DYNAMIC NEURAL NETWORKS FOR PARALLEL STEWART PLATFORMS

MOHAMMED AQUIL MIRZA

PhD

The Hong Kong Polytechnic University

2019

THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF COMPUTING

Dynamic Neural Networks for Parallel Stewart Platforms

MOHAMMED Aquil Mirza

A Thesis Submitted in Partial Fulfilment of

the Requirements for the Degree of

Doctor of Philosophy

Mar 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signature)

Mohammed Aquil Mirza (Name of Student)

ABSTRACT

Redundancy resolution is a critical problem in the control of parallel Stewart platform. The redundancy endows us with extra design degree to improve system performance. In this thesis, the kinematic control problem of Stewart platforms is formulated to a constrained quadratic programming. The KKT conditions of the problem is obtained by considering the problem in its dual space, and then a dynamic neural network is designed to solve the optimization problem recurrently. Theoretical analysis reveals the global convergence of the proposed neural network to the optimal solution in terms of the defined criteria. Simulation results verifies the effectiveness in the tracking control of the Stewart platform for dynamic motions.

Redundancy resolution of parallel manipulators is widely studied and have brought many challenges in the control of robotic manipulators. The dual neural network, which is categorized under the recurrent neural networks inherits parallel processing capabilities, are widely investigated for the control of serial manipulators in past decades and has been extended to the control of parallel Stewart platforms in our previous works. However, conventional dual neural network solutions for redundancy resolution requires prior knowledge of the robot, which may not be accessible accurately in real time applications. In this thesis, we establish a model-free dual neural network to control the end-effector of a Stewart platform for the tracking of a desired spacial trajectory, at the same time as learning the unknown time-varying parameters. The proposed model is purely data driven. It does not rely on the system parameters as apriori and provides a new solution for stabilization of the self motion of Stewart platforms. Theoretical analysis and results show that we can achieve a globally convergent neural model in this thesis. It is also shown to be optimal under the model free criterion. In this thesis, we carried out numerical simulations which highlight and

illustrate relateable performance capability in terms of model-free optimization. Simulation results provided, verify the tracking control of the end effector while controlling the dynamic motion of the Stewart platform.

Keywords: Stewart platform, kinematic redundancy, recurrent neural networks, constrained quadratic programming, redundancy resolution, kinematic control.

PUBLICATIONS

Books

1. Shuai Li, Long Jin, **Mohammed Aquil Mirza**, "Kinematic Control of Redundant Robot Arms Using Neural Networks", in *IEEE John Wiley Sons, Hoboken, NJ, Inc., 2019*.

Journals

1. **Mohammed Aquil Mirza**, Shuai Li, Long Jin, "Simultaneous learning and control of parallel Stewart platforms with unknown parameters", Neurocomputing, Volume 266, 2017.
2. **Mohammed Aquil Mirza**, Shuai Li, "Dynamic Neural Networks for Kinematic Redundancy Resolution of Parallel Stewart Platforms.", IEEE Transactions on Cybernetics, 2016.

Seed Funds

1. **PI Mohammed Aquil Mirza**, "Hypnos", in *PolyU Lean Launchpad 2019* (Institute of Entrepreneurship), The Hong Kong Polytechnic University, Mar 2019.
2. **PI Mohammed Aquil Mirza**, "A solution for automatic physical parameter measurement of industrial manipulators", *Creative Software Competition*, Tamkang University, Sep 2018.
3. **PI Mohammed Aquil Mirza**, "Effect Node Modelling", in *PolyU Microfund 2018* (Institute of Entrepreneurship), The Hong Kong Polytechnic University, May 2018.

ACKNOWLEDGEMENTS

This thesis has allowed me to explore the interesting areas of robotics. I wish to thank the Robotics and Intelligent Systems research group of Dr. Li Shuai. They have enabled me to study the most interesting fields I have experienced in my academic endeavors.

I would also like to sincerely thank my Ph.D. supervisor Dr. Li Shuai for his primary supervision, for guidance, but also for allowing me the freedom to influence the choice of topic to a large degree. I am thankful to his inputs and comments, encouragement and trust, and large amount of valuable time spent with me.

I would like to thank the department of computing department for their support and kind assistance during my study period. I recognize that this thesis would not have been possible without the financial assistance from the Hong Kong Polytechnic University. I appreciate Dr. Li and the Department of Computing for offering me the travel grants for study tours and to attend the conferences.

Finally, I would like to express my gratitude to the center of my universe, my mother for her endless encouragement, support and patience during my Ph.D. candidature. My father, who passed away during my Ph.D. study with a dream to attend my graduation ceremony and who encouraged me throughout my life. My brother and my sister who always supported me and sacrificed much more than I expected, for me, to achieve my dreams. I am really grateful to them for their endless love, patience, understanding and support.

TABLE OF CONTENTS

CERTIFICATE OF ORIGINALITY iii

ABSTRACT iv

PUBLICATIONS vi

ACKNOWLEDGEMENTS vii

LIST OF FIGURES xi

LIST OF TABLES xiii

CHAPTER 1. INTRODUCTION 1
1.1 Related Work 1
1.1.1 History of industrial robots 1
1.2 Serial robots 2
1.2.1 Disadvantages 2
1.2.2 Benefits 3
1.3 Parallel robots 3
1.3.1 Disadvantages 4
1.3.2 Benefits 5
1.4 History of parallel robots 5
1.4.1 First patents 5
1.4.2 Gough, Stewart, or Cappel's platform? 6
1.5 Industrial use 9
1.5.1 Machine tools 9
1.5.2 Forming machines 11
1.5.3 Positioning 11
1.5.4 Delta robot 12
1.5.5 Tricept 13
1.5.6 Simulator 16
1.5.7 Cable robot 16
1.6 Thesis Organization 18

CHAPTER 2. DYNAMIC NEURAL NETWORKS FOR KINEMATIC REDUNDANCY RESOLUTION OF PARALLEL STEWART PLATFORMS 19
2.1 Introduction 19
2.2 Preliminaries 21
2.3 Robot Kinematics 22
2.3.1 Geometric Relation 22
2.3.2 Velocity Space Resolution 25
2.4 Problem Formulation as Constrained Optimization 27
2.5 Dynamic Neural Network Model 28
2.5.1 Neural Network Design 29
2.6 Theoretical Results 31
2.6.1 Optimality 31
2.6.2 Stability 33
2.7 Numerical Investigation 35
2.7.1 Simulation Setups 35
2.7.2 Circular Trajectory 36
2.7.3 Square Trajectory 39
2.8 Conclusion 42

CHAPTER 3. SIMULTANEOUS LEARNING AND CONTROL OF PARALLEL STEWART PLATFORMS WITH UNKNOWN PARAMETERS 45
3.1 Introduction 45
3.2 Kinematic Modeling of Stewart Platforms 49
3.2.1 Geometric Relation 49
3.2.2 Velocity Space Resolution 52
3.3 Recurrent Neural Network Design 53
3.3.1 Problem Formulation from an Optimization Perspective 53
3.3.2 Neural Network Dynamics 55
3.3.3 Stability 57
3.3.4 Optimality 58
3.4 Numerical Investigation 59
3.4.1 Setups 59
3.4.2 Circular Path 60
3.4.3 Square Trajectory 64
3.5 Conclusions 64

CHAPTER 4. STEWART PLATFORM CONSTRUCTION AND ALGORITHM VERIFICATION 66
4.1 Design a 6 DOF Rocking Chair Mechanism for Rocking Chair 66
4.1.1 Evaluation of existing solutions and selecting an appropriate variant 66
4.1.2 Building a Stewart Platform 68
4.1.3 Building Material 69
4.1.4 Mechanical Equipment 69
4.2 Creating a Solid Works Model 69
4.2.1 Selection of suitable design 71
4.2.2 Linear actuator 72
4.2.3 Joints 72
4.2.4 Platform and Base 73
4.2.5 Model Actuator 73
4.2.6 User position in the simulator 74
4.3 Background of the Invention 74
4.3.1 Open Challenges 74
4.3.2 Expected Outcomes 74
4.3.3 Market Demand 75
4.3.4 Our Invention Claims 75
4.4 Experimental setup discussion and results 77
4.4.1 Simulation Results 78
4.4.2 Loading rocking chair platform and studying reaction and capacity 83
4.5 Remarks 84

CHAPTER 5. CONCLUSION AND FUTURE WORK 87
5.1 Conclusion 87
5.2 Future Work 88

REFERENCES 90

LIST OF FIGURES

1.1 Open string 3
1.2 Closed string 4
1.3 Gwinnet's patent [5] 6
1.4 Pollard's varnishing robot [5] 6
1.5 Gough's octahedral hexapod [5] 7
1.6 MAST facility at Minnesota University [5] 8
1.7 Stewart's Simulator Design [5] 8
1.8 Cappel's octahedral hexapod [5] 9
1.9 Metrom German P1423 [3] 10
1.10 Metrom German PM1400 [3] 10
1.11 Fraunhofer HexaBend [67] 11
1.12 Hexapd Telescope [8] 12
1.13 Stratospheric Aerosol and Gas Experiment III [77] 13
1.14 The FlexPicker 340 [4] 14
1.15 Surgical-Scope [4] 15
1.16 Orion Delta 3D Printer [56] 15
1.17 Orion Delta 3D Printer [71] 16
1.18 Orion Delta 3D Printer [68] 17
1.19 Orion Delta 3D Printer [63] 17

2.1 A schematic of the Stewart platform. 23
2.2 Two layer neural network architecture 31
2.3 Tracking of a circular motion. (a). The trajectory of the end-effector; (b). Time history of the position tracking error. 37
2.4 The time evolution of the Stewart platform state variables in the case of circular motion tracking. 38
2.5 The time evolution of the neural network state variables in the case of circular motion tracking. 40
2.6 Tracking of a square motion. (a). The trajectory of the end-effector; (b). Time history of the position tracking error. 41
2.7 The time evolution of the Stewart platform state variables in the case of square motion tracking. 43

2.8 The time evolution of the neural network state variables in the case of square motion tracking. 44

3.1 The schematic of a Stewart platform. 49
3.2 Stewart platform geometric representation. The red triangle on the top and the red hexagon below it represent the moving top plate and the fixed base plate, respectively. 50
3.3 The control diagram of the Stewart robot using the proposed neural network. . 54
3.4 Architecture of the proposed neural network. 57
3.5 Tracking of a circular motion. 60
3.6 Stewart platform state variables at different time in case of circular motion tracking. 61
3.7 Tracking of a square motion. 62
3.8 Stewart platform state variables at different times in case of square motion tracking. 63

4.1 The mechanical construction of a rocking chair Skeleton. 68
4.2 The material and equipment for Stewart Platform. 70
4.3 Linear Transmotec Actuator 73
4.4 Mechanical rocking chairs 76
4.5 Market demand 76
4.6 Model pose 77
4.7 Sine wave signal as input model 79
4.8 The pitch acceleration produced by mechanical platform 79
4.9 The vertical acceleration produced by moving platform 79
4.10 The pitch acceleration produced by moving platform 80
4.11 The pitch acceleration produced by moving platform 80
4.12 The pitch acceleration produced by mechanical platform 81
4.13 The vertical acceleration produced by moving platform 81
4.14 The pitch acceleration produced by moving platform 82
4.15 The vertical acceleration produced by moving platform 82
4.16 The output results of load capacity comparison 83
4.17 The output results of load capacity comparison - 5HZ and 2mm 84
4.18 The output results of load capacity comparison - 7HZ and 3mm 85
4.19 The output and input results comparison in frequency domain 85

LIST OF TABLES

4.1 Top and Base Plate of Stewart Platform 71
4.2 The Leg of Stewart Platform 71
4.3 Power Supply 72

CHAPTER 1
INTRODUCTION

Today we live in a world that is constantly accelerating. This is also the case of replacing the human labor force with a machine. It has been developing very lately the use of industrial (automated) robots in all sectors industry, they can replace even more workers and increase productivity and efficiency of the production process.

This trend makes robot makers more efficient, more economical and produce safer equipment in a shorter time, which is very costly, financially, for when utilizing human employees. Solving these problems by using virtual models.

1.1 Related Work

1.1.1 History of industrial robots

The word robot was used for the first time by Karel Capek in his theater play Rosum Universal Robots (R.U.R) in 1920. Since then, this word expanded around the world and used to name mechanical, automated devices that replace human activity. Exact definition of the term robot does not exist because it is a very diverse and complex technical device.

According to International Organization for Standardization (ISO) 8373, an industrial robot is defined as: “automatically controlled, re-programmable, multipurpose handling machine, stationary or based on traction, intended for use in industrial automation.” [69]

Since the beginning of the word robot, people have been trying to construct devices that would imitate them. In 1927, a Telovex robot appeared in the New York Exhibition with a few basic moves. This example was followed by many other robots. They were always in the shape of a man and their construction was at the top of that level. After the Second World

War, when demand grew heavily above supply, it was born with the need for rapid production in large series. For this reason, the development first began with industrial robots. In 1956 American engineers George Devol and Joshep Engelberg began with the development of its robot Unimate 1900, which was already successful in 1961 and was put into operation at General Motors as a replacement for workers operating pressurized casting machines. In 1962, he was followed by a second model Versatran. Robots have proven themselves well and have licensed after patenting manufactured in developed countries around the world such as the United Kingdom, Sweden, and Japan. Mechanically these robots were in good condition, but their electronics and control systems were the cause of low-reliability [69], [73].

A more detailed breakdown and description PRaM (Industrial robots and manipulators) can be found in [69] and [73]. However, we will be particularly interested in the division according to kinematic structures on:

- Serial robots
- Parallel robots

1.2 Serial robots

Robots with serial kinematics are characterized by their similar structure of the human limb, are thus formed from series-ordered kinematic pairs that form an open-loop chain. The resulting movement then consists of translation or rotation of individual members where each the member moves independently, as can be seen on the open chain diagram fig. 1.1.

1.2.1 Disadvantages

Robots with serial kinematics have been in development for decades and their features are constantly improving. However, they encounter limits that cannot overcome for their basic design which is one of the main shortcomings of this type. The construction is such that

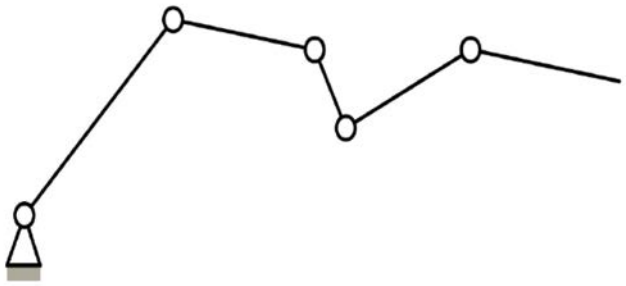


Figure 1.1: Open string

most of the robot parts are stressed on the bend and also that every previous member carries the weight of all the following members. To preserve the stiffness of the entire structure the model must be very robust. As a result of this, the moving mass is limited by axial dynamics, that is, speed and acceleration. Another unwanted feature is that the error in the end effector position is given by the sum errors of all previous members. Because of this, they are high in production accuracy requirements.

1.2.2 Benefits

The advantage of this type of construction is the large working area and the possibility of zooming the end effector to the desired object at different angles. Together with simple control and calibration, these features give serial kinematics high versatility of use. Approximately 90% of robots used in industry have a construct based on serial kinematics.

1.3 Parallel robots

Robots with parallel kinematics are, unlike serial ones, made with closed kinematic chains. In general, a parallel robot can be defined as follows: “It is closed kinematic chain mechanism whose end effector is connected to the base through several independent kinematic chains.” [60]

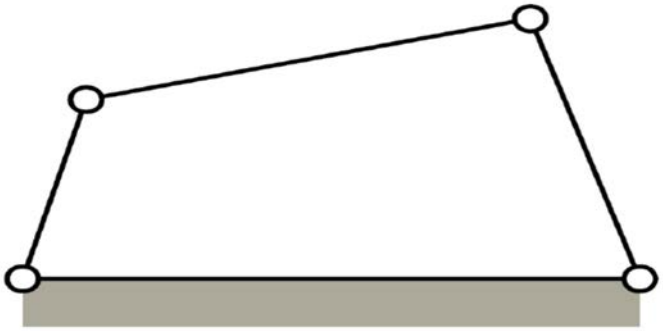


Figure 1.2: Closed string

This definition is not entirely accurate, as it includes, for example, redundant mechanisms that have a greater number of actuators than controlled degrees of freedom of the end effector. In Fig. 1.2 we can see a closed chain of parallel robot [60].

The following were introduced to define more precisely the term parallel robot characteristics listed in [6]:

- At least two strings support the end effector, each of which string contains at least one simple actuator (member allowing movement with one degree of freedom)
- The number of actuators is the same as the number of degrees of freedom of the end effector (platforms)
- If all actuators are blocked (motion is prevented), the number of degrees of freedom is zero

1.3.1 Disadvantages

Like any mechanism, a parallel robot has its drawbacks. The strings are stressed mainly on tension and pressure so that with the increasing length of the shoulders there is a risk of loss buckling stability. Long arm lengths combined with higher spindle speeds (machining center only) causes an inaccuracy of the end effector, such as due to thermal dilation of the shoulders. It is, therefore, necessary to monitor the change in shoulder length and compensate

for these errors in the robot control program. Another disadvantage is small workspace in proportion to the size of the device.

1.3.2 Benefits

Because the platform is supported by several arms it is a parallel robot to carry a heavier burden and achieve greater speed and acceleration than a serial robot at a total lower weight of the device. Also, the end-effector error is lower than the serial robot. Here the error is given by the average of errors in each shoulder. This type of device also has the advantage of having unified parts (all arms are identical), so their manufacture or repair is simpler.

1.4 History of parallel robots

The following chapter describes the history of parallel robots as outlined in [5]. The history of parallel robots goes further than expected at first glance. It is unknown who first coined the term parallel construction mechanism, but the first mention comes from 1645 English architect Christopher Wren. Though it was only theoretical problems associated with parallel structure. In the 19th century, the researchers L.A. Cauchy, H. Lebesgue, R. Bricard published work associated with a parallel mechanism.

1.4.1 First patents

In 1928, J. Gwinnet patented a movable platform. It was mounted on the spherical joint and the movement was to provide three pneumotors. As you can see Fig. 1.3, the device serves in the entertainment industry as a platform in the cinema. The patent was accepted in 1931, but it is not known whether this facility was ever built. Gwinnet was ahead of its time, and then the industry was not ready for such a complex mechanism.

The first industrial parallel robot is considered by Willard L. V. Pollard, which was patented in 1942. It was a robot that sprayed or applied color through the pistols. As can

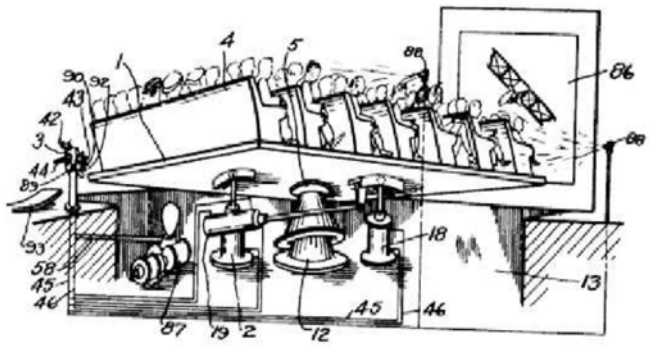


Figure 1.3: Gwinnet's patent [5]

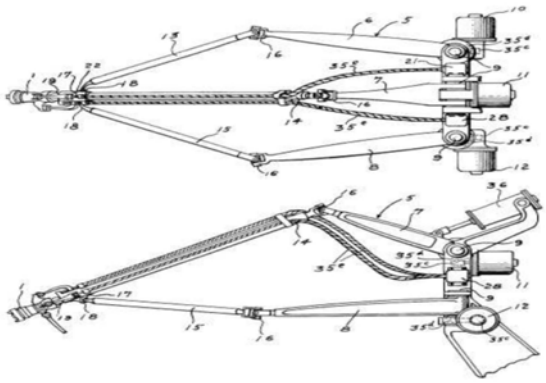


Figure 1.4: Pollard's varnishing robot [5]

be seen from Fig. 1.4, the robot had three arms that were the base is driven by head-space motors in the space and the other two motors associated with the head using Bowden for the shooting. This mechanism, therefore, had 5 degrees of freedom. However, this device has never been built.

1.4.2 Gough, Stewart, or Cappel's platform?

It was not until 1955 that a prototype of the octagonal hexapod, which can be seen in Fig 1.5. Dr. Eric Gough's first proposal came already in 1947. The device was designed to test drive for the company Dunlop Rubber Co. The design consists of a base and a platform that is based on six arms (hexapod). The arms are attached to the platform using ball joints and based on universal joints, according to their octagonal position (octahedron). The movement

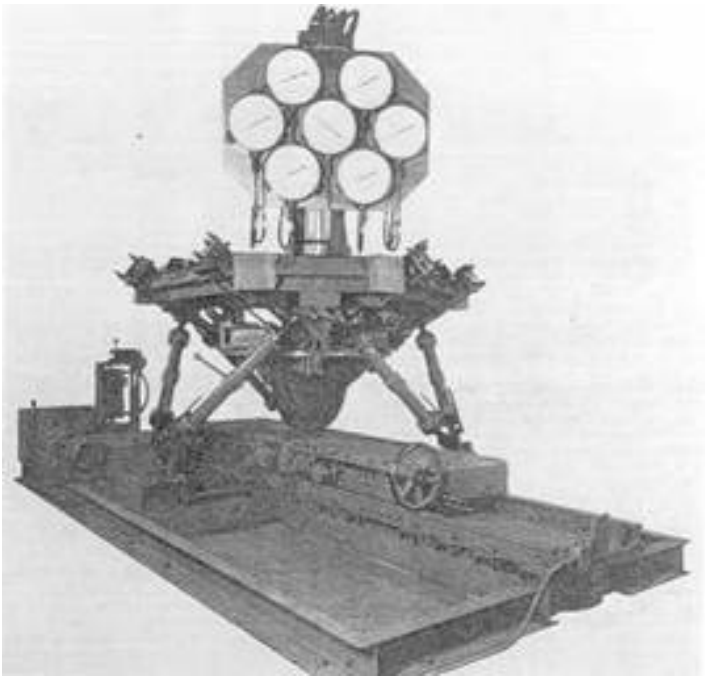


Figure 1.5: Gough's octahedral hexapod [5]

is ensured by a variable length shoulder. The device has a total of 6 degrees of freedom. Later the machine was reconstructed and controlled digitally. Until 2000, it was used in practice and is now exposed to the British National Museum.

However, as Dr. Gough research in his life, he worked on the idea of using six shoulders/legs which was inspired by MAST (Multi-Axis Simulation Table). This mechanism has three horizontal and three vertically positioned arms and used to simulate earthquakes or vibrations in general. Devices of this species are still used today. The MAST at Minesota University can be seen in Fig. 1.6.

In 1965, D. Stewart published his article describing the 6th gear of the freedom we see in Fig. 1.7. This is a parallel design proposal of kinematics as an air simulator whose platform has a triangular shape and is connected to three strings through ball joints. Each string consists of two arms of varying length which are connected to the base and rotationally connected to its binding. Gough suggested at the very end of his work, Stewart legs joining all 6 arms to the platform, creates a hexapod. The irony is that Gough's hexapod is now generally referred to as the Stewart platform.

Figure 1.6: MAST facility at Minnesota University [5]

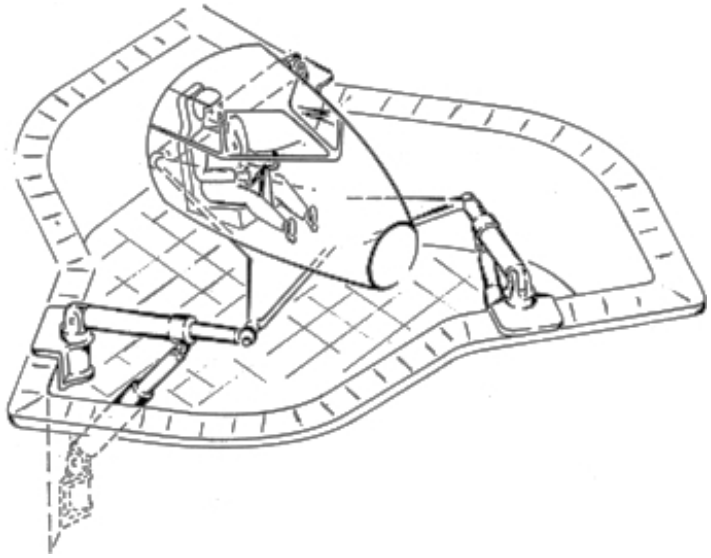


Figure 1.7: Stewart's Simulator Design [5]

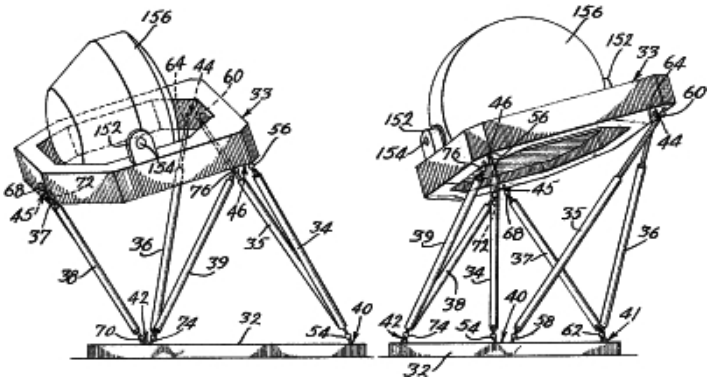


Figure 1.8: Cappel's octahedral hexapod [5]

For the third time in 1962, the American engineer Klaus Cappel was given the task of improving the MAST design to be suitable for use as a flight simulator. After unsuccessful experiments on seven-string devices (complicated control that led to the destruction of the machine) he came with the same octahedral such as Gough Fig. 1.8.

1.5 Industrial use

This chapter contains a selection of several parallel mechanism applications in industrial practice.

1.5.1 Machine tools

One of the largest manufacturers of machining centers with parallel kinematics is German company Metrom, which has developed a series of pentapod devices. Machine tool with 5 variable length strings where the spindle of the machine is the end effector. The strings are positioned so that the spindle can be rotated in one axis by up to 90°. So in combination with the rotary table, the center is capable of full machining from 5 hillsides. One of the latest models is Metro P1423 Fig. 1.9.

Metrom has also developed a compact and mobile version of its pentapod, PM1400 metric Fig. 1.10. It is designed to be simple and esay to be brought to a destination, to be folded when used for a short time, eg for machining large-dimensional components. The

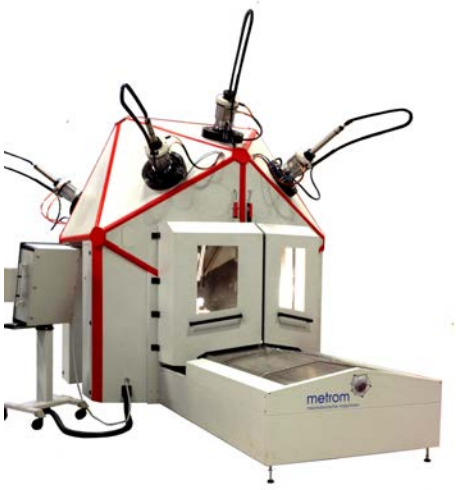


Figure 1.9: Metrom German P1423 [3]

Figure 1.10: Metrom German PM1400 [3]

Figure 1.11: Fraunhofer HexaBend [67]

design also allows the replacement of the terminal effector, eg for laser, grinding or painting [3], [85].

1.5.2 Forming machines

Parallel mechanisms have also found their way into shaping. At the Institute of Fraunhofer it has developed and patented HexaBend Fig. 1.11, which is a hydraulically controlled Stewart platform. You can move the platform shape, the preheated material emerging from the stationary part of the machine. The advantage of this equipment lies in its flexibility. For a given profile diameter, it is sufficient to bend of any radius of only one tool [67].

1.5.3 Positioning

The most common use of Stewart's platform in practice is device positioning in space that is used in various industries. In the examples that are described below, we will focus on space research.

The first device is the Hexapod-Telescope (HPT), which is seen in Fig. 1.12. The telescope, which developed between 1999 and 2006, is located in mountains in the north

Figure 1.12: Hexapd Telescope [8]

of Chile is unique in its construction. A primary mirror with a diameter of 1.5 meters, is placed on Stewart's platform with very precise positioning. Against the adverse effects of the dusty environment, its mechanics are stored in a carbon fiber-composite over-housing. At the primary, the mirror is the second Stewart platform at the end of which is the secondary optics [8].

The second is SAGE III - ISS (Stratospheric Aerosol and Gas Experiment III). It is a device capable of measuring the gas content in the upper one using a spectrometer atmosphere. This device is mainly focused on ozonosphere measurement. The decommissioning is planned for 2016, but it is not a separate satellite, like its predecessors, but a module prepared for assembly on ISS (International Space Station). Therefore, it is necessary to separate its positioning so that the Stewart platform will take care of as seen in Fig. 1.13 [77].

1.5.4 Delta robot

In the early 1980s, Dr. Reymond Clavel developed an idea to use the parallelogram on the assembly of a parallel 3-arm robot that has 3 translational degrees of freedom. This created a delta robot. In the original proposal, the four shoulders were used for transmission rotation between base and platform [4]. Delta robots have very quickly found their place in industrial

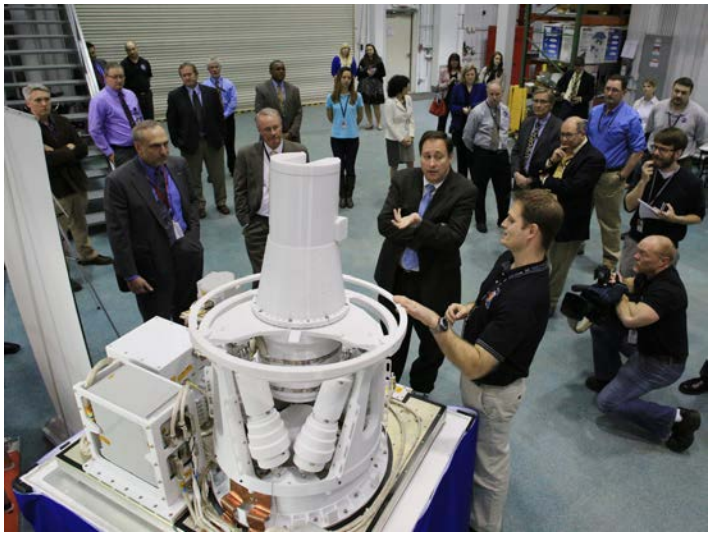


Figure 1.13: Stratospheric Aerosol and Gas Experiment III [77]

practice and today most commonly used on so-called Pick and Place operations (take and place), eg in the food industry. The leading manufacturer is Brown Boveri with a model FlexPicker and Fanuc with M1iA / 0.5SL as shown in Fig. 1.14.

Delta robot has other uses. For example, ISIS (Intelligent Surgical Instruments and Systems) introduced its SurgiScope Fig. 1.15. This is a delta robot that carries a 20 kg weighing microscope and other tools used in neurosurgery [4].

Parallel mechanisms have found their place in 3D printing as well. Lightly modified a delta robot that uses a linear feed instead of a rotary drive is used as a 3D printer. One of the first series-produced models to be used for home use is the Orion Delta 3D Printer Fig. 1.16 which has a linear displacement robot extended work area in height [56].

1.5.5 Tricept

PKMTricept is a Spanish company that develops and manufactures robots of the Tricept series. It is there are hybrids with hybrid kinematics, where they are on a parallel tripod (tripod) a serial string is provided to ensure rotation of the end effector. The whole mechanism has 5 or 6 degrees of freedom as required. Thanks to their large work area and good dynamics have found their way in many industries. Most of all, however, they are used in the automotive

Figure 1.14: The FlexPicker 340 [4]

Figure 1.15: Surgical-Scope [4]

Figure 1.16: Orion Delta 3D Printer [56]

Figure 1.17: Orion Delta 3D Printer [71]

industry where they are in the production line to support a wide range of operations such as machining, welding and grinding.

1.5.6 Simulator

As mentioned above, Stewart's platform was invented as an air simulator and is used for this purpose today. Its been already 70 years that Canadian Aviation Electronics (CAE) developed and operates simulators for both civilian and military purposes. Recently, FFS (Full-Flight Simulator) has been increasingly used. This simulator is capable of simulating all possible implications during the flight. In Fig. 1.17, there is a CEA simulator of Boeing 737 [71].

1.5.7 Cable robot

The IPAnem cable robot Fig. 1.18 is a special kind of parallel mechanism, which was developed at the Fraunhofer Institute INA. It uses the transformation to move the length of the arms formed from the cable wound on the drum. This robot, for now, found no industrial use and is still in development [68].

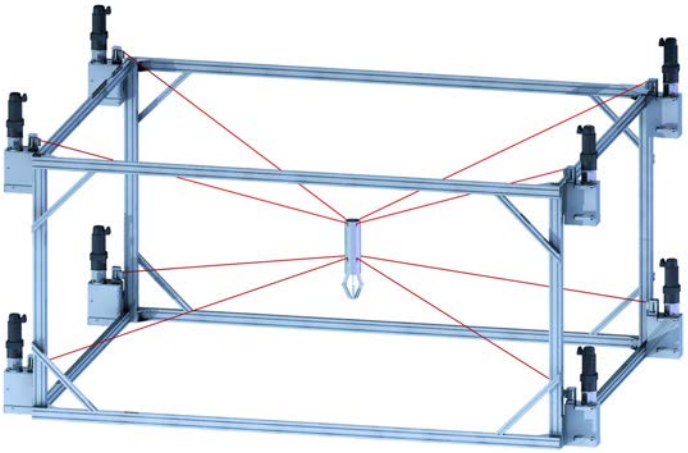


Figure 1.18: Orion Delta 3D Printer [68]

Figure 1.19: Orion Delta 3D Printer [63]

However, in 2016 China completed the FAST (Five-hundred-meter Aperture Spherical Telescope). It is the biggest and the most sensitive radio telescope in the world. As is already apparent from the title, the primary reflector will have a diameter of 500 meters and will consist of approximately 4400 triangular mirrors. Secondly, focusing optics will be placed in the cab to be positioned using a cable parallel mechanism consisting of 6 cables suspended at 6 towers. They will be evenly spaced in a circle with a diameter of 600m. Exact position of the secondary optics will be fine-tuned using the Stewart platform located in the cabin. Fig. 1.19 [63].

1.6 Thesis Organization

The rest of this thesis is organized as follows.

- In Chapter 2, the kinematic control problem of Stewart platforms is formulated to a constrained quadratic programming. The KKT conditions of the problem is obtained by considering the problem in its dual space, and then a dynamic neural network is designed to solve the optimization problem recurrently. Theoretical analysis reveals the global convergence of the proposed neural network to the optimal solution in terms of the defined criteria. Simulation results verifies the effectiveness in the tracking control of the Stewart platform for dynamic motions.

- In Chapter 3, the dual neural network, which is categorized under the recurrent neural network inherits parallel processing capabilities, are widely investigated for the control of serial manipulators in past decades and has been extended to the control of parallel Stewart platforms in our previous work. However, conventional dual neural network solutions for redundancy resolution requires prior knowledge of the robot, which may not accessible accurately in real applications. In this thesis, we establish a model-free dual neural network to control the end-effector of a Stewart platform for the tracking of a desired spacial trajectory, at the same time as learning the unknown time-varying parameters. The proposed model is purely data driven. It does not rely on the system parameters as a priori and provides a new solution for stabilization of the self motion of Stewart platforms. Theoretical analysis and results show that we can achieve a globally convergent neural model in this thesis.

- In Chapter 4, we propose for the first time the application of Stewart platform in the field of therapy to design and develop the rocking chair to improve the performance of mechanical rocking chairs and thereby offering convenient, portable and comfortable experience.

- In Chapter 5, we present conclusions and possible future directions of research arising from this work.

CHAPTER 2

DYNAMIC NEURAL NETWORKS FOR KINEMATIC REDUNDANCY RESOLUTION OF PARALLEL STEWART PLATFORMS

2.1 Introduction

Kinematically redundant manipulators are referred to those which have more degrees of freedom than acquired for the position and orientation. The redundancy of parallel manipulators can be utilized to overcome the obstacles, singularities [69], increasing workspace, improving dexterity and to optimize the performance and to achieve a smooth end-effector motion task [58]. As redundant robots have more degree-of-freedom (DOF) than required, there usually exists multiple solutions, which motivates us for the consideration of exploiting the extra DOFs to improve the control performance.

The inverse kinematics problem is one of the fundamental task in understanding the operability of parallel manipulators i.e., to find the actuator inputs, provided the desired end-effector trajectories. Conventional design of the parallel mechanisms often encounter singularity problem. The intentional design of redundancy in parallel mechanism often provides a effective remedy. In [43], the authors proposed a new 3-DOF symmetric spherical 3-UPS/S parallel mechanism with three prismatic actuators, and studied the kinematics, statics, and workspace of the mechanism. In [43], a 2(SP+SPR+SPU) serial-parallel manipulators was considered. Based on the analysis, they designed three new types of kinematically redundant parallel mechanisms, including a new redundant 7-DOF Stewart platform. In [62], the damped least square method was utilized to tackle singularity problem. However, it only modifies the end-effector path in terms of velocity. It was shown in [76] that it is important

to analyze the inverse dynamic of parallel manipulators to find the joint friction and actuator dynamics.

Soft computing techniques, including neural networks [65, 96], fuzzy logic [55, 70], and genetic algorithm [11, 24] have been extensive used for robotic control and modeling. Due to the outstanding performance in parallel processing and recursive computation, dynamic neural networks, as a special type of neural networks, have long been employed as a powerful tool for the control of conventional serial robot arms. In [82], an adaptive recurrent neural network was employed for the control of a mobile manipulator with unknown dynamics. It was shown that the control strategy guarantees the asymptotical convergence to the desired motion. In [101], a unified quadratic-programming-based dynamical system approach was proposed to optimize the joint torque of serial manipulators. A minimum-energy redundancy resolution methodology was proposed in [102] for the recurrent neural networks control of serial manipulators. Periodic oscillations of the various neural networks were reported in the thesis. In order to reduce the network complexity and further increase the computational efficiency, [99] introduced a single layered dual neural network was proposed for the control of kinematically redundant manipulators. In [26], the recurrent neural network approach was extended to solve the cooperation of multiple manipulators organized in a distributed network.

In spite of the great success of dynamic neural networks in the control of robotic systems, especially in controlling serial robot manipulators, there is rarely report on using dynamic neural network to address the kinematic resolution problem of stewart platform, which is a typical parallel robot platform. In this thesis, we make progress along this direction and propose a dynamic neural solution to solve the kinematic resolution problem of Stewart platform in compliance with the physical constraints. Actually, for the Stewart platform, its forward kinematics are highly nonlinear and heavily coupled, which impose great challenges to the neural dynamic design to reach the same control goal.

The remainder of the chapter is organized into eight sections. Section II describes the preliminaries of the Stewart platform. Section III provides the background information

on the robotic kinematics of the constrained manipulators. Section IV describes the problem formulation as constrained optimization for the physically constrained manipulators. Section V provides a dynamic neural network model for the kinematic redundancy resolution problem of the parallel manipulator. Section VI presents the theoretical results on global optimality and dynamic convergence. Section VII illustrates the simulation results for position tracking and orientation tracking and their performance. Section VIII concludes the chapter.

2.2 Preliminaries

The pose (position and orientation) of a rigid body in three-dimensional space is uniquely determined by a translation, represented by a three-dimensional vector, and a rotation, represented by a 3×3 rotational matrix in terms of three Euler angles. The rotational matrix $\Omega \in \mathbb{R}^{3\times 3}$ defined by the Euler angles $[\phi_x, \phi_y, \phi_z]^T \in \mathbb{R}^3$ has the following property for its time derivative,

$$\dot{\Omega}\Omega^T = \begin{bmatrix} 0 & -\dot{\phi}_z & \dot{\phi}_y \\ \dot{\phi}_z & 0 & -\dot{\phi}_x \\ -\dot{\phi}_y & \dot{\phi}_x & 0 \end{bmatrix} \tag{2.1}$$

This property holds for all rotational matrices. Additionally, the rotational matrix is orthogonal, i.e., $\Omega\Omega^T = I$, $\Omega^{-1} = \Omega^T$ with I denoting a 3×3 identity matrix.

For two vectors $u = [u_1, u_2, u_3]^T \in \mathbb{R}^3$ and $v = [v_1, v_2, v_3]^T \in \mathbb{R}^3$, their cross-product, denoted as '$\times$', is defined as,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix}. \tag{2.2}$$

The triple product of three vectors in three-dimensional space is defined based on the cross product. For three vectors $u = [u_1, u_2, u_3]^T \in \mathbb{R}^3$, $v = [v_1, v_2, v_3]^T \in \mathbb{R}^3$, and $w = [w_1, w_2, w_3]^T \in \mathbb{R}^3$, their triple product is defined as $(u\times v)^T w$, and equals the following in value,

$$(u \times v)^T w = \det \begin{bmatrix} u^T \\ v^T \\ w^T \end{bmatrix}. \tag{2.3}$$

The triple product is invariant under circular shifting:

$$(u \times v)^T w = (v \times w)^T u = (w \times u)^T v \tag{2.4}$$

The matrix on the right-hand side (2.1) is a 3×3 skew-symmetric matrix. For a general 3×3 skew-symmetric matrix, as can be simply verified,

$$\begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \alpha = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \times \alpha \tag{2.5}$$

always holds for any $x, y, z \in \mathbb{R}$, $\alpha = [\alpha_1, \alpha_2, \alpha_3]^T \in \mathbb{R}^3$. Due to the above relation, it is common to use $[x, y, z]^T_\times$ to represent a three-dimensional skew-symmetric matrix as,

$$\begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \alpha = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_\times . \tag{2.6}$$

2.3 Robot Kinematics

The Stewart platform is a typical parallel mechanism and can be extended to different forms by modifying its mechanisms. It includes a mobile platform on the top as shown in Fig. 2.1, a fixed base, and six independent driving legs connecting the aforementioned two parts. The two ends of each leg are fixed on the mobile platform and the fixed based respectively using universal joints. Each leg can be actuated to change its length for the adjustment of the distance between the two fixed points on the platform and the base. All together, the six legs collaborates to adjust the orientation and position of the mobile platform by changing their lengths.

2.3.1 Geometric Relation

For the Stewart platform, the global coordinate is fixed on the base and the platform coordinate is fixed on the mobile platform. $a_i \in \mathbb{R}^3$ for $i = 1, 2, ..., 6$ represents the position in global coordinates of the ith connection point on the base. $b_i' \in \mathbb{R}^3$ for $i = 1, 2, ..., 6$ represents the position in platform coordinates of the ith connection point on the platform. We

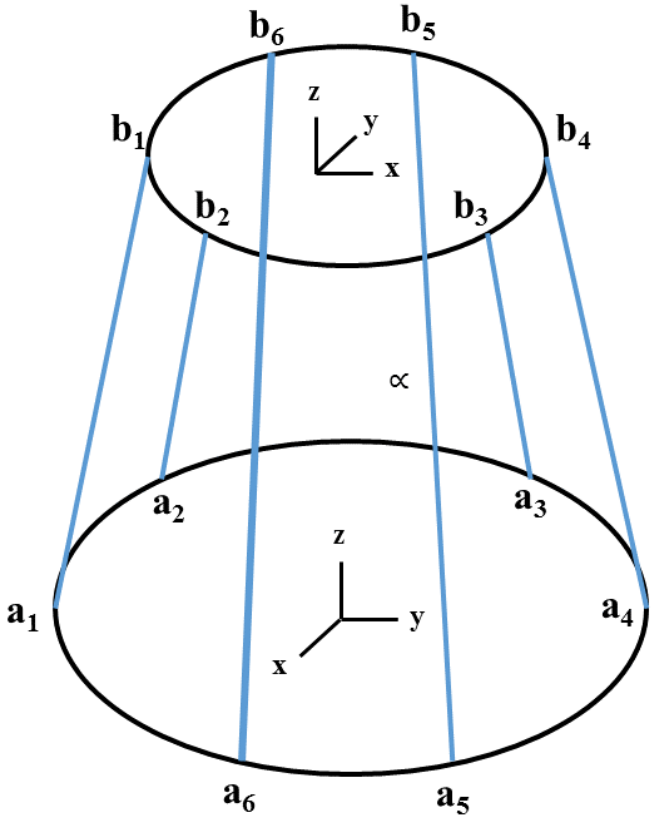


Figure 2.1: A schematic of the Stewart platform.

use b_i to represents its position in the global coordinate, as shown in Fig. 2.1. $d_i = b_i - a_i$ for $i = 1, 2, ..., 6$ represents the vector corresponding to the ith leg, which points from the base to the platform. For a point $x' \in \mathbb{R}^3$ in the platform coordinate, its position $x \in \mathbb{R}^3$ in global coordinate can be obtained after a rotational and translational transformation as follows,

$$x = p + Qx' \tag{2.7}$$

where $p = [x_p, y_p, z_p]^T \in \mathbb{R}^3$ is the global coordinate of the zero position in the platform coordinate, and it corresponds to the translational transformation, $Q \in \mathbb{R}^{3\times3}$ is the rotational matrix, which is uniquely defined by the Euler angles $\theta = [\theta_x, \theta_y, \theta_z]^T \in \mathbb{R}^3$,

$$\begin{aligned} Q &= Q_z Q_y Q_x \\ Q_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \\ Q_y &= \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \\ Q_z &= \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_y & \cos\theta_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.8}$$

Following (2.7), as to the ith connection point on the platform, i.e., the ones with $x = b_i$ in the global coordinates or the ones with $x' = b_i'$ in the platform coordinates, we have,

$$b_i = p + Qb_i' \tag{2.9}$$

Therefore, the ith leg vector can be further expressed as,

$$d_i = b_i - a_i = p + Qb_i' - a_i \tag{2.10}$$

For the vector d_i, we define $r_i = \|d_i\|$ to represent its length. Accordingly, we have

$$r_i = \|p + Qb_i' - a_i\| \tag{2.11}$$

Notice that both a_i and b_i are constants and are determined by the geometric structure. $p = [x_p, y_p, z_p]^T$ defines the translation of the platform, and Q as a function of the Euler angles

$\theta = [\theta_x, \theta_y, \theta_z]^T$, defines the rotation of the platform. Overall, the right-hand side of (3.14) depends on the pose variables of the platform $\pi = [x_p, y_p, z_p, \theta_x, \theta_y, \theta_z]^T \in \mathbb{R}^6$ while the left-hand side of (3.14) is the length of the leg, which is controlled for actuation. In this sense (3.14) for $i = 1, 2, .., 6$ defines the kinematic relation between the actuation variables and the pose variables. For a six-dimensional reference pose, the desired leg length r_i can be directly obtained from (3.14). However, in real applications, the reference are usually not six-dimensional. For example, for surgical applications of Stewart platform, people may only care about the position of an end-effector on the platform, instead of its orientation. In this situation, the reference is three-dimensional and we have three additional degree of freedom as redundancy. For such a situation, we usually have infinite number of feasible solutions of r_i for $i = 1, 2, ..., 6$ to reach the reference. Among the feasible solutions, we may be able to identify one, which outperforms others in terms of certain optimization criteria. This intuitive analysis motivates us to model it as an optimization problem and identify the optimal one for improved performance. However, due to the nonlinearity of (3.14), direct treatment of (3.14) is technically prohibitive. Instead of direct solution in position space, we turn to solve the problem in its velocity space to exploit the approximate linearity.

2.3.2 Velocity Space Resolution

For easy treatment, we convert (3.14) into the following equivalent form,

$$r_i^2 = (p + Qb_i' - a_i)^T (p + Qb_i' - a_i). \tag{2.12}$$

To obtain the velocity space relations, we first compute time derivative on both sides of (2.12), which yields,

$$\begin{aligned} r_i \dot{r}_i &= (p + Qb_i' - a_i)^T (\dot{p} + \dot{Q}b_i' + Q\dot{b}_i' - \dot{a}_i) \\ &= (p + Qb_i' - a_i)^T (\dot{p} + \dot{Q}b_i'). \end{aligned} \tag{2.13}$$

Recall that both a_i and b_i' are constants and their time derivatives, $\dot{a}_i$ and $\dot{b}_i'$, are equal to zero. For the rotational matrix Q, according to the preliminary equations (2.1) and (2.6), it has the

following property for its time derivative,

$$\dot{Q}Q^T = \begin{bmatrix} 0 & -\dot{\theta}_z & \dot{\theta}_y \\ \dot{\theta}_z & 0 & -\dot{\theta}_x \\ -\dot{\theta}_y & \dot{\theta}_x & 0 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix}_\times = \dot{\theta}_\times. \tag{2.14}$$

Therefore, $\dot{Q}$ writes as follows,

$$\dot{Q} = \dot{\theta}_\times (Q^T)^{-1} = \dot{\theta}_\times Q. \tag{2.15}$$

Substituting (2.15) into (2.13) yields,

$$\begin{aligned} r_i \dot{r}_i &= (p + Qb_i' - a_i)^T (\dot{p} + \dot{\theta}_\times Q b_i') \\ &= d_i^T (\dot{p} + \dot{\theta}_\times Q b_i') \\ &= d_i^T \dot{p} + \left((\dot{\theta} \times (Q b_i')\right)^T d_i \\ &= d_i^T \dot{p} + \left((Q b_i') \times d_i\right)^T \dot{\theta} \\ &= \begin{bmatrix} d_i^T & \left((Q b_i') \times d_i\right)^T \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix}. \end{aligned} \tag{2.16}$$

In the above equation, (2.10) and (2.4) are used for the derivation in the second line and the derivation in the second last line, respectively. Noticing that $r_i = \|d_i\| > 0$ could be guaranteed by the mechanical structure, we have the following result:

$$\begin{aligned} \dot{r}_i &= \frac{1}{r_i} \begin{bmatrix} d_i^T & \left((Q b_i') \times d_i\right)^T \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} \\ &= \frac{1}{r_i} \begin{bmatrix} d_i^T & \left((Q b_i') \times d_i\right)^T \end{bmatrix} \dot{\pi}. \end{aligned} \tag{2.17}$$

For the six-dimensional vector $r = [r_1, r_2, ..., r_6]^T$, we have the compact matrix form as follows,

$$\dot{r} = A_1 \dot{\pi}, \tag{2.18}$$

where

$$\begin{aligned} A_1 &= \begin{bmatrix} \frac{1}{r_1} d_1^T & \frac{1}{r_1}\left((Q b_1') \times d_1\right)^T \\ \frac{1}{r_2} d_2^T & \frac{1}{r_2}\left((Q b_2') \times d_2\right)^T \\ \cdots & \cdots \\ \frac{1}{r_6} d_6^T & \frac{1}{r_6}\left((Q b_6') \times d_6\right)^T \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{r_1}(p + Q b_1' - a_1)^T & \frac{1}{r_1}\left((Q b_1') \times (p - a_1)\right)^T \\ \frac{1}{r_2}(p + Q b_2' - a_2)^T & \frac{1}{r_2}\left((Q b_2') \times (p - a_2)\right)^T \\ \cdots & \cdots \\ \frac{1}{r_6}(p + Q b_6' - a_6)^T & \frac{1}{r_6}\left((Q b_6') \times (p - a_6)\right)^T \end{bmatrix}. \end{aligned} \tag{2.19}$$

Equation (2.18) gives the kinematic relation of a six-degree-of-freedom Stewart platform from the velocity of the pose variables to the speed of the legs.

2.4 Problem Formulation as Constrained Optimization

Compared to (3.14), (2.18) significantly simplifies the problem as $\dot{r}$ in (2.18) is now affine to the $\dot{\pi}$ while the relation between r_i and π (or p and Q) in (3.14) are nonlinear, or even non-convex to the pose variables. Similar to our analysis before, in the case that the reference pose velocity is given in six dimensions, the solution of $\dot{r}$ can be solved directly from (2.18). However, in the situation that the reference pose velocity is described in lower dimensions that six, the extra redundancies are available to reach improved performance. The following equality model the reference velocity constraint in reduced dimensions,

$$\alpha = A_2\dot{\pi} \tag{2.20}$$

where the reference vector $\alpha \in \mathbb{R}^m$ with $0 < m < 6$ is pre-given, the matrix $A_2 \in \mathbb{R}^{m\times 6}$ is the transformation matrix and is also pre-given. As an example, if we would like to maintain the platform at a given height, i.e., $\dot{p}_z = 0$, we set $\alpha = 0$ with $m = 1$ and $A_2 = [0, 0, 1, 0, 0, 0]^T$ in (2.20). Due to the extra design freedom, the value of $\dot{\pi}$ usually cannot be uniquely solved from (2.20). We thus define the following criteria to optimize the solution,

$$\min_{(\dot{\pi},\tau)} \frac{1}{2}\dot{\pi}^T\Lambda_1\dot{\pi} + \frac{1}{2}\tau^T\Lambda_2\tau \tag{2.21}$$

where $\Lambda_1 \in \mathbb{R}^{6\times 6}$ and $\Lambda_2 \in \mathbb{R}^{6\times 6}$ are both symmetric constant matrices and are both positive definite, $\tau = \dot{r}$ is the controllable speed of the platform legs. In application, the term $\dot{\pi}^T\Lambda_1\dot{\pi}$ can be used to specify the kinematic energy (including translational kinetic energy and rotational kinetic energy) by choosing proper weighting matrix Λ_1 (say choosing Λ_1 as one formed from the mass of the platform and its moment of inertia for the kinematic energy case), and the term $\tau^T\Lambda_2\tau$ characters the input power consumed by the robotic system. This objective function also follows the convention of control theory [74]. The decision variable τ, which is controlled by the actuators, is subjected to physical constraints. In this thesis, we

model the physical constraints as linear inequalities in the following form,

$$B\tau \leq b \tag{2.22}$$

where $B \in \mathbb{R}^{k\times 6}$ and $b \in \mathbb{R}^k$ with k being an integer. Note that constraints are not imposed to the variable $\dot{\pi}$ since its value usually is specified as a feasible one in the planning stage. In summary of (3.16) as the object function, (2.18) as the mapping relation, (2.20) to fulfil the reference tracking requirements, and (3.17) as physical constraints, we can formulate the kinematic control problem of the Stewart platform as the following constrained,

$$\begin{aligned}
\min_{(\dot{\pi},\tau)} \quad & \frac{1}{2}\dot{\pi}^T\Lambda_1\dot{\pi} + \frac{1}{2}\tau^T\Lambda_2\tau && \text{(2.23a)}\\
\text{s.t.} \quad \tau &= A_1\dot{\pi} && \text{(2.23b)}\\
\alpha &= A_2\dot{\pi} && \text{(2.23c)}\\
B\tau &\leq b. && \text{(2.23d)}
\end{aligned}$$

Due to the presence of both equation and inequality constraints in the optimization problem (2.23), usually it cannot be solved analytically. Conventional approaches introduces extra penalty terms formed by the constraints to the objective function and solve the problem numerically using gradient descent along the new objective function. However, penalty based approaches only reaches an approximate solution of the problem and thus are not suitable for error-sensitive applications. Instead of using this approximate approach, in the next section, we will propose a dynamic neural network solution, which asymptotically converges to the theoretical solution.

2.5 Dynamic Neural Network Model

In this section, we first consider the optimization problem (2.23) in its dual space and then present a neural network to solve it dynamically. After that, we investigate the hardware realization of the proposed model.

2.5.1 Neural Network Design

According to the Karush-Kuhn-Tucker (KKT) conditions [6], the solution of problem (2.23) satisfies,

$$\Lambda_1\dot{\pi} - A_1^T\lambda_1 - A_2^T\lambda_2 = 0 \tag{2.24a}$$

$$\Lambda_2\tau + \lambda_1 + B^T\mu = 0 \tag{2.24b}$$

$$\tau = A_1\dot{\pi} \tag{2.24c}$$

$$\alpha = A_2\dot{\pi} \tag{2.24d}$$

$$\begin{cases} \mu > 0 & \text{if } B\tau = b \\ \mu = 0 & \text{if } B\tau < b \end{cases} \tag{2.24e}$$

where $\lambda_1 \in \mathbb{R}^6$, $\lambda_2 \in \mathbb{R}^m$ (m is the number of rows in matrix A_2), and $\mu \in \mathbb{R}^6$ are dual variables to the equation constraint (2.23b), the equation constraint (2.23c) and the inequality constraint (2.23d), respectively. The expression (2.24e) can be simplified to the following one by introducing an saturation function,

$$\mu = (\mu + B\tau - b)^+ \tag{2.25}$$

where the nonlinear mapping '$(\cdot)^+$' is a function which maps negative values to zero and non-negative values to themselves. From (2.24a), $\dot{\pi}$ can be solved as,

$$\dot{\pi} = \Lambda_1^{-1}(A_1^T\lambda_1 + A_2^T\lambda_2). \tag{2.26}$$

Substituting $\dot{\pi}$ in (2.26) to (2.24c) and (2.24d) yields,

$$\tau = A_1\Lambda_1^{-1}(A_1^T\lambda_1 + A_2^T\lambda_2), \tag{2.27}$$

$$\alpha = A_2\Lambda_1^{-1}(A_1^T\lambda_1 + A_2^T\lambda_2). \tag{2.28}$$

To eliminate τ, we first represent it in terms of λ_1 and μ according to (2.24b) as,

$$\tau = -\Lambda_2^{-1}(\lambda_1 + B^T\mu) \tag{2.29}$$

then substitute (2.29) into (2.24c) and (2.27), which results in,

$$-\Lambda_2^{-1}(\lambda_1 + B^T\mu) = A_1\Lambda_1^{-1}(A_1^T\lambda_1 + A_2^T\lambda_2), \tag{2.30}$$

$$\mu = (\mu - B\Lambda_2^{-1}\lambda_1 - B\Lambda_2^{-1}B^T\mu - b)^+. \tag{2.31}$$

We use the following dynamics for the solutions of λ_1, λ_2 and μ in (2.28), (2.30) and (2.31):

$$\begin{aligned}
\epsilon\dot{\lambda}_1 &= -\Lambda_2^{-1}(\lambda_1 + B^T\mu) - A_1\Lambda_1^{-1}(A_1^T\lambda_1 + A_2^T\lambda_2) \\
\epsilon\dot{\lambda}_2 &= -A_2\Lambda^{-1}A_1^T\lambda_1 - A_2\Lambda^{-1}A_2^T\lambda_2 + \alpha \\
\epsilon\dot{\mu} &= -\mu + (\mu - B\Lambda_2^{-1}\lambda_1 - B\Lambda_2^{-1}B^T\mu - b)^+
\end{aligned} \tag{2.32}$$

where $\epsilon > 0$ is a scaling factor. Overall, the proposed dynamic neural network has λ_1, λ_2 and μ as state variables and τ in (2.27) as the output, which is expressed as follows in summary,

State equations:

$$\epsilon\dot{\lambda}_1 = -\Lambda_2^{-1}\lambda_1 - A_1\Lambda_1^{-1}A_1^T\lambda_1 - A_1\Lambda_1^{-1}A_2^T\lambda_2 - \Lambda_2^{-1}B^T\mu \tag{2.33a}$$

$$\epsilon\dot{\lambda}_2 = -A_2\Lambda^{-1}A_1^T\lambda_1 - A_2\Lambda^{-1}A_2^T\lambda_2 + \alpha \tag{2.33b}$$

$$\epsilon\dot{\mu} = -\mu + (-B\Lambda_2^{-1}\lambda_1 + \mu - B\Lambda_2^{-1}B^T\mu - b)^+ \tag{2.33c}$$

Output equation:

$$\tau = A_1\Lambda_1^{-1}A_1^T\lambda_1 + A_1\Lambda_1^{-1}A_2^T\lambda_2 \tag{2.33d}$$

About the proposed neural network model (3.18) for the kinematic redundancy resolution problem (2.23) of parallel manipulator, we have the following remark.

Remark 1. *Fig. 2.2 shows the architecture of the proposed model, for the situation with* $m = 3$, $k = 4$*(*m *is the dimension of* α, k *is the number of rows of matrix* B*). From this figure, it is clear that this neural network (3.18) is organized in a two-layer architecture. The first layer(feedback layer), which is composed of* $6 + m + k$ *neurons , is a nonlinear layer with dynamic feedback. This layer of neurons is associated with the state variables* $\lambda_1 \in \mathbb{R}^6$, $\lambda_2 \in \mathbb{R}^m$ *and* $\mu \in \mathbb{R}^k$*, and get input from the input variable* α*. It follows Eqs. (3.18a), (3.18b) and (3.18c) for dynamic updates. The second layer (output layer) is a static layer with linear mapping as described by (3.18d). It consists of* 6 *neurons and maps the state variables to the output, which is the platform leg speed* τ*.*

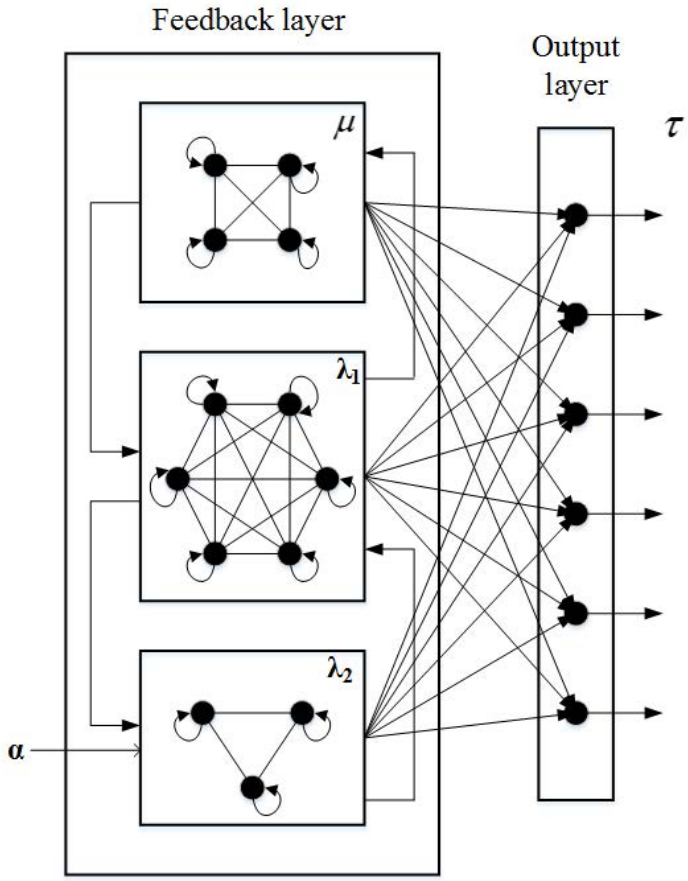


Figure 2.2: Two layer neural network architecture

2.6 Theoretical Results

In this section, we present theoretical results on the proposed neural networks for solving the redundancy resolution problem of parallel manipulators.

2.6.1 Optimality

In this part, we show the equilibrium point of the dynamic neural networks (3.18) ensures that the corresponding output τ given by (3.18d) is identical to the optimal solution of the problem (2.23). On this point, we have the following theorem,

Theorem 1. *Suppose* $(\lambda_1^*, \lambda_2^*, \mu^*)$ *is the equilibrium point of the dynamic neural network (3.18). Then the corresponding output* τ^* *obtained from the output equation (3.18d) is opti-*

mal to the constrained programming problem (2.23).

Proof. Letting the right-hand sides of the state equations (3.18a), (3.18b) and (3.18c) equal to zero, we find the following conditions about the equilibrium point $(\lambda_1^*, \lambda_2^*, \mu^*)$:

$$-\Lambda_2^{-1}\lambda_1^* - A_1\Lambda_1^{-1}A_1^T\lambda_1^* - A_1\Lambda_1^{-1}A_2^T\lambda_2^* - \Lambda_2^{-1}B^T\mu^* = 0 \tag{2.34}$$

$$-A_2\Lambda^{-1}A_1^T\lambda_1^* - A_2\Lambda^{-1}A_2^T\lambda_2^* + \alpha = 0 \tag{2.35}$$

$$-\mu^* + (-B\Lambda_2^{-1}\lambda_1^* + \mu^* - B\Lambda_2^{-1}B^T\mu^* - b)^+ = 0 \tag{2.36}$$

and the corresponding output is,

$$\tau^* = A_1\Lambda_1^{-1}A_1^T\lambda_1^* + A_1\Lambda_1^{-1}A_2^T\lambda_2^* \tag{2.37}$$

Define an auxiliary value,

$$\dot{\pi}^* = \Lambda^{-1}A_1^T\lambda_1^* + \Lambda^{-1}A_2^T\lambda_2^* \tag{2.38}$$

To show that τ^* is optimal to (2.23), we only need to show that (λ_1^*, λ_2^*, μ^*, τ^*, $\dot{\pi}^*$) satisfy the KKT condition (2.24) of the optimization problem (2.23), or equivalently the equation set composed of (2.26), (2.27), (2.28), (2.30) and (2.31) according to the analysis in Section 2.5.1, we can conclude that the equation set composed of (2.26), (2.27), (2.28), (2.30) and (2.31) is equivalent to the KKT condition (2.24). Comparing the equation set composed of (3.24), (3.25), (3.26), (3.27), (3.28), and the one composed of (2.26), (2.27), (2.28), (2.30) and (2.31), we find that they are identical and therefore are equivalent. The above procedure implies that the solution (λ_1^*, λ_2^*, μ^*, τ^*, $\dot{\pi}^*$) is optimal to (2.23). Therefore, we conclude that τ^* is optimal to the problem (2.23). □

2.6.2 Stability

In this part, we present theoretical results on the stability of the proposed dynamic neural network model. In Section 2.6.1, we have concluded that the equilibrium point of the neural network (3.18) is optimal solution of (2.23). Generally speaking, a dynamic system may not converge to its equilibrium points. It may happen for a dynamic system to evolve towards divergence, oscillation, or even chaos. It is necessary for the proposed neural network to converge for effective computation purpose. Before presenting the convergence results, we first present a lemma about a general projected dynamic system as shown below.

$$\dot{u} = -u + P_{\Omega}(u - F(u)) \tag{2.39}$$

where $\mu \in \mathbb{R}^l$, Ω is a closed convex set of $\mathbb{R}^l$, $P_{\Omega}(\cdot)$ is the projection operator onto the set Ω.

Lemma 1 (Xia, 2004 [87])**.** *If $\nabla F(u)$ is symmetric and positive semi-definite in $\mathbb{R}^l$, then the dynamic system (2.39) is stable in the sense of Lyapnov and is globally convergent to its equilibrium.*

The above lemma gives a general convergence results on dynamic systems with the presence of projection operators. In our system, the operator $(\cdot)^+$ is also a projection operator, which projects input values to non-negative ones. With Lemma 1, the following stability results on the proposed model (3.18) is provable.

Theorem 2. *The dynamic neural network (3.18) is stable in the sense of Lyapunov and is globally convergent to the optimal solution of (2.23).*

Proof. As we have proved that the output τ associated with the equilibrium points is optimal to problem (2.23) in Theorem 1, to draw the conclusion in this theorem we only need to show the convergence of (3.18) to its equilibrium points. To leverage the results presented in Lemma (1), we first convert (3.18) into a similar form as (2.39). Define a vector function

$F = [F_1^T, F_2^T, F_3^T]^T$, with F_1, F_2 and F_3 defined as follows,

$$\begin{aligned} F_1 &= \Lambda_2^{-1}\lambda_1 + A_1\Lambda_1^{-1}A_1^T\lambda_1 + A_1\Lambda_1^{-1}A_2^T\lambda_2 + \Lambda_2^{-1}B^T\mu \\ F_2 &= A_2\Lambda_1^{-1}A_1^T\lambda_1 + A_2\Lambda^{-1}A_2^T\lambda_2 - \alpha \\ F_3 &= B\Lambda_2^{-1}\lambda_1 + B\Lambda_2^{-1}B^T\mu + b \end{aligned} \tag{2.40}$$

and define a set Ω as,

$$\Omega = \{(\lambda_1, \lambda_2, \mu), \lambda_1 \in \mathbb{R}^6, \lambda_2 \in \mathbb{R}^m, \mu \in \mathbb{R}^k, \mu \geq 0\} \tag{2.41}$$

where $\mu \geq 0$ is defined in the piece-wise sense. We also define a new variable,

$$x = \begin{bmatrix} \lambda_1^T & \lambda_2^T & \mu^T \end{bmatrix}^T. \tag{2.42}$$

With the above definitions of F, Ω and x, the proposed neural network (3.18) can be converted as,

$$\epsilon\dot{x} = -x + P_\Omega(x - F(x)). \tag{2.43}$$

Define a new time scale $\phi = \frac{t}{\epsilon}$. With the new time scale, $\epsilon\dot{x} = \epsilon\frac{dx}{dt} = \frac{dx}{d\phi}$, and the neural dynamic (2.43) converts to,

$$\frac{dx}{d\phi} = -x + P_\Omega(x - F(x)) \tag{2.44}$$

which are in the nominal form of the projected dynamic systems. To prove the convergence, we only need to show the symmetricity of ∇F defined in (2.40) and its positive semi-definite.

$$\nabla F = \begin{bmatrix} \Lambda_2^{-1} + A_1\Lambda_1^{-1}A_1^T & A_1\Lambda_1^{-1}A_2^T & \Lambda_2^{-1}B^T \\ A_2\Lambda_1^{-1}A_1^T & A_2\Lambda_1^{-1}A_2^T & 0 \\ B\Lambda_2^{-1} & 0 & B\Lambda_2^{-1}B^T \end{bmatrix}. \tag{2.45}$$

Clearly, ∇F is symmetric. As to the positive semi-definiteness, we decompose ∇F in (2.45) into the following form,

$$\begin{aligned}\nabla F &= \begin{bmatrix} A_1\Lambda_1^{-1}A_1^T & A_1\Lambda_1^{-1}A_2^T & 0 \\ A_2\Lambda_1^{-1}A_1^T & A_2\Lambda_1^{-1}A_2^T & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \Lambda_2^{-1} & 0 & \Lambda_2^{-1}B^T \\ 0 & 0 & 0 \\ B\Lambda_2^{-1} & 0 & B\Lambda_2^{-1}B^T \end{bmatrix} \\ &= \begin{bmatrix} A_1\Lambda_1^{-\frac{1}{2}} \\ A_2\Lambda_1^{-\frac{1}{2}} \\ 0 \end{bmatrix} \begin{bmatrix} A_1\Lambda_1^{-\frac{1}{2}} \\ A_2\Lambda_1^{-\frac{1}{2}} \\ 0 \end{bmatrix}^T + \begin{bmatrix} \Lambda_2^{-\frac{1}{2}} \\ 0 \\ B_2\Lambda_2^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Lambda_2^{-\frac{1}{2}} \\ 0 \\ B_2\Lambda_2^{-\frac{1}{2}} \end{bmatrix}. \end{aligned} \tag{2.46}$$

The above expression implies that ∇F is indeed positive semi-definite. In summary, as ∇F defined in this proof is symmetric and positive semi-definite, the dynamic system (2.44), i.e., the proposed neural network (3.18) is stable and is convergent to the optimal solution of (2.24) according to Lemma 1.

□

2.7 Numerical Investigation

To validate the effectiveness of the proposed approach, in this section we apply the neural network model to the redundancy resolution of a physically constrained Stewart platform.

2.7.1 Simulation Setups

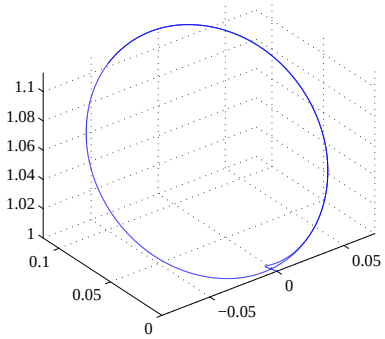
In the simulation, we consider a Stewart platform with the leg connectors on the mobile platform locating around a circle with radius 1.0m at $b_1' = [0.7386, 0.1302, 0]$, $b_2' = [0.7386, -0.1302, 0]$, $b_3' = [-0.4821, 0.5745, 0]$,$b_4' = [-0.2565, 0.7048, 0]$, $b_5' = [-0.2565, -0.7048, 0]$, and $b_6' = [-0.4821, -0.5745, 0]$ in the platform coordinate, and the leg connectors locating around a circle with radius 0.75m at $a_1 = [0.3750, 0.6495, 0]$, $a_2 = [0.3750, -0.6495, 0]$,

$a_3 = [-0.7500, 0.0000, 0]$, $a_4 = [0.3750, 0.6495, 0]$, $a_5 = [0.3750, -0.6495, 0]$, $a_6 = [-0.7500, 0.0000, 0]$ on the fixed base. For simplicity, the end-effector is put at the origin of the platform coordinate (this can always be achieved by defining the platform coordinate with its origin at the end-effector position). In the situation for position tracking in three dimensional space, the total redundancy is 3 as the input dimension is 6 while the output dimension is 3.

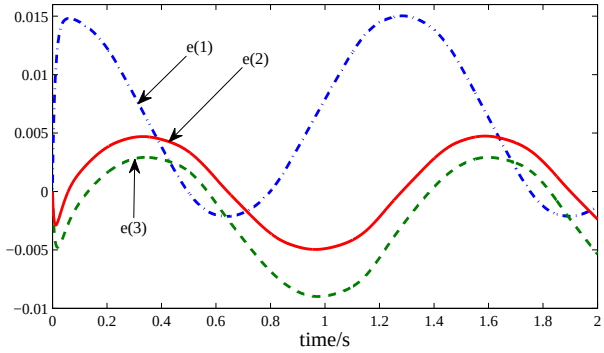
In the simulation, we use the tracking error, defined as the difference between the desired position and the real position at time t, to measure the tracking performance in both the circular motion tracking and the square motion tracking situation.

2.7.2 Circular Trajectory

In this part, we consider the tracking of a smooth circular path using the proposed method. The desired motion of the end-effector is to follow a circular trajectory at the speed of $2m/s$. The desired circle trajectory is centered at $[-0.04, 0.06, 1.05]$ with an radius of 0.08m, and has a revolute angle around the x-axis for 45 degrees. In the simulation, we simply choose Λ_1 and Λ_2 as an identity matrix. The value of the matrix A_1 is computed in real time according to (2.19). The matrix A_2 is chosen as $A_2 = [I_{3\times3}, 0_{3\times3}]$ such that the position tracking requirements can be achieved. In practice, the actuation speed of each leg is limited within a range due to the physical constraints of the actuators. To capture this property, we impose the constraint that the speed tau is no greater than $\eta > 0$ in its absolute value, i.e., $|\tau_i| \leq \eta$ for $i = 1, 2, ..., 6$, which is equivalent to $-\eta \leq \tau_i \leq \eta$. Organizing to a matrix form yields an inequality in the form of (2.23d) with $B = \begin{bmatrix} I_{6\times6} \\ I_{6\times6} \end{bmatrix}$, $b = \eta \mathbf{1}_{12}$ with $\mathbf{1}_{12}$ representing a twelve-dimensional vector with all entries equal one. In the simulation, we set the speed bound $\eta = 0.25m/s$. The scaling factor ϵ is chosen as $\epsilon = 0.01$. The tracking results are shown in Fig. 2.3 by running the simuation for 2 seconds. As shown in the Fig.2.3(a), the end-effector successful tracks the circular path with a small tracking error (as shown in Fig.2.3(b), where $e(1)$, $e(2)$, and $e(3)$ denotes the components of the position tracking error e, respectively, along the x, y, and z axes of the base frame, the errors are less than $0.015m$



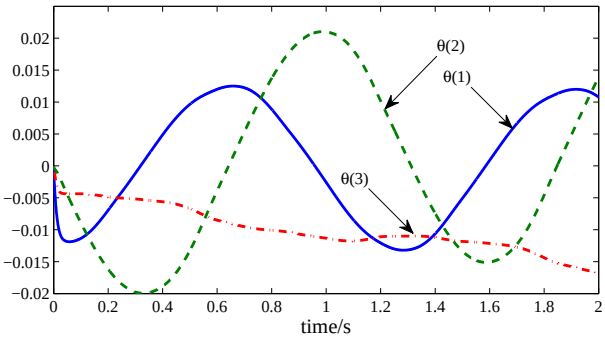
(a) End-effector trajectory



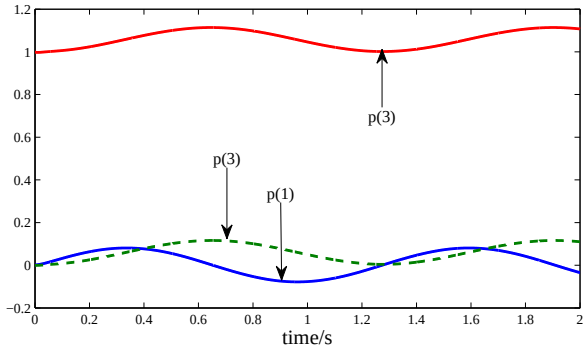
(b) Position tracking error

Figure 2.3: Tracking of a circular motion. (a). The trajectory of the end-effector; (b). Time history of the position tracking error.

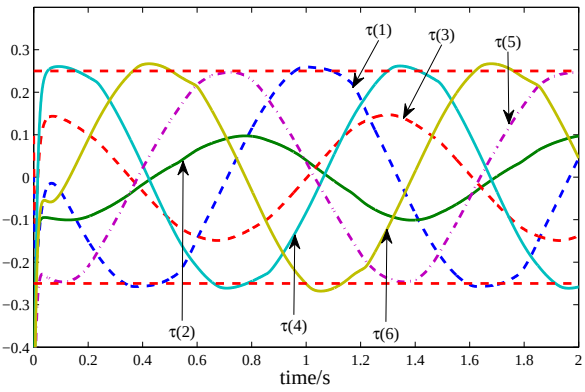
in amplitude). The circular-path following experiments demonstrated the capability of the proposed dual neural network for online resolving of kinematic redundancy of physically constrained manipulators. Considering the input motion for the 6 legs, the Fig.2.4 show the time profile of the Stewart platform state variables, i.e., the 3 Euler orientations of the mobile platform (Fig.2.4(a)), the end-effector position (Fig.2.4(b)), the speed of each leg (Fig.2.4(c)) and the length of each leg (Fig.2.4(d)). The attached moving frame to the upper platform is exactly in the middle and the $p(1)$ and $p(2)$ coordinates started from zero and the $p(3)$ oscillates between almost 0.005 meters and -0.006 meters. The $p(2)$ coordinate started from approximately 0.125 meters which is the altitude of the upper platform before the motion of the actuators. It was observed that the harmonic response is repeated every



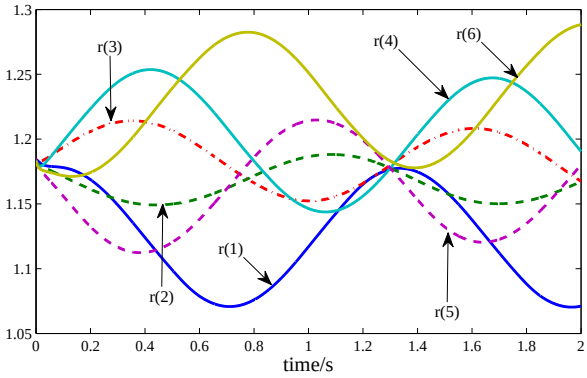
(a) Orientation of the platform θ



(b) Position of the end-effector p



(c) Control action τ



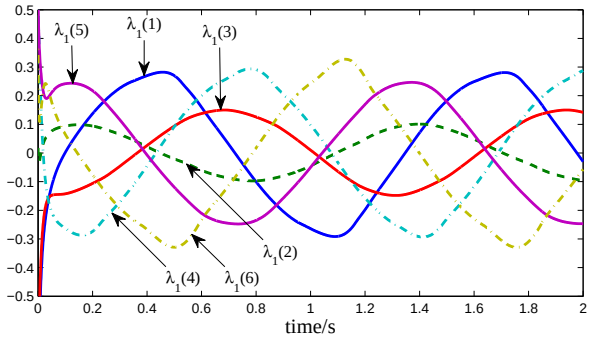
(d) Leg length r

Figure 2.4: The time evolution of the Stewart platform state variables in the case of circular motion tracking.

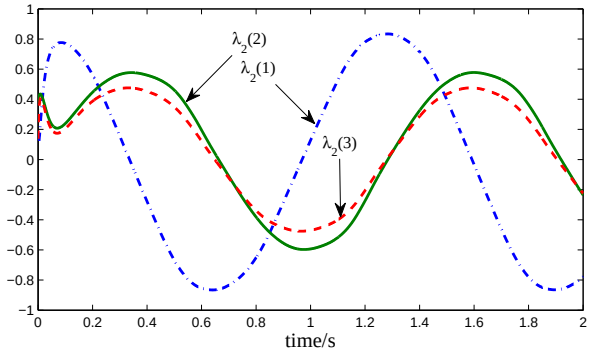
1.25 seconds or 2π after that the cycle is repeated again. From Fig.2.4(a), it is observed that there is a small drift from the neutral position. Although the presence of this drifting in orientation, the desired motion, which is given in terms of end-effector positions, are still achived as shown in Fig.2.3. This in turn validates the robustness of the proposed scheme in the presence of orientation drifting. The red dash line in Fig.2.4(c) depicts the bound ± 0.25 ($\eta = 0.25$), which is the bound for the action speed. It can be observed that τ converges to the region $[-0.25, 0.25]$ very fast and stays approximately inside this region through the runtime, except some short period (e.g., $\tau(6)$ at around time $t = 1.7$s) due to the dynamics of the desired motion. The neural network state variable are plotted in Fig.2.5. From this figure, we can clearly observer the dynamic evolution of the neural activities. It is noteworthy that the neural activities do not converge to a constant value. Instead, they varies with time as they are utilized to compensate and regulate the dynamic motion of the robot.

2.7.3 Square Trajectory

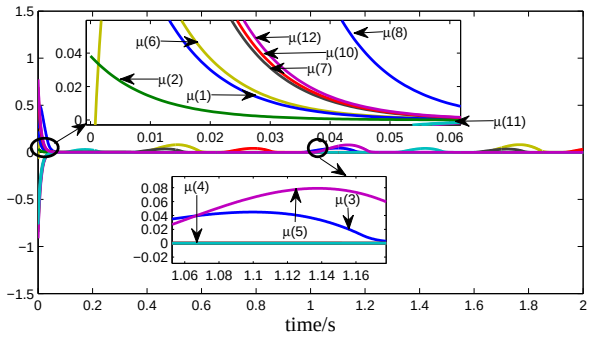
In this section, we investigate the square trajectory tracking using the proposed approach. Different from the case of smooth circular motion tracking, the desired square path is non-smooth at the four corners and poses challenges to the controller on its real-time performance. In the simulation, the desired motion of the end effector is to follow a square trajectory, which is centered at $[0.15, 0.075, 0.74]$ with the edge length of 0.08m, at the deisred speed $1.0m/s$. The square has a revolute angle around the x-axis for 60 degrees. We choose the parameterers Λ_1 and Λ_2, A_2 and B the same values as in the situation for circular motion for simplicity. The parameter A_1 is computed in real time according to (2.19). The speed limit bound η is chosen as $\eta = 0.6m/s$, and b is accordingly chosen as $b = 0.6\mathbf{1}_{12}$ with $\mathbf{1}_{12}$. The scaling factor ϵ is chosen as $\epsilon = 0.001$. The tracking results are shown in Fig. 2.6 by running the simuation for 2 seconds. As shown in the Fig. 2.6(a), the end-effector successful tracks the square motion with a small tracking error (as shown in Fig.2.6(b), where $e(1)$, $e(2)$, and $e(3)$ denotes the components of the position tracking error e, respectively, along the x, y, and z axes of the base frame, the errors are less than $0.006m$ in amplitude). Remark-



(a) Neural Network state variable λ_1

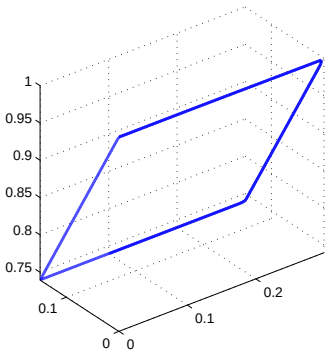


(b) Neural Network state variable λ_2

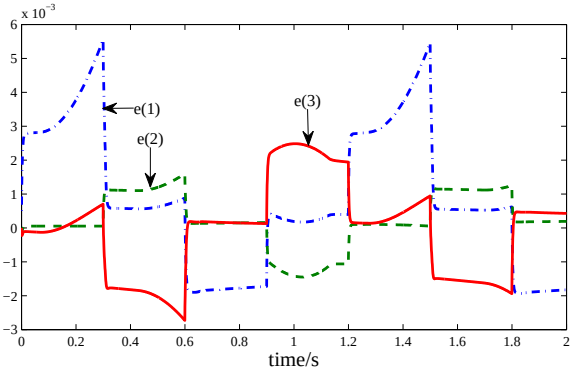


(c) Neural Network state variable μ

Figure 2.5: The time evolution of the neural network state variables in the case of circular motion tracking.



(a) End-effector trajectory



(b) Position tracking error

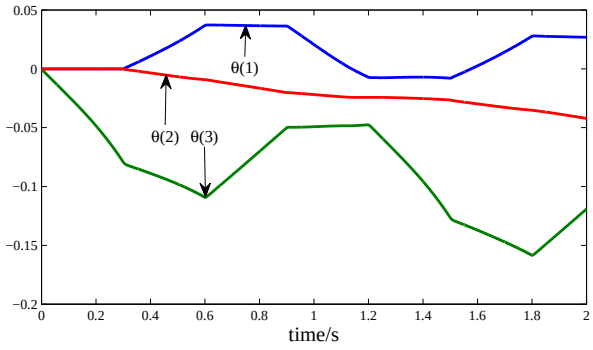
Figure 2.6: Tracking of a square motion. (a). The trajectory of the end-effector; (b). Time history of the position tracking error.

ably that the error curves have jerks at time $t = 0.3s, 0.6s, 0.9s, 1.2s, 1.5s, 1.8s$, which is a result of the velocity switching from one direction to another one around the corner of the square. Despect of the existence of the jerks, due to the nonlinear feedback mechanism in the neural network, the errors reduces swiftly to a very low value (much lower than $0.001m$ as shown in Fig. 2.3(b) after the jerk). The above observation in turn validates the effectiveness of the propsed neural scheme in dealing with non-smooth tracking problem. The real-time values on the Stewart platform states are shown in Fig. 2.7. Similar to the situation for circle tracking, the presence of drifting in orientations, as observed in Fig. 2.7(a), does not affect the tracking performance. The end-effector position evolves without drifting (Fig. 2.7(b)), and remains a very small error from its desired trajectory (Fig. 2.3(b)). Fig. 2.3(c) shows the

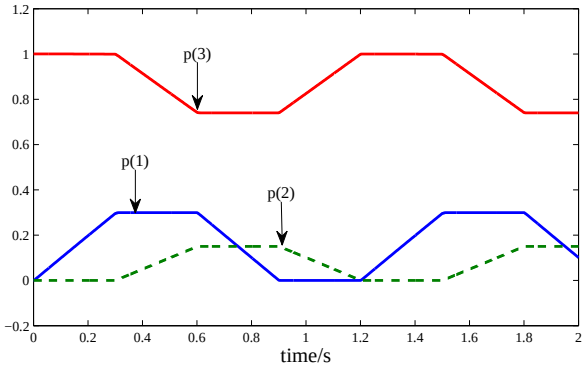
time profile of the control action, which is the speed of each leg. It is clear that the control action is approximately regulated inside the range $[-0.6, 0.6]$, which in turn validates the effect of the proposed solution in fulfilling the inequality regulation. The dynamic evolution of the neural activities are shown in Fig. 2.8.

2.8 Conclusion

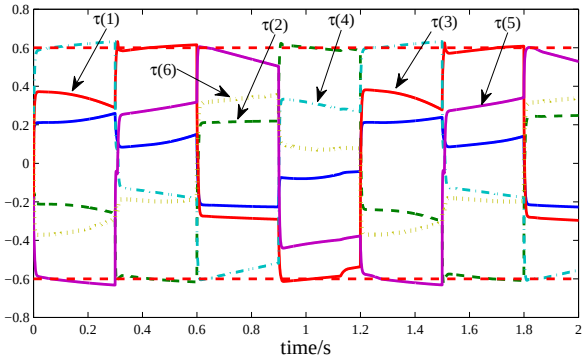
In this chapter, a dynamic neural network is designed to solve the redundancy resolution problem of Stewart platforms. The physical constraints and actuation are rigorously modelled as a constrained quadratic programming problem. To solve this problem in real time, a recurrent neural network is proposed to reach the equality constraints, inequality constraints, and optimality criteria simultaneously. Rigorous theoretical proof are supplied to verify the convergence of the proposed model. Simulation results validate the effectiveness of the proposed solution.



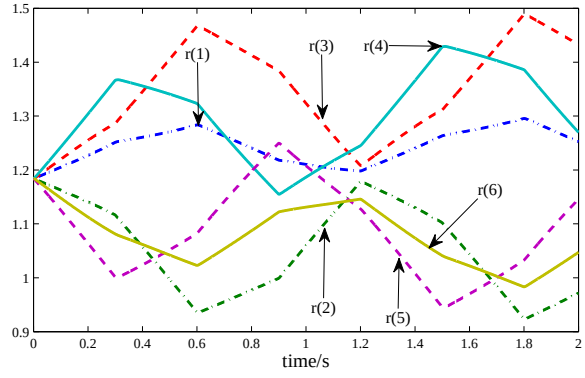
(a) Orientation of the platform θ



(b) Position of the end-effector p

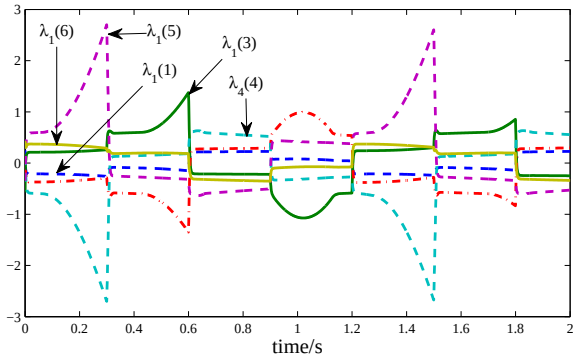


(c) Control action τ

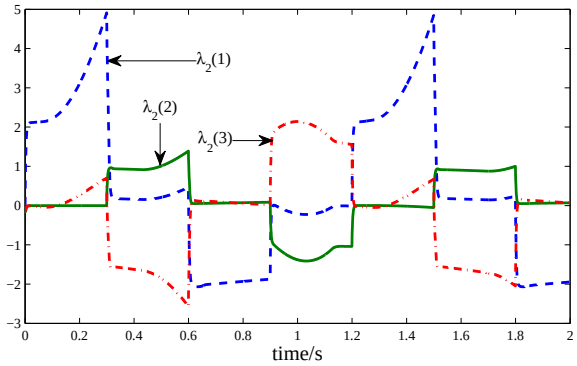


(d) Leg length r

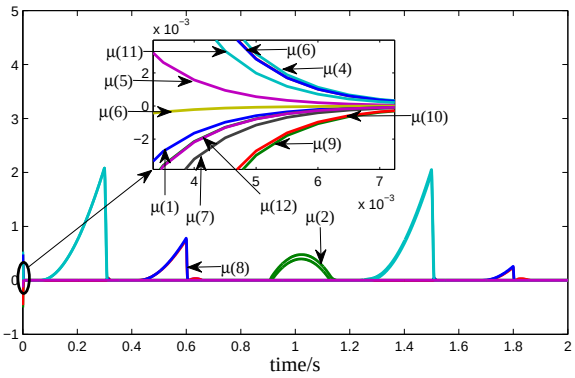
Figure 2.7: The time evolution of the Stewart platform state variables in the case of square



(a) Neural Network state variable λ_1



(b) Neural Network state variable λ_2



(c) Neural Network state variable μ

Figure 2.8: The time evolution of the neural network state variables in the case of square motion tracking.

CHAPTER 3

SIMULTANEOUS LEARNING AND CONTROL OF PARALLEL STEWART PLATFORMS WITH UNKNOWN PARAMETERS

3.1 Introduction

Kinematically redundant manipulators [103] are those manipulators that prove to have sufficiently higher degrees of freedom (DOF) than required for positioning and for orientation of the platform. With the advancement in the field of robotic technologies, robotic manipulators are widely used in the applications of factory automation which are required to carry out continuous and delayed work, such as lifting and transporting radioactive substances and executing the work in a hazardous, scattered or packed environments. In comparison to non-redundant manipulators, redundant ones offers extra DOFs [20, 25, 44, 95], and are often used to improvise the dexterity, in order they work efficiently by avoiding collisions with the obstacles. The research interests in the field of kinematically redundant manipulators [72] have gained popularity due to their ability to avoid obstacles, internal singular configurations, optimize the performance of the workspace and the end-effector motion task [59]. Among various types of redundant manipulators, parallel ones, which usually feature higher rigidity, higher precision and higher response speed than serial ones, have received popular applications in flight simulators, electrostatic magnetic lenses, etc. However, how to efficiently control the motion of redundant parallel manipulators, especially in the situation with parameter uncertainties, or even unknown, sets great practical significance but also remains as a challenging research problem. Parallel redundant manipulators [59] are broadly classified as parallel manipulators for whose the the task space coordinates and are lower/lesser than the number of actuators. These manipulators found many industrial applications like

robotics arms, surgical robots and so on. These manipulators offer greater advantages when incorporated because it makes the structure flexible, faster and lighter thereby improving the Cartesian stiffness and optimizing the distribution of the force. Due to the advantages of high speed and high acceleration, parallel manipulators have been studied and implemented widely.

Redundant manipulator holds the similar dynamic equation [83] like the serial one. Therefore, extending design works of serial manipulators to parallel manipulators becomes relatively obvious. Though, the designing the dynamics of the parallel manipulator becomes more complicated. Moreover, its become quite challenging to identify the unknown parameter which describes the dynamic behavior or properties of the system. Hence, the control schemes of the existing traditional methods for serial manipulators cannot be extended in real-time control applications to parallel redundant manipulators.

Parallel manipulators are confined to age-old and basic problems of identification and classification of singularities [2, 106]. A lot of work is developed along mathematical tools borrowed from serial manipulators for local analysis and to resolve the problem of singularities. Gosselin et. al were the first to define, study, analyze and report the singularity problem for the closed-loop kinematic chains. The structure and behavior of singularity problem for the parallel manipulator is indeed complex and challenging. Many works are incorporated to address the kinematic manipulability measure for design and control of parallel mechanisms. Actually, many methods dealing with nonlinear control [37, 79–81, 104, 109, 110] and nonlinear optimization [38, 45–51, 51–53, 98, 105] are potentially applicable to Stewart platform control. To name a few, a series of neural network approaches are established to solve serial robot control and extended to address multiple robot coordination problems [15, 17–19, 31, 40, 41, 57]; some of the representative works in serial manipulator control are also extended to parallel robot control [61, 107]; recent works on winner-take-all neural networks, and their applications to address competitive control of multiple robots cast lights to the competition control of multiple Stewart platforms [14, 34, 35, 42]; works on multiple mobile robot cooperation and consensus help us understand the control principle of coordinated systems and investigate their applications to Stewart control [21–23, 27–30, 32, 36];.con-

vergence acceleration techniques may also applicable to speed up the control performance, resulting in a timely and accurate solution [89–92, 94]; the universal nonlinear approximation power of feed-forward neural networks allow us to consider them as candidates to fit the nonlinearity of Stewart platform [33, 39].

Recurrent neural networks [13, 93, 100], as a powerful parallel computation method, are proven effective and efficient for the applications of real-time solutions to the inverse kinematics problem. In the literature of past decade, a variety of dynamical system solvers are proposed to resolve the problems of online constrained quadratic programming, including primal dual network, Lagrange neural networks, the gradient network and projected network [93, 100, 111]. For example, in [108], joint-constrained inverse kinematics, redundancy resolution have employed numerical algorithms with inclusion of physical constraints. By incorporating the velocity and the joint constraints Jacobian matrix augmented in the constraints are violated. There are also traditional approaches which considers joint and velocity constraints. For expressing a general solution in the form of redundant join velocities Gram-Schmidt orthogonalization procedure is utilized. Cheng [7] formulates the quadratic programming form from a constrained kinematic redundancy problem. This methods improved the computational efficiency by incorporating Gaussian eliminations with partial pivoting. Generally speaking, the numerical solutions to redundant kinematics are computationally intensive.

Constraints in soft computing techniques [16, 66, 97], introduce majorally two categories of difficulties in obtaining the solution to the problem. Firstly, the challenge is of the independence where the coordinate are not independent and secondly, a prior information of the constraints forces is not sufficiently provided and they are regarded among the unknowns of the system. Hence, control of Stewart platform as a constraint system becomes complicated due to the complex nature of the neural dynamics [10]. Another limiting factor in conventional robotic manipulation researches using dual neural network approaches is the requirement for the design model, which involves constructing a mathematical model that highlights the controlled dynamics of the model [12, 78]. The initial stage of the design usually requires to establish the inter dependence between different parts and their histori-

cal dependence on previous states. Later stage requires the analytical controller design with the mathematically modeled system dynamics. Although the designed control law gives a promising performance for the mathematical model, this might not be the case in real time applications as the exact representation of the model is hard to obtain. This may be due to various reasons, .e.g., the fact of sheer complexity of the designed model or the uncertainty involved in the area [9]. However, modeling of the feedback control for the physical system brings out the tradeoff between the ease of model and its precision of the physical system in matching. Due to the improvement of the parameters in the controller depending upon the convergence and stability factors, adaptive techniques usually demonstrate outstanding results in the face of complex systems. This motivates us to devise an adaptive and model free neural controller to steer the motion of a Stewart platform.

In this chapter we enlighten the close gap between two disjoint research areas of model-free recurrent neural networks for machine learning and model-based dual neural networks for accurate model control. The proposed network interacts with the learning and control parts [64]. An excitation noise is added to avoid the learning degradation. This deliberate design offers precise convergence of the estimated variables to their true values. The stability of the network is proved theoretically and by simulations and is proved that the bounded error is achieved to be arbitrarily small by scaling the additive noise.

The rest of the chapter is structured as defined; Section II is about the mathematical foundation of the parallel stewart system. Section III reviews the literature on the structural geometry and kinematics of the system. Section IV formulate the problem of model-free redundancy resolution as an optimization problem. This sections further elaborates the global stability of the proposed neural network. Section V describes the mathematical conclusions on the convergence to optimal solutions. Section VI illustrates the simulation results. Section VII gives the conclusion of this chapter.

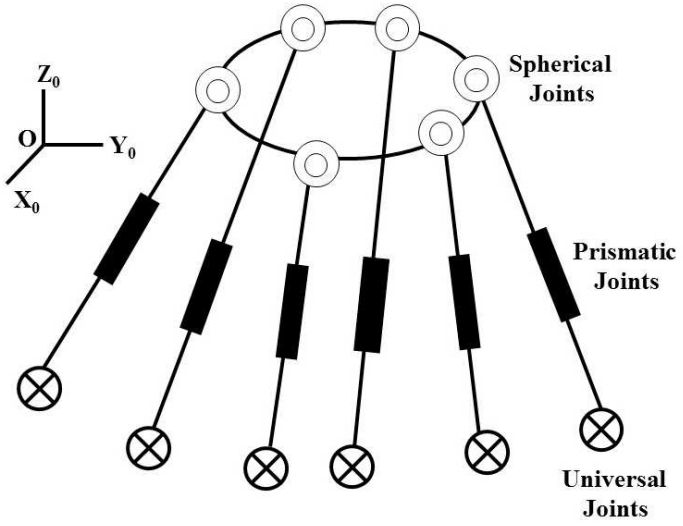


Figure 3.1: The schematic of a Stewart platform.

3.2 Kinematic Modeling of Stewart Platforms

A Stewart platform, as sketched in Fig. 3.1, consists of a six DOFs platform comprising two plates namely a fixed base plate and a flexible or moving top plate, which is in turn connected with the series of prismatic actuators and passive joints. Each of the prismatic actuators are connected by a spherical joint to the base plate. A base plate is connected by universal joints to each of its actuators. This specific arrangement of actuators and joints allows the top moving plate to move on either sides depending upon the lengths of the prismatic actuators or leg joints.

3.2.1 Geometric Relation

There are two coordinate systems associated with a Stewart platform, namely, base coordinate system which is fixed as a global system and a moving coordinate system of the platform [86]. We use x' to distinguish a variable defined in the base coordinate system from the corresponding one defined as x in global coordinate . The diagram shown in Fig. 3.2, the position vectors b_i indicates the position of the center of the universal joint of the leg. Thus, a_i, defined as the position vectors, represents the moving platform positions in global coordinates of the base's ith connection point. The vector $d_i = b_i - a_i$ represents the ith

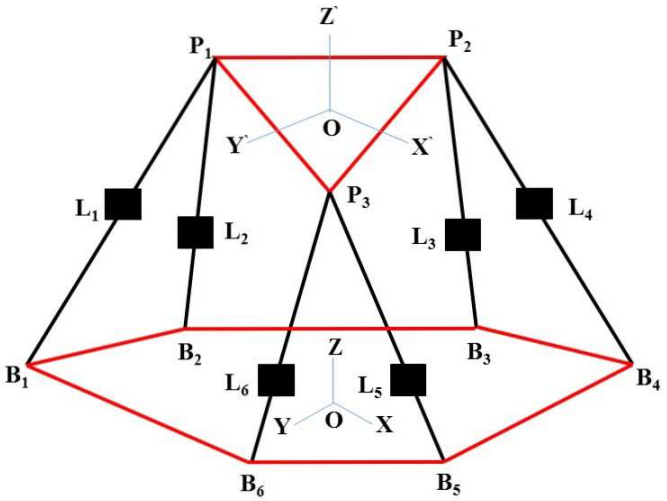


Figure 3.2: Stewart platform geometric representation. The red triangle on the top and the red hexagon below it represent the moving top plate and the fixed base plate, respectively.

leg of the actuator pointing from the base to the platform. The global coordinates and the platform coordinates are fixed to the base and the mobile platform, respectively. $a_i \in \mathbb{R}^3$ for $i = 1, 2, \cdots, 6$ dictates the position in global coordinates of the ith connection point on the base. $b_i' \in \mathbb{R}^3$ for $i = 1, 2, \cdots, 6$ denotes the position in platform coordinates of the ith connection point on the platform. Hence b_i is defined to represents the global coordinate (see Fig. 3.2). $d_i = b_i - a_i$ for $i = 1, 2, \cdots, 6$ is defined to represent the ith leg vector from the base to the platform. For $x' \in \mathbb{R}^3$ in platform coordinate, the corresponding global coordinates $x \in \mathbb{R}^3$ in can be derived after a translational and rotational transformation:

$$x = p + Qx', \tag{3.1}$$

where $p = [x_p, y_p, z_p]^{\mathrm{T}} \in \mathbb{R}^3$ representing the global coordinates of the origin position in the platform coordinate, and corresponding to the translational transformation, $Q \in \mathbb{R}^3$ is the rotational matrix, which is predominantly defined by the Euler angles $\theta = [\theta_x, \theta_y, \theta_z]^{\mathrm{T}} \in \mathbb{R}^3$,

$$Q = Q_z Q_y Q_x,$$

$$Q_x = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \cos\theta_x & \sin\theta_x \\ 1 & -\sin\theta_x & \cos\theta_x \end{bmatrix}$$

$$Q_y = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix}$$
$$Q_z = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_y & \cos\theta_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

Following (3.1), as to the ith connection point on the platform, i.e., the ones with $x = b_i$ in the global coordinates or the ones with $x' = b'_i$ in the platform coordinates, we have

$$b_i = p + Qb'_i, \tag{3.3}$$

Therefore, the ith leg vector can be further expressed as

$$d_i = b_i - a_i = p + Qb'_i - a_i \tag{3.4}$$

For the vector d_i, we define $r_i = \|d_i\|$ to represent its length. Accordingly, we have

$$r_i = \|p + Qb'_i - a_i\|, \tag{3.5}$$

It is to be observed that both a_i and b_i are constants. Henceforth, they can be derived by the geometric structure. $p = [x_p, y_p, z_p]^{\mathrm{T}}$ denotes the coordinate frame which defines the translation of the platform, and Q is representing a function of the Euler angles $\theta = [\theta_x, \theta_y, \theta_z]^{\mathrm{T}}$, denoting the rotation of the Stewart platform. Taking into consideration, the righthand side of (3.5) which depends on the pose variables of the Stewart platform $\pi = [x_p, y_p, z_p, \theta_x, \theta_y, \theta_z]^{\mathrm{T}} \in \mathbb{R}^6$ while the left-hand side of the (3.5) is the distant-length of the leg, which is in turn controlled for actuation. Hence we highlight that in this way, (3.5) for $i = 1, 2, \cdots, 6$ defines the relationship of kinematics between the actuation variables and the pose variables. Now, for a defined six-dimensional reference position, the desired leg length r_i can be directly obtained from (3.5). However, in real-time applications, the reference are usually not defined in six-dimensional. This could be explained as for example, in medical industry say; for use of surgical applications which includes Stewart platform, doctors only care about the position of an end-effector on the platform, and are not eagerly motivated to its orientation. Owing to this fact usually in these scenarios, the reference is three-dimensional and therefore we have three additional degrees of freedom as redundancy. For these kind

of upcoming challenges, we ultimately have infinite number of feasible solutions of r_i for $i = 1, 2, \cdots, 6$ to reach the reference. Among all of the feasible options and solutions, we may be able to identify a unique solution, which outperforms others and existing solutions in terms of certain criterion of optimization. This intuitive and mathematical approach analyzes and motivates for modeling of optimization problem and to identify the optimal solution for the improvised performance. Due due to presence of the nonlinearity of (3.5), treating of (3.5) directly is technically impractical and prohibitive. Hence, we formulate the problem in terms of velocity space to explore the linearity approximation rather than studying the problem in position space for a direct solution.

3.2.2 Velocity Space Resolution

For easy of simplicity and for the treatment, (3.5) is re-arranged as follows

$$r_i^2 = (p + Qb_i' - a_i)^{\mathrm{T}}(p + Qb_i' - a_i). \tag{3.6}$$

We define, represent and obtain the velocity space relations by computing the time derivative on both sides of (3.6), which yields

$$r_i\dot{r}_i = (p + Qb_i' - a_i)^{\mathrm{T}}(\dot{p} + \dot{Q}b_i' + Q\dot{b}_i' - \dot{a}_i) = (p + Qb_i' - a_i)^{\mathrm{T}}(\dot{p} + \dot{Q}b_i'). \tag{3.7}$$

Therefore, if we recollect and recall that both a_i and b_i' are constants and their time derivatives, $\dot{a}_i$ and $\dot{b}_i'$, are equivalent to zero. For achieving the purpose rotational matrix Q, we consider the above discussed preliminary equations, and the following represent the time derivative property

$$\dot{Q}Q^{\mathrm{T}} = \begin{bmatrix} 0 & -\dot{\theta}_z & \dot{\theta}_y \\ \dot{\theta}_z & 0 & -\dot{\theta}_x \\ -\dot{\theta}_y & \dot{\theta}_x & 0 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_y \end{bmatrix}_{\times} = \dot{\theta}_{\times}. \tag{3.8}$$

In this way, the moving platform rotation matrix coordinate system with respect to base platform is achieved. However, the position vector specified at origin of the moving platform denotes the translation vector with respect to the base platform.

$$\dot{Q} = \dot{\theta}_{\times}(Q^{\mathrm{T}})^{-1} = \dot{\theta}_{\times}Q. \tag{3.9}$$

Substituting (3.9) into (3.7) yields

$$\begin{aligned} r_i\dot{r}_i &= (p + Qb'_i - a_i)^{\mathrm{T}}(\dot{p} + \dot{\theta}_{\times}Qb'_i) \\ &= d_i^{\mathrm{T}}(\dot{p} + \dot{\theta}_{\times}Qb'_i) \\ &= d_i^{\mathrm{T}}\dot{p} + (\dot{\theta}{\times}Qb'_i)^{\mathrm{T}}d_i \\ &= d_i^{\mathrm{T}}\dot{p} + ((Qb'_i){\times}d_i)^{\mathrm{T}}\dot{\theta} \\ &= [d_i^{\mathrm{T}} \;\; ((Qb'_i){\times}d_i)^{\mathrm{T}}] \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix}. \end{aligned} \tag{3.10}$$

As mentioned, in the above equation (3.4) are used for the derivation in the second line and the derivation in the second last line, respectively. Noticing that $r_i = \|d_i\| > 0$ could be guaranteed by the mechanical structure, we have the following result

$$\dot{r}_i = \frac{1}{r_i}[d_i^{\mathrm{T}} \;\; ((Qb'_i){\times}d_i)^{\mathrm{T}}] \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} = \frac{1}{r_i}[d_i^{\mathrm{T}} \;\; ((Qb'_i){\times}d_i)^{\mathrm{T}}]\dot{\pi}. \tag{3.11}$$

For the six-dimensional vector $r = [r_1, r_2, \cdots, r_6]^{\mathrm{T}}$, we have the compact matrix form as follows

$$\dot{r} = A_1\dot{\pi}, \tag{3.12}$$

where

$$A_1 = \begin{bmatrix} \frac{1}{r_1}d_1^{\mathrm{T}} & \frac{1}{r_1}((Qb'_1){\times}d_1)^{\mathrm{T}} \\ \frac{1}{r_2}d_2^{\mathrm{T}} & \frac{1}{r_2}((Qb'_2){\times}d_2)^{\mathrm{T}} \\ \cdots & \cdots \\ \frac{1}{r_6}d_6^{\mathrm{T}} & \frac{1}{r_6}((Qb'_6){\times}d_6)^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \frac{1}{r_1}(p + Qb'_1 - a_1)^{\mathrm{T}} & \frac{1}{r_1}((Qb'_1){\times}(p - a_1))^{\mathrm{T}} \\ \frac{1}{r_2}(p + Qb'_2 - a_2)^{\mathrm{T}} & \frac{1}{r_2}((Qb'_2){\times}(p - a_2))^{\mathrm{T}} \\ \cdots & \cdots \\ \frac{1}{r_6}(p + Qb'_6 - a_6)^{\mathrm{T}} & \frac{1}{r_6}((Qb'_6){\times}(p - a_6))^{\mathrm{T}} \end{bmatrix}. \tag{3.13}$$

Equation (3.12) projects the kinematic relation of a Stewart platform with six-degree-of-freedom from the velocity of the pose variables to the speed of the legs.

3.3 Recurrent Neural Network Design

3.3.1 Problem Formulation from an Optimization Perspective

In this section, we introduce a numerical and non-linear [54] gradient decent optimization method to resolve the real kinematic parameters from the measurement data. The digital indicator are defined as measurement devices in order to rectify and verify the location of

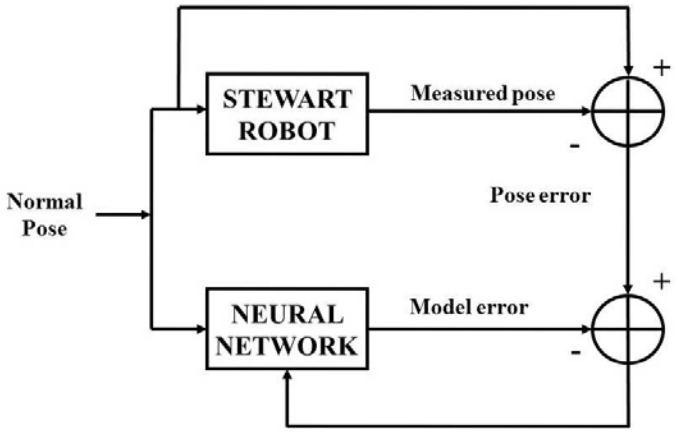


Figure 3.3: The control diagram of the Stewart robot using the proposed neural network.

end-effector of the stewart platform. This in turn can determine the error between the desired and actual locations. The equation corresponding to kinematics of parallel stewart platform can be expressed as follows

$$r_i = \|p + Qb_i' - a_i\| \tag{3.14}$$

where a_i representing the ith position vector in the mobile platform with respect to A and b_i representing the ith position vector with respect to the base B. r_i representing the ith link length. This equation denotes the peculiar inverse kinematic equation of a stewart platform. In general, it is infeasible to derive a forward kinematic model for the parallel manipulators due to non-availability of close form solution, and some numeral algorithms must be incorporated to derive the parameters of the forward kinematics.

The non-linear or non-convex relationship between r_i and π to the position variable is affine to $\dot{\pi}$. We observed in our analysis that $\dot{r}$ can be solved directly from compact matric form equation $\dot{r} = A_1\dot{\pi}$. The reference velocity constrained in reduced dimensions for the equality model is defined as

$$\alpha = A_2\dot{\pi} \tag{3.15}$$

where $\alpha \in \mathbb{R}^m$ is the reference vector with $0 < m < 6$. The pre-given transformation matrix $A_2 \in \mathbb{R}^{m\times 6}$. The optimized solution is defined as follows:

$$\min_{(\dot{\pi},\tau)} \frac{1}{2}\dot{\pi}^T \Lambda_1 \dot{\pi} + \frac{1}{2}\tau^T \Lambda_2 \tau \tag{3.16}$$

where the symmetric matrices $\Lambda_1 \in \mathbb{R}^{6\times6}$ and $\Lambda_2 \in \mathbb{R}^{6\times6}$ are both constant and positive definite, the speed of the platform legs to be control is denoted as$\tau = \dot{r}$. In practice, the term $\dot{\pi}^T \Lambda_1 \dot{\pi}$, which is in a quadratic form, specify the kinematic energy when choosing Λ_1 properly, and the term the input power can be characterized by $\tau^T \Lambda_2 \tau$. The formulation of the objective function is consistent with the convention of control theory in defining quadratic cost functions [75]. The actuator can directly change the value of the decision variable τ. Its value is under physical constraints, which are modeled as inequalities in the form below,

$$B\tau \leq b. \tag{3.17}$$

In this expression, $B \in \mathbb{R}^{k\times6}$ and $b \in \mathbb{R}^k$ with k as an integer. It is noteworthy that it is not imposed to the variable $\dot{\pi}$ for the constraint as in the planning stage it usually has already been specified. In summary of (3.16) as the object function, and (3.17) as physical constraints, and also with the nonlinear dynamic equation constraints, a constrained programming can thus be formulated to solve the control as mentioned in (2.23) Since there is a presence of two types of constraints, namely equality and inequality constraints it is not feasible to solve the optimization analytically. By incorporating traditional approaches it incurs extra penalty in terms formed by the constraints to the objective function. Hence to resolve the problem numerically using approach of gradient descent along the new objective function is also expensive. Hence we conclude that penalty based approaches are expensive and can only reach an approximate solution of the problem and therefore are not feasible to tackle error-sensitive applications [88]. Hence, it worths attempt to devise a dynamic differential equation in the type of neural networks to approach the solution iteratively.

3.3.2 Neural Network Dynamics

In this part, we present the neural network model used in this chapter. This is adynamic neural model that can be described by an ordinary differential equation. The dynamics is as

follows:

State equations:

$$\epsilon\dot{\lambda}_1 = -\Lambda_2^{-1}\lambda_1 - \hat{A}_1\Lambda_1^{-1}\hat{A}_1^{\mathrm{T}}\lambda_1 - \hat{A}_1\Lambda_1^{-1}A_2^{\mathrm{T}}\lambda_2 - \Lambda_2^{-1}B^{\mathrm{T}}\mu \tag{3.18a}$$

$$\epsilon\dot{\lambda}_2 = -A_2\Lambda^{-1}\hat{A}_1^{\mathrm{T}}\lambda_1 - A_2\Lambda^{-1}A_2^{\mathrm{T}}\lambda_2 + \alpha \tag{3.18b}$$

$$\epsilon\dot{\mu} = -\mu + (-B\Lambda_2^{-1}\lambda_1 + \mu - B\Lambda_2^{-1}B^T\mu - b)^+ \tag{3.18c}$$

Output equation:

$$\bar{\tau} = \hat{A}_1\Lambda_1^{-1}\hat{A}_1^{\mathrm{T}}\lambda_1 + \hat{A}_1\Lambda_1^{-1}A_2^{\mathrm{T}}\lambda_2 \tag{3.18d}$$

$$\tau = \bar{\tau} + w \tag{3.18e}$$

$$\zeta\dot{\hat{A}}_1 = -(\hat{A}_1\dot{\pi} - \tau)\dot{\pi}^{\mathrm{T}} \tag{3.18f}$$

where $\zeta > 0$ is a scaling factor. By replacing A_1 with $\hat{A}_1$ as obtained in (3.1), the whole system obtained so far can be expressed as

$$\zeta\dot{\hat{A}}_1 = -(\hat{A}_1\dot{\pi} - \tau)\dot{\pi}^{\mathrm{T}}. \tag{3.19}$$

Consider a special case when the initial value of $\hat{A}_1$ and π at time t=0 are both set a t 0. In this situation, the immediate derivative of state variables can be obtained as $\dot{A}_1 = 0$. Then

$$\zeta\dot{A}_1 = -(A_1\dot{\pi} - \tau)\dot{\pi}^{\mathrm{T}}). \tag{3.20}$$

Subtracting (3.2) from (3.3) yields

$$\zeta(\dot{\hat{A}}_1 - \dot{A}_1) = (A_1 - \hat{A}_1)\dot{\pi}\pi^{\mathrm{T}}. \tag{3.21}$$

Since, $\hat{A}_1 - A_1 = \tilde{A}_1$,

$$\zeta\dot{\hat{A}}_1 = (-\tilde{A}_1)\dot{\pi}\pi^{\mathrm{T}}. \tag{3.22}$$

The state variables of the neural network are plotted in fig. 3.4. These figures depicts the dynamic redundancy of neural network.

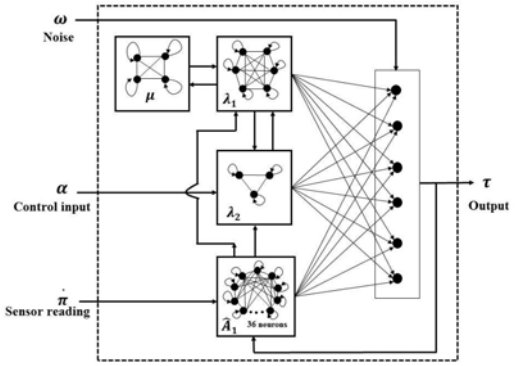


Figure 3.4: Architecture of the proposed neural network.

3.3.3 Stability

Stability is the most important issue in the dynamic systems. Non-stable systems may oscillate or even diverge. In this section we discuss the stability of the proposed architecture in namely learning convergence as follows.

Define $\hat{A}_1 - A_1 = \tilde{A}_1$ and use $V_1 = \|\tilde{A}_1\|_F^2 = \text{trace}(\tilde{A}^{\text{T}}\dot{\hat{A}}_1)/2$, where $\|.\|_F$ denotes the Frobenius norm of a matrix, as an estimation error metric. The time derivative of V_1 along the system dynamic is $\dot{V}_1 = trace(\tilde{A}^{\text{T}}\dot{\hat{A}}_1) = -\frac{1}{\zeta}trace[\tilde{A}^{\text{T}}(\hat{A}_1\dot{\pi} - \tau)\dot{\pi}^T]$

Note that the equality $\text{trace}(E(AB)) = trace(E(BA))$ for any A and B of appropriate sizes is utilized in the above derivation. Thus, $\dot{V}_1 = -\frac{1}{\zeta}trace(\tilde{A}\dot{\pi}^{\text{T}})$. For $C = \tilde{A}_1\dot{\pi}$ $\dot{V} = -\frac{1}{\zeta}trace(C^+C) = -\frac{1}{\zeta}\|\tilde{A}_1\dot{\pi}\|_F^2 \leq 0$. Note that $V \geq 0$ and is monotonically decreasing according to above equations.

We have

$$\tau = A_1\dot{\pi}$$

$$\tau = \bar{\tau} + w$$

$$\bar{\tau} + w = A_1\dot{\pi}$$

$$\dot{\pi} = A^+(\bar{\tau} + w).$$

We know that $\tilde{A}\dot{\pi} = 0$, we can multiply $\tilde{A}_1^{\text{T}}$, $(\bar{\tau} + w)^{\text{T}}$ and computing the expected value yields

$$E[\text{trace}[\tilde{A}_1^{\text{T}}\tilde{A}_1\tilde{A}_1^+(\bar{\tau} + w)(\bar{\tau} + w)^{\text{T}}]] = 0. \tag{3.23}$$

We note that since $(\bar{\tau}+w)(\bar{\tau}+w)^{\mathrm{T}} = E[(\bar{\tau}\bar{\tau}^{\mathrm{T}}) + w\bar{\tau}^{\mathrm{T}} + \bar{\tau}w^{\mathrm{T}} + ww^{\mathrm{T}}] \Rightarrow \bar{\tau}\bar{\tau}^{\mathrm{T}} + E(ww^{\mathrm{T}}) = \bar{\tau}\bar{\tau}^{\mathrm{T}} + \sigma^2 I$. These two equations upon equating result as follows: $E[\text{trace}[\tilde{A}_1^{\mathrm{T}}\tilde{A}_1(\bar{\tau}\bar{\tau}^{\mathrm{T}} + \sigma^2 I)]] = 0$

$= trace(\tilde{A}_1^{\mathrm{T}}\tilde{A}_1\bar{\tau}\bar{\tau}^{\mathrm{T}}) + \text{trace}(\tilde{A}_1^{\mathrm{T}}\tilde{A}_1\sigma^2 I)$, which further implies $\|A_1\tau\|_F^2 + \|\tilde{A}_1\sigma\|_F^2 = 0$.

3.3.4 Optimality

This part shows the optimal solution of the original optimization problem can be arrived at by converging to the equilibrium point of the dynamic neural networks (3.18).

Theorem 1. *For dynamic neural network (3.18) with $(\lambda_1^*, \lambda_2^*, \mu^*)$ as the equilibrium point, the output τ^* from (3.18d) is optimal to the constrained programming (2.23).*

Proof 1. *The equilibrium point $(\lambda_1^*, \lambda_2^*, \mu^*)$ meets the following condition according to state equations (3.18a):*

$$-\Lambda_2^{-1}\lambda_1^* - A_1\Lambda_1^{-1}A_1^T\lambda_1^* - A_1\Lambda_1^{-1}A_2^T\lambda_2^* - \Lambda_2^{-1}B^T\mu^* = 0 \tag{3.24}$$

$$-A_2\Lambda^{-1}A_1^T\lambda_1^* - A_2\Lambda^{-1}A_2^T\lambda_2^* + \alpha = 0 \tag{3.25}$$

$$-\mu^* + (-B\Lambda_2^{-1}\lambda_1^* + \mu^* - B\Lambda_2^{-1}B^T\mu^* - b)^+ = 0 \tag{3.26}$$

and the corresponding output is,

$$\tau^* = A_1\Lambda_1^{-1}A_1^T\lambda_1^* + A_1\Lambda_1^{-1}A_2^T\lambda_2^* \tag{3.27}$$

Define an auxiliary value,

$$\dot{\pi}^* = \Lambda^{-1}A_1^T\lambda_1^* + \Lambda^{-1}A_2^T\lambda_2^* \tag{3.28}$$

τ^* is optimal to (2.23) can be obtained by showing the satirisation of KKT conditions. Comparing the equation set composed of (3.24), (3.25), (3.26), (3.27), (3.28) we find that they are identical and therefore are equivalent. The above procedure implies that the solution (λ_1^*,

λ_2^*, μ^*, τ^*, $\dot{\pi}^*$) is optimal to (2.23). Therefore, we conclude that τ^* is optimal to the problem (2.23).

3.4 Numerical Investigation

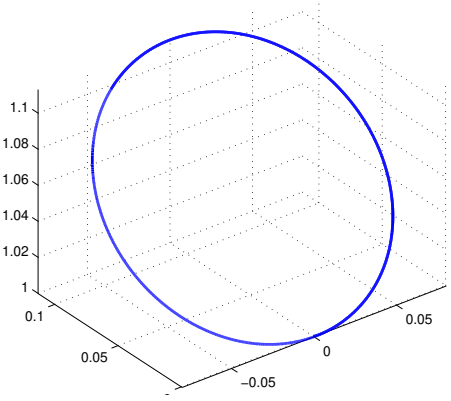
To demonstrate the efficiency and effectiveness of the proposed model free neural network approach applied to redundancy resolution of the stewart platform, we implemented it in MATLAB.

3.4.1 Setups

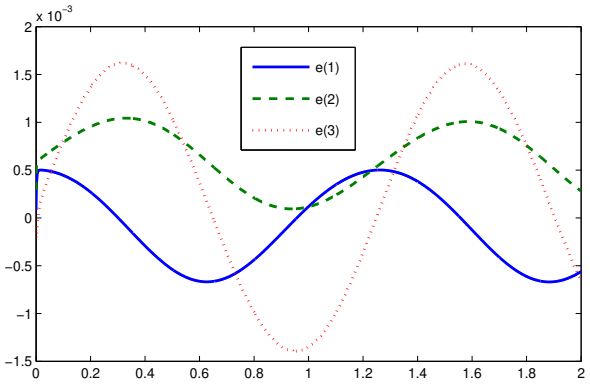
A Stewart platform with the following specifications will be considered: leg connectors are located around a circle with radius of 1.0 meter at $b_1' = [0.7386, 0.1302, 0]$, $b_2' = [0.7386, -0.1302, 0]$ $b_3' = [-0.4821, 0.5745, 0]$, $b_4' = [-0.2565, 0.7048, 0]$, $b_5' = [-0.2565, -0.7048, 0]$, $b_6' = [-0.4821, 0.5745, 0]$ are specifically the platform coordinates and the leg connectors located around the circle with radius 0.75 metres at $a_1 = [0.3750, 0.6495, 0]$, $a_2 = [0.3750, -0.6495, 0]$, $a_3 = [-0.7500, 0.0000, 0]$, $a_4 = [0.3750, 0.6495, 0]$, $a_5 = [0.3750, -0.6495, 0]$, $a_6 = [-0.7500, 0.0000, 0]$ are placed on the fixed base. For the ease of simulations the end effector is rotated and placed at origin with respect to the platform coordinate. The total expected redundancy is assumed to be 3, for the position tracking in the 3-D space. The input and output dimensions are 6 and 3, respectively.

The desired angular motion speed is set as 0.2 rad/s. The control scaling factor ϵ of the neural model is set as $\epsilon = 10^{-2}$ and the learning scaling factor ζ is set as $\zeta = 10^{-4}$. The excitation signal w is set as random noise with zero mean and deviation of 10^{-3}. The basic idea is to set the noise at small value to ensure a minimal impact to the system performance.

In the simulation, we consider two tracking trajectories, namely, a square path and a circular path.



(a) Trajectory of the end-effector



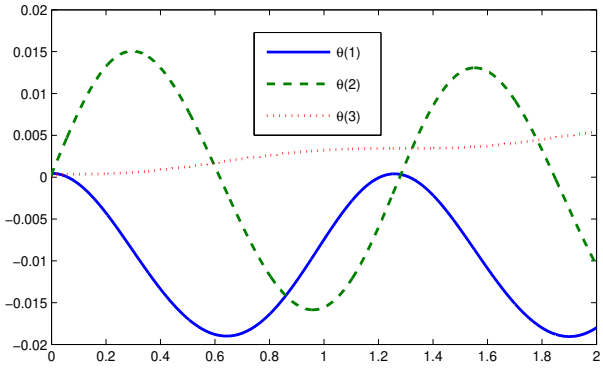
(b) Time history of the position of tracking error

Figure 3.5: Tracking of a circular motion.

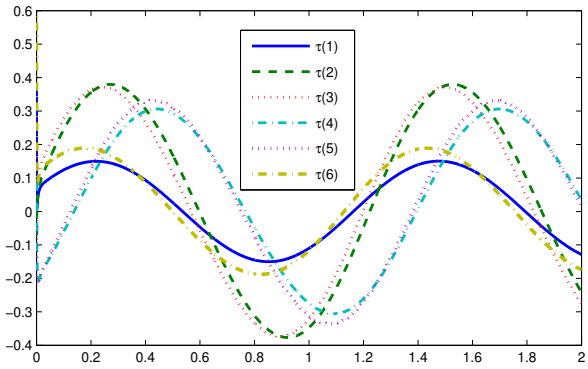
3.4.2 Circular Path

In this section we simulate the tracking of smooth path around circular trajectory using the model-free approach. It is desired to follow the path of a circle at the minimal speed of $2m/s$. The circle is centered at $[-0.04, 0.06, 1.05]$ with an radius of 0.08m, and has a revolutionary angle around the x-axis for 45 degrees. In the simulation setup, Λ_1 and Λ_2 are chosen as identity matrix. The values of the matrix A_1 is computed in real time accordingly. The matrix A_2 is chosen as $A_2 = [I_{3x3}, 0_{3x3}]$ so that the position tracking requirements are obtained. In real time, the legs actuation speed is limited to a certain range as it is associated with the real time constraints of the actuators.

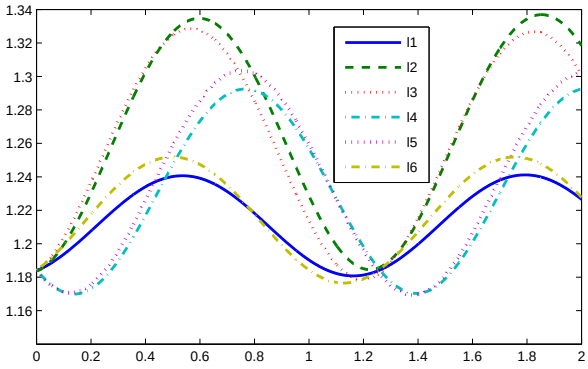
The results are obtained by executing the simulation for 2 s. The Fig. 3.5(a) shows the



(a) Orientation of the platform θ

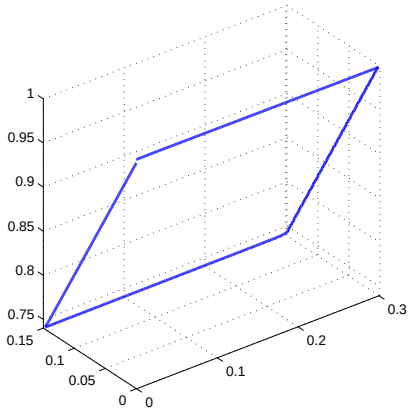


(b) Control action τ

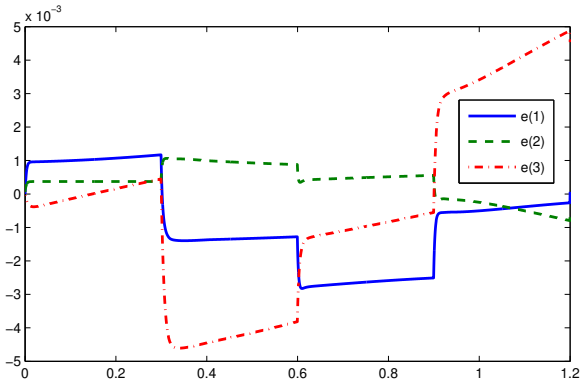


(c) Leg length r

Figure 3.6: Stewart platform state variables at different time in case of circular motion tracking.



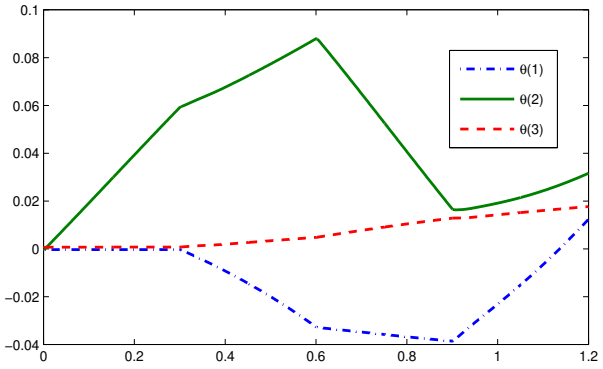
(a) Trajectory of the end-effector



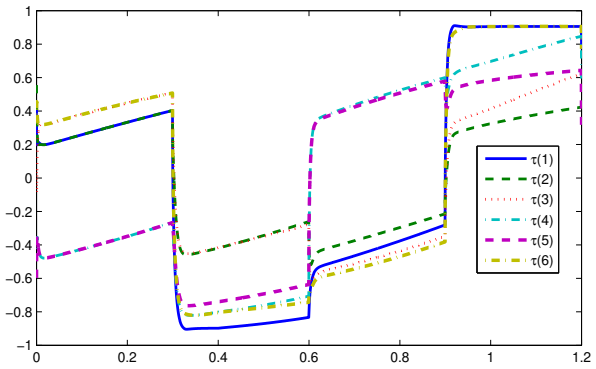
(b) Time history of the position of tracking error

Figure 3.7: Tracking of a square motion.

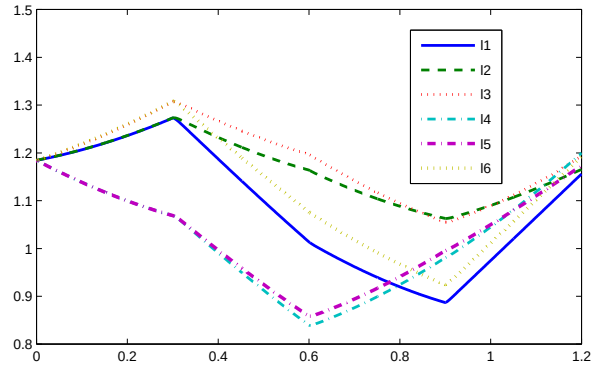
completed tracks in circular motion of an end effector with a least tracking error as shown in Fig. 3.5(b). The position tracking error components $e(1), e(2), e(3)$ are plotted against x, y and z axes of the base frame of the platform. The errors depicted from the figure are less than 0.015 m in terms of the amplitude. This path tracking simulation demonstrated the capability of the proposed model for resolving the kinematic redundancy of the physically constrained Stewart platform. The input motion for the legs are shown in Fig. 3.6 which depicts the time evolution of the Stewart platform state variables, e.g., 3 Euler orientations of the platform, the position of the end effector and the leg speed and its related length. The coordinates $p(1)$ and $p(2)$ of the attached moving frame starting from zero and $p(3)$ varies between 0.005 and -0.006 m. Fig. 3.6(c) states the bound ± 0.25 ($\eta = 0.25$) for action speed.



(a) Orientation of the platform θ



(b) Control action τ



(c) Leg length r

Figure 3.8: Stewart platform state variables at different times in case of square motion tracking.

It is to be observed that τ converges within the boundary region of [-0.25, 0.25].

3.4.3 Square Trajectory

In this section we will discuss the simulated results of the square trajectory incorporating our model-free approach, which are shown in Figs. 3.7 and 3.8. In circular motion the path traveled by the end effector in 360^o. However, in square trajectory the path is non-smooth when switches from one straight line to the next and how to reach timely control becomes a challenging issue. The end effector follows the trajectory which is places in the center at [0.15, 0.075, 0.74] where the length of the edge is defined as 0.08 m, at 1.0 m/s as the speed. The revolute angle of the square around the x-axis is 60 degrees. The chosen parameters Λ_1 and Λ_2, A_2 and B have the similar values as that mentioned for the circular motion. The parameter A_1 is computed in real-time. The limit of speed $\eta = 0.6$ m/s and $b = 0.6\mathbf{I}_{12}$ and $\epsilon = 0.001$. The tracking results for the square trajectory is shown in figures below (position tracking error vector [$e(1)$, $e(2)$, $e(3)$], which is very small).

3.5 Conclusions

A model free dual neural network is proposed to investigate and resolve the redundancy resolution problem of manipulators. In this chapter we also establish dynamic model for an model-free network which is designed for a general case of modular manipulator. The proposed controller model is designed such that priori knowledge is not required for dynamic parameters and can suppress bounded external disturbances effectively in presence of the external noise added. The instability problem caused by self-motions of the redundant robot can be resolved by the presented dual-neural control algorithm. Theoretical results are presented to verify the stability of the proposed models. The simulation is carried out on a redundant manipulator, which has verified the effectiveness of the dynamic modeling method and the controller design method. In our future work, the following aspects will be considered:

1. In this chapter, we only considered deterministic uncertainty that can be learned. In applications, there may exist some non-deterministic uncertainties that demonstrate stochastic or probabilistic properties that cannot be learned directly. We consider to use robust control strategy to optimize the control parameters such that the L2 gain from the non-learnable uncertainty to the resulting control error could be minimized.

2. The controller developed in this chapter is in a continuous time form. In implementation with micro-controllers, the control actions need to be truncated into discrete time. However, the selection of a proper sampling period for control becomes an issue. To resolve this problem, we consider to employ event-triggered control mechanism to design an improved version of this control for adaption of the sampling period.

3. One trend in the research of parallel robots is to design a hybrid structure with both serial and parallel mechanisms involved. How to extend the current results to hybrid mechanisms remains to be a challenging issue.

4. This chapter considers a single Stewart platform. For complicated tasks, more than one platforms may be required and the coordination of them emerges to be a challenging issue.

CHAPTER 4

STEWART PLATFORM CONSTRUCTION AND ALGORITHM VERIFICATION

4.1 Design a 6 DOF Rocking Chair Mechanism for Rocking Chair

This chapter deals with the design of a driving chair simulator for real-time simulation time. Under this proposal, the intention is to build the simulator and it will be necessary to carry out some other strength analysis based on the dynamics of the moving part, which, as shown in Fig. 4.1. In the future, we plan to improvise the control units, rocking chair behavior and also the rider under certain conditions and possibly for simulation and analysis for minor accidents.

A power-driven 6-DOF rocking chair is designed for people with a sleeping disorder that enables them to fall asleep in short duration is one of the medical applications. The rocking chair, in turn, can be classified as a therapy. Furthermore, the power-driven rocking chair also provides people (elderly, adults, kidult and new-born) whose act of sleeping is inconvenient, a comfortable, economic and portable moving means. Generally, the base platform of the power-driven rocking chairs has fewer degrees of freedom. The motivation leads to less applicability when the rocking chair is operating in different terrains. To adapt different criteria (in terms of flexibility, portability, and weight carrying capacity) and promise a good comfortable and convenient performance, we propose the design of the 6 DOF rocking chair.

4.1.1 Evaluation of existing solutions and selecting an appropriate variant

Based on a search in the related parts of the work we can find out that there are currently differences in the implementation of driving simulators with varying degrees of freedom.

The simplest and the most elaborate are the significant differences in the complexity of the construction. Generally, the higher the degree of the simulator's freedom, its construction is more complicated, its dimensions and total weight increase. This results in a higher price and a more demanding management of the movement system. Usually, the weight is increased at the same time as the moving mass, which increases the energy needed to perform the motion and therefore requires a more powerful movement system (motors).

As mentioned above, various simulators use a different approach to modulate the movement of the human mind. Each variant has its advantages and disadvantages, and there is probably no clear answer to the question of which one is better or the most simulating reality. It can be argued that the most advanced driving simulators (National Advanced Driving Simulator (NADS) and so on) use the widest possible anabolic mobility options (hexapod, which is optionally dop). However, as reported by researchers, according to some studies, the use of fewer than 6 degrees of freedom in a driving simulator could be more appropriate in terms of more realistic simulation experience. Even in the field of aerospace simulators, there is no clear view of the need to use all 6 stages of freedom [29]. It depends on the specific purpose and task to which the simulator is intended, and of course, no movement is always better than a poorly tuned motion.

As we planned to build a simulator using the financial resources of the department, it would be inconceivable to deal with too complicated design solutions under the given conditions and taking into account the scope of this thesis. Taking into account the above circumstances, a variant of the "seat" simulator was chosen. This is a 6-degree simulator with a movable seat and a stationary rest the frame on which the controls and the screen are clamped. Its advantage is simple construction, relatively simple control of the movement system, small dimensions that allow its operation, for example, in the home, office and the low weight of the moving mass, which reduces the demands on battery performance.

(a) The skeleton of a rocking chair

Figure 4.1: The mechanical construction of a rocking chair Skeleton.

4.1.2 Building a Stewart Platform

The motivation for building this robot is to connect the digital world to the exciting physical world. This chapter describes the 3D models and the necessary equipment and technique required to create the Stewart platform.

Stewart platforms find ways in numerous industry and social applications like flight simulators, civil constructions, sorting of vehicles, and many other fields and for various purposes.

A Stewart platform is usually defined as a parallel robot with six degrees of freedom (6 DOF). The six degrees of freedom describes its movement and direction. The movement along the three axes and the rotation along with the three directions. The axis is denoted as X, Y, and Z and the directions are referred to as yaw, roll, and pitch.

4.1.3 Building Material

Ordering, building and testing the parts and material is the most time consuming and challenging task. The building material could be classified as into three main streams;

- The mechanical parts in the construction of the Stewart Platform (Table 4.1)
- The electronics; including relays, servo motors, hinges, boards, plates, wiring (Table 4.2)
- The computer for programming, controlling and testing the platform (Table 4.3)

The estimated cost of all the required material and equipment is approximated $10k$ Hong Kong dollars. Due to the presence of a 3D printer, the cost is minimized. The 3D printer helps in printing the brackets and other required hinges for connecting actuators.

4.1.4 Mechanical Equipment

The mechanical equipment required for constructing the Stewart platform includes the 2 platforms; top plate and bottom plate, actuators (acting as controlling rods) connecting servo motors and the 3D brackets connected as hinges and joints.

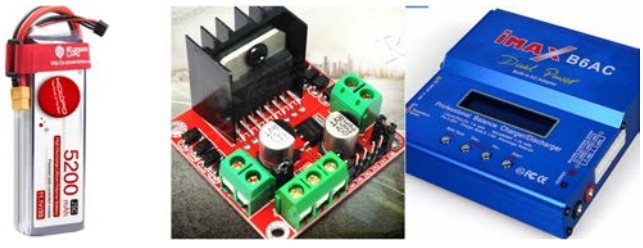
The estimated market price of this mechanical equipment is approximated $2k$ Hong Kong dollars.

4.2 Creating a Solid Works Model

This section deals with the creation of a Stewart platform for subsequent export into one complete package.

(a) Parts of Stewart Platform

(b) The Leg of the Stewart Platform



(c) Power Supply

Figure 4.2: The material and equipment for Stewart Platform.

Table 4.1: Top and Base Plate of Stewart Platform

S.No.	Qty	Name	Description
1	20	Aluminum bar	20 bars of 0.5m with 20X20mm for the intersection
2	20	Aluminum connection hinge	20x20mm Angle Bracket Aluminum alloy
3	20	Aluminium board	200*250*2mm aluminum alloy sheet aluminum plate
4	100	Slider on the aluminum bar	aluminum pre-installed nut T-type
5	20	Connection hinge	Narrow strip hinge multi-purpose
6	30	Connector	Corner fittings right angle connector
7	100	Screws	special round head bolt screw
8	200	Screw washer	cylindrical head Hexagon socket bolts

Table 4.2: The Leg of Stewart Platform

S.No.	Qty	Name	Description
1	10	Linear actuator	20 bars of 0.5m with 20X20mm for the intersection
2	20	Linear actuator connector	rod bracket for motor mounting
3	50	Bearings	Edge bearings ball bearings
4	20	distance sensors	Infrared distance sensor 10 80cm

4.2.1 Selection of suitable design

When creating a simulation model, several factors have to be taken into account. The model should not be too complicated (may not contain all the elements that are part of a real machine). At the same time, however, all factors should be taken into account affecting its functionality.

Table 4.3: Power Supply

S.No.	Qty	Name	Description
1	1	Charger	Multi-function battery charger
2	2	Batteries	5200mah lithium battery
3	20	Drivers	L298N motor drive board module

4.2.2 Linear actuator

The main part of Stewart's platform is his shoulders/legs. These consist of joints and linear actuators to ensure their movement. It can be implemented in several versions:

- Hydraulic or pneumatic drive
- Linear electric motor
- Mechanical drive (conversion from rotary motion to sliding motor movement of the matrix)

As a model, the DLA 200 linear actuator has been selected from the company Transmotec. It is an actuator driven by a DC electric motor, upon rotation whose movement is converted to a sliding ball screw. The conversion between the engine and the screw are made with a gearbox fitted with aluminum gears.

4.2.3 Joints

As mentioned above, the shoulders are connected to the platform using joints. Attach the knee joint to the base. A joint GLK 2 (Gelndewagen Luxus Kompaktklasse) Fig. 4.3 was chosen which has the possibility of moving in one axis $\pm 45^{\circ}$ and in the other $\pm 90^{\circ}$, which is more than sufficient for this model. When calculating degrees of freedom, we considered the second link as a ball joint, but to simplify the design and use the cardan joint design which

(a) Prismatic Actuators - The skeleton of

a rocking chair

Figure 4.3: Linear Transmotec Actuator

is identical to the GLK 2 model, thereby using a combination axial radial bearing which enables rotation of the third axis.

4.2.4 Platform and Base

When designing a platform and base shape, we try to place adjacent joints closer to each other, so that the structure approaches the octave as suggested by Gough. We will choose the shape of Stewart's platform from the French of the Symmetry company, which is visible in Fig. 4.1.

4.2.5 Model Actuator

For simulation purposes, several adjustments were made to the actuator model. For simplicity, gearboxes have been replaced by gears toothed belt transmission. Also, most of the components were omitted that are not necessary for the actuator function.

4.2.6 User position in the simulator

Of course, ergonomics also plays an important role in designing the simulator. The goal is to secure its convenient control for people of different ages. As has already been mentioned, the effort is adapted to control the chair seat for user's comfort. For maximum convenience, it is advisable to observe certain ranges of these angles, which are defined by standards in the field of ergonomics [37].

4.3 Background of the Invention

This section describes the need for the invention of automated rocking chairs and the market demand value and supply needs.

4.3.1 Open Challenges

50-70 million US adults have a sleeping disorder according to the Institute of Medicine. Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem. [1]

The report [1] highlights sleeping deprivation statistics:

- 37% of 20-39 year-old reports short sleep duration
- 40% of 40-59 year-old reports short sleep duration
- 35.3% adults reports < 7 hours of sleep during a typical 24-hour period.

Over 1,00,000 deaths occur each year in US hospitals due to sleep deprivation, which has been shown to make a significant contribution.

4.3.2 Expected Outcomes

Rocking chair therapy has a positive effect on the mental and physical well-being inpeople with dementia [84] according to a Journal of Alzheimers Disease. Research studies of Rock-

ing chairs show a measurable increase in well-being in people suffering from dementia and also prove to be effective for anti-anxiety medication. Improved quality of life without the need for medication is yet another addition to the development of the rocking chair. Rocking Chair Therapy contributes to a calmer environment for dementia sufferers and proves out to be a quicker recovery factor [84]

4.3.3 Market Demand

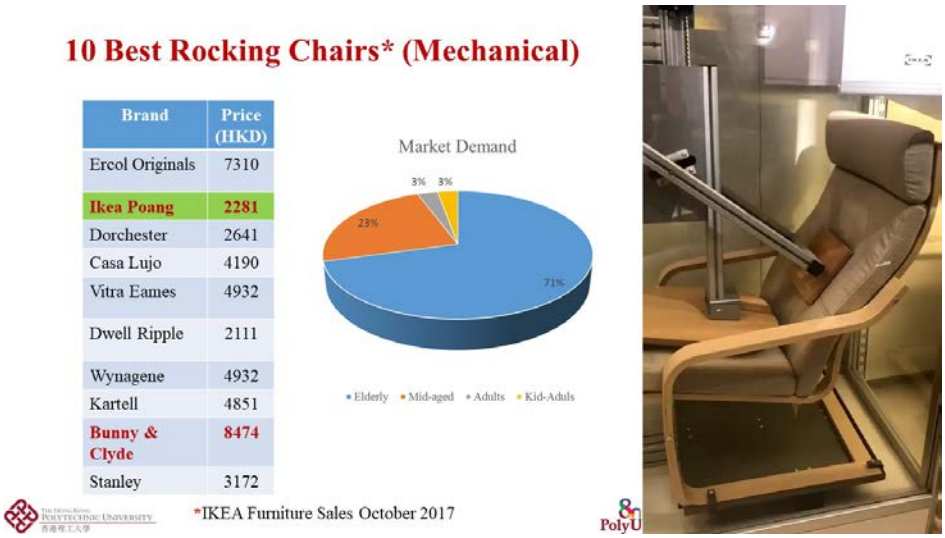
Rocking chairs are broadly classified into mechanical and electronic rocking chairs. The mechanical rocking chairs have a strong market than electronic ones due to high maintenance cost, durability, relativity, and portability.

The mechanical chairs are usually made from materials like cut-wood, bent-wood, metal, laminated constructive material, a combination of paper and composites, etc. The biggest competition market for mechanical chairs is IKEA. Fig. 4.1 shows the top 10 market chairs in the market and their market price. The figure shows the market demand of the rocking chairs where 71% is captured by the elderly and 23% is captured by the mid-aged group. This is viable from the figure that these chairs have high demand ranging from elderly to kidults as presented in Fig. 4.5.

4.3.4 Our Invention Claims

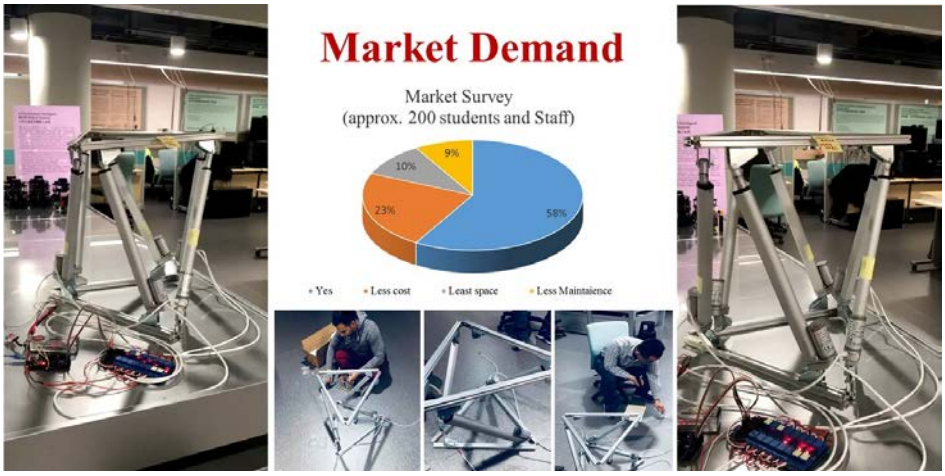
An automated wellness rocking chair is developed. We use the term “wellness” as this chair also contributes to therapy in the field of medical engineering. The model pose is shown in Fig. 4.6. The automated rocking chair consists of;

- a ground contraction base for supporting and maintaining the balance of the cushion seat
- the joints mounted at the six legs/shoulders are apex uppermost to the rocking chair
- a pair of straight flat-metal reinforced ball joints for movement and rotation



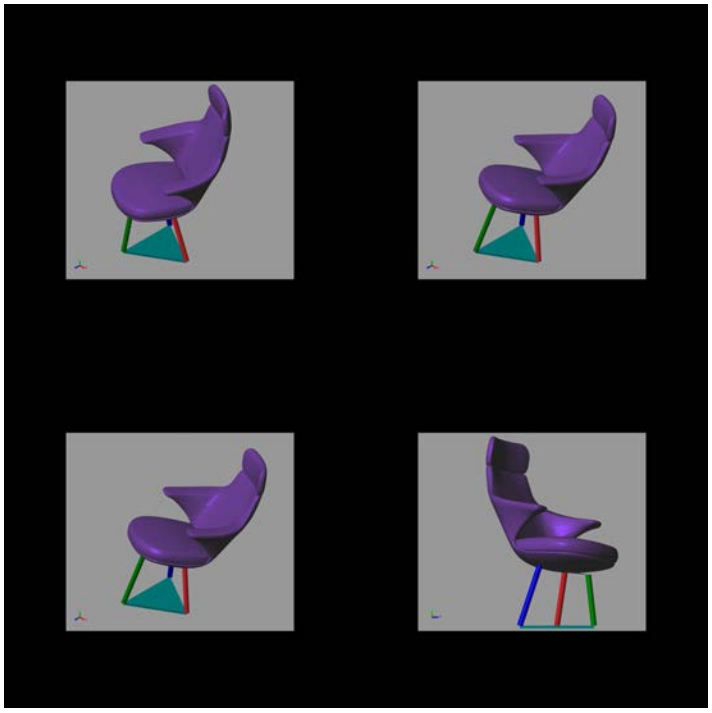
(a) IKEA Rocking chair

Figure 4.4: Mechanical rocking chairs



(a) The skeleton of a rocking chair

Figure 4.5: Market demand



(a) 3-legs driven model

Figure 4.6: Model pose

- affixing the top and bottom ends of these joints to rearward and forward shoulder surfaces which connects the top of the box seat

4.4 Experimental setup discussion and results

In this section, we describe the stages of testing and the modeling of the setup followed with discussions made in section 4.2. The rocking chair platform is controlled via Bluetooth using Arduino and working with relays.

To observe the movement and rotation of the shoulders a sine wave signal is passed as the input of the virtual prototyping rocking chair model and the speed of the chair was maintained between $0.1 to 0.5$ Km/h. The chair produces 2-DOF vibration, acceleration (vertical) and pitch.

The platform is confirmed with the loading capacity and the maximum requirement

of the load that can be implanted on the chair that generates the required movements. The idea is to test the capability of the chair that can hold the maximum weight of a human. As an average person weight between $65 to 85$ Kg. In the process of testing the seating capacity, the weight is determined by placing a 100 Kg load given 1 Hz frequency sine signal. When the frequency of the signal is increased (up to 5 Hz) the platform still handles a higher weight capacity. This is because of the shoulders electric cylinders.

In the experimental results, we applied various weights as inputs to test the strength of the rocking chair. These weights inputs will lead our study of the platform vibration in different conditions of weight applications. The results of the real vibration will be compared to that of the existing real-time Stewart platform and thus the outcome will be analyzed and compared with the smoothness of the rocking chair movement.

4.4.1 Simulation Results

In the initial stage, we applied a pure sine wave signal to the platform as shown in Fig. 4.7 as the input. The speed of the moving platform is fixed to 20 km/h. All the signals are transmitted via Arduino and in connection with the relays attached to the platform. The rocking chair platform gives a 2-DOF vibration output with vertical acceleration and pitch. The output of the platform is sketched in Fig. 4.8. It is observed that the number value decreases in the case of pitch acceleration. However, there is a gradual increase in comparison to the resonance point.

A point of reference is highlighted in the above figure where there is continuous all time increase in the vertical acceleration. Fig. 4.9 shows this vividly. These vibrations and the output waves are the results of computer simulation.

The vibrations produced by the moving rocking chair Stewart platform can be measured by using a 6-DOF Inertial Measuring Unit sensor. The following figures Fig. 4.10 and Fig. 4.11 captures and highlights the real vibrations produced by the platform through simulation. This shows a similarity between the two vibrations. The sensor produced a

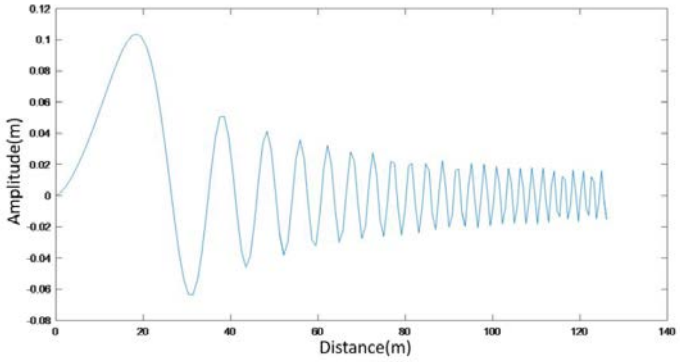


Figure 4.7: Sine wave signal as input model

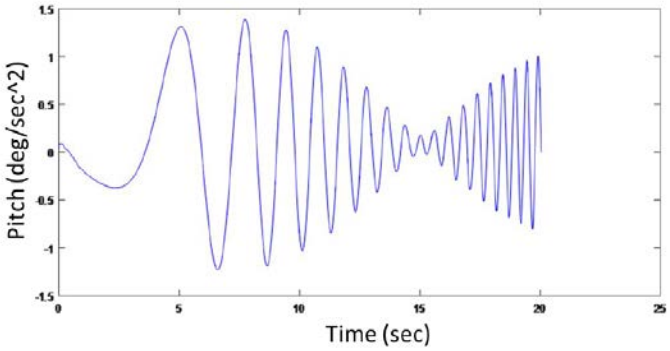


Figure 4.8: The pitch acceleration produced by mechanical platform

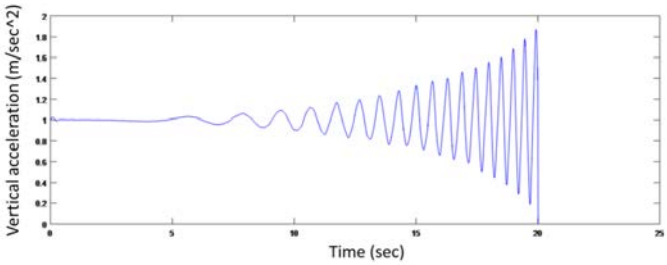


Figure 4.9: The vertical acceleration produced by moving platform

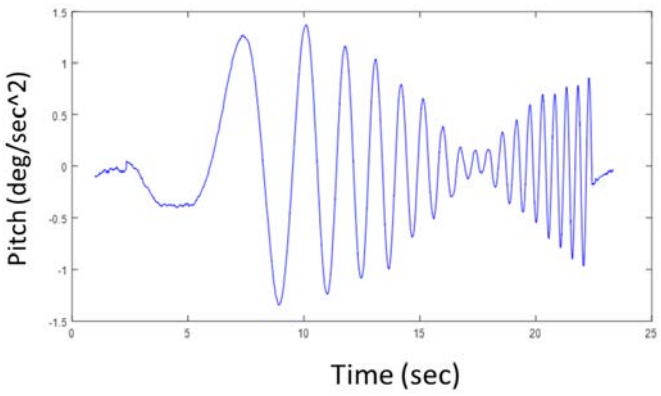


Figure 4.10: The pitch acceleration produced by moving platform

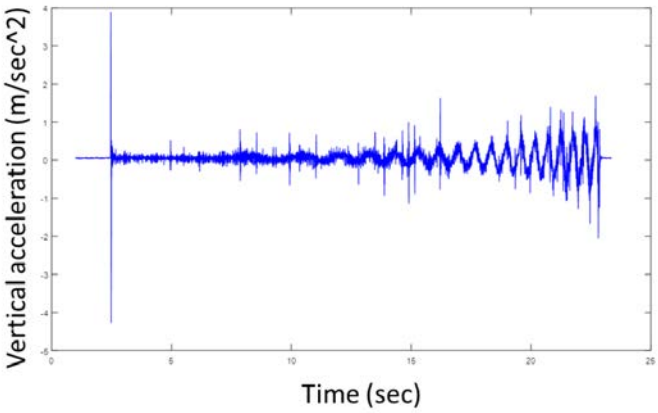


Figure 4.11: The pitch acceleration produced by moving platform

noisy signal wave on the vertical acceleration, it can be easily sighted that the obtained curve matches to that generated by computer simulation.

The vibrations of the mechanical rocking chair with speed of 20km/h movement by a 70kg person are recorded and the simulation results can be seen from the graphs below in Fig. 4.12 and Fig. 4.13. These two figures show the acceleration in terms of vertical and pitch acceleration.

We also study the vibrations of the automated rocking chair controlled via a handheld mobile device and the vibrations of the movements are displayed in Fig. 4.14 and Fig. 4.15. The results of the mechanical chair to that of the automated chair show that the actual movement of the chair matches that of the automated chair. Thereby proving that an automated

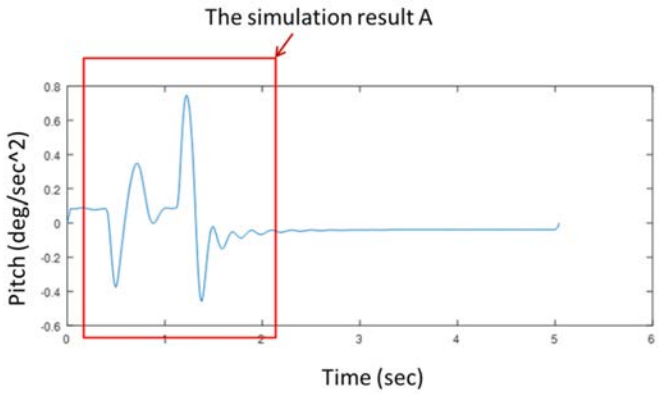


Figure 4.12: The pitch acceleration produced by mechanical platform

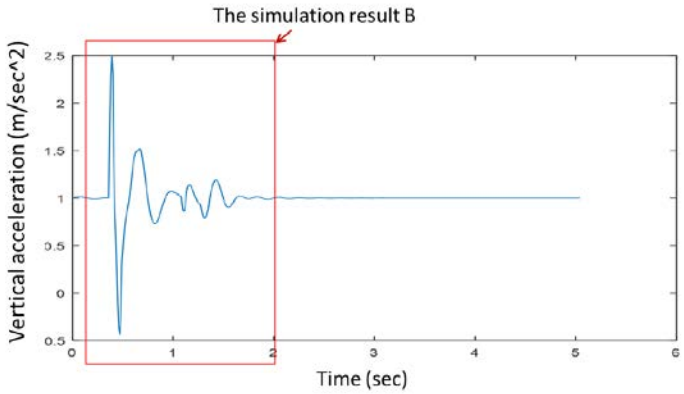


Figure 4.13: The vertical acceleration produced by moving platform

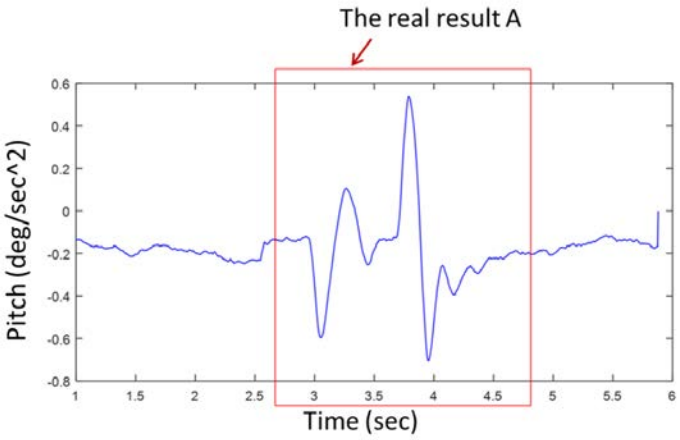


Figure 4.14: The pitch acceleration produced by moving platform

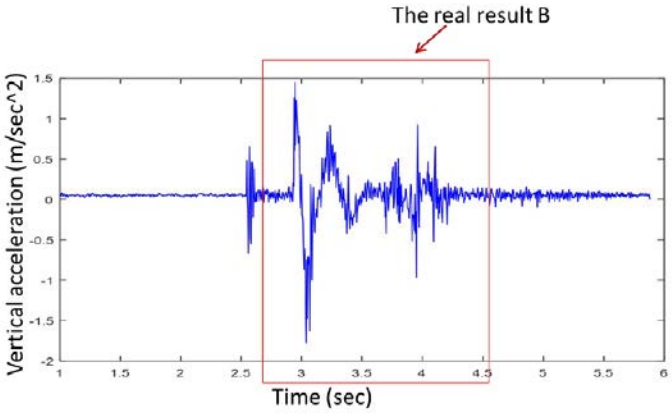


Figure 4.15: The vertical acceleration produced by moving platform

chair is as effective as a mechanical chair. However, there are many added advantages to the automated chair like flexibility, automation, portable, etc.

The results above also highlight that the rocking chair Stewart platform can produce the necessary vibrations according to the movement of the human body (input value). This is to say, that the quality of the rocking chair control system depends on the accuracy of the simulation model and to that of the design of the Stewart platform that can precisely measure the vibrations of the moving chair.

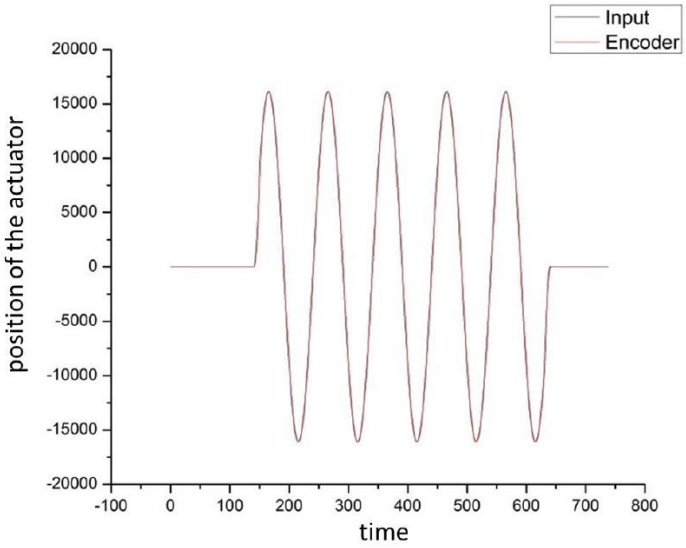


Figure 4.16: The output results of load capacity comparison

4.4.2 Loading rocking chair platform and studying reaction and capacity

In this section, we study if the designed platform is capable of meeting the sitting load capacity of an individual human body ranging from a newborn to an elderly where the weigh deviates from 10 pounds to 180 pounds. We also study the required vibrations necessary for movements thereby providing the comfort of sleeping.

In the process of load testing of the seat, the weight of approximately 220 pounds was tested to see the upper limit of the platform. After a weight of 220 pounds is applied with the frequency of 1 HZ and 7 to 8 mm sine wave amplitude, the rocking chair vibration, and the generated input signal appear to overlap as shown in Fig. 4.16

We also experimented to change the frequency signals to say 5 HZ and provided a sine wave single of 2mm the sine wave signals for the moving platform overlap perfectly with each other. The vibration is assumed to be the encoder output. This is reflected in Fig. 4.17.

In the next level, the data values deviate for higher number values of 7 HZ with a sine wave signal of 3mm as in Fig. 4.18. The waves in the figure dont overlap perfectly to that of the generated input. The vibration produced by the moving platform was lagging behind the

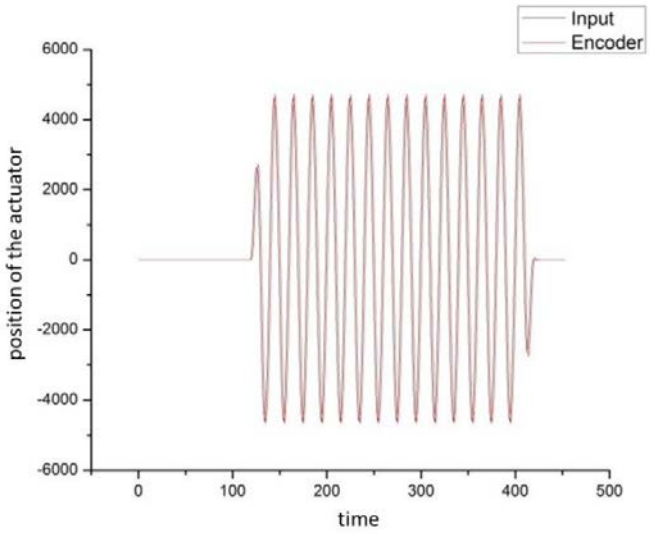


Figure 4.17: The output results of load capacity comparison - 5HZ and 2mm

input sine wave.

This is to conclude that the weight of the load on the moving platform plays a vital role in giving a satisfactory performance of the platform. The reason for these deviations is due to the cylinder/actuator of the platform that could not tolerate the rated power provided as input.

The generated results lag behind the input sine wave signals in the frequency domain as drawn in Fig. 4.19. However, upon inspection, it is observed that the amplitude of the vibration produced by the rocking chair platform was considerably lesser than that of the input sine wave signal.

4.5 Remarks

Rocking chairs are usually designed not only to provide comfort but to also provide to relax the lower limb areas of the people; especially the elderly. The research studies show that the fundamental designs of both automated rocking chair and mechanical rocking chair are neither fully adapted nor do they offer the same benefits. The rocking chair manufacturers in the United States are of mass production for imports and exports, thought these lack of

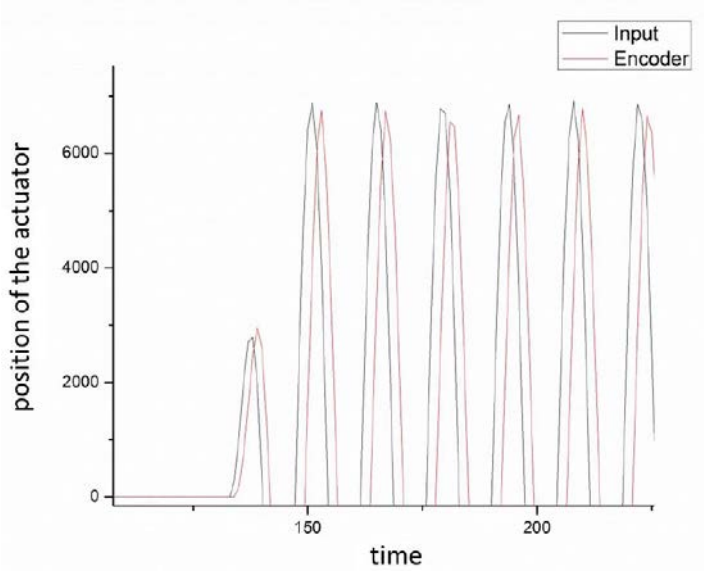


Figure 4.18: The output results of load capacity comparison - 7HZ and 3mm

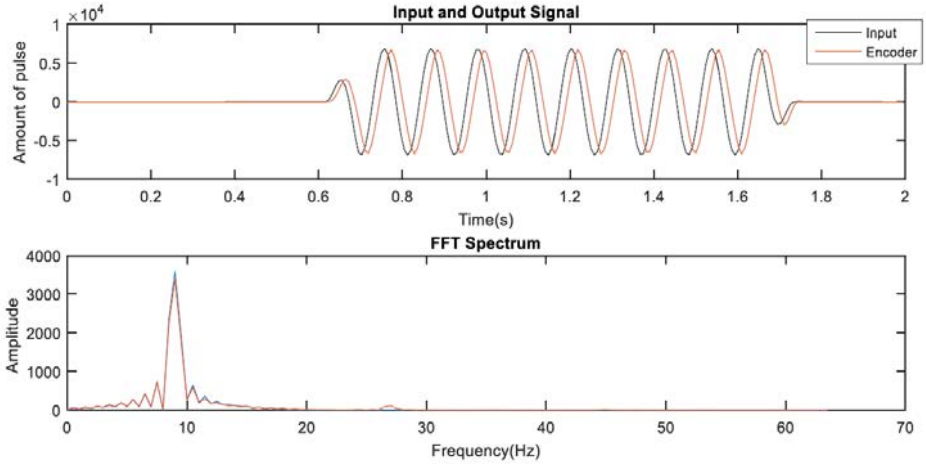


Figure 4.19: The output and input results comparison in frequency domain

research and participation of clinical therapists and mechanical engineers. There are still many open challenges in the field of rocking chairs and moreover, these automated rocking chairs prove to be more efficient, reliable and cheaper than mechanical rocking chairs. The main purpose of the study on the automated rocking chair is to design new mechanisms for the existing rocking chairs in the market. With the present research of mechanisms, rocking chair advantages can overcome simple daily obstacles that help the elderly and people with sleeping disorders to fall asleep in a shorter duration. In this chapter, we demonstrate the 6-DOF rocking chair automated mechanism and the workspace analysis in the above chapters demonstrates that the mechanisms can be good design tactics. This design can be used as a benchmark resource for the designers to update the automated rocking chair.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

- In the first scheme, a dynamic neural network is designed to solve the redundancy resolution problem of Stewart platforms. The physical constraints and actuation are rigorously modelled as a constrained quadratic programming problem. To solve this problem in real time, a recurrent neural network is proposed to reach the equality constraints, inequality constraints, and optimality criteria simultaneously. Rigorous theoretical proof are supplied to verify the convergence of the proposed model. Simulation results validate the effectiveness of the proposed solution.

- For second scheme, a model free dual neural network is proposed to investigate and resolve the redundancy resolution problem of manipulators. In this thesis we also establish dynamic model for an model-free network which is designed for a general case of modular manipulator. The proposed controller model is designed such that priori knowledge is not required for dynamic parameters and can suppress bounded external disturbances effectively in presence of the external noise added. The instability problem caused by self-motions of the redundant robot can be resolved by the presented dual-neural control algorithm. Theoretical results are presented to verify the stability of the proposed models. The simulation is carried out on a redundant manipulator, which has verified the effectiveness of the dynamic modeling method and the controller design method. In our future work, the following aspects will be considered:

  1. In this work, we only considered deterministic uncertainty that can be learned. In applications, there may exist some non-deterministic uncertainties that demonstrate

stochastic or probabilistic properties that cannot be learned directly. We consider to use robust control strategy to optimize the control parameters such that the L2 gain from the non-learnable uncertainty to the resulting control error could be minimized.

2. The controller developed in this thesis is in a continuous time form. In implementation with micro-controllers, the control actions need to be truncated into discrete time. However, the selection of a proper sampling period for control becomes an issue. To resolve this problem, we consider to employ event-triggered control mechanism to design an improved version of this control for adaption of the sampling period.

3. One trend in the research of parallel robots is to design a hybrid structure with both serial and parallel mechanisms involved. How to extend the current results to hybrid mechanisms remains to be a challenging issue.

4. This work considers a single Stewart platform. For complicated tasks, more than one platforms may be required and the coordination of them emerges to be a challenging issue.

5.2 Future Work

The work presented in this thesis can be extended in different directions in the future.

- First, Stewart-Gough platform is an important mechanism of studying the dynamics of the system. How to combine our approach to effectively do redundancy resolution of multi-degree of freedom of parallel stewart platform can be a future direction for us to explore.

- Second, our approach is how kinematic control problem of Stewart platforms is formulated to a constrained quadratic programming based on the KKT conditions. We will extend our approach to other non-constrained conditions. We will explore how to integrate these into rocking chair mechanism to further improve their performances.

- Third, tracking control of the Stewart-platform with multiple edges is critical for erros.

We will further explore how to integrate to customize new effective dynamic motion with least errors.

- Finally, efficiently control the motion of redundant parallel manipulators, especially in the situation with parameter uncertainties, or even unknown, sets great practical significance but also remains as a challenging research problem. In the future, we will extend the Stewart design to make its translation more efficient and complete.

REFERENCES

[1] Bruce M Altevogt, Harvey R Colten, et al. Sleep disorders and sleep deprivation: an unmet public health problem. National Academies Press, 2006.

[2] Helon Vicente Hultmann Ayala and Leandro dos Santos Coelho. Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator. Expert Systems with Applications, 39(10):8968–8974, 2012.

[3] Joshua Barnfather. Using" process-to-part" techniques in large nuclear power plant component manufacturing to improve supplier competitiveness.

[4] Ilian Bonev. Delta parallel robotthe story of success. Newsletter, (4):1–8, 2001.

[5] Ilian Bonev. The true origins of parallel robots. ParalleMIC, 2003.

[6] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, March 2004.

[7] Fan-Tien Cheng, Tsing-Hua Chen, and York-Yih Sun. Resolving manipulator redundancy under inequality constraints. IEEE Transactions on Robotics and Automation, 10(1):65–71, 1994.

[8] Rolf Chini. The hexapod telescope–a never-ending story. In Reviews in Modern Astronomy, volume 13, pages 257–268, 2000.

[9] Thomas Geike and John McPhee. Inverse dynamic analysis of parallel manipulators with full mobility. Mechanism and Machine Theory, 38(6):549–562, 2003.

[10] Dongsheng Guo and Yunong Zhang. Neural dynamics and newton–raphson iteration for nonlinear optimization. Journal of Computational and Nonlinear Dynamics, 9(2):021016, 2014.

[11] H. Vicente H. Ayala and D.S. Coelho. Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator. Expert Systems with Applications, 39(10):8968–8974, 2012.

[12] Xiaolin Hu and Jun Wang. An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application. IEEE Transactions on Neural networks, 19(12):2022–2031, 2008.

[13] Xiaolin Hu, Jun Wang, and Bo Zhang. Motion planning with obstacle avoidance for kinematically redundant manipulators based on two recurrent neural networks. In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, pages 137–142. IEEE, 2009.

[14] Long Jin and Shuai Li. Distributed task allocation of multiple robots: A control perspective. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016.

[15] Long Jin, Shuai Li, Hung Manh La, and Xin Luo. Manipulability optimization of redundant manipulators using dynamic neural networks. IEEE Transactions on Industrial Electronics, 64(6):4710–4720, 2017.

[16] Long Jin and Yunong Zhang. G2-type srmpc scheme for synchronous manipulation of two redundant robot arms. IEEE transactions on cybernetics, 45(2):153–164, 2015.

[17] Long Jin, Yunong Zhang, and Shuai Li. Integration-enhanced zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. IEEE transactions on neural networks and learning systems, 27(12):2615–2627, 2016.

[18] Long Jin, Yunong Zhang, Shuai Li, and Yinyan Zhang. Modified znn for time-varying quadratic programming with inherent tolerance to noises and its application to kine-

matic redundancy resolution of robot manipulators. IEEE Transactions on Industrial Electronics, 63(11):6978–6988, 2016.

[19] Long Jin, Yunong Zhang, Shuai Li, and Yinyan Zhang. Noise-tolerant znn models for solving time-varying zero-finding problems: A control-theoretic approach. IEEE Transactions on Automatic Control, 62(2):992–997, 2017.

[20] Long Jin, Yunong Zhang, Tianjian Qiao, Manchun Tan, and Yinyan Zhang. Tracking control of modified lorenz nonlinear system using zg neural dynamics with additive input or mixed inputs. Neurocomputing, 196:82–94, 2016.

[21] Muhammad Umer Khan, Shuai Li, Qixin Wang, and Zili Shao. Cps oriented control design for networked surveillance robots with multiple physical constraints. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35(5):778–791, 2016.

[22] Muhammad Umer Khan, Shuai Li, Qixin Wang, and Zili Shao. Distributed multi-robot formation and tracking control in cluttered environments. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 11(2):12, 2016.

[23] Muhammad Umer Khan, Shuai Li, Qixin Wang, and Zili Shao. Formation control and tracking for co-operative robots with non-holonomic constraints. Journal of Intelligent & Robotic Systems, 82(1):163–174, 2016.

[24] R. Köker. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. Information Sciences, 222:528–543, 2013.

[25] Kene Li and Yunong Zhang. State adjustment of redundant robot manipulator based on quadratic programming. Robotica, 30(3):477–489, 2012.

[26] S. Li, S. Chen, B. Liu, Y. Li, and Y. Liang. Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. Neurocomputing, 91:1–10, 2012.

[27] Shuai Li and Yi Guo. Distributed source seeking by cooperative robots: All-to-all and limited communications. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 1107–1112. IEEE, 2012.

[28] Shuai Li and Yi Guo. Distributed consensus filter on directed switching graphs. International Journal of Robust and Nonlinear Control, 25(13):2019–2040, 2015.

[29] Shuai Li and Yi Guo. Dynamic consensus estimation of weighted average on directed graphs. International Journal of Systems Science, 46(10):1839–1853, 2015.

[30] Shuai Li, Yi Guo, and Brian Bingham. Multi-robot cooperative control for monitoring and tracking dynamic plumes. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pages 67–73. IEEE, 2014.

[31] Shuai Li, Jinbo He, Yangming Li, and Muhammad Usman Rafique. Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective. IEEE transactions on neural networks and learning systems, 28(2):415–426, 2017.

[32] Shuai Li, Ruofan Kong, and Yi Guo. Cooperative distributed source seeking by multiple robots: Algorithms and experiments. IEEE/ASME Transactions on mechatronics, 19(6):1810–1820, 2014.

[33] Shuai Li and Yangming Li. Nonlinearly activated neural network for solving time-varying complex sylvester equation. IEEE Transactions on Cybernetics, 44(8):1397–1407, 2014.

[34] Shuai Li, Yangming Li, and Zheng Wang. A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application. Neural Networks, 39:27–39, 2013.

[35] Shuai Li, Bo Liu, and Yangming Li. Selective positive–negative feedback produces the winner-take-all competition in recurrent neural networks. IEEE transactions on neural networks and learning systems, 24(2):301–309, 2013.

[36] Shuai Li, Yuesheng Lou, and Bo Liu. Bluetooth aided mobile phone localization: a nonlinear neural circuit approach. ACM Transactions on Embedded Computing Systems (TECS), 13(4):78, 2014.

[37] Shuai Li, Max QH Meng, Wanming Chen, Yangming Li, Zhuhong You, Yajin Zhou, Lei Sun, Huawei Liang, Kai Jiang, and Qinglei Guo. Sp-nn: A novel neural network approach for path planning. In Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on, pages 1355–1360. IEEE, 2007.

[38] Shuai Li, Zhu-Hong You, Hongliang Guo, Xin Luo, and Zhong-Qiu Zhao. Inverse-free extreme learning machine with optimal information updating. IEEE transactions on cybernetics, 46(5):1229–1241, 2016.

[39] Shuai Li, Zhu-Hong You, Hongliang Guo, Xin Luo, and Zhong-Qiu Zhao. Inverse-free extreme learning machine with optimal information updating. IEEE transactions on cybernetics, 46(5):1229–1241, 2016.

[40] Shuai Li, Yunong Zhang, and Long Jin. Kinematic control of redundant manipulators using neural networks. IEEE transactions on neural networks and learning systems, 28(10):2243–2254, 2017.

[41] Shuai Li, Yunong Zhang, and Long Jin. Kinematic control of redundant manipulators using neural networks. IEEE transactions on neural networks and learning systems, 28(10):2243–2254, 2017.

[42] Shuai Li, MengChu Zhou, Xin Luo, and Zhu-Hong You. Distributed winner-take-all in dynamic networks. IEEE Transactions on Automatic Control, 62(2):577–589, 2017.

[43] Q. Liu, Q. He, and Z. Shi. Extreme support vector machine classifier. In T. Washio, E. Suzuki, K. Ting, and A. Inokuchi, editors, Advances in Knowledge Discovery and Data Mining, volume 5012 of Lecture Notes in Computer Science, pages 222–233. Springer Berlin Heidelberg, 2008.

[44] Qiuge Liu, Qing He, and Zhongzhi Shi. Extreme support vector machine classifier. In Pacific-Asia conference on knowledge discovery and data mining, pages 222–233. Springer, 2008.

[45] Xin Luo, Zhong Ming, Zhuhong You, Shuai Li, Yunni Xia, and Hareton Leung. Improving network topology-based protein interactome mapping via collaborative filtering. Knowledge-Based Systems, 90:23–32, 2015.

[46] Xin Luo, Mingsheng Shang, and Shuai Li. Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications. In Data Mining (ICDM), 2016 IEEE 16th International Conference on, pages 311–319. IEEE, 2016.

[47] Xin Luo, Zhuhong You, Mengchu Zhou, Shuai Li, Hareton Leung, Yunni Xia, and Qingsheng Zhu. A highly efficient approach to protein interactome mapping based on collaborative filtering framework. Scientific reports, 5:7702, 2015.

[48] Xin Luo, MengChu Zhou, Hareton Leung, Yunni Xia, Qingsheng Zhu, Zhuhong You, and Shuai Li. An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering. IEEE Transactions on Automation Science and Engineering, 13(1):333–343, 2016.

[49] Xin Luo, MengChu Zhou, Shuai Li, YunNi Xia, Zhu-Hong You, QingSheng Zhu, and Hareton Leung. Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing qos data. IEEE transactions on cybernetics, 48(4):1216–1228, 2018.

[50] Xin Luo, Mengchu Zhou, Shuai Li, Yunni Xia, Zhuhong You, Qingsheng Zhu, and Hareton Leung. An efficient second-order approach to factorize sparse matrices in recommender systems. IEEE Transactions on Industrial Informatics, 11(4):946–956, 2015.

[51] Xin Luo, MengChu Zhou, Shuai Li, Zhuhong You, Yunni Xia, and Qingsheng Zhu. A nonnegative latent factor model for large-scale sparse matrices in recommender

systems via alternating direction method. IEEE transactions on neural networks and learning systems, 27(3):579–592, 2016.

[52] Xin Luo, Mengchu Zhou, Mingsheng Shang, Shuai Li, and Yunni Xia. A novel approach to extracting non-negative latent factors from non-negative big sparse matrices. IEEE access, 4:2649–2655, 2016.

[53] Xin Luo, MengChu Zhou, Zidong Wang, Yunni Xia, and Qingsheng Zhu. An effective scheme for qos estimation via alternating direction method-based matrix factorization. IEEE Transactions on Services Computing, 2016.

[54] Yanhong Luo, Qiuye Sun, Huaguang Zhang, and Lili Cui. Adaptive critic design-based robust neural network control for nonlinear distributed parameter systems with unknown dynamics. Neurocomputing, 148:200–208, 2015.

[55] Jose Luis M., V. Santibáñez, R. Soto, and M.A. Llama. Fuzzy self-tuning pid semiglobal regulator for robot manipulators. Industrial Electronics, IEEE Transactions on, 59(6):2709–2717, 2012.

[56] Joseph Madamesila, Philip McGeachy, J Eduardo Villarreal Barajas, and Rao Khan. Characterizing 3d printing in the fabrication of variable density phantoms for quality assurance of radiotherapy. Physica Medica: European Journal of Medical Physics, 32(1):242–247, 2016.

[57] Mingzhi Mao, Jian Li, Long Jin, Shuai Li, and Yunong Zhang. Enhanced discrete-time zhang neural network for time-variant matrix inversion in the presence of bias noises. Neurocomputing, 207:220–230, 2016.

[58] O. Masory and W. Jian. Workspace evaluation of stewart platforms. Advanced Robotics, 9(4):443–461, 1994.

[59] Oren Masory and Jian Wang. Workspace evaluation of stewart platforms. Advanced robotics, 9(4):443–461, 1994.

[60] Jean-Pierre Merlet. Parallel robots, volume 128. Springer Science & Business Media, 2006.

[61] Aquil Mirza Mohammed and Shuai Li. Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms. IEEE transactions on cybernetics, 46(7):1538–1550, 2016.

[62] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. Trans. ASME Journal of Dynamic System, Measures and Control, 108, 1986.

[63] Rendong Nan, Di Li, Chengjin Jin, Qiming Wang, Lichun Zhu, Wenbai Zhu, Haiyan Zhang, Youling Yue, and Lei Qian. The five-hundred-meter aperture spherical radio telescope (fast) project. International Journal of Modern Physics D, 20(06):989–1024, 2011.

[64] Zhen Ni, Haibo He, Xiangnan Zhong, and Danil V Prokhorov. Model-free dual heuristic dynamic programming. IEEE transactions on neural networks and learning systems, 26(8):1834–1839, 2015.

[65] N. Nikdel, P. Nikdel, M.A. Badamchizadeh, and I. Hassanzadeh. Using neural network model predictive control for controlling shape memory alloy-based manipulator. Industrial Electronics, IEEE Transactions on, 61(3):1394–1401, 2014.

[66] Nazila Nikdel, Parisa Nikdel, Mohammad Ali Badamchizadeh, and Iraj Hassanzadeh. Using neural network model predictive control for controlling shape memory alloy-based manipulator. IEEE Transactions on Industrial Electronics, 61(3):1394–1401, 2014.

[67] Viera Poppeova, Vladimir Bulej, and Juraj Uricek. Parallel kinematic structures and their innovative applications. Annals of DAAAM & Proceedings, pages 353–355, 2011.

[68] Andreas Pott, Hendrick Mütherich, Werner Kraus, Valentine Schmidt, Philipp Miermeister, and Alexander Verl. Ipanema: a family of cable-driven parallel robots for industrial applications. In Cable-Driven Parallel Robots, pages 119–134. Springer, 2013.

[69] Liu S., Li W., Du Y., and Fang L. Forward kinematics of the stewart platform using hybrid immune genetic algorithm. In Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on, pages 2330–2335, June 2006.

[70] Nasri S., F. Piltan, S. Haghighi, I. Nazari, and S. Siamak. Artificial control of puma robot manipulator: A-review of fuzzy inference engine and application to classical controller. International journal of Robotics and Automation (IJRA), 2(5):406–429, 2011.

[71] Carsten Walter Schwarz and Klaus-Uwe Hahn. Full-flight simulator study for wake vortex hazard area investigation. Aerospace Science and Technology, 10(2):136–143, 2006.

[72] Liu Sheng, Li Wan-long, Du Yan-chun, and Fang Liang. Forward kinematics of the stewart platform using hybrid immune genetic algorithm. In Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on, pages 2330–2335. IEEE, 2006.

[73] B Siciliano, L Sciavicco, L Villani, and G Oriolo. Robotics–modelling, planning and control. advanced textbooks in control and signal processing series, 2009.

[74] E.D. Sontag. Mathematical control theory: deterministic finite dimensional systems, volume 6. Springer, 1998.

[75] Eduardo D Sontag. Mathematical control theory: deterministic finite dimensional systems, volume 6. Springer Science & Business Media, 2013.

[76] G. Thomas and M. John. Inverse dynamic analysis of parallel manipulators with full mobility. Mechanism and Machine Theory, 38(6):549 – 562, 2003.

[77] R Trucco, PC Galeone, and F Pepe. Hexapod pointing system. In Spacecraft Guidance, Navigation and Control Systems, volume 381, page 201, 1997.

[78] Fei-Yue Wang, Huaguang Zhang, and Derong Liu. Adaptive dynamic programming: An introduction. IEEE computational intelligence magazine, 4(2), 2009.

[79] Huanqing Wang, Xiaoping Liu, and Kefu Liu. Adaptive fuzzy tracking control for a class of pure-feedback stochastic nonlinear systems with non-lower triangular structure. Fuzzy Sets and Systems, 302:101–120, 2016.

[80] Huanqing Wang, Xiaoping Liu, and Kefu Liu. Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems. IEEE transactions on neural networks and learning systems, 27(3):510–523, 2016.

[81] Huanqing Wang, Xiaoping Liu, and Kefu Liu. Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems. IEEE transactions on neural networks and learning systems, 27(3):510–523, 2016.

[82] Z.P. Wang, T. Zhou, Y. Mao, and Q.J. Chen. Adaptive recurrent neural network control of uncertain constrained nonholonomic mobile manipulators. International Journal of Systems Science, 45(2):133–144, 2014.

[83] ZP Wang, T Zhou, Y Mao, and QJ Chen. Adaptive recurrent neural network control of uncertain constrained nonholonomic mobile manipulators. International Journal of Systems Science, 45(2):133–144, 2014.

[84] Nancy M Watson, Thelma J Wells, and Christopher Cox. Rocking chair therapy for dementia patients: Its effect on psychosocial well-being and balance. American Journal of Alzheimer's disease, 13(6):296–308, 1998.

[85] Manfred Weck and Dirk Staimer. Parallel kinematic machine tools–current state and future potentials. CIRP Annals-Manufacturing Technology, 51(2):671–683, 2002.

[86] Youshen S Xia, Gang Feng, and Jun Wang. A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control. IEEE

Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(1):54–64, 2005.

[87] Y.S. Xia. Further results on global convergence and stability of globally projected dynamical systems. Journal of Optimization Theory and Applications, 122(3):627–649, 2004.

[88] YS Xia. Further results on global convergence and stability of globally projected dynamical systems. Journal of Optimization Theory and Applications, 122(3):627–649, 2004.

[89] Lin Xiao. A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation. Neurocomputing, 167:254–259, 2015.

[90] Lin Xiao. A new design formula exploited for accelerating zhang neural network and its application to time-varying matrix inversion. Theoretical Computer Science, 647:50–58, 2016.

[91] Lin Xiao. A nonlinearly activated neural dynamics and its finite-time solution to time-varying nonlinear equation. Neurocomputing, 173:1983–1988, 2016.

[92] Lin Xiao. A nonlinearly-activated neurodynamic model and its finite-time solution to equality-constrained quadratic optimization with nonstationary coefficients. Applied Soft Computing, 40:252–259, 2016.

[93] Lin Xiao and Bolin Liao. A convergence-accelerated zhang neural network and its solution application to lyapunov equation. Neurocomputing, 193:213–218, 2016.

[94] Lin Xiao and Rongbo Lu. Finite-time solution to nonlinear equation using recurrent neural dynamics with a specially-constructed activation function. Neurocomputing, 151:246–251, 2015.

[95] Lin Xiao and Yunong Zhang. Dynamic design, numerical solution and effective verification of acceleration-level obstacle-avoidance scheme for robot manipulators. International Journal of Systems Science, 47(4):932–945, 2016.

[96] A. Yorukoglu and E. Altug. Estimation of unbalanced loads in washing machines using fuzzy neural networks. Mechatronics, IEEE/ASME Transactions on, 18(3):1182–1190, 2013.

[97] Ahmet Yörükoğlu and Erdinç Altuğ. Estimation of unbalanced loads in washing machines using fuzzy neural networks. IEEE/ASME Transactions on Mechatronics, 18(3):1182–1190, 2013.

[98] Zhu-Hong You, MengChu Zhou, Xin Luo, and Shuai Li. Highly efficient framework for predicting interactions between proteins. IEEE transactions on cybernetics, 47(3):731–743, 2017.

[99] X. Youshen and W. Jun. A dual neural network for kinematic control of redundant robot manipulators. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 31(1):147–154, Feb 2001.

[100] Songchuan Zhang, Youshen Xia, and Weixing Zheng. A complex-valued neural dynamical optimization approach and its stability analysis. Neural Networks, 61:59–67, 2015.

[101] Y. Zhang, S.S. Ge, and T. Lee. A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 34(5):2126–2132, 2004.

[102] Y. Zhang, W. Jun, and Y. Xia. A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. IEEE Transactions on Neural Networks, 14(3):658–667, 2003.

[103] Y. Zhang and J. Wang. A dual neural network for convex quadratic programming subject to linear equality and inequality constraints. Phys. Lett. A, pages 271–278, 2002.

[104] Yinyan Zhang and Shuai Li. Predictive suboptimal consensus of multiagent systems with nonlinear dynamics. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7):1701–1711, 2017.

[105] Yinyan Zhang and Shuai Li. Predictive suboptimal consensus of multiagent systems with nonlinear dynamics. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7):1701–1711, 2017.

[106] Yunong Zhang, Shuzhi Sam Ge, and Tong Heng Lee. A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(5):2126–2132, 2004.

[107] Yunong Zhang, Liangyu He, Shuai Li, Dechao Chen, and Yaqiong Ding. Zeroing dynamics based motion control scheme for parallel manipulators. Electronics Letters, 53(2):74–75, 2016.

[108] Yunong Zhang, Jun Wang, and Youshen Xia. A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. IEEE transactions on neural networks, 14(3):658–667, 2003.

[109] Zhijun Zhang, Aryel Beck, and Nadia Magnenat-Thalmann. Human-like behavior generation based on head-arms model for robot tracking external targets and body parts. IEEE transactions on cybernetics, 45(8):1390–1400, 2015.

[110] Zhijun Zhang, Zhijun Li, Yunong Zhang, Yamei Luo, and Yuanqing Li. Neural-dynamic-method-based dual-arm cmg scheme with time-varying constraints applied to humanoid robots. IEEE transactions on neural networks and learning systems, 26(12):3251–3262, 2015.

[111] Cong Zheng and Jinde Cao. Robust synchronization of coupled neural networks with mixed delays and uncertain parameters by intermittent pinning control. Neurocomputing, 141:153–159, 2014.