, the rotation is disassembled into a rotation around the y-axis and another rotation around the z-axis, then the calculation of the final rotation matrix would be:

$$R = R_{y,\varphi} \cdot R_{z,\theta} \quad (4.9)$$

Where:

$$R_{y,\varphi} = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \quad (4.10)$$

And

$$R_{z,\theta} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.11)$$

And the final rotation matrix would be:

$$R = \begin{pmatrix} \cos\varphi\cos\theta & -\cos\varphi\sin\theta & \sin\varphi \\ \sin\theta & \cos\theta & 0 \\ -\sin\varphi\cos\theta & \sin\varphi\sin\theta & \cos\varphi \end{pmatrix} \quad (4.12)$$

One thing to be noticed is that once the disassembly order of the rotation is decided, the matrix calculation would have to be multiplied in that order. Change of multiply order would change the final result. For example, if the above calculation is reversed, the rotation around the z-axis is calculated first, then the matrix would be:

$$R' = \begin{pmatrix} \cos\varphi\cos\theta & -\sin\theta & \cos\theta\sin\varphi \\ \sin\theta\cos\varphi & \cos\theta & \sin\theta\sin\varphi \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \quad (4.13)$$

After comparing (4.12) and (4.13), $R \neq R'$ And this leads to the conclusion that the order of calculation cannot be altered.

There is another situation, however. In the previous section, the rotation movement was all in order, the later rotation was built on the previous rotation. In other words, the rotation axis of the second movement was formed by the first rotation. Now, in this second situation, the rotation is dissected into several steps still, but all around the three axis of a fixed coordinate system. in this case, the calculation of the rotation matrix is different from the previous situation. The multiplication of matrix order should be changed.

For example, the rotation is divided into one rotation around the y-axis of the base coordinate and one rotation around the z-axis of the base coordinate in that order, then the rotation matrix can be calculated as follow:

$$R = R_{z,\theta} \cdot R_{y,\varphi} \quad (4.14)$$

As shown in the equation (4.14), the later rotation is multiplied in front of the previous rotation.

The above shown the modelling of three-dimensional revolute movement. In real cases though, most of the robotic joints have only one rotation axis, which means that to realize a three-dimensional movement, two independent joints are required.

This leads to a further problem: the inevitable distance between two joints, as shown in figure (4.2) below:

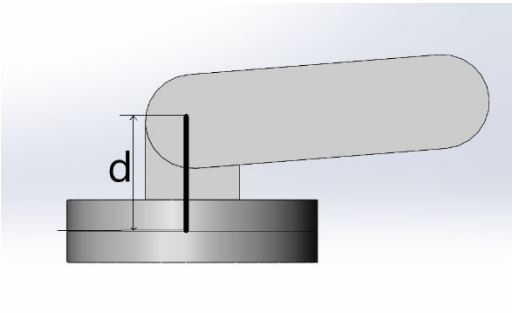

Figure 4. 2Distance between joints

To solve the problem, two coordinate systems are assigned to two joints, and the vector of the first origin pointing to the second origin is assumed to be d_1^0. Thus, a random point on link one, assumed to be p^1, will have a coordinate relative to the first coordinate system $ox_0y_0z_0$ as p^0. According to the rigid movement, this relative coordinate p^0 can be expressed as:

$$p^0 = R_1^0 p^1 + d^0 \quad (4.15)$$

This is the method to present the end-effector position for the one-link robotic arm.

4.1.1.2 Four degree-of-freedom arm

One additional joint and one additional link are added onto the previous arm, adding one more degree-of-freedom. The end effector will be placed on the end of the second link and be marked as p^2. As shown in the figure, the configuration of the manipulator consists of three coordinates: $o_0x_0y_0z_0$, $o_1x_1y_1z_1$ and $o_2x_2y_2z_2$. The vector from o_0 to o_1 is d_1^0, and the vector from o_1 to o_2 is d_2^1. To determine the position of end effector relative to the base coordinate $o_0x_0y_0z_0$, the calculation can be carried out as below:

$$p^1 = R_2^1 p^2 + d_2^1 \quad (4.16)$$

$$p^0 = R_1^0 p^1 + d_1^0 \quad (4.17)$$

By replacing p^1 into p^0, the equation between p^2 and p^0 can be found:

$$p^0 = R_2^0 p^2 + d_2^0 \quad (4.18)$$

Where:

$$R_2^0 = R_1^0 R_2^1 \quad (4.19)$$

And:

$$d_2^0 = d_1^0 + R_1^0 d_2^1 \quad (4.20)$$

Then the coordinate of end effector in the global coordinate system can be calculated and thus, the movement of end effector can be monitored. With this technique, more joints and

links can be added, and the dynamic modelling can be finished by the above method. However, with more joints means more separate coordinate systems, which leads to more and more complicated matrix calculation. For that, the homogeneous transformation is introduced. It transfers the calculation equation into matrix form:

$$H = \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix}, R \in SO(3); d \in R^3 \quad (4.21)$$

And to present end effector:

$$P^0 = H_1^0 P^1 \quad (4.22)$$

In this equation, the position vector of end effector is expanded with an extra element 1:

$$P^0 = \begin{pmatrix} p^0 \\ 1 \end{pmatrix} \quad (4.23)$$

$$P^1 = \begin{pmatrix} p^1 \\ 1 \end{pmatrix} \quad (4.24)$$

The basic homogeneous transformation matrixes can be written as follow:

$$Trans_{x,a} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Rot_{x,\alpha} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

（4.25）

$$Trans_{y,b} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Rot_{y,\beta} = \begin{pmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

（4.26）

$$Trans_{z,c} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Rot_{z,\gamma} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.27)$$

Each corresponding to the translation and rotation movement of the x, y, and z axis. In the case of robotic arms, the rotations of joints are usually limited within the rotation around one particular joint and transaction movement along the length of links. So, with proper configuration of the coordinate system, the movement of each joint can be simplified into a multiplication of four basic homogeneous transformation matrixes:

$$A_i = Rot_{z,\theta} Trans_{z,d} Rot_{x,\alpha} Trans_{x,a}$$

=

$$\begin{pmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & d\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & a\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & \cos a & -\sin a & 0\\ 0 & \sin a & \cos a & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$
（4.28）

$$=\begin{pmatrix}\cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta\\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta\\ 0 & \sin\theta & \cos\theta & d\\ 0 & 0 & 0 & 1\end{pmatrix}$$（4.29）

This is the dynamic matrix for one link, for every link, there is one matrix like the above one and with all the matrixes multiplied, the general dynamic model is established. This is the most commonly used tool in robotic modelling called "Denavit-Hartenberg convention", it provides a uniform way of presenting the configuration and variation of the robotic manipulators. The four variations in the matrix: a, α, d, θ are called "D-H Parameters", they present the only four variations in the D-H convention that are required to establish the dynamic model: Link Length, Link Twist, Link Offset and Link Angle. For each link on the robotic arm, there is a set of four parameters, and eventually, all the parameters fall into a table. Take a robotic manipulator with the following configuration as an example:
The following figure (4.3) shows a simplified drawing of a Stanford Robotic arm, which is an RRP type with a ball joint wrist.

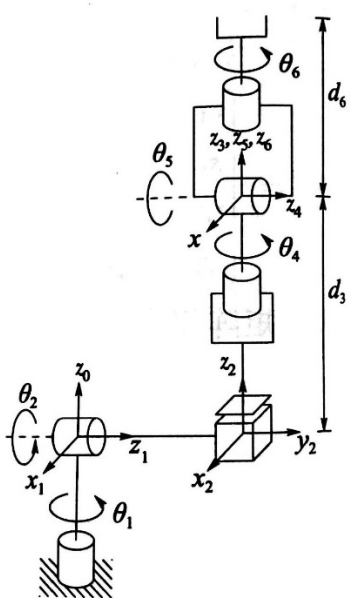

Figure 4. 3 Stanford Robotic arm[43]

The ball joint cannot be presented in one single set of DH parameters. So, it is divided into two separate joints with two orthogonal rotation axials instead. With the established notation

in the figure, the DH parameter table is presented as follow:

link	d_i	a_i	α_i	θ_i
1	0	0	-90	θ_i
2	d_i	0	+90	θ_i
3	d_i	0	0	0
4	0	0	-90	θ_i
5	0	0	+90	θ_i
6	d_i	0	0	θ_i

Table 4. 1D-H Parameter of Stanford Robotic arm

Eventually, the transformation matrix can be calculated as follows:

$$T_6^0 = A_1 \cdots A_6 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.30)$$

Where each r and d represent a complex multiplication of trigonometric functions, the detailed calculation, along with other examples of applying DH convention, will be shown in appendix 1.

The above described the basic process of establishing the kinematic model of the manipulator. The model was established in an order from the base to the end effector and this is called **Forward Kinematics (FK)**. However, in this case, the end effector with welding head bears a higher priority, so it is reasonable to try to establish the model from end effector. This is where the **Inverse Kinematic (IK)** is involved. In FK, the transformation matrix T is calculated by multiplication of transformation matrixes, the IK tries to solve the equation of $T_n^0(q_1 \dots q_n) = H$, where H is the expected position and gesture of the end effector.

$$T_n^0(q_1 \dots q_n) = A_1(q_1) \cdots A_n(q_n) \quad (4.31)$$

In this scenario, q can be either the angle or the transaction distance, in general, it is called the "joint variable". So, the objective is to solve q_1 to q_n. Consider that the end effector is usually in a certain movement, it is better if the solution of q_1 to q_n is in a close loop form where the only variable is the variable that describes the movement of end effector. However, to solve the Inverse Kinematic problem usually means to solve over a dozen of equations with even more variations. To make things easier, a technique called decoupling is applied. It divides the Inverse Kinematic problem into two subsections: inverse position kinematics and inverse orientation kinematics. The following uses a six degree of freedom manipulator with a ball joint wrist as an example:

4.1.1.3 Six degree-of-freedom manipulator with ball joint wrist.

The first step is to work out the origin point for the wrist. The ball joint is divided into three separate joints with three orthogonal rotation axes, the point for these three axes to meet is notified as o_c and the position of end effector in the global coordinate system is marked as o.
The origin can be calculated in following:

$$o_c^0 = o - d_6 R \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.32)$$

Expand the equation into matrix form:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} o_x - d_6 r_{13} \\ o_y - d_6 r_{23} \\ o_z - d_6 r_{33} \end{pmatrix} \quad (4.33)$$

By solving the equation above, the first three joint variables can be obtained, thus the transfer matrix R_3^0 can be calculated.
The general transfer matrix is $R = R_3^0 R_6^3$, with R_3^0 established, R_6^3 can be calculated as well:

$$R_6^3 = (R_3^0)^T R \quad (4.34)$$

The generally applied method for solving the matrix is to use the geometric method. The principle of this method is to project the arm onto the $x_0 - y_0$ plane, as shown in figure (4.4) below:

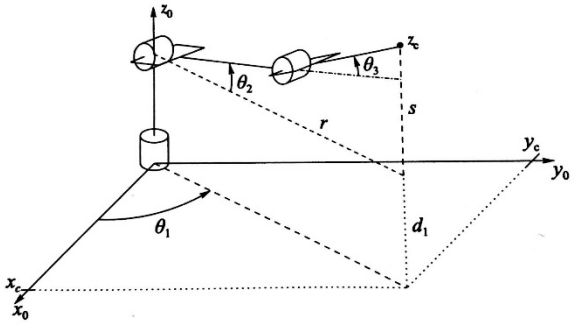

Figure 4. 4 A six-degree-of-freedom arm[43]

From figure (4.4), the solution to θ_1 can be $\theta_1 = A\tan 2(x_c, y_c)$ or $\theta_1 = \pi + A\tan(x_c, y_c)$. These two solutions correspond to two sets of solutions for θ_2 and θ_3.
After θ_1 is calculated, the same method can be applied to θ_2 and θ_3. Usually, for θ_3 there are two solutions:

$$\theta_3 = \mathrm{Atan}\,2\left(D, \pm\sqrt{1 - D^2}\right) \quad (4.35)$$

Where:

$$D = \cos\theta_3 = \frac{r^2+s^2-a_2^2-a_3^2}{2a_2a_3} \quad (4.36)$$

And for θ_2:

$$\theta_2 = \text{Atan}\,2(r,s) - A\tan 2(a_2 + a_3c_3, a_3s_3) \quad (4.37)$$

In some cases, there is an offset between the base and the joint. The offset will result in a more complicated solution to the matrix but in this case would not be considered since the structure is symmetric thus there is no difference between left position or right position.

By calculating the transfer matrix of the first three joints, the result determines the position of the wrist, also known as **inverse position kinematics**. Then the status of the wrist itself will be pursuit, which is the **inverse orientation kinematics**.

By applying a similar method, the orientation can be obtained:

$$\theta_1 = A\tan 2(x_c, y_c) \text{ or } \theta_1 = \pi + A\tan 2(x_c, y_c) \quad (4.38)$$

$$\theta_2 = \text{Atan}\,2(r,s) + \frac{\pi}{2} \quad (4.39)$$

With every angular variable presented in algebra form, an established position matrix of end effector can lead to numerical solution of joint variables. The inverse kinematics is usually applied after the forward kinematics is established to test if there are other possible solutions to the acquired end effector position.

4.1.2 The Jacobian

The DH convention is good enough to establish a mathematical model of the robotic in a three-dimensional space. However, to describe the movement of joints and end effector, velocity-related variables need to be taken into consideration. This is where the Jacobian Matrix is involved.

For an n-degree-of-freedom manipulator with joint variable $q_1 \cdots q_n$, the transfer matrix for end effector to the base can be written as:

$$T_n^0(q) = \begin{pmatrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{pmatrix} \quad (4.40)$$

In the above matrix, the $R_n^0(q)$ and $o_n^0(q)$ are both time-related functions, which represent the orientation and position of the end effector. To relate these two functions to the linear and angular velocity of joints, the following assumptions are made:

$$S(\omega_n^0) = \dot{R_n^0}(R_n^0)^T \quad (4.41)$$

$$v_n^0 = \dot{o_n^0} \quad (4.42)$$

Where ω_n^0 is the angular velocity of the end effector and v_n^0 is the linear velocity of the end effector. The expression for these two variables is expected to be:

$$v_n^0 = J_v \dot{q} \quad (4.43)$$

$$\omega_n^0 = J_\omega \dot{q} \quad (4.44)$$

To further simplify:

$$\varepsilon = J\dot{q} \quad (4.45)$$

$$\varepsilon = \begin{pmatrix} v_n^0 \\ \omega_n^0 \end{pmatrix} \quad (4.46)$$

$$\text{and } J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \quad (4.47)$$

And J will be the Jacobian in question.

For an n-link manipulator, the angular velocity can be calculated as:

$$\omega_n^0 = \rho_1 \dot{q}_1 k + \rho_2 \dot{q_2} R_1^0 k + \cdots + \rho_n \dot{q_n} R_{n-1}^0 k = \sum_{i=1}^n \rho_i \dot{q}_\iota z_{i-1}^0 \quad (4.48)$$

$$\rho_i = \begin{cases} 1 & when\ joint\ i\ is\ revolute \\ 0 & when\ joint\ i\ is\ linear \end{cases}$$

This leads to the angular velocity element of the Jacobian:

$$J_\omega = (\rho_1 z_0 \cdots \rho_n z_{n-1}) \quad (4.49)$$

For the linear velocity, $J_{vi} = \frac{\partial o_n^0}{\partial q_i}$. (4.50)

In linear joint, $\dot{o_n^0} = \dot{d}_\iota R_{i-1}^0 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \dot{d}_\iota z_{i-1}^0$, and thus $J_{vi} = z_{i-1}$ (4.51)

In revolute joint, $v = \omega \times r$, $\omega = \dot{\theta}_i z_{i-1}$ and $r = o_i - o_{i-1}$. Thus $J_{vi} = z_{i-1} \times (o_n - o_{i-1})$.

So, the general equation for Jacobian Matrix will be:

$$J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \quad (4.52)$$

$$\text{While } J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & for\ revolute\ joint \\ z_{i-1} & for\ linear\ joint \end{cases}$$

$$\text{And } J_{\omega_i} = \begin{cases} z_{i-1} & for\ revolute\ joint \\ 0 & for\ linear\ joint \end{cases}$$

Usually, for manipulators, there is expected to have another matrix called Tool Speed, which is the relative transfer matrix for tool placed on the end effector and the end effector. In this case, the welding head is hard connected to the end effector, so a tool matrix will be added but will not be taken into consideration during the following calculation and simulation to simplify the process.

As the D-H parameter and Jacobian describes the position and velocity of the robotic system, the dynamics of the system remain unsolved. In the following part, the dynamics of the robotic arm will be solved by applying the Euler-Lagrange equation. With a general appliable dynamic Euler-Lagrange equation, the multi-variable control method will be introduced and thus leads to robust adaptive control.

4.1.3 Euler-Lagrange equation in the dynamic system

By applying Newton's second law of motion, the motion equation will be:

$$m\ddot{y} = f - mg \quad (4.53)$$

The left side of the equation can be expanded:

$$m\ddot{y} = \frac{d}{dt}(m\dot{y}) \quad (4.54)$$

$$= \frac{d}{dt}\frac{\partial}{\partial \dot{y}}\left(\frac{1}{2}m\dot{y}^2\right) = \frac{d}{dt}\frac{\partial K}{\partial \dot{y}} \quad (4.55)$$

Where K in the equation presents the kinematic energy of the system.

And similarly, the gravitational force in the original motion equation can be expressed as:

$$mg = \frac{\partial}{\partial y}(mgy) = \frac{\partial P}{\partial y} \quad (4.56)$$

Where P stands for gravitational potential energy.

By calculating $K - P$, the Lagrange Multiplier, $\mathcal{L}$, is obtained.

$$\mathcal{L} = K - P = \frac{1}{2}m\dot{y}^2 - mgy \quad (4.57)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{y}} = \frac{\partial K}{\partial \dot{y}}, \frac{\partial \mathcal{L}}{\partial y} = -\frac{\partial P}{\partial y} \quad (4.58)$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{y}} - \frac{\partial \mathcal{L}}{\partial y} = f \quad (4.59)$$

And thus, the Euler-Lagrange equation is obtained, there is a more detailed show of the process to obtain the Euler-Lagrange equation, which will be shown in appendix 2.

After the general equation is obtained, the kinematic energy and potential energy specifically for the robotic arm are required.

For a rigid body, the kinematic energy can be expressed as:

$$K = \frac{1}{2}mv^T v + \frac{1}{2}\omega^T I\omega \quad (4.60)$$

Where I in the equation presents the inertia tensor.

As for the potential energy, it can be presented as:

$$P_i = m_i g^T r_{ci} \quad (4.61)$$

In which r_{ci} stands for the position vector of the mass centre

for the rigid body.

The n-degree-of-freedom robotic arm can be considered as a combination of multiple rigid bodies. Thus, the kinematic energy and potential energy of the system can be calculated by integral as below:

$$K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \quad (4.62)$$

$$D(q) = \left[\sum_{i=1}^{n} m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)\right] \quad (1.63)$$

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} m_i g^T r_{ci} \quad (4.64)$$

With K and P acquired, the Lagrange Multiplier can be calculated:

$$\mathcal{L} = K - P = \frac{1}{2}\sum_{i,j} d_{ij}(q)\dot{q}_i\dot{q}_j - P(q) \quad (4.65)$$

And to calculate Euler-Lagrange equation, substitute equation (1.65) into equation (1.59)

$$\sum_j d_{kj}\ddot{q}_j + \sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j + \frac{\partial P}{\partial q_k} = \tau_k \quad (4.66)$$

One of the terms can be expressed as:

$$\sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j = \sum_{i,j}\frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j = \sum_{i,j} c_{ijk}\dot{q}_i\dot{q}_j \quad (4.67)$$

The whole Euler-Lagrange equation can be re-written as:

$$\sum_{j=1}^{n} d_{kj}(q)\ddot{q}_j + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ijk}(q)\dot{q}_i\dot{q}_j + gk(q) = \tau_k, k = 1, \cdots, n \quad (4.68)$$

In order to simplify the display of the equation, the above equation is usually shown as follow:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \quad (4.69)$$

Which is the generalized form of Euler-Lagrange equation for robotic arms.

For a walking-welding robot, the priority is the movement stability of the welding head since the welding quality depends on it. In the theoretical environment, a traditional controller like open-loop control is sufficient enough, with the consolidated gait of the moving platform, the movement of the end effector can also be predetermined. When it comes to reality, there will be various factors, no matter expected or unexpected, to interfere with the actual movement. Possible factors include vibration from the motors, deformation of materials during the working process and unexpected variation of the terrains. This is where robust and adaptive control is involved.

The core theory of these two controllers is to track the uncertainty of the system and by applying inner-loop/outer-loop control to minimize it so that the end effector can carry the required movement as stable as possible. An inner-loop/outer-

loop control is actually a control system architect, it consists of four parts: Trajectory planner, outer-loop controller, inner-loop controller and the robot. The non-linear control is accomplished in the inner-loop controller where, as shown in the Euler-Lagrange equation, $q, \dot{q}, \ddot{q}$ are input and τ is the output, or in this part, τ can also be noted as u. The inner-loop controller deals with the fundamental movement of the system as if no interference factor mentioned above is involved. The actual interference is dealt with by the outer-loop controller. The control diagram is shown in figure (4.5) below:

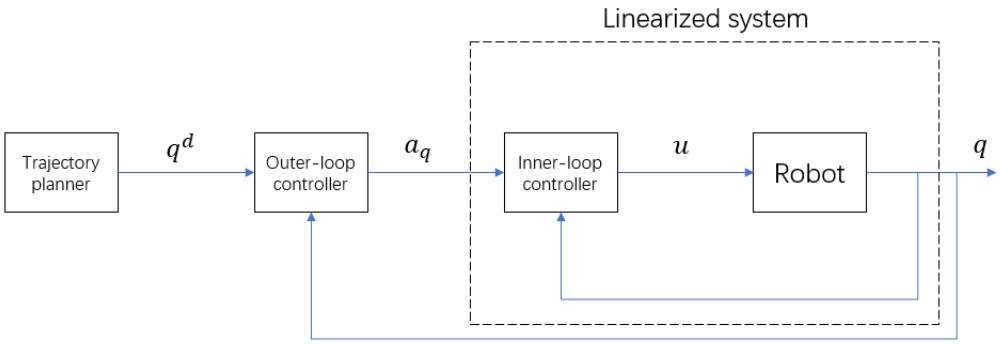

Figure 4. 5 Control System

Robust and adaptive control is achieved by adding extra modification on the outer-loop controller so that it can self-adapt to the change from the system itself or from the outer environment. It has been generally recognized that robust control performs better when dealing with environmental impact or unmodelled dynamic features, while adaptive control deals better with variations from the system itself. Consider that the application situation is most likely to be in a closed indoor environment and welding most likely will not encounter various force changes during the working procedure, adaptive control will be the focus here.

Consider the Euler-Lagrange equation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \quad (4.70)$$

In which u stands for the inverse dynamics input

$$u = \widehat{M}(q)a_q + \hat{C}(q,\dot{q}) + \hat{g}(q) \quad (4.71)$$

In the equation, $\hat{(\cdot)}$ stands for the representation value of $(\cdot)$, which means that due to the uncertainty of the system, an accurate inverse dynamic control cannot be achieved, and deviation and errors need to be considered. $\tilde{(\cdot)}$ stands for the

error of the system and can be calculated as $\tilde{(\cdot)} = \hat{(\cdot)} - (\cdot)$
Combine the above two equations together, it can be calculated that:

$$\ddot{q} = a_q + \eta\left(q, \dot{q}, a_q\right) \quad (4.72)$$

In which

$$\eta = M^{-1}\left(\tilde{M}a_q + \tilde{C}\dot{q} + \tilde{g}\right) \quad (4.73)$$

And η stands for uncertainty.
In adaptive control, the system parameter cannot be fixed, so they need to be estimated.

Set a_q

$$a_q = \ddot{q}^d - K_1(\dot{q} - \dot{q}^d) - K_0(q - q^d) \quad (4.74)$$

Where K_1 and K_0 are the gain matrix

$$K_0 = \begin{pmatrix} \omega_1^2 & 0 & \cdots & 0 \\ 0 & \omega_2^2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_n^2 \end{pmatrix} \quad (4.75)$$

$$K_1 = \begin{pmatrix} 2\omega_1 & 0 & \cdots & 0 \\ 0 & 2\omega_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 2\omega_n \end{pmatrix} \quad (4.76)$$

With linearized parameters of the system, it can be calculated:

$$\ddot{\tilde{q}} + K_1\dot{\tilde{q}} + K_0\tilde{q} = \hat{M}^{-1}Y(q, \dot{q}, \ddot{q})\tilde{\theta} \quad (4.77)$$

In which Y is the regression function.
And there is $\tilde{\theta} = \hat{\theta} - \theta$
Rewrite the equation (4.77) in state space:

$$\dot{e} = Ae + B\Phi\tilde{\theta} \quad (4.78)$$

Where:

$$A = \begin{pmatrix} 0 & I \\ -K_0 & -K_1 \end{pmatrix} \quad (4.79)$$

$$B = \begin{pmatrix} 0 \\ I \end{pmatrix} \quad (4.80)$$

$$\Phi = \hat{M}^{-1}Y(q, \dot{q}, \ddot{q}) \quad (4.81)$$

5 Simulation process and results

A six-degree-of-freedom robotic manipulator was applied with a welding torch hard connected to the end effector in the simulation process, which was designed by another student in the same project group. For this research, the robotic manipulator was simplified into joints and links. The mass of the arm will be concentrated onto mass centres located on each link or joint.
The simulation process is divided into two parts. The first part involves SolidWorks and ADAMS for a visualized simulation result. The second part involves MATLAB, where the control theory is applied in a simulating model in Simulink and the simulation result is compared with the result from ADAMS to verify the designed control system.

5.1 Numerical and 3D modelling of the robotic arm.

5.1.1 SolidWorks

Although ADAMS provides the function of 3D modelling inside itself, it is still easier and quicker to use SolidWorks to build a model first and import it into ADAMS. As is mentioned, this research only focuses on the configuration of the robotic arms, not the design detail, so the parts modelled inside SolidWorks can be simple as shown below:

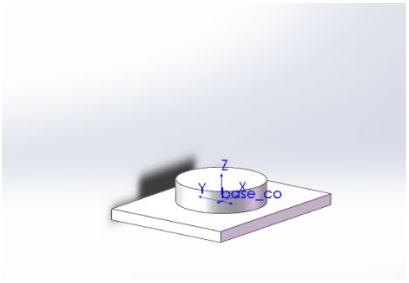

Figure 5. 1 Base joint

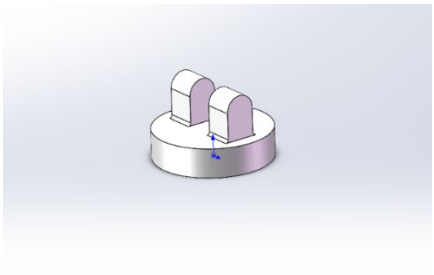

Figure 5. 2 Joint 1

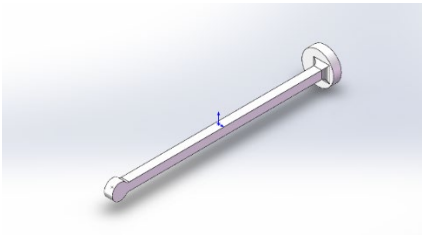

Figure 5. 3 Link 1

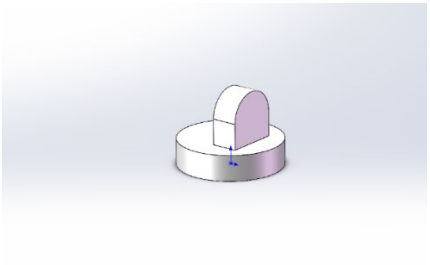

Figure 5. 4 Joint 2 and 3

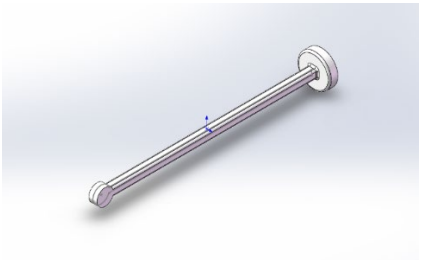

Figure 5. 5 Link 2

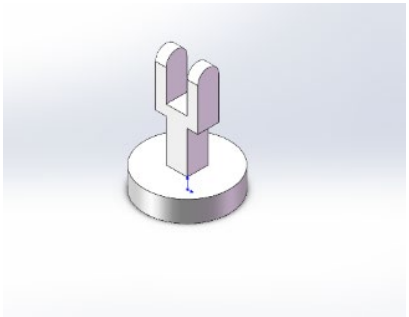

Figure 5. 6 Joint 4 and 5

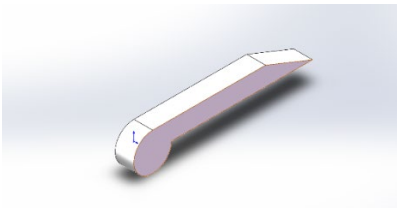

Figure 5. 7 Link 3

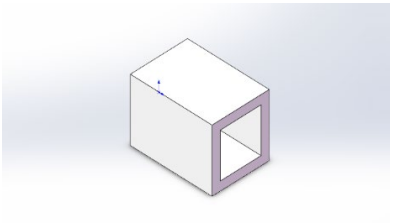

Figure 5. 8 End effector

The assembled robotic arm is shown as follow:

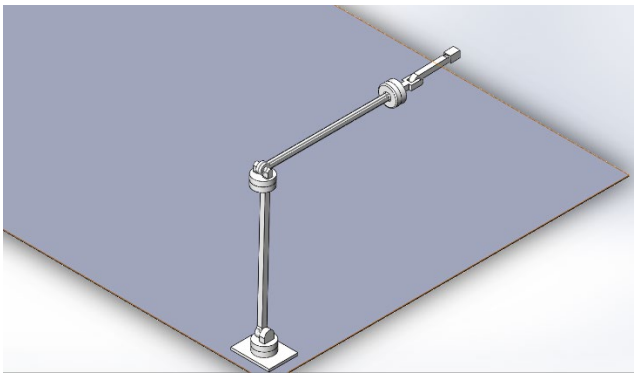

Figure 5. 9 Assembled arm

As is shown in figure (5.9), the robotic arm has six-degree-of-freedom, each joint consists of two independent revolute joints and according to a generally accepted naming method, joints are named as “shoulder”, “elbow” and “wrist”.

After the 3D modelling is finished, it can be taken into ADAMS for dynamic modelling. The first step is to establish a numerical model of the arm.

5.1.2 D-H Parameter:

To give a full D-H Parameter of the system, a three-dimensional coordinate system needs to be added to each separate joint. In this case, a six-degree-of-freedom robotic manipulator would require at least six-coordinate systems:

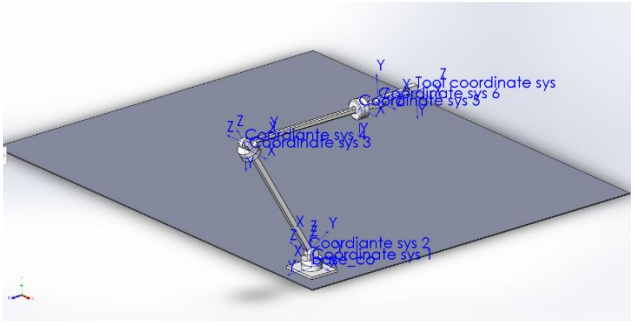

Figure 5. 10 Coordinate configuration

And to further simplify the diagram:

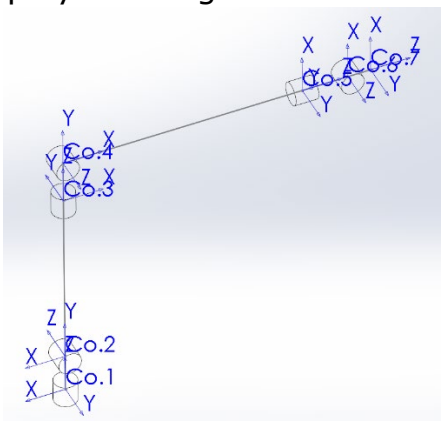

Figure 5. 11 Simplified coordinate system

As was mentioned in the mathematical modelling section, the D-H parameter of the arm would be:

Link	a_i	α_i	d_i	θ_i
1	0	90	0	θ_1
2	0	90	0	θ_2
3	0	90	0	θ_3
4	0	90	0	θ_4
5	0	90	0	θ_5
6	0	90	0	θ_6

Table 5. 1 D-H Parameter of the arm

And the Jacobian of the end effector would be calculated as follow:

5.1.3 The Jacobian

For this robotic manipulator, there are total of six degree-of-freedoms. So, the transfer matrix of the end effector can be written as:

$$T_6^0(q) = \begin{pmatrix} R_6^0(q) & o_6^0(q) \\ 0 & 1 \end{pmatrix} \quad (5.1)$$

The angular velocity and linear velocity for the end effector can

be calculated as below:

$$S(\omega_6^0) = \dot{R_6^0}(R_6^0)^T \quad (5.2)$$

$$v_6^0 = \dot{o_6^0} \quad (5.3)$$

$$v_6^0 = J_v\dot{q} \quad (5.4)$$

$$\omega_6^0 = J_\omega\dot{q} \quad (5.5)$$

To further simplify:

$$\varepsilon = J\dot{q} \quad (5.6)$$

$$\varepsilon = \begin{pmatrix} v_6^0 \\ \omega_6^0 \end{pmatrix} \text{ and } J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \quad (5.7)$$

The matrix J is the Jacobian of the system
To solve for the angular velocity:

$$\omega_n^0 = \rho_1\dot{q}_1k + \rho_2\dot{q_2}R_1^0k + \cdots + \rho_n\dot{q_n}R_{n-1}^0k = \sum_{i=1}^n \rho_i\dot{q}_\iota z_{i-1}^0 \quad (5.8)$$

$$\rho_i = \begin{cases} 1 & when\ joint\ i\ is\ revolute \\ 0 & when\ joint\ i\ is\ linear \end{cases}$$

In this robotic manipulator, all joints are revolute, so the ρ_i will always be 1. Thus the angular velocity is:

$$\omega_6^0 = \sum_{i=1}^6 \dot{q}_\iota z_{i-1}^0 \quad (5.9)$$

The angular velocity element in the Jacobian matrix is:

$$J_\omega = (z_0 \cdots z_{n-1}) \quad (5.10)$$

Where $z_{n-1} = R_{i-1}k$
And $k = (0,0,1)^T$

As for the linear velocity of the end effector, $J_{vi} = \frac{\partial o_6^0}{\partial q_i}$ (5.11)

Again, all the joints are revolute joint so:

$$J_{vi} = z_{i-1} \times (o_6 - o_{i-1}) \quad (5.12)$$

To sum up
The Jacobian of this robotic manipulator can be written as:

$$J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \quad (5.13)$$

Where:

$$J_v = (J_{v1} \ldots J_{v6}) \quad (5.14)$$

$$J_\omega = (J_{\omega 1} \ldots J_{\omega 6}) \quad (5.15)$$

5.2 Motion simulation

5.2.1 ADAMS

At this point, the kinematic model of the arm is finished and can be taken into the software for simulation of motion.
In ADAMS, import the .parasolid version of the model. An

additional part named ”ground” is added to the assembly as the base reference of movement.

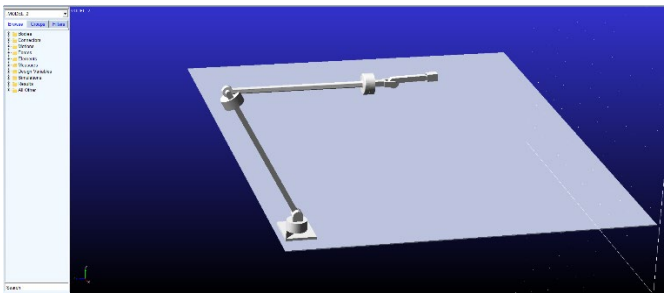

Figure 5. 12 ADAMS window

In ADAMS, the original properties set up in SolidWorks will all be lost, thus there need to be several steps to re-define the features before the simulation process can be carried out.

First, all the connections need to be recognized.

In SolidWorks, all the connections are established by the action of “mate”. However, such boundaries cannot be recognized by ADAMS, so the connections need to be set manually again. The following lists the connections on this model in ADMAS:

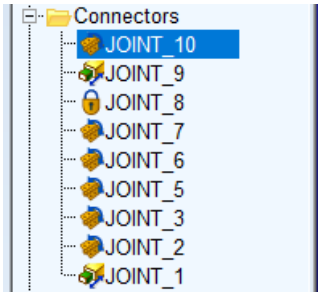

Figure 5. 13 Joints

Joint 1 is the linear movement between the arm base and the ground. There should be the model of the robot platform and legs, but they are not the priority of this research, so they will not be in this part of the simulation.

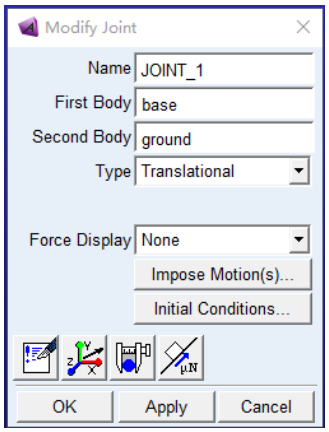

Figure 5. 14 Configuration of joint

Joint 9 is the translational movement between the end effector and the ground. This movement needs to be separated from the

base-ground movement.

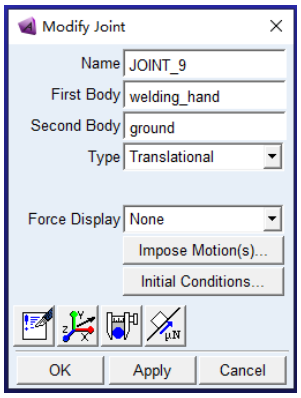

Figure 5. 15 Joint 9 configuration

The rest of the joints are all revolute joints.

With all the features set and gravitational force set along negative Y-axis of the global coordinate system, the kinematic simulation can be in process.

The movement of the base is dependent on the gait of the robot. In this research, the gait in assumption is the insect-wave gait, which is one of the most used gaits for hexagon walking robots. For this gait, each move involves three legs while the other three remain steady to support the platform. With the movement of legs, the platform moves in steps, and the velocity of the platform needs to be determined by actual simulation. In this research, at first, the speed of the base will be set to a step function with a time-related variable to reproduce the step movement of the platform. More complexity will be added later to try to reproduce the real movement of the platform.

In ADAMS, there are two types of function that can generate a step function, IF function and $STEP$ function. For this simulation, the IF function is applied to generate a repeatable movement of the platform, show as $IF(sin(time):3,0.5,0)$, which means when $sin(time)$ is positive, the velocity of the base is 3, as for the rest situation, the velocity of the base is 0.

For the end effector, the speed depends on the requirement of the welding head. The type of welding applied is SMAW, which is the Shielded Metal Arc Welding. The suggested travel speed for SMAW is 75 to 150 mm per minute, which is 1.25 to 2.5 mm per second. In this simulation, the speed is set to 1.5 mm/sec.

After these two movements are set up, run the simulation:

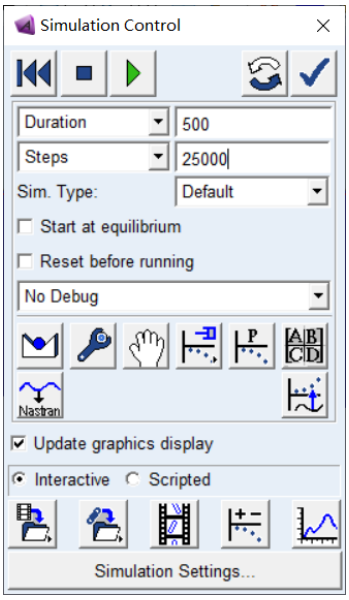

Figure 5. 16 Simulation control

The time duration is set to 500 seconds and divided into 25000 steps, so 50 steps per second. This can provide several full cycles of the movement and good accuracy of velocities and forces in the plotted diagram later.

Figure (5.17) to Figure (5.22) shows the angel of joints and Figure (5.23) to Figure (5.28) shows the angular velocity of joints:

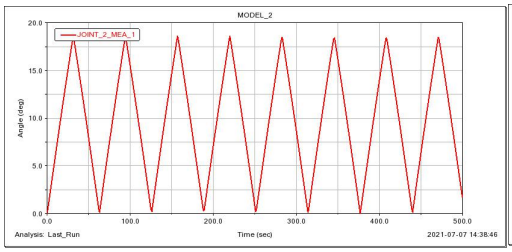

Figure 5. 17Joint 1 angle

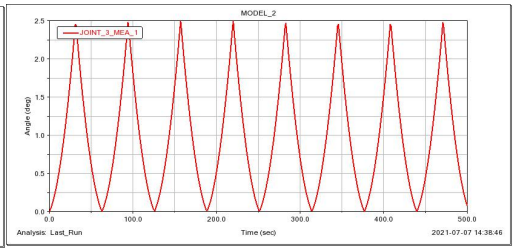

Figure 5. 18Joint 2 angle

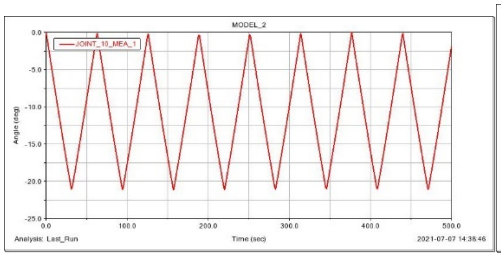

Figure 5. 19Joint 3 angle

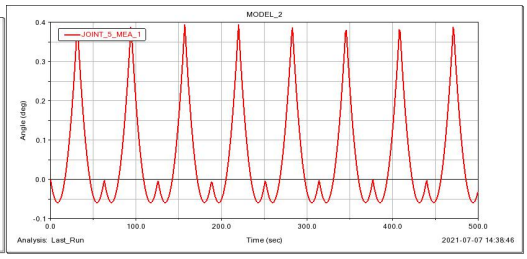

Figure 5. 20Joint 4 angle

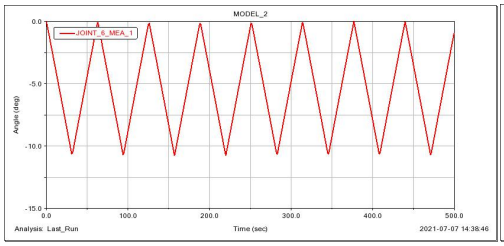

Figure 5. 21Joint 5 angle

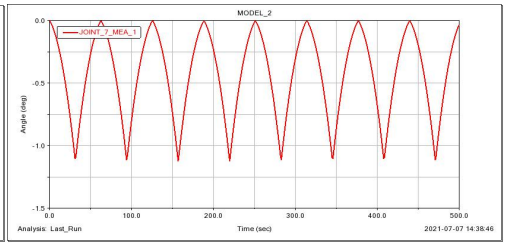

Figure 5. 22Joint 6 angle

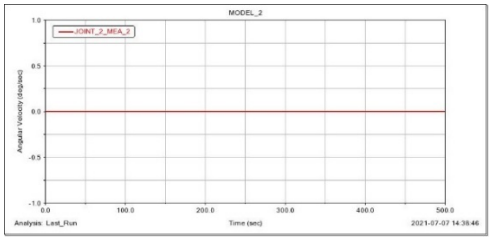

Figure 5. 23Joint 1 velocity

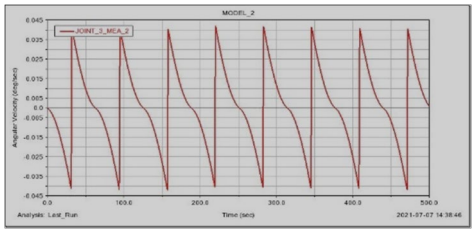

Figure 5. 24Joint 2 velocity

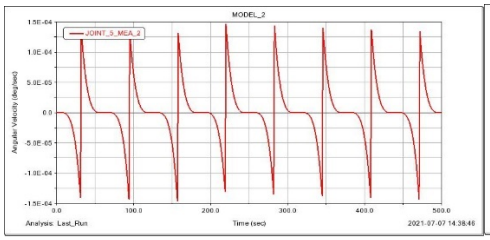

Figure 5. 25Joint 3 velocity

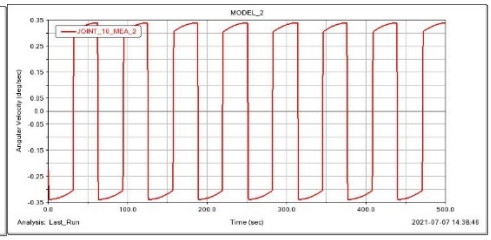

Figure 5. 26Joint 4 velocity

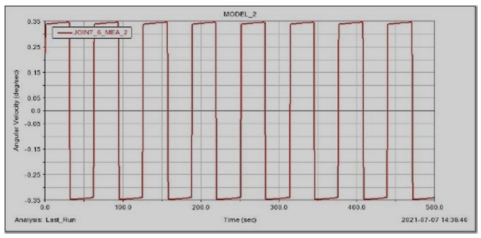

Figure 5. 27Joint 5 velocity

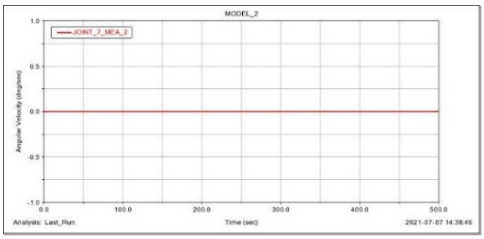

Figure 5. 28Joint 6 velocity

The above figures provide the speed and force on each joint as required to carry out such movement. These results are deducted from the result, which is the movement of the base and end effector. The next step is to put the system into MATLAB and verify the function calculated from the above section. Consider that in a practical situation, the control variables are the torque and angles of the motors. In this step, the input will be the motion function of the base, and the output will be the torque and angular velocity of each joint. Furthermore, the result in this section will be compared with the result in the above section to determine the accuracy and thus the feasibility of the function.

To introduce the ADAMS model into MATLAB, there are few steps

that need to be followed:
The first step is to set the driving mechanism on the model. In this case, to determine the force on the system.
In ADAMS, add the applied force on each of the rotational connections as shown below:

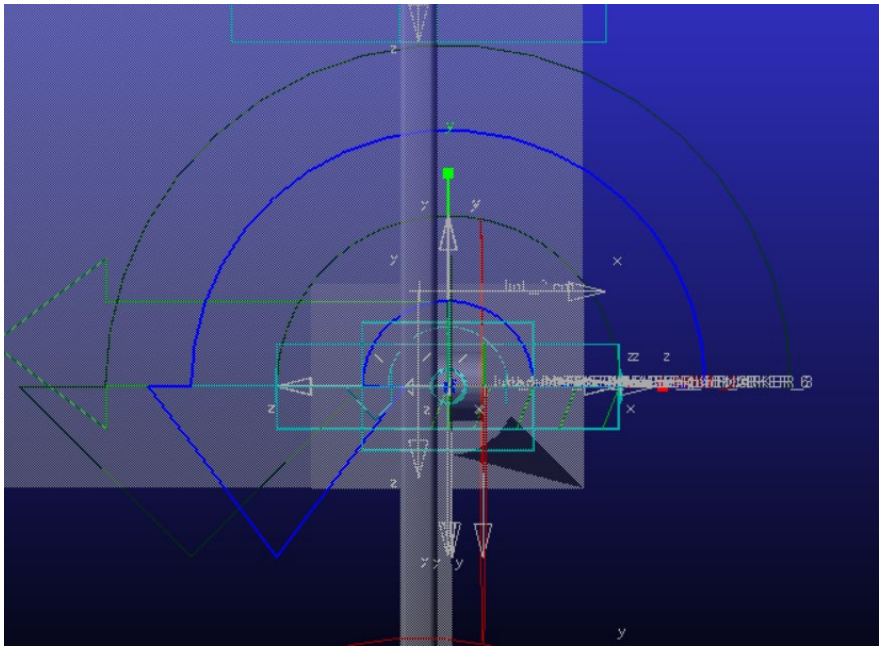

Figure 5. 29Applied torque on first joint

The above shows the first torque applied on the joint (the blue arrow), in general, six torques like the above will be added to each of the joints.
Next, add elements to the system, including the torque, the angle of each joint and angular velocity. Also, set up the measurements for the torque, angle and angular velocity as the monitor.

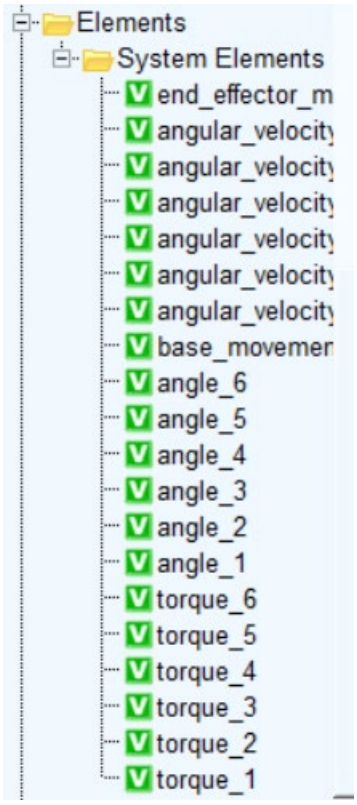

Figure 5. 30Element list

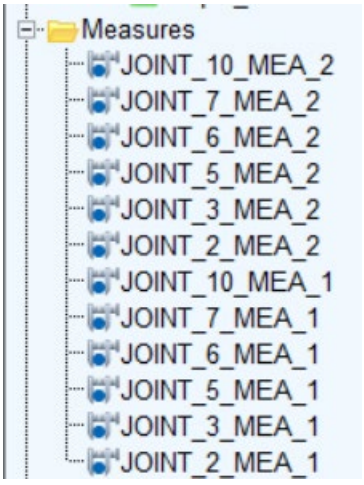

Figure 5. 31Output list

Then, substitute the set elements into the value of all the applied forces established above. These will be listed as “inputs” and “outputs” later.

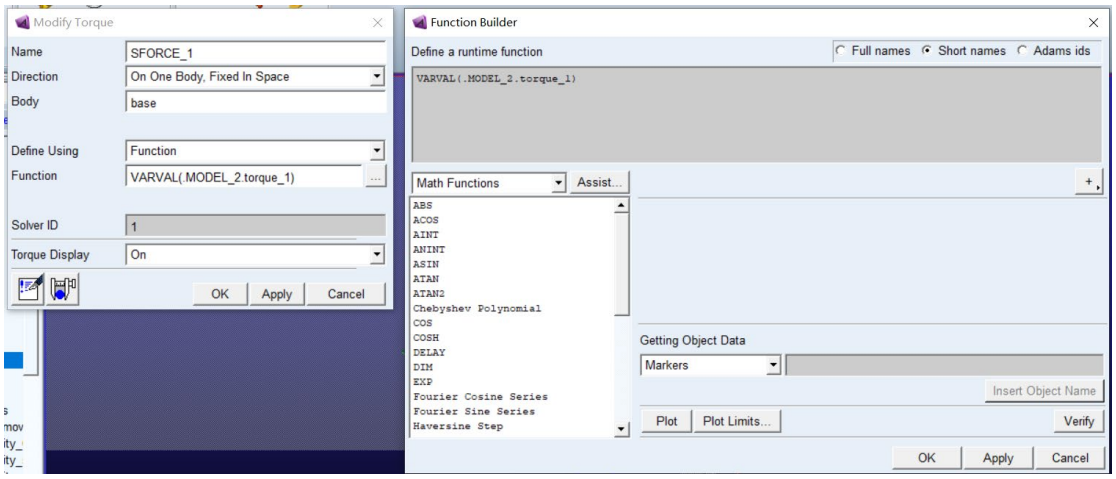

Figure 5. 32User defined function for the applied torque

Use the "controls" function in plugins to generate the system that can be imported into MATLAB. Mark the six torques as the input of the system and the angle and angular velocity as the output. This procedure produces several files that can be read by MATLAB.

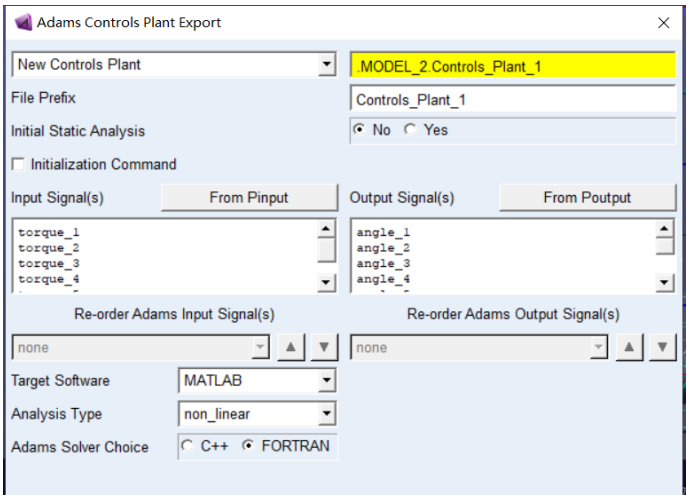

Figure 5. 33Control function for interface with MATLAB

5.2.2 MATLAB

Turn to MATLAB and open the .m file just generated in the command window. It provides the variables that are marked as input and output just now in ADAMS, then generate the Simulink and open the system:

```
%%% INFO : ADAMS plant actuators names :
1 torque_1
2 torque_2
3 torque_3
4 torque_4
5 torque_5
6 torque_6
%%% INFO : ADAMS plant sensors   names :
 1 angle_1
 2 angle_2
 3 angle_3
 4 angle_4
 5 angle_5
 6 angle_6
 7 angular_velocity_1
 8 angular_velocity_2
 9 angular_velocity_3
10 angular_velocity_4
11 angular_velocity_5
12 angular_velocity_6

>> adams_sys
>>
```

Figure 5. 34Element check in Simulink

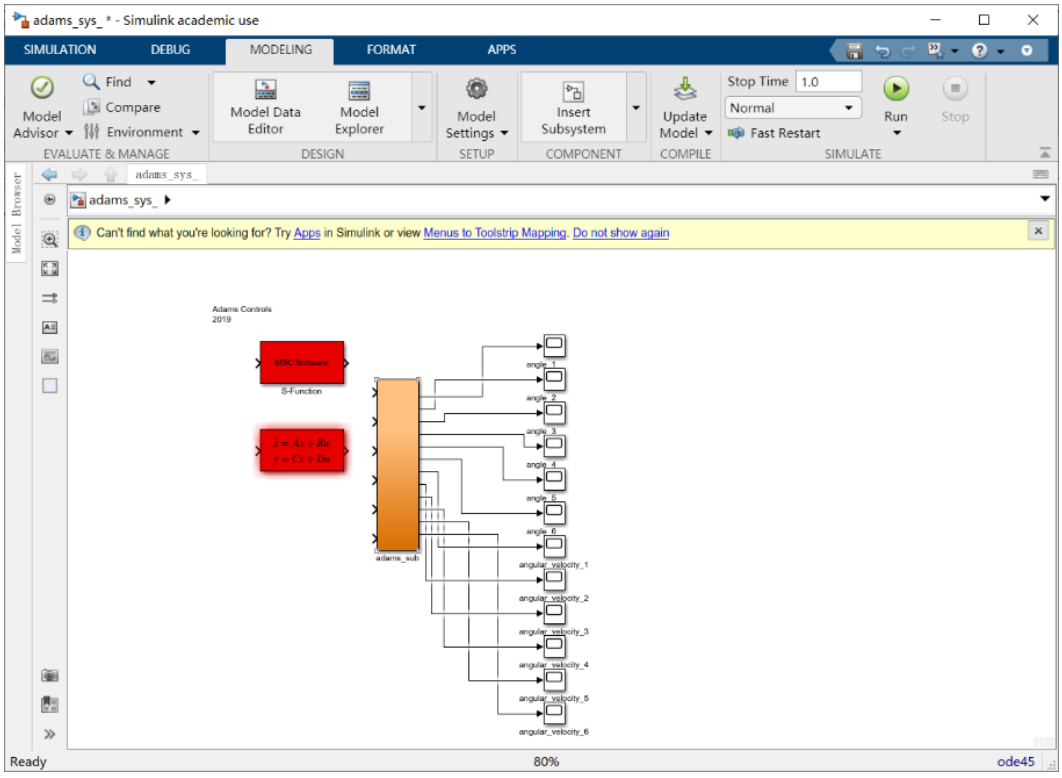

Figure 5. 35Origin system from ADAMS

As shown in figure (5.35), the generated system from ADAMS only provides measures set up earlier. The control system is designed and set up manually. In this case, a PD control system is applied as the primary adaptive control system.

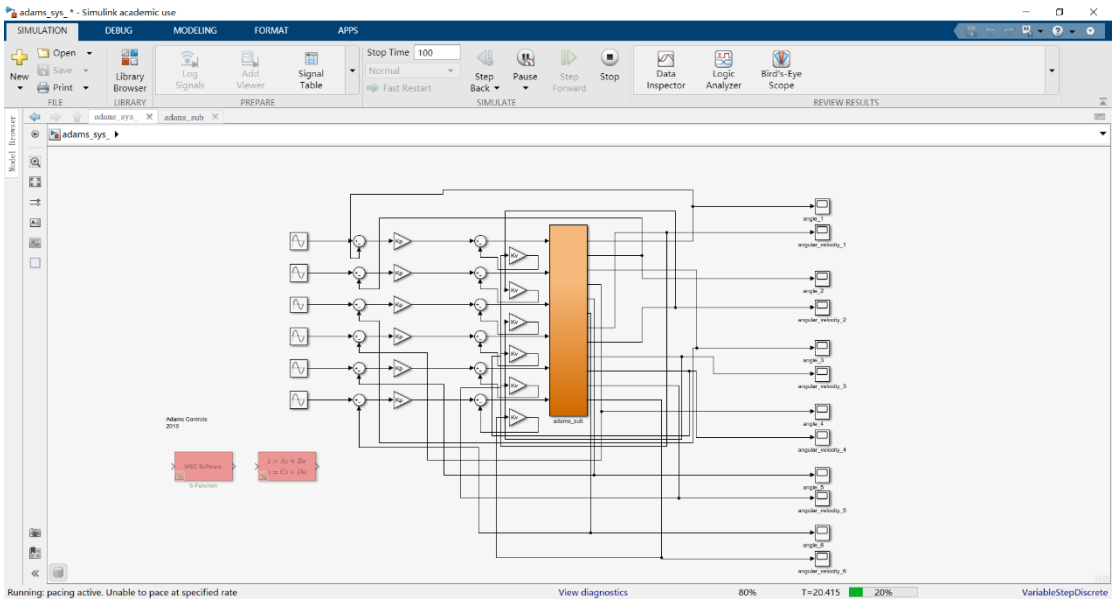

Figure 5. 36PD control system

As shown in figure (5.36), a PD controller is added to each of the joints, and six of those controllers are combined to form the general control system.

Set the input as a sin wave as the lurking variable. Then run the simulation for a time of 200 seconds, which is approximately one cycle of the working procedure. After the simulation, the position and angular velocity of each joint are shown through the scope, and the diagrams are displayed below:

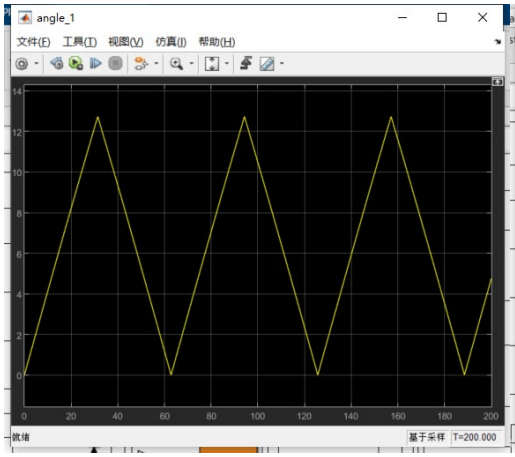

Figure 5. 37Joint 1 angle

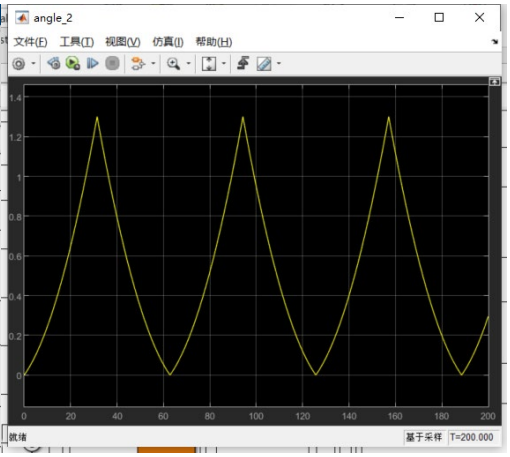

Figure 5. 38Joint 2 angle

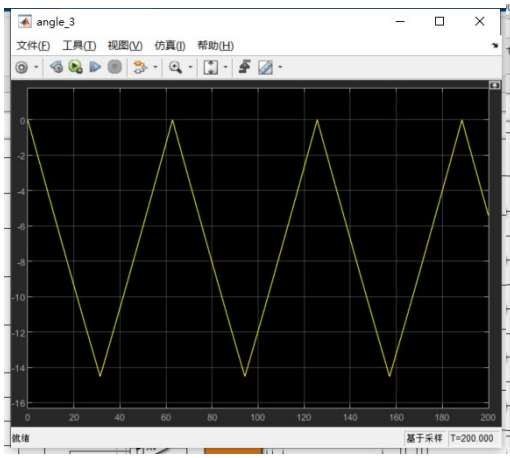

Figure 5. 39 Joint 3 angle

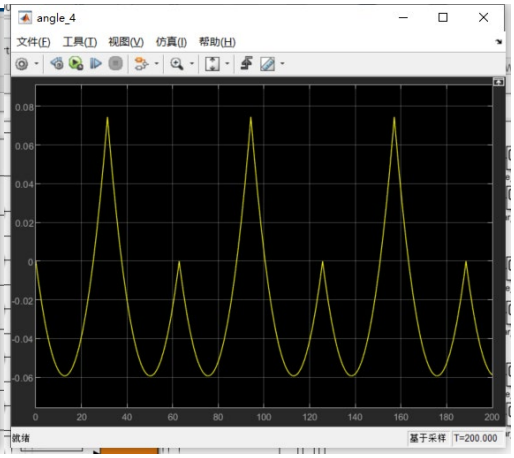

Figure 5. 40 Joint 4 angle

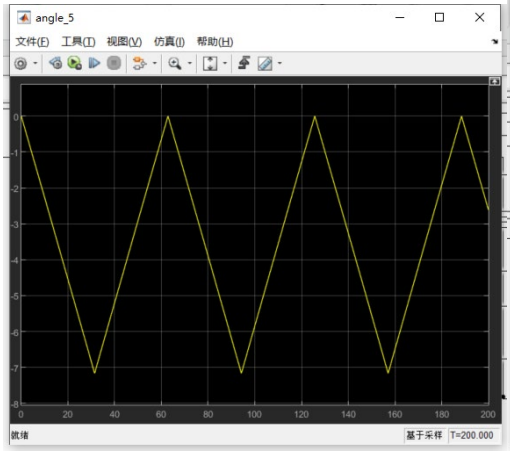

Figure 5. 41 Joint 5 angle

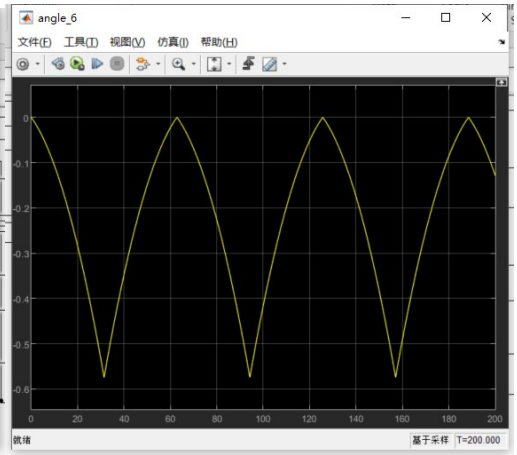

Figure 5. 42 Joint 6 angle

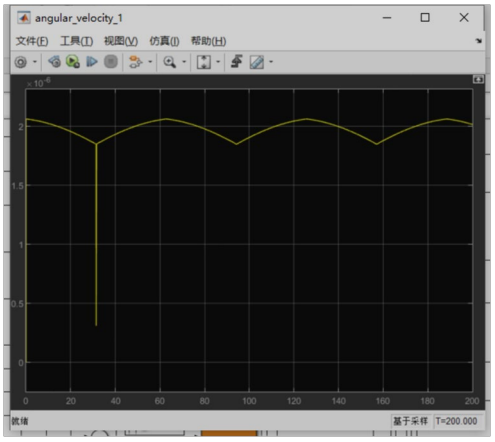

Figure 5. 43Joint 1 velocity

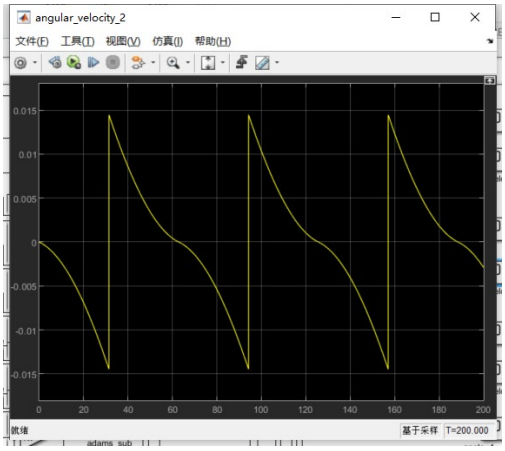

Figure 5. 44 Joint 1 velocity

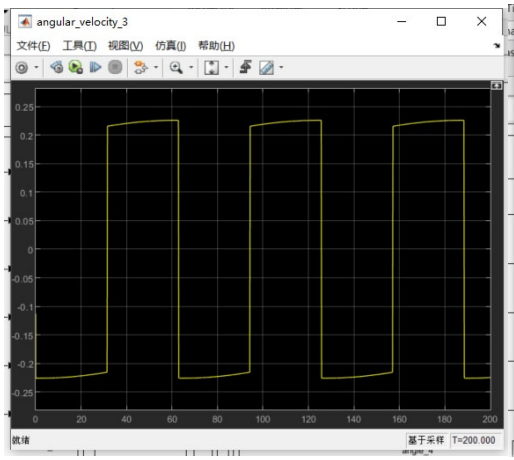

Figure 5. 45 Joint 1 velocity

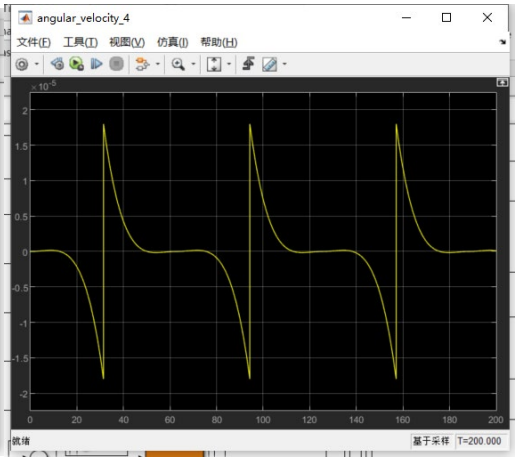

Figure 5. 46 Joint 1 velocity

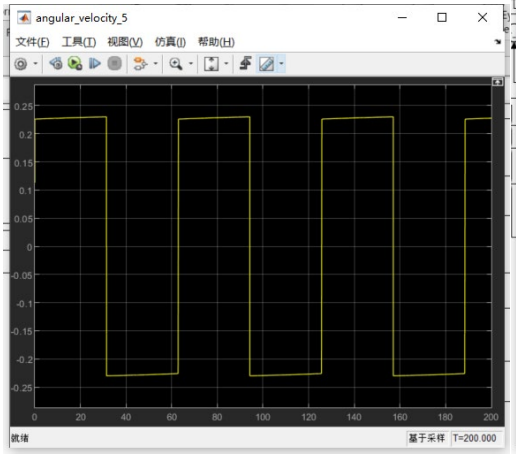

Figure 5. 47 Joint 1 velocity

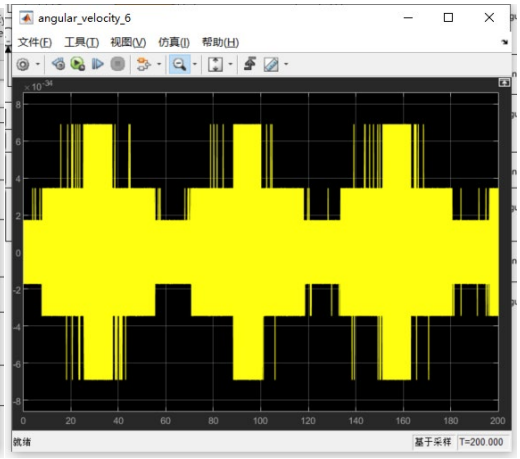

Figure 5. 48 Joint 1 velocity

Compare the results shown by MATLAB and the above results from ADAMS, and it can be seen that the error on the diagram is rather small between the ADAMS results and the MATLAB results. As a matter of fact, some of the peaks and turns on the diagram become curves after the PD controller is added, which indicates a smoother transaction of the motor rotation speeds. This means the movement of the arm is more fluent than the system without the controller, and since there are no sudden changes in torque, less wear to the motor and every part in contact. A detailed error analysis is in the following section. So, it can be concluded that the primary adaptive controller, which is the PD controller, is effective for this robotic manipulator.

5.2.3 Error analysis

In order the illustrate the effect of the PD controller, an error analysis is required. In this case, the angular velocity of the joint of the end effector matters, which is joint 7 in ADAMS.
Here presents the result extracted from the two software:

	A	B
1	ADAMS	MATLAB
2	-3.88E-14	1.95E-13
3	-2.13E-05	-1.4E-05
4	-4.26E-05	-2.8E-05
5	-6.38E-05	-4.3E-05
6	-8.51E-05	-5.7E-05
7	-1.06E-04	-7.1E-05
8	-1.28E-04	-8.5E-05
9	-1.49E-04	-9.9E-05
10	-1.70E-04	-0.00011
11	-1.92E-04	-0.00013
12	-2.13E-04	-0.00014
13	-2.34E-04	-0.00016
14	-2.56E-04	-0.00017
15	-2.77E-04	-0.00018
16	-2.98E-04	-0.0002
17	-3.20E-04	-0.00021
18	-3.41E-04	-0.00023
19	-3.63E-04	-0.00024
20	-3.84E-04	-0.00026
21	-4.05E-04	-0.00027
22	-4.27E-04	-0.00028
23	-4.48E-04	-0.0003
24	-4.70E-04	-0.00031
25	-4.91E-04	-0.00033
26	-5.12E-04	-0.00034
27	-5.34E-04	-0.00036
28	-5.55E-04	-0.00037

Figure 5. 49Velocity output of joint7 from both software

There are 25000 lines of data for the simulation since the step was set to 25000, so it will not be displayed here. The diagram is drawn from these two sets of results, as shown in diagram (5.50)

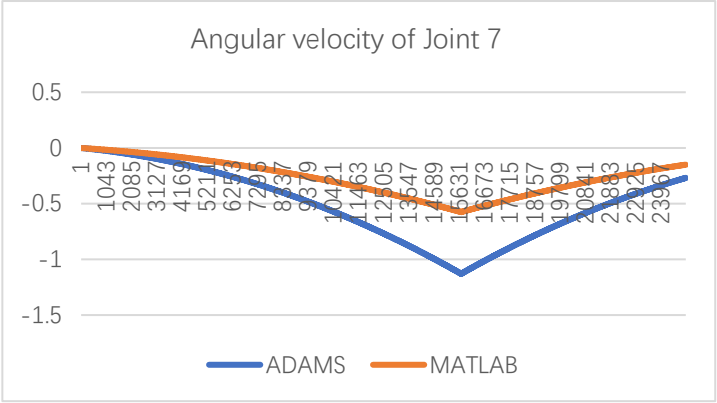

Figure 5. 50Angular velocity of joint 7 in one time cycle

And the error from these results can be plotted as follow:

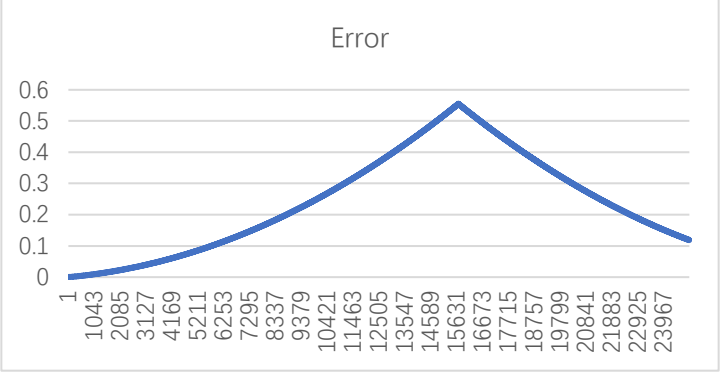

Figure 5. 51Error

This shows the error in the first 50 seconds of movement, which contains the two peaks and turning points of the joint velocity, and the maximum error also happens at the peak of the diagram, which is 0.551 deg/sec.
Another run for 150 seconds provides the data as shown below:

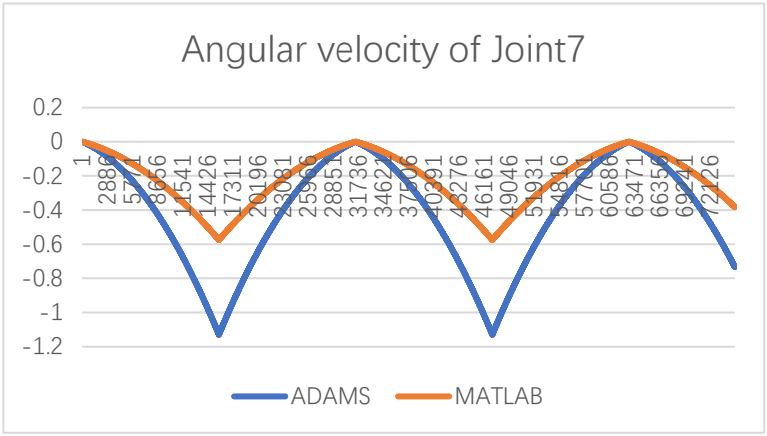

Figure 5. 52Velocity output of joint 7 in multiple time cycles

And the error between two sets of results:

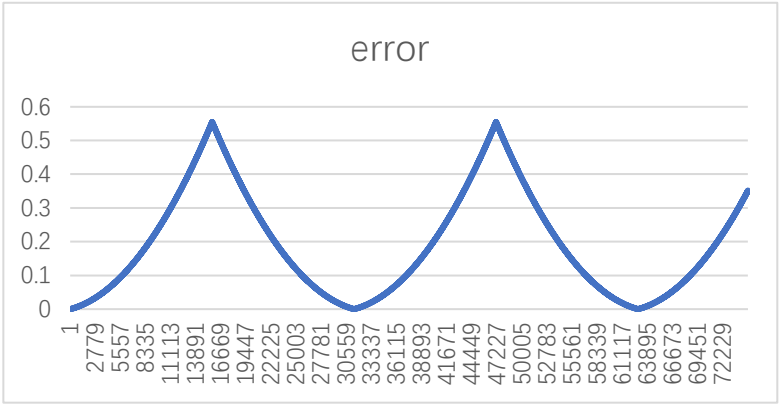

Figure 5. 53Error

The error of the results cannot be called perfect. Whether it is within the tolerable range needs further study to be verified. Compare the two sets of data, and the error repeatedly appears with a fixed cycle. The peak of error also appears at the turning point of the angle of joint 7. This could be because when the joint is changing the rotation direction, the inertia of motion reaches the maximum at the turning point and thus leads to the peak of error since the PD controller cannot handle the interference with extra inertia on the joint. It can be concluded that with the interference of motor added at each joint, the piled error reaches the peak when the joint connecting manipulator and welding head is changing rotation direction.
However, it should be noticed that the controller applied to the

system is relatively straightforward at this stage. The PD controller does have some higher variations like the PID controller that can cope with complicated environmental influences thus, once applied, it should come up with some better results with less error.

5.2.4 Direct controller design from SolidWorks to MATLAB

There is another method that enables a direct import from the SolidWorks model into MATLAB files. It involves an establishment of a new file. The target file type is called *".urdf"*, **Unified Robot Description Format**, as is shown in the name. This file is created to describe the robot system and can be recognized by most of the developing software, like SolidWorks, ADAMS, and MATLAB. However, SolidWorks does not have the function of setting 3D models into *.urdf* files. This is where a third-party add-in is involved:

SolidWorks to URDF Exporter

This is a third-party add-in developed by ROS.org[44], which enables the user to impose unique features for robotics onto SolidWorks models. Generally, it takes four steps to finish building a *.urdf* file by this add-in. The first step is to set up the configuration of the robotic system, including the joints, links, and tree. The second step is to establish the joint properties like joint types (revolute or prismatic) and movement limits. The third step is to set up the link properties. It includes the original position, moment of inertia, and material of the links. The final step is to generate the file. The *.urdf* file will also contain the mesh and texture of the robot, and this is the step in which all these features are generated.

With the above established D-H parameter of the robotic arm, the establishing process is shown as follow:

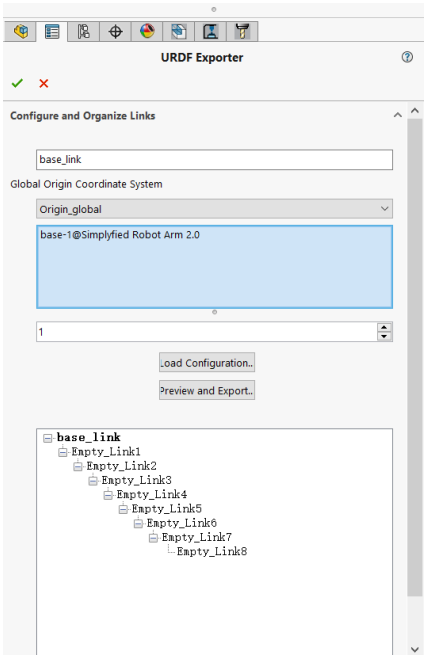

Figure 5. 54 Establish the joints and links of the arm

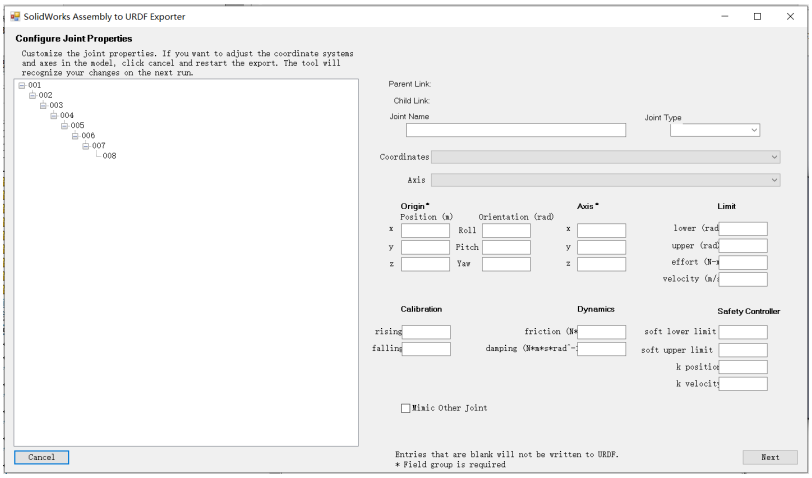

Figure 5. 55 Set physical properties of joints and links

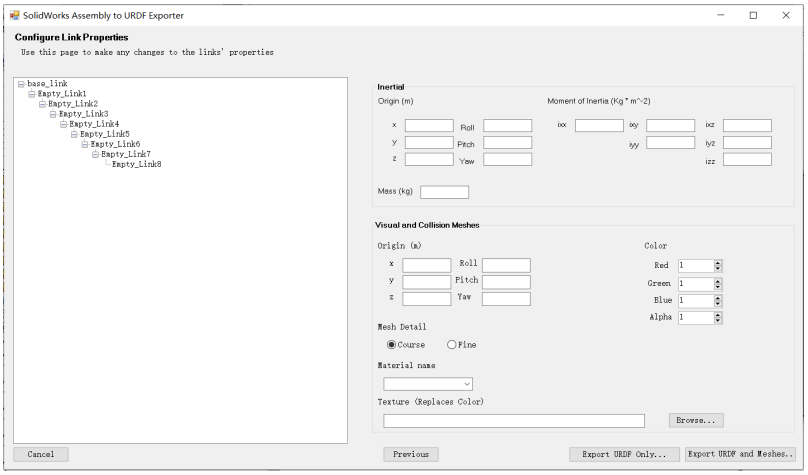

Figure 5. 56 Set boundaries of motion for joints

Eventually, the add-in will create several files that contain all features of the robotic system and the *.urdf* file can be recognized in most software. Take MATLAB as an example. The generated *.urdf* file can be open in MATLAB with Simulink. When open, the model can be plotted with an assigned coordinate system to each joint. In the Simulink interface, a rigid body tree will be shown so that the design of the control system and simulations can be carried out. By skipping the ADAMS simulation, this method provides better flexibility in establishing the motion function and translation matrix of the system, which enables the simulation to be carried out under more complicated and unique situations.

6 Discussion and future work

In this research, a control system for a specific robotic manipulator for the shipbuilding welding procedure was designed and tested. First, the background of the project was reviewed, including current types of welding robots, different designs of the platform and the possible gait they would apply and welding techniques that are currently applied in the industry. Next, the foundation of modelling and control of robotic manipulators was introduced. There are several key features for modelling, like D-H parameters and the Jacobian. The dynamic of the system was also established for the design of the control system. Then, key features mentioned above for the specific robotic manipulator were calculated, the system was then taken into ADAMS for motion simulation to determine the velocity and power output of each motor. Eventually, the model was imported into MATLAB Simulink. In the Simulink, a PD controller was added to the system and simulation was run upon the system. It can be concluded from the result that for the current stage where a specific design was not established, a control system with a basic adaptive controller (in this case, a PD controller) is sufficient for certain configurations of a robotic arm to carry out welding. With that being said, there are still several aspects that can require further study and improvement.

Modelling

In this study, a simple joint and link manipulator is used for basic features and simulation. As is mentioned in the previous section, the configuration comes from a design from another group of students on campus. The design is yet to be finished, not to mention to be manufactured. So, for the design of the robotic arm, there is a lot of work to be done. At this stage, the design does have a general idea of the configuration and shape of the arm, but the details need a lot of refining. The position of the fastener, the selection of the motor and bearings are just two among the various undefined details on the design and require further study.

Tool selection

For this project, the selected function is welding, and the type is SMAW, shielded metal arc welding. There are, however, many other types of welding techniques, each type requires its own welding head and moving speed, some even require a second head for feeding or protective gas injection. The various welding technique comes with more motion equation of the end effector.

To take a step further, the function of the tool can be more than welding. Other functions like fastening bolts or applying adhesive materials are also frequently required in the industry.

Design of controller

In order to come up with a general solution for the welding procedure, the manipulator should be able to deal with as many types of movement as possible. And to achieve this goal, two of the most obvious ways are either adding the degree of freedom to the arm to enable more precise movement or designing a more advanced control system. The robotic manipulator already has six-degree-of-freedom. Adding more degree-of-freedom would result in kinematical redundancy. The kinematically redundant arm is usually applied in a situation where the arm needs to operate across certain obstacles. Consider that the arm is already mounted on a hexagon walking platform, there is no need for the arm to bypass obstacles in most situations. Thus, for this manipulator, a more advanced control system can be designed to improve performance. During the research, the adaptive control is applied, and the PD controller is added to the system. A step more of the PD controller is the PID controller, **Proportion Integration Differentiation**. The extra integration term in the PID control equation enables a smaller gain factor ($K_p and K_v$ in this case, for example) but better performance in suppressing the environmental interfere and tracking the step-index input. As shown in the simulation result from MATLAB Simulink, the angular velocity of the last two joints still has much error. This could have resulted from a pile-up from previous errors on other joints or the incapability of a simple PD controller. The PID controller can be a great improvement in the performance of the manipulator.

Other factors

During this research, there are variables that were considered 0 or neglected. For example, the material of the arm was unified to aluminum, but in reality, it should be a combination of metal alloy and polymers. The actual material properties like the density and strength of the parts need to be considered separately. The friction of joints was also neglected, which actually can be the main contribution to the error of movement. These factors can be taken into consideration during future simulations for a better and more accurate result. With that being said, it still requires massive experiments and measurement on the real manipulator.

7 Conclusion

In this thesis, the basic theory of modelling and controlling the robotic system is introduced, key factors including D-H parameter, Jacobian, adaptive control and PD controller are introduced. A simplified robotic arm aimed for welding in the shipbuilding industry is discussed. The controller of this particular arm is designed and simulated for feasibility. After that, future work and possible improvement of the control system and the arm itself are presented. It can be concluded that under the current PD controller, the arm can fulfil the requirements of the basic welding procedure. For a more complicated application, a better controller is needed.

During the research, several objectives were established, during the research, the required objectives were reached step by step: A mathematical model and a physical model were established, the relative elements for later simulation were obtained, including the joint configurations, D-H parameters, moving pattern of the base and end effector. The MSC ADMAS was applied for a kinematic simulation of the arm with established parameters. The required joint angle and angular velocity for the movement were obtained through the kinematic simulation. The obtained parameters were then put into MATLAB Simulink. With a primary PD control system applied, a clear reduce in end effector joint angular speed was observed. With the same base moving gait and end effector moving speed, the angular velocity variation for joint 7 was reduce to 0.5 rad/s, which met the preset objective. For further study, it is recommended to apply a PID controller for higher precision. Also, different tool selection for the end effector and different joint configuration need to be discussed.

8 Reference

[1] D. Lee, "Robots in the shipbuilding industry," *Robotics and computer-integrated manufacturing,* vol. 30, no. 5, pp. 442-450, 2014, doi: 10.1016/j.rcim.2014.02.002.

[2] N. Ku *et al.*, "Development of a mobile welding robot for double-hull structures in shipbuilding," *Journal of Marine Science and Technology,* vol. 15, pp. 374-385, 2010, doi: 10.1007/s00773-010-0099-5.

[3] P. Gonzalez de Santos, M. A. Armada, and M. A. Jimenez, "Ship building with ROWER," *IEEE robotics & automation magazine,* vol. 7, no. 4, pp. 35-43, 2000, doi: 10.1109/100.894031.

[4] T. Mori, K. Miyawaki, N. Shinohara, and Y. Saito, "NC painting robot for narrow areas," *Advanced Robotics,* vol. 15, no. 3, pp. 323-326, 2001, doi: 10.1163/156855301300235850.

[5] J. Kim *et al.*, "Rail Running Mobile Welding Robot 'RRX3' for Double Hull Ship Structure," *IFAC Proceedings Volumes,* vol. 41, no. 2, pp. 4292-4297, 2008, doi: 10.3182/20080706-5-kr-1001.00722.

[6] S. Bieller, C. Müller, S. Lampe, and N. Kutzbach. "Standardization." https://ifr.org/standardisation (accessed.

[7] B. Brumson. "Scara vs. Cartesian Robots: Selecting the Right Type for Your Applications." (accessed.

[8] F. Neri and E. Mininno, "Memetic Compact Differential Evolution for Cartesian Robot Control," *IEEE Computational Intelligence Magazine,* vol. 5, no. 2, pp. 54-65, 2010, doi: 10.1109/mci.2010.936305.

[9] M. Shariatee, A. Akbarzadeh, A. Mousavi, and S. Alimardani, "Design of an economical SCARA robot for industrial applications," 2014: IEEE, doi: 10.1109/icrom.2014.6990957.

[10] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and Control of Articulated Robot Arms with Redundancy," *IFAC Proceedings Volumes,* vol. 14, no. 2, pp. 1927-1932, 1981, doi: 10.1016/s1474-6670(17)63754-6.

[11] L. Rey and R. Clavel, "The Delta Parallel Robot," Springer London, 1999, pp. 401-417.

[12] C. Mueller. "WR Industrial Robots 2019 Chapter 1." (accessed.

[13] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of applied mechanics,* vol. 22, no. 2, pp. 215-221, 1955, doi: 10.1115/1.4011045

[14] G. Gao, G. Sun, J. Na, Y. Guo, and X. Wu, "Structural parameter identification for 6 DOF industrial robots," *Mechanical systems and signal processing,* vol. 113, pp. 145-155, 2018, doi: 10.1016/j.ymssp.2017.08.011.

[15] J.-D. Sun, G.-Z. Cao, W.-B. Li, Y.-X. Liang, and S.-D. Huang, "Analytical inverse kinematic solution using the D-H method for a 6-DOF robot," 2017: IEEE, doi: 10.1109/urai.2017.7992807.

[16] D. E. Orin and W. W. Schrader, "Efficient Computation of the Jacobian for Robot Manipulators," *The International Journal of Robotics Research,* vol. 3, no. 4, pp.

66-75, 1984, doi: 10.1177/027836498400300404.
[17] M. Renaud, "Geometric and kinematic models of a robot manipulator: Calculation of the Jacobian matrix and its inverse," *Proc. 11th int. Symp. Industrial Robot,* 1981.
[18] R. Featherstone and D. Orin, "Robot dynamics: equations and algorithms," IEEE, doi: 10.1109/robot.2000.844153.
[19] K. J. Åström, "Theory and applications of adaptive control—A survey," *Automatica,* vol. 19, no. 5, pp. 471-486, 1983, doi: 10.1016/0005-1098(83)90002-X
info:doi/10.1016/0005-1098(83)90002-X.
[20] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive Jacobian Tracking Control of Robots With Uncertainties in Kinematic, Dynamic and Actuator Models," *IEEE Transactions on Automatic Control,* vol. 51, no. 6, pp. 1024-1029, 2006, doi: 10.1109/tac.2006.876943.
[21] J. Iqbal, "Modeling and Analysis of a 6 DOF Robotic Arm Manipulator," vol. 3, R. u. Islam and H. Khan, Eds., ed. Canadian Journal on Electrical and Electronics Engineering, 2012, pp. 300-306.
[22] I. Alkahla, "Sustainability assessment of shielded metal arc welding (SMAW) process," vol. 244, S. Pervaiz, Ed., ed. IOP Conference Series: Materials Science and Engineering, 2017.
[23] N. R. W., "Weld Thermal Efficiency of the GTAW Process," J. C. E., Ed., ed. Welding journal, 1975.
[24] H.-y. Shen, J. Wu, T. Lin, and S.-b. Chen, "Arc welding robot system with seam tracking and weld pool control based on passive vision," *International journal of advanced manufacturing technology,* vol. 39, no. 7, pp. 669-678, 2008, doi: 10.1007/s00170-007-1257-8.
[25] F. Lu, S. Yao, S. Lou, and Y. Li, "Modeling and finite element analysis on GTAW arc and weld pool," *Computational Materials Science,* vol. 29, no. 3, pp. 371-378, 2004, doi: 10.1016/j.commatsci.2003.10.009.
[26] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems,* vol. 16, no. 2, p. 172988141983959, 2019, doi: 10.1177/1729881419839596.
[27] F. Tedeschi and G. Carbone, "Design Issues for Hexapod Walking Robots," *Robotics,* vol. 3, no. 2, pp. 181-206, 2014, doi: 10.3390/robotics3020181.
[28] J. Bares *et al.*, "Ambler: an autonomous rover for planetary exploration," *Computer,* vol. 22, no. 6, pp. 18-26, 1989, doi: 10.1109/2.30717.
[29] T. Owen, "Machines That Walk: The Adaptive Suspension Vehicle by Shin-Min Song and Kenneth J. Waldron, The MIT Press, Massachusetts1989, 314pages incl. index" *Robotica,* vol. 7, no. 4, pp. 368-368, 1989, doi: 10.1017/S0263574700006937.
[30] F. Pfeiffer, J. Eltze, and H. J. Weidemann, "The Tum-Walking Machine," *Intelligent Automation & Soft Computing,* vol. 1, no. 3, pp. 307-323, 1995, doi: 10.1080/10798587.1995.10750637.
[31] F. Delcomyn and M. E. Nelson, "Architectures for a biomimetic hexapod robot," *Robotics and Autonomous Systems,* vol. 30, no. 1-2, pp. 5-15, 2000, doi:

10.1016/s0921-8890(99)00062-7.

[32] M. R. Fielding, R. Dunlop, and C. J. Damaren, "Hamlet: force/position controlled hexapod walker - design and systems," IEEE, doi: 10.1109/cca.2001.973998.

[33] A. Roennau, G. Heppner, L. Pfotzer, and R. Dillmann, "LAURON V: OPTIMIZED LEG CONFIGURATION FOR THE DESIGN OF A BIO-INSPIRED WALKING ROBOT," *Nature-Inspired Mobile Robotics,* pp. 563-570, 2013, doi: 10.1142/9789814525534_0071

[34] C. Georgiades *et al.*, "AQUA: an aquatic walking robot," IEEE, doi: 10.1109/iros.2004.1389962.

[35] M. Oku, "Development of Hydraulically Actuated Hexapod Robot COMET-IV-The 1st Report : System Design and Configuration," *Proceedings of the 2007 JSME Conference on Robotics and Mechatronics,* 2007.

[36] M. Denton. "Mantis homepage." http://www.mantisrobot.com/ (accessed.

[37] BostonDynamics. "Spot|Boston Dynamics." https://www.bostondynamics.com/spot (accessed.

[38] H. Adachi, N. Koyachi, T. Arai, A. Shimiza, and Y. Nogami, "Mechanism and control of a leg-wheel hybrid mobile robot," IEEE, doi: 10.1109/iros.1999.811738.

[39] Y. Li, "Dynamic Simulation Analyses of a Six-Leg-Wheel Hybrid Mobile Robot under Uneven Terrains," 2010: IEEE, doi: 10.1109/icinis.2010.176.

[40] J. P. Barreto, A. Trigo, P. Menezes, J. Dias, and A. T. De Almeida, "FED-the free body diagram method. Kinematic and dynamic modeling of a six leg robot," IEEE, doi: 10.1109/amc.1998.743574.

[41] E. Celaya and J. M. Porta, "Control of a six-legged robot walking on abrupt terrain," IEEE, doi: 10.1109/robot.1996.506575

[42] L. Donghun *et al.*, "Development and application of a novel rail runner mechanism for double hull structures of ships," ed: IEEE, 2008, pp. 3985-3991.

[43] "Robot Modeling and Control," *Industrial Robot: An International Journal,* vol. 33, no. 5, pp. 403-403, 2021/09/15 2006, doi: 10.1108/ir.2006.33.5.403.1.

[44] S. Brawner. "SolidWorks to URDF Exporter." http://wiki.ros.org/sw_urdf_exporter (accessed.

9 Appendix

9.1 Appendix 1

The DH Parameter calculation for the Stanford arm
The parameter assigned for the robotic arm is shown below:

link	d_i	a_i	α_i	θ_i
1	0	0	-90	θ_i
2	d_i	0	+90	θ_i
3	d_i	0	0	0
4	0	0	-90	θ_i
5	0	0	+90	θ_i
6	d_i	0	0	θ_i

Table 10. 1

For each joint, the transfer matrixes are calculated:

$$A_1 = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.1)$$

$$A_2 = \begin{pmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.2)$$

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.3)$$

$$A_4 = \begin{pmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.4)$$

$$A_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.5)$$

$$A_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.6)$$

Thus the transfer matrix for the end effector can be calculated as:

$$T_6^0 = A_1 \cdots A_6 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.7)$$

And the elements of the matrix can be interpreted as:

$r_{11} = c_1[c_2c_4c_5c_6 - s_4s_6 - s_2s_5c_6] - s_1s_4c_5c_6 + c_4s_6$
$r_{21} = s_1[c_2c_4c_5c_6 - s_4s_6 - s_2s_5c_6] + c_1s_4c_5c_6 + c_4s_6$
$r_{31} = -s_2(c_4c_5c_6 - s_4s_6) - c_2s_5c_6$
$r_{12} = c_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] - s_1(-s_4c_5s_6 + c_4c_6)$
$r_{22} = -s_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] + c_1(-s_4c_5s_6 + c_4c_6)$
$r_{32} = s_2(c_4c_5s_6 + s_4c_6) + c_2s_5s_6$
$r_{13} = c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5$
$r_{23} = s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5$
$r_{33} = -s_2c_4s_5 + c_2c_5$
$d_x = c_1s_2d_3 - s_1d_2 + d_6(c_1c_2c_4s_5 + c_1c_5s_2 - s_1s_4s_5)$
$d_y = s_1s_2d_3 + c_1d_2 + d_6(c_1s_4s_5 + c_2c_4s_1s_5 + c_5s_1s_2)$
$d_z = c_2d_3 + d_6(c_2c_5 - c_4s_2s_5)$

9.2 Appendix 2

The Newton-Euler method

In this section, a different approach for pursuing the dynamic equation for the system is introduced. Both methods end up with the same equation, the difference lies in the analytical perspective they use. The Euler-Lagrange equation tends to consider the system as a whole, each element in the equation presents a summed dynamic feature of the whole system. But in the Newton-Euler method, the problem is analyzed for each link and joint of the arm, and they are eventually combined through forward-backwards recursion.

In classical mechanics, there are several conclusions that can be applied for solving the problem:

1. For every force, there is an equal and opposite reaction force
2. The velocity of change of linear momentum is equal to the joint force imposed on the object.
3. The velocity of change of angular momentum equals the join torque on the objective.

To apply the second and third conclusion on the movement of a certain object, it can be written in the equation that:

$$ma = f \quad (9.8)$$

$$\frac{d(I_0\omega_0)}{d_t} = \tau_0 \quad (9.9)$$

In which I_0 is the moment of inertia for the inertia frame of reference of the object and τ_0 is the join torque.

The moment of inertia can be calculated as:

$$I_0 = RIR^T \quad (9.10)$$

Where R is the transfer matrix, and I is the moment of inertia for the local coordinate system. This indicates that the moment of inertia for the inertia frame of reference is not a constant value function to the time. To overcome this problem, the torque can be calculated from the local coordinate system:

$$l\dot{\omega} + \omega \times (I\omega) = \tau \quad (9.11)$$

The calculation process is shown below:

$$\dot{R}R^T = S(\omega_0) \quad (9.12)$$

$$\omega_0 = R\omega, \omega = R^T\omega_0 \quad (9.13)$$

So the angular momentum can be expressed as:

$$h = I_0\omega_0 = RIR^TR\omega = RI\omega \quad (9.14)$$

$$\dot{h} = \dot{R}I\omega + RI\dot{\omega} \quad (9.15)$$

$$\dot{R} = S(\omega_0)R \quad (9.16)$$

$$\dot{h} = S(\omega_0)RI\omega + RI\dot{\omega} \quad (9.17)$$

$$R^T\dot{h} = R^TS(\omega_0)RI\omega + I\dot{\omega} = S(R^T\omega_0)I\omega + I\dot{\omega} \quad (9.18)$$

$$= S(w)I\omega + l\dot{\omega} = \omega \times (I\omega) + I\dot{\omega} \quad (9.19)$$

Which is the same as equation (9.11).

Having calculated the join torque for single object, the robotic system can also be analyzed. The first step is to set up a coordinate system for each link and one inertia reference system 0. Then several variables of the system need to be established:

$a_{c,i} = acceleration\ of\ the\ mass\ center\ for\ link\ i$

$a_{e,i} = acceleration\ of\ the\ end\ for\ link\ i$

$\omega_i = angular\ velocity\ of\ coordinate\ system\ i\ to\ reference\ syetem\ 0$

$\alpha_i = angular\ acceleration\ of\ system\ i\ to\ reference\ system\ 0$

$g_i = \ gravitational\ acceleration$

$f_i = force\ impose\ from\ link\ i-1\ to\ link\ i$

$\tau_i = torque\ imposed\ from\ link\ i-1\ to\ link\ i$

$R_{i+1}^{i} = transfer\ matrix\ from\ system\ i+1\ to\ system\ i$

$m_i = mass\ of\ link\ i$

$I_i = moment\ of\ inertai\ of\ link\ i$

$r_{i,ci} = the\ vector\ of\ point\ o_i\ to\ mass\ center\ of\ link\ i$

$r_{i+1,c_i} = the\ vector\ of\ point\ o_1\ to\ the\ mass\ center\ of\ link\ i$

$r_{i,i+1} = the\ vector\ from\ o_i\ to\ o_{i+1}$

With all the elements established, write the force equilibrium function of link i.

$$f_i - R_{i+1}^{i} f_{i+1} + m_i g_i = m_i a_{c,i} \quad (9.20)$$

And the torque equilibrium function:

$$\tau_i - R_{i+1}^{i}\tau_{i+1} + f_i \times r_{i,c_i} - \left(R_{i+1}^{i} f_i\right) \times r_{i+1,Ci} = I\dot{\omega}_i + \omega_i \times (I_i\omega_i) \quad (9.21)$$

The core of the Newton-Euler method is to acquire the solution of the above two functions, which correspond to a set of global coordinates and its first and second-order derivative.

Before the beginning of the calculation, set a boundary condition of $f_{n+1} = 0, \tau_{n+1} = 0$, which is to state that n is the maximum number of links and there is no link after the n^{th} link.

From previous kinematic analysis, the angular velocity of the coordinate system i can be expressed as:

$$\omega^{o}{}_{i} = \omega^{o}{}_{i-1} + z_{i-1}\dot{q}_i \quad (9.22)$$

$$\omega_i = \left(R_i^{i-1}\right)^T \omega_{i-1} + b_i\dot{q}_i \quad (9.23)$$

$$b_i = (R_i^0)^T z_{i-1} \quad (9.24)$$

As for the angular acceleration:

$$\alpha_i = (R_i^0)^T \dot{\omega}_i \quad (9.25)$$

$$\dot{\omega^{o}}{}_{i} = \dot{\omega^{o}}{}_{i-1} + z_{i-1}\ddot{q}_i + \omega^{o}{}_{i} \times z_{i-1}\dot{q}_i \quad (9.26)$$

Put the above equation into coordinate i

$$\alpha_i = \left(R_i^{i-1}\right)^T \alpha_{i-1} + b_i \ddot{q}_i + \omega_i \times b_i \dot{q}_i \quad (9.27)$$

Having determined the angular velocity and acceleration, the next step is the linear velocity and acceleration:

$$v^o{}_{c,i} = v^o{}_{e,i-1} + v^o{}_i \times r^o{}_{i \cdot c_i} \quad (9.28)$$

$$a^o{}_{c,i} = a^o{}_{e,i-1} + \dot{\omega^o}{}_i \times r^o{}_{i,c_i} + \omega^o{}_i \times \left(\omega^o{}_i \times r^o{}_{i,ci}\right) \quad (9.29)$$

$$a_{c,i} = (R_i^0)^T a_{c,i}^{(0)} \quad (9.30)$$

$$a_{c,i} = \left(R_i^{i-1}\right)^T a_{e,i-1} + \dot{\omega}_i \times r_{i,c_i} + \omega_i \times \left(\omega_i \times r_{i,ci}\right) \quad (9.31)$$

$$a_{e,i} = \left(R_i^{i-1}\right)^T a_{e,i-1} + \dot{\omega}_i \times r_{i,i+1} + \omega_i \times \left(\omega_i \times r_{i,i+1}\right) \quad (9.32)$$

So, with all the velocities and accelerations established, the Newton-Euler method can be used:

From the initial condition:

$$\omega_0 = 0, \alpha_0 = 0, a_{c,0} = 0, a_{e,0} = 0$$

Calculate the angular velocity, angular acceleration, endpoint linear acceleration and mass center linear acceleration (in that order) from $i = 1$ to $i = n$

From the ending condition:

$$f_{n+1} = 0, \tau_{n+1} = 0$$

To calculate the join force and torque of each link from $i = n$ to $i = 1$.

Thus, the dynamic feature of the system is analyzed.