Simulation of a Humanoid Robot

A thesis presented to

the faculty of

the Russ College of Engineering and Technology of Ohio University

In partial fulfillment

of the requirements for the degree

Master of Science

Chandana Venkatayogi

November 2007
This thesis titled

Simulation of a Humanoid Robot

by

CHANDANA VENKATAYOGI

has been approved for

the Department of Mechanical Engineering

and the Russ College of Engineering and Technology by

Robert L. Williams II

Professor of Mechanical Engineering

Dennis Irwin

Dean, Russ College of Engineering and Technology
ABSTRACT

VENKATAYOGI, CHANDANA, M.S., November 2007, Mechanical Engineering

Simulation of a Humanoid Robot (150 pp.)

Director of Thesis: Robert L. Williams II

A 35 degrees-of-freedom humanoid robot is constructed in a modular fashion with repeated seven degrees of freedom serial robot chain with spherical-revolute-spherical joints for the trunk, arms and legs. The major benefit of the repeated serial structure is the ease of computations resulting from the similar kinematic parameters for the five serial robot chains. Simulations are done to test the behavior of the humanoid. The humanoid robot arm is simulated using the resolved rate control motion, with a constant velocity input for the hand. The inverse dynamics solution for the resolved rate control motion of the humanoid arm is developed using an iterative Newton-Euler formulation and the required actuator torques are calculated. The humanoid robot is also simulated in the context of resolved rate control, with motion constrained to the left arm. The proposed humanoid robot is kinematically simulated for walking and stair climbing with trajectory planning, from literature, in the form of joint angles for each step of locomotion.

Approved: _____________________________________________________________

Robert L. Williams
Professor of Mechanical Engineering
ACKNOWLEDGMENTS

Throughout my graduate career, I will be indebted to my advisor, Dr. Robert L. Williams for his immense patience and dedication, which has made this research possible. I would like to express my sincere and deep gratitude for his invaluable suggestions and comments, throughout all stages of my academic work. I also wish to express my appreciation to Dr. Hajrudin Pasic, Dr. Ben Stuart and Dr. Premjit Singh for being on my thesis committee.

Thanks to my wonderful family: Mom, Dad, Shaanu and Lachi for everything. Finally, I would like to thank Aaru, Sri and Tinku for all the strength and support.
# Table of Contents

Abstract ............................................................................................................................... 3  

Acknowledgments ............................................................................................................... 4  

List of Tables ...................................................................................................................... 7  

List of Figures ..................................................................................................................... 8  

CHAPTER 1: Introduction ........................................................................................... 11  

1.1 Statement of Purpose ................................................................................ 11  
1.2 Background ............................................................................................... 11  
1.3 Literature Review ...................................................................................... 12  
1.4 Thesis Objectives ...................................................................................... 21  
1.5 Thesis Organization .................................................................................. 21  

CHAPTER 2: Humanoid Robot Development ............................................................ 23  

2.1 Human Structure ....................................................................................... 23  
2.2 Proposed Humanoid Robot ....................................................................... 28  

CHAPTER 3: Kinematics ............................................................................................ 33  

3.1 D-H Parameters: ........................................................................................ 33  
3.2 Forward Kinematics .................................................................................. 39  

CHAPTER 4: Resolved Rate Control .......................................................................... 43  

4.1 Jacobian Matrix ......................................................................................... 44  
4.2 Programming and Simulation Results ...................................................... 45  
4.3 Application of Resolved Rate Control to Humanoid .............................. 52
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3-1 Humanoid Arm D-H Parameters</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Table 3-2 D-H Parameters for the Humanoid Robot</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>Table 5-1 Lengths, Radii and Initial Joints of the Links of the Humanoid</td>
<td></td>
<td>61</td>
</tr>
<tr>
<td>Table 6-1 Lengths, Radii and Initial Joint Angles of the Humanoid Arm</td>
<td></td>
<td>86</td>
</tr>
<tr>
<td>Figure 1-1</td>
<td>Humanoid Robots from Honda Company</td>
<td>13</td>
</tr>
<tr>
<td>Figure 1-2</td>
<td>Humanoid Robots from Waseda University</td>
<td>14</td>
</tr>
<tr>
<td>Figure 1-3</td>
<td>QRIO</td>
<td>15</td>
</tr>
<tr>
<td>Figure 1-4</td>
<td>Partner Robots from Toyota Motor Company</td>
<td>16</td>
</tr>
<tr>
<td>Figure 1-5</td>
<td>Humanoid Robots from MIT</td>
<td>17</td>
</tr>
<tr>
<td>Figure 1-6</td>
<td>GuRoo from University of Queensland</td>
<td>18</td>
</tr>
<tr>
<td>Figure 1-7</td>
<td>HRP-2 from Kawada Industries Inc.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 2-1</td>
<td>Plane, Saddle and Ellipsoidal joints</td>
<td>24</td>
</tr>
<tr>
<td>Figure 2-2</td>
<td>Hinged Joint</td>
<td>25</td>
</tr>
<tr>
<td>Figure 2-3</td>
<td>Pivot Joint</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2-4</td>
<td>Condyloid Joint</td>
<td>27</td>
</tr>
<tr>
<td>Figure 2-5</td>
<td>Ball and Socket Joint</td>
<td>28</td>
</tr>
<tr>
<td>Figure 2-6</td>
<td>Proposed Humanoid Robot</td>
<td>30</td>
</tr>
<tr>
<td>Figure 2-7</td>
<td>Exploded View of the Humanoid Robot</td>
<td>31</td>
</tr>
<tr>
<td>Figure 2-8</td>
<td>Front and Side Views of the Proposed Humanoid Robot</td>
<td>32</td>
</tr>
<tr>
<td>Figure 3-1</td>
<td>Coordinate Frames for Basic Serial Chain</td>
<td>34</td>
</tr>
<tr>
<td>Figure 3-2</td>
<td>Nomenclature used for D-H Parameters of the Humanoid Robot</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4-1</td>
<td>Initial Position of the Humanoid Arm</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4-2</td>
<td>Final Position of the Humanoid Arm</td>
<td>47</td>
</tr>
</tbody>
</table>
Figure 4-3 \( \theta \) (radians) vs. time (sec) Plots ................................................................. 49
Figure 4-4 Joint rates (rad/sec) vs. time (sec) plots ......................................................... 50
Figure 4-5 Cartesian Position (m) and Euler Angles (rad) vs. time (sec) for End Effector
................................................................................................................................... 51
Figure 4-6 Det \((J * J_t)\) vs. time (sec) ........................................................................... 52
Figure 4-7 Initial Position of the Humanoid for Resolved Rate Control Example.......... 54
Figure 4-8 Final Position of the Humanoid for Resolved Rate Control Example ........ 55
Figure 5-1 Lengths and Radii of the Links of the Humanoid ......................................... 60
Figure 5-2 Initial Vertical Position of the Humanoid Robot for Walking ..................... 64
Figure 5-3 Notation for the Joint Angles of the Leg Joints ............................................ 66
Figure 5-4 Trajectories of Joint Angles of \(q_1, q_2, q_3\) and \(q_4\) vs. time for Simulating
Humanoid Walking .................................................................................................... 67
Figure 5-5 Trajectories of Joint Angles of Leg and Hand Joints from Simulation of
Humanoid Walking .................................................................................................... 70
Figure 5-6 Humanoid at the End of the Starting Step after 0.4 seconds ....................... 71
Figure 5-7 Humanoid at the End of the First Steady Step at the End of 0.9 seconds ...... 71
Figure 5-8 Humanoid in Walking with Plots at Increments of 0.1 secs for 0.9 secs ....... 72
Figure 5-9 Initial Vertical Position of the Humanoid for Stair Climbing ....................... 75
Figure 5-10 Trajectories of Joint Angles of \(q_1, q_2, q_3\) and \(q_4\) with time used for
Simulation of Stair Climbing ..................................................................................... 76
Figure 5-11 Trajectories of Joint Angles of Leg and Hand Joints from Simulation of
Humanoid Stair Climbing ......................................................................................... 78
Figure 5-12 Humanoid at the end of the starting step after 0.5 seconds of stair climbing...

Figure 5-13 Humanoid at the end of the first steady step at the end of 1 second of stair climbing

Figure 5-14 Humanoid in Stair Climbing with Plots at Increments of 0.1 seconds for 1.5 seconds

Figure 6-1 Plot of forces vs. time of the 7 joints of the humanoid arm

Figure 6-2 Plot of Forces vs. time of Various Joints of the Humanoid Arm

Figure 6-3 Plot of Torques vs. time for Joints 1 through 7 (Joints of the Trunk)

Figure 6-4 Plot of Torques vs. time for joints 8 through 14 (Joints of Left Hand)

Figure 6-5 Plot of Torques vs. time for joints 15 through 21 (Joints of the Right Hand)

Figure 6-6 Plot of Torques vs. time for joints 22 through 28 (Joints of the Left Leg)

Figure 6-7 Plot of Torques vs. time for joints 29 through 35 (Joints of the Right Leg)
CHAPTER 1: INTRODUCTION

1.1 Statement of Purpose

The purpose of this thesis is to simulate a Humanoid robot. A model for a humanoid robot has to be developed. The kinematic model for the humanoid will be built. The dynamics model and an inverse dynamic solution has to be developed. The goal is to simulate the proposed humanoid robot for representative motions.

1.2 Background

According to The Robot Institute of America (1979), a robot is defined as "A reprogrammable, multifunctional manipulator designed to move materials, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks." The root word for Robot is ‘Robota’ which is Czech for ‘forced labor’. The word has its origins from Karel Capek’s 1921 play ‘Rossum’s Universal Robots’. This play provided a vision that a robot looks like a human. Starting in the 1930s, Isaac Asimov proposed humanoid robots in his science fiction robot short stories and novels. These humanoid robots are projected indistinguishable from human beings. In a short story called ‘Runaround’, Asimov presented the Three Laws of Robotics for the safety of humans and robots.

A humanoid robot (also called android) has a structure based on the human body. Generally, the humanoids have a torso with a head, a pair of arms and a pair of legs. In
some cases, the legs are replaced by wheels for easier locomotion. Additionally the humanoids may also have eyes, mouth, and ears, adding vision, speech, and hearing.

Research in robotics dates back to 1960s. In the decade from 1960 to 1970 the research was focused on industrial robots and manipulators for automatic manufacturing processes. Human arms were imitated to design mechanical manipulators for various automated tasks like gripping, welding, spray painting, grinding, etc. In the 1980s, mechatronic mobile systems with wheels or legs were developed. In the 1990s, concepts like artificial intelligence, vision, speech and voice recognition were incorporated making the robots safer to use in hazardous environments.

1.3 Literature Review

This section presents the literature review for this project. Humanoid robots developed by various laboratories and research in the field of biped locomotion are mentioned.

After ten experimental robots, Honda Company came up with humanoid robots like P1, P2, P3 and ASIMO. P1 is the first Honda prototype of a humanoid robot. This robot can turn electrical and computer switches, grab door knobs, pick up and carry things. The height of P1 is 1.915 m (6’ 3 4/10”) and 175 kg (385 lbs). P2 is a self-regulating battery-operated android which can walk independently, walk up and down the stairs, and push carts. Wireless techniques were used with a computer torso, motors, battery, wireless radio and other built-in devices. P2 is 1.82 m (5’ 11 7/10”) tall and 210
kg (462 lbs). P3 has changes in component materials and a decentralized control system which resulted in decrease in height, 1.6 m (5.25 ft) and weight, 130 kg (286 lbs). It has 16 joints in total. It has 30 degrees of freedom; 12 for legs, 14 for arms and 4 for hands. The maximum walking speed of P3 is 2 km/hr. ASIMO (Advanced Step in Innovative Mobility) has 26 degrees of freedom. This walking humanoid is 1.2m tall (3’ 11 2/10") and a mass of 52 kg (114.4 lbs). The walking speed of ASIMO is 0-1.6 km/hr. This robot can recognize moving objects. Other features include posture, gesture, sound, and face recognition.

The Humanoid Robotics Institute of Waseda University developed a full scale human-like robot in the 1970s called WABOT-1. This robot is capable of stable walking by shifting its center of gravity from one leg to another. WABOT-2 is a musician robot that played the piano. WABIAN is a biped with a complete human configuration capable of walking and carrying objects. HALADY is another robot from Waseda University.
which can interact with humans. It has voice recognition and voice synthesis capabilities along with gesture behavior recognition and conversation capability. HALADY-2 works together with a human partner. Apart from the technology used for HALADY, it also possesses physical interaction functions enabling direct contact with humans. Waseda University has another humanoid robot called iSHA with 26 degrees of freedom mostly driven by electric motors. This robot is capable of two hours autonomous operation. WENDY (Waseda Engineering Designed Symbiotic) is a human symbiotic robot. It is capable of physical and emotional interaction with humans. It is a 52 degrees of freedom mechanism with a height of 1.5 m (4’ 11 1/10”) and mass 170kg (374 lbs ). It can recognize the environment using CCD cameras. WAMOeba (Waseda Artificial Mind on Emotion Base) is designed to emerge intelligence and emotion on par with humans. Also, the communication ability of the robot is enhanced.

Figure 1-2 Humanoid Robots from Waseda University

Source: [http://www.humanoid.rise.waseda.ac.jp](http://www.humanoid.rise.waseda.ac.jp)
QRIO from Sony Corporation is a biped humanoid robot that can walk on uneven surfaces, dance, recognize faces and voices of people, and also carry on conversations. The pinch detection feature enables QRIO to sense if anything is caught in its joints. A special feature of QRIO is that it reacts to protect itself against an impact. In the event of falling, it gets back up by itself after checking in all directions. The features are attained by Intelligent Servo Actuator (ISA), a drive system with motors, gears, a computer and a set of sensors.

Figure 1-3 QRIO

Source: [http://www.sony.net/SonyInfo/QRIO/top_nf.html](http://www.sony.net/SonyInfo/QRIO/top_nf.html)
Toyota Motor Company has 4 models of humanoid robots called the partner robots and are designed for personal assistance and entertainment. The first robot is designed for walking, assistance, and elderly care. This android is 1.2m (4’) tall and weighs 35 kg (77 lbs). The second model is a mountable robot intended for elderly care and mobility. It stands 1.8 m (6’) tall and weighs 75 Kg (165 lbs). This robot looks like a chair and can carry passengers. The third robot is a rolling version. The areas of application include manufacturing and mobility. It is 1 m (40”) tall and also weighs 35 kg (77 lbs). The fourth model being produced by Toyota is a wire-operated version. It is lighter than the others and can move more quickly. The actuators in the torso act as power sources for arm and leg movements.

![Partner Robots from Toyota Motor Company](http://www.toyota.co.jp/en/special/robot/)

<table>
<thead>
<tr>
<th>WALKING</th>
<th>MOUNTABLE</th>
<th>ROLLING</th>
<th>WIRELESS OPERATED</th>
</tr>
</thead>
</table>

Figure 1-4 Partner Robots from Toyota Motor Company


The Massachusetts Institute of Technology has a humanoid COG that has a head, two arms and torso. The head has four eyes, two of which are intended for close-up
vision and the other two for distant viewing. The M2 project from MIT’s leg lab constructed a set of legs that could be attached to the torso of COG. Coco from MIT is a 15 degree-of-freedom robot. It weighs around 10 kgs. The MIT Artificial Intelligence Laboratory has a human-like robotic hand. It has vision and sound input and output, arms and dexterous manipulation. The hand has four fingers, with two joints on with finger and a thumb. These are controlled by tendon like cables. The system is of a human-scale cable-driven tool with actuators and sensors for tactile positioning and torquing and a computer.

![Figure 1-5 Humanoid Robots from MIT](http://www.ai.mit.edu/projects/humanoid-robotics-group/leglab/robots/m2/m2.html)

The GuRoo project of the University of Queensland Robotics Laboratory is a humanoid which capable of balancing, turning, crouching and standing from a prostrate position. It is a 1.2 m robot with a total of 23 joints, fifteen in the legs and abdomen and eight joints in the head and neck assembly. The other 8 joints drive the head and neck
assembly and the arms. The joints in the legs and abdomen produce significant mechanical power with large torques and relatively low speeds.

Figure 1-6 GuRoo from University of Queensland


BiOMAN-1, Jirou is a biologically-inspired, bipedal, dynamic, humanoid robot that was developed at the Artificial Life and Robotics Laboratory of Oita University. This robot can walk dynamically and go up and down stairs. The central pattern generator produces different types of walking patterns. The robot has a pair of small CMOS color CCD cameras, a speaker, and a microphone in the head, a GPS receiver, a portable telephone, and sensors in the body part, so that the integration of locomotion and behaviors to achieve specific demonstrations will be realized. Speed is at least 12cm/s. There is an artificial brain for BiOMAN-1.
HRP-2 from Kawada Industries Inc. is a humanoid that is 154 cm (60") tall, weighs 58 kg (127 lbs) and has 30 degrees-of-freedom. It can cope with uneven surfaces, walk at two-thirds of human speed on a narrow path. HRP-2 is designed to be adult human feminine size. The hip joint of HRP-2 has a cantilever-type structure. It has a waist with 2 degrees of freedom and three CCD cameras inside of a head module.

![Figure 1-7 HRP-2 from Kawada Industries Inc.](http://www.kawada.co.jp/global/ams/hrp_2.html)

Many researchers have developed several locomotion algorithms for bipeds. Lum et al (1999) presented a trajectory planning of a five link biped for walking and also for stair climbing. Their model describes five links, which consists of the two legs and the torso driven by independent DC motors. Hasegawa et al (2000) have proposed a natural stable walking motion for a biped robot in different environments. The algorithm
for walking motion is based on Zero Moment Point (ZMP) at which the moments around 
\( x \) and \( y \) axis resulted from the floor reaction and torque are zero.

A passive dynamic walking three-dimensional biped model is built by Collins et al (2001). The model has two legs and knees. The feet of the biped are curved and the arms are constrained mechanically for a stable gait.

Chew and Pratt (2002) use a Virtual Model Control approach for the formulation of motion control method and SD-FAST, a program for dynamic simulation program of two biped robots to demonstrate the application of their algorithm. The first biped Spring Flamingo has two legs with three actuated rotary joints each and feet. The second biped M2 has six degrees-of-freedom for each leg, with three at the hip, two at ankles and the remaining one at the knee joint. The locomotion is in the sagittal plane. Chew and Pratt (2002) also developed two frontal plane algorithms for dynamic biped walking on level ground. This is applied to a simulated model of M2.

Azevedo (2004) proposed a control law for biped machines. The walking gait is based on biomechanics, neuroscience and physiology. The authors opine that the biped locomotion problem has been solved during human evolution. The data for human walking is recorded from the motion of a human subject.
1.4 Thesis Objectives

The thesis research work is involved with the kinematic and dynamic design and simulation for a humanoid robot constructed in a modular fashion with repeated seven degrees-of-freedom serial robot chain. The specific thesis objectives are:

- Design of humanoid robot including DH parameters, plus approximate dynamic parameters.
- Kinematic modeling and solutions including forward pose kinematics and resolved-rate control.
- Development of a dynamics model and inverse dynamics solution.
- MATLAB simulation of humanoid robot control for representative tasks and motions.
- Development of 3D graphics to support MATLAB simulation.

1.5 Thesis Organization

Chapter 2 of this thesis describes the development of the humanoid robot based on human structure. A model for humanoid robot is presented. Chapter 3 gives the formulation of the Denavit and Hartenberg parameters (DH parameters) and the development of forward pose kinematics solution of the proposed humanoid robot. The model is simulated using the resolved rate control algorithm. The method used and the programming results of the resolved rate control are presented in the Chapter 4. Chapter 5 discusses the trajectory planning for walking and stair climbing. The humanoid is
simulated using these trajectories. The simulation and results are also presented in that chapter. Chapter 6 has the inverse dynamics of the humanoid robot in the context of resolved rate control of the humanoid arm, where the torques are computed for the required dynamic motions. Finally, conclusions derived from this research and recommendations for future study are given in Chapter 6.
CHAPTER 2: HUMANOID ROBOT DEVELOPMENT

2.1 Human Structure

The human skeleton is a framework of bones, cartilage, ligaments and joints of the body. The skeletal system provides many levers of movement. Bones are the majority of the structures in the skeleton. A human adult is comprised of 206 bones connected at joints. The skeletal system provides the levers (bones) and axes of rotation (formed due to the joints) about which movements are generated. Hamill & Knutzen (1995) is used for the information about human structure and the related figures in this chapter.

These joints are of three different types: Fixed, partially-movable, and freely-movable joints. Fixed joints do not allow any movement. Examples of these joints are the joints in the adult skull. Slightly movable joints are the joints which allow only a small amount like the joints between the vertebrae.

Most of the joints in the human body are freely movable joints. They are also known as the synovial or the diarthrodial joints. The diarthrodial joints are of seven types based on the type of movement provided and the number of degrees of freedom allowed.

1. Plane or gliding joints: These are non-axial joints. Due to these joints, the bones can slide over each other. These joints provide flexion, extension, radial deviation, and ulnar deviation. These joints are found at the foot among tarsals and at hand among the carpals. An example of these joints is shown in figure 2-1.
2. Saddle joints: These joints are functionally similar to the ellipsoidal joints except that a small amount of rotation is also provided. The carpometacarpal joint of thumb is an example of a saddle joint. Figure 2.1 shows a saddle joint.

3. Ellipsoidal joints: These are biaxial joints with two degrees of freedom, providing motion in two planes. Flexion and extension are provided in one plane. The motions in the second plane are abduction and adduction. The radio carpal joint at the wrist is an example of an ellipsoidal joint. Figure 2-1 shows an Ellipsoidal joint.
4. Hinged joints: These are uniaxial joints with one degree of freedom. Example of such a joint is the ulnohumeral articulation at the elbow. Figure 2-2 gives an example of hinged joint.

![Figure 2-2 Hinged Joint](source)

3. Pivot joints: These are also uniaxial with one degree of freedom. The motion provided is rotation, pronation, and supination. These joints can be found at the superior and inferior radioulnar joint. Figure 2-2 shows a pivot joint.
4. Condyloid joints: These joints provide primary movement in one plane and a small amount of rotational movement in another plane, giving two degrees-of-freedom. Flexion and extension are provided in the first plane and a limited rotation is provided in another plane. An example for this joint is the knee. Figure 2-4 shows a Condyloid joint.
7. Ball and socket joints: These are the most mobile joints in the human body with three degrees-of-freedom. The movement is in three planes. Flexion and extension, abduction & adduction, and rotation are the movements provided. Examples of ball and socket joints are the hip and shoulder joints. Figure 2-5 shows a ball and socket joint.
2.2 Proposed Humanoid Robot

For the design of a humanoid robot, only the joints which provide basic locomotion are considered. The shoulder joints are ball and socket joints. The elbow is a hinged joint. The human wrist with all the joints has three degrees of freedom, hence it is approximated with a ball and socket joint. All the slightly movable joints of the vertebrae are approximated to a single hinged joint. The neck and the pelvis are approximated as ball and socket joints. With these approximations, the structure of this set of joints
becomes similar to the shoulder-elbow-wrist set. The hip-knee-ankle are also considered as ball and socket, hinged, and ball and socket joints.

Based on these assumptions, a 35 degrees-of-freedom humanoid robot design is proposed. The three-dimensional humanoid is modeled in Solid Edge, as shown in Figures 2-6 through 2-8. From the figure 2-6, it can be seen that the proposed humanoid has the torso with torso with one joint for the vertebrae for simplification. The Humanoid has head, neck, two legs and arms.
Figure 2-6 Proposed Humanoid Robot
Figure 2-7 shows the exploded view of the proposed humanoid. It can be seen that the humanoid has similar kinematic structure for the spine, arms and legs. The joint structure comprises a combination of spherical-revolute-spherical joints in series. In this humanoid, similar serial robot chains with seven degrees of freedom each are repeated five times.
Figure 2-8 Front and Side Views of the Proposed Humanoid Robot
In this chapter we discuss the kinematics of the humanoid robot. The Denavit-Hartenberg (D-H) notation (Craig, 2005) is used to describe the location of each link of the manipulator with respect to its neighboring link. Then, for a given set of joint angles, the resultant pose (position and orientation) is calculated. In formulating the manipulator kinematics, we focus on the motion of the links without the influence of masses and forces.

3.1 D-H Parameters:

Any robot can be described kinematically with values of four quantities for each link/joint. Two of these quantities describe the link geometry and the other two describe the link’s connection to its neighboring link. For a revolute joint, $\theta_i$ is called the joint variable, and the other three quantities would be fixed link parameters. The definition of serial robots by these quantities is known as Denavit-Hartenberg (D-H) notation. The D-H parameters are used to define the pose of each link coordinate frame with respect to its previous neighboring link frame. D-H convention from Craig (2005) is used.

The D-H parameters of the human arm are presented first. The human arm is comprised of a spherical shoulder, a revolute elbow, and a spherical wrist. Each spherical joint is represented as a set of three intersecting revolute joints, perpendicular to each other and separated by a zero distance. Thus, the spherical shoulder consists of three revolute joints (1 through 3), the elbow consists of a single revolute joint (4), and the
spherical wrist consists of three revolute joints (5 through 7). Table 3-1 shows the D-H parameters from Figure 3.1.

![Figure 3-1 Coordinate Frames for Basic Serial Chain](image)

**Table 3-1 Humanoid Arm D-H Parameters**

<table>
<thead>
<tr>
<th>$I$</th>
<th>$a_{i-1}$</th>
<th>$d_i$</th>
<th>$\theta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0^\circ$</td>
<td>0</td>
<td>$\theta_1$</td>
</tr>
<tr>
<td>2</td>
<td>$+90^\circ$</td>
<td>0</td>
<td>$\theta_2+90$</td>
</tr>
<tr>
<td>3</td>
<td>$+90^\circ$</td>
<td>0</td>
<td>$\theta_3$</td>
</tr>
<tr>
<td>4</td>
<td>$0^\circ$</td>
<td>$L_1$</td>
<td>$\theta_4$</td>
</tr>
<tr>
<td>5</td>
<td>$0^\circ$</td>
<td>$L_2$</td>
<td>$\theta_5$</td>
</tr>
<tr>
<td>6</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>$\theta_6-90$</td>
</tr>
<tr>
<td>7</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>$\theta_7$</td>
</tr>
</tbody>
</table>
The model considered for the humanoid robot has an interesting feature of a repeated serial robot chain structure, where the two arms, two legs and the torso have the same arrangement of joints. Hence, the set of spherical-revolute-spherical joints is repeated five times for the overall humanoid robot design. The D-H parameters extended to whole human body are given in Table 3.2. This is a major advantage since the same kinematics terms may be applied five times.
Figure 3-2 Nomenclature used for D-H Parameters of the Humanoid Robot
Table 3-2 D-H Parameters for the Humanoid Robot

<table>
<thead>
<tr>
<th>Notation</th>
<th>Joint</th>
<th>$i$</th>
<th>$\alpha_{i-1}$</th>
<th>$a_{i-1}$</th>
<th>$d_i$</th>
<th>$\theta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRUNK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_1$</td>
<td>Pelvis</td>
<td>1</td>
<td>$0^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_1$</td>
</tr>
<tr>
<td>$P_2$</td>
<td>Pelvis</td>
<td>2</td>
<td>$+90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_{2+90}$</td>
</tr>
<tr>
<td>$P_3$</td>
<td>Pelvis</td>
<td>3</td>
<td>$+90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_3$</td>
</tr>
<tr>
<td>$T$</td>
<td>Vertebra</td>
<td>4</td>
<td>$0^0$</td>
<td>$L_{T1}$</td>
<td>0</td>
<td>$\theta_4$</td>
</tr>
<tr>
<td>$N_1$</td>
<td>Neck</td>
<td>5</td>
<td>$0^0$</td>
<td>$L_{T2}$</td>
<td>0</td>
<td>$\theta_5$</td>
</tr>
<tr>
<td>$N_2$</td>
<td>Neck</td>
<td>6</td>
<td>$-90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_{6-90}$</td>
</tr>
<tr>
<td>$N_3$</td>
<td>Neck</td>
<td>7</td>
<td>$-90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_7$</td>
</tr>
<tr>
<td><strong>LEFT HAND</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$LS_1$</td>
<td>Left Shoulder</td>
<td>8</td>
<td>$0^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_8$</td>
</tr>
<tr>
<td>$LS_2$</td>
<td>Left Shoulder</td>
<td>9</td>
<td>$+90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_{9+90}$</td>
</tr>
<tr>
<td>$LS_3$</td>
<td>Left Shoulder</td>
<td>10</td>
<td>$+90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_{10}$</td>
</tr>
<tr>
<td>$LE$</td>
<td>Left Elbow</td>
<td>11</td>
<td>$0^0$</td>
<td>$L_{LA1}$</td>
<td>0</td>
<td>$\theta_{11}$</td>
</tr>
<tr>
<td>$LW_1$</td>
<td>Left Wrist</td>
<td>12</td>
<td>$0^0$</td>
<td>$L_{LA2}$</td>
<td>0</td>
<td>$\theta_{12}$</td>
</tr>
<tr>
<td>$LW_2$</td>
<td>Left Wrist</td>
<td>13</td>
<td>$-90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_{13-90}$</td>
</tr>
<tr>
<td>$LW_3$</td>
<td>Left Wrist</td>
<td>14</td>
<td>$-90^0$</td>
<td>0</td>
<td>0</td>
<td>$\theta_{14}$</td>
</tr>
</tbody>
</table>
Table 3-2 continued

<table>
<thead>
<tr>
<th>RIGHT HAND</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RS₁</strong> Right Shoulder</td>
<td>15</td>
<td>0°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{15})</td>
</tr>
<tr>
<td><strong>RS₂</strong> Right Shoulder</td>
<td>16</td>
<td>+90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{16}+90)</td>
</tr>
<tr>
<td><strong>RS₃</strong> Right Shoulder</td>
<td>17</td>
<td>+90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{17})</td>
</tr>
<tr>
<td><strong>RE</strong> Right Elbow</td>
<td>18</td>
<td>0°</td>
<td>(L_{RA1})</td>
<td>0</td>
<td>(\theta_{18})</td>
</tr>
<tr>
<td><strong>RW₁</strong> Right Wrist</td>
<td>19</td>
<td>0°</td>
<td>(L_{RA2})</td>
<td>0</td>
<td>(\theta_{19})</td>
</tr>
<tr>
<td><strong>RW₂</strong> Right Wrist</td>
<td>20</td>
<td>-90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{20}-90)</td>
</tr>
<tr>
<td><strong>RW₃</strong> Right Wrist</td>
<td>21</td>
<td>-90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{21})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEFT LEG</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LHP₁</strong> Left Hip</td>
<td>22</td>
<td>0°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{22})</td>
</tr>
<tr>
<td><strong>LHP₂</strong> Left Hip</td>
<td>23</td>
<td>+90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{23}+90)</td>
</tr>
<tr>
<td><strong>LHP₃</strong> Left Hip</td>
<td>24</td>
<td>+90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{24})</td>
</tr>
<tr>
<td><strong>LK</strong> Left Knee</td>
<td>25</td>
<td>0°</td>
<td>(L_{LL1})</td>
<td>0</td>
<td>(\theta_{25})</td>
</tr>
<tr>
<td><strong>LA₁</strong> Left Ankle</td>
<td>26</td>
<td>0°</td>
<td>(L_{LL2})</td>
<td>0</td>
<td>(\theta_{26})</td>
</tr>
<tr>
<td><strong>LA₂</strong> Left Ankle</td>
<td>27</td>
<td>-90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{27}-90)</td>
</tr>
<tr>
<td><strong>LA₃</strong> Left Ankle</td>
<td>28</td>
<td>-90°</td>
<td>0</td>
<td>0</td>
<td>(\theta_{28})</td>
</tr>
</tbody>
</table>
Table 3-2 continued

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RIGHT LEG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHP1</td>
<td>Right Hip</td>
<td>29</td>
<td>$0^\circ$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RHP2</td>
<td>Right Hip</td>
<td>30</td>
<td>$+90^\circ$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RHP3</td>
<td>Right Hip</td>
<td>31</td>
<td>$+90^\circ$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RK</td>
<td>Right Knee</td>
<td>32</td>
<td>$0^\circ$</td>
<td>$L_{RL1}$</td>
<td>0</td>
</tr>
<tr>
<td>RA1</td>
<td>Right Ankle</td>
<td>33</td>
<td>$0^\circ$</td>
<td>$L_{RL2}$</td>
<td>0</td>
</tr>
<tr>
<td>RA2</td>
<td>Right Ankle</td>
<td>34</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RA3</td>
<td>Right Ankle</td>
<td>35</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### 3.2 Forward Kinematics

This section presents the forward pose kinematic solution for the proposed humanoid robot. For a forward pose kinematics problem, the Cartesian pose (position and orientation) of the manipulator links is found for a given set of joint angles. For a specific configuration of joint angles, the Cartesian pose solution is unique.

Craig (2005) is used for finding the forward position kinematics solution. An initial step in the solution requires the neighboring transformation matrices. The general form of transformation matrix that defines frame $i$ with respect to neighboring frame $i-1$ is given by (3.1) (Craig, 2005). The size of this matrix is 4x4. The transformation matrix has the elements are functions of D-H parameters, $a_{i-1}$, $a_{i}$, $d_{i}$, and $\theta_{i}$. 
$$\begin{bmatrix}
\cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\
\sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\
\sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\
0 & 0 & 1 & 1
\end{bmatrix} \quad (3.1)$$

The homogenous transformation matrix (3.1) is constructed such that the rotation matrix and translation vector are included in a single matrix:

$$\begin{bmatrix}
^A_R^R \\
^A_P_B \\
0 & 0 & 0 & 1
\end{bmatrix} \quad (3.2)$$

In (3.2), $^A_R^R$ is the rotation matrix which gives the orientation of frame B with respect to frame A. The size of this matrix is 3x3. The vector $^A_P_B$ is the 3x1 position vector giving the Cartesian position of the origin of frame B with respect to frame A.

The forward pose kinematic solution for the humanoid robot arm is calculated first. Since the transformation matrix is given for the current joint with respect to the previous link, the number of neighboring transformation matrices will be equal to the number of links in the manipulator. In order to include the hand, an additional
transformation must be used. Equation 3.3 shows the transformation matrix of frame \{H\} with respect to the last active frame \{7\}. This transformation matrix is constant.

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & h \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(3.3)

Using the equation 3.1, the transformation matrices \(\begin{array}{llll}
_0^1T, _1^2T, _2^3T, _3^4T, _4^5T, _5^6T \text{ and } _6^7T
\end{array}\) are found. These transformations are with respect to the neighboring frames. The transformations are converted to express in terms of the base frame using the set of equations (3.4). The forward pose kinematics transformation \(\begin{array}{llll}
_0^\text{n}T
\end{array}\) recursively relates the moving frame \{n\} with respect to the base frame \{0\}.

\[
\begin{align*}
_0^2T &= _1^2T \\
_0^3T &= _2^3T \\
_0^4T &= _3^4T \\
_0^5T &= _4^5T \\
_0^6T &= _5^6T \\
_0^7T &= _6^7T \\
_0^\text{H}T &= _7^\text{H}T
\end{align*}
\]

(3.4)
As explained earlier, in (3.2), $^{\wedge}P_B$ gives the position of the frame \{B\} with respect to \{A\}. Thus, $^0P_3$ gives the position of shoulder with respect to the base frame. $^0P_4$, $^0P_7$ and $^0P_{II}$ give the positions of elbow, wrist and hand, respectively, with respect to the base frame. The resultant transformation matrices with respect to the neighboring link are listed in the Appendix A.

The same method is now extended to obtain the forward position kinematics solution for the overall 35 degrees-of-freedom humanoid robot. The transformation matrices are calculated for all the joints with respect to pelvis, the base frame. The positions of the joints are plotted. The transformation matrices with respect to the neighboring link, are listed in the Appendix B.
CHAPTER 4: RESOLVED RATE CONTROL

This section presents the resolved rate control of the humanoid robot arm. Resolved rate control is one of the most commonly used methods for commanding motion in case of arms. In this method, the velocity of the end effector is given as input. This is resolved into joint angle rates of all the links that are needed to achieve the movement of the end effector. The resolved rate control problem is based on the inverse velocity solution. The problem is defined as to find the joint rates for the given Cartesian rates. This is achieved by the formation of manipulator kinematics using Jacobian matrix. The linear and angular velocity of each of the links in the model under study is calculated. When calculating the velocity of a link, the velocity due to the previous link should be considered along with the components of its own. The method for the velocity propagation from link to link in Craig (2005) is followed. The humanoid robot arm is simulated using resolved rate control and graphs of various parameters with time are plotted.

Resolved rate motion control commands the movement of the end-effector of the manipulator in a required Cartesian direction and orientation, controlling the rate at which the joints move. This can be useful in applications in which the accurate motion of the end-effector is important as in case of welding, spray painting, etc.,. The motions of the various joints are resolved and separately controlled. Several motors at various joints run together, but at different rates to achieve the desired motion. This method gives a simplified sequence of motions for a required task.
4.1 Jacobian Matrix

The Jacobian matrix is a multi-dimensional form of derivative that maps the velocities in the joint space to Cartesian description, thus describing the geometric relationship between the joint rates and the velocity of end-effector. The size of a Jacobian matrix is the number of Cartesian degrees of freedom the robot operates in by the number of robot joints. Hence, for a seven axis manipulator such as in this thesis, the size of the Jacobian matrix $J$ is 6x7. Singularities are a potential problem in the Jacobian matrix and thus in resolved rate control.

In the resolved rate control problem, we calculate the joint rates using the following equation.

$$\dot{\Theta} = J^{-1} \cdot V$$  \hspace{1cm} (4.1)

In the equation above, $\dot{\Theta}$ is an array of joint rates of all the links in the manipulator. This can be used to find the desired joint angles of the links. The equations used to find the joint angles are given in the next section of this chapter.

$V$ is the vector of Cartesian velocities. The Cartesian velocities of the end effector are constituted by the linear and angular velocities. With linear velocity $v$ and angular velocities $w$, the Cartesian velocity is given as follows.

$$V = [v \  w]^T$$  \hspace{1cm} (4.2)
4.2 Programming and Simulation Results

The humanoid robot arm is simulated in MATLAB using the resolved rate control method. By this method, we can find the joint velocity profiles for the specified Cartesian velocity trajectory for all time. Given the required joint rates at each simulated control step, we use numerical integration to find the next joint angles to command to the robot. In the following equation, $\Theta$ is the set of joint angles of the links of the manipulator and $\dot{\Theta}$ is the set of joint rates.

$$\Theta = \Theta + \dot{\Theta} \Delta t$$ (4.3)

The inputs for a resolved rate control problem are the Cartesian velocities, the current joint angles and the DH-parameters. The initial position is defined and the arm is simulated over a period of time. The simulation is for a certain task (moving a ball from one point to another for demonstration in this thesis). The invalid configurations, where Jacobian is a singular matrix, are avoided in the simulation. Also, in this case since the Jacobian is not a square matrix, a MATLAB function \textit{pinv} (pseudoinverse) is used.
The simulation is done with an initial joint angle configuration of

\[ \Theta = \{0, 90, 0, 70, 0, -90, 0\} \text{ in degrees.} \]

The linear velocity of the end effector is given as

\[ \nu = [0.05 \ 0.05 \ 0.05]^T. \]

The angular velocity of the end effector is \( w = [0.05 \ 0 \ 0]^T. \) Thus, the
Cartesian velocity of the end effector is $V = [0.05\ 0.05\ 0.05\ 0.05\ 0\ 0]^T$. The arm is simulated for a time period of 3 seconds with an increment of 0.1 seconds. Figure 4.1 shows the initial configuration of the humanoid arm. The plots show the arm in four different views for clarity. The associated m-file code for the program can be found in Appendix C.

![Figure 4-2 Final Position of the Humanoid Arm](image-url)
Figure 4.2 shows the arm after a time period of 3 seconds which is the final position. In this configuration, the manipulator arm has already moved the ball from one surface to another with the given Cartesian velocity for the hand.

The following figures show the associated plots of the variables in this resolved rate simulation: joint angles, angular velocities, Cartesian pose, Euler angles, and det(J*J') with time. Figure 4.3 shows the values of the joint angles of all the joints of the manipulator arm plotted against time in seconds.
Figure 4-3 \( \theta \) (radians) vs. time (sec) Plots
Figure 4.4 shows the values of the joint rates of all the seven joints of the manipulator arm plotted against time in seconds.
The left column of the figure 4.5 in the following page shows the values of the Cartesian position in meters for the end effector expressed with respect to the base, plotted against time in seconds. The right column of the figure is a plot between the Euler angles of the end effector in radians versus time in seconds.

Figure 4-5 Cartesian Position (m) and Euler Angles (rad) vs. time (sec) for End Effector

Figure 4.6 shows the values of det(J*J’) plotted against time. This graph is to observe the singularities in the Jacobian matrix. When the value of det(J*J’) becomes zero, it means that the configuration cannot be achieved. In this resolved rate control
demonstration, the singular configurations are avoided. In this case, the manipulator is simulated for a time of only 3 seconds to avoid running into singularities.

![Graph showing Det (J * Jt) vs. time (sec)](image)

**Figure 4-6 Det (J * Jt) vs. time (sec)**

### 4.3 Application of Resolved Rate Control to Humanoid

For demonstration in this thesis, the resolved rate control method is also applied to the Humanoid. The programming is done as explained in the earlier section. The humanoid is simulated for a movement in the left arm, for a task of moving a ball from
one point to another. In this case, the base of the Humanoid is the pelvis and the left hand is considered the end effector.

The size of the Jacobian matrix for the humanoid is 6 x 35, since the humanoid has 35 degrees of freedom. For the simulation, the Cartesian velocity of the end-effector is taken as $V=[50 \ 50 \ 50 \ 0 \ 0 \ 0]^T$. These are converted into the required joint rates using the equation 4.1. The corresponding joint angles are calculated using the equation 4.3. Since, the task is to move a ball with the left hand; motion is constrained only to the joints of the shoulder, elbow and the hand of the left hand. The forward pose kinematic solution gives the position and orientation of the humanoid for these joint angles. The configurations of the humanoid are plotted in four different views for clarity. The initial configuration of the humanoid is the vertical position where the legs of the Humanoid are aligned together and the elbow is bent to hold the ball from surface it is initially placed on. This configuration is shown in the figure 4.7.
The Humanoid is simulated for a time period of 2 seconds. Figure 4.8 shows the final position of the humanoid after the hand moves to place the ball on the surface it had to be placed. Thus, with its left hand, the Humanoid has moved the ball from its initial surface to the final surface.
In this chapter, the theory and the programming for the simulation of humanoid arm using the resolved rate is presented. The humanoid is also simulated for walking and stair climbing using a different control algorithm from the literature. This is discussed in the following chapter.
CHAPTER 5:  HUMANOID ROBOT KINEMATIC SIMULATION

The humanoid robot discussed in this thesis is simulated for walking and also for stair climbing in this chapter. These common motions are particularly interesting because they are responsible for balancing and moving the humanoid from one point to another. Such simulations could be used to demonstrate human-like mobility attained by rhythmic movements of the body parts.

We discuss the simulation of the above-mentioned motions of the humanoid using MATLAB as the programming tool. The trajectory of the joint angles for walking is modified from Lum (1999). In this paper, the authors present trajectory planning of the joint angles of a five link biped for stable walking and also for stair climbing. The locomotion is constrained to a plane perpendicular to the ground that separates the biped into left and right halves (Sagittal Plane). Their model describes five links, which consists of the two legs and the torso driven by independent DC motors.

5.1 Trajectory for Walking

The human gait cycle for walking can be broken into various phases depending on how the feet are in contact with the ground. The three phases in the biped walking are the starting step from the vertical position and two steady walking steps. For demonstration in this thesis, the humanoid walking is started with right leg. In the first step, from the
vertical position of the humanoid, the right leg is moved forward and placed on the ground with support of the left leg. The first steady walking step involves lifting the left leg with single leg support of the right leg until the left leg is planted on the ground again. The second steady walking step is similar to the first steady walking step, but this step has single leg support of the left leg until the right leg is lifted and planted on the ground again. The consecutive repetitions of steady walking steps after the implementation of first step result in a continued locomotion in the sagittal plane. At the end of each step the biped has two legged support. This walking can be defined as one starting step and a repetition of steady walking steps with left and right feet alternately switched for supports. The results are provided in the form of graphs with joint angles against time giving the trajectory, adapted from Lum (1999).

The first step of the walking cycle in Lum (1999) takes 0.5 seconds for execution. But there is a dwell in the stride after a time period of 0.2 seconds for 0.3 seconds, when there is no joint angle change in any of the joints. In the following steps, with an execution time of 0.5 seconds, there is a dwell time after 0.25 seconds for 0.25 seconds. To avoid these phases where the humanoid will be stationary, each stride is considered without the dwell. This results in a continuous motion of the humanoid. For motion appearance, a displacement of pelvis is implemented. This is achieved by incrementing the pelvis displacement coordinates in the base frame, which has the pelvis in a horizontal plane for each step.
Along with the movement of the hip and leg joints, the walking cycle of the humanoid also includes movement in the shoulder and hand joints. Thus for a human-like motion, a swinging of the arms is added from a general observation of the human walking, where the human moves the opposite arm in the same direction as the leg to add stability. Humans move the left arm back when moving the left leg forward and vice-versa. A flexion and extension of the joint angles of the arms alternating between right and left arms is added to the simulation to demonstrate the swinging of arms. Similar to a walking cycle, the arm motion can be divided into three phases as the arms movements have to be synchronized with the leg movements. This is constituted by a first step in which the arms are moved away from the vertical starting position where the arms are aligned with each other and the two steady movements. In the first step the arm opposite to the leg moving forward is also moved forward. Then alternate arms are moved slightly in opposite directions.

5.2 Simulation and Results for Humanoid Walking

The humanoid model discussed in this thesis is simulated using MATLAB, due to the simplicity of its use and good quality graphics. A program ‘humanoid_walking’ created for simulation and various other functions that support the main program are discussed in this section. The corresponding m-file program for ‘humanoid_walking’ is shown in the Appendix D.
The lengths and the radii of the links of the humanoid robot in meters are obtained by measuring a human subject. These are shown in the figure 5.1. Initial joint angles (in radians) are defined for vertical position where the hands and legs of the humanoid hands are aligned to each other. Table 5.1 gives the lengths, radii and the initial joint angles of the humanoid. Figure 5.2 shows this initial configuration of the Humanoid robot.
Figure 5-1 Lengths and Radii of the Links of the Humanoid
Table 5-1 Lengths, Radii and Initial Joints of the Links of the Humanoid

<table>
<thead>
<tr>
<th>Joint</th>
<th>i</th>
<th>L (meters)</th>
<th>R (meters)</th>
<th>Θ (Radians)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUNK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pelvis</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pelvis</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pelvis</td>
<td>3</td>
<td>0.25</td>
<td>0.1194</td>
<td>0</td>
</tr>
<tr>
<td>Vertebrae</td>
<td>4</td>
<td>0.3</td>
<td>0.1194</td>
<td>0</td>
</tr>
<tr>
<td>Neck</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Neck</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Neck</td>
<td>7</td>
<td>0.1194</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>LEFT HAND</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left Shoulder</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Shoulder</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Shoulder</td>
<td>10</td>
<td>0.275</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>Left Elbow</td>
<td>11</td>
<td>0.25</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>Left Wrist</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Wrist</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>π/2</td>
</tr>
<tr>
<td>Left Wrist</td>
<td>14</td>
<td>0.175</td>
<td>0.0358</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 5-1 continued

<table>
<thead>
<tr>
<th>RIGHT HAND</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Shoulder</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Right Shoulder</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Right Shoulder</td>
<td>17</td>
<td>0.275</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>Right Elbow</td>
<td>18</td>
<td>0.25</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>Right Wrist</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Right Wrist</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>$\pi/2$</td>
</tr>
<tr>
<td>Right Wrist</td>
<td>21</td>
<td>0.175</td>
<td>0.0358</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEFT LEG</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Hip</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Hip</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Hip</td>
<td>24</td>
<td>0.45</td>
<td>0.0517</td>
<td>0</td>
</tr>
<tr>
<td>Left Knee</td>
<td>25</td>
<td>0.475</td>
<td>0.0517</td>
<td>0</td>
</tr>
<tr>
<td>Left Ankle</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Ankle</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Left Ankle</td>
<td>28</td>
<td>0.2125</td>
<td>0.0517</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 5-1 continued …

<table>
<thead>
<tr>
<th>Right Hip</th>
<th>29</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Hip</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Right Hip</td>
<td>31</td>
<td>0.45</td>
<td>0.0517</td>
<td>0</td>
</tr>
<tr>
<td>Right Knee</td>
<td>32</td>
<td>0.475</td>
<td>0.0517</td>
<td>0</td>
</tr>
<tr>
<td>Right Ankle</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Right Ankle</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Right Ankle</td>
<td>35</td>
<td>0.2125</td>
<td>0.0517</td>
<td>0</td>
</tr>
</tbody>
</table>
The trajectory of the joint angles obtained from the paper is converted into a set of equations defining the relationship between the joint angles and time for various links of the humanoid during a walk cycle. A user-defined function ‘legangledata’ is created to define these equations. In this function, the equations are given for various links at different time periods. For each link the times and the joint angles are collected as arrays in steps of time which is same as the time increment for every iteration of the simulation. Thus for a time period of 1.4 seconds, there are two sets of arrays for the time and joint
angles for the hip, knee, shoulder and elbow joints which define motion of legs and the swinging of arms. The first 0.4 seconds represent the starting step. The motion between 0.4 to 1.4 seconds can be repeated thereafter, to define a continuous motion. The inputs for the user-created function, ‘legangledata’ is ‘step’ which is assigned in the main program, ‘humanoid_walking’ and the outputs are q1, q2, q3, q4, ank, and shldr which are arrays of joint angle values of the leg and hand joints. The notations for these joints angles are explained in the figure 5.3. The trajectories of these joint angles for simulation of humanoid walking are shown in figure 5.4.
Figure 5-3 Notation for the Joint Angles of the Leg Joints
The humanoid is simulated for a time period of 2.4 seconds to demonstrate walking. A variable ‘increment’ is introduced to represent the motion of the pelvis in the horizontal direction. This is to show the movement of the humanoid in the sagittal plane. This is initially set to zero. A variable ‘time’ which tracks the time is also set to zero. An increment in time is defined as ‘step’ which is given a value of 0.1. Hence for each iteration ‘time’ is increased by ‘step’, until ‘time’ is equal to 2.4. This is done using a for loop. Another parameter “j” is introduced to read the values for the arrays returned.
by the function “legangledata”. In the walking motion the joints that are varying are the shoulder, ankle, hip and the knee joints. As explained in the figure 5.3, the corresponding joint angles are t8, t11, t15, t18, t24, t25, t31 and t32. These are assigned to the corresponding arrays from the function ‘legangledata’ using equations for each ‘j’ by the following formulae.

\[
\begin{align*}
    t_{24} &= q_2(j), \\
    t_{25} &= q_1(j), \\
    t_{31} &= -q_3(j), \\
    t_{32} &= q_4(j).
\end{align*}
\]  

(5.1)

For this set of joint angles for each iteration corresponding to an increment of ‘step’, the transformations are found using the function ‘coordinates’ which returns the x, y, z coordinates of all the joints of the humanoid and the transformations with respect to the base frame which is the pelvis. The inputs for the function ‘coordinates’ are the joint angles and lengths of the various joints and links. For each iteration, the coordinates of all the links are obtained for the corresponding configuration of the humanoid robot based on the joint angles at that instant governed by the equations from ‘legangledata’. The corresponding m-file program is shown in the Appendix E.

Plotting the coordinates of the joints obtained by this function would result in stick figure. To obtain better graphics, the links of the humanoid robot can be represented
as cylinders. From the radii of all the links of the humanoid, the points on the cylinder are
to be calculated. This is obtained by using a function ‘cylcoords’ which gives the points
on a cylinder. The ball and socket joints of the humanoid with three coincident degrees of
freedom can be represented as spheres. The points of these spheres are obtained using the
function ‘ballnsocket’. The head and neck coordinates are also represented by a sphere
and cylinder respectively. The functions ‘ballnsocket’ and ‘cylcoords’ are used to obtain
the points on the surfaces of the head and neck. Another function by name ‘finalcoords’
is created to filter out the coordinates of the joints obtained the ‘cylcoords’ that are used
for plotting. Also, a surface is created to represent the floor.

To plot the results for each walking iteration, a built-in MATLAB function, ‘surf’
is called. This command plots the three dimensional surface for the matrix arguments of
the ‘finalcoords’ in the form of \( \text{surf}(X,Y,Z) \). The graphics are shown in four
different views for clarity. These are shown on the same screen using subplots.

Figure 5.5 shows the plots of the joint angles of the leg and arm joints versus
time. These are the joint angles resulted from the simulation of the humanoid for walking
for a time of 2.4 seconds. It can be seen from these plots how after the first 0.4 seconds of
simulation, the steps between 0.4 and 1.4 seconds are repeated.
The following figures show the humanoid in walking at different points of the simulation. Figure 5.6 shows the humanoid at the end of 0.4 seconds which is the end of the first starting step. Figure 5.7 shows the humanoid at the end of 0.9 seconds which is the end of the first steady step. Figure 5.8 shows the simulation of the Humanoid Robot for a time period of 0.9 seconds. The increment for each iteration of simulation is 0. Seconds. At the end of 0.9 seconds, the humanoid completes the first steady step.
Figure 5-6 Humanoid at the End of the Starting Step after 0.4 seconds

Figure 5-7 Humanoid at the End of the First Steady Step at the End of 0.9 seconds
Figure 5-8 Humanoid in Walking with Plots at Increments of 0.1 secs for 0.9 secs
5.3 Trajectory for Stair Climbing

The stair climbing cycle can also be broken into similar phases as the walking cycle. The three phases in the biped stair climbing are the starting step and the two steady steps that repeat. In the starting step, the humanoid lifts one leg and plants it on the first stair, with support from on the other leg on the ground. At the end of the first step, the humanoid has two-legged support. The first steady walking step involves the step where the leg on the ground is lifted up with single-leg support of the leg on the first stair till the other leg is planted again on the next stair. The second steady step is similar to the first steady step, but with the leg support shifted to the other leg. After the starting step is implemented, the two steady walking steps can be repeated to continue the locomotion in the sagittal plane. Similar to walking, the biped has two-legged support at the end of each step. Stair climbing can be defined as one starting step and a repetition of steady walking steps with left and right feet switched for support. The motion planning for stair climbing is provided in the form of graphs with joint angles against time giving the trajectory, adapted from Lum (1999).

Each cycle of stair climbing in Lum (1999) takes 0.5 seconds for execution. Similar to the walking cycle, a dwell in each stride after a time period of 0.25 seconds for 0.25 seconds is noticed. To avoid this phase where the humanoid will be stationary, each stride is considered without the dwell. This results in a continuous motion of the humanoid. In Lum (1999), at the start of each step the non-supporting leg is folded back
in 0.05 seconds. This results in an impulsive motion. In order to avoid this, in the simulation, the leg is folded back in 0.1 seconds. Another modification made in the joint angle trajectories obtained from the paper is to constrain the joint angle of the supporting leg such that the leg is always perpendicular to the stairs. Unlike walking on a horizontal plane, the displacement of pelvis in case of stair climbing is in both horizontal and vertical directions. This is achieved by incrementing the coordinates of the base in both these directions for each step. Arm swinging motion is also included for the stair climbing, just like for the walking gait.

5.4 Simulation and Results for Stair Climbing

As discussed earlier, the humanoid model discussed in this thesis is simulated using MATLAB. A program ‘humanoid_stairclimbing’ is created for simulation and various other functions that support the main program are discussed in this section. The lengths, radii and the initial joint angles of the links are same as mentioned for ‘humanoid_walking’. Figure 5.9 shows the initial vertical position of the humanoid robot, with the legs and arms aligned to each other.
The trajectory of the joint angles obtained from the paper is converted into a set of equations with a user-defined function ‘legangledat aclimbing’. The corresponding m-file program is shown in the Appendix F. The inputs for the function, is ‘step’ which is assigned in the main program, ‘humanoid_stairclimbing’ and the outputs are q1,q2,q3,q4,ank and shldr which are arrays of joint angle values of the leg and hand joints. This is shown in the figure 5.10, with plots of the joint angles versus time.
The humanoid is simulated for a time period of 1.5 seconds to demonstrate stair climbing. Variables ‘incrementx’ and ‘incrementy’ are introduced to represent the motion of pelvis in horizontal and vertical directions. This is initially set to zero. A variable ‘time’ which tracks the time is also initially set to zero. An increment in time is defined as ‘step’ which is given a value of 0.1. Hence for each iteration ‘time’ is increased by ‘step’, until ‘time’ is equal to 1.5. This is done using a for loop. Another parameter “j”
is introduced to read the values for the arrays returned by the function “legangledatalimbing”. In the stair climbing motion the joints that are varying are the shoulder, ankle, hip and the knee joints. These corresponding joint angles are t8, t11, t15, t18, t24, t25, t31 and t32. These are assigned to the corresponding arrays from the function ‘legangledatalimbing’ using equations for each ‘j’ in the program. For the set of joint angles for each iteration corresponding to an increment of ‘step’, the transformations are found using the function ‘coordinatesclimbing’ which returns the x, y, z coordinates of all the joints of the humanoid and the transformations with respect to the base frame which is the pelvis. The function is similar to ‘coordinates’ with a difference that accounts for movement in the vertical direction also. For each iteration, the coordinates of all the links are obtained for the corresponding configuration of the humanoid based on the joint angles at that instant governed by the equations from ‘legangledatalimbing’. The same functions, ‘cylcoords’, ‘ballnsocket’ and ‘finalcoords’ are used to obtain the coordinates of surface the cylinders and spheres for showing the links and joints. The built-in MATLAB function is used to plot the humanoid in 3-d. The graphics are shown in four different views on the same screen using subplots. For the stair climbing motion, the plots of the joint angles versus time can be seen in figure 5.11.
The following figures show the humanoid in stair climbing motion at different points of the simulation. Figure 5.12 shows the humanoid at the end of 0.5 seconds which is the end of the first starting step. Figure 5.13 shows the humanoid at the end of 1 second, which is the end of the first steady step. Figure 5.14 shows the simulation of the Humanoid Robot for a time period of 1.5 seconds. The increment for each iteration of simulation is 0.1 seconds. At the end of 1.5 seconds, the humanoid completes the second steady step of stair climbing.
Figure 5-12 Humanoid at the end of the starting step after 0.5 seconds of stair climbing
Figure 5-13 Humanoid at the end of the first steady step at the end of 1 second of stair climbing
Figure 5-14 Humanoid in Stair Climbing with Plots at Increments of 0.1 seconds for 1.5 seconds
In this chapter, we consider the development of dynamic equations of motion of the humanoid arm and the humanoid robot. An inverse dynamics model is developed for the humanoid arm. An attempt has been made to develop an inverse dynamics model for the overall humanoid in discussion. The problem of inverse dynamics is defined as to calculate the actuators’ torques that are required for the given kinematic motions (angles, angular velocities, and angular accelerations).

Inverse dynamics are essential for the control of a robot as they let us find the required inputs for the desired motion outputs. Development of a realistic dynamics model can help achieve precise control of a manipulator. Inverse dynamics is the basis of the popular computed-torque, or feedback linearization, control method.

6.1 Newton-Euler Formulation

There are two basic approaches to develop a realistic dynamic model, the Lagrange-Euler and the Newton-Euler formulation. Of the two methods, the Newton-Euler formulation is chosen for its computational efficiency for a large number of axes. With the Newton-Euler formulation, the effect of the forces and the torques on each individual link can be seen since each link of a serial-link manipulator is isolated at every instance.
The Newton-Euler formulation follows the algorithm in which the generalized torques are solved numerically. The algorithm is governed by the following equations as presented in Craig (2005).

**Outward Iterations for Kinematics and Inertial Loads (i = 0 to n)**

\[ i+1 \omega_{i+1} = i+1 R^i \omega_i + \dot{\theta}_{i+1} Z_{i+1} \]

\[ i+1 \omega_{i+1} = i+1 R^i \omega_i + i+1 R^i \omega_i \times \dot{\theta}_{i+1} Z_{i+1} + \ddot{\theta}_{i+1} Z_{i+1} \]

\[ i+1 \omega_{i+1} = i+1 R^i \omega_i \times i+1 P_{i+1} + i+1 \omega_i \times \left( \omega_i \times i+1 P_{i+1} \right) + v_i \]

\[ i+1 \omega_{i+1} = i+1 \omega_{i+1} \times i+1 P_{\omega i+1} + i+1 \omega_{i+1} \times \left( i+1 \omega_{i+1} \times i+1 P_{\omega i+1} \right) + v_{i+1} \]

\[ i+1 F_{i+1} = m_{i+1} v_{C_{i+1}} \]

\[ i+1 N_{i+1} = C_{i+1} I_{i+1} \omega_{i+1} + C_{i+1} I_{i+1} \omega_{i+1} \times C_{i+1} I_{i+1} \omega_{i+1} \]

\[(6.1)\]
Inward Iterations for Kinetics ($i = n+1$ to 1)

\[ i \mathbf{f}_i = i \mathbf{R}^{i+1} \mathbf{f}_i + i \mathbf{F}_i \]

\[ i \mathbf{n}_i = i \mathbf{N}_i + i \mathbf{R}^{i+1} \mathbf{n}_{i+1} + i \mathbf{P}_c \times \mathbf{F}_i + i \mathbf{P}_l \times i \mathbf{R}^{i+1} \mathbf{f}_{i+1} \]

\[ \tau_i = i \mathbf{n}_i \mathbf{Z}_i \]  \hspace{1cm} (6.2)

In the above equations,

- $\omega$ is the rotational velocity,
- $\dot{\omega}$ is the rotational acceleration,
- $\mathbf{v}$ is the linear acceleration,
- $\mathbf{v}_c$ is the linear acceleration of the center of mass,
- $\mathbf{P}_c$ is the position vector of the centroid from the local frame,
- $\mathbf{F}$ is the inertial force of the center of mass,
- $\mathbf{N}$ is the inertial torque of the center of mass,
- $\mathbf{f}_i$ is the force exerted by the link $i$-1 on link $i$,
- $\mathbf{n}_i$ is the torque exerted by the link $i$-1 on link $i$,
- $\tau_i$ is the actuator torque (for revolute joints, the $Z$ components of $\mathbf{n}_i$).
The algorithm has two sets of iterations, the outward and the inward. For the outward iterations, the velocity and the accelerations of each of the links starting from the base to the end-effector are calculated recursively. Starting with the first link, for the joint displacement, joint velocity and acceleration, the linear and angular velocities and accelerations of link centroid are found. The kinematic parameters of the first link are used to find the velocities and accelerations of the neighboring proximal link, which is the link closer to the base. The same is repeated for all the links. Using the results from the outward iterations, the forces and moments are calculated for all the links from the end-effector towards the base using the kinetics equations from the inward iterations. The required joint torques are determined in this phase.

6.2 Programming Results for Dynamics of Humanoid Arm

The dynamic model of the humanoid arm is developed in the simulated resolved rate framework. The base of the humanoid arm is the shoulder. Since this base is fixed, the initial linear velocity, angular velocity, and angular acceleration are set to zero. As discussed by Craig (2005), the effect of gravity can be included by accelerating the base by $g$ upwards. The lengths, radii and the initial joint angles of various links are given in the table 6.1.
Table 6-1 Lengths, Radii and Initial Joint Angles of the Humanoid Arm

<table>
<thead>
<tr>
<th>Joint</th>
<th>I</th>
<th>L (meters)</th>
<th>R (meters)</th>
<th>Θ (Radians)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shoulder</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Shoulder</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>π/2</td>
</tr>
<tr>
<td>Shoulder</td>
<td>3</td>
<td>0.275</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>Elbow</td>
<td>4</td>
<td>0.25</td>
<td>0.0358</td>
<td>0</td>
</tr>
<tr>
<td>Wrist</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wrist</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>-π/2</td>
</tr>
<tr>
<td>Wrist</td>
<td>7</td>
<td>0.175</td>
<td>0.0358</td>
<td>0</td>
</tr>
</tbody>
</table>

From the lengths and the joint angles, transformations are calculated and are expressed in base frame coordinates. For this, a function named ‘coordinates’ is called in the m-file. The function solves the translational and rotational positions of the various links and returns the transformations with respect to base frame. The mass of the humanoid model is taken as 150 pounds and is divided by the entire volume of the humanoid model and the density is calculated. The mass of each individual link is found by multiplying their volume with this density. These masses are used to find the moment of inertia of the links. In the principal axis form, the moment of inertia of a cylinder is given by the following formula.
\[
I = \begin{bmatrix}
\frac{1}{2} Mh^2 + \frac{1}{4} MR^2 & 0 & 0 \\
0 & \frac{1}{2} Mh^2 + \frac{1}{4} MR^2 & 0 \\
0 & 0 & \frac{1}{2} MR^2
\end{bmatrix}
\]

Where \( M \) is the mass, \( h \) is the length and \( R \) is the radius of each cylindrical link. The centers of the mass of each link are calculated by finding the midpoints of the links in \( X \), \( Y \) and \( Z \) axes.

The velocities of the model are taken exactly as they were in the resolved rate control the dynamic model in the resolved rate frame. Along with the initial conditions, the above-mentioned values of velocities, accelerations, moments of inertia, transformations, and centers of mass are substituted in the outward and inward iterations of the Newton-Euler algorithm given by equations (6.1) and (6.2). The resulting torques that should be applied at the joints is obtained. For the next iteration, the forces, angular velocities and accelerations from the earlier iteration are used. The M-file for the Newton- Euler Iteration is shown in Appendix G.

Figure 6.1 shows the change in the forces (Newtons) applied on various joints of the humanoid arm versus time, for the resolved rate control of the humanoid arm, presented in section 3.2 of Chapter 3.
Figure 6.2 shows the plot of the torques (Newton-meters) vs. time in seconds. These are the required torques for achieving the commanded resolved rate control motion to move the humanoid arm.
Figure 6-2 Plot of Forces vs. time of Various Joints of the Humanoid Arm

From the figures 6.1 and 6.2 it can be seen that the forces and torques on the last two links of the manipulator arm are zeroes. This is because there are no external forces acting in the direction of these links.

6.3 Inverse Dynamics for the Humanoid Walking

An attempt has been made to find the inverse dynamics for the overall humanoid robot. The Newton-Euler algorithm (Craig, 2005) was again implemented for the humanoid robot. Unlike the humanoid arm which is a set of serial links, the structure of
the humanoid proposed in this thesis is a branching set of serial links. The torso of the humanoid can be considered as a serial chain with the arms and legs as the serial sub-chains branching from the torso.

When implementing the Newton-Euler method for the humanoid, the outward and the inward iterations are calculated separately for each serial chain. A set of iterations is done for the torso with the neck as end-effector and the resultant forces, velocities and accelerations are used as the initial conditions for the outward and inward iterations for the humanoid arms. The same method is implemented to the legs. The results obtained for the inverse dynamics of the simulation of Humanoid walking, presented in section 5.2 of chapter 5 for 1.4 seconds, are shown in the figures 6.3 through 6.7. The torques obtained by this method are not accurate, since the moments and the reaction forces of the all the links are not included in finding the torques for the torso. From Figure 6.3, it can be observed that the torques on the joints 1 to 7, which are the joints of the trunk, are constant. This is because there is no change in the joint angles, for these joints during humanoid walking. The M-file for the Newton- Euler Iteration is shown in Appendix G.
Figure 6-3 Plot of Torques vs. time for Joints 1 through 7 (Joints of the Trunk)
Figure 6-4 Plot of Torques vs. time for joints 8 through 14 (Joints of Left Hand)
Figure 6-5 Plot of Torques vs. time for joints 15 through 21 (Joints of the Right Hand)
Figure 6-6 Plot of Torques vs. time for joints 22 through 28 (Joints of the Left Leg)
The dynamics solution of the overall humanoid should be obtained in the conventional way using the free body diagrams, to include the reacting forces and moments from each leg and arm onto the torso (ignored in the above modeling). Because of time constraints, this will be included in the Future Work Section.
CHAPTER 7: CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In this thesis, a 35 degrees-of-freedom humanoid robot design is proposed. The humanoid robot has five serial robot chains of seven degrees-of-freedom each. This joint structure of repeated spherical-revolute-spherical joints results in the same kinematics terms for the trunk, arms and legs of the humanoid. The D-H parameters are defined for the humanoid arm first and are extended to the entire humanoid robot. The parameters are similar for the trunk, arms and legs of the robot because of the repeated seven degrees-of-freedom structure. A forward position kinematics solution for the humanoid arm and the entire humanoid is calculated based on Craig (2005). The humanoid arm is simulated using the resolved rate control method.

The humanoid robot is kinematically simulated for walking and stair climbing. The trajectories of the joint angles for the walking and stair climbing motions are adapted from Lum (1999). These simulations demonstrate human-like locomotion in the sagittal plane. Finally, the inverse dynamics problem (where the torques required for obtaining the resolved rate control motion of the humanoid arm) is solved using the iterative Newton-Euler formulation obtained from Craig (2005). An attempt has also been made to find the Inverse dynamics of the humanoid for walking. In the implementation of the Newton-Euler method for the humanoid, the outward and the inward iterations are calculated separately for each serial chain of the trunk, legs and the arms of the
humanoid. The resultant torques obtained by the application of Newton- Euler Algorithm obtained are extremely high. This is because the reacting forces and moments from the arms and legs are not accounted in the assumptions in the implementation of the Newton-Euler formulation explained above.

7.2 Future Work

It is found that the results obtained, from the application of the Newton-Euler formulation with the outward and inward iterations calculated separately for each serial chain, are not accurate. The equations of the motion of the overall humanoid for walking and stair climbing should be attempted using the conventional method with free body diagrams to include the reacting forces and moments from the arms and legs of the trunk of the humanoid.

Future work should focus on building the proposed humanoid robot with DC motors for the joints. The programs developed in the thesis are for observing the behavior of the humanoid for representative tasks. The software can be developed for controlling the humanoid in real time. Also, in addition to observing the behavior of the humanoid locomotion, the programs can be extended to observe the behavior of human locomotion, for applications in the field of bio-mechanics.
Walking and climbing can be combined with grasping resulting in more useful applications. Walking and stair climbing motions are presented in only one plane. These can be extended to 3-D locomotion.
REFERENCES


Web References:

http://world.honda.com/ASIMO/

http://www.humanoid.rise.waseda.ac.jp

http://www.sony.net/SonyInfo/QRIO/top_nf.html

http://www.toyota.co.jp/en/special/robot/

http://www.ai.mit.edu/projects/humanoid-robotics-group/

http://www.ai.mit.edu/projects/leglab/robots/m2/m2.html

http://www.itee.uq.edu.au/~damien/GuRoo/

http://www.kawada.co.jp/global/ams/hrp_2.html
APPENDIX A: TRANSFORMATION MATRICES OF THE LINKS OF THE HUMANOID ARM

\[ T_{01} = \begin{bmatrix} \cos(t_1), & -\sin(t_1), & 0, & 0; \\ \sin(t_1), & \cos(t_1), & 0, & 0; \\ 0, & 0, & 1, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]

\[ T_{12} = \begin{bmatrix} -\sin(t_2), & -\cos(t_2), & 0, & 0; \\ 0, & 0, & -1, & 0; \\ \cos(t_2), & -\sin(t_2), & 0, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]

\[ T_{23} = \begin{bmatrix} \cos(t_3), & -\sin(t_3), & 0, & 0; \\ 0, & 0, & -1, & 0; \\ \sin(t_3), & \cos(t_3), & 0, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]

\[ T_{34} = \begin{bmatrix} \cos(t_4), & -\sin(t_4), & 0, & 11; \\ \sin(t_4), & \cos(t_4), & 0, & 0; \\ 0, & 0, & 1, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]

\[ T_{45} = \begin{bmatrix} \cos(t_5), & -\sin(t_5), & 0, & 12; \\ \sin(t_5), & \cos(t_5), & 0, & 0; \\ 0, & 0, & 1, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]

\[ T_{56} = \begin{bmatrix} \sin(t_6), & \cos(t_6), & 0, & 0; \\ 0, & 0, & 1, & 0; \\ \cos(t_6), & -\sin(t_6), & 0, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]

\[ T_{67} = \begin{bmatrix} \cos(t_7), & -\sin(t_7), & 0, & 0; \\ 0, & 0, & 1, & 0; \\ -\sin(t_7), & -\cos(t_7), & 0, & 0; \\ 0, & 0, & 0, & 1 \end{bmatrix}; \]
APPENDIX B: TRANSFORMATION MATRICES OF THE LINKS OF THE HUMANOID

\[
o_{T_p1}= \begin{bmatrix}
sin(t1), & cos(t1), & 0, & 0; \\
0, & 0, & -1, & 0; \\
-cos(t1), & sin(t1), & 0, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
p_{1_{T}p2}= \begin{bmatrix}
-sin(t2), & -cos(t2), & 0, & 0; \\
0, & 0, & -1, & 0; \\
cos(t2), & -sin(t2), & 0, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
p_{2_{T}p3}= \begin{bmatrix}
cos(t3), & -sin(t3), & 0, & 0; \\
0, & 0, & -1, & 0; \\
sin(t3), & cos(t3), & 0, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
p_{3_{T}t}= \begin{bmatrix}
cos(t4), & -sin(t4), & 0, & l_{t1}; \\
sin(t4), & cos(t4), & 0, & 0; \\
0, & 0, & 1, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
t_{T}n1= \begin{bmatrix}
cos(t5), & -sin(t5), & 0, & l_{t2}; \\
sin(t5), & cos(t5), & 0, & 0; \\
0, & 0, & 1, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
n_{1_{T}n2}= \begin{bmatrix}
sin(t6), & cos(t6), & 0, & 0; \\
0, & 0, & 1, & 0; \\
cos(t6), & -sin(t6), & 0, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
n_{2_{T}n3}= \begin{bmatrix}
cos(t7), & -sin(t7), & 0, & 0; \\
0, & 0, & 1, & 0; \\
-sin(t7), & -cos(t7), & 0, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
n_{3_{T}l_{s}1}= \begin{bmatrix}
-cos(t8), & sin(t8), & 0, & l_{s}; \\
sin(t8), & cos(t8), & 0, & 0; \\
0, & 0, & -1, & -d_{8}; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]

\[
l_{s1_{T}l_{s}2}= \begin{bmatrix}
-sin(t9), & -cos(t9), & 0, & 0; \\
0, & 0, & -1, & 0; \\
cos(t9), & -sin(t9), & 0, & 0; \\
0, & 0, & 0, & 1; \\
\end{bmatrix}
\]
\[ ls2Tls3 = [\cos(t10), -\sin(t10), 0, 0; 0, 0, -1, 0; \sin(t10), \cos(t10), 0, 0; 0, 0, 0, 1]; \]

\[ ls3Tle = [\cos(t11), -\sin(t11), 0, lla1; 0, 0, 0, 0; \sin(t11), \cos(t11), 0, 0; 0, 0, 1, 0]; \]

\[ leTlw1 = [\cos(t12), -\sin(t12), 0, lla2; 0, 0, 0, 0; \sin(t12), \cos(t12), 0, 0; 0, 0, 0, 1]; \]

\[ lw1Tlw2 = [\sin(t13), \cos(t13), 0, 0; 0, 0, 1, 0; \cos(t13), -\sin(t13), 0, 0; 0, 0, 0, 1]; \]

\[ lw2Tlw3 = [\cos(t14), -\sin(t14), 0, 0; 0, 0, 0, 0; -\sin(t14), -\cos(t14), 0, 0; 0, 0, 0, 1]; \]

\[ n3Trs1 = [-\cos(t15), \sin(t15), 0, -ls; 0, 0, 0, 0; \sin(t15), \cos(t15), 0, 0; 0, 0, -1, d8; 0, 0, 0, 1]; \]

\[ rs1Trs2 = [-\sin(t16), -\cos(t16), 0, 0; 0, 0, 0, 0; \cos(t16), -\sin(t16), 0, 0; 0, 0, 0, 1]; \]

\[ rs2Trs3 = [\cos(t17), -\sin(t17), 0, 0; 0, 0, 0, 0; -\sin(t17), \cos(t17), 0, 0; 0, 0, 0, 1]; \]

\[ rs3Tre = [\cos(t18), -\sin(t18), 0, lra1; 0, 0, 0, 0; \sin(t18), \cos(t18), 0, 0; 0, 0, 1, 0]; \]

\[ reTrw1 = [\cos(t19), -\sin(t19), 0, lra2; 0, 0, 0, 0; \sin(t19), \cos(t19), 0, 0; 0, 0, 1, 0]; \]
\[ \begin{bmatrix}
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{rw1Trw2} = \begin{bmatrix}
sin(t20), & \cos(t20), & 0, & 0;
0, & 0, & 1, & 0;
\cos(t20), & -\sin(t20), & 0, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{rw2Trw3} = \begin{bmatrix}
\cos(t21), & -\sin(t21), & 0, & 0;
0, & 0, & 1, & 0;
-\sin(t21), & -\cos(t21), & 0, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{lhp1Tlhp2} = \begin{bmatrix}
-\sin(t23), & -\cos(t23), & 0, & 0;
0, & 0, & -1, & 0;
\cos(t23), & -\sin(t23), & 0, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{lhp2Tlhp3} = \begin{bmatrix}
\cos(t24), & -\sin(t24), & 0, & 0;
0, & 0, & -1, & 0;
\sin(t24), & \cos(t24), & 0, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{lhp3Tlkl} = \begin{bmatrix}
\cos(t25), & -\sin(t25), & 0, & 1111;
\sin(t25), & \cos(t25), & 0, & 0;
0, & 0, & 1, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{lklTlan1} = \begin{bmatrix}
\cos(t26), & -\sin(t26), & 0, & 1112;
\sin(t26), & \cos(t26), & 0, & 0;
0, & 0, & 1, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{lan1Tlan2} = \begin{bmatrix}
\sin(t27), & \cos(t27), & 0, & 0;
0, & 0, & 1, & 0;
\cos(t27), & -\sin(t27), & 0, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]

\[ \text{lan2Tlan3} = \begin{bmatrix}
\cos(t28), & -\sin(t28), & 0, & 0;
0, & 0, & 1, & 0;
-\sin(t28), & -\cos(t28), & 0, & 0;
0, & 0, & 0, & 1;
\end{bmatrix} \]
\[ p_{1Trhp1} = \begin{bmatrix} \cos(t29), \sin(t29), 0, 0; \\ \sin(t29), -\cos(t29), 0, 0; \\ 0, 0, -1, d8; \\ 0, 0, 0, 1 \end{bmatrix} \]

\[ r_{hp1Trhp2} = \begin{bmatrix} \sin(t30), -\cos(t30), 0, 0; \\ 0, 0, 0, 0; \\ \cos(t30), \sin(t30), 0, 1; \\ 0, 0, 0, 1 \end{bmatrix} \]

\[ r_{hp2Trhp3} = \begin{bmatrix} \cos(t31), -\sin(t31), 0, 0; \\ \sin(t31), \cos(t31), 0, 0; \\ 0, 0, 0, 0; \\ 0, 0, 0, 1 \end{bmatrix} \]

\[ r_{hp3Trk} = \begin{bmatrix} \cos(t32), -\sin(t32), 0, 0; \\ \sin(t32), \cos(t32), 0, 0; \\ 0, 0, 0, 0; \\ 0, 0, 0, 1 \end{bmatrix} \]

\[ r_{kTran1} = \begin{bmatrix} \cos(t33), -\sin(t33), 0, 0; \\ \sin(t33), \cos(t33), 0, 0; \\ 0, 0, 0, 0; \\ 0, 0, 0, 1 \end{bmatrix} \]

\[ r_{an1Tran2} = \begin{bmatrix} \sin(t34), \cos(t34), 0, 0; \\ \cos(t34), -\sin(t34), 0, 0; \\ 0, 0, 0, 0; \\ 0, 0, 0, 1 \end{bmatrix} \]

\[ r_{an2Tran3} = \begin{bmatrix} \cos(t35), -\sin(t35), 0, 0; \\ -\sin(t35), \cos(t35), 0, 0; \\ 0, 0, 0, 0; \\ 0, 0, 0, 1 \end{bmatrix} \]
APPENDIX C: CODE FOR THE RESOLVED RATE CONTROL OF
HUMANOID ARM

% Resolved rate control of the Humanoid arm

clear all
clc

AZ = -90-37.5+180;  EL = 30;  % 3D view angles
xV = [-.75 .75];                % View limits
yV = [-.75 .75];
zV = [-.75 .75];

%lengths in meters
l1=11*0.0254;
l2=10*0.0254;
LH=7*0.0254;

%Angles in radians
t1=0;
t2=90*pi/180;
t3=0;
t4=70*pi/180;
t5=0;
t6=-90*pi/180;
t7=0;

L=[0 0 0 11 12 0 LH];
R=0.0254*[0 0 0 1.4324 1.4324 0 1.4324];

%Initial angular velocity of the end effector
td=[0,0,0,0,0,0,0;0,0,0,0,0,0,0;0,0,0,0,0,0,0];

%Initial angular acceleration of the end effector
tdd=[0,0,0,0,0,0,0;0,0,0,0,0,0,0;0,0,0,0,0,0,0];

VH=[0.05;0.05;0.05];
wH=[0.05;0;0];
g=9.81;

wl=[0; 0; 0];
dw=[0; 0; 0];
dv=[0; g; 0];

%Initialising time and counter for animation
count=3;
time=0;


dt=0.1;

%initialising a counter for reading arrays which are later used for
%plotting graphs
num=1;

f=[0,0,0,0,0,0; 0,0,0,0,0,0; 0,0,0,0,0,0];

n1=f
n1loop=f
floop=f
w1loop=w1
dwloop=dw
dvloop=dv

%Loop for running animation and calculation of various variables for 5
%seconds
while time<=count
    [Torque,f,w1,dw,dv,n1]=dynamicsequations(t1,t2,t3,t4,t5,t6,t7,L,R,w1loop,dwloop,dvloop,td,tdd,floop,n1loop)

    floop=f
    w1loop=w1
dwloop=dw
dvloop=dv
    n1loop=n1

    [J,R03,R04,R07,T03,T04,T07,T0H]=jacob_arm(t1,t2,t3,t4,t5,t6,t7,11,12,LH)

    Vw=VH-cross(wH,R07*[LH;0;0])
    V=[VH;wH];

    tdot=pinv(J)*V;

    if time<=.75
        V=(time/.75)*V;
    else
        V=V;
    end
x1=[T03(1,4) T04(1,4)]
y1=[T03(2,4) T04(2,4)]
z1=[T03(3,4) T04(3,4)]

x2=[T04(1,4) T07(1,4)];
y2=[T04(2,4) T07(2,4)];
z2=[T04(3,4) T07(3,4)];

x3=[T07(1,4) T0H(1,4)];
y3=[T07(2,4) T0H(2,4)];
z3=[T07(3,4) T0H(3,4)];

xcoord1(num)=x1(1);
ycoord1(num)=y1(1);
zcoord1(num)=z1(1);

xcoord2(num)=x2(1);
ycoord2(num)=y2(1);
zcoord2(num)=z2(1);

xcoord3(num)=x3(1);
ycoord3(num)=y3(1);
zcoord3(num)=z3(1);

rad=.025
[xs1,ys1,zs1] = ballnsocketrad(x3(2),y3(2),z3(2),.025);

x32i=1.5264e-017
y32i=-0.4058
z32i=0.4271

x32f=0.144
y32f=-0.2643
z32f=.5415

sd=.05*side
sxi=[x32i+sd x32i+sd x32i-sd x32i-sd; x32i-sd x32i+sd x32i+sd x32i-sd];
syi=[y32i-sd y32i+sd y32i-sd y32i-sd; y32i+sd y32i+sd y32i-sd y32i-sd];
szi=[z32i-rad z32i-rad z32i-rad z32i-rad; z32i-rad z32i-rad z32i-rad z32i-rad];

sxf=[x32f+sd x32f+sd x32f-sd x32f-sd; x32f-sd x32f+sd x32f+sd x32f-sd];
syf=[y32f-sd y32f+sd y32f-sd y32f-sd; y32f+sd y32f+sd y32f+sd y32f-sd];
szf=[z32f-rad z32f-rad z32f-rad z32f-rad; z32f-rad z32f-rad z32f-rad z32f-rad];
%%ZYX EULER ANGLES ; Row Pitch Yaw

b1=atan2(R03(3,1),sqrt(R03(1,1)^2+R03(2,1)^2))
a1=atan2(R03(2,1),R03(1,1))
g1=atan2(R03(3,2),R03(3,3))

b2=atan2(R04(3,1),sqrt(R04(1,1)^2+R04(2,1)^2));
a2=atan2(R04(2,1),R04(1,1));
g2=atan2(R04(3,2),R04(3,3));

b3=atan2(R07(3,1),sqrt(R07(1,1)^2+R07(2,1)^2));
a3=atan2(R07(2,1),R07(1,1));
g3=atan2(R07(3,2),R07(3,3));

alpha1(num)=a1;
beta1(num)=b1;
gamma1(num)=g1;

alpha2(num)=a2;
beta2(num)=b2;
gamma2(num)=g2;

alpha3(num)=a3;
beta3(num)=b3;
gamma3(num)=g3;

subplot(2,2,1);
p=plot3(x1,y1,z1,'r',x2,y2,z2,'b',x3,y3,z3,'g',xs1,ys1,zs1,'y');
hold on;
surf(sxi,syi,szi);
surf(sxf,syf,szf);
hold off;

% Plot 3 main robot lengths (L1, L2, LH)
set(p,'LineWidth',3); grid;
set(gca,'FontSize',16); xlabel('\itX'); ylabel('\itY');
zlabel('\itZ');
axis([xV(1) xV(2) yV(1) yV(2) zV(1) zV(2)]);
axis('square');
%axis([xV(1) xV(2) yV(1) yV(2) zV(1) zV(2)]);
%title('3D View');
view(AZ,EL);

subplot(2,2,2);
p=plot3(x1,y1,z1,'r',x2,y2,z2,'b',x3,y3,z3,'g',xs1,ys1,zs1,'y');
hold on;
surf(sxi,siy,szi);
surf(sxf,syf,szf);
hold off;

% Plot 3 main robot lengths (L1, L2, LH)
set(p,'LineWidth',3); grid;
set(gca,'FontSize',16); xlabel(\itX'); ylabel(\itY');
zlabel(\itZ');
axis('square'); axis([xV(1) xV(2) yV(1) yV(2) zV(1) zV(2)]);
%title('XY Plane');
view(-90,90);

subplot(2,2,3);
p=plot3(x1,y1,z1,'r',x2,y2,z2,'b',x3,y3,z3,'g',xs1,ys1,zs1,'y');
hold on;
surf(sxi,siy,szi);
surf(sxf,yyf,szf);
hold off;

% Plot 3 main robot lengths (L1, L2, LH)
set(p,'LineWidth',3); grid;
set(gca,'FontSize',16); xlabel(\itX'); ylabel(\itY');
zlabel(\itZ');
axis('square'); axis([xV(1) xV(2) yV(1) yV(2) zV(1) zV(2)]);
%title('XZ Plane');
view(180,0);

subplot(2,2,4);
p=plot3(x1,y1,z1,'r',x2,y2,z2,'b',x3,y3,z3,'g',xs1,ys1,zs1,'y');
hold on;
surf(sxi,siy,szi);
surf(sxf,yyf,szf);
hold off;

% Plot 3 main robot lengths (L1, L2, LH)
set(p,'LineWidth',3); grid;
set(gca,'FontSize',16); xlabel(\itX'); ylabel(\itY');
zlabel(\itZ');
axis('square'); axis([xV(1) xV(2) yV(1) yV(2) zV(1) zV(2)]);
%title('YZ Plane');
view(-90,0);
pause(1/4);
if time==0
  pause;
end


ttime(num)=time
tl1(num)=t1;
ttdot1(num)=tdot(1,1);
tt2(num)=t2;
ttdot2(num)=tdot(2,1);
tt3(num)=t3;
ttdot3(num)=tdot(3,1);
tt4(num)=t4;
tt4(num) = tdot(4, 1);
tt5(num) = t5;

tt5(num) = tdot(5, 1);
tt6(num) = t6;

tt6(num) = tdot(6, 1);
tt7(num) = t7;

tt7(num) = tdot(7, 1);

t1 = t1 + tdot(1, 1) \times 0.1;
t2 = t2 + tdot(2, 1) \times 0.1;
t3 = t3 + tdot(3, 1) \times 0.1;
t4 = t4 + tdot(4, 1) \times 0.1;
t5 = t5 + tdot(5, 1) \times 0.1;
t6 = t6 + tdot(6, 1) \times 0.1;
t7 = t7 + tdot(7, 1) \times 0.1;

detj(num) = det(J \times J');

% Incrementing the counter

time = time + dt;

td = (tdot \times [0 0 1])';

Torq1(num) = Torque(1);
Torq2(num) = Torque(2);
Torq3(num) = Torque(3);
Torq4(num) = Torque(4);
Torq5(num) = Torque(5);
Torq6(num) = Torque(6);
Torq7(num) = Torque(7);

Force = f' \times [0 0 1]';

f1(num) = Force(1);
f2(num) = Force(2);
f3(num) = Force(3);
f4(num) = Force(4);
f5(num) = Force(5);
f6(num) = Force(6);
f7(num) = Force(7);

v1(num) = V(1);
v2(num) = V(2);
v3(num) = V(3);
v4(num) = V(4);
v5(num) = V(5);
v6(num) = V(6);

num = num + 1;
td1\( (num) = td(3,1); \)
td2\( (num) = td(3,2); \)
td3\( (num) = td(3,3); \)
td4\( (num) = td(3,4); \)
td5\( (num) = td(3,5); \)
td6\( (num) = td(3,6); \)
td7\( (num) = td(3,7); \)

tddl\( (num) = [td1\( (num) - td1\( (num-1) ))/dt; \)
tdd2\( (num) = [td2\( (num) - td2\( (num-1) ))/dt; \)
tdd3\( (num) = [td3\( (num) - td3\( (num-1) ))/dt; \)
tdd4\( (num) = [td4\( (num) - td4\( (num-1) ))/dt; \)
tdd5\( (num) = [td5\( (num) - td5\( (num-1) ))/dt; \)
tdd6\( (num) = [td6\( (num) - td6\( (num-1) ))/dt; \)
tdd7\( (num) = [td7\( (num) - td7\( (num-1) ))/dt; \)

tdd = ([td1\( (num) \quad tdd2\( (num) \quad tdd3\( (num) \quad tdd4\( (num) \quad tdd5\( (num) \quad tdd6\( (num) \quad tdd7\( (num) ]) \) '* [0 0 1])'); \)

end

figure;
subplot(4,2,1);
plot(ttime,tt1);
ylabel('Theta 1');
subplot(4,2,3);
plot(ttime,tt2);
ylabel('Theta 2');
subplot(4,2,5);
plot(ttime,tt3);
ylabel('Theta 3');
subplot(4,2,7);
plot(ttime,tt4);
xlabel('Time')
ylabel('Theta 4');
subplot(4,2,2);
plot(ttime,tt5);
ylabel('Theta 5');
subplot(4,2,4);
plot(ttime,tt6);
ylabel('Theta 6');
subplot(4,2,6);
plot(ttime,tt7);
xlabel('Time')
ylabel('Theta 7');
figure;
subplot(4,2,1);
plot(ttime,ttdot1);
ylabel('Theta dot 1');
subplot(4,2,3);
plot(ttime,ttdot2);
ylabel('Theta dot 2');
subplot(4,2,5);
plot(ttime,ttdot3);
ylabel('Theta dot 3');
subplot(4,2,7);
plot(ttime,ttdot4);
xlabel('Time')
ylabel('Theta dot 4');
subplot(4,2,2);
plot(ttime,ttdot5);
ylabel('Theta dot 5');
subplot(4,2,4);
plot(ttime,ttdot6);
ylabel('Theta dot 6');
subplot(4,2,6);
plot(ttime,ttdot7);
xlabel('Time')
ylabel('Theta dot 7');

figure;
plot(ttime,detj);
xlabel('Time')
ylabel('det (J*Jt)');

figure;
subplot(3,2,1);
plot(ttime,xcoord3);
ylabel('x3');
subplot(3,2,3);
plot(ttime,ycoord3);
ylabel('y3');
subplot(3,2,5);
plot(ttime,zcoord3);
ylabel('z3');
xlabel('Time');
subplot(3,2,2);
plot(ttime,alpha3);
ylabel('alpha3');
subplot(3,2,4);
plot(ttime,beta3);
ylabel('beta3');
subplot(3,2,6);
plot(ttime,gamma3);
ylabel('gamma3');
xlabel('Time');
figure;
subplot(4,2,1);
plot(ttime,f1);
ylabel('Force 1');
subplot(4,2,3);
plot(ttime,f2);
ylabel('Force 2');
subplot(4,2,5);
plot(ttime,f3);
ylabel('Force 3');
subplot(4,2,7);
plot(ttime,f4);
xlabel('Time')
ylabel('Force 4');
subplot(4,2,2);
plot(ttime,f5);
ylabel('Force 5');
subplot(4,2,4);
plot(ttime,f6);
ylabel('Force 6');
subplot(4,2,6);
plot(ttime,f7);
xlabel('Time')
ylabel('Force 7');

figure;
subplot(4,2,1);
plot(ttime,Torq1);
ylabel('Torque 1');
subplot(4,2,3);
plot(ttime,Torq2);
ylabel('Torque 2');
subplot(4,2,5);
plot(ttime,Torq3);
ylabel('Torque 3');
subplot(4,2,7);
plot(ttime,Torq4);
xlabel('Time')
ylabel('Torque 4');
subplot(4,2,2);
plot(ttime,Torq5);
ylabel('Torque 5');
subplot(4,2,4);
plot(ttime,Torq6);
ylabel('Torque 6');
subplot(4,2,6);
plot(ttime,Torq7);
xlabel('Time')
ylabel('Torque 7');
APPENDIX D: CODE FOR THE SIMULATION OF HUMANOID FOR WALKING

clear all
clc

%Lengths and radii of various links of the Humanoid in meters
lt1=10*0.0254;
lt2=12*0.0254;
ls=0;
lla1=11*0.0254;
lla2=10*0.0254;
lra1=lla1;
lra2=lla2;
wais=0;
lll1=18*0.0254;
lll2=19*0.0254;
lrl1=lll1;
lrl2=lll2;
d81=30*0.0254/(2*pi)+11*0.0254/(2*pi);
d82=30*0.0254/(2*pi)-13*0.0254/(2*pi);
lh=7*0.0254;
lf=8.5*0.0254;
radhead=22*0.0254/(2*pi);
neck=3*0.0254;
r=[.5*0.0254 .5*0.0254]';
rth=[13*0.0254/(2*pi) 13*0.0254/(2*pi)]';
rwst=[30*0.0254/(2*pi) 30*0.0254/(2*pi)]';
rhnd=[9*0.0254/(2*pi) 9*0.0254/(2*pi)]';
rad=1.5*0.0254
floor=lll1+lll2+rad;

%Initialising
increment=0;
num=1;
time=0;

%Input Angles; conversion of angles in degrees into radians
t1=0*pi/180;
t2=0*pi/180;
t3=0*pi/180;
t4=0*pi/180;
t5=0*pi/180;
t6=0*pi/180;
t7=0*pi/180;
t8=0*pi/180;
t9=0*pi/180;
t10=0*pi/180;
t11=0*pi/180;
t12=0*pi/180;
t13=0*pi/180+pi/2;
t14=0*pi/180;

 t15=0*pi/180;
t16=0*pi/180;
t17=0*pi/180
 t18=0*pi/180;
t19=0*pi/180;
t20=0*pi/180+pi/2;
t21=0*pi/180;

 t22=0*pi/180;
t23=0*pi/180;
t24=0*pi/180;
t25=0*pi/180;
t26=0*pi/180;
t27=0*pi/180;
t28=0*pi/180;

 t29=0*pi/180;
t30=0*pi/180;
t31=0*pi/180;
t32=0*pi/180;
t33=0*pi/180;
t34=0*pi/180;
t35=0*pi/180;

step=.1;
fast=step/.01;
n=20
option=1;
j=1

[q1,q2,q3,q4,ank,shldr]=legangledata(option);

for t=0:step:2.4
    t11=-ank(j);
    t18=-ank(j);
    t8=-shldr(j);
    t15=shldr(j);
    t24=q2(j);
    t25=q1(j);
    t31=-q3(j);
    t32=q4(j);
    j=j+1;
end

%Transformations
if t>0 & t<=0.4
    increment=increment+([sin(q2(j))-sin(q2(j-1))]*(lll1+lll2));
else
    increment=increment+[(lll1+lll2)*sin(0.2750)+(lll1)*sin(0.0962)]*step/4;
end

m=length(r);
theta = (0:20)/20*2*pi;
theta2 = (0:4)/4*2*pi;
mat=[ones(1,n+1);ones(1,n+1)];

%Joint Coordinates of the cylinder 1
[xc,yc,zc]=cylcoords(m,n,theta,-2*d82,r,3);
[xcf1,ycf1,zcf1]=finalcoords(xc,yc,zc,oTlhp1,n);
[xs1,ys1,zs1] = ballnsocketrad(xc,yc,zc,oTlhp1,n);

%Joint Coordinates of the cylinder 2
[xc,yc,zc]=cylcoords(m,n,theta,lt1,rwst,1);
[xcf2,ycf2,zcf2]=finalcoords(xc,yc,zc,oTp3,n);
[xs2,ys2,zs2] = ballnsocketrad(xc,yc,zc,oTp3,n);

%Joint Coordinates of the cylinder 3
[xc,yc,zc]=cylcoords(m,n,theta,lla1,rhnd,1);
[xcf3,ycf3,zcf3]=finalcoords(xc,yc,zc,oTls1,n);
[xs3,ys3,zs3] = ballnsocketrad(xc,yc,zc,oTls1,n);

%Joint Coordinates of the cylinder 4
[xc,yc,zc]=cylcoords(m,n,theta,-2*d81,r,3);
[xcf4,ycf4,zcf4]=finalcoords(xc,yc,zc,oTls3,n);
[xs4,ys4,zs4] = ballnsocketrad(xc,yc,zc,oTls3,n);

%Joint Coordinates of the cylinder 5
[xc,yc,zc]=cylcoords(m,n,theta,lla1,rhnd,1);
[xcf5,ycf5,zcf5]=finalcoords(xc,yc,zc,oTls3,n);
[xs5,ys5,zs5] = ballnsocketrad(xc,yc,zc,oTls3,n);
%Joint Coordinates of the cylinder 6
[xc, yc, zc] = cylcoords(m, n, theta, lla2, rhnd, 1);
[xcf6, ycf6, zcf6] = finalcoords(xc, yc, zc, oTle, n);
[xs6, ys6, zs6] = ballnsocketrad(x6, y6, z6, rad);

%Joint Coordinates of the cylinder 7
[xc, yc, zc] = cylcoords(m, n, theta, lra1, rhnd, 1);
[xcf7, ycf7, zcf7] = finalcoords(xc, yc, zc, oTrs3, n);
[xs7, ys7, zs7] = ballnsocketrad(x7, y7, z7, rad);

%Joint Coordinates of the cylinder 8
[xc, yc, zc] = cylcoords(m, n, theta, lra2, rhnd, 1);
[xcf8, ycf8, zcf8] = finalcoords(xc, yc, zc, oTre, n);
[xs8, ys8, zs8] = ballnsocketrad(x8, y8, z8, rad);

%Joint Coordinates of the cylinder 9
[xc, yc, zc] = cylcoords(m, n, theta, lll1, rth, 1);
[xcf9, ycf9, zcf9] = finalcoords(xc, yc, zc, oTlh3, n);
[xs9, ys9, zs9] = ballnsocketrad(x9, y9, z9, rad);

%Joint Coordinates of the cylinder 10
[xc, yc, zc] = cylcoords(m, n, theta, lll2, rth, 1);
[xcf10, ycf10, zcf10] = finalcoords(xc, yc, zc, oTlk, n);
[xs10, ys10, zs10] = ballnsocketrad(x10, y10, z10, rad);

%Joint Coordinates of the cylinder 11
[xc, yc, zc] = cylcoords(m, n, theta, lrl1, rth, 1);
[xcf11, ycf11, zcf11] = finalcoords(xc, yc, zc, oTrhp3, n);
[xs11, ys11, zs11] = ballnsocketrad(x11, y11, z11, rad);

%Joint Coordinates of the cylinder 12
[xc, yc, zc] = cylcoords(m, n, theta, lrl2, rth, 1);
[xcf12, ycf12, zcf12] = finalcoords(xc, yc, zc, oTrk, n);
[xs12, ys12, zs12] = ballnsocketrad(x12, y12, z12, rad);

%Joint Coordinates of the cylinder lh
[xc, yc, zc] = cylcoords(m, n, theta2, lh, r, 3);
[xcf1h, ycf1h, zcf1h] = finalcoords(xc, yc, zc, oTlw3, 4);
[xs1h, ys1h, zs1h] = ballnsocketrad(x1h, y1h, z1h, rad);

%Joint Coordinates of the cylinder rh
[xc, yc, zc] = cylcoords(m, n, theta2, lh, r, 3);
[xcfrh, ycfrh, zcfrh] = finalcoords(xc, yc, zc, oTrw3, 4);
[xsrrh, ysrh, zsrrh] = ballnsocketrad(xrh, yrh, zrh, rad);

%Joint Coordinates of the cylinder lf
[xc, yc, zc] = cylcoords(m, n, theta2, lf, r, 2);
[xcf1f, ycf1f, zcf1f] = finalcoords(xc, yc, zc, oTlan3, 4);
[xs1f, yslf, zslf] = ballnsocketrad(xlf, ylf, zlf, rad);
%Joint Coordinates of the cylinder rf
[xc,yc,zc]=cylcoords(m,n,theta2,lf,r,2);
[xcfrf,ycrf,zcfrf]=finalcoords(xc,yc,zc,oTran3,4);
[xsrf,ysrf,zsrf] = ballnsocketrad(xrf,yrf,zrf,rad);

%Joint Coordinates of the cylinder for neck
[xcn,ycn,zcn]=cylcoords(m,n,theta,neck,r,1);
[xcnf3,ycnf3,zcnf3]=finalcoords(xcn,ycn,zcn,oTtneck,n);

[xsheadf,ysheadf,zsheadf] = ballnsocketrad(xhead,yhead,zhead,radhead);

lim=1.3;
clr=summer;

subplot(2,2,1);
sx=[-lim lim lim -lim;lim lim -lim -lim]
sy=[-lim -lim lim lim;lim -lim -lim lim]
sz=[-floor -floor -floor -floor;-floor -floor -floor -floor];
surf(sx,sy,sz);
colormap(clr);
hold on;
plot3(x1,y1,z1,x2,y2,z2,x3,y3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7,x8 ,y8,z8,x9,y9,z9,x10,y10,z10,x11,y11,z11,x12,y12,z12);
hold on;
plot3(xlh,ylh,zlh,'r',xrh,yrh,zrh,'r',xlf,ylf,zlf,'r',xrf,yrf,zrf,'r')
hold off;
xlabel('X');
ylabel('Y');
zlabel('Z');
axis('square');
axis([-lim lim -lim lim -lim lim]);
view(37.3,30)

hold on;
surf(xcf1,ycf1,zcf1);
hold on;
surf(xs1,ys1,zs1);
hold on;
surf(xcf2,ycf2,zcf2);
hold on;
surf(xcf3,ycf3,zcf3);
hold on;
surf(xs3,ys3,zs3);
hold on;
surf(xcf4,ycf4,zcf4);
hold on;
surf(xs4,ys4,zs4);
```matlab
hold on;
surf(xcf5,ycf5,zcf5);
hold on;
surf(xs5,ys5,zs5);
hold on;
surf(xcf6,ycf6,zcf6);
hold on;
surf(xcf7,ycf7,zcf7);
hold on;
surf(xs7,ys7,zs7);
hold on;
surf(xcf8,ycf8,zcf8);
hold on;
surf(xcf9,ycf9,zcf9);
hold on;
surf(xs9,ys9,zs9);
hold on;
surf(xcf10,ycf10,zcf10);
hold on;
surf(xcf11,ycf11,zcf11);
hold on;
surf(xs11,ys11,zs11);
hold on;
surf(xcf12,ycf12,zcf12);
hold on;
surf(xs1h,ys1h,zs1h);
hold on;
surf(xs1r,ys1r,zs1r);
hold on;
surf(xs1f,ys1f,zs1f);
hold on;
surf(xsrf,ysrf,zsrf);
hold on;
surf(xcf1h,ycf1h,zcf1h);
hold on;
surf(xcf1r,ycf1r,zcf1r);
hold on;
surf(xcf1f,ycf1f,zcf1f);
hold on;
surf(xcf1rf,ycf1rf,zcf1rf);
hold on;
surf(xsheadf,ysheadf,zsheadf);
hold on;
surf(xcnf3,ycnf3,zcnf3);

hold off;

subplot(2,2,2);
sx=[-lim lim lim -lim;lim lim -lim -lim];
sy=[-lim -lim lim -lim];
sz=[-floor -floor -floor -floor;floor -floor -floor floor];
surf(sx,sy,sz);
colormap(clr);
```
hold on;

plot3(x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7,x8
,y8,z8,x9,y9,z9,x10,y10,z10,x11,y11,z11,x12,y12,z12);
hold on;
plot3(xlh,ylh,zlh,'r',xrh,yrh,zrh,'r',xlf,ylf,zlf,'r',xrf,yrf,zrf,'r')
hold off;
xlabel('X');
ylabel('Y');
zlabel('Z');
axis('square');
axis([-lim lim -lim lim -lim lim]);
view(0,0);

hold on;
surf(xcf1,ycf1,zcf1);
hold on;
surf(xsl1,ys1,zs1);
hold on;
surf(xcf2,ycf2,zcf2);
hold on;
%surf(xs2,ys2,zs2);
hold on;
surf(xcf3,ycf3,zcf3);
hold on;
surf(xs3,ys3,zs3);
hold on;
surf(xcf4,ycf4,zcf4);
hold on;
surf(xs4,ys4,zs4);
hold on;
surf(xcf5,ycf5,zcf5);
hold on;
surf(xs5,ys5,zs5);
hold on;
surf(xcf6,ycf6,zcf6);
hold on;
surf(xcf7,ycf7,zcf7);
hold on;
surf(xs7,ys7,zs7);
hold on;
surf(xcf8,ycf8,zcf8);
hold on;
surf(xcf9,ycf9,zcf9);
hold on;
surf(xs9,ys9,zs9);
hold on;
surf(xcf10,ycf10,zcf10);
hold on;
surf(xcf11,ycf11,zcf11);
hold on;
surf(xs11,ys11,zs11);
hold on;
surf(xcf12, ycf12, zcf12);
hold on;
surf(xslh, yslh, zslh);
hold on;
surf(xsrh, ysrh, zsrh);
hold on;
surf(xs1f, yslf, zslf);
hold on;
surf(xsr1f, ysr1f, zsr1f);
hold on;
surf(xcflh, ycflh, zcflh);
hold on;
surf(xcfrh, ycfrh, zcfrh);
hold on;
surf(xcflf, ycflf, zcflf);
hold on;
surf(xcfrf, ycfrf, zcfrf);

hold on;
surf(xsheadf, ysheadf, zsheadf);

hold on;
surf(xcnf3, ycnf3, zcnf3);
hold off;

subplot(2,2,3);

sx=[-lim lim lim -lim;lim lim -lim -lim];
sy=[-lim -lim lim lim;lim lim -lim -lim];
sz=[-floor -floor -floor -floor; -floor -floor -floor -floor];
surf(sx, sy, sz);
colormap(clr);
hold on;
plot3(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4, x5, y5, z5, x6, y6, z6, x7, y7, z7, x8,
y8, z8, x9, y9, z9, x10, y10, z10, x11, y11, z11, x12, y12, z12);
hold on;
plot3(xlh, ylh, zlh, 'r', xrh, yrh, zrh, 'r', xsf, ysf, zsf, 'r', xrf, yrf, zrf, 'r')
hold off;

xlabel('X');
ylabel('Y');
zlabel('Z');
axis('square');
axis([-lim lim -lim lim -lim lim -lim lim]);
view(90,0);

hold on;
surf(xcf1, ycf1, zcf1);
hold on;
surf(xs1, ysl, zsl);
hold on;
surf(xcf2, ycf2, zcf2);
hold on;  
%surf(xs2,ys2,zs2);  
hold on;  
surf(xcf3,ycf3,zcf3);  
hold on;  
surf(xs3,ys3,zs3);  
hold on;  
surf(xcf4,ycf4,zcf4);  
hold on;  
surf(xs4,ys4,zs4);  
hold on;  
surf(xcf5,ycf5,zcf5);  
hold on;  
surf(xs5,ys5,zs5);  
hold on;  
surf(xcf6,ycf6,zcf6);  
hold on;  
surf(xcf7,ycf7,zcf7);  
hold on;  
surf(xs7,ys7,zs7);  
hold on;  
surf(xcf8,ycf8,zcf8);  
hold on;  
surf(xcf9,ycf9,zcf9);  
hold on;  
surf(xs9,ys9,zs9);  
hold on;  
surf(xcf10,ycf10,zcf10);  
hold on;  
surf(xcf11,ycf11,zcf11);  
hold on;  
surf(xs11,ys11,zs11);  
hold on;  
surf(xcf12,ycf12,zcf12);  
hold on;  
surf(xs1h,ys1h,zs1h);  
hold on;  
surf(xsrh,ysrh,zsrh);  
hold on;  
surf(xs1f,ys1f,zs1f);  
hold on;  
surf(xsrh,yrsr,zrsr);  
hold on;  
surf(xsrf,ysrf,zsrh);  
hold on;  
surf(xcf1h,ycf1h,zcf1h);  
hold on;  
surf(xcf1h,ycf1h,zcf1h);  
hold on;  
surf(xsrh,yrsr,zsrh);  
hold on;  
surf(xcf1f,ycf1f,zcf1f);  
hold on;  
surf(xcfrh,ycfrh,zcfrh);  
hold on;  
surf(xcfrf,ycfrf,zcfrf);  
hold on;  
surf(xsheadf,ysheadf,zsheadf);
hold on;
surf(xcnf3,ycnf3,zcnf3);
hold off;

subplot(2,2,4);

sx=[-lim lim lim -lim;lim lim -lim lim];
sy=[-floor -floor -floor -floor;floor -floor -floor -floor];
sz=[-floor -floor -floor -floor;floor -floor -floor -floor];
surf(sx,sy,sz);
colormap(clr);
hold on;

plot3(x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7,x8
 ,y8,z8,x9,y9,z9,x10,y10,z10,x11,y11,z11,x12,y12,z12);
hold on;
plot3(xlh,ylh,zlh,'r',xrh,yrh,zrh,'r',xlf,ylf,zlf,'r',xrf,yrf,zrf,'r')
hold off;
xlabel('X');
ylabel('Y');
zlabel('Z');
axis('square');
axis([-lim lim -lim lim -lim lim -lim lim]);
view(90,90);
hold on;

hold on;
surf(xcf1,ycf1,zcf1);
hold on;
surf(xs1,ys1,zs1);
hold on;
surf(xcf2,ycf2,zcf2);
hold on;
surf(xcf3,ycf3,zcf3);
hold on;
surf(xs3,ys3,zs3);
hold on;
surf(xcf4,ycf4,zcf4);
hold on;
surf(xs4,ys4,zs4);
hold on;
surf(xcf5,ycf5,zcf5);
hold on;
surf(xs5,ys5,zs5);
hold on;
surf(xcf6,ycf6,zcf6);
hold on;
surf(xcf7,ycf7,zcf7);
hold on;
surf(xs7,ys7,zs7);
hold on;
surf(xcf8,ycf8,zcf8);
hold on;
surf(xcf9,ycf9,zcf9);
hold on;
surf(xs9,ys9,zs9);
hold on;
surf(xcf10,ycf10,zcf10);
hold on;
surf(xcf11,ycf11,zcf11);
hold on;
surf(xs11,ys11,zs11);
hold on;
surf(xcf12,ycf12,zcf12);
hold on;
surf(xslh,yslh,zslh);
hold on;
surf(xsrh,ysrh,zsrh);
hold on;
surf(xslf,yslf,zslf);
hold on;
surf(xsrh,ysrh,zsrh);
hold on;
surf(xsrf,ysrf,zsrf);
hold on;
surf(xcflh,ycflh,zcflh);
hold on;
surf(xcfrh,ycfrh,zcfrh);
hold on;
surf(xcflf,ycflf,zcflf);
hold on;
surf(xcfrf,ycfrf,zcfrf);
hold on;
surf(xsheadf,ysheadf,zsheadf);
hold on;
surf(xcnf3,ycnf3,zcnf3);
hold off;

pause(0.1);
if j==2
    pause;
end
t33;
t34;
t35;

ttime(num)=time;
tt8(num)=t8;
tt11(num)=t11;
tt15(num)=t15;
tt18(num)=t18;
tt24(num)=t24;
tt25(num)=t25;
tt31(num)=t31;
tt32(num)=t32;

time=time+step;
num=num+1;

end

timer=2.4

figure;
subplot(4,2,1);
plot(ttime,tt8);
ylabel('Theta 8');
axis([0 timer -0.5 1]);
subplot(4,2,3);
plot(ttime,tt11);
ylabel('Theta 11');
axis([0 timer -0.5 1]);
subplot(4,2,5);
plot(ttime,tt15);
ylabel('Theta 15');
axis([0 timer -0.5 1]);
subplot(4,2,7);
plot(ttime,tt18);
xlabel('Time')
ylabel('Theta 18');
axis([0 timer -0.5 1]);
subplot(4,2,2);
plot(ttime,tt24);
ylabel('Theta 24');
axis([0 timer -0.5 1]);
subplot(4,2,4);
plot(ttime,tt25);
ylabel('Theta 25');
axis([0 timer -0.5 1]);
subplot(4,2,6);
plot(ttime,tt31);
xlabel('Time')
ylabel('Theta 31');
axis([0 timer -0.5 1]);
subplot(4,2,8);
plot(ttime,tt32);
xlabel('Time')
ylabel('Theta 32');
axis([0 timer -0.5 1]);
APPENDIX E: M-FILE OF THE FUNCTION LEGANGLEDATA

function [qone, qtwo, qthr, qfour, ank, shldr] = legangledatanodwellslower(step);

i=1;
times=2

for t=0:step:0.2*times
    q1ini=0;
    time1(i)=t;
    q1in(i)=q1ini;
    i=i+1;
end

i=1;
for t=0:step:0.2*times
    m=.265/(.2*times);
    q2ini=m*t;
    time1(i)=t;
    q2in(i)=q2ini;
    i=i+1;
end

i=1;
for t=0:step:0.2*times
    m=.385/(.2*times);
    q3ini=m*t;
    time1(i)=t;
    q3in(i)=q3ini;
    i=i+1;
end

i=1;
for t=0:step:0.2*times
    m=.385/(.2*times);
    q4ini=m*t;
    time1(i)=t;
    q4in(i)=q4ini;
    i=i+1;
end

i=1;
for t=0:step:0.2*times
    an1=2*0.875*t/times;
    time1(i)=t;
    ank1(i)=an1;
    i=i+1;
end
i=1;
for t=0:step:0.2*times
    shldr=0.875*t/times;
    time1(i)=t;
    shldr1(i)=shldr;
    i=i+1;
end

% i=1;
for t=0.2*times:step:0.25*times
    m=.708/((.25-.2)*times);
    q1sla=m*(t-.2*times);
    time21(i)=t;
    q1s11(i)=q1sla;
    i=i+1;
end

i=1;
for t=0.25*times+step:step:0.35*times
    q1sla=0.708;
    time22(i)=t;
    q1s12(i)=q1sla;
    i=i+1;
end

i=1;
for t=0.35*times+step:step:0.5*times
    m=0.708/((.35-.5)*times);
    q1sla=m*(t-.5*times);
    time23(i)=t;
    q1s13(i)=q1sla;
    i=i+1;
end

i=1;
for t=0.5*times+step:step:.7*times
    q1sla=0;
    time24(i)=t;
    q1s14(i)=q1sla;
    i=i+1;
end

timeq1=[time21 time22 time23 time24]
q1s1=[q1s11 q1s12 q1s13 q1s14]

% i=1;
for t=0.2*times:step:0.4*times
    m=(.265+.385)/((.2-.4)*times);

q2s1a = m*(t-.2*times)+0.265;
    time21(i)=t;
    q2s11(i)=q2s1a;
    i=i+1;
end

i=1;
for t=0.4*times+step:step:0.45*times
    q2s1a=-.385;
    time22(i)=t;
    q2s12(i)=q2s1a;
    i=i+1;
end

i=1;
for t=0.45*times+step:step:0.7*times
    m=(.265+.385)/((.7-.45)*times)
    q2s1a=m*(t-.7*times)+.265;
    time23(i)=t;
    q2s13(i)=q2s1a;
    i=i+1;
end

timeq2=[time21 time22 time23];
q2s1=[q2s11 q2s12 q2s13]

% i=1;
t=0.2;
for t=0.2*times:step:0.45*times
    m=(.385+.265)/(0.7-.45)*times)
    q3s1a=m*(t-.2*times)+.385;
    time21(i)=t;
    q3s11(i)=q3s1a;
    i=i+1;
end

i=1;
for t=0.45*times+step:step:0.7*times
    m=(-.265-.385)/((.45-.7)*times)
    q3s1a=m*(t-.45*times)-.265;
    time22(i)=t;
    q3s12(i)=q3s1a;
    i=i+1;
end
timeq3=[time21 time22];
q3s1=[q3s11 q3s12];

i=1;
for t=0.2*times:step:0.25*times
m = .385/((.2-.25)*times);
q4sla=m*(t-.25*times);
timeq41(i)=t;
q4sl1(i)=q4sla;
i=i+1;
end
i=1;
for t=0.25*times+step:step:0.45*times
q4sla=0;
timeq42(i)=t;
q4sl2(i)=q4sla;
i=i+1;
end
i=1;
for t=0.45*times+step:step:0.5*times
m = .708/((.5-.45)*times);
q4sla=m*(t-.45*times);
timeq43(i)=t;
q4sl3(i)=q4sla;
i=i+1;
end
i=1;
for t=0.5*times+step:step:0.6*times
q4sla=0.708;
timeq44(i)=t;
q4sl4(i)=q4sla;
i=i+1;
end
i=1;
for t=(0.6*times+step):step:0.7*times
m = (.708-.385)/((.6-.7)*times);
q4sla=m*(t-.6*times)+.708;
timeq45(i)=t;
q4sl5(i)=q4sla;
i=i+1;
end
timeq4=[timeq41 timeq42 timeq43 timeq44 timeq45];
q4s1=[q4sl1 q4sl2 q4sl3 q4sl4 q4sl5];

i=1;
for t=0.2*times:step:0.7*times
an2=2*0.175;
timeq5(i)=t;
ank2(i)=an2;
i=i+1;
end
i=1;
t=0.2;
for t=0.2*times:step:0.45*times
shld2=-1.75/times*t+.525;
timesd1(i)=t;
shldra(i)=shld2;
i=i+1;
end

i=1;
for t=0.45*times+step:step:0.7*times
  shld2=1.75/times*t-.875;
timesd2(i)=t;
  shldrb(i)=shld2;
i=i+1;
end

timeq6=[timesd1 timesd2];
shldr2=[shldra shldrb];
%
timemore=timeq1+.5*times
time=[time1 timeq1 timemore];

qone=[q1in q1s1 q1s1]
qtwo=[q2in q2s1 q2s1]
qthr=[q3in q3s1 q3s1]
qfour=[q4in q4s1 q4s1]
anl=[ank1 ank2 ank2]
shldr=[shldr1 shldr2 shldr2]
function [qone, qtwo, qthr, qfour, ank, shldr] = legangledataclimbing_slower(step);

i = 1;
times = 2;
step = 0.1

for t = 0:step:0.25*times
    qlini = 0;
    time1(i) = t;
    qlin(i) = qlini;
    i = i + 1;
end

i = 1;
for t = (0.25*times + step):step:0.3*times
    m = (pi/2)/(.55*times -.5*times);
    q1s1a = (t -.25*times)*m;
    time11(i) = t;
    q1s11(i) = q1s1a;
    i = i + 1;
end

i = 1;
for t = 0.3*times + step:step:0.45*times
    q1s1a = pi/2;
    time12(i) = t;
    q1s12(i) = q1s1a;
    i = i + 1;
end

i = 1;
for t = 0.45*times + step:step:0.5*times
    m = (.85615 - (pi/2))/[(.75 -.7)*times];
    q1s1a = (t -.45*times)*m + (pi/2);
    time13(i) = t;
    q1s13(i) = q1s1a;
    i = i + 1;
end

i = 1;
for t = .5*times + step:step:.55*times
    m = -.85615/[(1.05 - 1)*times];
    q1s1a = (t -.55*times)*m;
    time15(i) = t;
    q1s15(i) = q1s1a;
    i = i + 1;
end
\[i=1;\]
\[
\text{for } t=0.25\times \text{times} \text{ step: step: } 0.25\times \text{times}
\]
\[
\quad m=0.1136/(0.25\times \text{times});
\quad q2ini=m\times t;
\quad time2(i)=t;
\quad q2in(i)=q2ini;
\quad i=i+1;
\]
\[\text{end}\]
\[
\text{for } t=0.45\times \text{times} \text{ step: step: } 0.45\times \text{times}
\]
\[
\quad m=(-0.85615-0.1136)/(0.7\times \text{times}-0.5\times \text{times});
\quad q2s1a=m\times (t-0.45\times \text{times})-0.85615;
\quad time22(i)=t;
\quad q2s12(i)=q2s1a;
\quad i=i+1;
\]
\[\text{end}\]
\[
\text{for } t=0.5\times \text{times} \text{ step: step: } 0.5\times \text{times}
\]
\[
\quad q2s1a=-0.85615;
\quad time23(i)=t;
\quad q2s13(i)=q2s1a;
\quad i=i+1;
\]
\[\text{end}\]
\[
\text{for } t=0.75\times \text{times} \text{ step: step: } 0.75\times \text{times}
\]
\[
\quad m=(-0.85615-0.1136)/(1\times \text{times}-1.25\times \text{times});
\quad q2s1a=m\times (t-0.75\times \text{times})+0.1136;
\quad time24(i)=t;
\quad q2s14(i)=q2s1a;
\quad i=i+1;
\]
\[\text{end}\]

\[\text{timeq2}=[\text{time2 time22 time23 time24}];\]
\[\text{q2s1}=[\text{q2in q2s12 q2s13 q2s14}];\]
\% i=1;
for t=0:step:0.2*times
    m=.85615/(.2*times);
    q3ini=m*t;
    time3(i)=t;
    q3in(i)=q3ini;
    i=i+1;
end

i=1;
for t=.2*times+step:step:0.25*times
    q3sla=.85615;
    time31(i)=t;
    q3s31(i)=q3sla;
    i=i+1;
end

i=1;
for t=0.25*times+step:step:0.5*times
    m=(-.1136-.85615)/(.5*times-.25*times);
    q3sla=m*(t-.25*times)+.85615;
    time32(i)=t;
    q3s32(i)=q3sla;
    i=i+1;
end

i=1;
for t=.5*times+step:step:.7*times
    m=(+.1136+.85615)/(1.2*times-1*times);
    q3sla=m*(t-.5*times)-.1136;
    time34(i)=t;
    q3s34(i)=q3sla;
    i=i+1;
end

i=1;
for t=.7*times+step:step:.75*times
    q3sla=.85615;
    time35(i)=t;
    q3s35(i)=q3sla;
    i=i+1;
end
timeq3=[time3 time31 time32 time34 time35];
q3s1=[q3in q3s31 q3s32 q3s34 q3s35];

\% i=1;
for t=0:step:0.05*times
    m=(pi/2)/(.05*times);
    q4ini=m*t;
    time4(i)=t;
    q4in(i)=q4ini;
% i=1;
for t=0.05*times+step:step:0.2*times
    q4sla=pi/2;
    timeq41(i)=t;
    q4s11(i)=q4sla;
    i=i+1;
end
i=1;
for t=0.2*times+step:step:0.3*times
    m=(-pi/2)/(.3*times-.2*times);
    q4sla=m*(t-.3*times);
    timeq42(i)=t;
    q4s12(i)=q4sla;
    i=i+1;
end
% i=1;
for t=0.3*times+step:step:.5*times
    q4sla=0;
    timeq45(i)=t;
    q4s15(i)=q4sla;
    i=i+1;
end
i=1;
for t=.5*times+step:step:.55*times
    m=(pi/2)/(.55*times-.5*times);
    q4sla=m*(t-.5*times);
    timeq46(i)=t;
    q4s16(i)=q4sla;
    i=i+1;
end
i=1;
for t=.55*times+step:step:0.7*times
    q4sla=pi/2;
    timeq47(i)=t;
    q4s17(i)=q4sla;
    i=i+1;
end
i=1;
for t=.7*times+step:step:0.75*times
    m=(pi/2-.85615)/(1.2*times-1.25*times);
    q4sla=m*(t-.75*times)+.85615;
    timeq48(i)=t;
    q4s18(i)=q4sla;
    i=i+1;
end
timeq4=[time4 timeq41 timeq42 timeq45 timeq46 timeq47 timeq48];
q4s1=[q4in q4s11 q4s12 q4s15 q4s16 q4s17 q4s18];

% i=1;
for t=0:step:0.25*times
    an1=(0.175/0.5)*t;
    time5(i)=t;
    ank1(i)=an1;
    i=i+1;
end

i=1;
for t=(0.25*times)+step:step:0.75*times
    an2=0.175;
    timeq5(i)=t;
    ank2(i)=an2;
    i=i+1;
end

% i=1;
for t=0:step:0.25*times
    shldr=(0.0875/.5)*t;
    time6(i)=t;
    shldr1(i)=shldr;
    i=i+1;
end

i=1;
t=0.2;
for t=(0.25*times)+step:step:0.5*times
    m=-.875;
    shld2=m*(t-0.5)+0.175;
    %shld2=-0.875*t+0.35;
    timesd1(i)=t;
    shldra(i)=shld2;
    i=i+1;
end

i=1;
for t=(0.5*times+step):step:0.75*times
    m=(0.35-0.2625)/.5;
    shld2=m*(t-1)+0.2625;
    timesd2(i)=t;
    shldr(i)=shld2;
    i=i+1;
end

timeq6=[timesd1 timesd2];
shldr2=[shldr shldr b];
time=[timel timeq6];
qone=[q1s1];
qtwo=[q2s1];
qthr=[q3s1];
qfour=[q4s1];
ank=[ank1 ank2];
%Newton- Euler Iteration
%Inverse Dynamics Solution

function [Torque,f,w1,dw,dv,n1]=dynamicsequations(t1,t2,t3,t4,t5,t6,t7,L,R,w1loop,dwloop,dvloop,td,tdd,floop,n1loop)

f=floop;
w1=w1loop;
dw=dwloop;
dv=dvloop;
n1=n1loop;

l1=L(1);
r1=R(1);
l2=L(2);
r2=R(2);
l3=L(3);
r3=R(3);
l4=L(4);
r4=R(4);
l5=L(5);
r5=R(5);
l6=L(6);
r6=R(6);
l7=L(7);
r7=R(7);

%Transformations
[T01,T02,T03,T04,T05,T06,T07,T0H]=coordinates(t1,t2,t3,t4,t5,t6,t7,L,R);
T=[T01 T02 T03 T04 T05 T06 T07 T0H]

lcm1=[(T01(1,4)+T02(1,4))/2;
      (T01(2,4)+T02(2,4))/2;
      (T01(3,4)+T02(3,4))/2]

lcm2=[(T02(1,4)+T03(1,4))/2;
      (T02(2,4)+T03(2,4))/2;
      (T02(3,4)+T03(3,4))/2]
\[
lcm_3 = \frac{(T_{03}(1,4) + T_{04}(1,4))}{2}; \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad<br>
\[ dv(:,i+1) = R \cdot (\text{cross\,\,(\,dw(:,i+1),P)} + \text{cross\,\,(wl(:,i),cross\,(wl(:,i+1),P))) + dv(:,i))} \]

\[ dvc(:,i+1) = \text{cross\,\,(\,dw(:,i+1),CM1)} + \text{cross\,\,(wl(:,i+1),cross\,(wl(:,i+1),CM1))) + dv(:,i+1)} \]

\[ F(:,i+1) = (m(i) \cdot dvc(:,i+1)) \]

\[ N(:,i+1) = (I \cdot dw(:,i+1) + \text{cross\,\,(wl(:,i+1),I \cdot wl(:,i+1)))} \]

\[
%\text{Initializing}
\]

\[
\begin{align*}
& i = n - 1; \\
%\text{Inward Iterations}
& \text{while } i >= 1; \\
& \quad R_i = [T(1,j) \ T(1,j+1) \ T(1,j+2); T(2,j) \ T(2,j+1) \ T(2,j+2); T(3,j) \ T(3,j+1) \ T(3,j+2)]; \\
& \quad P_i = [T(1,j+3); T(2,j+3); T(3,j+3)]; \\
& \quad CMI = \text{CM(:,i)} \\
& \quad f(:,i) = R_i \cdot f(:,i+1) + F(:,i+1) \\
& \quad n1(:,i) = (N(:,i+1) + R_i \cdot n1(:,i+1) + \text{cross\,\,(CMI, F(:,i+1))) + cross\,(P, R_i \cdot f(:,i+1)))}; \\
& \quad \text{torque1}(i) = [n1(:,1)]' \cdot [0; 0; 1]; \\
& \quad i = i - 1; \\
& \quad j = 4 \cdot i - 3; \\
& \text{end}
\]

\[
\text{Torque} = n1' \cdot [0; 0; 1]
\]
APPENDIX H: CODE FOR NEWTON-EULER ITERATION FOR HUMANOID

function [Torque, f, w, dw, dv] = humanoid_inv_dyn_tree(t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24, t25, t26, t27, t28, t29, t30, t31, t32, t33, t34, t35, L, R, w1loop, dwloop, dvloop, dt, ddt, floop, increment, num)

% Newton-Euler Iteration
% Inverse Dynamics Solution

w1 = w1loop;
dw = dwloop;
dv = dvloop;
f = floop;

k = 1;
j = 1;
f1 = [f(1, j) f(1, j+1) f(1, j+2) f(1, j+3) f(1, j+4) f(1, j+5) f(1, j+6) f(1, j+7);
f(2, j) f(2, j+1) f(2, j+2) f(2, j+3) f(2, j+4) f(2, j+5) f(2, j+6) f(2, j+7);
f(3, j) f(3, j+1) f(3, j+2) f(3, j+3) f(3, j+4) f(3, j+5) f(3, j+6) f(3, j+7)]

f1loop = f1;

w1 = [0; 0; 0];
dw = [0; 0; 0];
dv = [0; 9.81; 0];

% Transformations

[T, oTp1, oTp2, oTp3, oTt, oTn1, oTn2, oTn3, oTls1, oTls2, oTls3, oTle, oTlw1, oTlw2, oTlw3, oTlh, oTrs1, oTrs2, oTrs3, oTre, oTrw1, oTrw2, oTrw3, oTrh, oTlhp1, oTlhp2, oTlhp3, oTlk, oTlan1, oTlan2, oTlan3, oTlf, oTrhp1, oTrhp2, oTrhp3, oTrk, oTran1, oTran2, oTran3, oTrf] = coordinates_dyn(t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24, t25, t26, t27, t28, t29, t30, t31, t32, t33, t34, t35, increment);

Tall = [oTp1 oTp2 oTp3 oTt oTn1 oTn2 oTn3 oTls1 oTls2 oTls3 oTle oTlw1 oTlw2 oTlw3 oTlh oTrs1 oTrs2 oTrs3 oTre oTrw1 oTrw2 oTrw3 oTrh oTlhp1 oTlhp2 oTlhp3 oTlk oTlan1 oTlan2 oTlan3 oTlf oTrhp1 oTrhp2 oTrhp3 oTrk oTran1 oTran2 oTran3 oTrf];

Tall = [oTp1 oTp2 oTp3 oTt oTn1 oTn2 oTn3 oTls1 oTls2 oTls3 oTle oTlw1 oTlw2 oTlw3 oTlh oTrs1 oTrs2 oTrs3 oTre oTrw1 oTrw2 oTrw3 oTrh oTlhp1 oTlhp2 oTlhp3 oTlk oTlan1 oTlan2 oTlan3 oTlf oTrhp1 oTrhp2 oTrhp3 oTrk oTran1 oTran2 oTran3 oTrf];
oTlhp2 oTlhp3 oTlk oTlan1 oTlan2 oTlan3 oTlf oTrhp1 oTrhp2 oTrhp3 oTrk
oTran1 oTran2 oTran3 oTrf;

lcm1=[(oTp1(1,4)+oTp2(1,4))/2;
(oTp1(2,4)+oTp2(2,4))/2;
(oTp1(3,4)+oTp2(3,4))/2];

lcm2=[(oTp2(1,4)+oTp3(1,4))/2;
(oTp2(2,4)+oTp3(2,4))/2;
(oTp2(3,4)+oTp3(3,4))/2];

lcm3=[(oTp3(1,4)+oTt(1,4))/2;
(oTp3(2,4)+oTt(2,4))/2;
(oTp3(3,4)+oTt(3,4))/2];

lcm4=[(oTt(1,4)+oTn1(1,4))/2;
(oTt(2,4)+oTn1(2,4))/2;
(oTt(3,4)+oTn1(3,4))/2];

lcm5=[(oTn1(1,4)+oTn2(1,4))/2;
(oTn1(2,4)+oTn2(2,4))/2;
(oTn1(3,4)+oTn2(3,4))/2];

lcm6=[(oTn2(1,4)+oTn3(1,4))/2;
(oTn2(2,4)+oTn3(2,4))/2;
(oTn2(3,4)+oTn3(3,4))/2];

lcm7=[oTn3(1,4);
oTn3(2,4);
oTn3(3,4)];

lcm8=[(oTls1(1,4)+oTls2(1,4))/2;
(oTls1(2,4)+oTls2(2,4))/2;
(oTls1(3,4)+oTls2(3,4))/2];

lcm9=[(oTls2(1,4)+oTls3(1,4))/2;
(oTls2(2,4)+oTls3(2,4))/2;
(oTls2(3,4)+oTls3(3,4))/2];

lcm10=[(oTls3(1,4)+oTle(1,4))/2;
(oTls3(2,4)+oTle(2,4))/2;
(oTls3(3,4)+oTle(3,4))/2];

lcm11=[(oTle(1,4)+oTlw1(1,4))/2;
(oTle(2,4)+oTlw1(2,4))/2;
(oTle(3,4)+oTlw1(3,4))/2];

lcm12=[(oTlw1(1,4)+oTlw2(1,4))/2;
(oTlw1(2,4)+oTlw2(2,4))/2;
\( \frac{(oTlw1(3,4) + oTlw2(3,4))}{2} \);

\( lcm_{13} = \left[ \frac{(oTlw1(1,4) + oTlw3(1,4))}{2}; (oTlw1(2,4) + oTlw3(2,4))}{2}; (oTlw1(3,4) + oTlw3(3,4))}{2} \); 

\( lcm_{14} = \left[ \frac{(oTlw3(1,4) + oTlh(1,4))}{2}; (oTlw3(2,4) + oTlh(2,4))}{2}; (oTlw3(3,4) + oTlh(3,4))}{2} \); 

\( lcm_{15} = \left[ \frac{(oTlw2(1,4) + oTlw3(1,4))}{2}; (oTlw2(2,4) + oTlw3(2,4))}{2}; (oTlw2(3,4) + oTlw3(3,4))}{2} \); 

\( lcm_{16} = \left[ \frac{(oTlw3(1,4) + oTlh(1,4))}{2}; (oTlw3(2,4) + oTlh(2,4))}{2}; (oTlw3(3,4) + oTlh(3,4))}{2} \); 

\( lcm_{17} = \left[ \frac{(oTlw2(1,4) + oTlw3(1,4))}{2}; (oTlw2(2,4) + oTlw3(2,4))}{2}; (oTlw2(3,4) + oTlw3(3,4))}{2} \); 

\( lcm_{18} = \left[ \frac{(oTlw3(1,4) + oTlh(1,4))}{2}; (oTlw3(2,4) + oTlh(2,4))}{2}; (oTlw3(3,4) + oTlh(3,4))}{2} \); 

\( lcm_{19} = \left[ \frac{(oTlw2(1,4) + oTlw3(1,4))}{2}; (oTlw2(2,4) + oTlw3(2,4))}{2}; (oTlw2(3,4) + oTlw3(3,4))}{2} \); 

\( lcm_{20} = \left[ \frac{(oTlw3(1,4) + oTlh(1,4))}{2}; (oTlw3(2,4) + oTlh(2,4))}{2}; (oTlw3(3,4) + oTlh(3,4))}{2} \); 

\( lcm_{21} = \left[ \frac{(oTlw2(1,4) + oTlw3(1,4))}{2}; (oTlw2(2,4) + oTlw3(2,4))}{2}; (oTlw2(3,4) + oTlw3(3,4))}{2} \); 

\( lcm_{22} = \left[ \frac{(oTlw3(1,4) + oTlh(1,4))}{2}; (oTlw3(2,4) + oTlh(2,4))}{2}; (oTlw3(3,4) + oTlh(3,4))}{2} \); 

\( lcm_{23} = \left[ \frac{(oTlw2(1,4) + oTlw3(1,4))}{2}; (oTlw2(2,4) + oTlw3(2,4))}{2}; (oTlw2(3,4) + oTlw3(3,4))}{2} \); 

\( lcm_{24} = \left[ \frac{(oTlw3(1,4) + oTlh(1,4))}{2}; (oTlw3(2,4) + oTlh(2,4))}{2}; (oTlw3(3,4) + oTlh(3,4))}{2} \);
\( \text{lcm25} = \frac{\text{oTlk}(1,4) + \text{oTlan1}(1,4)}{2}; \\
\frac{\text{oTlk}(2,4) + \text{oTlan1}(2,4)}{2}; \\
\frac{\text{oTlk}(3,4) + \text{oTlan1}(3,4)}{2}; \\
\)

\( \text{lcm26} = \frac{\text{oTlan1}(1,4) + \text{oTlan2}(1,4)}{2}; \\
\frac{\text{oTlan1}(2,4) + \text{oTlan2}(2,4)}{2}; \\
\frac{\text{oTlan1}(3,4) + \text{oTlan2}(3,4)}{2}; \\
\)

\( \text{lcm27} = \frac{\text{oTlan2}(1,4) + \text{oTlan3}(1,4)}{2}; \\
\frac{\text{oTlan2}(2,4) + \text{oTlan3}(2,4)}{2}; \\
\frac{\text{oTlan2}(3,4) + \text{oTlan3}(3,4)}{2}; \\
\)

\( \text{lcm28} = \frac{\text{oTlan3}(1,4) + \text{oTlf}(1,4)}{2}; \\
\frac{\text{oTlan3}(2,4) + \text{oTlf}(2,4)}{2}; \\
\frac{\text{oTlan3}(3,4) + \text{oTlf}(3,4)}{2}; \\
\)

\( \text{lcm29} = \frac{\text{oTrhp1}(1,4) + \text{oTrhp2}(1,4)}{2}; \\
\frac{\text{oTrhp1}(2,4) + \text{oTrhp2}(2,4)}{2}; \\
\frac{\text{oTrhp1}(3,4) + \text{oTrhp2}(3,4)}{2}; \\
\)

\( \text{lcm30} = \frac{\text{oTrhp2}(1,4) + \text{oTrhp3}(1,4)}{2}; \\
\frac{\text{oTrhp2}(2,4) + \text{oTrhp3}(2,4)}{2}; \\
\frac{\text{oTrhp2}(3,4) + \text{oTrhp3}(3,4)}{2}; \\
\)

\( \text{lcm31} = \frac{\text{oTrhp3}(1,4) + \text{oTrk}(1,4)}{2}; \\
\frac{\text{oTrhp3}(2,4) + \text{oTrk}(2,4)}{2}; \\
\frac{\text{oTrhp3}(3,4) + \text{oTrk}(3,4)}{2}; \\
\)

\( \text{lcm32} = \frac{\text{oTrk}(1,4) + \text{oTran1}(1,4)}{2}; \\
\frac{\text{oTrk}(2,4) + \text{oTran1}(2,4)}{2}; \\
\frac{\text{oTrk}(3,4) + \text{oTran1}(3,4)}{2}; \\
\)

\( \text{lcm33} = \frac{\text{oTran1}(1,4) + \text{oTran2}(1,4)}{2}; \\
\frac{\text{oTran1}(2,4) + \text{oTran2}(2,4)}{2}; \\
\frac{\text{oTran1}(3,4) + \text{oTran2}(3,4)}{2}; \\
\)

\( \text{lcm34} = \frac{\text{oTran2}(1,4) + \text{oTran3}(1,4)}{2}; \\
\frac{\text{oTran2}(2,4) + \text{oTran3}(2,4)}{2}; \\
\frac{\text{oTran2}(3,4) + \text{oTran3}(3,4)}{2}; \\
\)

\( \text{lcm35} = \frac{\text{oTran3}(1,4) + \text{oTrf}(1,4)}{2}; \\
\frac{\text{oTran3}(2,4) + \text{oTrf}(2,4)}{2}; \\
\frac{\text{oTran3}(3,4) + \text{oTrf}(3,4)}{2}; \\
\)

CM = [lcm1 lcm2 lcm3 lcm4 lcm5 lcm6 lcm7 lcm8 lcm9 lcm10 lcm11 lcm12 lcm13 lcm14 lcm15 lcm16 lcm17 lcm18 lcm19 lcm20 lcm21 lcm22 lcm23 lcm24 lcm25 lcm26 lcm27 lcm28 lcm29 lcm30 lcm31 lcm32 lcm33 lcm34 lcm35];
\[
M = \begin{bmatrix}
0 & 0 & 10 & 10 & 0 & 0 & 5 & 0 & 0 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 5 & 5 & 0 & 0 & 2 & 0 & 0 & 5 & 5 & 0 & 0 & 2
\end{bmatrix};
\]

\[
[Icc] = \text{Iccs}(M, L, R);
\]

\[g = 9.81;\]

\[s = 1;\]

\[\text{--------------------------------------------} \]
\[\text{FOR JOINTS OF THE TRUNK} \]
\[n = 7; \]
\[\text{--------------------------------------------} \]

\[\text{FORWARD ITERATIONS} \]
\[f = f_1; \]

\[\text{for } i = 1:n\]
\[j = 4s - 3; \]
\[k = 3s - 2; \]
\[R = \begin{bmatrix}
T(1, j) & T(1, j + 1) & T(1, j + 2); \\
T(2, j) & T(2, j + 1) & T(2, j + 2); \\
T(3, j) & T(3, j + 1) & T(3, j + 2)
\end{bmatrix}; \]
\[P = \begin{bmatrix}
T(1, j + 3); \\
T(2, j + 3); \\
T(3, j + 3)
\end{bmatrix}; \]
\[CM_1 = CM(:, s); \]
\[Icc(2, k + 2); Icc(3, k) Icc(3, k + 1) Icc(3, k + 2)]; \]
\[m(i) = M(s); \]
\[w_1(:, i + 1) = (R \cdot w_1(:, i) + dt(:, i)); \]
\[dw(:, i + 1) = (R \cdot dw(:, i) + \text{cross}(R \cdot w_1(:, i), dt(:, i)) + ddt(:, i)); \]
\[dv(:, i + 1) = R^* \text{cross}(dw(:, i + 1), P) + \text{cross}(w_1(:, i), \text{cross}(w_1(:, i + 1), P)) + dv(:, i)); \]
\[dvc(:, i + 1) = \text{cross}(dw(:, i + 1), CM_1) + \text{cross}(w_1(:, i + 1), \text{cross}(w_1(:, i + 1), CM_1)) + d \]
\[v(:, i + 1); \]
\[F(:, i + 1) = (m(i) \cdot dvc(:, i + 1)); \]
\[N(:, i + 1) = (I \cdot dw(:, i + 1) + \text{cross}(w_1(:, i + 1), I \cdot w_1(:, i + 1))); \]
\[s = s + 1; \]
\[\text{end} \]
\[\text{INITIALIZING} \]
\[n = n - 1; \]
\[n_1 = f; \]
\[\text{INWARD ITERATIONS} \]
\[\text{while } i > 1; \]
\[R_i = \begin{bmatrix}
T(1, j) & T(1, j + 1) & T(1, j + 2); \\
T(2, j) & T(2, j + 1) & T(2, j + 2); \\
T(3, j + 1) & T(3, j + 2)
\end{bmatrix}; \]
\[P_i = \begin{bmatrix}
T(1, j + 3); \\
T(2, j + 3); \\
T(3, j + 3)
\end{bmatrix}; \]
\[CM_1 = CM(:, i); \]
\[f(:, i) = R_i \cdot f(:, i + 1) + F(:, i + 1); \]
\[n_1(:, i) = (N(:, i + 1) + R_i \cdot n_1(:, i + 1) + \text{cross}(CM_1, F(:, i + 1)) + \text{cross}(P, R_i \cdot f(:, i + 1))) ; \]
i=i-1;
j=4*i-3;
end
f1=f;
wl1=wl1;
dw1=dw;
dv1=dv;
Torqueln1'*[0;0;1];

%FOR JOINTS OF THE LEFT HAND

f2=f1;
wl2=wl1;
dw2=dw;
dv2=dv;

n=7;

%Initializing
i=n-1;
n1=f2;

%Inward Iterations
while i>=1;
    Ri=[T(1,j) T(1,j+1) T(1,j+2); T(2,j) T(2,j+1) T(2,j+2);T(3,j) T(3,j+1) T(3,j+2)]';
    Pi=[T(1,j+3); T(2,j+3); T(3,j+3)];
    CMl=CM(:,i);
    f2(:,i)=Ri*f2(:,i+1)+F(:,i+1);
    n1(:,i)=(N(:,i+1)+Ri*n1(:,i+1)+cross(CM1,F(:,i+1))+cross(Pi,Ri*f2(:,i+1)));
    i=i-1;
end

dv2(:,i+1)=R*(cross(dw2(:,i+1),P)+cross(wl2(:,i),cross(wl2(:,i+1),P)))+dv2(:,i);
dvc(:,i+1)=cross(dw2(:,i+1),CM1)+cross(wl2(:,i+1),cross(wl2(:,i+1),CM1))
    +dv2(:,i+1);
    F(:,i+1)=(m(i)*dvc(:,i+1));
    N(:,i+1)=(I*dw2(:,i+1)+cross(wl2(:,i+1),I*wl2(:,i+1)));
s=s+1;
end

%Initializing
i=n-1;
n1=f2;

%Inward Iterations
while i>=1;
    Ri=[T(1,j) T(1,j+1) T(1,j+2); T(2,j) T(2,j+1) T(2,j+2);T(3,j) T(3,j+1) T(3,j+2)]';
    Pi=[T(1,j+3); T(2,j+3); T(3,j+3)];
    CMl=CM(:,i);
    f2(:,i)=Ri*f2(:,i+1)+F(:,i+1);
    n1(:,i)=(N(:,i+1)+Ri*n1(:,i+1)+cross(CM1,F(:,i+1))+cross(Pi,Ri*f2(:,i+1)));
    i=i-1;
j=4*i-3;
end
Torque2=n1'*[0;0;1];

%FOR JOINTS OF THE RIGHT HAND

f3=f1;
w13=w11;
dw3=dw1;
dv3=dv1;

n=7;

%Outward Iterations
for i=1:n
    j=4*s-3;
k=3*s-2;
R=[T(1,j) T(1,j+1) T(1,j+2); T(2,j) T(2,j+1) T(2,j+2); T(3,j) T(3,j+1) T(3,j+2)];
P=[T(1,j+3); T(2,j+3); T(3,j+3)];
CM1=CM(:,s);
I=[Icc(1,k) Icc(1,k+1) Icc(1,k+2); Icc(2,k) Icc(2,k+1) Icc(2,k+2)];
m(i)=M(s);
w13(:,i+1)=(R*w13(:,i)+dt(:,s));
dw3(:,i+1)=(R*dw3(:,i)+cross(R*w13(:,i),dt(:,s))+ddt(:,s));
dv3(:,i+1)=R*(cross(dw3(:,i+1),P)+cross(w13(:,i),cross(w13(:,i+1),P))+dv3(:,i));
dvc(:,i+1)=cross(dw3(:,i+1),CM1)+cross(w13(:,i+1),cross(w13(:,i+1),CM1));
F(:,i+1)=(m(i)*dvc(:,i+1));
N(:,i+1)=(I*dw3(:,i+1)+cross(w13(:,i+1),I*w13(:,i+1)));s=s+1;
end
%Initializing
i=n-1;
n1=f3;

%Inward Iterations
while i>1
    R1=[T(1,j) T(1,j+1) T(1,j+2); T(2,j) T(2,j+1) T(2,j+2); T(3,j) T(3,j+1) T(3,j+2)];
P1=[T(1,j+3); T(2,j+3); T(3,j+3)];
CM1=CM(:,i);
f3(:,i)=R1*f3(:,i+1)+F(:,i+1);
n1(:,i)=(N(:,i+1)+R1*n1(:,i+1)+cross(CM1,F(:,i+1))+cross(P,R1*f3(:,i+1)));
i=i-1;
j=4*i-3;
end
Torque3=n1'*[0;0;1];

%FOR JOINTS OF THE LEFT LEG
\begin{verbatim}
\texttt{n=7; 
\%Outward Iterations 
\texttt{for} \ i=1:n 
\quad \texttt{j=4*s-3; \ k=3*s-2; \ R=[T(1,j) \ T(1,j+1) \ T(1,j+2); T(2,j) \ T(2,j+1) \ T(2,j+2);T(3,j) \\
T(3,j+1) \ T(3,j+2)]; \ P=[T(1,j+3); T(2,j+3); T(3,j+3)]; \\
CM1=CM(:,s); \\
I=[Icc(1,k) Icc(1,k+1) Icc(1,k+2);Icc(2,k) Icc(2,k+1) Icc(2,k+2);Icc(3,k) Icc(3,k+1) Icc(3,k+2)]; \\
m(i)=M(s); \\
w14(:,i+1)=(R*w14(:,i)+dt(:,s)); \\
dw4(:,i+1)=(R*dw4(:,i)+cross(R*w14(:,i),dt(:,s))+ddt(:,s)); \\
dv4(:,i+1)=R*(cross(dw4(:,i+1),P)+cross(w14(:,i),cross(w14(:,i+1),P))+dv4(:,i)); \\
dvc(:,i+1)=cross(dw4(:,i+1),CM1)+cross(w14(:,i+1),cross(w14(:,i+1),CM1))+dv4(:,i+1); \\
F(:,i+1)=(m(i)*dvc(:,i+1)); \\
N(:,i+1)=(I*dw4(:,i+1)+cross(w14(:,i+1),I*w14(:,i+1))); \\
s=s+1; 
\texttt{end} 
\%Initializing 
i=n-1; 
nl=f4; 
\%Inward Iterations 
\texttt{while} \ i>1; 
\quad \texttt{Ri=[T(1,j) \ T(1,j+1) \ T(1,j+2); T(2,j) \ T(2,j+1) \ T(2,j+2);T(3,j) \\
T(3,j+1) \ T(3,j+2)]; \ Pi=[T(1,j+3); T(2,j+3); T(3,j+3)]; \\
CM1=CM(:,i); \\
f4(:,i)=Ri*f4(:,i+1)+F(:,i+1); \\
n1(:,i)=(N(:,i+1)+Ri*n1(:,i+1)+cross(CM1,F(:,i+1))+cross(P,Ri*f4(:,i+1))); \\
i=i-1; 
\quad \texttt{j=4*i-3; 
\texttt{end} 
Torque4=n1'*[0;0;1]; 
\texttt{\%FOR JOINTS OF THE RIGHT LEG 
wl5=wlloop; 
\texttt{dwl5=dwlloop; 
\texttt{dv5=dvloop; 
\texttt{f5=f1loop; 
}}} \end{verbatim}
n=7;
%Outward Iterations
for i=1:n
  j=4*s-3;
  k=3*s-2;
  R=[T(1,j) T(1,j+1) T(1,j+2); T(2,j) T(2,j+1) T(2,j+2);T(3,j)
    T(3,j+1) T(3,j+2)];
  P=[T(1,j+3); T(2,j+3); T(3,j+3)];
  CM1=CM(:,s);
  I=[Icc(1,k) Icc(1,k+1) Icc(1,k+2); Icc(2,k) Icc(2,k+1)
    Icc(2,k+2); Icc(3,k) Icc(3,k+1) Icc(3,k+2)];
  m(i)=M(s);
  w15(:,i+1)=(R*w15(:,i)+dt(:,s));
  dw5(:,i+1)=(R*dw5(:,i)+cross(R*w15(:,i),dt(:,s))+ddt(:,s));
  dv5(:,i+1)=R*(cross(dw5(:,i+1),P)+cross(w15(:,i),cross(w15(:,i+1),P))+
    dv5(:,i));
  dvc(:,i+1)=cross(dw5(:,i+1),CM1)+cross(w15(:,i+1),cross(w15(:,i+1),CM1))
    +dv5(:,i+1);
  F(:,i+1)=(m(i)*dvc(:,i+1));
  N(:,i+1)=(I*dw5(:,i+1)+cross(w15(:,i+1),I*w15(:,i+1)));
  s=s+1;
end

%Initializing
i=n-1;
n1=f5;

%Inward Iterations
while i>=1;
  Ri=[T(1,j) T(1,j+1) T(1,j+2); T(2,j) T(2,j+1) T(2,j+2);T(3,j)
    T(3,j+1) T(3,j+2)]';
  Pi=[T(1,j+3); T(2,j+3); T(3,j+3)];
  CM1=CM(:,i);
  f5(:,i)=Ri*f5(:,i+1)+F(:,i+1);
  n1(:,i)=(N(:,i+1)+Ri*n1(:,i+1)+cross(CM1,F(:,i+1))+cross(P,Ri*f5(:,i+1)));
  i=i-1;
  j=4*i-3;
end
Torque5=n1'*[0;0;1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w1=[w11 w12 w13 w14 w15];
dw=[dw1 dw2 dw3 dw4 dw5];
dv=[dv1 dv2 dv3 dv4 dv5];
f=[f1 f2 f3 f4 f5];
Torque=[Torque1; Torque2; Torque3; Torque4; Torque5];