MULTIBODY DYNAMICS MODEL OF A FULL HUMAN BODY FOR SIMULATING WALKING

A Thesis

Submitted to the Faculty

of

Purdue University

by

Zahra Khakpour

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

May 2017

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Hazim El-Mounayri, Chair

Department of Mechanical Engineering

Dr. Sohel Anwar

Department of Mechanical Engineering

Dr. Ali Razban

Department of Mechanical Engineering

Approved by:

Dr. Jie Chen

Head of the Departmental Graduate Program

This thesis is dedicated to my parents and my brother for all of their continued encouragement, love, and support.

ACKNOWLEDGMENTS

This thesis could not have been achievable without the assistance and support of several individuals who in one way or another contributed their valuable help in the preparation and completion of this study.

First and foremost, I would like to express my special thanks to Dr. Tamer Wasfy for providing this opportunity for me to learn and gain this much experience throughout my study. Without his technical advice and financial support, this work could not be done.

Also, I want to thank my advising committee, Dr. Hazim El-Mounayri, Dr. Ali Razban and Dr. Sohel Anwar for their time and direction during the completion of this thesis.

In Addition, I am indebted to Jafari family for helping, trusting, and supporting me during these years. I might not have the opportunity to study at IUPUI without supporting of the Jafaris Fellowship program created through the generosity of Dr. Ali Jafari and his wife Mrs. Maymanat Montaser.

I would also like to extend my gratefulness to Jerry Mooney for his instructions and advice through my graduate courses.

Moreover, I would like to take this opportunity to thank my friends for their solid support: Sara Makki-Alamdari, Bahareh Abbasi, Shadi Hassanzadeh, Ali Delbari, and Nojan AliAhmad.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Khakpour, Zahra M.S.M.E., Purdue University, May 2017. Multibody Dynamics Model of A Full Human Body For Simulating Walking. Major Professor: Hazim El-Mounayri.

Bipedal robotics is a relatively new research area which is concerned with creating walking robots which have mobility and agility characteristics approaching those of humans. Also, in general, simulation of bipedal walking is important in many other applications such as: design and testing of orthopedic implants; testing human walking rehabilitation strategies and devices; design of equipment and facilities for human/robot use/interaction; design of sports equipment; and improving sports performance & reducing injury. One of the main technical challenges in that bipedal robotics area is developing a walking control strategy which results in a stable and balanced upright walking gait of the robot on level as well as non-level (sloped/rough) terrains.

In this thesis the following aspects of the walking control strategy are developed and tested in a high-fidelity multibody dynamics model of a humanoid body model:

1. Kinematic design of a walking gait using cubic Hermite splines to specify the motion of the center of the foot.

2. Inverse kinematics to compute the legs joint angles necessary to generate the walking gait.

3. Inverse dynamics using rotary actuators at the joints with PD (Proportional-Derivative) controllers to control the motion of the leg links.

The thee-dimensional multibody dynamics model is built using the DIS (Dynamic Interactions Simulator) code. It consists of 42 rigid bodies representing the legs, hip,

spine, ribs, neck, arms, and head. The bodies are connected using 42 revolute joints with a rotational actuator along with a PD controller at each joint. A penalty normal contact force model along with a polygonal contact surface representing the bottom of each foot is used to model contact between the foot and the terrain. Friction is modeled using an asperity-based friction model which approximates Coulomb friction using a variable anchor-point spring in parallel with a velocity dependent friction law.

In this thesis, it is assumed in the model that a balance controller already exists to ensure that the walking motion is balanced (i.e. that the robot does not tip over).

A multi-body dynamic model of the full human body is developed and the controllers are designed to simulate the walking motion. This includes the design of the geometric model, development of the control system in kinematics approach, and the simulation setup.

1. INTRODUCTION

1.1 Motivation

The field of robotics has been extensively studied for many decades. Robots have been used in numerous industries and applications, such as autonomous vehicles, surgical robots, educational robots, military robots, manufacturing robots, and entertainment and media robots.

In recent decades, some research projects in the field of robotics have focused on the improvement of walking robots, and bipedal robots in particular [1]. This is the focus of this thesis. Specifically, this thesis is focused on developing a high-fidelity multibody dynamic model of the human body along with bipedal walking control strategy. In spite of the fact that humans take it for granted, walking is a very complicated process which involves: nonlinear/discontinuous dynamics and complex control of more than fifty-seven joints and muscles during the walking locomotion [2,3] to achieve an upright balanced walking motion. Recently, the study of biped and four-legged robots has attracted the attention of many research groups'.

Research topics include: design and development of testing walking machines and emulating human walking in terms of morphology, gait appearance, energy disbursed and control. The major advantage over other forms of locomotion, such as wheels, is that humanoid biped robots can easily handle real-world obstacles where the surrounding surfaces are unpredictable and irregular with many obstacles. Due to the human body anatomy and their capacity to communicate via body language, humanoid robots are discussed primarily in the context of service applications. The human-like robot, and in general, the biped robot, walk with two legs in different environments [4, 5]. The capacity of walk motion is essential to biped robots which need the ability to control their stability while walking [6].

The simulation of bipedal walking, and walking is important in numerous applications, like:

- **Robotics**
  - **Autonomous and Semi-autonomous Human-like Robots with Human-like Mobility**

    The main applications of a humanoid robot and the control strategy presented in this thesis are: autonomous robots, biomedical studies, and computer animation. Autonomous humanoid robots have potential in service applications, including domestic help applications, flexible manufacturing applications, exploring harsh environments, planetary exploration, military operations, security, surveillance, etc. One up and-coming application of a service robot would be assisting humans' in their daily activities within domestic applications such as, simple tasks and household responsibilities.

- **Biomedical**

  Multibody dynamics for biped robots, when treated as biological research, is part of this research effort. A multibody system encompasses a set of components that are interconnected. Each component helps with translational and rotational motions. Connections might be closed-loop configuration, or open-loop configurations [7].

  Highlighting the importance of safe and natural interactions with human users, many studies consider designing anthropomorphic robots. Anthropomorphic robots are effective in performing human tasks like walking and running [8], which was approved with a wide range of human medical needs from surgery through therapy and rehabilitation.

  - **Orthopedics:** testing and design of orthopedic implants
  - **Ergonomics:** design of equipment and facilities for human use/interaction

- **Rehabilitation:** testing of rehabilitation strategies and devices

  In the healthcare industry, the modeling of human body for biomedical, orthopedic and ergonomic applications has been studied. For example, there is a need to evaluate orthopedic and artificial limb solutions for seniors and disabled persons (in particular, the lower limbs). Yet, substantial challenges exist in the evaluation of anthropometric methods due to the problems related to safety in experimental settings and measurement accuracy.

- **Sports Medicine:**
  - Design of sports equipment
  - Improving sports performance & reducing injury
- **Domestic Applications:**

  A domestic robot must have different abilities like human robot interaction. Most robot platforms, however, support only one or two of the above skills [9–11].

- **Computer Generated Imagery (CGI) and Computer Animation**

  Another application of humanoid robots is in the animation and robotics industry, where automated human computer animations are used in movies and video games. Virtual environments are involved in computer graphics to provide a background for continuous interactions between characters and objects. In this case, even though creating an effective animation is challenging, it does play an important role to provide realism [12, 13].

  Artificial intelligence can be of value in this context. Infusing robotics and artificial intelligence into the gaming experience promises to be transformative, [14] redefining what is possible with virtual video games for real life applications in the physical world.

In order to have realistic animation, interactive character animation and simulated physics [12] are important. Although simulated physics are used widely to animate passive phenomena, kinematics-based approaches are still applied in commercial applications cite15. Significant improvements have recently occurred in usability and visual quality, which encourages the application of simulated physics in animation of an interactive character.

Also, there are recent research studies on the Man-machine Animation Real-Time Interface (MARTI) for automated special effects animations and human-computer interaction applications. MARTI introduces novel research in a number of engineering disciplines, such as speech recognition, facial modeling, and computer animation. These disciplines essentially need locomotion, for example, biped robots. One study utilized this method to control a human-like robot to play weightlifting and sprint games in the FIRA HuroCup league [15].

- **Educational**

  Other robots exist for entertainment purposes or for educational activities.

1.2 Literature Review

Various biped robots were created over the last several years and numerous biped robot control algorithms appear in the scholarly literature [16]. One important focus in existing literature is answering the question of how to create realistic and flexible locomotion controllers for physics-based characters. In this case, there are different approaches that have been described previously [9, 17–19]. The most common approaches are joint-space approaches. In this literature review the researches has been classified and reviewed based on the following group:

- Robots can be classified according to:
  - 2D/3D
  - Number of DOFs

- Actuator type

- Inverse kinematics and dynamics or Forward dynamics (multibody dynamics)
  - Frictional contact modeling

- Gait design and planning

- Balance control

Underlining individual joint actions tune the controller's parameters. There are some specific characteristics regarding individual joint actions. Since these actions are combined in a non-linear and interrelated technique, with regard to the joint controllers, it is hard to explain coordinated motion [20]. To modify the controllers for novel tasks, it is necessary to have substantial retuning or utilize ad-hoc methods, which helps to address redundancy. Providing target poses for control, to overcome these difficulties, motion capture is employed in some methods [21]. Gait planning for biped robots and path planning are significantly different. A biped robot refers to a ballistic mechanism. This mechanism interacts with the ground/environment. Its feet are used for these interactions. Due to the absence of control inputs and attractive forces, the joint of foot and ground is unilateral and under-actuated. Locomotion mechanics have some characteristics including unilaterality and under-actuation. These characteristics are inherent for the given mechanics, and rely on factors explaining postural instability and fall. Postural instability can cause serious consequences. Thus, thorough analysis is crucial to effectively calculate and address the chance of reduction [22]. Regarding under-actuated mechanical systems, there is a considerable amount of research examining walking control [18, 21, 23–26].

Meanwhile, the control strategies developed in other research [26], allowed for the foot rotation, as part of a natural gait and explicit control of the ZMP position, to be addressed.

The problem with making a biped robot able to dynamically walk stably is attention-grabbing because of the complication of the model:

- Biped robots are mechanical systems that classically have high degree of freedoms, that is, numerous links and joints recorded as coordinates in order to achieve locomotion [27].
- Impacts of the feet with the ground.
- Their variable structure: certainly the state dimension is different between various walking environment [28–31].

In order to have a stable biped robot, the study in Ref. [32] provide new criterion for rotational harmony of the foot is, in this way, an essential foundation for the assessment and control of step and security in biped robots [33]. Actually, foot turn seems to mirror lost adjust and be a possible reason for monopods and bipeds falling; two different types of biped robots with the most inclination to fall. Therefore, another rule named the FRI point demonstrates the condition of dependability of the robot [34, 35].

There are different learns about the strolling control of the mechanical frameworks and biped robots. Most of these systems are controlled by mechanisms that use tracking reference motions [26]. In the existing research on multi-body dynamics and the control of full human body robots, the control methods are designed for the invariant references trajectory. There is a comparison between the existing research and most contemporary relevant studies, listed in references and in regard to the following categories, which shows the motivation and objective of this research. Following is a list of researched topics:

Applications, 2D/3D Modeling, Solving kinematics, Degree of freedom, Experimental or theoretical result, Positioning system, Landmark navigation, trajectory.

Those that studied articles on 2D characters/robots [1,9,26,36,37], may not be able to apply their control law on 3D model, which is far from the goal of anthropomorphic robots.

Solving kinematics is one of the approaches that can be applied to the controller algorithms on the mode: Rather than processing the increasing speeds for a known

arrangement of joint torques, done may likewise have the capacity to do this the a different way, torques and powers are required for a character to play out a specific given movement [38, 39]. This procedure is called backwards kinematics. It is regularly utilized as a part of biomechanics to investigate the progression of human movement, utilizing movement information that is enlarged with outside drive estimations [40–43]. For the controller algorithms aspects, the pose-control graphs method has been utilized effectively to animate 2D characters [44]. However, the control of the 3D biped robots needs extra adjust amendment. The fundamental purpose behind this is with no following of worldwide interpretation and pivot of the objective stances [26]. Laszlo et al. utilizes posture control charts for adjusted 3D movement controllers utilizing limit-cycle control [45]. To look after adjust, direct stance amendments are connected toward the start of every strolling cycle, in view of pelvis turn or focus of mass. In later research, posture control charts have been utilized to produce various intuitive aptitudes, for example, climbing stairs [46] or client controlled skiing and snowboarding [47]. Considerable research on the control of biped robots, using the time-variant control law [26], is given. The target of building up a period variation control law prompts to asymptotically stable strolling. At the point when the reference strolling movement is produced, as for a scalar valued function of the condition of the robot rather than time, the controller does not fluctuate crosswise over time which enhances scientific tractability. Moreover, after control converges, the setup of the planar robot becomes the desired one. Likewise, for the same robot with the same cyclic motion, there is an important effect of the control law used. Control law based on a reference trajectory that derives from the current state of the robot leads to a balanced gait. Whereas, a control law whose reference motion is a function of time causes off-balanced walking [48]. For reasons like this, the virtual constraints method is the basis for the successful control of the planar bipeds in the previous work [26, 48–52]. In couple of research studies [17, 20], such a methodology was used to extend control to a walking robot moving in 3D with 2 degrees of under-actuation.

A review of the control of the biped robot leads to lump different control design methods includes 3 different categories. The first category regards to intelligent learning control, including controllers with implementing the fuzzy logic [53] and neural network methods [54]. The other control method regards to an intuitive control strategies [53].Finally, the third class involves pre-computed reference trajectories, as well as the development of a controller that enables following the trajectories.

In a previous reference [26], T. Wang presents a simple planar bipedal low-dimensional 2D model consisting of five links connected to form two legs with knees and a torso with five degree of freedom and all rotational joint.

In this research study, each revolute joint angle controlled by a a PD controller, proportional-derivative control is the most common feedback control method used in studies of joint-space motion control [22].

In spite of the high hopes on the basis of early advances, recent achievements in motion control that rely on stimulusresponse networks are few and far between. One effective technique in alternate frameworks is off-line parameter optimization [19, 55–58]. In a previously described article, [12] it is claimedthat off-line parameter optimization has not been a valid example of full-body humanoid biped motion relied upon stimulusresponse networks.

Although appearing to be unsuitable for the control of anything, with the possible exception of the simple biped characters(3D), the latest research studies imply that the use of advanced optimization for motion control is still reasonable [19, 57–59], but only so long as the framework used for control is appropriate to the desired task. This research applies the strategy of stimulusresponse network control.

Regarding the DOF of the models, a particularly clear and simple actuation model makes joint torques for all given DOF. Through this method, the quantity of model's DOF will be the number of the character's active DOFs.

Since muscles can just wrench, at minimum of 2 muscles are needed in order to control a DOF. Biological systems frequently have many more duplicative muscles [59]. It is unusual to utilize muscle-based actuation in animation because of the high

number of DOFs that need to be controlled and lower levels of simulation performance [36, 60–62]. In this research study, the mode has forty-two degrees of freedom.

Moreover, in contrast with other studies on the topic of the simulation the motion through the appropriate joints and links, we can see that several authors deal with walking and running gaits using toe rotation [63–65]. Yet, we now understand the human lower limb much better than in the past, in particular, the knee and ankle joints [66]. A benefit of using these two joints is that their complex architecture involves non-symmetric surfaces. Therefore, their motion is significantly more complex than the motion of a revolute joint.

The work of [37, 67–69] tells us that toe joints enable humans to perform longer strides, reduce energy consumption, and walk faster [1].

We often simplify character models as a means of enhancing simulation performance and improving the ease of control. Sometimes humans are treated as simple biped characters, in which a single body includes the head and the trunk [43, 70]. Wang et al. [19] included a different parts for toes in proposed foot model as a means of overcoming the drawbacks of common models attaching feet to a single body.

All knee joints, ankle joints, toe joints, and foot segments are simulated in these studies to approach a more stable full human body model.

One of the steps which should be considered a real-time physics simulation for this biped robot model, is collision detection between feet and ground [12]. Collision detection methodology is well covered in textbooks and scientific literature (e.g. [71]). Authors tend to ignore self-collision when examining physics-based character animation, however. Scholars have also neglected to study collisions between bodies that touch each other with a joint. In this research, the directions of two legs are checked to avoid any self-collision.

Human-Robot Interaction:

There are many different types of robots and functionalities but among all, a humanoid biped robot has particularly strong capacity for action, which allows it to mimic human behavior, such as, walking. Consequently, the interfaces and interac-

tions between humans and all different types of humanoid robots have received much attention recently. Humanoid biped robots are used to simplify everyday human life, help human workers avoid safety risks, provide entertainment, and benefit future human societies. Many researchers are active in this field and are drawn in by the promise of exciting future advances [72].

One of the methods used to check the performance of a system and a robot's inference with humans, such as, getting and analyzing feedback, is video games.

Model dimension, 2D/3D:

There are numerous studies working on 2D models and the design of robotic systems. Also there are studies on the patterns with two dimensional approaches. But over the last several decades studies on this area have grown and changed to 3D studies as well. Also, there are several studies working and comparing their control methods on both 2D and 3D patterns [73].

Low DOF/High DOF:

In robotics, the term "degree of the freedom" (DOF) denotes the complexity of a robot to control its joints. The more degrees of the freedom the more difficult it will be to maintain control. However, the modeled movement will be smoother and the results will be closer to reality. In this way, studies have a more accurate response to different inputs from the environment. The whole robot system can be molded as various DOF rigid bodies, but most of the realistic active walking robots have a high degree of freedom which makes controlling them a complex task [48]. Thus, a large body of research has been devoted to developing and simulating the biped robots with lower degrees of the freedom.

In a recent research study [48], a six-degrees of the freedom active biped walking robot was fabricated, in a previously described journal article [69] a two-legged robot where each leg had seven degrees of the freedom or LOLA robot [74]. Lastly, in [30], seven degrees of the freedom leg was used. In other publications, researchers report different values for the degrees of the freedom like seven degrees of the freedom [26,53], ten [18, 30, 75], and fourteen [26, 66].

Type of Actuator:

Most of the modules that have been utilized in the past use two common actuators: Rotational actuators and linear actuators. A linear actuator creates motion in a straight line, unlike a conventional electric motor which generates circular motion. Applications for linear actuators include robotic designs, industrial devices, and machine tools, among others [76]. An actuator that produces a rotational motion, or torque, is known as a rotational actuator. In its most modest form, an actuator can be completely mechanical, with linear motion input in one direction providing the escalation that permits rotation. However, most actuators use some sort of energy source [77].

One recent study [73] investigated both 2D and 3D models, both under and full actuation. V. Lebastard, in one of his more recent studies observed control of motion for a biped robot by using only actuated variables measurement. Thus the robot is under-actuated in a single support since there is no motor (actuator) in the ankle location of the robot.

In one of the biped walking robot studies, M. Hardt [78] presents a method to minimize the actuation energy as all while proving the natural walking motion by solving the minimum energy motion problem including algebraic and saturation constraints values to answer Hamilton-Jacobi-Bellman equation considering the optimal path [22].

Another study [49] uses a combination of both "Series Elastic Actuation" and "Limit Cycle Walking" to provide the real motion of a robot.

Boston Dynamics [79] is an active company in the field of robotics and developed a 28 DOF humanoid robot which implemented hydraulic actuators.

Robotic Inverse Kinematic:

In order for biped robots to follow a desired gait motion, the problem of solving inverse kinematics needs to be solved. A great deal of research exists on both general and specic robotic congurations. For systems with no closed-form (analytical) solutions [80], numerical solution strategies are used. Several strategies exist, including

Fig. 1.1. Atlas Robot, Twenty-eight Hydraulically Actuated DOF

the Secant method, the Newton-Raphson method, and other optimization methods. Since the Newton-Raphson method is used in this project, the following literature review primarily includes studies which utilize this method.

An important note about the Newton-Raphson approach is that it uses the pseudo-inverse of the Jacobian matrix to solve for the joint angles and that the singularity problem was avoided using the singular value decomposition (SVD) of the Householder matrix [81].

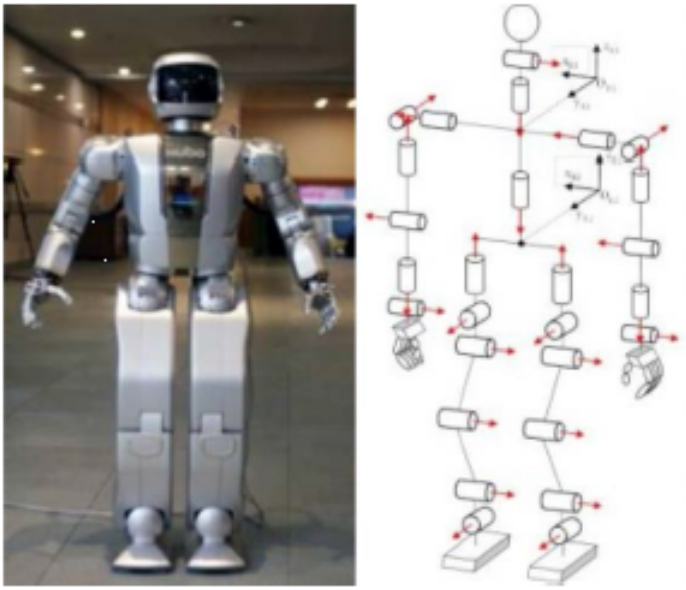

Fig. 1.2. Closed Form Inverse Kinematic [82]

Scholars have also sought to develop a generalized solution for the inverse kinematics of robots [83]. A recent study suggested one solution could be an arbitrary number of DOF. The study used an iterative process with a modied Newton-Raphson algorithm in which the number of DOFs is above six. This is because the Jacobian is not square and the inverse, or psuedo-inverse, cannot be found.

Control Model:

A large mass of research has used the zero moment point (ZMP) method; for example in a previously described study [84], S. Kajita introduced a new control method of ZMP which tracked serve controller considering of dynamics of the robot as a full model. Also, Y.J. Kim in his study [85] represents a balance control strategy for the 2D modeled biped robot.

There are several studies providing the ZMP method in view of ground reaction control and foot planting location control. In the Figure 1.3, the interface between these three control methods are demonstrated.

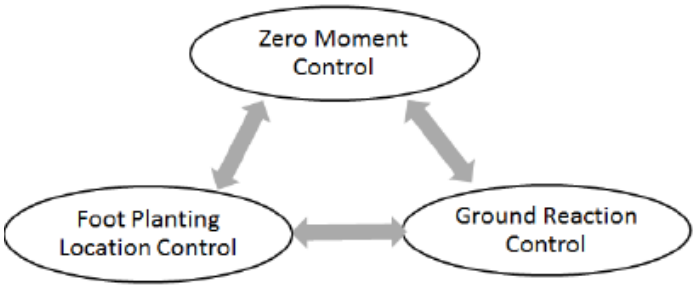

Fig. 1.3. Interfaces Between Three Main Control Methods

Among all these studies, T.zuu-Hseng S. Li presents [6] the newest new zero-moment point control methodology with modifiable parameter both in lateral and sagittal planes and makes the zero-moment point control trajectory more exible.

A recent focus in this area is the study of real-time modeling and trajectory generation. In a previous research study [20], an innovative method for real time trajectory generation is provided and allows for the tuning of the Fourier series parameters.

There are other studies [9, 45, 71] utilizing additional methods besides the ZMP, like the concept of center of pressure (CoP) with respect to ground feet contact forces. The foot rotation indicator (FRI) is an added novel method [43].

There are other related studies utilizing the comprehensive useful control methods to control the walking motion of robots in both a real environment (natural way) and simulation.

Balance Control Strategy:

The motion of an object results from how it interacts with the environment. Motion tells us a lot about how objects contact [43, 73]. Physical models with the objective of tracking people have successfully used information about inelastic ground planes [2, 15, 27]. The ability to detect contact between surfaces through motion is not well studied in the literature on computer vision, however [86].

Given limitations on what we can model using the various available commercial codes, biomechanical models of how the human body moves and those used to assess injuries are not yet interchangeable. The representation of the anatomical joints

by mechanical joints rather than contact pairs illustrate the limitations of current modeling techniques, as do the muscle actions that produce joint torques rather than extraneous muscle forces. Similarly, we cannot effectively model contact between human motion and environmental forces [87].

This study provides an overview of selected methodologies to study human motion, paying close attention to multibody dynamics approaches as a means of understanding significant areas of motion. I also describe how finite element procedures help us with the representation of abnormalities in physiological structures.

Most of the researches as mentioned, studied on the balance control of a robot or human body model with low degree of freedom and modeling the lower part of the body not full human body, in this study it has been worked on a multi-body dynamic model of full human body with 42-DOF.

1.3 Objectives

The main objectives in this research are as follows:

- Create a high-fidelity three-dimensional multibody dynamics humanoid body model using a multibody dynamics software, namely, DIS (Dynamics Interaction Simulator) [88]. The multibody dynamics model is used to predict the "forward dynamics" of the robot. The model consists of 42 rigid bodies representing the legs, hip, spine, ribs, neck, arms, and head. The bodies are connected using 42 revolute joints with a rotational actuator with a PD controller at each joint. A penalty normal contact force model along with a polygonal contact surface representing the bottom of each foot is used to show contact between the foot and theground. Friction is modeled using an asperity-based friction model whit approximating the Coulomb friction using a variable anchor-point spring in parallel with the velocity dependent friction law. DIS is explicit time-integration multibody dynamics code. The main advantage of the DIS code is

that it integrates the following computational techniques into one solver, which allows modeling flexible robots and real-world terrains:

  - Finite element method for modeling flexible bodies. Thus it can be used in case some of the robots' links are flexible (long /slender).
  - Discrete element method (DEM) for modeling granular and soil-type materials. Thus, the robot walking strategy can be tested on soft soil terrains.
  - Smoothed particle hydrodynamics (SPH) for modeling fluid flow. Thus the robot can be tested while walking in shallow water.

- Develop and test the following aspects of the bipedal walking control strategy and integrate them into the high-fidelity multibody dynamics robot model:

  - Kinematic design of a walking gait using cubic Hermite splines to specify the motion of the center of the foot.
  - Inverse kinematics to compute the legs joint angles necessary to generate the walking gait.
  - Inverse dynamics using rotary actuators at the joints with PD (Proportional-Derivative) controllers to control the motion of the leg links.

Note that in this thesis, it is assumed in the model that a balance controller already exists to ensure that the walking motions is balanced (i.e. that the robot does not tip over).

1.4 Contribution

This research presents a high fidelity multibody dynamics human body model along concurrent with a walking control strategy. The model includes 42 rigid bodies, 42 revolute joints and rotary actuators, and frictional contact surfaces for the feet. An inverse kinematics algorithm was used to make a target point on each foot that would

follow a desired walking path modeled using Hermite splines. The inverse kinematics algorithm uses Newton's method, a numerically calculated Jacobian, and a Moore-Penrose Pseudoinverse of the Jacobian to calculate the leg joint angles necessary to follow the walking path [89]. Numerical simulations of a human model walking on a flat surface are presented to demonstrate the model.

The main contribution of this thesis is creating and integrating a high-fidelity flexible multibody dynamics model of a full human body bipedal walking capability and DEM, using the multibody dynamics code.

2. MULTIBODY DYNAMICS FORMULATION

DIS, or Dynamic Interactions Simulator, was used to model the human body throughout this research. This was accomplished using multibody dynamics software and will be explained in this chapter. Additionally, the rigid body dynamics and control system were modeled with this software. Throughout this chapter, the integration of time and the motion equationswhich are utilized in the DIS code will be summarized.

For completeness, in the all equations in this chapter, these rulesare used (considering the indicial notations):

- For vector component numbers the lower case indices are used
- For node numbers the upper case indices are used
- Time derivative is represented by a superposed
- Time is represented by the superscript
- The convention of the Einstein summation is used for subscipte indices in the equations.

By definition, both humanoid robots and robot mechanisms are multibody systems where the actual multiple bodies are linked. From here on, the body will be assumed to be rigid.

2.1 Equations of Motion

Bones, when the entire human body is taken into account, will be represneted as rigid bodies. Thus, this rigid body is modeled as a finite element node. Additionally, this finite element node will be positioned at the center of mass, COM, of a single

rigid body. As shown previously in other study [7], the motion equations may be composed using explicit finite element code for 3-Dimensional rigid bodies. Every rigid body, or node, will contain three degrees of freedom. The definition of these degrees of freedom will be with reference to the rotational matrix and the inertial reference matrix . In Euler angles and Euler parameters [90, 91], for example, the utilization of all body rotation matrix to quantity the rigid body rotation may aid in avoiding the singularity issues related to three and four parameter rotation.

In terms of the nodes, the translational equations can be composed regarding the global inertial reference frame. They will be attained through gathering the individual rigid body equations and are written as:

$$M_W \ddot{x}^t_{Wi} = F^t_{s_{Wi}} + F^t_{s_{Wi}} \qquad \text{(Equation 2-1)}$$

Where:

M_W: Node W's lumped mass

x: Nodal Cartesian vector with respect to the global inertial reference coordinates

$\ddot{x}$: Nodal accelerations vector with respect to the global inertial reference coordinates

t: Time

W: Global node number to represent the nodes and start from 1 to N which is total number of all nodes

i : number of coordinates (i=1,2,3)

F_s: Internal structural force vector

F_a: external force vectorlike surface forces to rigid body

For every node a body-fixed material frame is defined. With respect to the body frame, the inertia of the body will be defined as I_{ij}, also called the inertia tensor. At time t_0, the body frame orientation can be defined as $R_W^{t_0}$. With respect to the global inertial reference coordinates, the body frame orientation is the rotation matrix at time t_0. Lastly, with respect to the body-fixed material coordinates, the equations of the rotational motion can be written for each node and are as follows:

$$I_{Wij}\ddot{\theta}^t_{Wj} = T^t_{s_{Wi}} + T^t_{a_{Wi}} - (\dot{\theta}^t_{Wi} * I_{Wij}\dot{\theta}^t_{Wj})_{Wi} \qquad \text{(Equation 2-2)}$$

Where:

I_W: rigid body W's inertia tensor

$\ddot{\theta}_{Wj}$: rigid body W's angular acceleration vector component with respect to its material coordinates in direction j $\dot{\theta}_{Wj}$: rigid body W's angular velocity vector component with respect to its material coordinates in direction j.

$T_{s_{Wi}}$: Components of the vector of internal torque in direction i at node W

$T_{a_{Wi}}$: Components of the vector of external torque applied to the rigid body

For only the indices i and j, a summation procedure is utilized. I_W, the inertia tensor, will remain constant due the rigid body equations being written in the body frame. In order to solve Equation 2-1 for the global nodal position x, the trapezoidal rule is utilized and is shown as:

$$\dot{x}^t_{Kj} = \dot{x}^{t-\Delta t}_{Kj} + 0.5\Delta t(\ddot{x}^t_{Kj} + \ddot{x}^{t-\Delta t}_{Kj}) \qquad \text{(Equation 2-3)}$$

$$x^t_{Kj} = x^{t-\Delta t}_{Kj} + 0.5\Delta t(\dot{x}^t_{Kj} + \dot{x}^{t-\Delta t}_{Kj}) \qquad \text{(Equation 2-4)}$$

In the equation above, Δt is defined as the time step. In Equation 2-2, the trapezoidal rule can likewise be used as the formula with time integration for the rotation increments of the nodesand is written as:

$$\dot{\theta}^t_{Kj} = \dot{\theta}^{t-\Delta t}_{Kj} + 0.5\Delta t(\ddot{\theta}^t_{Kj} + \ddot{\theta}^{t-\Delta t}_{Kj}) \quad \text{(Equation 2-5)}$$

$$\Delta\theta^t_{Kj} = 0.5\Delta t(\dot{\theta}^t_{Kj} + \dot{\theta}^{t-\Delta t}_{Kj}) \quad \text{(Equation 2-6)}$$

Where, for body K, $\Delta\theta_{Kj}$ is the incremental rotation angle around all three body axe. In order to produce the incremental rotation angles, the motion rotational equations are integrated. The rotation matrix, which parallels the incremental rotation angles, is utilized to give the new value to the rotation matrix of body K as described by R_K, and is written as:

$$R^t_K = R^{t-\Delta t}_K R(\Delta\theta^t_{Ki}) \quad \text{(Equation 2-7)}$$

$R(\Delta\theta^t_{Ki})$ from Equation 2-6, is the rotation matrix. This rotation matrix is equivalent to the incremental rotation angles. Revealed in the following section is the process for resolving Equations 2-1 to 2-7. The constraint equations, which define the nodes' position or velocity, are presented. In short, the constraint equations are: Contact/impact constraints:

$$f(\{x\}) \geq 0 \quad \text{(Equation 2-8)}$$

Joint constraints:

$$f(\{x\}) = 0 \quad \text{(Equation 2-9)}$$

Prescribed motion constraints:

$$f(\{x\}, t) = 0 \quad \text{(Equation 2-10)}$$

2.2 Joint Modeling

This section will serve to provide a short synopsis of modeling joints through rigid body kinematics simulation environment. In order to comprehend human motion

kinematics, the study of general motion in a multibody system needs to be considered. This needs to be completed with a specific emphasis on kinematic pair restrictions. These restrictions are equivalent to anatomical joints in the human body.

For practical purposes, a joint is fundamentally an association between joining points. Thus, joints introduce restrictions on motion between rigid body points, of which a rigid body can have several connection points.

A connection point, or joint location on human body, will not introduce extra DOF to a model. Equations 2-11 and 2-12 provide connection points position and its velocity, respectively, where x_{LP} is the position of the connection point comparative to the body's frame. Additionally, x_{GP} and $\dot{x}_{GP}$ are the position and velocity of the connection point relative to the global reference coordinates. Finally, the position and velocity of a point on a rigid body relative to the global reference frame Figure 2.1 can be written as:

$$x_{GP_i} = X_{B_i} + R_{B_{ij}} x_{LP_j} \qquad \text{(Equation 2-11)}$$

$$\dot{x}_{GP_i} = \dot{X}_{B_i} + R_{B_{ij}} (W_{BF} * x_{LP})_j \qquad \text{(Equation 2-12)}$$

2.2.1 Spherical Joint

The study of spherical joints aids this work in the ability to construct connections among bone. Furthermore, spherical joints can be utilized to formulate other types of joints, including revolute joints. These spherical joints attach two separate points on the other two separate bodies. In order to obtain similar translational coordinates, relative to the global reference frame, the two points are restricted by the spherical joint.

Consequently, between two connection points, a spherical joint can be written as:

$$X^t_{c1_i} = X^t_{c2_i} \quad \Rightarrow X^t_{c1_i} - X^t_{c2_i} = 0 \qquad \text{(Equation 2-13)}$$

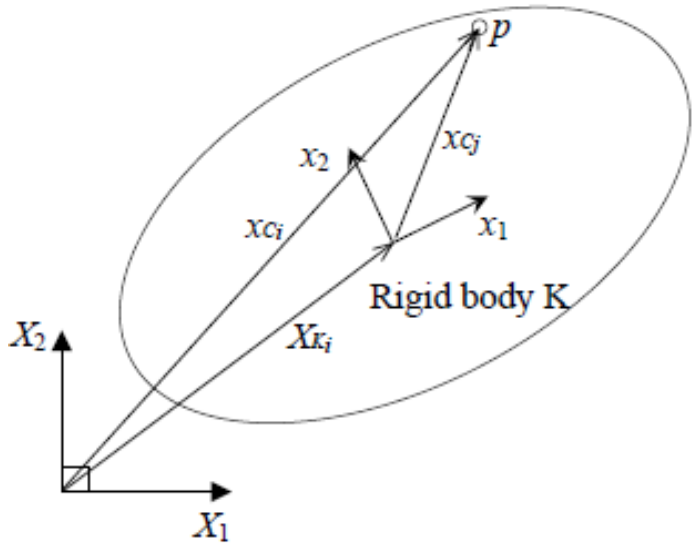

Fig. 2.1. Demonstrating Rigid Body by Relation to Connection Points

In the equation, $X^t_{c1_i}$ is the first point $c1$ position(global). Additionally, $X^t_{c2_i}$ is the second point $c2$ poition (global). Then, between two rigid bodies, a spherical joint plants three relative rotational DOF free and restricts three relative translational DOF. Accordingly, the procedure that implements this penalty is as follows.

$$F_P = k_p d + c_p d_i \dot{d}_i \qquad \text{(Equation 2-14)}$$

$$d_i = X^t_{c1_i} - X^t_{c2_i} \qquad \text{(Equation 2-15)}$$

$$\dot{d}_i = \dot{X}^t_{c1_i} - \dot{X}^t_{c2_i} \qquad \text{(Equation 2-16)}$$

$$d = \sqrt{d_1^2 + d_2^2 + d_3^2} \qquad \text{(Equation 2-17)}$$

$$F_{p_t} = F_p d_i / d \qquad \text{(Equation 2-18)}$$

Where:

$\dot{X}^t_{c1_i}$: Global velocity vector for point $c1$

$\dot{X}^t_{c2_i}$: Global velocity vector for point $c2$

F_P: Penalty force magnitude

F_{P_i}: Force of penalty reaction on $c1$

k_p: The stiness of penalty spring

c_p: Penalty damping

d_i: distance vector between $c1$ to$c2$

$\dot{d}_i$: rate of change of distance between $c1$ to $c2$ in time

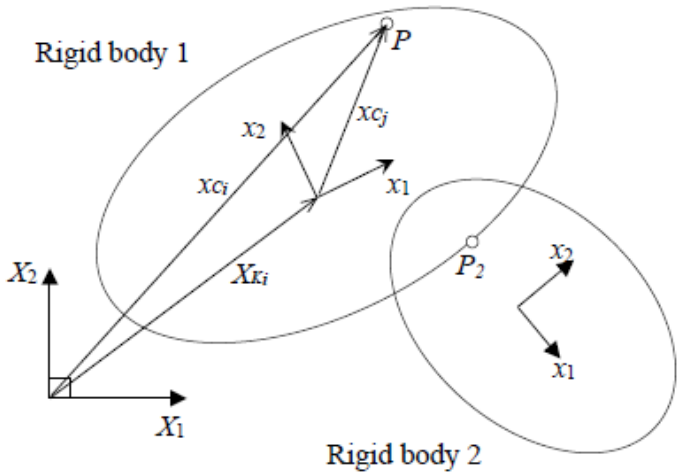

Fig. 2.2. Spherical Joint Demonstration

The joint penalty force, in opposing directions, is affixed to the two connection points. This force, then, attempts to correspond points $X^t_{c1_i}$ and $X_{c2_i})^t$ by implementing the restraint shown below:

$$X^t_{c1_i} = X^t_{c2_i} \quad d_i = X^t_{c1_i} - X^t_{c2_i} \qquad \text{(Equation 2-19)}$$

The joint force is then shifted to the center of gravity of the matching rigid body, which is equivalent to the center of body frame. This is completed through utilizing the following:

$$F_i = F_{p_i} \qquad \text{(Equation 2-20)}$$

$$T_i = -(x_{c1_i} * R_{ji} F_{p_i}) \quad d_i = X^t_{c1_i} - X^t_{c2_i} \qquad \text{(Equation 2-21)}$$

Where:

F_i: The force at the center of the gravity of the rigid body

T_i: Moment on the rigid body

R_{ji}: Rigid body rotation matrixx_{c1_i}: The point position(to the rigid body's coordinates)

2.2.2 Revolute Joint

Revolute joints comprise the majority of the joints in a robotic arm. These joints are regularly referred to as hinge, or rotational joints, and serve to link two rigid bodies. A revolute joint can be presented through aligning two spherical joints in a line.

This serves to restrain the three DOF in both the translational and rotational direction amongst the two rigid bodies. This then, allows a singular DOF in the rotational direction (see Figure 2.3). Finally, revolute joints can be employed as either active or passive joints.

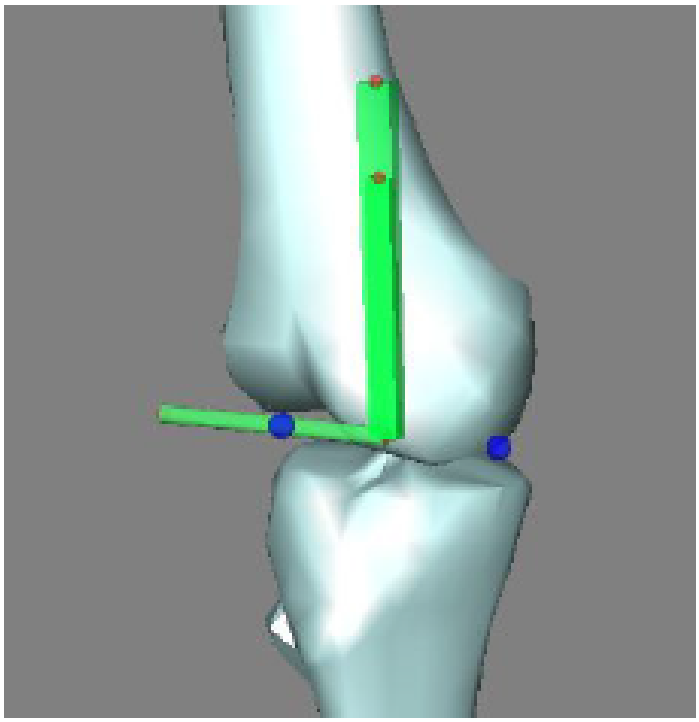

Fig. 2.3. Revolute Joint Used in Human Body Model

2.3 Rotational Actuators

Actuators are of paramount importance to this work. They serve to produce forces on rigid bodies among individual points. Only rotational actuators that will be utilized in this study.

As shown in Figure 2.4, the rotational actuator joins 3 points on two individual rigid bodies. Through the use of a PD controller, the value of the torque representing by T created by the actuator can be written as:

$$T = k(\theta - \theta_{des}) + c(\dot{\theta} - \dot{\theta}_{des}) \qquad \text{(Equation 2-23)}$$

Where:

θ: Actuator's angle (current value)Current angle of the actuator

θ_{des}: Desired actuator's angle

k: Proportional gain

c: Derivative gain

Making use of both Equations 2-20 and 2-21, it can be shown that the rotary spring forces are shifted to the rigid body's center. This transfer is completed as force and moment of inertia.

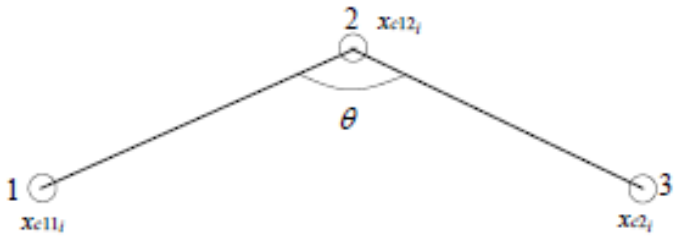

Fig. 2.4. Rotational Actuator

2.4 Contact Model

Throughout this study, contact modeling was used to calculate the interactions between the human body model's feet and the ground. This methodology requires using articular surface geometry and developing effective methods to obtain the distances between two contact surfaces.

The contact model used alters depending on several factors ranging from different surfaces with various friction coefficients to different type of toes and ground interfaces. Depending on how the model is designed, whether or not shoes are used may affect the design equation.

The purpose of this section is to outline the method for modeling contact between rigid bodies, which is based on previous work by Wasfy et al. [90, 91].

Contact constraints are of the form:

$$f(\{x\}) \geq 0 \qquad \text{(Equation 2-24)}$$

As explained previously, the penalty technique that is utilized will present the normal contact restraints between all the points of two rigid bodies.

In this method, to define the value of the position and velocity of the contact point:

$$X_{ci}^{t} = X_{Ki}^{t} + R_{Kij}^{t} x_{cj} \qquad \text{(Equation 2-25)}$$

$$\dot{X}_{ci}^{t} = \dot{X}_{Ki}^{t} + R_{Kij}^{t}(\{\dot{\theta}_{K}^{t}\} * \{x_c\})_j \qquad \text{(Equation 2-26)}$$

Where:

x_{cj} : Coordinates of c: contact point with respect to local frame of first body

$\dot{X}_{c_i}^{t}$: Velocity components of c : contact point (to global coordinates at t)

$X_{c_i}^{t}$: Coordinates of c :contact point with respect to global frame at time t

$\dot{X}_{K_i}^{t}$: Velocity components of center of first body frame (to global coordinates at t)

$X_{K_i}^{t}$: Coordinates first body frame center (to global coordinates at t)

$R_{K_i}^{t}$: Rotation matrix of the first rigid body at time t

$\dot{\theta}_{K}^{t}$: Angular velocity components of the first body at time t

$$F_{c_i} = F_{t_i} + F_{n_i} \qquad \text{(Equation 2-27)}$$

F_c is transferred as a force and a torque to the center of the node through using the following equations:

$$F_i = F_{c_i} \qquad \text{(Equation 2-28)}$$

$$T_i = (x_{LP_i} * R_{BF_{ji}} F_{c_i}) \qquad \text{(Equation 2-29)}$$

$$x_{LP_j} = BF_{ji}(x_{GP_i} - X_{BF_i}) \qquad \text{(Equation 2-30)}$$

Where:

F_i: This is the contact force at the rigid body's center of gravity

$x_{L_{cp}}$: The contact point position (to definedrigid body's coordinates)

T_i: The contact moment

Correspondingly, the opposite of $= F_i$, is shifted to the other rigid body center (contacting body) in place of a force and moment of inertia.

2.4.1 Frictional Contact

In cases where joint and contact friction is involved, asperity-spring friction models can be utilized.

The model of friction is then formed using a piecewise linear velocity(Coulomb friction element).

Asperity-Friction Theory

If, on a microscopic scale, the contact surface between objects is viewed, a surface that may appear smooth actually has asperities. These asperities can range from one to several thousand molecules. The asperities will interlock between two surfaces when they are in contact.

The block's motion will be prevented once there is a tangential force placed on it. The motion will be prevented due to the normal contact forces between the asperities on both surfaces. Therefore, the tangential force and the friction force will be equal.

There is however a chance that the asperities will either break or deform and allow motion of the block. This may occur if the magnitude of the tangential force is more than the product of the friction coefficient, (μ), and the normal reaction force, N.

If the case that the normal force, N, is increased, then the surface asperities interlock more compactly. In this circumstance, a superior tangential force is essential to break or alter the shape of the asperities in order for the block to move. Chemical bonding and electrical interactions are assumed to be negligible in this situation.

Asperity-Friction Model

Asperity friction is approximated by the asperity friction model. More specifically, the friction can be modeled where friction forces between two contacts rise because of the surface asperitiesthe interaction of the surfaces. $F_{tangential_i}$ is the tangential friction contact force and written as:

$$F_{tangential_i} = F_{tangent} t_i \qquad \text{(Equation 2-31)}$$

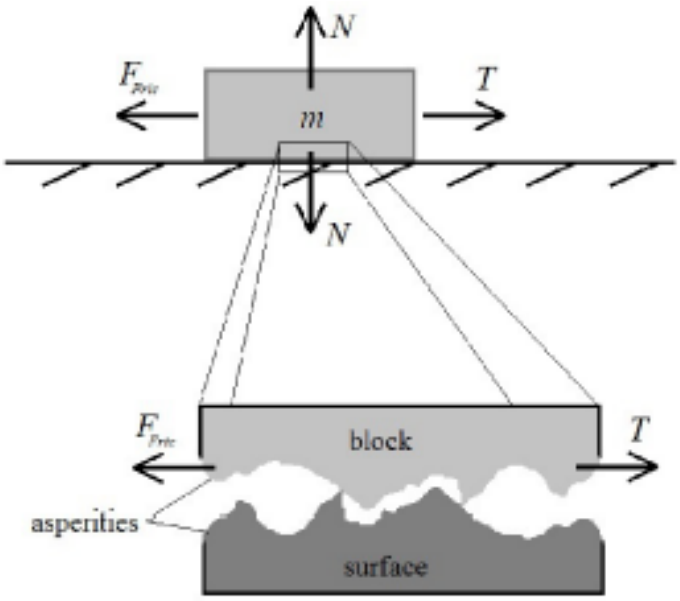

Fig. 2.5. Friction's Physical Interpretation

The tangential friction force, or ($F_{tangent}$), can be calculated from the asperity friction model by making use of the normal force. The surface asperities will be as tangential springs when two surfaces are in static contact. Additionally, the springs

will deform make the surfaces back to initial positions of each ones when a tangential force is applied.

When the tangential force becomes great enough, the surface asperities yield, or in other words, the springs break. Consequently, there will be sliding between the surfaces. The breakaway force is then equivalent to the contact pressure (normal value). Asperities will introduce resistance to the motion while surfaces slide in opposite directions. This motion is a velocity friction (sliding), acceleration, and contact pressure (normal value).

The provided Figure 2.5 displays a schematic diagram of the asperity friction model. In short, the model consists of a simple piecewise linear velocity-dependent approximate Coulomb friction element, that only includes two linear segments, in parallel with a variable anchor point spring.

2.4.2 Contact Point Search

There is a significant importance that goes along with contact detection in this work. This is due to the fact that contact detection will control when to apply surface interaction forces. Contact will be detected between the slave contact surface, or polygonal surface, and the master contact surface. The master contact points are localized on a rigid body where the slave contact surface is on a separate rigid body. A binary tree contact search algorithm may be implemented to perceive contact between the contact points of the mainpath and the rest paths. This algorithm provides fast contact searching [92].

2.4.3 Penalty Normal Contact

In order to impose the normal contact constraint, the penalty normal technique may be utilized.

Through the use of this method, a normal reaction force, (F_{normal}), may be created when a point enters a contact surface. The magnitude of this aforementioned normal

reaction force will be equivalent to the defined distance. Lastly, the normal reaction force will also be proportional to the velocity of penetration:

$$F_{normal} = Ak_p + A\begin{cases} c_p\dot{d}, & \dot{d} \geq 0 \\ s_p c_p \dot{d}, & \dot{d} < 0 \end{cases} \quad \text{(Equation 2-32)}$$

$$\dot{d} = v_i n_i \quad \text{(Equation 2-33)}$$

$$vn_i = \dot{d}n_i \quad \text{(Equation 2-34)}$$

Where:

A: The rectangular area(for each contact point)

$\vec{n}$: Unit vector normal to the surface

d: The minimum distance between the contact surface and the contact point

$\dot{d}$: the change rate of the value d respect to time

k_p: Penalty stiffness

c_p: Damping coefficient per unit area

s_p: Separation damping factor that establishesdetermines the sticking amount between the contact surface and the contact point at the same point , this factor is between 0 and 1.

$\vec{v_i}$: The velocity between the contact surface and the contact point (relative velocity)

The normal contact force vector can be written as:

$$F_{n_i} = n_i F_{normal} \quad \text{(Equation 2-35)}$$

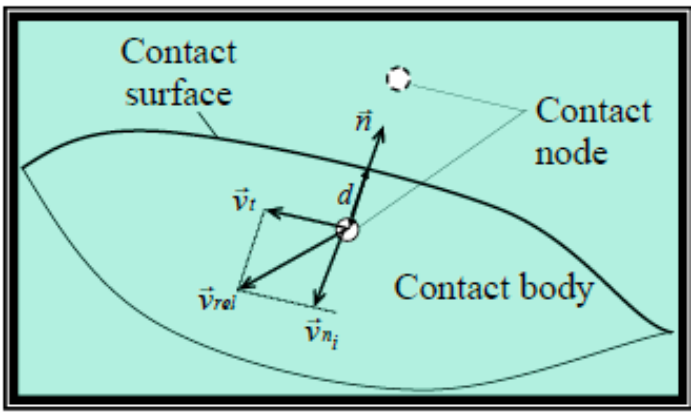

Fig. 2.6. Contact Node / Contact Body / Contact Surface [88, 91]

The total force on the node generated due to the frictional contact between the point and surface can be written as:

$$F_{point_i} = F_{t_i} + F_{n_i} \qquad \text{(Equation 2-36)}$$

2.5 Modeling Foot Pad Friction for a Full Human Body Model

In this research, modeling friction is used for modeling the contact friction. Furthermore, the model friction path has been modeled with regard to different types of contact surfaces. Contrary to other foot long models in literature where penalty-type formulations are used without experimental justification, this model offers the appropriate experimental justifications.

It is important to consider the lack of sufficient data on experiments regarding the friction and shear determination of a human heel pad. Analyzing ground reaction forces in virtual feet usually yield the findings of excessively high initial transient forces. Even with normal initial transient force readings, ground reaction force profiles are found to differ from experimental results.

Friction Model for a Full Human Body Model

Arguably the most important part of friction modeling is developing the contact dynamic characteristics. Typically, this process involves the development of equations between two asperities. This contact involves elastic, platstoc-elastic and plastic-plastic deformation.

Friction Parameters a Full Human Body Model

In this study, for the contact between foot and ground, although the coefficient of friction is function of relative velocity, is 0.7. The time multiplier for coefficient of friction and the normal pressure multiplier for coefficient of friction are 1.

The friction/contact model is defined as "friction asperity spring model", so the friction spring stiffness/unit area is $5e7$ and the velocity stiffness/unit area is $1e6$; the maximum asperity breaking factor is 0.975.

For normal contact parameters, the normal contact stiffness per unit area, the normal contact force per unit area, and the normal stiction force are 0. The normal contact damping per unit area is 8000. The separation damping factor is 1. The normal averaging factor is 0.75.

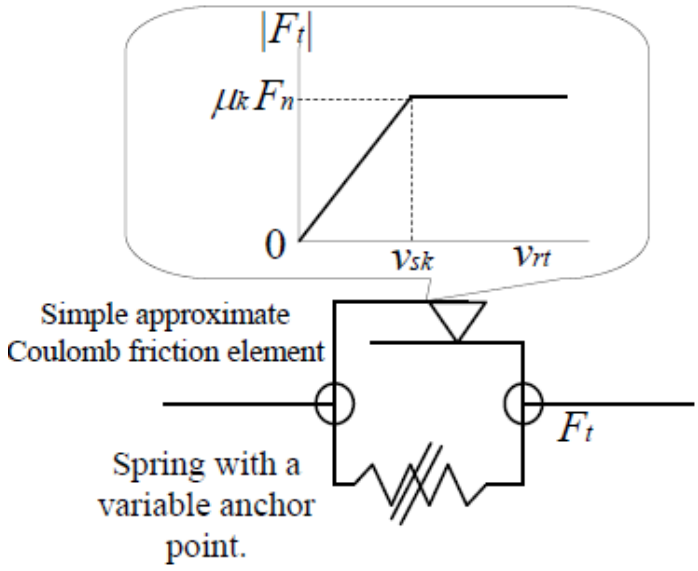

Fig. 2.7. Simple Approximate Coulomb Friction Element [93]

2.6 Explicit Solution Procedure

The model nodes are where the solution fields for the multi-body modeling systems are defined. It is important to mention that a rigid body will be modeled as a finite element node. Accordingly, these solutions include the following global translational vectors:

$X^t_{K_i}$: Position

$\dot{X}^t_{K_i}$: Velocity

$\ddot{X}^t_{K_i}$: Acceleration

Additionally, these solution fields include rotational vectors and matrices, where:

$\overset{t}{}_{K_i}$: Local body velocity

$\overset{t}{}_{K_i}$: Local body acceleration

$R^t_{K_{ij}}$: Global to local rotation matrices

Thus, the following needs to be stated:

$K = 1$ to N.

N : total number of rigid bodies.

The mentioned response quantities' time can be predicted though the use of an explicit time integration. A benefit of utilizing this procedure is that it is very parallel. The following process describes how to achieve approximately linearly growwith total of processors on shared memory of the parallel computers. Lastly, this process is executed in the DIS [94] (Dynamic Interactions Simulator) commercial software code.

1- Formulate test:

   a. For solution fields (described above), set initial conditions.

   b. Including master contact surfaces, itemize all finite elements.

c. Itemize elements to be run on each processor (use an algorithm for equal computational involves each processor).

d. Itemize all represented motion constraints.

e. Calculate all finite element nodes's solid masses of (Note: masses are constant).

f. Locate minimum time step (loop over all the elements).

2- Loop over the solution time (Note: Time increment = Δt):

a. Set the values of all nodal at the last time step = consider for all solution fields the current values of nodal .

b. Conduct a predictor iteration and a corrector iteration:

i Set moments of inertia = 0 and nodal forces = 0 (with a corrector iteration).

ii Execute contact search algorithm.

iii Calculate the moments and the nodal forces.

1. Loop through all the elements while calculating and collecting the forces of the element nodal.

2. Run each elements list from part 1.c on one processor.

iv Locate nodal values at present time step (use the equations of motion (semi-discrete) and integration rule of trapezoidal time).

v Implement the represented motion constraints (nodal values = prescribed values).

vi Loop back to the step 2(skip step 1) [89, 93, 95]

3. DESCRIPTION OF THE MULTIBODY HUMAN BODY MODEL

This chapter describes the multibody model of the human body. It is possible to use a system of bodies in order to systematically model the complex dynamics of a human body. Each member of the set may undergo translation or rotation in the three dimensional space. Momentum as well as the kinematics of the members may be used in order to formalize a system of equations that describe a complex system.

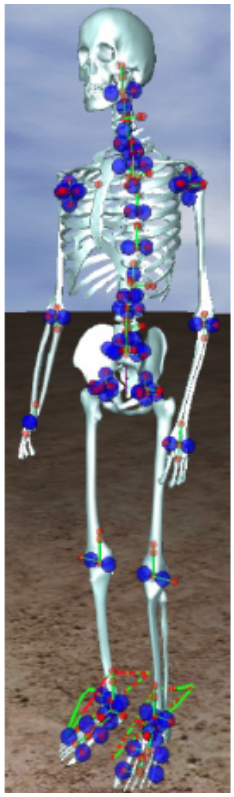

Fig. 3.1. Full Human Body Modeled in DIS

It is possible to divide multimode human body models into complex and simple models. Regardless of the pick, replication of the human-like gait remains to be

extremely essential. It is also important for the modeling technique to be able to capture human walking locomotion dynamics as well as synchronization.

Numerous approaches exist to model the bipedal walking motion, like human walking, for example. Selecting the appropriate modeling strategy requires one to fully understand the advantages and shortcomings of each considered model. Walking motion is a remarkable type of the multibody dynamic system with the existence of friction and slipping. All these occurrences detected during the foot ground contact should be presented into the realistic model to. In order to mathematically determine the foot-ground, interaction forces have to be taken into account. Furthermore, in order to account for forces, both material properties and contact kinematics of structures are used as inputs to the modeling strategy. Various geometries could be used in order to study the aforementioned interaction. Point-plane elements are popular due to their simplicity.

This research develops a three-dimensional (3D) model of a full human body.

The simplified multibody dynamic of a human model in this study is characterized by ten main kinematic chains with each of the chains is consisting of a one or more rigid bodues:

- Right Leg,
- Left Leg,
- Pelvis,
- Human Back,
- Right Hand,
- Left Hand,
- Ribs,
- Scapula,

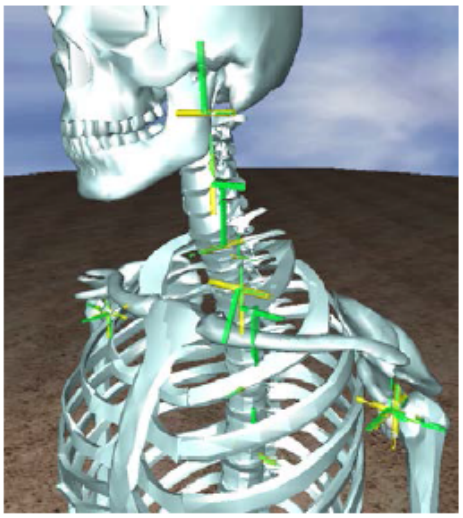

Fig. 3.2. Joints of Neck and Head of Full Human Body Model

- Neck and Head.

Each body is a group of multiple objects. The right and left leg each have different bodies including:

- Foot Toes
- Foot Middle PRT

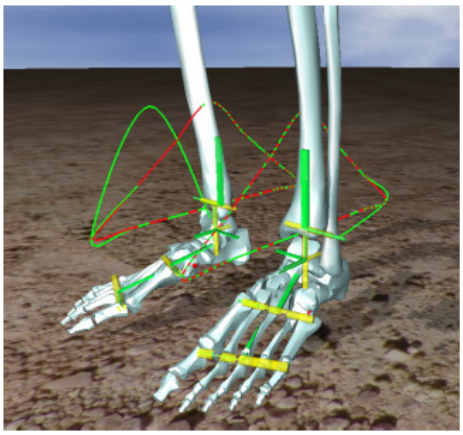

Fig. 3.3. Foot Modeled in DIS

- Tibia
- Femur-Thigh
- 2 Rigid bodies to connect the leg to pelvis
- Ankle(3 rigid bodies)

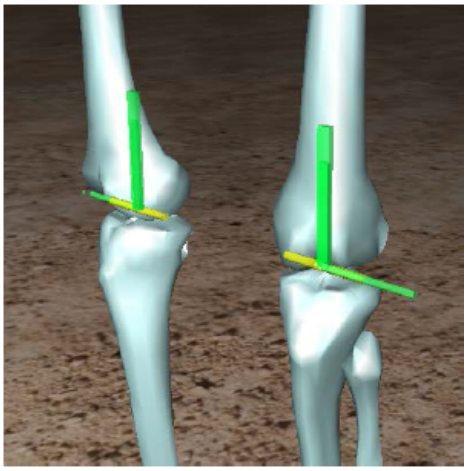

Fig. 3.4. Presenting Knee Model of Full Human Body Model

Also, the back includes all lumbar vertebrae and thoracic vertebrae bones, but to simplify the full human body model and reduce the number of DOF, a group of three thoracic vertebrae bones represents as one rigid body in this model.

On the other hand, to have flexible movement and multiple functionalities such as running, supporting for sit-to-stand, and other motions, this model specified one rigid body for each lumbar vertebrae:

- Thoracic_Vertebrae 1, 2, 3
- Thoracic_Vertebrae 4, 5, 6
- Thoracic_Vertebrae 7, 8, 9
- Thoracic_Vertebrae 10, 11, 12

- Lumbar_Vertebrae 1
- Lumbar_Vertebrae 2
- Lumbar_Vertebrae 3
- Lumbar_Vertebrae 4
- Lumbar_Vertebrae 5
- Lumbar_Vertebrae 6

One of the control strategies to balance the walking motion is utilizing both hands in case of a lack of balance; while a human is walking, back and forth alternating movements of hands help in regard to tuning the center of body mass. The right and left hands have multiple rigid bodies, they include:

- Hand and fingers all as one rigid body
- UlnaRadius_Forearm
- Humor_Upper_Arm
- Two Rigid bodies to connect the hand to the shoulder

The neck is made of two rigid bodies:

- Cervical_Vertebrae 3 ,4 Axis
- Cervical_Vertebrae 5, 6, 7

The pelvis, head and scapula each one are a single rigid body; also all of the ribs are represented by one single rigid body in this model.

In this type of model, the friction and normal forces between both the feet and the ground are modeled to have a more realistic simulation. Although control of this model is more difficult compared to the nonslip walking motion model, since the control model duty is to prevent the human body model from slipping and the

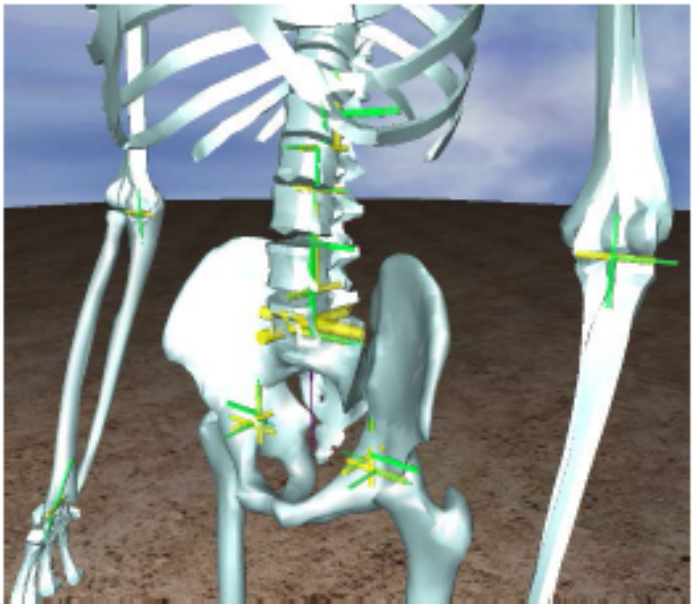

Fig. 3.5. Pelvis

model has to engage the reaction produced by impacts. The model studied in this research is three dimensional and demonstrates the kinematic configuration and weight properties of the human full body model.

The initial step in this study was to develop the design of the full human body in the multibody dynamic software. This exercise in design included an iterative process of developing a concept, rening the details, and deciding on a nalized design. The reason for developing a new geometric design was to demonstrate the entire development process of the humanoid robotic system and its control system.

To have a more realistic and natural model, all of the mass properties and bone sizes are based on a body of a $30 - 40$ year old adult male. Each bone and its mass properties are measured from "Weight, Volume, and Center of Mass of Segment of the Human Body" medical study by Charles E. Clauser, et al [96].

3.1 Design Goal

The main goal of this study is to design a full human body and model it similar as possible to a human being, considering all bones shapes, weights and sizes. The next step is to provide a different balance control strategy for walking motion.

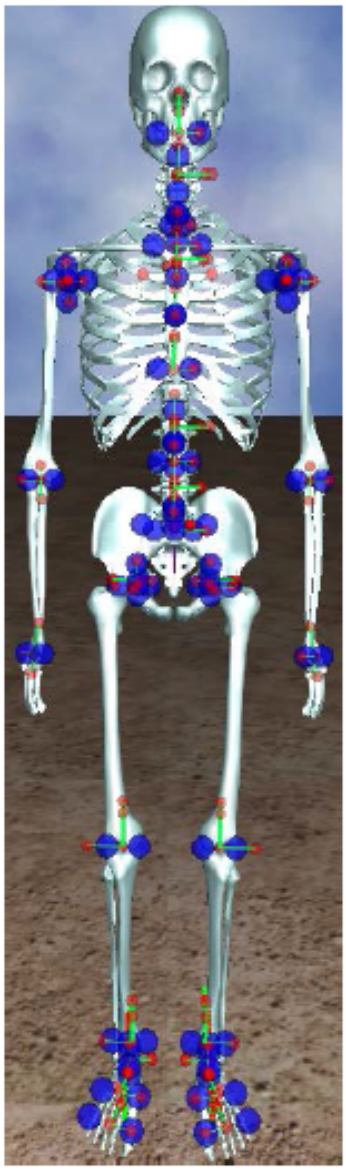

Fig. 3.6. Model of a Full Human Body for Simulating Walking

3.2 Topology of the Model

As formerly mentioned, this model contains numerous anatomical segments. The linkage between the segments is through ideal rotational joints, described by a 42 degree of freedom (DOF) model.

It is not an easy task to describe the contact surfaces of a real human joint. This problem arises due to complex shapes as well as changing characteristics of the contact point while the body is in motion. To explain further, the adjacent segments change contact points resulting in an appearance of relative translations between them. To avoid the complications that result from relative translations and further to have a fixed center of rotation, it is beneficial to model body joints as ideal joints. This simplification is especially helpful for large motions, like the ones present during gait. In this research study all the used joints are modeled as revolute joints.

In order to provide a three degree of freedom in joints, such as wrists, ankles and shoulders, three rotational joints are modeled in three different directions (axis). Consider the connection between the shoulder and arm. Three rotational joints could be utilized in order to provide three degrees of freedom for the joint. This action would essentially replicate the joint under study as it functions in the human body.

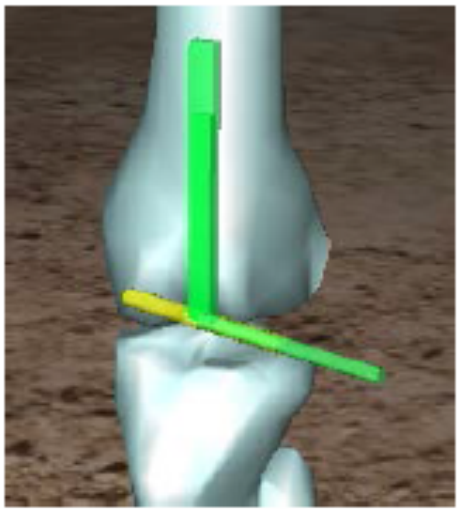

Fig. 3.7. Rotational Joints Applied in Knee

3.3 Kinematical and Mass Properties

This section describes how this research models the volume, mass, weight, and COG (center of mass) of each parts of the human body.

Knowledge of these variables is important when conducting research in such diverse fields, such as space technology and robotics. While the specific information needed may vary from one specialty to another, common to all is the objective of understanding more fully the biomechanics of man either as an entity or as a component of some complex system; and this helps to get realistic results. It is important to tune the simulation model as accurate as possible in this research study.

In this model, the toes have 250 grams, the middle section of foot has 410 grams each, and the pelvis as one rigid body has 7.56 kilograms. The Ankle includes three different rigid bodies; each one has equally 200 grams. The hands of the model including the fingers weights 400 grams.

The tibia is 3.08 kilograms, ribs as a single rigid body is 10 kilograms, and the vert is 2.58 Kg.

3.3.1 Human Body Measurements (Anthropometry)

There exists some anthropometric measurements in order to estimate the joint positions and the body segment mass, volume, and height parameters. The inertial properties of the segments are taken, for lower limbs, from a reduced set of measurements extracted on the subject and by scaling issued data according to his weight and height [97].

The model is selected to perform the experiments is an adult male, mass $70kg$ and height $1.75m.$, the errors in the estimation of body parts parameters are expected to be low.

To model all body bones and segments which are in different shapes and masses, this study simplifies the bones into two different shape categories: cylinder and rectangular prism (cuboid).

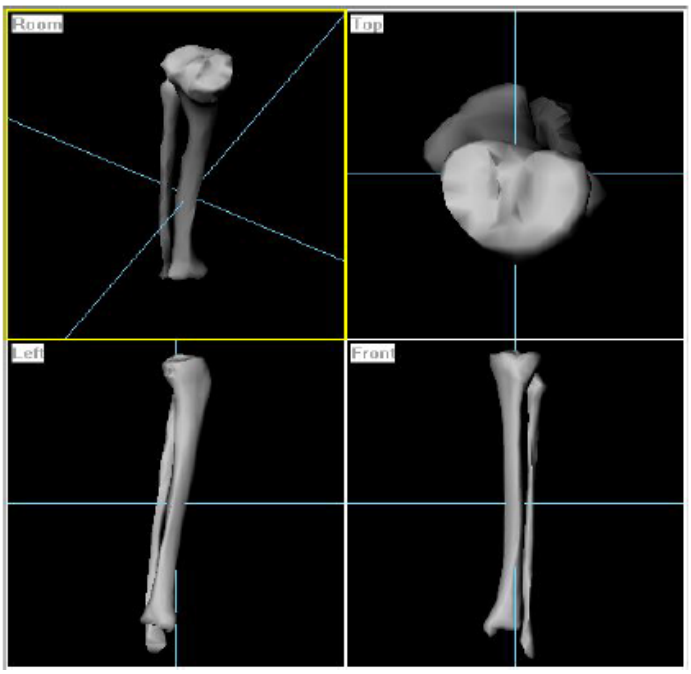

Fig. 3.8. Top, Left, Front and 3D View of Right Tibia Bone Model

Some of the bones are modeled as a circular cylinder with height h and radius r.

And other bones are modeled as a rectangular prism with height h, depth d, and width w; including the foot toes, the middle part of each foot, hands (with a low amount of depth), and pelvis.

Table 3.1.
Mass and Inertia of Rectangular Prism Rigid Bodies

Rigid Body	h(cm)	d(cm)	w(cm)	M(kg)	$I_h(kg.m^2)$ * 10^{-4}	$I_w(kg.m^2)$ * 10^{-4}	$I_d(kg.m^2)$ * 10^{-4}
FootToes	3.5185	6.24232	10.0962	0.25	2.94	1.07	2.38
FootMiddle	6.36254	8.89482	9.5017	0.41	5.79	4.09	4.47
Hand	9.6334	7.88498	4.0807	0.4	2.63	5.17	3.65
Pelvis	21.1809	1.7291	32.2739	7.56	844.57	470.99	938.85

Where moments of inertia for rectangular prism rigid bodies are calculated by:

$$I_h = \frac{1}{2}m(w^2 + d^2), \quad I_w = \frac{1}{2}m(h^2 + d^2), \quad I_d = \frac{1}{2}m(w^2 + h^2), \qquad \text{(Equation 3-1)}$$

Table 3.2.
Mass and Inertia of Rigid Bodies in Shape of Cylinder

Rigid Body	r(cm)	L(cm)	m(kg)	$I_r(kg.m^2)$ * 10^{-4}	$I_l(kg.m^2)$ * 10^{-4}
Anke	4.79659	12.771	0.2	2.30073	0.71831
Anke_Part1	2.85715	05.03935	0.2	0.816	0.423
Anke_Part2	2.85717	05.03935	0.2	0.816	0.423
Tibia	5.46879	43.9767	3.08	46.0578	0496.38054
Femur_Thigh	7.21932	47.4971	7.84	204.30484	1473.90335
Humor_upper_arm	4.99598	35.071	2.24	27.95499	229.59534
UlnaRadius_Forearm	4.53482	29.2366	1.26	12.95569	89.75177
Vert	5.0	12.0	2.58	32.25	30.96
Ribs	18.0	30.0	10	1620.0	750.0

Where the moment of inertia is derived by:

$$I_r = \frac{1}{2}mr^2, \quad I_l = \frac{1}{2}ml^2 \qquad \text{(Equation 3-2)}$$

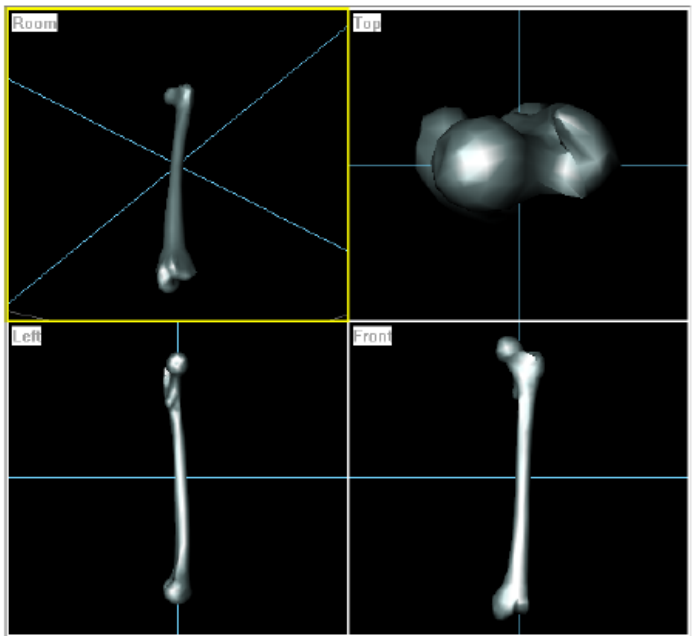

Fig. 3.9. Top, Left, Front and 3D View of Left Femur Bone Model

4. WALKING CONTROL

This chapter reviews the walking control strategies and parameters used in the PID controller for a 3D full human body stable walking functionality.

4.1 Inverse Kinematics

When a kinematic model of a robot is hard to achieve, inverse kinematics is a solution. In the robotics field, the kinematic equations are used to define parameters of each joint in the system with the goal of delivering the desired position for the robot's end-effectors, the segment of the robot which interacts with the environment.

$$\{\theta\} = \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_N \end{Bmatrix} \qquad \text{(Equation 4-1)}$$

Kinematic chain joint angles of the robotic systems are:

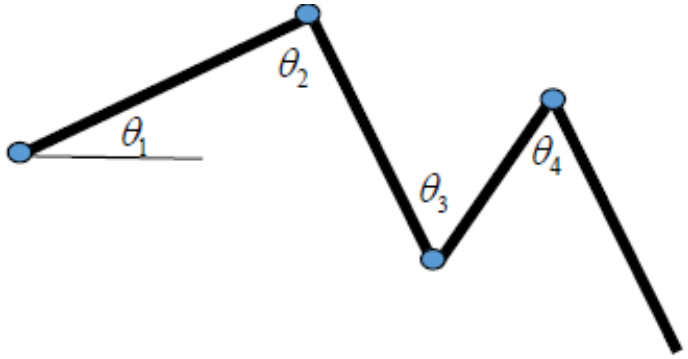

Fig. 4.1. Kinematic Chain Joint Angles

The end-effector homogeneous transformation matrix is called $H(\{\theta\})$ and it demonstrates mapping a position vector from one coordinate to another coordinate of the system with the format of

$$ {}^{j}H_i(\{\theta\}) = \begin{bmatrix} {}^{j}R_i(\{\theta\}) & {}^{j}p_i \\ 0\ 0\ 0 & 1 \end{bmatrix} \qquad \text{(Equation 4-2)} $$

to transfer from position i to j where $R(\{\theta\})$ is the end-effector rotation matrix. Also, the desired parameters are:

$\vec{X}_d$: Desired end-effector point

$\vec{N}_d$, $\vec{T}_d$: Desired end-effector normal and tangential vectors

And the desired end-effector Euler-angle is the function of both the desired end-effector normal and tangential vectors $\vec{a}_d = f(\vec{N}_d, \vec{T}_d)$.

And the target parameters are as follows:

$\vec{x}_p$: End-effector target point in the local end-effector frame

$\vec{n}_p$, $\vec{t}_p$: End-effector target normal and tangential vectors in the local end-effector frame

From forward kinematics:

$\vec{X}_p(\{\theta\}) = [H(\{\theta\})]\vec{x}_p$: End-effector target point in the global frame

$\vec{N}_p(\{\theta\}) = [R(\{\theta\})]\vec{n}_p$: End-effector target normal vectors

$\vec{T}_p(\{\theta\}) = [R(\{\theta\})]\vec{t}_p$: End-effector target tangential vectors

$\vec{\alpha}_p(\{\theta\}) = f(\vec{N}_p, \vec{T}_p)$: End-effector Euler-angles

Given $\vec{X}_d$ and $\vec{\alpha}_d$ to calculate $\{\theta_d\}$.

$$Let \begin{cases} \vec{g}_1(\{\theta_d\}) = \vec{X}_p(\{\theta_d\}) - \vec{X}_d \\ \vec{g}_2(\{\theta\}) = \vec{\alpha}_p(\{\theta\}) - \vec{\alpha}_d \end{cases} \quad then\ G(\{\theta_d\}) = \begin{Bmatrix} \vec{g}_1(\{\theta_d\}) \\ \vec{g}_2(\{\theta_d\}) \end{Bmatrix} = \begin{Bmatrix} \vec{g}_{11}(\{\theta_d\}) \\ \vec{g}_{12}(\{\theta_d\}) \\ \vec{g}_{13}(\{\theta_d\}) \\ \vec{g}_{21}(\{\theta_d\}) \\ \vec{g}_{22}(\{\theta_d\}) \\ \vec{g}_{23}(\{\theta_d\}) \end{Bmatrix}$$

(Equation 4-3)

So $\{\theta_d\}$ will be determined such that $G(\{\theta_d\}) = \Big\{0\Big\}$.

Newton's method

Considering Jacobian matrix $[J] = \frac{\partial G(\{\theta\})}{\partial\{\theta\}}$ which is a rectangular matrix, typically with more rows than columns.

To deal with the problem of more unknowns than equations, partial-least squares method is one of the solutions used in this study

$$E = \{r\}^T\{r\} \tag{Equation 4-4}$$

where $\{r\} = \{G(\{\theta\} + \{\Delta\theta\})\} = \{G(\{\theta\})\} + [J]\{\Delta\theta\}$ is residual.

$$[J]^T[J]\{\Delta\theta\} = -[J]^T\{G(\{\theta\})\} \tag{Equation 4-5}$$

To simplify the above equation in this format $[B]\{\Delta\theta\} = \{C\}$, matrix B defines as $[B] = [J]^T[J]$ and matrix C defines as $[C] = -[J]^T\{G(\{\theta\})\}$.

To obtain $\{\theta_d\}$ iteration keep continue until $\{r\} = \{0\}$

4.2 Walking Motion Path/Pattern

To have a realistic modeling for human walking, in this study has a variable for walking speed of human, which the default value is $1.40m/s$ ($5.00km/h$ or $3.10mph$), and this speed is a preferred speed of walking for human approved by many research

studies [98–100]. Although there are other factors impacts the walking speeds, like environmental elements, weather, surface texture and others.

The other element of waking motion which has been to be considered in modeling is walking pattern of human. In this research, two numerical analysis methods are applied to create a smooth continuous walking pattern include uniform quadratic/cubic B-Spline interpolation and Hermit cubic spline.

There are two main different patterns for backward and forward waling motion for both right and left legs.

4.2.1 Uniform Quadratic/Cubic B-Spline

In the mathematical subfield of numerical analysis, in order to generate the interpolation curve a series of low order polynomials interpolation functions are used instead of one high-order polynomial. Quadratic or cubic polynomials are the most common types used. This type of curve is called a "spline". Splines can be used to generate 3D or 2D curves.

In the B-spline algorithm every control point will be defined with a basis function.

$$point(u) = \sum_{i=1}^{n+1} N_{i,j}(u) point_i \quad while \quad u_{min} \leq u < u_{max} \qquad \text{(Equation 4-6)}$$

Each basis function, $N_{i,j}$, with degree of $j-1$ and order j, and the continues function of Point is associated with $n+1$ control points $Point_1$, $Point_2$, ..., $Point_{n+1}$.

The minimum value for k is 2 to create a linear equation, 3 for quadratic and 4 for cubic equation; and the maximum value is note more than the total of $n+1$ control points. However, for curves of Bezier j is same as $n+1$ control points. So, in this algorithm the order of the curve is not function of the number of control points.

This method is used to interpolate the desire waking diagram for each leg. For each step motion there a series of points: $(x_{p1}, y_{p1}, z_{p1} = 0)$, $(x_{p2}, y_{p2}, z_{p2} = 0)$,

$(x_{p3}, y_{p3}, z_{p3} = 0)$,..., $(x_{pn}, y_{pn}, z_{pn} = 0)$, which total is 102 points for right step and 117 points for left one by 12 control points.

Uniform Quadratic B-Spline Interpolation

For each successive 3 points , a curve is generated using:

$$x(u) = w_1(u)x_1 + w_2(u)x_2 + w_3(u)x_3 \qquad \text{(Equation 4-7)}$$

$$y(u) = w_1(u)y_1 + w_2(u)y_2 + w_3(u)y_3 \qquad \text{(Equation 4-8)}$$

$$z(u) = w_1(u)z_1 + w_2(u)z_2 + w_3(u)z_3 \qquad \text{(Equation 4-9)}$$

where u is a real number between 0 and 1. The weights w_i are given by:

$$w_1 = 0.5 - u + 0.5u^2 \qquad \text{(Equation 4-10)}$$

$$w_2 = 0.5 + u - u^2 \qquad \text{(Equation 4-11)}$$

$$w_3 = 0.5u^2 \qquad \text{(Equation 4-12)}$$

In matrix form the Equations 4-10,4-11, and 4-12:

$$\begin{Bmatrix} w_1 \\ w_2 \\ w_3 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u^2 \\ u^1 \\ 1 \end{Bmatrix} \qquad \text{(Equation 4-13)}$$

4.2.2 Hermit Cubic Spline

For each successive four points we generate a curve using:

$$x(u) = w_1(u)x_1 + w_2(u)x_2 + w_3(u)x_3 + w_4(u)x_4 \qquad \text{(Equation 4-14)}$$

$$y(u) = w_1(u)y_1 + w_2(u)y_2 + w_3(u)y_3 + w_4(u)y_4 \quad \text{(Equation 4-15)}$$

$$z(u) = w_1(u)z_1 + w_2(u)z_2 + w_3(u)z_3 + w_4(u)z_4 \quad \text{(Equation 4-16)}$$

the weights w_i are given by:

$$w_1 = \frac{1}{6}(1 - 3u + 3u^2 - u^3) \quad \text{(Equation 4-17)}$$

$$w_2 = \frac{1}{6}(4 - 6u^2 + 3u^3) \quad \text{(Equation 4-18)}$$

$$w_3 = \frac{1}{6}(1 + 3u + 3u^2 - 3u^3) \quad \text{(Equation 4-19)}$$

$$w_4 = \frac{1}{6}(u^3) \quad \text{(Equation 4-20)}$$

In matrix form the above equations can be written as:

$$\begin{Bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{Bmatrix} = \frac{1}{6} \begin{bmatrix} 11 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u^3 \\ u^2 \\ u^1 \\ 1 \end{Bmatrix} \quad \text{(Equation 4-21)}$$

Cubic Hermite splines numerical method is usually used for interpolation of numeric data definite in assumed values x_1, x_2,...,x_n with goal of an even unbroken output, like walking motion path.

4.2.3 Walking Motion Pattern

Studying on different simulation results and outcomes from the curves and patterns generated by both numerical methods as mention before, the Hermit cubic Spline is selected as the final numerical method to generate the walking motion for

this study. The Hermit Spine has better result in smooth motion since the curves actually pass through all the control points and better approach to control the height and length of the steps.

To simulate the human walking motion, one leg moves forward and another one moves backward. In the programming code (MATLAB in this study) multiple variables utilized to generate the walking motion curves by Hermit cubical spline numerical method.

Each step is triangle shape with step height H and step length L as variables, in this research the height of step is 20 centimeters and the length is 50 cm. for starting cycle of motion 5 cm offset is considered. Walking velocity is equal to 1.4 m/s or 140 cm per second. To calculate the weights of Hermit method sample rate t or (delta_t) is considered 0.01 in respect of t in this study, and the value is between 0 to 1.

The triangles are created by three lines:

$$\begin{cases} line_1: & y = 0; \\ line_2: & y = (2 * \frac{StepHeight}{StepLength})x; \\ line_3: & y = (-2 * \frac{StepHeight}{StepLength})x + 2 * StepHeight; \end{cases} \quad \text{(Equation 4-22)}$$

Both right and left legs start from same point, in this study the initial point is assumed (Length/2, -offset) which is $(25, -5)$.

As it is shown in the figures, the start of backward motion goes from (middle point of Step Length,-offset) to origin point $(0, 0)$ and the forward motion goes from initial point $(25, -5)$ to the apex of the triangle for this step (3/4 of Step Length ,Step Height).

Each walking path includes two significant cycles, the starting cycle and one steady state motion cycle following it.

Figure 4.1 and 4.3 demonstrates the combination of the starting motion and steady state motion into one curve.

Forward Step:

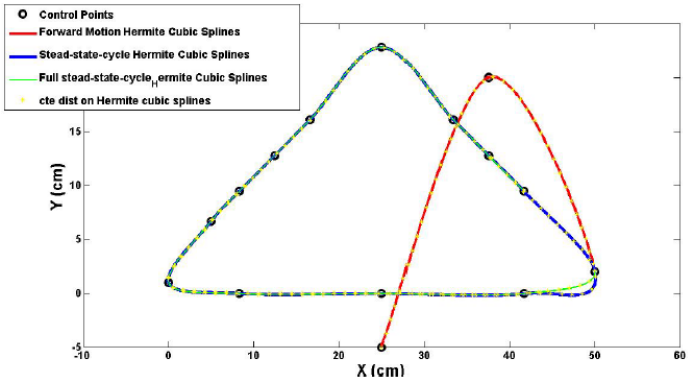

Fig. 4.2. Forward Step Figure with Control Points

The 12 control points for forward step are (Step Length/2, -offset), (3/4 of Step Length ,Step Height) and (Step Length , 0) for starting cycle and the following points for steady state cycle:

Table 4.1.
List of Control Points for Forward Step

#	$X_{ControlPoint}$	$Y_{ControlPoint}$	$Z_{ControlPoint}$
$ControlPoint_1$	$\frac{5}{6}$ of Step Lenght	0	0
$ControlPoint_2$	$\frac{1}{2}$ of Step Lenght	0	0
$ControlPoint_3$	$\frac{1}{6}$ of Step Lenght	0	0
$ControlPoint_4$	0	epsilon	0
$ControlPoint_5$	$\frac{1}{6}$ of Step Lenght	$\frac{1}{3}$ of Step Height+C	0
$ControlPoint_6$	$\frac{1}{4}$ of Step Lenght	$\frac{1}{2}$ of Step Height+C	0
$ControlPoint_7$	$\frac{1}{3}$ of Step Lenght	$\frac{2}{3}$ of Step Height+C	0
$ControlPoint_8$	Step Lenght	Step Height+C	0
$ControlPoint_9$	$\frac{1}{2}$ of Step Lenght	$\frac{2}{3}$ of Step Height+C	0
$ControlPoint_10$	$\frac{2}{3}$ of Step Lenght	$\frac{1}{2}$ of Step Height+C	0

Where $C = \frac{Step\ \ Lengh}{18}$.

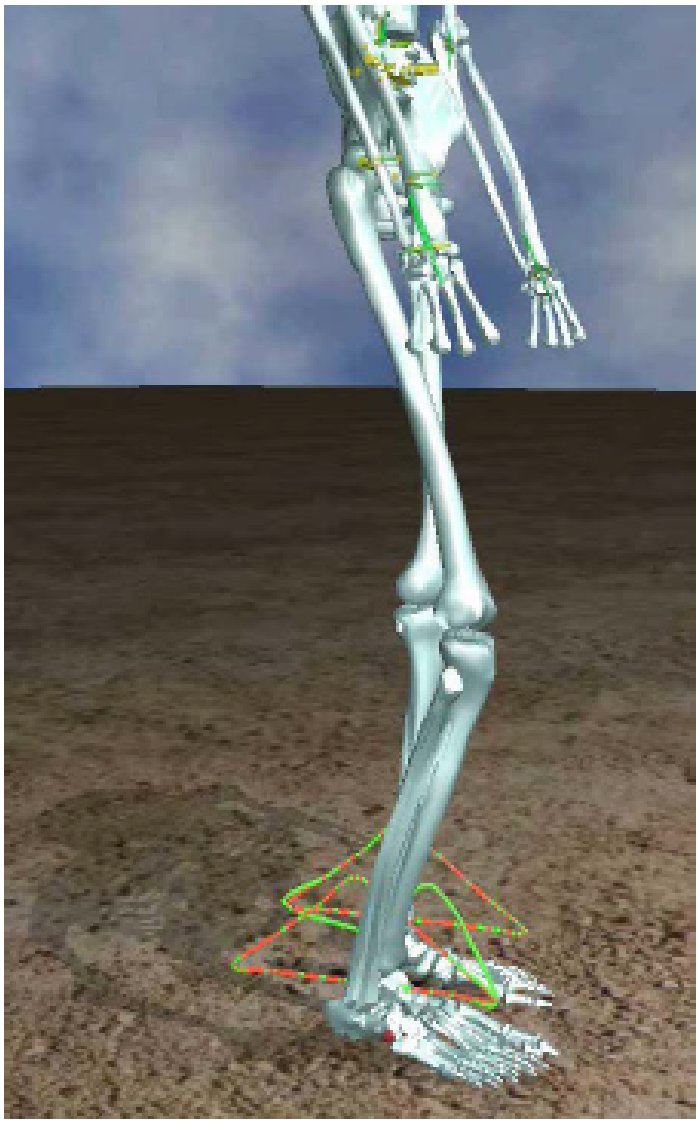

Fig. 4.3. Forward Step (Start)

Also, the time for the forward step from initial point to apex point and then (Step Length ,0) is the same as the time for the backward step from initial point to origin point $(0, 0)$, and the speed is constant for generation the forward step.

Backward Step:

The 12 control points for backward step are (Step Length/2, -offset) and $(0, 0)$ for starting cycle and the following points for steady state cycle:

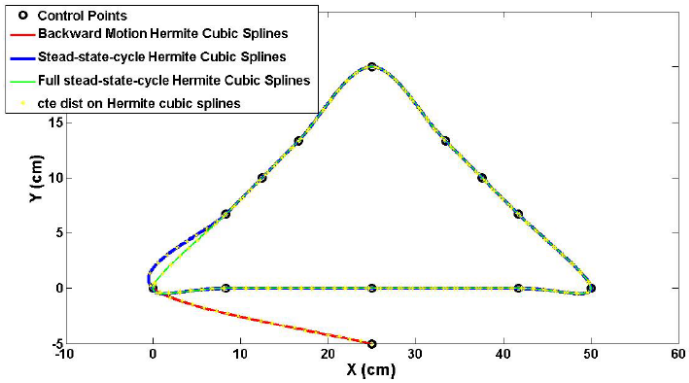

Fig. 4.4. Backward Step Figure with Control Points

Table 4.2.
List of Control Points for Backward Step

#	$X_{ControlPoint}$	$Y_{ControlPoint}$	$)Z_{ControlPoint}$
$ControlPoint_1$	$\frac{1}{6}$ of Step Lenght	$\frac{1}{3}$ of Step Height	0
$ControlPoint_2$	$\frac{1}{4}$ of Step Lenght	$\frac{1}{2}$ of Step Height	0
$ControlPoint_3$	$\frac{1}{3}$ of Step Lenght	$\frac{2}{3}$ of Step Height	0
$ControlPoint_4$	$\frac{1}{2}$ of Step Lenght	1 of Step Height	0
$ControlPoint_5$	$\frac{2}{3}$ of Step Lenght	$\frac{2}{3}$ of Step Height	0
$ControlPoint_6$	$\frac{3}{4}$ of Step Lenght	$\frac{1}{2}$ of Step Height	0
$ControlPoint_7$	$\frac{5}{6}$ of Step Lenght	$\frac{1}{3}$ of Step Height	0
$ControlPoint_8$	1 Step Lenght	0	0
$ControlPoint_9$	$\frac{5}{6}$ Step Lenght	0	0
$ControlPoint_10$	$\frac{1}{2}$ Step Lenght	0	0

The backward step from starting cycle is shown in figure below:

Set desired foot motion based on gait:

Subsequently, the list of coordinates of 225 points for forward step and 206 points for backward step which include set of x, y and z coordinates, tangential vector (with 3 coordinates), and normal vector of each point are applied as input of simulation

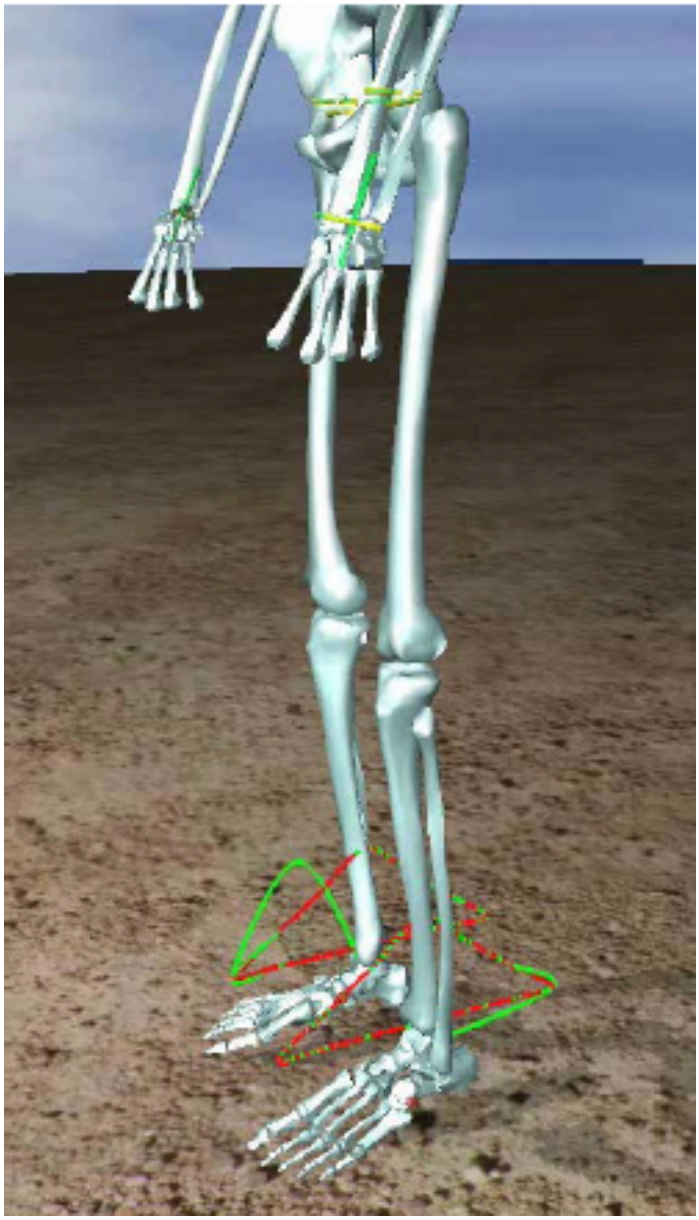

Fig. 4.5. Backward Step (Start)

software to create a point by point walking pattern for the full human body model with following assumptions:

- Hermite splines are used to trace the path of each foot.
- Foot target point is on the bottom of the foot.

- The foot orientation is kept constant.
- Velocity linearly ramped up from zero.
- Then, constant velocity is used along each path.

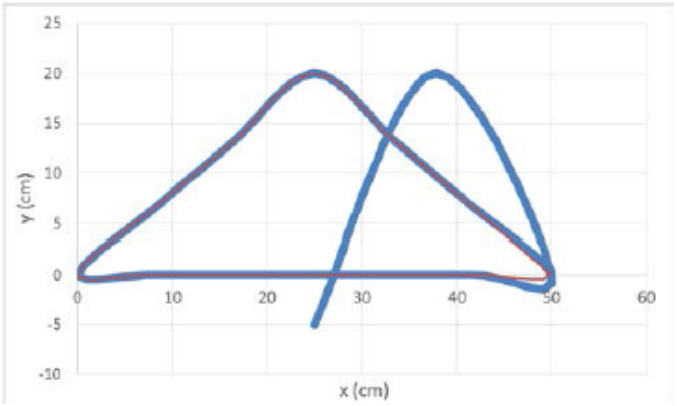

Fig. 4.6. Forward Step Plot

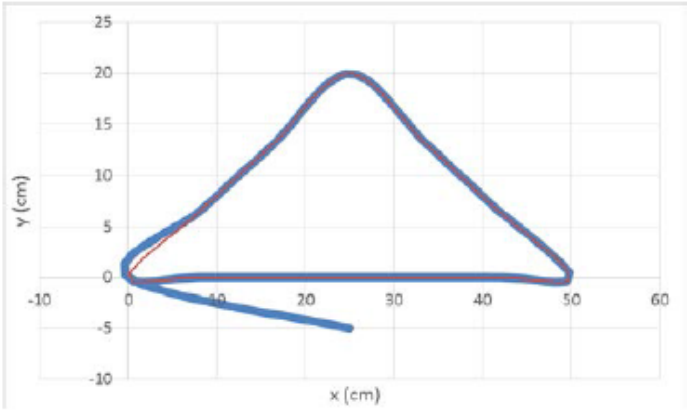

Fig. 4.7. Backward Step Plot

4.3 PD Joint Angle Control

In this study, the classical controller with single input and single output which has been used to control the joints is the proportional derivative controller known as PD

controller with using classical single input-single output (S.I.S.O.) design techniques. The input and output of the PD controllers are expressed in the local coordinate of each joint.

In this simulation environment PD controllers are designed by tuning the controller gains and simulating the system response. The walking motion controller is a finite state machine. Beside of the using the proportional derivative (PD) controllers for each joint in each state, there is a balance feedback controller. The PD controller for each revolute joint includes the proportional coefficient k_p, the damping coefficient k_d, and a target angle θ_d.

The torque τ for each revolute joint is generated by the joint angle θ (input) and the angular velocity $\dot{\theta}$:

$$\tau = k_p(\theta_d - \theta) + k_d\dot{\theta} \qquad \text{(Equation 4-23)}$$

The initial value of the PD controllers (proportional gain k_p and derivative gain k_d) and initial simulation parameters are assigned for each joint/actuator in simulation software.

For instance, for right ankle after initializing the mass and position of the body, the point joints and the actuator are assigned. To design the actuator, the Center point of the actuator, the axis point, and point 1(a point on the body which is right ankle in this example) and point 2 (a point on connecting body) should be initialized.

The controller type is "Angle" since all joints in this study are revolute joint. The joint angle θ unit is radian and the angular velocity $\dot{\theta}$ unit is radian per second. For this ankle joint, the proportional gain k_p is 10000 and the derivative gain k_d is 10. (with no integral gain for the controllers).

Therefore, the overall controlling strategy of this study is first to implement the forward dynamic of described model, all bodies coordinates and velocities based on external forces; then setting the desired foot motion based on gait gained from Hermit cubic spline numerical method. Next step is the first level of balance control to modify

Fig. 4.8. PD Controller for Right Ankle Joint

foot motion to achieve desired ground contact forces and then inverse kinematics to predict new joint angles to satisfy desired end-effector.

Next level of balance control is to control the upper body orientation to achieve balance. In this control strategy the COG (center of gravity) of body is calculated. If the orientation of body (the horizontal components of the COG), a vector from middle point of two feet and body COG is not normal to the ground, the balance control method apply different gains to push the body back to opposite direction to minimize the difference angle between normal vector to ground and body orientation which is α.

The last step of the iteration is PD control (calculating the joint torque (inverse dynamics)).

Balance controller is not part of this research study and it is suggested for future work of this research.

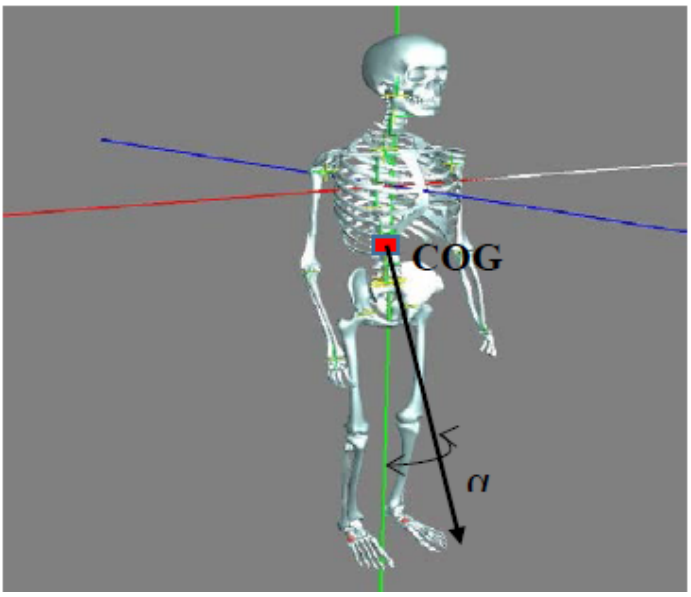

Fig. 4.9. Balance Controller Using COG of the Body

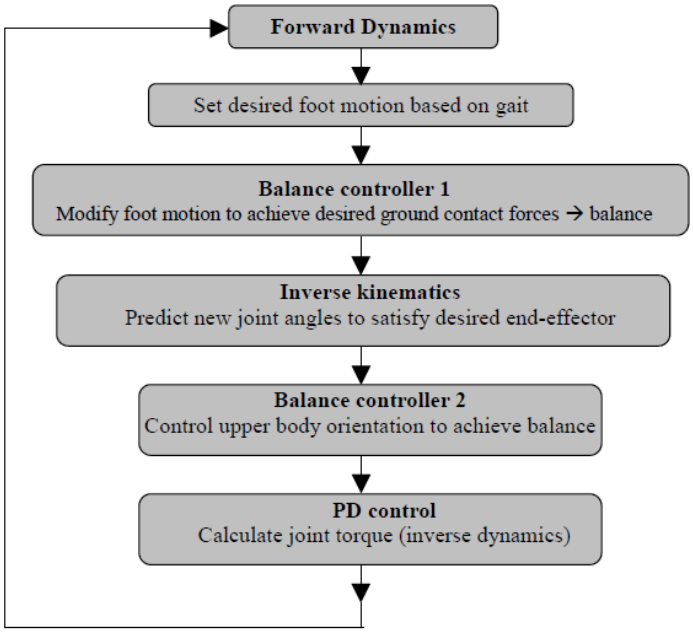

Fig. 4.10. Controlling Strategy Flowchart

5. SIMULATION RESULTS

In this project the model is designed and simulated in the DIS [7] simulation software,. Couple of methods are considered to control the stability and walking motion of the model. The functional sensitivity of the full human body model (multibody) is implemented to balance the control algorithms. Also, different PD controller gains were studied.

Fig. 5.1. Simulation Result

To validate the model, different external force objects from environment have been applied to the model to check the stability of the model with respect to different tests. The PD controller gains for all actuators have been tuned to balance the critical joints such as ankle-joint, knee-joint, and femur to pelvic joint.

The balance controllers are designed to make the model as robust as possible. The balance of the model is achieved by applying two phases of balance control strategies. First, the foot motion is modified to achieve desired ground contact forces. Second, the upper body orientation is controlled to achieve a good balance. Here considering the body center of mass and inertia, the human's shoulder and hands movements

are considered and the limitation of all bones rolling angle and movement has been taken into account. Based on the model response to different speed input, balance controllers are set.

Some simplifications in walking parameters have been made. The mass distributions of the bones are assumed uniform and symmetric. For inertia calculation, all bone shapes are simplified as cylindrical or rectangular prism shape. For example, the thigh bone (femur) is simplified as a rectangular prism with small width and height value and the depth of the actual bone.

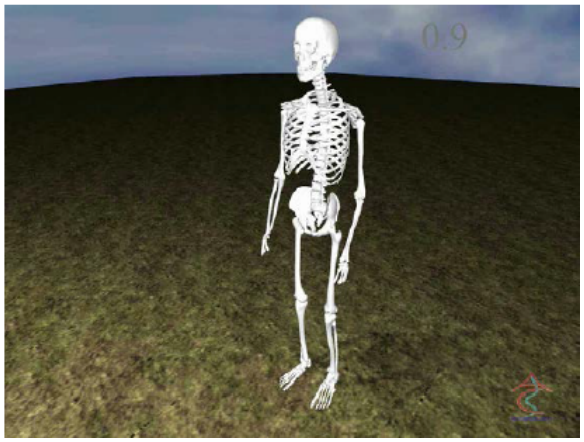

Fig. 5.2. Full Human Body Model in DIS - Front View

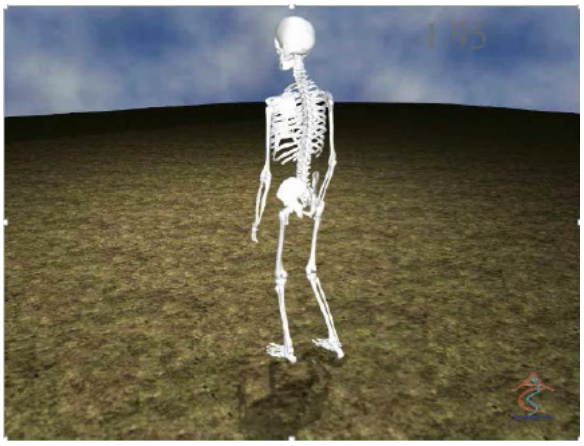

Fig. 5.3. Full Human Body Model in DIS - Back View

To test robustness and stability of the controllers multiple rounds of simulation have been performed.

The full human body model has been running in DIS, and figures below (Figures 5.4 and 5.5) show second by second of forward step motion:

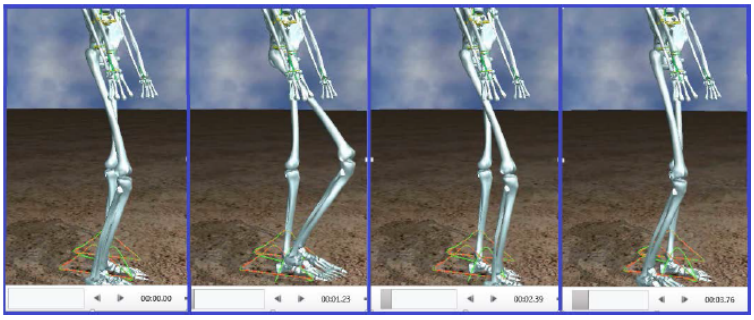

Fig. 5.4. Forward Step Motion (Start) - Part 1

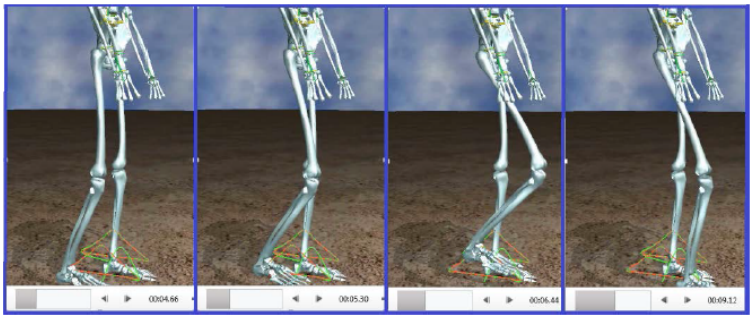

Fig. 5.5. Forward Step Motion (Start) - Part 2

Also, Figures 5.6 and 5.7 show the backward step motion, second by second: Different bones motion listed below are studied in the model walking motion:

- RightLegAnkleRollMotion
- RightLegFemurSideMotion
- LeftLegAnkleTwistMotion
- LeftLegAnkleRollMotion

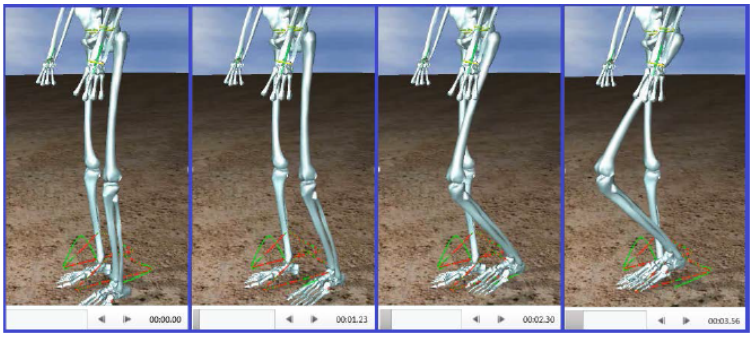

Fig. 5.6. Backward Step Motion (Start) - Part 1

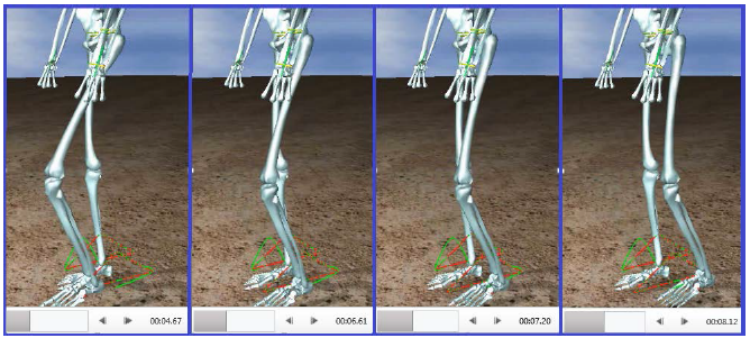

Fig. 5.7. Backward Step Motion (Start) - Part 2

- LeftLegFemurSideMotion
- RightLegFemurMotion
- RightLegTibiaMotion
- RightLegAnkleMotion
- LeftLegFemurMotion
- LeftLegTibiaMotion
- LeftLegAnkleMotion

The model in this research tracks the defined trajectory and walks with full stability.

Figure 5.8 demonstrates the knee angle and desired angle and Figure 5.9 shows the right knee torque in 4.0 second run:

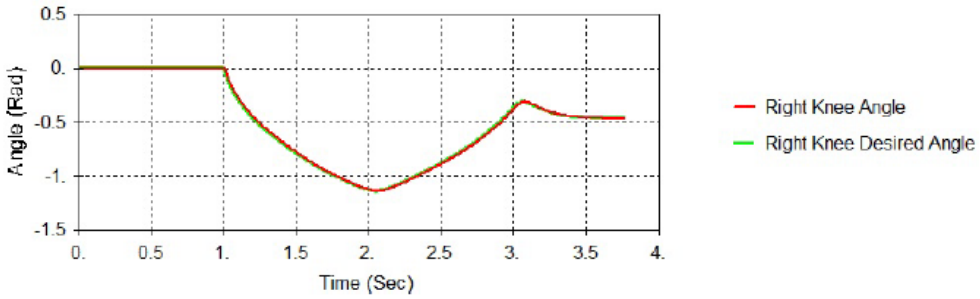

Fig. 5.8. Right Knee Angle and Desired Angle via Time (4 seconds)

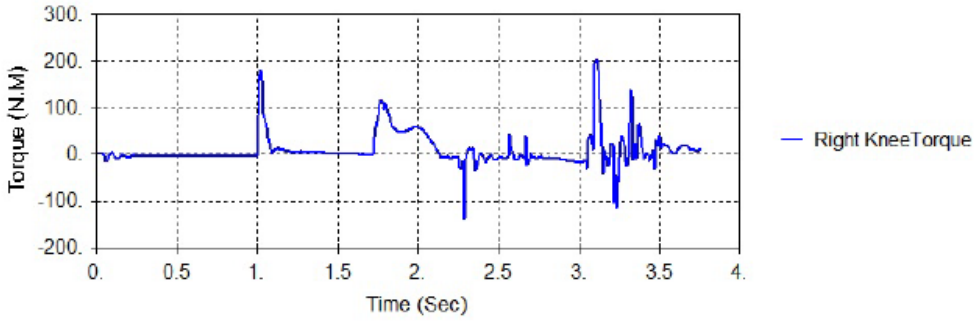

Fig. 5.9. Right Knee Torque via Time (4 seconds)

Also, Figure 5.10 represents right ankle angle and desires angle via time in 4.0 seconds:

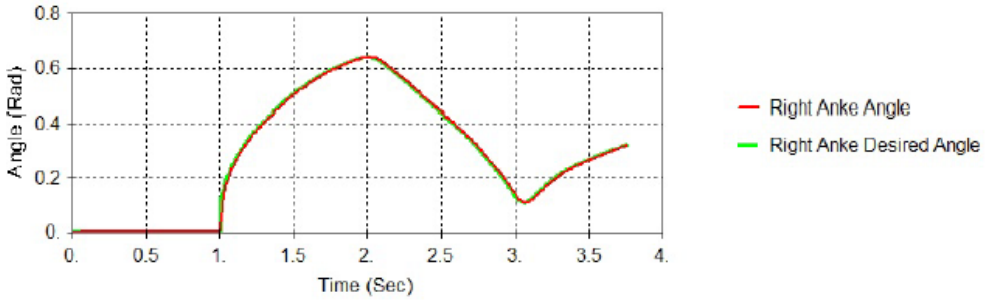

Fig. 5.10. Right Ankle Angle and Desired Angle via Time (4 seconds)

In Figure 5.11, the right ankle torque via time is shown:

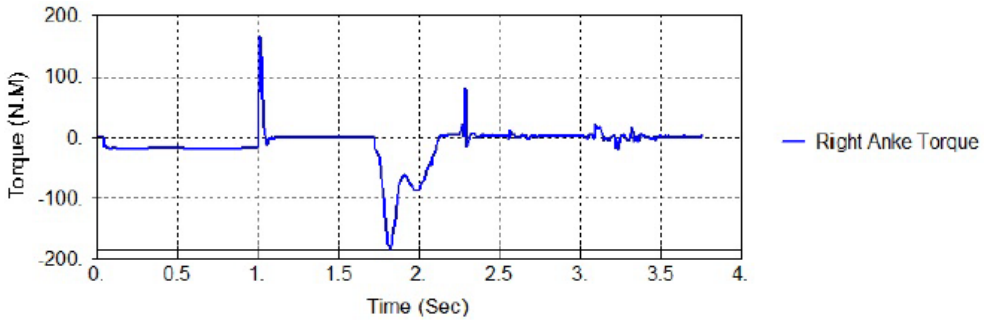

Fig. 5.11. Right Ankle Torque via Time (4 seconds)

Figures below [Figures 5.12 - 5.17] show the time-histories of right and left foot positions for 15 seconds simulation time:

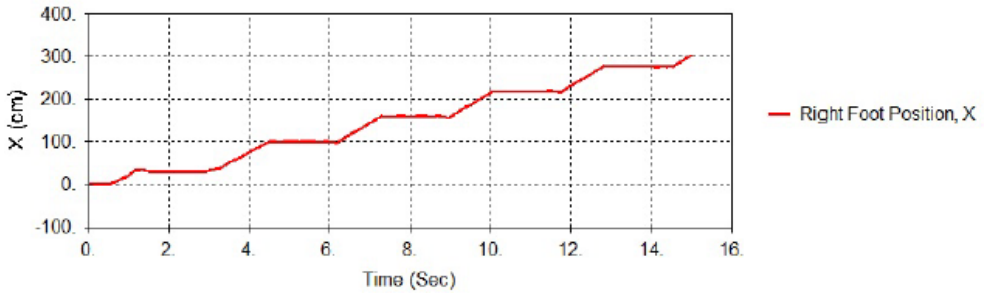

Fig. 5.12. Right Foot Position, X Value

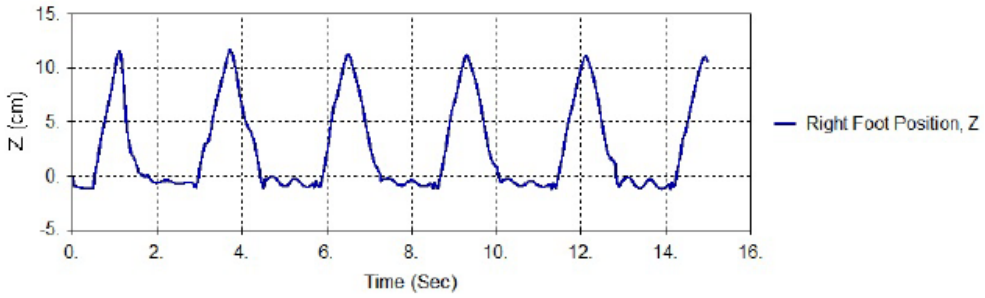

Fig. 5.13. Right Foot Position, Z Value

Fig. 5.14. Right Foot Position, Y value

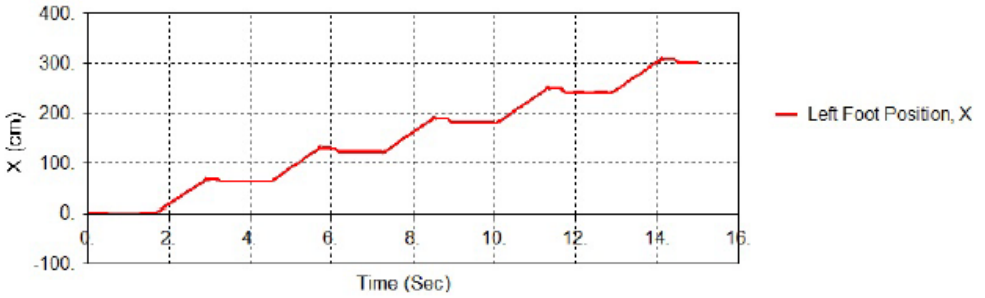

Fig. 5.15. Left Foot Position, X Value

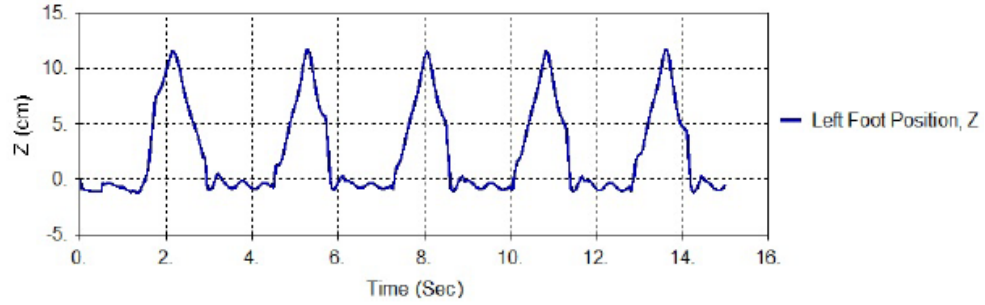

Fig. 5.16. Left Foot Position, Z Value

As shown in the figures, model is walking toward the X axis and keeps the step height (Z axis) about 12 cm; to stay in the line toward the X axis the model does not move in Y axis and the threshold [Figure 5.17] is 0.5 cm.

Also, time-histories of joint angles, desired joint angles and torques of right knee for 15 second simulation time are as shown in the next pages.

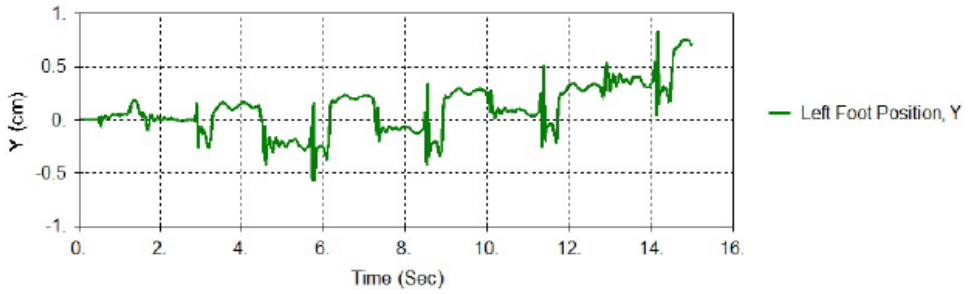

Fig. 5.17. Left Foot Position, Z Value

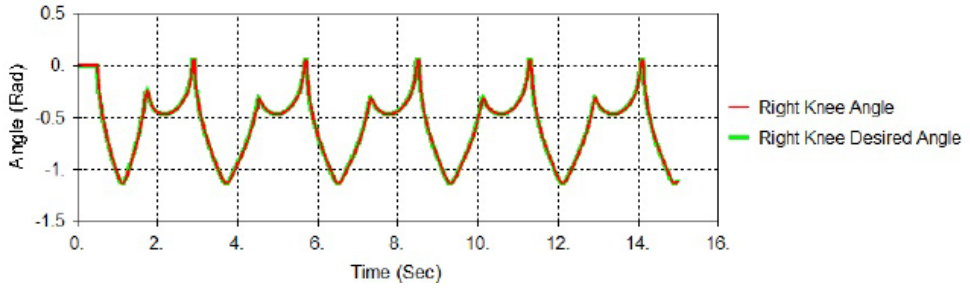

Fig. 5.18. Right Knee Angle and Desired Angle via Time (15 seconds)

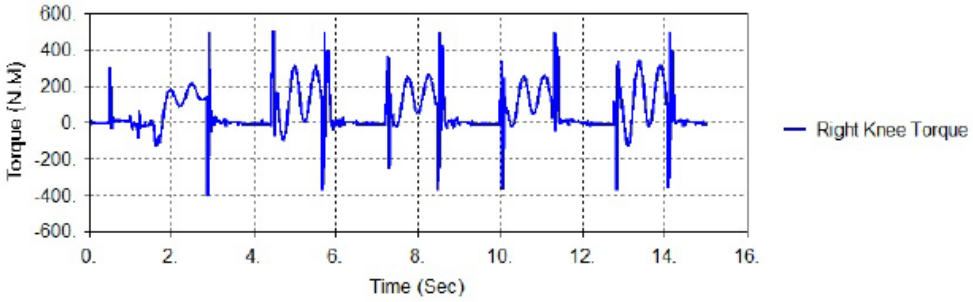

Fig. 5.19. Right Knee Torque via Time (15 seconds)

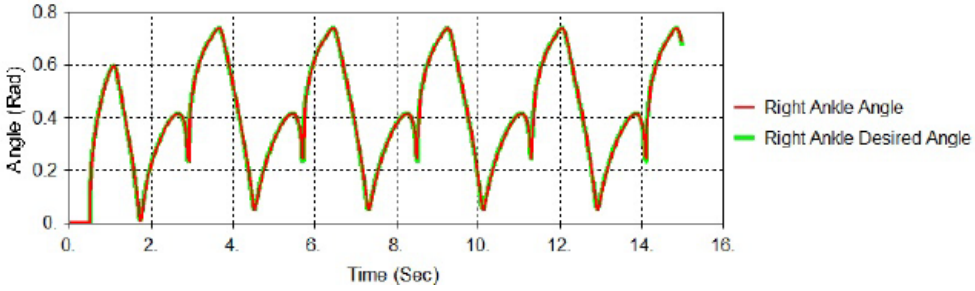

Fig. 5.20. Right Ankle Angle and Desired Angle via Time (15 seconds)

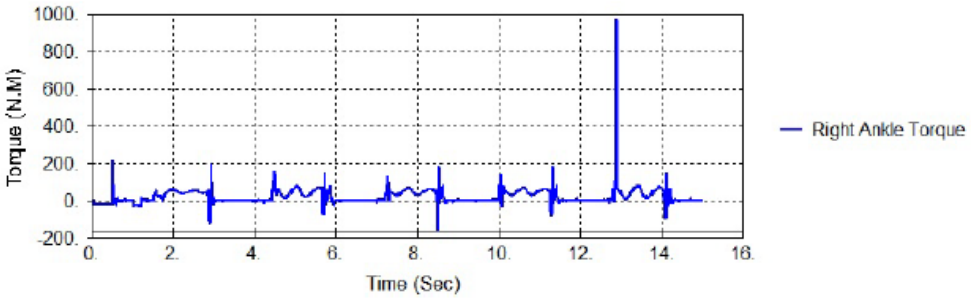

Fig. 5.21. Right Ankle Torque via Time (15 seconds)

6. CONCLUSIONS AND FUTURE WORK

This thesis focuses on the practice of inverse kinematics analysis and the synthesis of walking. A multi-body dynamic model of the full human body was developed and the controllers are designed to simulate the walking motion. This includes the design of the geometric model, development of the control system in kinematics approach, and the simulation setup.

6.1 Concluding Remarks

The human body bipedal walking capability was integrated in a high-fidelity flexible multibody dynamics/DEM. The bipedal walking capabilities include:

- Inverse kinematics:
    - Using Newton's method & least-squares method.
- PD angle control for the joints.
- Penalty technique with an asperity-based friction model for modeling foot frictional contact with ground.

The model was demonstrated using a human body model

6.2 Future Work

This section provides some related ideas are presented that may be helpful to further study the human body modeling and walking motion. In particular, we focus on extension of the current model to improve the stability and tune the PD controller.

Particularly, this chapter addresses improvements and extensions to this work in the areas of using the present simulation code to run several important classes of test cases, extending the model for improved stability analysis, and using PD controllers.

There are plenty of improvements that could be done with the aim of achieving better results in the accuracy of the current simulation and balance control methodology; this section gives a brief outline of topics of planned research related to this work.

Further stairs climbing simulations can improve correlation between the simulation and a human walking kinematics.

New control mechanisms can be developed;

Robustness of the controllers can be improved by formulating a better objective function to optimize the controller gains, the proportional coefficient k_p, the damping coefficient k_d.

Also more complex controller routine such as PID controller or adaptive controller may be applied.

The stability methodology can be further improved by the natural human body stability control rules like implementing the feedback controllers.

In particular, the full humanoid model should be capable of walking in arbitrary directions: forward and backwards, sideways and around corners. It should also be able to step over obstacles while coping with unknown disturbances such as slight pushes or uneven ground. Also; the full model should be capable of running motion.

Furthermore, the system can be sufficiently general to model different human physical type without requiring excessive parameter tuning.

Human motion is governed by multi-objectives in nature. The simulation in this study is based on skeletal model in the joint space. The advantages of skeletal model are its computational efficiency, robustness and stability. However, it lacks the important muscle activity and recruitment information.

In addition, the musculoskeletal model suffers from severe computational burdens for use in the optimization process. One method is to integrate control method and

motion capture to study muscle activities. The other one is to use a skeletal model integrated muscles only on particular joint such as knee and spine.

Most of the desired methodologies need wide-ranging familiarity and information of human anatomy in order to have a natural human robot/model interaction style and control of walking motion.

The research and balance control strategies presented in this work possibly can provide the opportunity for further research on humanoid robotics and other relevant applications. Also, the other study is research on 4-legged robots.

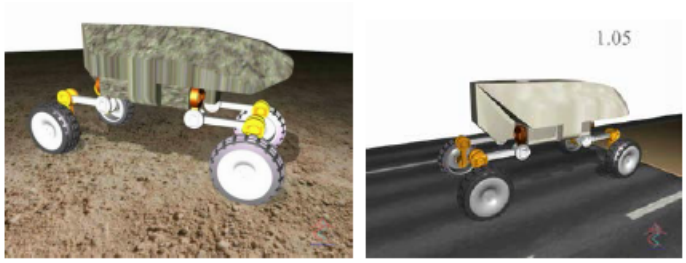

Fig. 6.1. 4-legged Robots

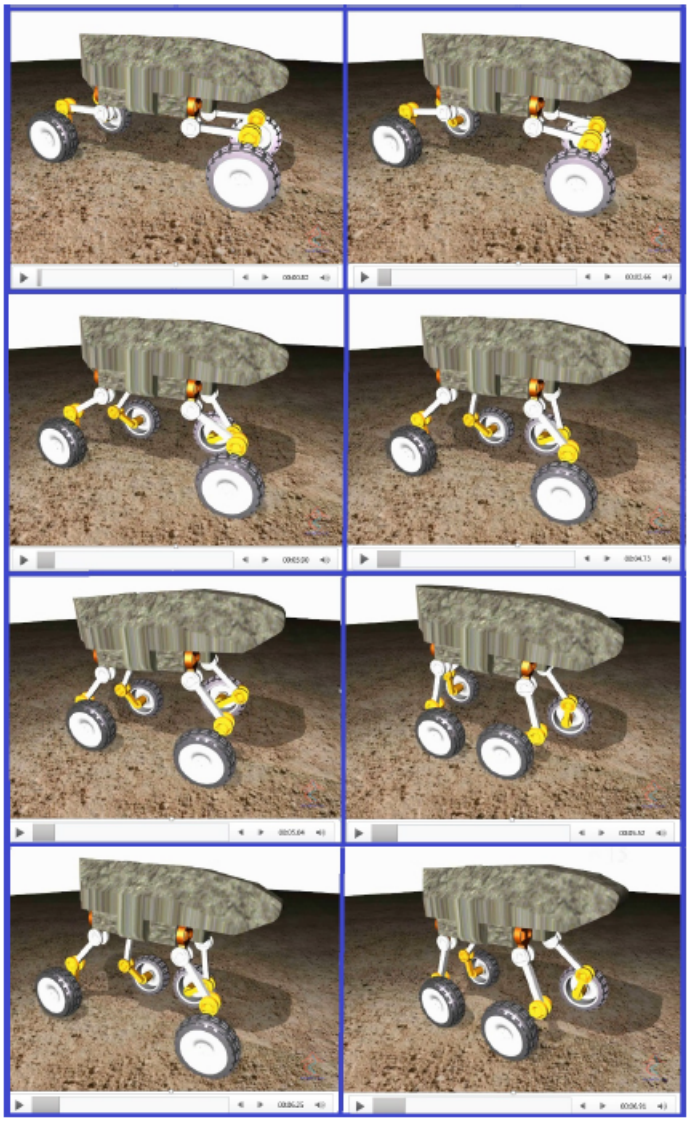

Fig. 6.2. 4-legged Robots Walking on the Ground, Simulated in DIS

REFERENCES

REFERENCES

[1] Y. Aoustin and A. Hamon, "Human like trajectory generation for a biped robot with a four-bar linkage for the knees," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1717–1725, 2013.

[2] S. Hwang, H.-S. Jeon, C.-h. Yi, O.-y. Kwon, S.-h. Cho, and S.-h. You, "Locomotor imagery training improves gait performance in people with chronic hemiparetic stroke: a controlled clinical trial," *Clinical rehabilitation*, vol. 24, no. 6, pp. 514–522, 2010.

[3] T. Upjohn, R. Turcotte, D. J. Pearsall, and J. Loh, "Three-dimensional kinematics of the lower limbs during forward ice hockey skating," *Sports Biomechanics*, vol. 7, no. 2, pp. 206–221, 2008.

[4] X. Luo and W. Xu, "Planning and control for passive dynamics based walking of 3d biped robots," *Journal of Bionic Engineering*, vol. 9, no. 2, pp. 143–155, 2012.

[5] P. R. Vundavilli, S. K. Sahu, and D. K. Pratihar, "Dynamically balanced ascending and descending gaits of a two-legged robot," *International Journal of Humanoid Robotics*, vol. 4, no. 04, pp. 717–751, 2007.

[6] T.-H. S. Li, Y.-T. Su, S.-H. Liu, J.-J. Hu, and C.-C. Chen, "Dynamic balance control for biped robot walking using sensor fusion, kalman filter, and fuzzy logic," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4394–4408, 2012.

[7] T. Wasfy, "Modeling spatial rigid multibody systems using an explicit-time integration finite element solver and a penalty formulation," pp. 901–908, 2004.

[8] S. Y. Nof, "Robot ergonomics: optimizing robot work," *Handbook of Industrial Robotics, Second Edition*, pp. 603–644, 2007.

[9] H. Chuangfeng and F. Yuefa, "Robust control for stable dynamic walking of biped robot," pp. 1750–1754, 2006.

[10] T. Breuer, G. R. Giorgana Macedo, R. Hartanto, N. Hochgeschwender, D. Holz, F. Hegger, Z. Jin, C. Müller, J. Paulus, M. Reckhaus *et al.*, "Johnny: An autonomous service robot for domestic environments," *Journal of intelligent & robotic systems*, vol. 66, no. 1, pp. 245–272, 2012.

[11] J. Stückler and S. Behnke, "Dynamaid, an anthropomorphic robot for research on domestic service applications," *Autom. Control Meas. Electron. Comput. Commun*, vol. 52, pp. 233–243, 2011.

[12] T. Geijtenbeek and N. Pronost, "Interactive character animation using simulated physics: A state-of-the-art review," vol. 31, no. 8, pp. 2492–2515, 2012.

[13] J. Pettré, T. Siméon, and J.-P. Laumond, "Planning human walk in virtual environments," vol. 3, pp. 3048–3053, 2002.

[14] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," pp. 1203–1234, 2016.

[15] C.-H. Kuo, Y.-C. Kuo, T.-S. Chen, Y.-P. Shen, and C.-C. Cheng, "Petri-net-based implementations for fira weightlifting and sprint games with a humanoid robot," *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 282–293, 2014.

[16] L. Liu, M. V. D. Panne, and K. Yin, "Guided learning of control graphs for physics-based characters," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 3, pp. 29–30, 2016.

[17] C.-L. Shih, J. Grizzle, and C. Chevallereau, "Asymptotically stable walking of a simple underactuated 3d bipedal robot," pp. 2766–2771, 2007.

[18] A. Chemori and A. Loría, "Control of a planar underactuated biped on a complete walking cycle," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 838–843, 2004.

[19] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 168–169, 2009.

[20] C. Chevallereau, J. W. Grizzle, and C.-L. Shih, "Asymptotically stable walking of a five-link underactuated 3-d bipedal robot," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 37–50, 2009.

[21] J. Zikmund and C. H. Moog, "The structure of 2-bodies mechanical systems," pp. 6454–6459, 2006.

[22] A. K. Gupta, V. Jha, and V. K. Gupta, "Design and development of remote controlled autonomous synchronic hexaroter aerial (asha) robot," *Procedia Technology*, vol. 14, pp. 51–58, 2014.

[23] I. Fantoni and R. Lozano, *Non-linear control for underactuated mechanical systems*. Springer Science & Business Media, pp. 101-211, 2002.

[24] M. Anderle *et al.*, "Analytical design of the acrobot exponential tracking with application to its walking," pp. 163–168, 2009.

[25] J. Grizzle, C. H. Moog, and C. Chevallereau, "Nonlinear control of mechanical systems with an unactuated cyclic variable," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 559–576, 2005.

[26] T. Wang and C. Chevallereau, "Stability analysis and time-varying walking control for an under-actuated planar biped robot," *Robotics and Autonomous Systems*, vol. 59, no. 6, pp. 444–456, 2011.

[27] S. Bououden and F. Abdessemed, "Walking control for a planar biped robot using 0-flat normal form," *Robotics and Autonomous Systems*, vol. 62, no. 1, pp. 68–80, 2014.

[28] C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica*, vol. 19, no. 05, pp. 557–569, 2001.

[29] N. Chaillet, G. Abba, and E. Ostertag, "Double dynamic modelling and computed-torque control of a biped robot," vol. 2, pp. 1149–1155, 1994.

[30] L. Roussel, C. Canudas-de Wit, and A. Goswami, "Generation of energy optimal complete gait cycles for biped robots," vol. 3, pp. 2036–2041, 1998.

[31] L. Sentis, "Synthesis and control of whole-body behaviors in humanoid systems," Ph.D. dissertation, Citeseer, pp. 35-126, 2007.

[32] A. Goswami, "Foot rotation indicator (fri) point: A new gait planning tool to evaluate postural stability of biped robots," vol. 1, pp. 47–52, 1999.

[33] Z. Yu, J. Li, Q. Huang, X. Chen, G. Ma, L. Meng, S. Zhang, Y. Liu, W. Zhang, W. Zhang *et al.*, "Slip prevention of a humanoid robot by coordinating acceleration vector," pp. 683–688, 2014.

[34] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padır, S. Feng *et al.*, "Human-in-the-loop control of a humanoid robot for disaster response: A report from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015.

[35] M. Sadedel, A. Yousefi-Koma, M. Khadiv, and M. Mahdavian, "Adding low-cost passive toe joints to the feet structure of surena iii humanoid robot," *Robotica*, pp. 1–23, 2016.

[36] R. Weinstein, E. Guendelman, and R. Fedkiw, "Impulse-based control of joints and muscles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 37–46, 2008.

[37] D. Tlalolini, C. Chevallereau, and Y. Aoustin, "Comparison of different gaits with rotation of the feet for a planar biped," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 371–383, 2009.

[38] P. A. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. D. Karssen, C. Paul, and A. Ruina, "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, 2014.

[39] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[40] A. Momeni and Z. Rispoli, "Dranimate: Rapid real-time gestural rigging and control of animation," pp. 61–62, 2015.

[41] F. Multon, L. France, M.-P. Cani-Gascuel, and G. Debunne, "Computer animation of human walking: a survey," *The journal of visualization and computer animation*, vol. 10, no. 1, pp. 39–54, 1999.

[42] F. Faure, G. Debunne, M.-P. Cani-Gascuel, and F. Multon, "Dynamic analysis of human walking," pp. 53–65, 1997.

[43] K. Yin, K. Loken, and M. van de Panne, "Simbicon: Simple biped locomotion control," vol. 26, no. 3, pp. 105–106, 2007.

[44] M. Van de Panne, R. Kim, and E. Fiume, "Virtual wind-up toys for animation," pp. 208–208, 1994.

[45] J. Laszlo, M. van de Panne, and E. Fiume, "Limit cycle control and its application to the animation of balancing and walking," pp. 155–162, 1996.

[46] J. Laszlo, E. Fiume, and M. van de Panne, "Interactive control for physically-based animation," pp. 201–208, 2000.

[47] P. Zhao and M. van de Panne, "User interfaces for interactive control of physics-based 3d characters," pp. 87–94, 2005.

[48] C. Chevallereau, "Modern control theory, chapter 7," under actuated biped robot," *Esculapio, Bologna (Italy)*, pp. 214–251, 1999.

[49] J. W. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Transactions on automatic control*, vol. 46, no. 1, pp. 51–64, 2001.

[50] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-de Wit, and J. Grizzle, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[51] F. Plestan, J. W. Grizzle, E. R. Westervelt, and G. Abba, "Stable walking of a 7-dof biped robot," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 653–668, 2003.

[52] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 42–56, 2003.

[53] L. Magdalena and F. Monasterio-Huelin, "A fuzzy logic controller with learning through the evolution of its knowledge base," *International Journal of Approximate Reasoning*, vol. 16, no. 3-4, pp. 335–358, 1997.

[54] A. Kun and W. T. Miller, "Adaptive dynamic balance of a biped robot using neural networks," vol. 1, pp. 240–245, 1996.

[55] S. Coros, P. Beaudoin, K. K. Yin, and M. van de Pann, "Synthesis of constrained walking skills," vol. 27, no. 5, pp. 113–114, 2008.

[56] K. Yin, S. Coros, P. Beaudoin, and M. van de Panne, "Continuation methods for adapting simulated skills," vol. 27, no. 3, pp. 81–82, 2008.

[57] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers for uncertain inputs and environments," vol. 29, no. 4, pp. 73–74, 2010.

[58] J. Tan, Y. Gu, G. Turk, and C. K. Liu, "Articulated swimming creatures," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, pp. 58–59, 2011.

[59] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM transactions on graphics*, vol. 31, no. 4, pp. 168–169, 2012.

[60] T. Geijtenbeek, M. van de Panne, and A. F. van der Stappen, "Flexible muscle-based locomotion for bipedal creatures," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 206–207, 2013.

[61] H. Van Welbergen, B. J. Van Basten, A. Egges, Z. M. Ruttkay, and M. H. Overmars, "Real time animation of virtual humans: A trade-off between naturalness and control," vol. 29, no. 8, pp. 2530–2554, 2010.

[62] Y. Lee, M. S. Park, T. Kwon, and J. Lee, "Locomotion control for many-muscle humanoids," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 218–219, 2014.

[63] S. Kajita, K. Kaneko, M. Morisawa, S. Nakaoka, and H. Hirukawa, "Zmp-based biped running enhanced by toe springs," pp. 3963–3969, 2007.

[64] R. Tajima, D. Honda, and K. Suga, "Fast running experiments involving a humanoid robot," pp. 1571–1576, 2009.

[65] D. Tlalolini, Y. Aoustin, and C. Chevallereau, "Design of a walking cyclic gait with single support phases and impacts for the locomotor system of a thirteen-link 3d biped using the parametric optimization," *Multibody System Dynamics*, vol. 23, no. 1, pp. 33–56, 2010.

[66] D. Wilson, J. Feikes, and J. Oconnor, "Ligaments and articular contact guide passive knee flexion," *Journal of biomechanics*, vol. 31, no. 12, pp. 1127–1136, 1998.

[67] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Toe joints that enhance bipedal and fullbody motion of humanoid robots," vol. 3, pp. 3105–3110, 2002.

[68] S. Takao, H. Ohta, Y. Yokokohji, and T. Yoshikawa, "Function analysis of human-like mechanical foot, using mechanically constrained shoes," vol. 4, pp. 3847–3852, 2004.

[69] J. Yoon, N. Handharu, and G. Kim, "A bio-robotic toe, foot and heel models of a biped robot for more natural walking," pp. 63–68, 2007.

[70] K. W. Sok, M. Kim, and J. Lee, "Simulating biped behaviors from human motion data," vol. 26, no. 3, pp. 107–108, 2007.

[71] M. G. Coutinho, *Dynamic simulations of multibody systems*. Springer Science & Business Media, pp. 10-158, 2013.

[72] S.-M. Song and K. J. Waldron, "An analytical approach for gait study and its applications on wave gaits," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 60–71, 1987.

[73] A. Patla, J. Frank, and D. Winter, "Assessment of balance control in the elderly: major issues," *Physiotherapy Canada*, vol. 42, no. 2, pp. 89–97, 1990.

[74] S. Lohmeier, T. Buschmann, and H. Ulbrich, "Humanoid robot lola," pp. 775–780, 2009.

[75] A. K. Sanyal and A. Goswami, "Dynamics and control of the reaction mass pendulum (rmp) as a 3d multibody system: Application to humanoid modeling," pp. 589–596, 2011.

[76] M. Shahinpoor, "Ionic polymer–conductor composites as biomimetic sensors, robotic actuators and artificial musclesa review," *Electrochimica Acta*, vol. 48, no. 14, pp. 2343–2353, 2003.

[77] T. Buschmann, S. Lohmeier, and H. Ulbrich, "Humanoid robot lola: Design and walking control," *Journal of physiology-Paris*, vol. 103, no. 3, pp. 141–148, 2009.

[78] M. Hardt and O. Von Stryk, "The role of motion dynamics in the design, control and stability of bipedal and quadrupedal robots," pp. 206–223, 2002.

[79] "Atlas - the agile anthropomorphic robot," http://www.bostondynamics.com/robot_Atlas.html, accessed: 2017-02-15.

[80] M. Friedmann, J. Kiener, S. Petters, H. Sakamoto, D. Thomas, and O. von Stryk, "Versatile, high-quality motions and behavior control of humanoid soccer robots," pp. 9–16, 2006.

[81] S. Uddin, O. Krause, and D. Martin, "A linearization technique to estimate operational constraints states by pseudo inverse matrix," pp. 1–6, 2016.

[82] M. A. Ali, H. A. Park, and C. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots," pp. 704–709, 2010.

[83] J. K. Parker, A. R. Khoogar, and D. E. Goldberg, "Inverse kinematics of redundant robots using genetic algorithms," pp. 271–276, 1989.

[84] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," vol. 2, pp. 1620–1626, 2003.

[85] Y.-J. Kim, J.-Y. Lee, and J.-J. Lee, "A balance control strategy of a walking biped robot in an externally applied force," pp. 572–577, 2012.

[86] M. A. Brubaker, L. Sigal, and D. J. Fleet, "Estimating contact dynamics," pp. 2389–2396, 2009.

[87] J. Ambrósio and M. T. da Silva, "Multibody dynamic approaches in modeling and analysis of biomechanical systems for human motion and injury assessment," pp. 101–103, 2016.

[88] T. M. Wasfy and M. J. Leamy, "Modeling the dynamic frictional contact of tires using an explicit finite element code," pp. 2381–2388, 2005.

[89] G. S. Yesmunt, "Design, analysis, and simulation of a humanoid robotic arm applied to catching," Master's thesis, Purdue University, Indianapolis, pp. 12-39, 2014.

[90] M. J. Leamy and T. M. Wasfy, "Transient and steady-state dynamic finite element modeling of belt-drives," *Transactions-American Society of Mechnical Engineering Journal of Dynamic Systems Measurement and Control*, vol. 124, no. 4, pp. 575–581, 2002.

[91] T. M. Wasfy, "Asperity spring friction model with application to belt-drives," pp. 371–378, 2003.

[92] T. M. Wasfy and J. OKins, "Finite element modeling of the dynamic response of tracked vehicles," pp. 1999–2009, 2009.

[93] S. G. Ahmadi, T. M. Wasfy, H. M. Wasfy, and J. M. Peters, "High-fidelity modeling of a backhoe digging operation using an explicit multibody dynamics code with integrated discrete particle modeling capability," pp. V07AT10A031–V07AT10A031, 2013.

[94] C.-L. Shih, "The dynamics and control of a biped walking robot with seven degrees of freedom," *Transactions-American Society of Mechnical Engineering Journal of Dynamic Systems Measurement and Control*, vol. 118, pp. 683–690, 1996.

[95] S. A. Ghoohaki, "High-fidelity modeling of a backhoe digging operation using an explicit multibody dynamics finite element code with integrated discrete element method," Master's thesis, Purdue University, Indianapolis, pp. 8-30, 2013.

[96] C. E. Clauser, J. T. McConville, and J. W. Young, "Weight, volume, and center of mass of segments of the human body (AMRL TR 69-70)," DTIC Document (NTIS NO.AD-710622), Tech. Rep., pp. 8-9, 1969.

[97] G. J. Gormley, E. Stoner, R. C. Bruskewitz, J. Imperato-McGinley, P. C. Walsh, J. D. McConnell, G. L. Andriole, J. Geller, B. R. Bracken, J. S. Tenover *et al.*, "The effect of finasteride in men with benign prostatic hyperplasia," *New England Journal of Medicine*, vol. 327, no. 17, pp. 1185–1191, 1992.

[98] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of Applied Physiology*, vol. 100, no. 2, pp. 390–398, 2006.

[99] R. V. Levine and A. Norenzayan, "The pace of life in 31 countries," *Journal of cross-cultural psychology*, vol. 30, no. 2, pp. 178–205, 1999.

[100] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. Pick, and W. H. Warren, "Visual flow influences gait transition speed and preferred walking speed," *Experimental brain research*, vol. 181, no. 2, pp. 221–228, 2007.

APPENDICES

A. FORWARD STEP, HERMIT CUBIC SPLINE CODE

```
% people walk at about 1.4 m/s (5.0 km/h; 3.1 mph) %

clc;
close all; clear all;

T = 1; %sec, Step Time Length
Points_number = 100;
Offset = 5; %cm
L = 50; %cm, Step Length
H = 20; %cm, Step Height
du = 0.01;
Vel = 1.4*100; %cm/sec
dt = T/Points_number;  %sec
ds = Vel*dt;
u = 0:du:1;
w1 = 1 - 3*u.^2 + 2*u.^3;
w2 = u - 2*u.^2 + u.^3;
w3 = 3*u.^2 - 2*u.^3;
w4 = u.^3 - u.^2;
X_0 = 0;
Y_0 = 0;
% Line1 Eq.: Y=0
% Line2 Eq.: Y=(2H)/L*X
```

```
% Line3 Eq.: Y=-(2H)/L*X+(2H)
% 12 Control Points
X_ControlPoints = [L/2, (L/2+L/4), L, 5*L/6, L/2, L/6, 0, L
    /10, L/6, L/4, L/2-L/6, L/2, L/2+L/6, 3/4*L, 5*L/6, L] ;
Y_ControlPoints = [-Offset, H, 2*L/50, 0.*X_ControlPoints
    (4:6),1*L/50, L/10+L/30,... L/18+(2*H)/L.*X_ControlPoints
    (9:12) , ... L/18-(2*H)/L.*X_ControlPoints(13:15) + (2*H)
    ,2*L/50   ,...] ;
X_Test= [5*L/6 , L/2, L/6, 0,L/10, L/6, L/4, L/2-L/6, L/2, L
    /2+L/6, 3/4*L, 5*L/6, L];
Y_Test = [Y_ControlPoints(4:end)];
X_ControlPoints =[X_ControlPoints , X_Test];
Y_ControlPoints =[Y_ControlPoints , Y_Test];
% Transfer Points
X_ControlPoints = X_ControlPoints + X_0;
Y_ControlPoints = Y_ControlPoints + Y_0;
% Hermit Spline
X_Spline = w1*X_ControlPoints(1) + w2*(0.5*(X_ControlPoints
    (2)-X_ControlPoints(1))) + w3*X_ControlPoints(2) + w4
    *(0.5*(X_ControlPoints(3)-X_ControlPoints(1))) ;
Y_Spline = w1*Y_ControlPoints(1) + w2*(0.5*(Y_ControlPoints
    (2)-Y_ControlPoints(1))) + w3*Y_ControlPoints(2) + w4
    *(0.5*(Y_ControlPoints(3)-Y_ControlPoints(1))) ;
for i=2:26
    X_Spline = [X_Spline, w1*X_ControlPoints(i) + w2*(0.5*(
        X_ControlPoints(i+1)-X_ControlPoints(i-1))) + w3*
        X_ControlPoints(i+1) + w4*(0.5*(X_ControlPoints(i+2)-
        X_ControlPoints(i)))];
```

```
    Y_Spline = [Y_Spline, w1*Y_ControlPoints(i) + w2*(0.5*(
        Y_ControlPoints(i+1)-Y_ControlPoints(i-1))) + w3*
        Y_ControlPoints(i+1) + w4*(0.5*(Y_ControlPoints(i+2)-
        Y_ControlPoints(i)))];
end
X_Spline = [X_Spline, w1*X_ControlPoints(14+13) + w2*(0.5*(
    X_ControlPoints(15+13)-X_ControlPoints(13+13))) + w3*
    X_ControlPoints(15) + w4*(0.5*(X_ControlPoints(15+13)-
    X_ControlPoints(14+13)))];
Y_Spline = [Y_Spline, w1*Y_ControlPoints(14+13) + w2*(0.5*(
    Y_ControlPoints(15+13)-Y_ControlPoints(13+13))) + w3*
    Y_ControlPoints(15) + w4*(0.5*(Y_ControlPoints(15+13)-
    Y_ControlPoints(14+13)))];

%Plotting
figure()
plot(X_ControlPoints,Y_ControlPoints,'ko','MarkerSize',10,'
    LineWidth',3)
hold on;
ind = find(abs(X_Spline-L) < 0.001);
plot(X_Spline(1:ind(1)),Y_Spline(1:ind(1)),'r','LineWidth',3)
plot(X_Spline(202:ind(end-1)),Y_Spline(202:ind(end-1)),'b','
    LineWidth',4)
plot(X_Spline(ind(end-1):end),Y_Spline(ind(end-1):end),'g','
    LineWidth',2)

%Dist of points
Dist = [0];
for n = 2:size(X_Spline,2)
```

```
    Dist = [Dist, Dist(n-1) + sqrt((X_Spline(n)-X_Spline(n-1)
        )^2 + (Y_Spline(n)-Y_Spline(n-1))^2)];
end
n = 1;
eps = 0.01;
X_Final = [1;X_Spline(1)];
Y_Final = [1;Y_Spline(1)];
for counter = 2:size(X_Spline,2)
    x_new = X_Spline(counter);
    y_new = Y_Spline(counter);
    if(abs((Dist(counter) - Dist(X_Final(1,n))) - ds)< 0.2)
        X_Final = [X_Final, [counter;x_new]];
        Y_Final = [Y_Final, [counter;y_new]];
        n = n+1;
    end
end

hold on;
plot(X_Final(2,:),Y_Final(2,:),'y*')
max(Dist)
n
1
Final_Points_number = n;
dT = T/Final_Points_number;
Time = (0:dT:T-dT)';
X = zeros(n,1);
Y = Y_Final(2,:)';
Z = X_Final(2,:)';
n_1 = zeros(n,1); n_2 = ones(n,1); n_3 = zeros(n,1);
```

```
t_1 = zeros(n,1); t_2 = ones(n,1); t_3 = zeros(n,1);
% Table = table(Time, X, Y, Z, n_1, n_2, n_3, t_1, t_2, t_3);
% filename = ([datestr(date),' Coordinates.xlsx']);
% writetable(Table,filename)
% open(filename)

legend('Control Points','Forward Motion_ Hermite Cubic
    Splines ','stead-state-cycle_ Hermite Cubic Splines ','
    full stead-state-cycle_ Hermite Cubic Splines ','cte dist
    on Hermite cubic splines')
```

B. BACKWARD STEP, HERMIT CUBIC SPLINE CODE

```
% people walk at about 1.4 m/s (5.0 km/h; 3.1 mph) %

clc;
close all; clear all;

T = 5; %sec, Step Time Length
Points_number = 500;
Offset = 5; %cm
L = 50; %cm, Step Length
H = 20; %cm, Step Height
du = 0.01;
Vel = 1.4*100; %cm/sec
dt = T/Points_number; %sec
ds = Vel*dt;
u = 0:du:1;
w1 = 1 - 3*u.^2 + 2*u.^3;
w2 = u - 2*u.^2 + u.^3;
w3 = 3*u.^2 - 2*u.^3;
w4 = u.^3 - u.^2;
X_0 = 0;
Y_0 = 0;
% Line1 Eq.: Y=0
% Line2 Eq.: Y=(2H)/L*X
```

```
% Line3 Eq.: Y=-(2H)/L*X+(2H)
% 12 Control Points
X_ControlPoints = [L/2, 0, L/6, L/4, L/2-L/6 , L/2, L/2+L/6,
    3/4*L, 5*L/6, L, 5*L/6, L/2,L/6,0] ;
Y_ControlPoints = [-Offset , ... (2*H)/L.*X_ControlPoints
    (2:5) , ... -(2*H)/L.*X_ControlPoints(6:9) + (2*H) , ...
    0.*X_ControlPoints(10:13),0] ;
X_Test= [ L/6, L/4, L/2-L/6 , L/2, L/2+L/6, 3/4*L, 5*L/6, L,
    5*L/6, L/2,L/6,0,L/6];
Y_Test = [Y_ControlPoints(3:end),(2*H)/L.*X_Test(end)];
X_ControlPoints =[X_ControlPoints, X_Test];
Y_ControlPoints =[Y_ControlPoints, Y_Test];
% Transfer Points
X_ControlPoints = X_ControlPoints + X_0;
Y_ControlPoints = Y_ControlPoints + Y_0;
% Hermit Spline
X_Spline = w1*X_ControlPoints(1) + w2*(0.5*(X_ControlPoints
    (2)-X_ControlPoints(1))) + w3*X_ControlPoints(2) + w4
    *(0.5*(X_ControlPoints(3)-X_ControlPoints(1))) ;
Y_Spline = w1*Y_ControlPoints(1) + w2*(0.5*(Y_ControlPoints
    (2)-Y_ControlPoints(1))) + w3*Y_ControlPoints(2) + w4
    *(0.5*(Y_ControlPoints(3)-Y_ControlPoints(1))) ;
for i=2:13+12
    X_Spline = [X_Spline, w1*X_ControlPoints(i) + w2*(0.5*(
        X_ControlPoints(i+1)-X_ControlPoints(i-1))) + w3*
        X_ControlPoints(i+1) + w4*(0.5*(X_ControlPoints(i+2)-
        X_ControlPoints(i)))];
    Y_Spline = [Y_Spline, w1*Y_ControlPoints(i) + w2*(0.5*(
        Y_ControlPoints(i+1)-Y_ControlPoints(i-1))) + w3*
```

```
        Y_ControlPoints(i+1) + w4*(0.5*(Y_ControlPoints(i+2)-
        Y_ControlPoints(i)))];
end
X_Spline = [X_Spline, w1*X_ControlPoints(14+12) + w2*(0.5*(
    X_ControlPoints(15+12)-X_ControlPoints(13+12))) + w3*
    X_ControlPoints(15+12) + w4*(0.5*(X_ControlPoints(15+12)-
    X_ControlPoints(14+12)))];
Y_Spline = [Y_Spline, w1*Y_ControlPoints(14+12) + w2*(0.5*(
    Y_ControlPoints(15+12)-Y_ControlPoints(13+12))) + w3*
    Y_ControlPoints(15+12) + w4*(0.5*(Y_ControlPoints(15+12)-
    Y_ControlPoints(14+12)))];

%Plotting
figure()
plot(X_ControlPoints,Y_ControlPoints,'ko','MarkerSize',10,'
    LineWidth',3)
hold on;
plot(X_Spline(1:102),Y_Spline(1:102),'r','LineWidth',3)
plot(X_Spline(102:1414-100),Y_Spline(102:1414-100),'b','
    LineWidth',4)
plot(X_Spline(1314:end),Y_Spline(1314:end),'g','LineWidth',2)

%Dist of points
Dist = [0];
for n = 2:size(X_Spline,2)
     Dist = [Dist, Dist(n-1) + sqrt((X_Spline(n)-X_Spline(n-1)
         )^2 + (Y_Spline(n)-Y_Spline(n-1))^2)];
end
n = 1;
```

```
eps = 0.01;
X_Final = [1;X_Spline(1)];
Y_Final = [1;Y_Spline(1)];
for counter = 2:size(X_Spline,2)
    x_new = X_Spline(counter);
    y_new = Y_Spline(counter);
    if(abs((Dist(counter) - Dist(X_Final(1,n))) - ds)< 0.2)
        X_Final = [X_Final, [counter;x_new]];
        Y_Final = [Y_Final, [counter;y_new]];
        n = n+1;
    end
end

hold on;
plot(X_Final(2,1:end-8),Y_Final(2,1:end-8),'y*')
max(Dist)
n
n= n-8
Final_Points_number = n;
dT = T/Final_Points_number;
Time = (0:dT:T-dT)';
X = zeros(n,1);
Y = Y_Final(2,1:end-8)';
Z = X_Final(2,1:end-8)';
n_1 = zeros(n,1); n_2 = ones(n,1); n_3 = zeros(n,1);
t_1 = zeros(n,1); t_2 = ones(n,1); t_3 = zeros(n,1);
Table = table(Time, X, Y, Z, n_1, n_2, n_3, t_1, t_2, t_3);
filename = ([datestr(date),' Coordinates.xlsx']);
writetable(Table,filename)
```

```
open(filename)

legend('Control Points','Backward Motion_ Hermite Cubic
    Splines ','stead-state-cycle_ Hermite Cubic Splines ','
    full stead-state-cycle_ Hermite Cubic Splines ','cte dist
    on Hermite cubic splines')
```