UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA
DELL'INFORMAZIONE
-
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

Simulating a Flexible Robotic System based on Musculoskeletal Modeling

Candidato: **BORTOLETTO ROBERTO** (622359-IF)

Relatore: **Prof. ENRICO PAGELLO**
Correlatore: **PhD MASSIMO SARTORI**

Data di Laurea: 12 Dicembre 2011

ANNO ACCADEMICO 2011-2012

Abstract

Humanoid robotics offers a unique research tool for understanding the human brain and body. Human body representations have been used for centuries to help in understanding and documenting the shape and function of its compounding parts. The synthesis of human motion is a complex procedure that involves accurate reconstruction of movement sequences, modeling of musculoskeletal kinematics, dynamics and actuation, and characterization of reliable performance criteria. Many of these processes have much in common with the problems found in robotics research, with the recent advent of complex humanoid systems. Several technologies required for this new kind of robots reached the necessary level of performance. It is clear that these human-friendly robots will look very different than today's industrial robots and generating coordinated natural motion in human-like robotic structures has proved to be a challenging task. The characterization and control of humanoid systems has an impact beyond robotics. It can provide the support to understand biological functions of the human body (biomechanics), tools to design machines and spaces where humans operate (ergonomics), simulation environments to study the effects of musculoskeletal alterations (surgical simulation) and to design and study rehabilitation systems, and support to synthesize realistic computer animations.

This work presents the design and development of a new-generation bipedal robot. Its modeling and simulation has been realized by using an open-source software to create and analyze dynamic simulation of movement: OpenSim. Starting from a study by Fuben He, our model aims to be used as an innovative approach to the study of a such type of robot in which there are series elastic actuators represented by active and passive spring components in series with motors. It has provided of monoarticular and biarticular joint in a very similar manner to human musculoskeletal model. We have developed several model of this biped robot that allow us to study its movement from four major point of view: angle-driven, by prescribing the angle to each joint, torque-driven with and without springs, by prescribing force and moment to each joint and muscle-driven by using Compute Muscle Control (CMC), a built-in tool of OpenSim. In this way we can compare in a more exact manner what happens during human movement with that happens during robot movement and consequently improve the design of motion and balance controller in the real robot.

This thesis is only the starting point of a wide range of other possible future works: from the control structure completion and whole-body control application, to imitation learning and reinforcement learning for human locomotion, from motion test on flat ground to motion test on rough ground, and obviously the transition from simulation to practice with a real elastic bipedal robot biologically-inspired that can move like a human being.

Sommario

La Robotica umanoide offre uno strumento unico di ricerca per la comprensione del corpo e del cervello umano. Rappresentazioni del corpo umano sono state usate per secoli al fine di aiutare a comprendere e documentare la forma e la funzione delle sue componenti. La sintesi del movimento umano è una procedura complessa che coinvolge la ricostruzione accurata di sequenze di movimento, la modellazione cinematica, dinamica e dell'attuazione del sistema muscolo-scheletrico secondo i criteri di efficienza ed affidabilità che lo caratterizzano. Questi processi hanno molto in comune con i problemi riscontrati nella ricerca robotica, in particolare con l'avvento dei sistemi umanoidi complessi. E' chiaro che questi robot human-friendly sono e saranno molto diversi dai robot industriali conosciuti finora e l'obiettivo di fargli compiere movimenti tipicamente eseguiti dagli esseri umani si è dimostrato fin da subito un compito assai impegnativo. Oltre tutto, la caratterizzazione ed il controllo di un robot umanoide ha un impatto al di là della robotica pura. Può costituire uno strumento atto alla comprensione delle funzioni biologiche del corpo umano (biomeccanica), alla progettazione di macchine e spazi in cui gli esseri umani operano (ergonomia), all'implementazione di ambienti di simulazione per studiare gli effetti delle alterazioni muscolo-scheletriche (simulazione chirurgica) e per la progettazione e lo studio di sistemi di riabilitazione, nonché il supporto per sintetizzare animazioni al computer sempre più realistiche.

Questo lavoro descrive la modellazione, la simulazione e lo studio di un robot bipede di nuova generazione, eseguiti utilizzando un software open-source per creare e analizzare il movimento umano: OpenSim. Partendo da uno studio condotto da Fuben He, il nostro modello vuole essere un approccio innovativo allo studio di una tale tipologia di robot in cui sono presenti servo attuatori in serie a strutture elastiche che operano in maniera attiva e passiva, nel sintetizzare giunti mono- e bi-articolari, in maniera del tutto simile a quanto accade nell'essere umano. Abbiamo sviluppato diversi modelli che ci permettono di studiare il movimento da quattro principali punti di vista: angle-driven, prescrivendo l'angolo di ogni giuntura, torque-driven con e senza molle, prescrivendo forze e momenti di ciascun servo attuatore e muscle-driven utilizzando il controllo muscolare (Computed Muscle Control), uno strumento integrato di OpenSim. In questo modo possiamo confrontare più precisamente quello che succede durante il movimento umano con ciò che accade durante lo stesso movimento del robot e di conseguenza migliorare la progettazione dei controller di movimento e di equilibrio nel robot reale.

Questo è solo il punto di partenza di un'ampia gamma di possibili lavori futuri: dal completamento della struttura di controllo e della parte superiore del corpo, all'apprendimento per imitazione, da test di movimento su terreno piano a test su fondi sconnessi, e ovviamente il passaggio dalla simulazione alla pratica con un vero e proprio robot bipedi elastico, biologically-inspired che può muoversi come un essere umano.

Acknowledgments

I would like to thank my advisor, Prof. Enrico Pagello, for his continual drive to make me improve my work and for his faith that it would always work better.
I would also like to thank those who have helped me over these months, specially Massimo Sartori and Fuben He for their precious advices and teachings, and Andrea Rampin for his continuous support.
A special thanks goes to my parents, Livio and Fortunata who has given me the possibility to study and live an experience like that of the last years. Last but not least, I thank very much my girlfriend Serena who has been always close to me.

Will robots inherit the earth? Yes, but they will be our children.
- Marvin Minsky, 1995

Contents

Chapter 1

Introduction

For decades, popular culture has been enthralled with the possibility of robots that act and look like humans. We are promised by film, fiction and television that humanoids will cook for us, clean for us, become our best friends and teach our children. More recently, the media has covered a surprising number of new humanoid robots emerging on the commercial market. Like many new technologies, these early generations of commercially available humanoids are costly curiosities, useful for entertainment, but little else. Humanoid robots can already autonomously perform task decomposition necessary to carry out high-level, complex commands given through gesture and speech. Humanoids can adapt and orchestrate existing capabilities as well as create new behaviors using a variety of machine learning techniques. Humanoids may prove to be the ideal robot design to interact with people. After all, humans tend to naturally interact with other human-like entities; the interface is hardwire in our brains. Furthermore, while the industrial manipulator and mobile robots needs to adapt and drastically change their environment, the humanoid robots can work directly in the same human environment without any modification. Historically, we humans have adapted to the highly constrained modality of monitor and keyboard. In the future, technology will adapt to us. Humanoids probably will change the way we interact with machines and will impact how we interact with and understand each other.

Humanoid robotics also offers a unique research tool for understanding the human brain and body. Human body representations have been used for centuries to help in understanding and documenting the shape and function of its compounding parts. Since the Da Vinci drawings, human body atlases have evolved significantly and can nowadays describe the human anatomy with great precision, using multi-level biological scales spanning multiple dimensions. In parallel, the body physiology, its systems and their functions, the mechanisms of human motion, the pathological and healing processes are among the many topics being studied and described in different domains of science. The synthesis of human motion is a complex procedure that involves accurate reconstruction of movement sequences, modeling of musculoskeletal kinematics, dynamics and actuation, and characterization of reliable performance criteria. Many of these processes have much in common with the problems found in robotics research, with the recent advent of complex humanoid systems. Several technologies required for this new kind of robots reached the necessary level of performance. It is clear that these human-friendly robots will look very different than today's industrial robots and generating coordinated natural motion in human-like robotic structures has proved to be a challenging task. The characterization and control of humanoid systems has an impact beyond robotics. It can provide the support to understand biological functions of the human body (biomechanics), tools to design machines and spaces where humans operate (ergonomics), simulation environments to study the effects of musculoskeletal alterations (surgical simulation) and to design and study rehabilitation systems, and support to synthesize realistic computer animations.

Humanoid refers to any being whose body structure resembles that of a human: head, torso, legs, arms, hands. But it is also a robot made to resemble a human both in appearance and behavior. The difference between a robot and android is only skin-deep, looks exactly like humans on the outside, but with internal mechanics of humanoid robot. The main differences between humanoids and other kinds of robots (manipulators, mobile robots, multi-legged robot) are principally: bipedal human-like locomotion (stable gait, changing model during one/two feet support walking) and two legs + two arms + torso + head (hyper Degrees of Freedom (DoFs) system and complex kinematics and dynamics). Bipedism and cognition have very close relationship; the time line of human evolution shows three important events, called respectively: Erectus, Pre-Sapiens and Sapiens. For the human evolution bipedism frees the hands to create tools and start cognition and the humanoids robots nowadays status is focusing more in bipedism than in cognition: stable bipedal locomotion is not totally solved, we have only good lab example, it is mandatory to have robust biped platforms in order to implement cognitive robotics and there are few example of human-size humanoids in the market now and probably the same in the next future. We are in the Pre-Robotic stage comparing with the human evolution. Humanoid Robotics includes several type of projects where perception, processing and action are embodied in a recognizably anthropomorphic form in order to emulate some subset of the physical, cognitive and social dimensions of the human body and experience. The goal in humanoid robot design is to make a robot that acts safely alongside humans, extending our capabilities in a wide variety of tasks and environments. A robot that interacts socially with people in typical, everyday environments. So, from this point of view, Humanoid Robotics is not a well defined field, but rather a collaborative efforts that crosscut many disciplines. *Perception*, with computer vision and many other sensing modalities including taste, smell, sonar, IR, haptic feedback, tactile sensors and range of motion sensors. *Human-Robot interaction* with the study of human factors related to the tasking and control of humanoid robots. This area considers the ways in which humanoids can be profitably and safely integrated into everyday life. Such communications and interactions would imply not only explicit and conscious signal exchange, but also an implicit and subconscious one. Tomorrow's humanoids will operate in human environments, where efficient manipulation and locomotion skills, and safe contact interactions will be critical design factors. Citing Nakamura et al.: *"when the man-machine interface obtains access to the somatosensory information, machines would make the first step to understand humans. Such implicit communication between humans and machines could be termed as a cognitive-level communication"* [83]. *Legged locomotion* because they must be able to walk up stairs and steep inclines and over rough, uneven terrain. *Arm control and dexterous manipulation*, in which many researchers are working on dexterous tasks including catching balls, juggling, chopping vegetables, performing telesurgery and pouring coffee. *Learning and adaptive behavior*, to adapt existing capabilities to cope with environmental changes. Among all these disciplines, also *Artificial Intelligence* (AI) community looks to field such as neuroscience, cognitive psychology and biology for new insight. So, in the last years, a new behovior-based view of intelligence is emerged which has trasferred the emphasis from intelligent processing to robust real world action. As consequence come the question of how best to impart primitive behaviors to robots. Many researchers use biologically inspired technique such as Artificial Neural Newtworks (ANN), other use learning techniques such as reinforcement learning and genetic algorithms. In this way, robots can learn to solve problems, they have the potential to acquire new knowledge by adjusting parameters, exploiting patterns, devising new strategies, generating entire behaviors, predicting environmental changes and recognizing the strategies of opponents or exchanging knowledge with other robots. Japanese scientists are searching for learning techniques that can scale indefinitely. At the University of Tokyo, researchers have used a learning methodology they have called interactive teaching to give robots the ability to drive their own development. At the University of Southern California, Maja Mataric has worked to provide the Sarcos humanoid, DB, with a set of basis behaviors on which development learning can built. On the same robot, many researchers use motion capture to record human movement trajectories for shoulders, elbows, wrists, hips, knees and ankles. This data helps to identify the underlying principles that constrain and optimize body movement. These principles inform the way motion primitives are developed and used by humanoid designers to study smooth and efficient movement.

While only a few institutions are fully dedicated to the creation of humanoid robots, a host of projects around the world are meeting with encouraging success in particular areas. With the rise of the computer, people immediately began to envision the potential for encoding human intelligence into textual programs, but soon discovered that static programs and rule-based logic cannot capture the true essence of human intelligence. Estranged from perception and action, such intelligence derived meaning only as an extension of the human creator or user. Once embodied in real robots, such programs were confounded by noisy and all-to-often inconsistent data streaming in and out from a host of real-world sensors and actuators. However, robotic research community continued, mostly from a mechanical point of view, to develop new robotic tools for a variety of purposes. In 1973, the construction of the world's first humanoid robot was started at the Waseda University in Tokyo under the direction of Ichiro Kato. The first full-scale anthropomorphic robot in the world was called WABOT-1. It consisted of a limb-control system, a vision system and a conversation system. At the time, it was estimated that WABOT-1 had the mental faculty of a one half year old child. After it, in 1980, at Waseda University was developed also WL-9DR, the world's first robot to exhibit quasi-dynamic walking. Then, in 1985, with Hitachi Ltd. was developed WHL-11 (Waseda Hitachi Leg-11) biped, walking robot and in the same year, thanks again to Kato and his research group, also WASUBOT was built. It was a humanoid musician developed with Sumitomo Electric Industry Ltd. (WAseda SUmitomo roBOT). WASUBOT read a musical score and played a repertoire of 16 tunes on a keyboard instrument. Since these early successes, the Japanese electronics and automotive industries have played a key role in the emergence of humanoids.
Is generally assumed that a legged locomotion is much easier to accomplish on smaller humanoids. A significantly example is the SDR-4X developed by Sony as a domestic robot.

This was a remarkable project for the flexibility of the robot that used vision to adapt its walking behavior to cope with stairs and other features typical of a home. Then, in 1996 and 1997, Honda revealed P2 and P3 respectively. In particular, P3 was characterized with a sophisticated balance control mechanism allow it to perform complex actions such as walking sideways and kicking. Several interesting example come from also the arm control and exterous manipulation research area. From SAIKA, a ligh-weight full-sized humanoid robot developed at the University of Tokyo to ROBONAUT's upper body by NASA Johnson Space Center. From Robot-Human interaction investigators came KISMET, an head with eyebrows, eyelids, ears and mouth, etc. It was an MIT project for robot training. WENDY, a human symbiotic robot that consisted of two anthropomorphic arms, a head and torso; it had wheels intead of legs. It cames from S. Sugano Laboratory at Waseda University, as HADALY-2, an other humanoid robot designed for the purpose of interactive communication with humans. Human-robot interaction plays a crucial role in the burgeoning market for intelligent personal, service and entertainment robots. One example is MINERVA, a popular tour guide at the Smithsonian National Museum of American History. HARIS, a robotic arm and human interface designed to help disabled people move and fetch objects. URSULA, a full-sized robot Female Android that is remote controlled which can walk, talks, dances, plays music and more. It was designed by Utah based company Sarcos. This company has developed some of the world's most sophisticated humanoid robots and virtual reality interfaces. Sarcos also developed DB, a robot that used also by the ERATO brain project in Japan to enable motion learning. With its 20 DoFs, 80 Kg of weight and 1,85 meters of body is one of the most capable robots in history. More recently, a number of smaller, commercially available humanoid robots have emerged from Honda, Sony and Fujitsu: ASIMO, HRP-2P and HOAP-1, among others.

After this brief overview about humanoid robotics history, we come to the topic of this work: the movement, and in particular the legged locomotion and its simulation on a flexible humanoid robot system of new-generation. Moreover, simulating and modeling based on musculoskeletal human model. There are three main reasons for exploring the legged locomotion [4]. The first reason is to develop vehicles that can move on uneven and rough terrain. The second reason is to understand human and animal locomotion mechanics. The third reason which motivated the study of legged locomotion is the need to build artificial legs for amputees. The vast majority of humanoid and bipedal robots controls the joint angle profiles to carry out the locomotion. Active walking robots (robots with actuators) can do the above task with reasonable speed and position accuracy at the cost of high control efforts, low efficiences and most of the time unnatural gaits. WABIAN-2R is among the most successful bipedal walking humanoid robots. On the other hand, the state-of-the-art is represented by "*passive dynamic walking*" robots; mechanical contraptions that demonstrate extremely natural walking motions without the need for any control action. Three successful dynamic walking robots are the Cornell Robot, Denise and Toddler. The main goal of developing dynamic walking robots is to increase the efficiency of locomotion. Existing legged robots lack energy-efficiency, performance and adaptivity when compared with situations that animals cope with on a routine basis. Bridging the gap between artificial and natural systems requires not only better sensorimotor and learning capabilities but also a corresponding motion apparatus and intelligent actuators. A key prerequisite of versatile and energy-efficient legged robots that move in a-priori unknown environments are proper actuation modules [47]. Recent research has focused more and more on actuators with adaptable compliance that can change joint stiffness in order to adjust the overall leg properties with respect to robustness, energy efficiency and speed of motion, such as the MACCEPA which was implemented in the biped VERONICA [48]. A further concept to adjust joint stiffness and nominal angles was that of the pleated pneumatic muscles (PPAM) as installed in the biped LUCY. The first robot that was capable of jumping, which is a prerequisite for the ability to run, emerged from Raibert's pioneering work and consisted of telescopic springy legs. But the very first robot that was capable of both energy-efficient and human-like walking and jogging was the bipedal robot "**Jena Walker II**", an elastic, biologically inspired, three-segmented robot. In order to further investigate the realization of a human-like robot with human-like locomotion capabilities, recently the project **BioBiped** has been launched, in which the SIM Group of TU Darmstadt collaborates with the Locomotion Laboratory of University of Jena. One of the main aspect of this project is the energy-efficiency and affects of passive tendon-like structures. The use of springs in legged locomotion is generally accepted as important and has been promoted very early and subsequently deepened and promote by M. M. Williamson, J. E. Pratt, B. T. Krupp and many others, thanks to their works about *Elastic Series Actuators*. On this way, understanding the basis of human movement and reproducing it in robotic environments is a compelling challenge that has engaged a multidisciplinary audience. The understanding of how muscles are activated to actuate the human body will directly allow designing motion and balance controller to move humanoids in a more sophisticated way. However, synthesizing detailed descriptions of the elements of the neuromusculoskeletal (NMS) system with measurements of movement to create an integrated understanding of normal movement and to establish a scientific basis remains a major challenge. To achieve this goal, a theoretical framework is needed, and in the last decade several researchers have turned their attention to the realization of simulation platforms that give a such framework which must reveal the cause-effect relationships between neuromuscular excitation patterns, muscle forces and motions of the body. A dynamic simulation of movement that integrates models describing the anatomy and physiology of the elements of the NMS system and the mechanics of multijoint movement provides such a framework.

1.1 Statement of the Problem

This work presents the design and development of a new-generation biped robot. Its modeling and simulation has been realized by using an open-source software to create and analyze dynamic simulation of movement: OpenSim [73].

This biped robot has been projected by *Fuben He*, a PhD candidate of Mechanical Engineering. He studies at Dalian University of Technology (DUT), placed in Dalian City, Liaoning Province, China. He cames an year at the Department of Information Engineering as visiting student and this work burns from the collaboration with him.

Our model aims to be used as an innovative approach to the study of a such type of robot in which there are series elastic actuators represented by active and passive spring components in series with motors. It has provided of monoarticular and biarticular joint in a very similar manner to human musculoskeletal model. We have developed several model of this biped robot that allow us to study its movement from four major point of view: **angle-driven**, by prescribing the angle to each joint, **torque-driven** with and without springs, by prescribing force and moment to each joint and **muscle-driven** by using Compute Muscle Control (CMC), a built-in tool of OpenSim. In this way we can compare in a more exact manner what happens during human movement with that happens during robot movement and consequently improve the design of motion and balance controller in the real robot.

1.2 Novelty of this Research

This work proposes a novel approach to humanoid robot modeling that combines together the physiological accuracy of the model proposed by the biomechanics for studing of human movement, to the fast operation of those proposed by robotics researchers.
This work also demonstrates the efficacy on the use of a tool like OpenSim, originally thought for the study of human and animal movement, into the development of a humanoid robot. In fact, except for a previous model of Robovie-X that however isn't a robot provided of elastic actuators, this is the first robot studied with a similar approach, for what in our knowledge.
In this work we shows that it is possible to move our robot by actuating its motors and exploiting the compliant behavior given by its springs. Furthermore, with our muscle-driven model we will also exploit the possibility to move our robot by actuating its artificial muscles modeled with parameters accurately estimated from human musculoskeletal model. This fact open a possible way to create an highly performant human-machine interface where a man and a robot are in the same loop. With this approach a man can potentially control a robot such as presented in this work by using a NMS model. A such model can be seen as an extension or evolution of how done by Massimo Sartori in his PhD thesis in which he described the man in the loop approach using an EMG-driven NMS model to control a powered orthosis [55].

1.3 Delimitations

This work has to be seen as a starting point for more detailed studies in the future. So, we must to underlying some limitation given by the lack of an exact modeling of the real spring that will be assemble in the real robot. Our springs constitute only a good approximation of that is the reality. The same thing is valid for how regard the motors mechanical properties. On the other hand, for how regard the muscle-driven model, it is only a possible vision of that might be the future because actually there aren't actuators characterized by the human muscle physiologic and mechanic properties. Finally, in our work there isn't a model of contact between foot/toe and ground.

1.4 Thesis Overview

In chapter 2, we provide a brief background about movement modeling from human and robot point of view respectively.
In chapter 3, comes the time of simulation environments for humanoid movement. After an overview of the major robot simulators we focus our attention on the OpenSim platform that is also the software used for our studies.
In chapter 4, the elastic bipedal robot design is presented by deepening the state-of-the-art of musculoskeletal models in human and robot research areas and by discussing and comparing their applications.
In chapter 5, we continue with the description and comparison of different elastic actuation models. Finally a brief discussion about the expected control system design for our biped robot.
In chapter 6, a detailed description of the implementation of the robot model into OpenSim by reporting our analysis and comparison.
In chapter 7, finally, we draw some conclusions, but above all, we discuss in detail the possible future work.
At the end of this paper you can find also some appendix about model parts, assembly phases, code examples and the bibliography.

Chapter 2

Movement Modeling

The science that studies movement is generally driven by observation, but this one alone cannot explain principles of human and animal movement. Computer simulation and biomechanical modeling complement observations and provide a framework for building, validating and studying biological models. Furthermore, common access is needed so that investigators can contribute models to a broader community by exchangeing, testing, analyzing and improving through a multi-institutional collaboration [73, 74, 44]. Musculoskeletal modeling and simulation has high potential to improve patient care and reduce treatment costs by elucidating cause and effect relationship in individuals with neurological and musculoskeletal impairments and by predicting effective surgical [75] and rehabilitation treatments. Dynamic simulations of movement allow one to study neuromuscular coordination, analyze athletic performance and estimate internal loading of the musculoskeletal system. Muscle-actuated dynamic simulations are becoming a viable approach for determining how the elements of the musculoskeletal system interact to produce movement [6].

The challenge of ***synthesizing motion behaviors*** is a long-standing problem also within the robotics community. Motion tracking is a vital component of developing intelligent autonomous robots [53]. The datas recorded by tracking the precise position and orientation of points of interest at high frequency are used for several type of applications, from surveillance, control to analysis applications. A survey of advances in vision-based human motion capture and analysis can be found at [76]. From the older optical motion capture system with Pan-Tilt camera tracking [43], among markerless motion capture systems to study musculoskeletal biomechanics [70] and action capture systems with accelerometers which are ideal for a wearable performance animation system [67]. And still, low-cost systems as those presented in [8, 69] or those thought for identification and visualization of human segment parameters and mass properties [31, 30, 32, 33]; to more recent work where a multiple camera system with real-time volume reconstruction for articulated skeleton pose tracking [85]. Is note worth, from a robotics point of view, also that the use of such tools improves the possibility to control an external device, through the use of a range of analysis techniques [10].

Research in ***rehabilitation robotics*** strongly relies on the integration of information about the human physiology into assistive device control systems for the purpose of improving the interaction between the human and the machine. The Virtual Physiological Human (VPH) for example is an initiative, strongly supported by the European Com-

mission (EC), that seeks to develop an integrated model of human physiology at multiple scales from the whole body through the organ, tissue, cell and molecular levels to the genomic level. This is projected, since 2005, to be "*a methodological and technological framework that, once established, will enable the investigation of the human body as a single complex system*" [56].
The ***relationship between humans and machines*** includes also the understanding of the so-called somatosensory information. They includes all the stimuli sensed by the sensory organs located both outside and inside the body: tension, length and velocity of the muscles, tension of the tendon and ligaments, pressure of the cartilages and stress of the bones. Among others, Y. Nakamura and K. Yamane have explored the possibility of real-time computation of the somatosensory information by combining the motion-capture system, the musculoskeletal model of the human body and the efficient computation of kinematics and dynamics. Citing Nakamura et al.:"*When the man-machine interface obtains access to the somatosensory information, machines would make the first step to understand humans*" [83].
The design of ***intelligent assistive devices*** such as powered orthoses or exoskeletons requires a deep understanding of what force the patient's muscles are able to produce. Such knowledge will allow better defining the dynamics and the magnitude of support the machine will have to provide the user with. Power Assist System HAL, for gait disorder person is one of the most important project in this research area. The HAL is a walking aid system which capable of allowing the user to problem to movements such as standing up, sitting down, and going up and down stairs. With HAL-3, Y. Sankai et al. have proposed also a calibration method to identify a constant relating Electromyographic (EMG) signal to joint torque, and to realize an apparatus that enables power to be used for walking and standing up according to the intention of the operator [37, 36, 35]. A survey of the history and state-of-the-art of lower limb exoskeletons and active orthoses can be found at [3]. Also the new-generation of humanoids robot is moving toward human-inspired research. Several technologies required for this new kind of robots reached the necessary level of performance, e.g., computing power, communication technologies, sensors and electronics integration. Rich sensory information, lighweight design and soft-robotic features are required to reach the expected performance and safety during interaction with humans or in unknown environments. Many examples come from [1, 16, 34]. Furthermore, the ***new-generation of humanoids robot*** will have highly complex skeletal structures and actuator systems that increasingly resemble the human musculoskeletal system [38]. The understanding of how muscles are activated to actuate the human body will directly allow designing motion and balance controller to move humanoids in a more sophisticated way. Hydraulic, pneumatic, motor/gearbox, series-elastic electroactive polymer-based, springs, chemical-based and many other actuation schemes are also at varying stages of research and development [40, 49, 71].
In this chapter we provide an overview of the most popular approaches to human movements by reporting and describing some basic concepts and tools that allow these studies. Then we will also see in more detail the musculoskeletal models commonly used and we will provide an overview about muscle and tendon modeling. In this chapter we also discuss about the robot movement and, expecially, we treat some basic concept of robot locomotion before to concentrate our attention on legged locomotion and bipedal locomotion models.

2.1 Neuromusculoskeletal Human Modeling

There are several bodies of research about this topic. K. Yamane, Y. Nakamura et al. [50, 59, 22] have developed the first whole-body musculoskeletal human model with the aim of an essential tool for analyzing and simulating human motions in fields where whole-body coordination plays an important role. Such models are used in sport science, rehabilitation robotics or intelligent assistive devices such as powered orthoses. On the other hand, a wide range of more detailed neuromusculoskeletal (NMS) model but only of lower limbs has been provided in literature during the last years; among others, a significative example comes from [55] in which a neuromuscular Human-Machine interface for applications in rehabilitation robotics is presented together with novel techniques to estimate the musculotendon kinematics and to calibrate an elastic tendon NMS model. Work based on the NMS model developed by Lloyd et al. [77, 13, 11, 14], which is an EMG-driven NMS model composed of four fundamental componenets: anatomical musculoskeletal model, muscle activation model, elastic tendon Hill-type muscle model and calibration. It was created to estimate individual muscle force and subsequently joint moments and soft tissue loading, in conjunction with raw EMG and segmental movement. A body of effort has been also devoted to developing efficient algortithms for kinematics and dynamics computation of robotic mechanisms, and has been successfully applied to control and simulation of complex mechanisms including industrial manipulators and humanoid robotis. The human body has a number of different properties from robotic systems in its complexity, actuators, and controllers. Human joints are primarily actuated by muscles. As a result human movement is generated. The new-generation of humanoids robots have highly complex skeletal structures and actuator systems that increasingly resemble the human musculoskeletal system. The understanding of how muscles are activated to actuate the human body will directly allow designing motion and balance controller to move humanoids in a more sophisticated way. It is still an open research issue to understand the mechanism for generating and coordinating human motions. If in the brain science community, researchers have been tried to analyze and model how the brain coordinates the whole-body motion, on the other side, in the biomechanics community, the dynamics computation and motion analysis using musculoskeletal models have been investigated.

2.1.1 Whole-Body Musculoskeletal System

The musculoskeletal human model consists of a ***musculo-tendon network*** and a ***skeleton***. The skeleton, a set of bones grouped into a suitable number of body parts, is a set of rigid links connected by mechanical joints; the musculo-tendon network is composed of wires representing the elements to drive and/or constrain the bones including muscles, tendons, ligaments and cartilages [50]. There are various types of muscles, tendons, and ligaments. We can classify them into the following patterns: one to be replaced by a simple wire that connects an origin and an end; one to be replaced by a wire that has an origin, via-points, and an end; one to be replaced by several wires forming a fork at a virtual bone; one to be replaced by a set of simple wires; one to be replaced by a combination of the above. Each attribute of these elements is commonly modeled as follows:

1. **Bone**: Rigid-link element with mass and inertia;
2. **Muscle**: Active constrictive wire actuator;
3. **Tendon**: Passive constrictive wire coupled with a muscle to transmit its power;
4. **Ligament**: Passive constrictive wire that connects multiple bones to constrain their relative movement;
5. **Cartilage**: Passive linear spring with zero nominal length that connects multiple bones and constrains their relative movement.

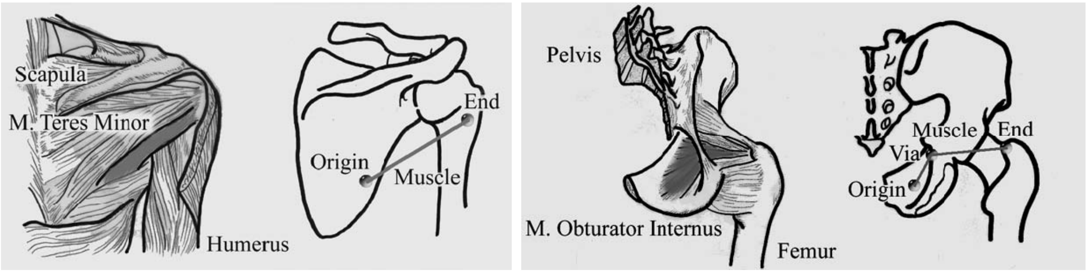


Figure 2.1: Two models: the first of simple muscle and the second of muscle with via-point [83].

Skeleton Model is realized with polygon geometric data of the bones commercially available for computer graphics, but there are also set of polygon data more or less detailed, obtained by reconstructing the geometry from computer tomography (CT) cross-sections of a human skeleton model. Both model are generally based on the standard body of European male. For the musculotendon model each wire has a pair of start and end points, and any number of *via-points* where the wire can slide back and forth without friction. In some musculoskeletal human model muscles can also actively change their length. The tension is often assumed uniform throughout the wire, and the via-points change only the direction of the wire without friction.
With reference to [50, 83], for example, the model covers most of the parts of the musculotendon networks except for those in the head, hands and feet but it includes all the minor muscles which are usually ignored in simplified models, allowing more precise computation of the tensions of individual muscles. In addition to the quantitative difference, the qualitative feature of the model is the introduction of ***virtual links*** for modeling branched muscles.
This model should be able to approximate the function of most types of muscles and tendons. However, it does not consider the following phenomena that might occur in the real human body:

- realtime addition or removal of via-points: moving the joints may introduce new via-points due to new contacts between a muscle and bone, or eliminate a via-point by taking off a muscle from a bone;
- interwire contatcs: it does not model the contact forces between multiple wires, although it does consider those between wires and bones by using the via-points.

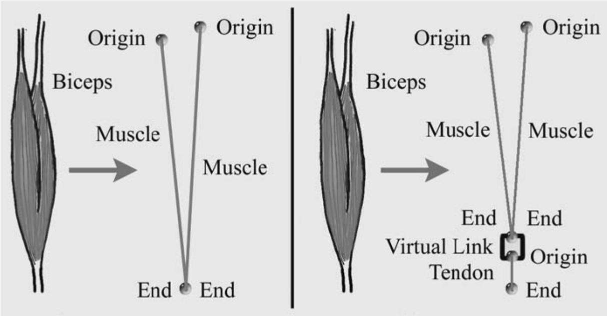


Figure 2.2: Two models of a muscle with furcation [83].

Table 2.1 summarizes the complexity of this full musculoskeletal model shown in Figure 2.3.

musculo-tendon network	
muscles	997
tendons	50
ligaments	117
cartilages	34
total wires	1198
muscle groups	78
virtual links	72

skeleton	
bones	200
bone groups	53
total DOF	155

Table 2.1: Complexity of the musculoskeletal model [83].

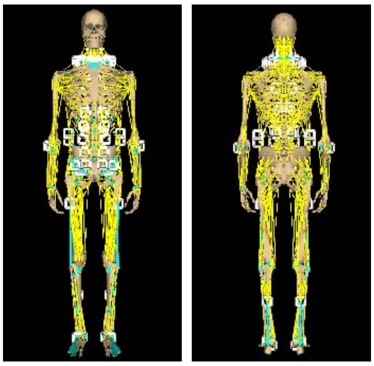


Figure 2.3: The musculoskeletal human model; yellow: muscles, blue: tendons and ligaments, white square: virtual links [83].

2.1.2 Muscles and Tendons

Muscles and tendons are the interface between the Central Nervous System (CNS) and the articulated body segments. The CNS excites muscles that subsequently develop forces that are transmitted by tendons to the skeleton, and so the body can performs a motor task.

Understanding how muscles activate and generate force about multiple joint and propel the human body towards a specific motion would significantly impact several research areas. The model of muscle and tendon to be invoked should depend on the objective. If the fundamental physical and chemical (micro) properties of muscle and tendon are to be understood, then the ***reductionist approach*** is clearly justified, to emphasize the study of an elemental unit of muscle or tendon tissue. If the I/O (macro) properties of the tissue are needed, and if the structure and function of the tissue are completely unknown, then the ***black-box approach*** is justified.
Substantially, all the researchers who study intermuscular coordination use a varition of the classical model of muscle developed in the first half of this century by Hill, Wilkie, and Ritchie and Wilkie. The brief background introduced in this thesis is based principally on [25].

Activation and Contraction Dynamics

The dynamics of muscle tissue can be divided into activation dynamics and contraction dynamics. Activation dynamics corresponds to the transformation of neural (or artificial) excitation to activation of the contractile apparatus, and muscle contraction dynamics to the transformation of activation to muscle force.
The steady-state property of muscle tissue is defined by its ***isometric force-length*** (fl) curve. This property can be studied when activation $a(t)$ and fiber length L^M are constant. In particular, we have full activation (i.e. $a(t) = 1$) when muscle tissue has been maximally excited for a long time. On the other hand, when muscle tissue has been neither neurally nor electrically excited for a long time then it is said to be inactivated or passive ($u(t) = a(t) = 0$). Active muscle force is the difference in force developed when muscle is activated and when muscle is passive. If we call L_0^M the ***muscle fiber length*** or ***optimal fiber length*** (the length at which active muscle force peaks), then the region where active muscle force is generated is (nominally) $0.5L_0^M < L^M < 1.5L_0^M$. L_0^M is also the shortest length at which passive muscle tissue develops force (nominally). The fl property of less then fully activated muscle tissue can be considered to be a scaled version of the fully activated one. In 1966, some experiments revealed that when fibers are shorter than optimal length, the force a muscle generates when maximally activated increases with fiber length (the ascending limb of the fl curve). Beyond the optimal length, the maximal active force a muscle generates decreases with fiber length (the descending limb of the fl curve), and muscle generates passive force. Near optimal fiber length, the muscle generates relatively consistent force when maximally activated (the plateau of the fl curve).
When muscle tissue undergoes an ***isotonic contraction***, or constant pull (tension), it first shortens and then stops. The length at which such a force can be sustained in steady-state corresponds to the length at which shortening terminates. By subjecting muscle to different tension and by imposing $0.5L_0^M < L^M < 1.5L_0^M$, an empirical ***force-velocity*** (fv) relation can be obtained from a set of length trajectories. In particular, when fibers shorten during activation (concentric), force generation decreases with increasing shortening velocity. When fibers lengthen during activation (eccentric), force generation increases with lengthening velocity. This increase in force is most pronounced at low lengthening velocities; at high lengthening velocity, the increase in lengthening

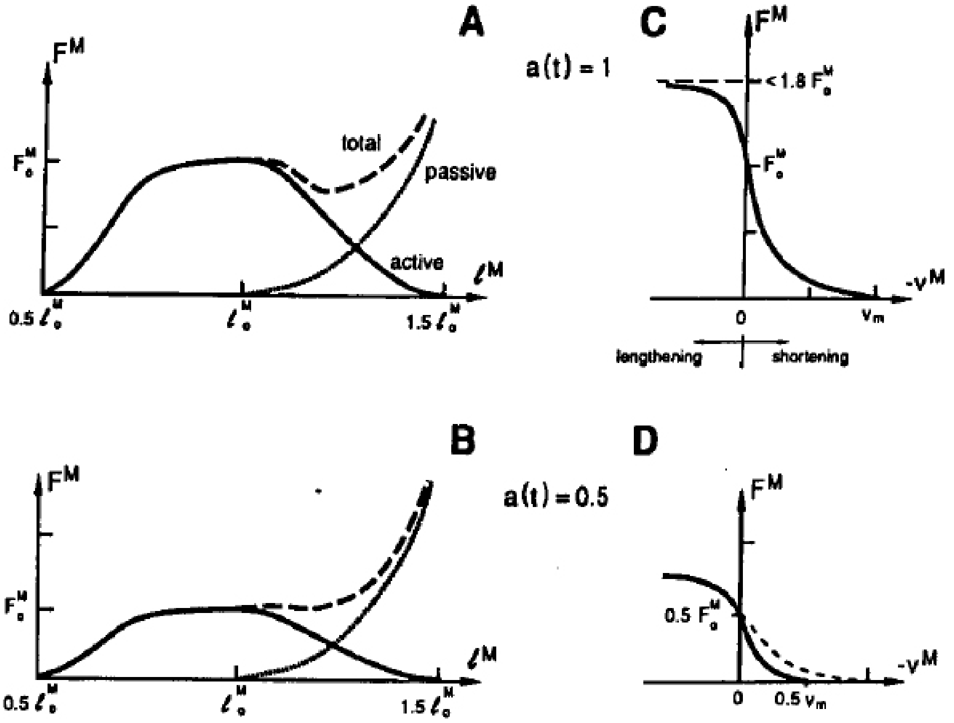


Figure 2.4: (A) Isometric Force Length (fl) relation of muscle tissue when passive (dotted curve) and when fully activated (dashed curve). Net force contributed by the activation of muscle tissue is the difference, called active force (solid line). By definition, peak active force F_0^M is developed when fibers are at their optimal length (i.e., when $L^M = L_0^M$). (B) Less than fully activated muscle tissue develops proportionately less active force (solid line), but passive force is unchanged (dotted line). (C) Force Velocity (fv) relation of fully activated muscle tissue when fibers are at a length $L^M = L_0^M$. An applied constant force less (greater) than F_0^M causes muscle tissue to shorten (lengthen). Limits exist to force generation ($<1.8F_0^M$) and to muscle fiber shortening (v_m). v_m is called the maximum shortening velocity. (D) Less than fully activated muscle tissue has been assumed either to shorten at a rate less than v_m when unloaded [25].

velocity has less effect on force production.
Finally, at optimal fiber length L_0^M, a maximum shortening velocity (v_m) can be defined from the fv relation. Even when fully activated, at v_m, muscle cannot sustain any tension. The ***mechanical power output*** ($F^M \cdot -v^M$) that active muscle delivers is determined by the shape of the fv curve. During shortening, muscle delivers power (positive power), with peak power output occuring when muscle shortens at $\approx 0.3v_m$.
Generally, all computer models of muscle used in studies of muscle coordination employ the same shaped fv curve. Common assumptions are:

- the fv relation scales with length and activation in one of two ways (i.e., either the velocity-axis intercept remains constant under all conditions or decreases with $a(t)$ and L^M);
- no discontinuity in slope at F_0^M (***peak isometric active-muscle force*** or ***maximum isometric force***) exists, even though experiments and cross-bridge theory

suggest one;

- the fv curve at any instant is unaffected by preceding events, even though it is known that prestretched muscle tissue subsequently shortens faster.

Mechanical (Hill-type) Model of Contraction Dynamics

This is a conceptual model in which the contractile properties of muscle tissue can be represented by a ***force-length-velocity*** (flv) relation controlled by muscle activation. Total muscle force F^M is the sum of passive force F^{PE} and active force F^{CE}. Structures responsible for these forces are called the passive element (PE) and the contractile element (CE). Force F^{CE} depends on muscle fiber length L^M, velocity v^M and the state activation of the muscle fibers $a(t)$. Commonly, Hill's equation is modified and used as the expression, though nothing precludes the use of other expressions.
Sometimes a ***muscle elastic element***, distinguischable from tendon elasticity, is included in series with the CE. Estimates of energy stored in muscle cross-bridges compared with energy stored in tendon is desired in studies of the biomechanics of movement, and this is the motivation to separate muscle elasticity from tendon elasticity. Cross-bridges store energy because active muscle tissue exhibits stiffness that arises from the cross-bridges themselves. However, the energy stored in cross-bridges is expected to be very small compared with the summed energy stored in the external and internal parts of tendon, in all but short tendon actuators. Thus, for many actuators, tendon compliance dominates and muscle ***Series Elastic Element*** (SEE) can be neglected.

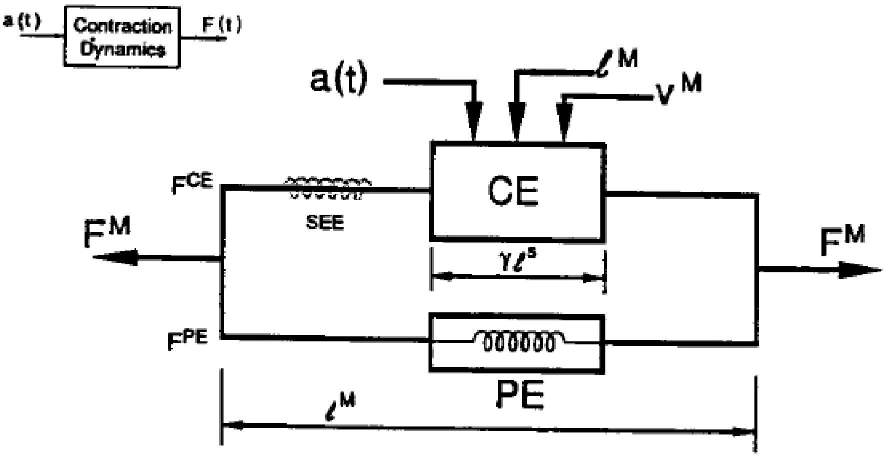


Figure 2.5: Hill-type model for contraction dynamics of muscle tissue (inset). Total muscle force F^M is the sum of passive force F^{PE} and active force F^{CE}. Structures putatively responsible for these forces are called the passive element (PE) and the contractile element (CE). Force F^{CE} depends on muscle fiber length L^M and velocity v^M, and the state of activation of the muscle fibers $a(t)$. Some models include a muscle series element (SEE). The length of the CE is the sum of the lengths of the γ sarcomeres of a muscle fiber and differs from muscle fiber length L^M by the length of the SEE [25].

2.1.3 Properties and Models of Tendons

Tendon, as generally defined, consists of a portion external to muscle and a portion internal to muscle. Common assumption are that:

- the same strain is experienced throughout internal and external tendon;
- the material (stress-strain) properties of external and internal tendon are the same.

Tendon strain ϵ^T is defined by the amount of tendon stretch relative to its resting, or slack length, $\epsilon^T = \Delta L^T / L_S^T = (L^T - L_S^t)/L_S^T$ where L_S^T is the length on elongation at which tendon just begins to develop force. L_S^T is called ***tendon slack length***. On the other hand, ***tendon stress*** σ^T is defined by the ratio of tendon force F^T to tendon Cross-Sectional Area A^T, $\sigma^T = F^T/A^T$.

One of the most important property relevant to coordination studies is the ***stress-strain*** relation (σ^T vs. ϵ^T curve). Tendon is assumed to be elastic or viscoelastic in almost all models used to study coordination, and, in some, its elasticity is assumed to be linear as well. In many models tendon elasticity is combined with the elasticity of muscle, and this overall musculotendon elasticity is defined and referred to as the SEE. Once ϵ_0^T is assumed and σ_0^T is specified, the generic force-strain (fs, $\tilde{F}$ vs. ϵ^T) curve can be found form the stress-strain curve of tendon by using σ_0^T to normalize stress. Data from many studies suggest that an appropriate (nominal) value for ϵ_0^T is 3.3% (2 - 9%), corresponding to a nominal value for σ_0^T of 32 MPa (14 - 84 MPa). These values for σ_0^T and ϵ_0^T give a safety factor of 1.9 to 2.7 times for tendon failure, since the highest force expected in tendon is 1.3 to $1.8F_0^M$ and since tendon fails at about 112 MPa and 10% strain.

In summary, tendon can be defined by a generic dimensionless fs curve that is musculotendon independent. To obtain the fl curve (F_T vs. L_T curve) of a specific tendon, from the fs curve, only two parameters are needed: L_S^T, tendon slack length (tendon-specific) and F_0^M, peak isomentric active-muscle force (muscle-specific).

Muscle Fiber Pennation

Muscle fiber pennation must be very high to affect the static and dynamic properties of actuators. The more pennated muscle fibers are the larger the change in musculotendon length caused by a change in the fiber length. This fact implies that the rate of change of musculotendon length is greater than the rate of change of muscle fiber length. The effects of pennation on musculotendon properties are significant only when muscle fibers are highly pennated (more than 23° - 25°).

2.1.4 The Musculotendon Actuator

Muscle and tendon don't work together only as an actuator, but also with the dynamics of the body segments. The force generating capability of an actuator is affected by its length L^{MT} and velocity v^{MT}, which depend on the position and motion of the body segments. In turn, body-segmental kinematics depend on the force F^T of each actuator. In this way, muscle, tendon and body segments constitute a coupled, Multiple-Input Multiple-Output (MIMO) feedback system.

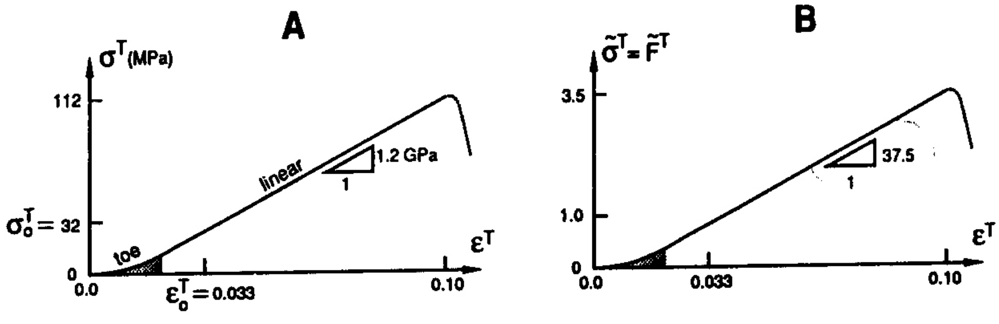


Figure 2.6: Material properties of tendon. (A) Nominal stress-strain curve (σ^T vs. ϵ^T curve). Tendon is more compliant in the toe region than in the linear region, where its tangent modulus is 1.2 GPa. The strain in tendon when $F^T = F_0^M$ is defined as ϵ_0^M (nominally, $\epsilon_0^T = 0.033$). σ^T is the stress in tendon when $F^T = F_0^M$ (nominally $\sigma_0^M = 32$ MPa). (B) Generic, dimensionless force-strain curve ($\tilde{F}^T$ vs. ϵ^T), which is also the normalized stress-strain curve. $\tilde{F}^T$ is the force in tendon relative to F_0^M ($\tilde{F}^T = F^T/F_0^M$). $\tilde{\sigma}^T$ is tendon stress normalized by σ_0^T ($\tilde{\sigma}^T = \sigma^T/\sigma_0^T$). Defining A^T to be the cross-sectional area of tendon, $\tilde{\sigma}^T = \tilde{F}^T$, since $\tilde{\sigma}^T = \sigma^T/\sigma_0^T = \sigma^T/(F_0^M/A^T) = F^T/F_0^M = \tilde{F}^T$. Tendon strain ϵ^T is defined by the amount of tendon stretch relative to its length, i.e. $\epsilon^T = \Delta L^T/L_s^T = [L^T - L_s^T]/L_s^T$, where L_s^T is the length on elongation at which tendon just begins to develop force (called tendon slack length) [25].

Strictly from the musculotendon actuator point of view, this system presents some interesting characteristics. Tendon interacts with only the contraction process of the muscle tissue and it has no interaction with the muscle activation process because activation is assumed to be uncoupled from the subsequent mechanical events. Musculotendon length L^{MT}, velocity v^{MT} and force F^T affect only musculotendon contraction dynamics. The understanding of musculotendon contraction demands knowledge of how tendon compliance affects the dynamics.
An individual actuator can be modeled by specifying five parameters, one tendon-specific and four muscle-specific:

1. tendon slack length (L_S^T);

2. peak isometric muscle force (F_0^M);

3. optimal muscle-fiber length (L_0^M);

4. optimal muscle-fiber pennation angle (α_0);

5. time scaling parameter derived from the maximum shortening velocity of muscle (τ_c).

From this one, a dimensionless model generic among all actuators can be formulated, since three physical quantities are associated with actuator dynamics (Force, Length and Time). Assuming that pennation angle is zero for simplicity and using F_0^M, L_0^M, τ_c as scaling parameters for force, length and time, the model has only one parameter: tendon slack length, normalized by optimal muscle fiber length, $\tilde{L}_S^T = L_S^T/L_0^M$. $\tilde{L}_S^T$ has no effect on activation dynamics because this one is assumed to be unaffected by muscle fiber

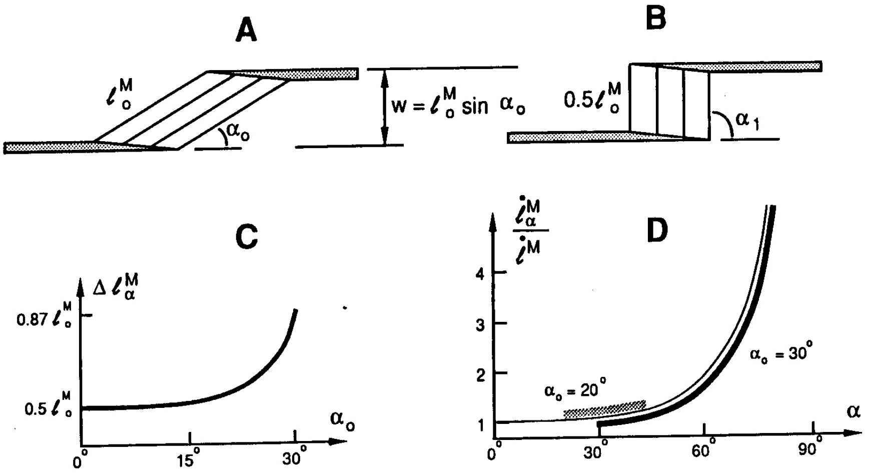


Figure 2.7: Effects of pennation on musculotendon velocity (D) and the range of musculotendon lengths corresponding to the ascending region of the muscle fl curve (C). Tendon is assumed to be compliantless so that the effects of pennation alone can be analyzed. Muscle thickness w is assumed to be constant, causing pennation to increase as fibers shorten. (A, B) Geometry of muscle fibers with respect to tendon when fibers are at their optimal length L_0^M, corresponding to a pennation angle of $\alpha = \alpha_0$, (e.g., $\alpha_0 = 30°$ in A), and when fibers are at their shortest length $L^M = 0.5L_0^M$, corresponding to an $\alpha = \alpha_1$, (e.g., $\alpha_1 = 90°$ in B) [25].

length or tendon slack length, but $\tilde{L}_S^T$ affects musculotendon contraction dynamics. From measurements of the length of its muscle-fibers (L_0^M) and its origin-to-insertion distance ($L_0^M + L_S^T$), an estimate of actuator $\tilde{L}_S^T$ can be obtained.

Because tendon is elastic, tendon compliance is proportional to tendon slack length. An actuator that has a ratio of $\tilde{L}_S^T = 1$ functions very differently than one having a ratio of $\tilde{L}_S^T = 10$. Thus, an actuator can be said to be highly compliant when it has a $\tilde{L}_S^T$ so large as to significantly affect its properties ($\tilde{L}_S^T = 10$). On the other hand, a very stiff actuator is one with a $\tilde{L}_S^T$ so small as to have no significant effect ($\tilde{L}_S^T = 1$). Is worth note that an actuator is not necessarily compliant just because it has a compliant or long tendon.

The ratio of tendon slack length to muscle fiber length $\tilde{L}_S^T$ affects also the fl relation of the musculotendon actuator. In fact, the actuator has to be longer than the length of muscle fibers by an amount equal to the tendon slack length L_S^T in order to sustain force. Otherwise the tendon is slack and muscle fibers are too short to generate force ($F^T = F^M > 0$, $L^{MT} > L_S^T + 0.5L_0^M$; or equivalently $\tilde{F}^T = \tilde{F}^M > 0$, $\tilde{L}^{MT} > \tilde{L}_S^T + 0.5$).

An other interesting aspect is the amount by which an actuator must be stretched, beyond that needed to stretch its muscle fibers, to accomodate an increase in force. For the stiff actuator, the muscle fl curve is distorted little, but the compliant actuator causes much distortion. The range of lengths where the compliant actuator operates on the ascending region of the fl curve of its muscle fibers is almost 2 times the range of the stiff actuator. In conclusion, because of tendon stretch, very long tendon actuators ($L_S^T = 15L_0^M$) need to be stretched beyond what muscle fibers alone would have to be to accomodate peak isometric force F_0^M.

2.2 Human Movement Modeling

In this section we provide a brief background of the biomechanics of human walking by inspiring to [3, 19, 20]. The period of time between any two identical events in the walking cycle is called ***gait cycle***. It is a reoccuring pattern of leg and foot movements, rotations and torques. The various events follow each other continuously and smoothly, so any event could be selected as the onset of the gait cycle. However, the human walking gait cycle is typically represented as starting (0%) and ending (100%) at the point of heel strike on the same foot, with heel strike on the adjacent foot occuring at approximately 62% of gait cycle. By contrast, the distance from initial contact of one foot to the following initial contact of the same foot is called gait stride. The shape, position and function of neuromuscular and musculoskeletal structures as well as the ligamentous and capsular constraints of the joints influence the gait characteristics.

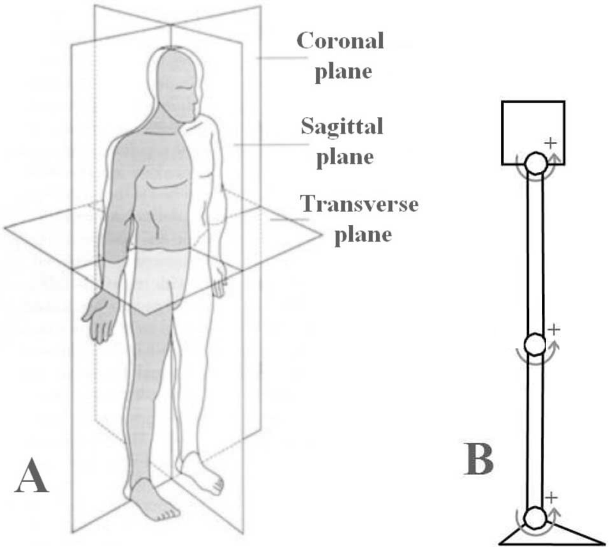


Figure 2.8: (A)Description of the anatomical planes. (B)Diagram of the leg shown in the rest position (0 degree at all joints) with the positive direction indicated. [Picture borrowed from [3] and modified]

The lower extremities and pelvis, which carry the head, arms and trunk (HAT), are reffered to as the ***locomotor apparatus***. The primary goal of this apparatus is energy efficiency in progression using a stable kinetic chain of joints and limb segments that work congruently. The Figure 2.8 shows a description of the human anatomical planes as well as a kinematic model of the human leg in the sagittal planes. Commonly, the sagittal plane is the dominant plane of motion during human locomotion. In this thesis, for how regards human models, joint motion in sagittal plane is referred to as hip flexion, knee extension and ankle dorsi-flexion. With respect to coronal plane we refer to hip internal rotation, hip adduction and ankle subtalar angle. In particular, motion of the hip in the coronal plane is referred to as abduction (away from the center of the body) and adduction as previously illustrated. Further, motion of the ankle in the coronal plane can be referred to as eversion (away from the center of the body) and inversion. The remaining Degrees of Freedom (DoF) of the hip and ankle can be simply also referred as "rotation". With this representation in mind, now we can describe in more detail the gait cycle. Generally, it is divided into two periods:

1. ***Stance***, during which the foot is in contact with the ground. This period constitutes 62% of the gait cycle;

2. ***Swing***, during which the foot is in the air. This period constitutes the remaining 38% of the gait cycle.

There is also another subdivision into the gait cycle: the period of time when both feet are in contact with the ground (***Double Support***), from the period of time when only one foot is in contact with the ground (***Single Support***). Double support configuration occurs twice in the gait cycle at the beginning and end of stance phase. It is also referred to as initial and terminal double-limb stance. Generally, the two periods of double-limb support represent 25% of the gait cycle, while running constitutes forward movement with no period of double-limb support. In fact, when velocity increases, double-limb support time decreases. For how regard the single support period, it is equal to the swing phase of the other limb during walking gait.
For completeness, we provide some other terms: the same side of the body is generally described by using the term *ipsilateral*, while to describe the opposite side of the body or the opposite limb the term *contralateral* is used; the direction of walking is referred to as the *line of progression*.

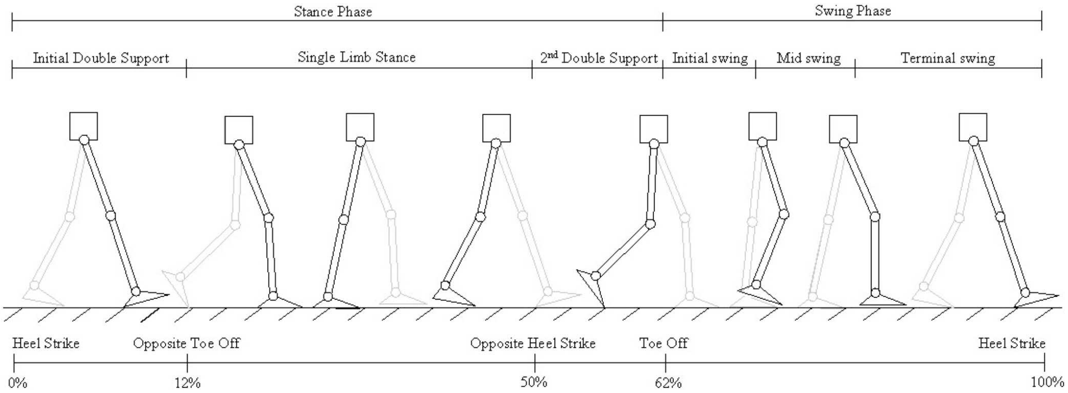


Figure 2.9: Human walking gait through one cycle, beginning and ending at heel strike. Percentages showing contact events are given at their approximate location in the cycle [3].

2.2.1 Functional Tasks and Phases of Gait

There are three functional tasks that describe a complete gait cycle. The first, called ***weight acceptance*** in which are involved two phases of the stance period: initial contact and loading response. When this task has been achieved, the individual is said to demonstrate a stable kinetic chain. This because initial limb stability and shock absorption, due to the transfer of body weight onto the limb, are required as soon as the contact between limb and ground is performed. Simultaneously, the momentum of progression must be preserved. The second, called ***single-limb support*** during which the

contralateral foot is in the swing period and total body weight is exclusively supported on the stance limb. It can be divided into three phases: midstance, terminal stance and a transitional phase, called preswing, that could be considered part of single-limb support as well. Then, the third functional task, that is called ***limb advancement***. Also for this task we have several phases: preswing, initial swing, midswing and terminal swing. Through these phases, the stance limb leaves the ground and advances forward to posture itself in preparation for the next initial contact.
A more accurate subdivision and description of the gait cycle can be provided by considering in more detail the phases of tasks. With reference to stance and swing periods, the first one consists of: *initial contact* (IC), *loading response* (LR), *midstance* (MSt), *terminal stance* (TSt) and *preswing* (PRw). On the other hand, the swing period can be divided into: *initial swing* (ISw), *midswing* (MSw) and *terminal swing* (TSw).

Initial Contact

It indicates the instantaneous point in time when occurs the instant the foot of the leading lower limb touches the ground. Initial Contact represents the beginning of the stance phase.

Loading Response

It constitutes the period of initial double-limb support, occupying about 10% of the gait cycle. During this phase, the body weight is fully transferred onto the stance limb because the foot comes in full contact with the floor. The term *foot flat* (FF) is the point in time when the foot becomes plantar grade.

Midstance

It occurs from the 10% to 30% periods of the gait cycle and represents the first half of single support. It begins when the contralateral foot leaves the ground and continues as the body weight travels along the length of the foot until it is aligned over the forefoot.

Terminal Stance

It begins with heel rise and ends when the contralateral foot contacts the ground and constitutes the second half of single-limb support. Terminal stance occurs from the 30% to 50% periods of the gait cycle. The term *heel off* (HO) is a descriptor useful in observational analysis and is the point during the stance phase when the heel leaves the ground.

Preswing

Preswing phase begins when the contralateral foot contacts the ground and ends with ipsilateral toe off. It occupies the last 12% of stance phase, from 50% to 62%. During this phase the body weight is transferred onto the contralateral limb and so the stance limb is unloaded. The termination of stance and the onset of swing is defined as the point where all portions of the foot have achieved motion relative to the floor.

Initial Swing

It begins the moment the foot leaves the ground and continues until maximum knee flexion occurs, when the swinging extremity is directly under the body and directly opposite the stance limb. It occupies one-third of the swing from 62% to 75% periods of the gait cycle.

Midswing

It occurs from the 75% to 85% periods of the gait cycle. It begins by following maximum knee flexion and ends when the tibia is in a vertical position.

Terminal Swing

From the 85% to 100% there is the final phase of terminal swing when the tibia passes beyond perpendicular and the knee fully extends in preparation for heel contact.

2.2.2 Temporal Parameters and Determinants of Gait

Gait parameters related to time are referred to as temporal parameters. Three interrelated temporal parameters are: ***stride length***, ***cadence*** and ***velocity***. Is note worth the fact that the *step length* is not a synonymous of *stride length*. This because a step length is the distance in meters from a given floor-contact point of the ipsilateral foot in stance to the same floor-contact point of the contralateral foot in stance. If we consider the segment of time in seconds taken for one step to occur and we measure it from an event of one foot to the following occurence of the same event with the other foot, then we obtain the step period.
Stride length is referred to as cycle length and contains both a left- and right-step length (in meters). Stride period or cycle time is the period of time in seconds from initial contact of one foot to the following initial contact of the same foot. A self-selected walking rhytm, then, can be described as the natural or free cadence. And we have the second temporal parameters, which refers to the number of steps taken per unit of time. It is the rate at which a person walks expressed in steps per minute. Finally, combining stride length and cadence we obtain the third temporal parameter: velocity. It is the resultant rate of forward progression. This is the best single index of walking ability and it is expressed in meters per second as the rate of chance of linear displacement along the direction of progression measured over one or more strides.
Now, if we consider the walking gait from an *energy efficency* point of view we have to pay attention to six determinants or variables that affect the energy expenditure, defined by Saunders et al. Who also defined walking as the translation of the Center of Mass (CoM) through space in a manner requiring the least energy expenditure. The six determinants of gait are: *pelvic rotation*, *pelvic tilt*, *knee flexion during midstance*, *foot and ankle motion*, *knee motion* and *lateral pelvic displacement*. They are based on two main principles:

1. Any displacement that elevates, depresses or moves the CoM beyond normal maximum excursion limits wastes energy;

2. Any abrupt or irregular movement will waste energy even when that movement does not exceed the normal maximum displacement limits of the CoM.

It is also particularly useful for the understanding exoskeleton and active orthosis design to note the *power requirements* of each joint. From walking gait data, it can be seen that, particularly at slow speeds, power at the hip is positive or near zero, power at the knee is predominantly negative (dissipative power) and power at the ankle is evently split between positive and negative. Since no net work is being done and resistance to motion is small, during steady-state (static) level ground walking the net mechanical power of the individual as a whole should be close to zero. A last observation is that the nature of the power at the individual joints can change dramatically if we consider a subject that walks at moderate to fast speeds, or on a positive incline or ascending stairs.

Pelvic Rotation

The total range of pelvic rotation is 8 degrees. The trailing extended weight-bearing limb is elastically linked through the joints of the pelvis with the advancing swing limb. Ligamentous constraints and muscular activity combine with forward momentum of the advancing swing limb to position the pelvis into 4 degrees of rotation from the line of progression prior to initial contact. During the reciprocating contralateral swing phase, the pelvis rotates in the opposite direction, first returning to its neutral position and then continuing to rotate an additional 4 degrees.

Pelvic Tilt

It is referred also as Pelvic List and is possible only in conjunction with adequate limb clearance in swing phase. At midastance, the CoM reaches its highest point as the body vaults over a planted leg.

Knee Flexion during Midstance

The stance limb enters initial contact with the knee in nearly full extension. It then flexes as the foot shifts to a plantar-grade position and continues moving into flexion until it reaches approximately 15 degrees. The knee then begins to extend but retains some flexion as it nears midstance; due to a relatively less extended knee as the tibia reaches verticality when the CoM is at its peak, the summit of the Center of Gravity (CoG) is depressed in its elevation.

If these three determinants were the only mechanisms affecting the progression of the CoM as it traverses through space, the CoG pathway would consist of a series of arcs at whose intersections an abrupt shift in the direction of the CoG would occur as it reached its lowest point. However, both foot and ankle motion as well as knee motion serve to smooth the pathway of the CoG. With these three determinants, pelvic rotation, pelvic tilt and knee flexion at midstance the vertical displacement of the CoM is sensible reduced.

Foot and Ankle Motion

At initial contact, the ankle is elevated due to the heel lever arm but falls as the foot becomes plantar grade. At heel rise, the ankle again is elevated, which continues through terminal stance and preswing. These ankle motions, coordinated with the knee and controlled by muscle action of pretibialis and triceps surae, smooth the pathway of the CoM during stance phase.

Knee Motion

Knee motion is intrisically associated with foot and ankle motion. Generally, at periods when the ankle center is depressed, the knee extends, and at periods when the ankle is elevated, the knee flexes. Knee motion, intimately associated with foot and ankle motion, smooths the pathway of the CoM and thus conserves energy.

Lateral Pelvic Displacement

To avoid extraordinary muscular and balancing demands, the pelvis shifts over the support point of the stance limb. The combination of femoral varus and anatomical valgum at the knee permits a vertical tibial posture with both tibias in close proximity to each other. The walking base or stride/step width typically is measured from one ankle joint center to the other although it often is described as the measurement from heel center to heel center.

Several gait analysis about normal and pathological function of human locomotion, conducted by J. Perry in 1992, describe the function of the rocker mechanisms. In particular, there are three mechanisms referred respectively as: heel rocker, ankle rocker and forefoot rocker, which describe the foot and ankle function with respect to the normal walking gait. As discussed in these sections, normal bipedal gait is achieved with a complex combination of automatic and volitional postural components. Normal walking requires stability, mobility of body segments and motor control to get energy-efficient forward progression.

2.2.3 Kinematics

Kinematics concerns itself with movement without consideration for the cause. The focus in gait analysis is on linear and angular displacements, velocities, accelerations and decelerations. The kinematics of walking can be quantitatively measured by means of instrumentation or qualitatively analyzed by means of observational gait assessment, such as a visual description of an individual's lower extremities, pelvis and trunk motion during ambulation. Most of what practitioners know today about kinematics of normal and pathological gait has been obtained from either an electrogoniometer, which uses electrical transducers attached to adjacent limb segments or, more likely, multicamera three-dimensional motion systems that track reflective markers placed on strategic anatomical landmarks. Motion analysis measures dynamic range of motion. Dynamic range indicates joint motion or excursion from the maximum angle to the minimum angle during a particular phase or phases in the gait cycle. Motion analysis markers are small spheres or balls placed at specific bony landmarks that, when tracked by a camera-based video system, can be used to dermine body segment and joint position.

2.2.4 Kinetics

Kinetic is the general term given to the study of forces that cause movement. Force may be defined as a push or a pull and is produced when one object acts on another. The units used to measure force are Newtons (N). Forces in walking can be internal or external. Forces such as muscle activity, ligamentous constraint or friction in muscles and joints belong to the first set, on the other hand, forces such as ground-reaction forces created from external loads belong to the second one.

The rotational potential of the forces acting on a joint is called torque, moment or moment of force. The internal joint moment is the net result of all of the internal forces acting about the joint, including moments due to muscles, ligaments, joint friction and structural constraints. The joint moment usually is calculated around a joint center. The units used to express moments or torques are Newton-meters (Nm) and for research purposes usually are normalized to the subject's body mass. Normalization is the process by which a relationship is established between initially collected data and some other basic reference data. Normalized to the subject's body mass, Newton-mass are expressed as Nm/Kg.

The term joint power is used to described the product of a joint moment and the joint angular velocity. Joint power is said to be generated when the moment and the angular velocity are in the same direction and said to be absorbed when they are in opposite directions. The units used to measure joint power is Watts (W).

The counterpart of internal joint moment is given from external moments which are the external ground-reaction force. Information on these forces is obtained from a force platform or force plate, which is a transducer set into the floor to measure the forces and torques applied by the foot to the ground. These devices provide quantified measures of the three componenets of the resultant ground-reaction force vector and the resultant torque vector about a given joint. The ground-reaction force line essentially is the vector summation of the three reaction forces resulting from the interaction between the foot and ground. The moment of force or torque is the cross product of the radius vector and the force. The radius vector, traditionally assigned the variable r, is a position vector from the point around which the calculations are made to the line of action for the force being considered, traditionally assigned the variable F. The length of r is the moment arm of the force F.

2.2.5 Electromyographics

Electromyographic (EMG) data provide important information in terms of understanding the direct physiological effect of prosthetic or orthotic design variants. EMG information is generally obtained by inserting fine wire electrodes directly into the muscle belly or by placing noninvasive surface electrodes over the muscle apex. Wire electrodes have the advantage of precise placement and are less likely to register "cross-talk" from adjacent muscle. They are essential for measuring deep muscles. Surface electrodes, on the other hand, provide a noninvasive alternative for measuring muscle activity of superficial groups.

2.3 Robot Kinematics and Dynamics

As we have seen a musculoskeletal model commonly used in literature is composed of rigid bodies connected by mechanical joints representing the skeleton, and a set of linear actuators representing the muscles.
There are two main streams of computation that correspond, in the robotics terminology, to:

- **INVERSE DYNAMICS** of musculoskeletal models - that it is used to estimate the joint torques and muscle tensions from motion, ground contact force, and/or electromyograph (EMG) measurements: the study of muscle function through the inquiry of the electrical signals the muscles emanate.

- **FORWARD DYNAMICS** computation - that it is performed to simulate the motion assuming a muscle activation pattern obtained by some control or optimization techniques.

While the inverse dynamics analysis has two inherent problems: inconsistency between the measured data due to the modeling error and indeterminate muscle tensions due to actuation redundancy, the forward dynamics computation has the main problems in control and optimization procedures, that would be difficult for complex models due to the instability and computational cost.
In the robotics field, a body of effort has been devoted to developing efficient algorithms for kinematics and dynamics modeling and computation of robotic mechanism. The pioneering work include compact representation of the joint kinematic parameters, realtime forward kinematics and inverse dynamics for control, and linear-time forward dynamics algorithms. However, applying these results is not straightforward due to the differences between the human body and robotic mechanisms. The main difference is that *the objective of the computations for human body dynamics is to know what is happening in the human body*, while that *for robotic systems is to drive the system as desidered.*
The results of analysis and simulation of the human body should therefore be both physically and physiologically consistent, although it is virtually impossible to collect all the data required to validate the results especially for human subjects.

In order to apply the robot dynamics algorithms, the *wire tensions* should be converted to *joint torques.* The conversion is commonly realized by utilizing the parameter called *moment arm* in biomechanics literature. Moment arm is defined as the distance from the joint axes to the muscle. The definition is usually limited to rotational joints and planar models because the distance is measured in the plane perpendicular to the joint axis.
Application of robot kinematics and mechanics can provide a more general way for converting the wire tensions to joint torques. The principle of virtual work and d'Alembert's Principle yield the following equation relating the wire tensions and the generalized forces:

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{f} \tag{2.1}$$

where $\boldsymbol{\tau}$ is the generalized forces, $\boldsymbol{f}$ is the wire tensions, and $\boldsymbol{J}$ is a Jacobian matrix defined as

$$\boldsymbol{J} = \partial \boldsymbol{l} / \partial \boldsymbol{q} \tag{2.2}$$

where $\boldsymbol{l}$ is the vector of wire lengths and $\boldsymbol{q}$ is the vector of the generalized coordinates [50, 83].

Inverse dynamics of musculoskeletal human models means to estimate the wire tensions $\boldsymbol{f}$ that realize a given set of joint accelerations at a given state. Because $\boldsymbol{\tau}$ can be obtained by applying any inverse dynamics algorithm for robots, this is the problem of solving the linear equation Eq.(2.1) for $\boldsymbol{f}$ with inequality constraints such as minimum and maximum muscle tensions.

Inverse dynamics of human musculoskeletal model has the following two inherent difficulties: the number of solutions can be from zero to infinity, depending upon the wire placements and number of wires, and it is very difficult to validate the results because no ground truth data are available. One of the solutions to the first problem is setting appropriate objective funcion and applying linear or nonlinear numerical optimization techniques. The second problem is still an open research issue. It would be possible to obtain the ground truth by precisely measuring the electromyography (EMG) of all muscles, but this approach is unrealistic especially for whole-body models because it requires needle EMG to access the inner muscles, as well as huge number of AD channels to obtain the data from all muscles at the same time. In addition, the tensions computed from EMG data are not reliable due to the noise in EMG measurements and uncertain muscle model parameters.

For the first problem, a formulation, proposed by Katsu Yamane and Yoshihiko Nakamura in the last year, includes a number of factors such as maximum and minimum muscle activities derived from muscle dynamics and tension distribution among agonist muscles. For the second one, they have tried to obtain both physically and physiologically reasonable whole-body muscle tensions by forming an optimization problem such that the error of the tension-torque transformation equation (2.1) becomes minimum while respecting the physiological muscle dynamics and the muscle activity information measured by EMG. The optimization is formulated as the following linear programming problem:

Find δ_τ, δ_f, δ_m, and $\boldsymbol{f}$ that minimize

$$Z = \boldsymbol{a}_\tau^T \delta_\tau + \boldsymbol{a}_f^T \delta_f + \boldsymbol{a}_m^T \delta_m \tag{2.3}$$

subject to

$$-\boldsymbol{f}_{max} \le \boldsymbol{f} \le -\boldsymbol{f}_{min} \tag{2.4}$$

$$-\delta_\tau \le \boldsymbol{\tau} - \boldsymbol{J}^T \boldsymbol{f} \le \delta_\tau \tag{2.5}$$

$$0 \le \delta_\tau \tag{2.6}$$

$$-\delta_f \le \boldsymbol{f} - \boldsymbol{f}^* \le \delta_f \tag{2.7}$$

$$0 \le \delta_f \tag{2.8}$$

$$-\delta_m \le \boldsymbol{E}_G \boldsymbol{f} \le \delta_m \tag{2.9}$$

$$\delta_m \le 0 \tag{2.10}$$

where $\boldsymbol{a}_\tau$, $\boldsymbol{a}_f$ and $\boldsymbol{a}_m$ are user-specified constant vectors with positive components, $\boldsymbol{f}^*$ is the vector of reference muscle tensions, $\boldsymbol{f}_{max} \ge 0$ and $\boldsymbol{f}_{min} \ge 0$ are the vectors of maximum and minimum muscle tensions respectively, and $\boldsymbol{E}_G$ is a constant matrix. The advantage of applying linear programming is that it is computationally efficient than other nonlinear methods such as quadratic programming. The problem is, on the other hand, that the solution often becomes both temporally and spatially jerky and therefore stressing the need to define a good set of constraints to obtain plausible muscle tensions. The details of the objective function and constraints are described in [50].

In a more general way, the forward dynamics computation of musculoskeletal models can be divided in the following three steps:

1. Compute $\boldsymbol{J}^* \in \mathbb{R}^{N_l \times N_G}$, the Jacobian matrix of the wire and spring lenghts with respect to the generalized coordinates, defined as Eq.(2.2). N_l and N_G are the number of wires and springs and DOF of the model.

2. Map the given wire tensions into the generalized force using the Jacobian matrix obtained at the point 1 above.

3. Compute the accelerations of the generalized coordinates from the generalized force using the method [84] and integrate them to obtain the whole-body motion.

On the other hand, the computation of inverse dynamics requires the following two steps:

1. Apply the standard Newton-Euler inverse dynamics computation to the given motion data, and compute the joint torques.

2. Map the joint torques to the wire tensions $\boldsymbol{f}$.

More details about specific methods are described in [83], and a comparative study about algorithms can be found at [51].

2.4 Robot Movement Modeling

Humanoid refers to any being whose body structure resembles that of a human: head, torso, legs, arms, hands. But it is also a robot made to resemble a human both in appearance and behavior. The difference between a robot and android is only skin-deep, looks exactly like humans on the outside, but with internal mechanics of humanoid robot. The main differences between humanoids and other kinds of robots (manipulators, mobile robots, multi-legged robot) are: bipedal human-like locomotion (stable gait, changing model during one/two feet support walking) and two legs + two arms + torso + head (hyper DOF system (>20) and complex kinematics and dynamics). While the industrial manipulator and mobile robots needs to adapt and drastically change their environment, the humanoids robots can work directly in the same human environment without any modification. While, if the environment is "empty" and the floor is flat, the wheeled humanoid is the best solution; if the environment is real (stairs, not flat floor, cluttered and without modification), then the biped humanoid is the best choice. Bipedism and cognition have very close relationship; the time line of human evolution shows three important events called: *Erectus*, *Pre-Sapiens*, and *Sapiens*. For the human evolution bipedism frees the hands to create tools and start cognition and the humanoids robots nowadays status is focusing more in bipedism than in cognition: stable bipedal locomotion is not totally solved, we have only good lab example, it is mandatory to have robust biped platforms in order to implement cognitive robotics and there are no human-size humanoids in the market now and probably the same in the next future. So, we are in the *Pre-Robotic stage* comparing with the human evolution. As we said previously, the synthesis of human motion is a complex procedure that involves accurate reconstruction of movement sequences, modeling of musculoskeletal kinematics, dynamics and actuation, and characterization of reliable performance criteria. Many of these processes have much in common with the problems found in robotics research, with the recent advent of complex humanoid systems.

There are research robots that can walk, jump, run, slide, etc. Also if most of these locomotion mechanisms have been inspired by their biological counterparts, there is one exception: the actively powered wheel is a human invention. This mechanism is not completely foreign to biological systems. Biological systems succeed in moving through a wide variety of harsh environments. Therefore it can be desirable to copy their selection of locomotion mechanisms. However, replicating nature in this regard is extremely difficult for several reasons. To begin with mechanical complexity that is easily achieved in biological systems through structural replication. Cell division, in combination with specialization, can readily produce a millipede with several hundred legs and several tens of thousands of individually sensed cilia. In manmade structures, each part must be fabricated individually, and so no such economies of scale exist. Additionally, the cell is a microscopic building block that enables extreme miniaturization. With very small size and weight, insects achieve a level of robustness that we have not been able to match with human fabrication techniques. Finally, the biological energy storage system and the muscular and hydraulic activation systems used by large animals and insects achieve torque, response time, and conversion efficiencies that far exceed similarly scaled man-made systems [66].

Locomotion is the complement of manipulation. While a robot arm is fixed but moves objects in a workspace by imparting force to them, in locomotion the environment is fixed and the robot moves by imparting force to the environment. However, in both cases, the goal is the study of actuators that generate interaction forces and mechanisms that implement desired kinematic and dynamic properties. *Stability*, *characteristics of contact* and *type of environment* are the case issues shared from locomotion and manipulation.

Our research interests are turned to humanoid robot locomotion and so, in the rest of this chapter we will provide a briefly background about legged locomotion with particular attention to bipedal and elastic bipedal locomotion.

2.4.1 Legged Locomotion

In general, legged locomotion requires higher degrees of freedom and therefore greater mechanical complexity than wheeled locomotion. On flat surfaces wheeled locomotion is one to two orders of magnitude more efficient than legged locomotion. But as the surface becomes soft, wheeled locomotion accumulates inefficiencies due to rolling friction whereas legged locomotion suffers much less because it consists only of point contacts with the ground. While the efficiency of wheeled locomotion depends greatly on environmental qualities like flatness and hardness of the ground, the efficiency of legged locomotion depends on the leg and body masses, both of which the robot must support at various points in a legged gait. This type of locomotion is characterized by a series of point contacts between the robot and the ground. The key advantages include adaptability and maneuverability in rough terrain, in addition a walking robot is capable of crossing a hole or chasm so long as its reach exceeds the width of the hole. A final advantage of legged locomotion is the potential to manipulate objects in the environment with great skill. On the other hand, the main disadvatages include power and mechanical complexity: we need of several degrees of freedom, the leg must be capable of sustaining part of the robots' total weight and the degrees of freedom have to be enough for impart forces in a number of different directions to achieve a high maneuverability.

A number of different leg configurations have been successful in a variety of organisms. Large animals, such as mammals and reptiles, have four legs, whereas insects have six or more legs. In some mammals, the ability to walk on only two legs has been perfected at the price of much more complex active control to maintain balance. In contrast, a creature with three legs can exhibit a static, stable pose provided that it can ensure that its CoG is within the tripod of ground contact. Static stability, demonstrated by a three-legged stool, means that balance is maintained with no need for motion. In order to achieve static walking, a robot must have at least six legs, so it is possible to design a gait in which a statically stable tripod of legs is in contact with the ground at all times.

A minimum of two degrees of freedom is generally required, in the case of legged mobile robots, to move a leg forward by lifting the leg and swinging it forward. More common is the addition of a third degree of freedom for more complex maneuvers and recently has been created bipedal walking robots with a fourth degree of freedom at the ankle joint. This joint enables more consistent ground contact by actuating the pose of the sole of the foot. Adding degrees of freedom to a robot leg increases the maneuverability of the robot and augments the range of terrains on which it can travel and the variety of gaits. But all additional joints and actuators constitutes disadvantages from energy, control and mass point of view. The number of possible gaits, in the case of a multilegged mobile robot, depends on the number of legs. The gait is a sequence of lift and release events for the individual legs. For a mobile robot with k legs, the total number of possible events N for a walking machine is

$$N = (2k - 1)! \tag{2.11}$$

2.4.2 Bipedal Locomotion

There are several theories of bipedal walking. In [9], Christopher L. Vaughan reviewes six of these theories: *evolution*, *minimizing energy consumption*, *maturation in children*, *central pattern generators*, *linking control and effect*, and *robots on two legs*.

In 1970s Honda started wide research on bipedal robots, which was followed by numerous research project performed by mainly Japan's companies and universities. Even though the research topic is relatively old, the bipedal walking robots need to be improved because the technology maturity has not been reached and this indicates that the bipedal robots overall design is a very complex task. This is particularly the case with the control aspect of the robots and it is also clear that the performance of the robot depends very much on the accuracy of the model, that is, how well the model represents the real-world system.

To understand the mechanical bipedal robots mechanics design, is necessary first to understand the bipedal walking process or bipedal locomotion. Also called walking gait, that is the repetitive sequence of leg movements to allow locomotion. The gait is characterized by the sequence of lift and release events of individual legs. As we said in the previous section, it depends on the number of legs. So, for a biped walker (k = 2), the number of possible events N is:

$$N = (2k - 1)! = 3! = 3 \cdot 2 \cdot 1 = 6 \tag{2.12}$$

where the six different events are:

1. lift right leg;
2. lift left leg;
3. release right leg;
4. release left leg;
5. lift both legs together;
6. release both legs together.

This is the reason for which many statical and dynamical models have been developed for the bipedal robots. The first bipedal walking robot from Kato, at Waseda University, in the 1970s was pioneering work on the field of two legged robots. In general, to synthesize gait and build biped walking system, two possible point of departure exist:

1. "**classical**" humanoid walking robots;
2. robots inspired by the concept of "**passive dynamic walking**" ("*limit cycle walkers*").

The standard ("classical") approach that can be used as a starting point for the construction and control of biped robots is the standard approach to robot control as applied in industrial manufactoring robots where a direct control over all degrees of freedom is ensured. The robot can be controlled to accurately follow prescribed trajectories by application of strong actuators and stiff structural components. Humanoid robots like ASIMO from Honda, SDR-3X from Sony, HRP-2P from Kawada, the robots from Toyota and LOLA from TU Munchen belong to this category of classical robots. For example, both Honda's ASIMO (mass 43 kg, height 1.2 m, velocity 0.44 m/s) and Sony's SDR-3X (mass 5 kg, height 0.5 m, velocity 0.25 m/s) are anthropomorphic robots with 24 degrees of freedom (cf. Fig. 12). Both have joint torque actuators, sophisticated feedback via accelerometers and rate sensors, plus an onboard computer to provide real-time control. The second type of robots are inspired by passive walkers. The original walkers were purely passive and walking down a slope fueled only by gravity.

Five decades ago Isaac Asimov, the science fiction writer formulated his three laws of robotics. The second of these stated that "*A robot must obey the orders given it by human beings*". One of these orders would no doubt have been to walk, a natural extension of the original function of a "robot", first introduced to the English language by the playwright Karel Capek in 1921. Based on his mother tongue of Czech, a robota is defined as a worker of forced labour. Over 30 years before Capek coined the term robot, George Fallis in the USA invented a bipedal walking toy (Fallis, 1888). The central claim of his patent stated "*This invention consists of a toy which is designed to simulate the human frame and which is a combined pendulum and rocker construction, whereby when placed upon an inclined plane it will be caused by the force of its own gravity to automatically step out and walk down the said plane*". What Fallis had described was a passive bipedal robot, where the word "passive" connotes the lack of active power. Another feature of the Fallis walker was that its gait was almost certainly static. Static walking assumes that the robot is statically stable. This means that, at any time, if all motion is stopped the robot will stay indefinitely in a stable position. A bipedal robot gait is said to be statically and a humanoid posture is said to be balanced if the ground projection of its ***CoM***, falls within the convex hull of the ***foot support area*** (the support polygon). In particular, the support area is commonly defined as the foot surface in case of one supporting leg or the minimum convex area containing both foot surfaces in case both feet are on the ground (single and double support phases, respectively); while the CoM is calculated according to its distance-weighted average location of the individual mass particles in the robot:

$$CoM = \frac{\sum P_{mi} \cdot M_i}{\sum M_i} \tag{2.13}$$

It is note worth that CoM is not a synonimous of ***Center of Pressure*** (CoP) that is the pivot point of the human/humanoid foot, the center point of the convex hull of the foot where it supports the most pressure. The CoP is calculated according to its distance-weighted average location of the individual pressures on the foot:

$$CoP = \frac{\sum P_{pi} \cdot P_i}{\sum P_i} \tag{2.14}$$

It is clear that static walking is in contrast to dynamic walking where the CoM can be outside of the support area, but the ***Zero Momentum Point*** (ZMP), which is the point where the total angular momentum is zero, cannot. The *passive* vs. *active* debate is an interesting one, as is the discourse on *static* vs. *dynamic* gait.

The idea of this type of robots is originally based on the observation that human walking looks like a double inverted pendulum and scientifically all started with an hypothesis posed by Weber and Weber as early as 1836: "*The leg can swing back and forth like a pendulum suspended from the body...Our attention is not required to produce this swinging motion*". Mochon and McMahon arrived at the same conclusion after comparing the swing leg motion with a passive double pendulum. Then, another hint, is given by Ralston who discovered that there exists an optimal walking velocity for humans; at approximately 5 Km/h the specific resistance (also termed specific cost of transport; i.e. energy cost per weight per distance traveled) is minimal, a phenomenon that indicates the useful of the natural frequencies of the mechanical system [58]. In 1989, McGeer showed that a completely unactuated and therefore uncontrolled robot can perform a stable walk when walking down a gentle slope. Since then, Ruina's group at Cornell University extended his work up to the point where the passive approach can be regarded beyond doubt as a valid starting point for bipedal gait synthesis and robot construction. His work has led to more recent efforts to explore the potential of passive gait to provide biomechanical insight (Garcia et al., 1998; Kuo, 2001). Nowadays robots like the Cornell biped from Cornell University and FLAME from TU Delft still comprise passive joints. Additionally some actuation is applied, such that they are able to walk in the horizontal plane. These robots belong to the so called group of limit cycle walkers. This definition comes from the fact that the gait of these systems appears to follow a stable limit cycle, a closed trajectory to which all neighboring trajectories converge. Thus limit cycle walkers exhibit a periodic repetitive motion, which restores itself after small perturbations have occured [12].

The benefits of the passive approach are the inherent efficiency of the walking motion, the natural-looking motions and the simplicity of the required construction. With this approach the robots are high energy efficient but there is no change in velocity or direction and their diversity is limited (rhythmic movements). Moreover, it requires a nonlinear dynamics and complex design process.

Inverted Pendulum Model

The inverted pendulum is a classic problem in dynamics and control theory and is widely used as a benchmark for testing control algorithms. It is often implemented with the pivot point mounted on a cart that can move horizontally; it may be called a cart and pole.

As we said previously, the human walking motion shows some similarities with the inverted pendulum mechanics. The pendulum pivot point is placed approximately at the CoP on the foot. The pendulum mass is placed approximately at the CoM. In this way, a simple pendulum model of bipedal walking can be represents as in Figure 2.10, where m represents the *CoM*, l is the *length of the leg* and δ represents the *stance leg angle.*

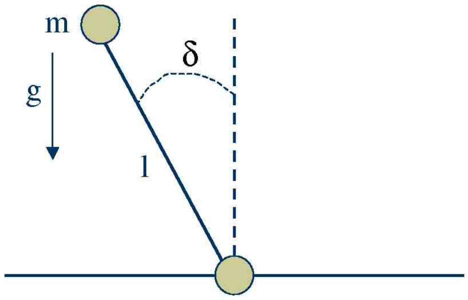


Figure 2.10: A simple pendulum model.

A pendulum has an equilibrium point in the straight up position and will accelerate in the direction of whichever side it is on. The further the mass is from the vertical, the faster it will accelerate. Note that there is no torque at the pivot point. Well, if now suppose that the mass is travelling from left to right, then there are two possibilities:

1. if the mass is on the left-hand side, it will slow down towards the vertical;

2. if the mass has passed the vertical, it will accelerate to its right.

At this point the system is converting kinetic energy into gravitational energy when it travels from left to vertical and convert it back into kinetic energy from the vertical to the right. We can describe the pendulum dynamics as follow:

- The *initial kinetic energy* is:
$$E_k = \frac{1}{2}mv^2 \tag{2.15}$$
- The *change to potential energy* is:
$$\Delta E_p = mgl(1 - cos\delta) \tag{2.16}$$
- By setting the change of potential energy equal to kinetic energy:
$$cos\delta = 1 - \frac{v^2}{2 \cdot gl} \tag{2.17}$$
- For *small angle approximation*, we get:
$$\delta = \frac{v}{\sqrt{gl}} \tag{2.18}$$

Now, if let's add a linear actuator along the length of the pendulum we obtain the situation represented on the left side in Figure 2.11. The force on the point mass lies along the length of the leg. The acceleration of the mass in the radial direction depends on the actuator force F, the gravitational force, and the fictitious centrifugal force due to the rotation of the pendulum. As previously, assume the mass of the pendulum is travelling from left to right, then with the actuator pulling the mass, the rotation motion will accelerate. On the other hand, extending the mass will decelerate the speed. This is the same as sitting on a spinning chair where the pivot point is underneath the chair, when opening the arms during spinning, the rotation will slow down, while closing up the arms will increase the rotational speed.

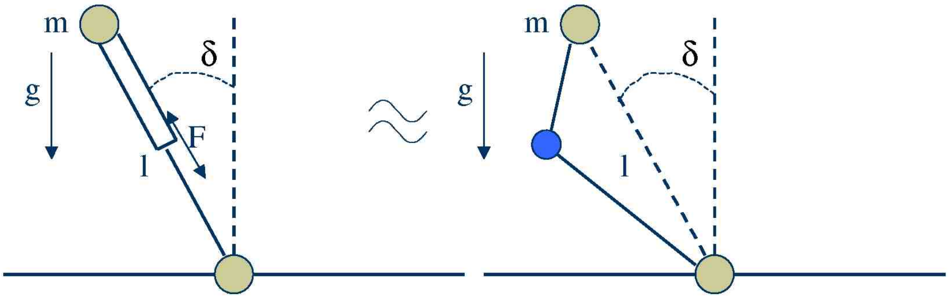


Figure 2.11: A linear actuator pendulum model, on the left side. A multi joint pendulum model, on the right side.

The next step is to analyze the multi joint model, represented on the right side of Figure 2.11. This model shares some characteristics with the linear actuator model, but it is implemented with a different mechanism. In order to transform a linear actuator model

into a multi joint pendulum model, we have to show that both dynamics are identical. We illustrate the similarities by using a two DoF multi joint model with the parameters mass point m and length l of the leg.
With different mechanical design, to obtain the same paramters we have to use inverse kinematics to calculate the angles for each individual joint. The only difference between the two models is the force gain from the actuator, for linear actuator model it is a linear force. Instead, this last model has a torque generated by the knee servo. One of the most difficult problem in robot walking is stability and now we have to balance this system to achieve stability. Controlling the speed of the rotation motion of the inverted pendulum by adjusting the length of the leg is not a well-balanced system. On the other hand, adding a foot and an ankle to the multi joint leg we obtain a better solution.

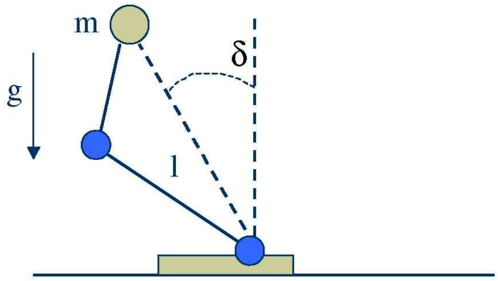


Figure 2.12: Ankle pendulum model.

In this way we obtain many benefits:

- it yields to a larger supporting area for the CoM to stay below when traveling;
- it also helps to control the speed of the pendulum and leads to a more stable system.

Balancing the mass is made by shifting it left and right from the vertical above the pivot point. The pivot point of a human is the CoP, thus: if the CoP is left of the CoM then the mass point will accelerate to the right; while if the CoP is right of the CoM then the mass point will accelerate to the left. So, by controlling the joints torque, we can arbitrarily control the location of the CoP.

Dynamical Walking Model

Irrespective of their structure and number of DoF involved, the basic characteristics of all biped locomotion systems are:

- the possibility of rotation of the overall system about one of the foot edges caused by strong disturbances, which is equivalent to the appearance of an unpowered (passive) DoF;
- gait repeatability, which is releated to regular gait only;
- regular interchangeability of single- and double-support phases.

Since, the locomotion mechanism change its structure during a single walking cycle from an open to a closed kinematic chain, all these circumstances have to be taken into account in artificial gait synthesis [57]. Biped dynamic walking allows the CoM to be outside the support region for limited amounts of time. Consequently, there is no absolute criterion that determines whether the dynamic walking is stable or not. Indeed a walker can be designed to recover from different kinds of instabilities. However, if the robot has active ankle joints and always keeps at least one foot flat on the ground, then the ZMP can be used as a stability criterion. The foot cannot be controlled directly but in an indirect way, by ensuring the appropriate dynamics of the mechanism above the foot.

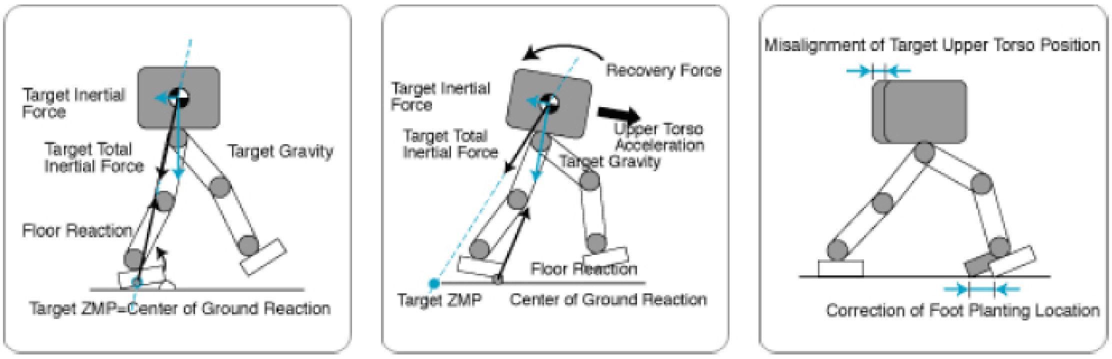

Target Inertial Force
Target Total Inertial Force
Target Gravity
Floor Reaction
Target ZMP=Center of Ground Reaction
Recovery Force
Target Inertial Force
Target Total Inertial Force
Upper Torso Acceleration
Target Gravity
Floor Reaction
Target ZMP
Center of Ground Reaction
Misalignment of Target Upper Torso Position
Correction of Foot Planting Location

Figure 2.13: Dynamic walking schema.

The ZMP is the point with respect to which dynamic reaction force at the contact of the foot with ground does not produce any moment, i.e. total reaction forces equals to zero, as stablished by Vukobratovic in 1968. It is clear that for robots that do not continuously keep at least one foot on the ground or that do not have active ankle joints (walking on stilts), the notion of support area does not exist, therefore the ZMP criterion cannot be applied. We must pay attention to the fact that the ZMP exists only when the foot is not rotating, while the CoP always exists when there is a contact between foot and ground. These two point coincide when the ZMP exists.

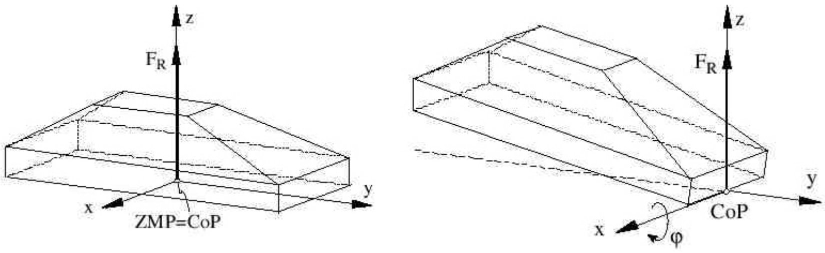


Figure 2.14: Support area schema.

The position of the ZMP is computed by finding the point (X,Y,Z) where the total torque is zero. If we consider only the ground plane, for simplicity, we can assume that Z = 0. We also assume that the robot has n links and that each link is subject to: a total force F_i applied at a point determined by the vector R_i relative to the CoG of the link; a vector T_i that determines the total motor torque applied to the link; and finally, we call R_z the ZMP vector and T the robots total torque. In this way, the force, torque and position vectors have the following coordinates: $F_i : (F_{xi}, F_{yi}, F_{zi})$, $T_i : (T_{xi}, T_{yi}, T_{zi})$, $R_i : (x_i, y_i, z_i)$, $R_z : (X, Y, Z)$, respectively. Then, the total torque is computed as:

$$T = \sum_{i=1}^{n}(R_i + R_z) \times F_i + \sum_{i=1}^{n} T_i = 0 \tag{2.19}$$

where × represents the cross product. The equation is then expanded as:

$$\sum_{i=1}^{n}(y_i + Y)F_{zi} - \sum_{i=1}^{n}(z_i + Z)F_{yi} + \sum_{i=1}^{n} T_{xi} = 0 \tag{2.20}$$

$$\sum_{i=1}^{n}(z_i + Z)F_{xi} - \sum_{i=1}^{n}(x_i + X)F_{zi} + \sum_{i=1}^{n} T_{yi} = 0 \tag{2.21}$$

$$\sum_{i=1}^{n}(x_i + X)F_{yi} - \sum_{i=1}^{n}(y_i + Y)F_{xi} + \sum_{i=1}^{n} T_{zi} = 0 \tag{2.22}$$

Making Z = 0 and solving these equations for X and Y we obtain the ZMP coordinates:

$$X = \frac{\sum_{i=1}^{n}(z_i F_{xi} - x_i F_{zi}) + \sum_{i=1}^{n} T_{yi}}{\sum_{i=1}^{n} F_{zi}} \tag{2.23}$$

$$Y = \frac{\sum_{i=1}^{n}(z_i F_{yi} - y_i F_{zi}) + \sum_{i=1}^{n} T_{xi}}{\sum_{i=1}^{n} F_{zi}} \tag{2.24}$$

The computed ZMP position is just a candidate to be a regular ZMP and its position should be compared with the real support polygon size. If the computed acting point of the ground reaction force is within the real support polygon, this point is ZMP and the mechanism is in equilibrium. If this is not the case, the ground reaction force acting point will be on the support polygon border and the distance from it to the computed ZMP position is proportional to the intensity of the perturbation moment that acts on the foot.

A very detailed elaboration of ZMP notion is given in [57], with a special review concerning "boundary cases" when the ZMP is close to the edge of the support polygon and "fictious cases" (also called FZMP) when the ZMP should be outside the support polygon. On the other hand, a geometrical interpretation of the ZMP can be found at [29].

2.4.3 Elastic Bipedal Locomotion

Elasticity in the actuation of robotic arms was for a long time seen as undesirable. When introducing a series elasticity in the joint actuation, reduced torque and force bandwidth and increased controller complexity for oscillation damping and tracking control are the result. However, research on series elastic actuators [54, 39] showed that mechanical compliance in the joint actuation can simplify force control in constrained situations, increase safety because of the low-pass filtering of torque and force peaks between the decoupled joint and gearbox, and increase performance of specific tasks because of the possibility to store mechanical energy in the elasticity. A meaningful example is given

from BioRob Arm, an equilibrium-controlled stiffness manipulator [79].
On the other hand, for how regard legged locomotion we have seen that the works in the field of legged robots can be categorized into three groups. The first one comprises conventionally built robots such as ASIMO, which are based on kinematic chains of rigid rotary joints and links. This group is currently still dominant and makes particularly use of the ZMP criterion and its variations to make sure that the robot does not fall over. The second group is dominated by the computer-controlled walking machines of M. Raibert. To achieve stable gaits, these robots use controllers based on a spring-loaded inverted pendulum model. Then, the third group is represented by the so-called passive dynamic walkers, pioneered by McGeer who introduced the concept of natural cyclic behavior.
Recently, this principle has been used to develop powered bipedal walkers that walk high efficiency in a more human-like way than the predecessors by exploiting natural dynamics. On this way, there were also several studies about the so-called "*spring-mass walking*", which demonstrated walking dynamics with a considerable similarity to that of human. The spring-mass model was originally proposed for characterizing running behavior of animals. The model consists of a body represented as a point mass and a leg approximated by a linear spring. This model was also extended for walking behavior, which explained a few aspects of human bipedal locomotion and then was elaborated for robotic implementation. An exploration of mechanically realistic model considering the theroretical spring-mass walking model and the anatomical structure of biological systems can be found at [27] in which there is also a simulation that involves the biped robot Jena Walker I. Nowadays, the state-of-the-art is represented by the equation below:

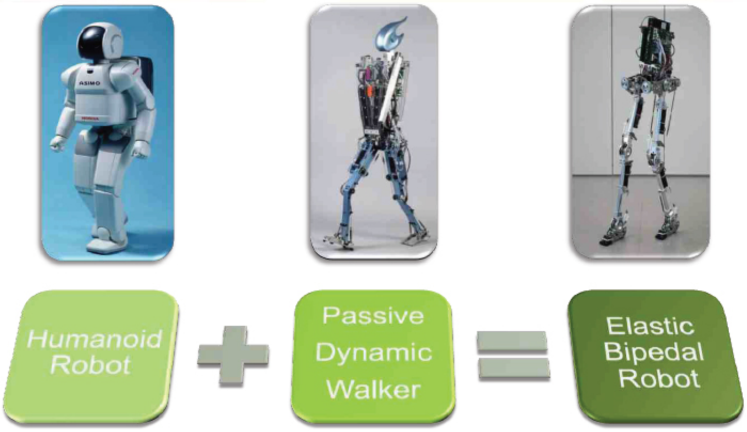


Figure 2.15: The state-of-the-art about Elastic Bipedal Robot Design.

There are three main challenges in developing biologically inspired systems with partially passive dynamics:

1. Systematical exploration of the basic mechanisms of self-stabilization including additional functional elements such as adjustable spring-damper regulators and basic feedback loops. After disturbances of the periodic locomotion pattern, if the CoM returns to the limit-cycle trajectory without any, or only a minimal amout of feedback control, processing sensory information on the actual disturbance, then this is referred with the term "self-stabilization". The use of electrical motors coupled to

spring-damper systems can reduce the slack in the joint, while a rapid adaptation to small unpredictable bumps in the ground can be taken over by passive compliance of the muscle-tendon system.

2. Roles of morphology in under actuated systems with respect to behavioral diversity. Which can be significantly influenced by the dynamics induced by the interactions with simple motor action and the ground reaction force. Phase-dependent activation of elasticities represents one out of many techniques to achieve behavioral diversity.

3. Using computational optimization tools to tackle with the problem of controlling nonlinear dynamics. Analysis of the main factors affecting the chosen objective function and a simple controller with preferably small parameter space are key factors that help reducing the number of iterations during the optimization process.

Previous and existing projects have shown that more use of springs in legged robots, particularly robots that are designed to run fast. Elasticity of legs, partially storing and releasing energy during contact with the ground, allow to achieve not only stable, but also rapid and energy-efficient locomotion. Consequently, the use of spring and more generally of elastic actuators, in legged locomotion, is nowadays generally accepted as important. However, also if mechanical elasticity is a prerequisite for ballistic human-like movements, it does not necessarily result in low energy requirements. The built-in elasticities strongly interact with the actuator modules causing a higher payload. Therefore, it is important to study well before the development of a human-like robot where to integrate elasticities. Neglecting this issue during the design process may lead to an even higher energy consumption than detected in conventional robots.
Particularly, three main uses of springs are suggested:

1. pogo stick principle in order to bounce along on springs. In this way energy is saved and unwanted heat production is reduced;

2. return springs to halt the legs at the end of each forward or backward swing and start them swinging the other way. So, further energy is saved;

3. compliant foot pads to moderate forces at impact of feet with the ground. This improves road holding by preventing vibrations.

These uses can be observed in animals and have also frequently appeared in existing robots. Bridging the gap between artificial and natural systems requires still addressing many conceptual and technological challenges and involves interdisciplinary knowledge. BioRob arm for that regards robot manipulators, and Jena Walker II, BioBiped R1 for that regards legged locomotion represent withoud doubt the state-of-the-art in these research areas. We discuss in more detail these robots in chapter 4 when we provide the design of our elastic bipedal robot. Moreover, in chapter 5 we also present a brief overview about some control approach and series elastic actuators.

Chapter 3

Simulation Environment: OpenSim

Computer simulations provide a framework for exploring the biomechanics of movement. Dynamic simulations of movement allow one to study neuromuscular coordination, analyze athletic performance, and estimate internal loading of the musculoskeletal system. Simulations can also be used to identify the sources of pathological movement and establish a scientific basis for treatment planning.

The idea behind the basic steps involved in the production of voluntary movement is quite simple. Commands initiated in the brain are transmitted along nerves to muscles. When activated by nerves, muscles generate forces. Muscle forces are transferred to bones and produce angular motions of the joints. When the nervous system properly coordinates the activation of many muscles, the results is smooth, purposeful movement. Scientists fascinated by human and animal movement have examined each of these steps and performed an extensive range of experiments to record neuromuscular excitation patterns, characterize muscle-contraction mechanics, describe musculoskeletal geometry, and quantify movement dynamics. However, linking detailed knowledge of neuro-musculoskeletal elements to create an integrated understanding of movement remains a challenge. Researchers need simulations to complement experimental studies because important elements of movement, including neural signals and muscle forces, are extremely difficult to measure experimentally. Using experiments alone to understand movement dynamics has two fundamental limitations: first, important variables, including the forces generated by muscles, are not generally measurable in experiments; second, it is difficult to establish cause-effect relationships in complex dynamic systems from experimental data alone. A theoretical framework is needed, and a dynamic simulation of movement that integrates models describing the anatomy and physiology of the elements of the neuro-musculoskeletal system and the mechanics of multi-joint movement provides such a framework.

On the other hand, developing accurate simulations of human and animal movements is challenging because of the intrinsic complexity of biologic systems. Understanding how the nervous system coordinates movement is especially difficult because many muscles work together to produce movement, and any individual muscle can accelerate all of the joints of the body. Therefore, developing and testing biologically realistic models requires collaboration between biologists and engineers. In the last ten years, the efforts of many researchers have been addressed to realize open and close software systems that let users develop models of musculoskeletal structures and create dynamic simulations of a wide variety of movements. Scott L. Delp, Frank C. Anderson, Allison S. Arnold, Peter Loan, Ayman Habib, Chand T. John, Eran Guendelman, Darryl G. Thelen and many others have developed a freely available, open-source software system that allows all that: OpenSim.

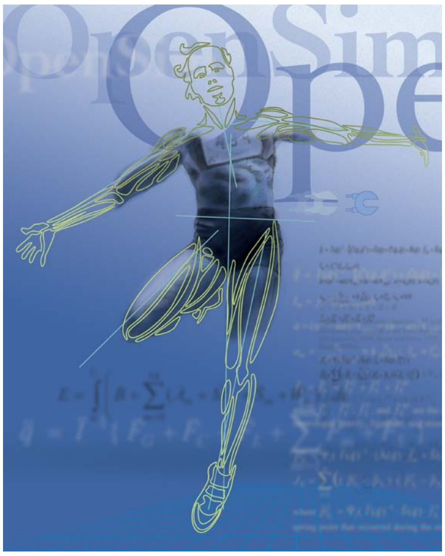


Figure 3.1: Website: https://simtk.org/home/opensim

This section deals the most important aspects of the OpenSim platform: from the concepts of capabilities and compatibility to the core components as Scaling, Inverse Kinematics, Inverse Dynamics, Static Optimization, Computed Muscle Control (CMC) and many others. And at the end we will see what is an OpenSim models. We use this platform in our research from a robotic point of view, also if it thought for study the human movement. In fact, its capabilities give us a tool for study a multibody system like our new humanoid robot in which we have elastic components very similar to muscles from a function point of view.

3.0.4 A computational framework

OpenSim is an open-source platform for modeling, simulating, and analyzing the neuromusculoskeletal system. With it the biomechanics community can build a library of simulations that can be exchanged, tested, analyzed, and improved through a multi-institutional collaboration. OpenSim version 1.0 was introduced at the American Society of Biomechanics Conference in 2007, and with version 2.0, an application programming interface (API) has been added, allowing researchers to access and customize OpenSim core functionality. Since the initial release, the software has been used in a wide variety of applications, including biomechanics research, medical device design, orthopedics and rehabilitation science, neuroscience research, ergonomic analysis and design, sports science, computer animation, and robotic research.
The core software is written in ANSI C++, and the graphical user interface (GUI) is written in Java and is built on the Netbeans Platform. OpenSim plug-in technology makes it possible to develop customized controllers, analyses, contact models, and muscle models

among other things. These plugins can be shared without the need to alter or compile source code. Use of plug-in technology allows computational components such as integrators and optimizers to be updated as appropriate without extensive restructuring. You can analyze existing models and simulations and develop new models and simulations from within the GUI. Open-source, third-part tools are used for some basic functionality, including the Xerces Parser from the Apache Foundation for reading and writing XML files and the Visualization Toolkit from Kitware for visualization.
OpenSim includes a wide variety of features. Some of the most useful features include:

- Taking pictures of musculoskeletal models and making animated movies;
- Plotting results of your analysis;
- Scaling the size of a musculoskeletal model;
- Performing inverse kinematics analyses to calculate joint angles from marker positions;
- Performing inverse dynamics analyses to calculate joint moments from joint angles and external forces;
- Generating forward dynamics simulations of movement;
- Analyzing dynamic simulations;
- Inspect Constraints in the model;
- Enable and disable model forces and constraints;
- Allow multiple geometries to be associated with one body and be able to assign the color, texture, visibility or transform of each piece of geometry independently.

As we said previously, you can create your own models of musculoskeletal structures and dynamic simulations of movements, as well as take advantage of computer models and dynamic simulations that other users have developed, shared and posted at Simtk.org where also OpenSim is maintained by a growing group of participants. Simtk.org serves as a public repository.
OpenSim is built using SimTK, an open-source simulation toolkit developed to create mathematical models of biological dynamics. SimTK is being developed by Simbios, an NIH National Center for Biomedical Computation based at Stanford University. The purpose of SimTK is to enable groundbreaking biomedical research by providing open access to high-quality tools for modeling and simulating biological structures. It is the low-level, domain independent computational and we will briefly cite the base functionality needed for the OpenSim API: numerical objects (numbers, constants, vectors, matrices), numerical methods (linear algebra, optimization, integration), multibody dynamics (Simbody), simulation (system and state). A noteworthy aspect is the Simbody package, that is a full-featured, high-performance multibody dynamics toolset using internal coordinates and capable of modeling open- and closed-topology systems. Computation is performed using a recursive O(n) method so that performance scales linearly with problem size. OpenSim's dynamics capability is based on the open-source Simbody package and on

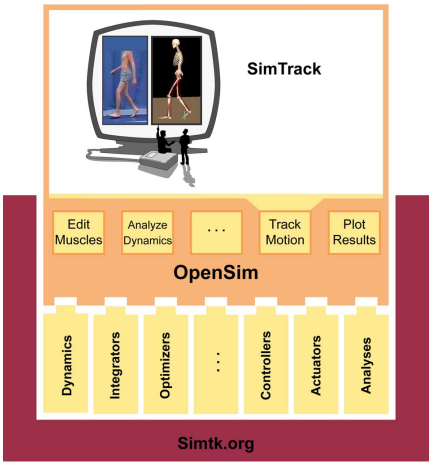


Figure 3.2: Schematic of OpenSim.

its basic concepts: Body (mass properties and geometry), Mobilizer (internal coordinate joint), Constraint, and Force.

The primary objects involved in computational simulation of a physical system in SimTK are: System, State and Study. The OpenSim Model class implements a SimTK System, and the OpenSim Manager represents a Study. OpenSim uses a SimTK::State object directly to represent the state of an OpenSim Model. A System is the computational embodiment of a mathematical model of the physical world. The System itself is an unchangeable, state-free ("const") object. Instead, the values of its variables are stored in a separate object, called a State. Finally, a Study couples a System and one or more States, and represents a computational experiment intended to reveal something about the System. It is important to note that from this point of view, a State, means everything variable about a System. That includes not only the traditional continuous time, position and velocity variables, but also discrete variables, memory of past events, modeling choices, and a wide variety of parameters that are called instance variables. A System is composed of a set of interlocking pieces, called subsystems, each of which may have its own state variables. By design this is not a hierarchical structure. It is a flat partitioning of a System into a small number of Subsystems, so the computational resources are flat, and the SimTK System/Subsystem scheme is a computational device, not a modeling system. SIMM (Software for Interactive Musculoskeletal Modeling) from Motion Analysis Corp. is a widely used software application for biomechanical simulation, surgical planning, and ergonomic analysis which was introduced in the early 1990s by Scott L. Delp and Peter Loan. Using SIMM, models of the lower and upper extremities were developed to examine the biomechanical consequences of surgical procedures including tendon surgeries, osteotomies and total joint replacements. A lower-extremity model was used to estimate muscle-tendon lengths, velocities, moment arms, and induced accelerations during normal and pathologic gait. Although SIMM helps users formulate models of the musculoskeletal system and dynamic simulations of movement, it provides no assistance with the computation of muscle excitations that produce coordinated movement and has limited tools for analyzing the results of dynamic simulations. Furthermore, SIMM as well as other commercial packages do not provide full access to source code, which makes it difficult for biomechanics researchers to extend their capabilities. OpenSim complements

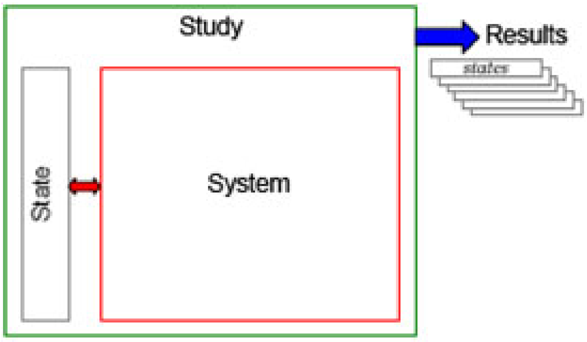


Figure 3.3: SimTK Simulation Concepts.

and augments the functionality of SIMM and the SIMM Dynamics Pipeline by providing advanced simulation and control capabilities. The joint (*.jnt) and muscles (*.msl) files used by SIMM to describe models of the musculoskeletal system can be converted into OpenSim models (*.osim) and brought into the OpenSim framework. In addition, the object-oriented, modular design of OpenSim allows users to extend its functionality and share functionality with other OpenSim users. OpenSim is a self-contained modeling and simulation environment that does not require additional software components or licenses to generate dynamic simulations.

To create a muscle-driven simulation of a movement, one must first formulate a dynamic model of the musculoskeletal system and its interactions with the environment. The elements of the musculoskeletal system are modeled by sets of differential equations that describe muscle contraction dynamics, musculoskeletal geometry, and body segmental dynamics. These equations characterize the time-dependent behavior of the musculoskeletal system in response to neuro-muscular excitation. Once a dynamic model of the musculoskeletal system is formulated, the next step is to find a pattern of muscle excitations that produce a coordinated movement. Excitations may be found by solving an optimization problem in which the objective of a motor task is defined or in which the objective is to drive a dynamic model to "track" experimental motion data.

Determining a set of muscle excitations that produce a coordinated movement is one of the major challenges in creating a dynamic simulation. Historically, the computational cost of generating coordinated muscle-actuated simulations of movement has been high, requiring days, weeks, or months of computer time. Recent breakthroughs in the application of robotic control techniques to biomechanical simulation have dramatically reduced the time needed to generate such simulations. In this scenario OpenSim offers a tool called SimTrack, which guides users through four steps to create a dynamic simulation.

As input, SimTrack takes a dynamic model of the musculoskeletal system, experimentally-measured kinematics, reaction forces and moments. As we can see in the figure below, in Step 1, a dynamic musculoskeletal model is scaled to match the anthropometry of an individual subject. In Step 2, an Inverse Kinematics (IK) problem is solved to determine the model generalized coordinate values (joint angles and translations) that best reproduce the raw marker data obtained from motion capture. In Step 3, a Residual Reduction Algorithm (RRA) is applied to make the model generalized coordinates computed in Step 2 more dynamically consistent with the measured ground reaction forces and moments. In Step 4, Computed Muscle Control (CMC) is used to generate a set of muscle excitations that produce a coordinated muscle-driven simulation of the subject's movement.

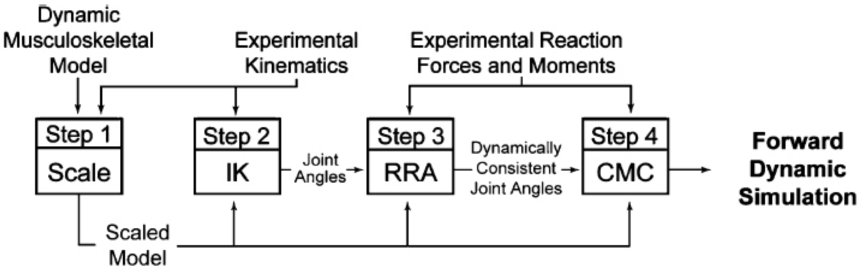

Dynamic Musculoskeletal Model
Experimental Kinematics
Experimental Reaction Forces and Moments
Step 1
Scale
Step 2
IK
Joint Angles
Step 3
RRA
Dynamically Consistent Joint Angles
Step 4
CMC
Forward Dynamic Simulation
Scaled Model

Figure 3.4: Steps for generating a muscle-driven simulation of a subject's motion with SimTrack.

3.0.5 Scaling

The Scale Tool alters the anthropometry of a model so that it matches a particular subject as closely as possible. Scaling is typically performed based on a comparison of experimental marker data with virtual markers placed on a model. In addition to scaling a model, the scale tool can be used to adjust the locations of virtual markers so that they better match the experimental data.

Scaling is performed based on a combination of measured distances between x-y-z marker locations and manually-specified scale factors. The marker locations are usually obtained using motion capture equipment. The unscaled model has a set of virtual markers placed in the same anatomical locations as the experimental markers. The dimensions of each segment in the model are scaled so that the distances between the virtual markers match the distances between the experimental markers. Tool can be used to move some or all of the virtual markers on the model so that they coincide with the experimental marker locations. The scaling step scales both the mass properties (mass and inertia tensor), as

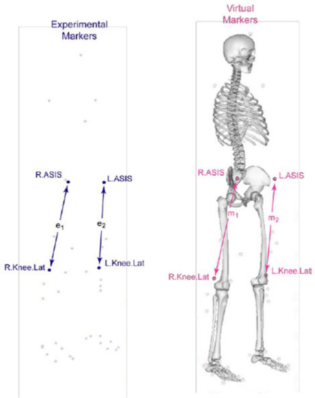


Figure 3.5: Experimental and Virtual Markers.

well as the dimensions of the body segments. This is accomplished using a combination of measurement-based and manual scaling. In measurement-based scaling, scale factors are determined by comparing distances between markers on the model and experimental marker positions provided in a *.trc file. A single scale factor is computed using one or more marker pairs: the experimental distance between a marker pair is computed by looking at each frame of experimental marker data in the given *.trc file, computing the distance between the pair for that frame, and taking the average across all frames in a user-specified time range. The overall scale factor is then the average of the scale factors computed due to all of the pairs. As an alternative to computing scale factors using measured marker positions, it is possible to specify the x-y-z scale factors for a segment manually. This is useful if the actual scale factors for segments are known, or were computed using some alternative algorithm. In addition, the masses of the segments are adjusted so that the total mass of the body equals the specified subject mass. There are two different ways: one approach is to preserve the mass distribution, which ensures that the masses of the subject-specific model segments are in the same proportion as they were in the generic model; the alternative approach incorporates the size scale factors, still ensuring the total mass equals the subject mass, but having the mass of the scaled segments reflect their scale in size. In the first case, the masses are scaled using a constant factor independent of the scale factors that were used to scale the individual segment sizes. In any case, the inertia tensor of each segment is updated to reflect its new size and mass.

After scaling the model, the next step is to move the model's markers to match experimental marker locations in a static pose. The static pose is computed by trying to match some combination of experimental marker positions and generalized coordinate values. The marker locations corresponding to the static pose are computed by averaging the marker positions in a given *.trc file across a user-specified time range. Once a static pose is computed using the IK-based algorithm, all model markers, that are not designated as fixed, are moved to the averaged "static pose" positions of the experimental markers.

3.0.6 Inverse Kinematics

The purpose of the IK step, as we said previously, is to find the set of generalized coordinates (joint angles and positions) for the model that best match the experimental kinematics recorded for a particular subject. This step is formulated as a least-squares problem that minimizes the differences between the measured marker locations and the model's virtual marker locations, subject to joint constraints. The inverse kinematics problem is to minimize the weighted squared error:

$$SquarredError = \sum_{i=1}^{markers} w_i(\vec{x}_i^{subject} - \vec{x}_i^{model})^2 + \sum_{j=1}^{joint\ \ angles} w_j(\theta_j^{subject} - \theta_j^{model})^2 \quad (3.1)$$

where $\vec{x}_i^{subject}$ and $\vec{x}_i^{model}$ are the three-dimensional positions of the ith marker or joint center for the subject and model, $\theta_j^{subject}$ and θ_j^{model} are the values of the jth joint angle for the subject and model, and w_i and w_j are factors that allow markers and joint angles to be weighted differently.

A marker error is the distance between an experimental marker and the corresponding

marker on the model when it is positioned using the generalized coordinates computed by the IK solver. On the other hand, a coordinate error is the difference between and experimental coordinate value and the coordinate value computed by IK. Where the experimental coordinate values can be joint angles obtained from a motion capture system, or may be computed from experimental data by various specialized algorithms or by other measurement techniques that involve other measurement devices. A distinction should be made between prescribed and unprescribed coordinates. The first one is a generalized coordinate whose trajectory is known and which will not be computed using IK, while the second one is a coordinate which is not prescribed, and whose value is computed using IK. From this point of view, only unprescribed coordinates can vary and so only they appear in the least squares equation solved by IK.
The least squares solution is affected by the choice of length and angle units. The units used by IK are the model's units, which are meters for length and radians for angles.

3.0.7 Inverse Dynamics

The inverse dynamics tool determines the generalized forces at each joint responsible for a given movement. Given the kinematics describing the movement of a model and perhaps a portion of the kinetics applied to the model, the tool uses these data to perform an inverse dynamics analysis. Classical mechanics mathematically expresses the mass-dependent relationship between force and acceleration, $F = ma$, with equations of motion. The inverse dynamics tool solves these equations, in the inverse dynamics sense, to yield the net forces and torques at each joint which produce the movement.
The classical equations of motion may be written in the following form

$$A(q)\ddot{q} + b(q, \dot{q}) + g(q) = \Gamma \tag{3.2}$$

where q is the $n \times 1$ vector of generalized coordinates, $A(q)$ is the $n \times n$ mass matrix, $b(q, \dot{q})$ is the $n \times 1$ vector of centrifugal and Coriolis terms, $g(q)$ is the $n \times 1$ vector of gravity terms, and Γ is the $n \times 1$ vector of generalized control forces (torques) with n the number of degrees of freedom.
The motion of the model is completely defined by the generalized positions, velocities, and accelerations. So, all of the terms on the left-hand side of the equations of motion are known. The remaining term on the right-hand side of the equations of motion is unknown. The inverse dynamics tool uses the known motion of the model to solve the equations of motion for the unknown generalized forces.

3.0.8 Static Optimization

Static optimization is an extension to inverse dynamics that further resolves the net joint moments into individual muscle forces at each instant in time according to a model with muscles. The Static Optimization tool uses the known motion of the model to solve the equations of motion for the unknown generalized forces subject to one of the following muscle activation-to-force conditions, ideal force generators and constrained by force-length-velocity properties respectively:

$$\sum_{m=1}^{nm} (a_m F_m^0) r_{m,j} = \tau_j \tag{3.3}$$

or

$$\sum_{m=1}^{nm} \left[a_m f(F_m^0, l_m, v_m)\right] r_{m,j} = \tau_j \tag{3.4}$$

while minimizing the objective function:

$$J = \sum_{m=1}^{nm} (a_m)^P \tag{3.5}$$

where nm is the number of muscles in the model; a_m is the activation level of muscle m at a discrete time step; F_m^0 is its maximum isometric force; l_m is its length; v_m is its shortening velocity; $f(F_m^0, l_m, v_m)$ is its force-length-velocity surface; $r_{m,j}$ is its moment arm about the jth joint axis; τ_j is the generalized force acting about the jth joint axis; and p is a user defined constant.

3.0.9 Residual Reduction

The purpose of Residual Reduction is to minimize the effects of modeling and marker data processing errors that aggregate and lead to large nonphysical compensatory forces called residuals. Residual reduction is a form of forward dynamics simulation that utilizes a tracking controller to follow model kinematics determined from the inverse kinematics. It is primarily intended for gait, i.e. movements like walking and running where the model is displaced relative to the ground while subject to ground reaction forces and torques. From Newton's second law, the following equation relates the measured ground reaction force and gravitational acceleration to the accelerations of the body segments:

$$\vec{F}_{external} = \sum_{i=1}^{segments} m_i \vec{a}_i - \vec{F}_{residual} \tag{3.6}$$

where $\vec{F}_{external}$ is the measured ground reaction force minus the body weight vector, $\vec{a}_i$ is the translational acceleration of the center of mass of the ith body segment, m_i is the mass of the ith body segment, and $\vec{F}_{residual}$ is the residual force. An analogous equation relates the ground reaction moment to the model kinematics and the residual moment. To reduce the residual forces and moments, the residuals are computed and averaged over the duration of the movement. Based on these averages, the algorithm recommends changes in the model mass parameters, such as the location of the center of mass of the trunk, that reduce the average values of the residuals over the duration of the movement.

Residual Reduction Algorithm (RRA)

Tracking Simulation. In the first step, the algorithm places the model in the starting configuration, by setting the value of the model's generalized coordinates to the values computed by the IK tool for the user-specified initial time. Repeatedly, RRA takes small steps forward in time until the user-specified final time is reached, computing the force values for all model's actuators. At each step, the actuator forces are computed by choosing force and torque values that minimize an objective function. There are two possible objective functions to use during static optimization in RRA. The slow target consists of an objective function (J) that is a weighted (w) sum of squared actuator controls, x, plus the sum of desired acceleration ($\ddot{q}_j$) errors:

$$J = \sum_{i=1}^{nx} x_i^2 + \sum_{j=1}^{nq} w_j(\ddot{q}_j^* - \ddot{q}_j) \tag{3.7}$$

The fast target is the sum of squared controls augmented by a set of equality constraints (c) that requires the desired accelerations to be achieved within the tolerance set for the optimizer:

$$J = \sum_{i=1}^{nx} x_i^2; \quad C_j = \ddot{q}_j^* - \ddot{q}_j, forallj \tag{3.8}$$

The fast target is both faster and must produce perfect tracking. However, if the constraints cannot be met, the fast target will fail, resulting in RRA exiting with an error message.

Mass Center Adjustment. At the end of the simulation, the average value for each residual actuator is computed. The average values are used to adjust the torso mass center to correct excessive "learning" of the model due to inaccurancies in the mass distribution and geometry of the torso in the model. A new model file is created.

Mass Adjustment Recommendation. This step is not applied to the model automatically, but it is recommended to the user, who can change the OpenSim model file by hand. Specifically, the desired mass change is: F_y/g, where F_y is the force applied along the Y (vertical) axis, and $g = -9.80665m/s^2$. This mass change is then divided up proportionally among the body segments.

Adjusted Kinematics. The same tracking simulation process is then repeated with three important differences:

1. The model with the adjusted torso mass center is used;

2. The residuals are weighted more heavily to make the optimizer choose smaller values for the residuals when minimizing the objective function;

3. Minimum and maximum limits are placed on the residual values;

These restrictions aim to reduce the need for residuals to the absolute minimum that is necessary to closely follow the desired kinematics so that the motion is generated purely by internal joint moments. Therefore, if the minimum and/or maximum allowed residual values aare too restrictive, the motion will be altered so dramatically that the results of RRA cannot be used to generate a realistic simulation with Computed Muscle Control (CMC), the next stage of OpenSim. But, if the residual minimum and/or maximum values are too lenient, then the residuals will still be large enough to exert forces that might normally be exerted by muscles, and thus the results would lead to unrealistic muscle function from CMC.

3.0.10 Computed Muscle Control

The purpose of Computed Muscle Control (CMC) is to compute a set of muscle excitations (or more generally actuator controls) that will drive a dynamic musculoskeletal model to

track a set of desired kinematics in the presence of applied external forces. It attempts to bridge the gap between forward and inverse methods by combining: PD feedback control to track experimental kinematic, static optimization to estimate the feed forward controls (muscle excitations) in order to generate desired accelerations at a small time (T) in the future, and then forward integration to generate new states and step forward in time.
With musculoskeletal models, we are also typically interested in estimating muscle forces and controls. Traditional approaches to solve for muscle controls computed, like static optimization, often fail to reproduce the observed motion (the inputs to inverse dynamics and static optimization) when applied in a forward dynamics simulation. There are three principle causes for this discrepancy:

1. forward and inverse musculoskeletal models do not share identical dynamics;
2. experimental noise and sampling results in dynamically inconsistent kinematics;
3. musculoskeletal models are nonlinear dynamical systems and inherently chaotic.

Cause 3) is often overlooked but it is important to realize that even if identical models where used in an inverse and then forward analysis with noiseless and error-free kinematics (i.e. synthetic data) a forward simulation will fail to reproduce the initial performance if the initial states of the simulation are not identical, since even the smallest of differences (to machine precision) can lead to diverging solutions. Cause 2) stems from the reality that data acquired (from a subject) does not match what could be generated by the model (satisfying modeled dynamics) and the estimates of joint kinematics (from IK) does not take into the continuity of system dynamics from one instant to the next given discrete samples of position data. The largest source of discrepancies is the fact that different models are used to perform inverse dynamics and static optimization versus that of a forward simulation. Even when static optimization includes force-length and force-velocity relationships, the estimate of muscle length and velocity are determined by the length of the whole muscle-tendon unit (inelastic tendon) and activations do not satisfy excitation-to-activation dynamics present in forward.
Before starting the CMC algorithm, initial states for the model are computed. The states comprise the generalized coordinates (joint angles), generalized speeds (joint angular velocities), plus any muscle states (for example, muscle activation levels and fiber lengths). While the initial values of the generalized coordinates and speeds can be taken from the desired kinematics that you specify, the initial values of the muscle states are generally unknown. To compute viable starting muscle states, CMC is applied to the first 0.030 seconds of the desired movement. Because the muscle states are generally out of equilibrium and muscle forces can change dramatically during this initial time interval, the simulation results during this interval are generally not valid.
First step is to compute a set of desired accelerations, using the following PD control law:

$$\ddot{\vec{q}}^*(t+T) = \ddot{\vec{q}}_{exp}(t+T) + \vec{k}_v \cdot \left[\dot{\vec{q}}_{exp}(t) - \dot{\vec{q}}(t)\right] + \vec{k}_p \cdot \left[\vec{q}_{exp}(t) - \vec{q}(t)\right] v \tag{3.9}$$

where $\vec{k}_v$ and $\vec{k}_p$ are the feedback gains on the velocity and position errors, respectively. Because the forces that muscles apply to the body cannot change instantaneously, the desired accelerations are computed for some small time in the future. For musculoskeletal models, is typically chosen to be about 0.010 seconds. This time interval is short enough to allow adequate control, but long enough to allow muscle forces to change. In fact, this value corresponds to the amount of timer required before an actuator can generate the required forces to match the desired accelerations. Differently from RRA, where all actuators are ideal, so this look-ahead window can be arbitrarily small. The error between the model coordinates and experimentally-derived coordinates will be driven to zero choseing the velocity gains with the following relation: $\vec{k}_v = 2 \cdot \sqrt{\vec{k}_p}$. For musculoskeletal models, it works well if the error gains are chosen to drive any errors to zero slowly.

The next step in CMC is to compute the actuator controls, $\vec{x}$, that will achieve the desired accelerations $\ddot{\vec{q}}^*(t + T)$. Any kind of actuator can be used with CMC and static optimization is used to distribute the load across synergistic actuators. Two formulations of the static optimization problem are currently available in CMC. The first formulation is called the slow target Eq.(3.7) and the second formulation is called the fast target Eq.(3.8).

The final step in the CMC algorithm is to use the computed controls to conduct a standard forward dynamic simulation, advancing forward in time by T.

These steps, are repeated until time is advanced to the end of the desired movement interval.

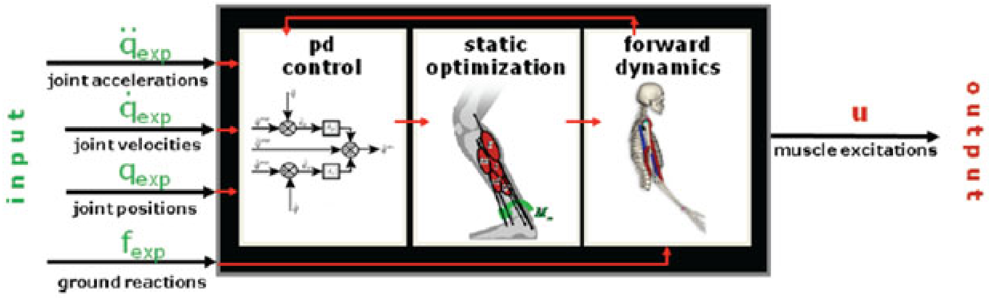


Figure 3.6: Overview of Computed Muscle Control.

3.0.11 Forward Dynamics

Given the controls computed by the Computed Muscle Control (CMC), the Forward Dynamics Tool can drive a forward dynamic simulation. It uses the same model and actuator set used in CMC, together with the initial states and controls computed during the CMC step, to run a muscle-driven forward dynamic simulation that aims to reproduce the same motion tracked by CMC.

As in CMC and RRA, a 5° order Runge-Kutta-Feldberg integrator is used. In contrast to CMC, which used PD controllers in a closed-loop system to ensure tracking of the desired trajectories, the Forward Dynamics Tool is an open-loop system.

The main purpose for the forward dynamics step, besides validating the CMC results, is to record additional simulation data. And this is done through the use of analyses that can be added also to RRA and CMC as well. The basic analyses of interest are: Kinematics, BodyKinematics and Actuation; which will be describe in the next section in more detail.

3.0.12 Analyzing Simulations

The Analyze Tool enables you to analyze a model or simulation based on a number of inputs that can include time histories of model states, controls, and external loads applied to the model. A typical use case is to analyze an existing simulation, which may have been computed using computed muscle control, without having to rerun the simulation. This not only saves compute time, but more importantly allows a simulation to be analyzed exactly as it occurred, avoiding the numerical drift that frequently occurs when rerunning a forward integration, particularly if the duration of the simulation is long. The Analyze Tool steps in time through a set of input data specifying the state of a model; at each time step, the tool runs a set of analyses on the model. As we said previously, available analyses include:

- **Kinematics**: Records the generalized coordinates (q's), generalized speeds (u's), and the accelerations (i.e., derivatives of the generalized speeds: du/dt)

- **BodyKinematics**: Records the configuration (center of mass position and orientation) of each body, as well as their velocities (linear and angular) and accelerations (linear and angular). Additionally, it records the overall center of mass of the model, as well as the velocity and acceleration of this center of mass.

- **Actuation**: Records the generalized force, speed, and power developed by each actuator of the model. The generalized force can either be a force (with units N) or a torque (with units Nm). The actuator speed is the rate at which the actuator shortens. Depending on the actuator, a speed can be either a translational speed (m/s) or an angular speed (deg/s). An actuator power (Watts) is the rate at which an actuator does work. Positive work means that the actuator is delivering energy to the model; negative power means that the actuator is absorbing energy from the model.

- **JointReaction**: Reports the joint reaction loads from a model. For a given joint, the reaction load is calculated as the forces and moments required to constrain the body motions to satisfy the joint as if the joint did not exist. The reaction load acts at the joint center (mobilizer frame) of both the parent and child bodies and either force can be reported and expressed in either the child, parent or ground frames. The default behavior is to express the force on the child in the ground frame.

The input data typically are loaded from files, and may come from experiments or simulations. The results are collected and written to file, usually storage (.sto) files, which you can open with other programs, such as Matlab or Microsoft Excel, for further analysis or plotting.

3.0.13 Perturbation

The Perturbation Tool can be used to compute accelerations induced by the actuators of a model, i.e., the contribution of individual actuators to the measured acceleration. Typically, induced accelerations of generalized coordinates (e.g., knee angle) or body positions (e.g., hip center) may be desired, and the actuators consist of the muscles and

any additional actuators (e.g., residual actuators, reserve actuators, etc.).
The formula that it uses to compute induced accelerations, that depends linearly on force ($\frac{\partial \ddot{x}}{\partial F_m}$), is:

$$\ddot{x}_m(t_i) \approx 2 \cdot \frac{x(F_m + \Delta F_m, t_i + \Delta t) - x(F_m, t_i + \Delta t)}{\Delta t^2 \Delta F_m} F_m \tag{3.10}$$

where $x(F_m, t_i + \Delta t)$ is the observed unperturbed positions, $x(F_m + \Delta F_m, t_i + \Delta t)$ is the observed perturbed positions.
Each foot has a linear and torsional spring added to it which attempts to pull it towards its location and orientation in the unperturbed simulation. The linear (translational) spring connects a point in the foot's frame to a point in world space. This point is the center of pressure. Its foot frame location is computed using the unperturbed trajectory. As the actuator forces are perturbed, the foot's trajectory will be perturbed, and this spring will try to pull the foot back to its unperturbed position.
The torsional spring tries to restore a foot's orientation to its orientation in the unperturbed simulation.
Since these springs are trying to model a rudimentary foot-contact model, they are not on at all times. The linear spring is primarily active between heel strike and toe off. The torsional spring is primarily active between foot flat and heel off. In order to make the transition between the springs smooth, a smooth falloff function is used to modulate their effects, rather than discontinuously turning them on/off at the right time. Due to this smoothing, the springs are technically on at all times, but outside of the intervals mentioned above, they are heavily attenuated so that their force is close to negligible. The falloff equation is defined as follows:

$$1.0 - \frac{1.0}{1.0 + \exp(-(t - t_0)/\tau)} \tag{3.11}$$

where t = time, t_0 = location of the midpoint of the step, and τ = rise and fall time constant.

Chapter 4

A Novel Elastic Bipedal Robot Design

This chapter begins the part of the thesis concerning the development of robots designed by Fuben He and then the modeling and simulation of movement realized by the author of this thesis through the OpenSim platform, based on the human musculoskeletal models. After a brief presentation of previous works that inspired this research we discuss in detail the structural components of our two-legged robots, the human musculoskeletal system modeling, and finally the joint modeling of mono- and bi-articular systems adopted in our humanoid robot.

In the last years the studies about elastic bipedal robot design have presented several bipedal locomotion model with compliant legs that utilize monoarticular and biarticular arrangement of tension springs. With experiments in simulation and in the real-world robotic platform, scientiests have shown that their model provides some eminent fueatures that could not be explained by the other simple models such as the ***ballistic walking***, generally known as ***compass gait model*** or ***passive dynamic walking***. With regard to this, in [26] Iida et al. have been shown that the compliant elements in their robot leg structure make the model possible to generate both walking and running gaits, and secondly, owing to the biarticular arrangements of the tension springs, their model is able to achieve more human-like leg movements compared with those of ballistic walking, by showing the potential roles of biarticular muscle arrangements and concluding that biarticular muscles do support energy transfer between the joints for the self-stabilization of walking and running gaits. In [48] a simulation model in the sagittal plane using an extended series actuation principle is presented by Radkhah et al. It extends the well known ***Series Elastic Actuator*** (SEA) principle and has several advantages over the conventional SEA. Also in this case, the effect of monoarticular structures in the robot model was explored, showing that they lead to reductions in the energy requirements.

4.1 Former Robots

In this first section we present the characteristics of projects that represent the state-of-the-art in research on humanoid robotics modeling and elastic actuators. In particular, we will present *BioRob arm*, the evolution of *Jena Walker* in his versions I and II, and finally *BioBiped R1*, the newest member borns thanks to the collaboration of SIM Group TU Darmstadt and the Locomotion Laboratory of University of Jena.

4.1.1 BioRob Arm

The BioRob arm is an equilibrium-controlled stiffness manipulator. The arm consists of a very lightweight structure with rigid links, elastically actuated by DC motors driving the joints by pulleys and cables with built-in mechanical compliances. The specific properties of the BioRob arm concept compared to other series elastic concepts are reduced link mass and inertia (a total mass of 4 Kg), reduced power consumption, and a significantly lower joint stiffness (ranging between 4 and 20 Nm), in total resulting in increased safety for applications with direct human-robot interaction. As a downside, the use of cable and pulley actuation increases friction and the series elasticity with particularly low joint stiffness demands special efforts regarding oscillation damping.

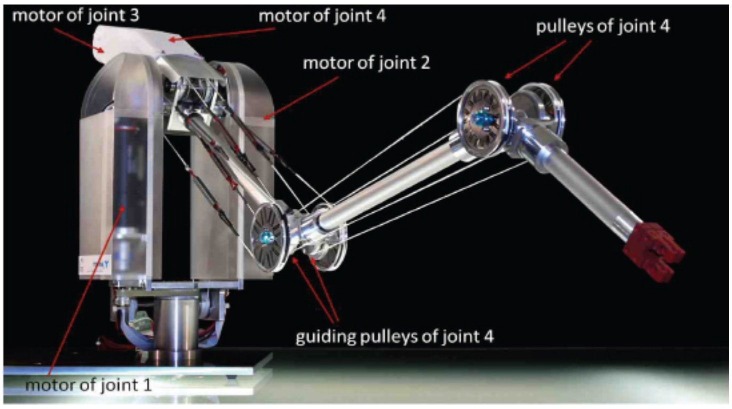


Figure 4.1: Position of motors on the BioRob arm [79]

Alternative actuation concepts such as pneumatic muscles exhibit inherent compliance and omit the need for gearboxes, but are slower, have a restricted range of operation and are suited for mobile applications only to a very limited extent. Electrical motors on the other hand are robust, allow high speeds, exhibit excellent controllability and are well suited for highly mobile applications. A more detailed description of the robot arm dynamics and its control can be found at [79].

4.1.2 Jena Walker I

This bipedal locomotion model and the respective real-world robotic platform consist of seven limb segments (three segments in each leg and one body segment), two motors at the hip joints, four passive knee and ankle joints and eight linear tension springs. Two ground contact points are defined in each foot segment.

The configuration of springs are determined such that they can constrain the passive joints for natural locomotion behavior and support the body weight of the entire system. The springs in each leg correspond to biarticular muscles, *rectus femoris* (RF: hip joint flexor and knee joint extensor), *biceps femoris* (BF: hip joint extensor and knee joint flexor) and *gastrocnemius* (GAS: knee joint flexor and ankle joint extensor), in human legs. Additionally, a monoarticular spring, corresponding to the *tibialis anterior* (TA: ankle joint flexor), is implemented.

More details about simulation of walking and running movement on this robot platform can be found at [26].

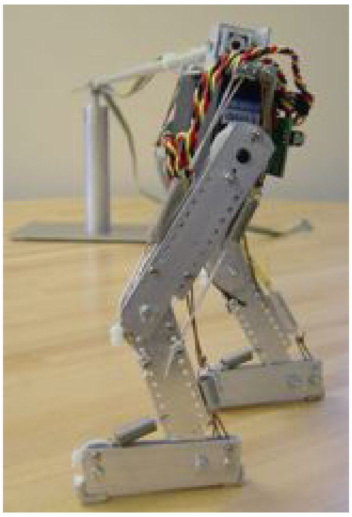


Figure 4.2: A picture of Jena Walker I [26].

4.1.3 Jena Walker II

Jena Walker II represents without doubt the very first robot that is capable of both energy-efficient and human-like walking and jogging. It is a novel elastic and biologically inspired, three-segmented robot that is attached at the trunk to a lateral guide.

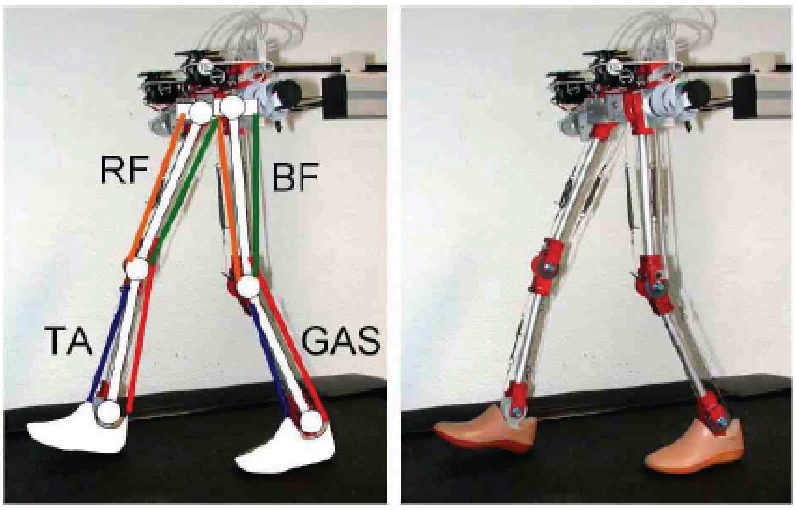


Figure 4.3: Three-segmented elastic legs of Jena Walker II with only one actuated joint in each hip module. On the left-hand side the used passive elastic structures are marked [45].

Its total mass is about 2 Kg and its hip height is 45 cm. The robot systems consists of rigid segments, actuated hip joint modules and adjustable elastic strings spanning including a prosthetic foot (SACH child foot, Otto Bock), shank and thigh. The elastic structures represent the major muscle groups: *tibialis anterior* (TA), *gastrocnemius* (GAS), *rectus femoris* (RF) and *biceps femoris* (BF). Except for the TA, all muscle groups span two joints leading to an inter-joint coupling within the leg. Furthermore, friction in the cables spanning the ankle joint contribute to damping in this joint. This damping is necessary to avoid vibrations of the foot during swing phase.

At the hip, two DC-motors introduce sinusoidal oscillations imitating the altering activity of the hip joint muscles during locomotion, and servo motors are used for tuning the rest length of the springs representing the action of GAS, RF and BF resulting in postural adjustments of knee and ankle joints.

4.1.4 BioBiped R1

This is a project recently launched, in order to further investigate the realization of different gaits without changing the kinematics of the bipedal robot. In this work the SIM Group of TU Darmstadt collaborates with the Locomotion Laboratory of University of Jena with the goal of to build a humanoid robot that is as large as a child and can autonomously change its gaits without any lateral guidance.

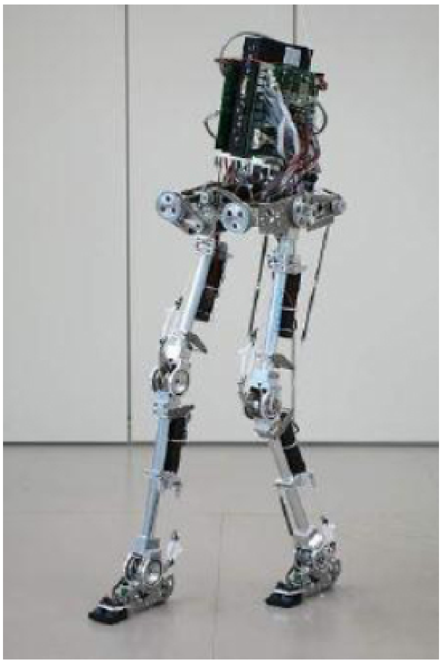


Figure 4.4: BiBiped1. Photo by A. Karguth/TETRA GmbH [48].

The legs consist of three segments, three biarticular structures and five monoarticular ones. The hip motors in the sagittal and lateral plane are actuated by bionic drives such as used in the BioRob manipulator. These drives give the possibility of pretension/preload which reduces/avoids the problem of backlash and play. Moreover, their visco-elastic property introduced between motors and joints provides a better damping reaction in case of collisions or hits.

While the monoarticular structure TA in the shank and the biarticular structures RF, GAS and BF are passive, the elastic structures SOL and VAS are actuated. This, in order to feed the ankle and the knee joint with energy, and so to support them during bending with motor power. A simple trunk that can tilt forwards and backwards has been also considered, in the first prototype.

The sensory data provided by the encoders at the driving ends and in the joints helps comparing different control approaches. Force contact sensors at the feet heel and ball

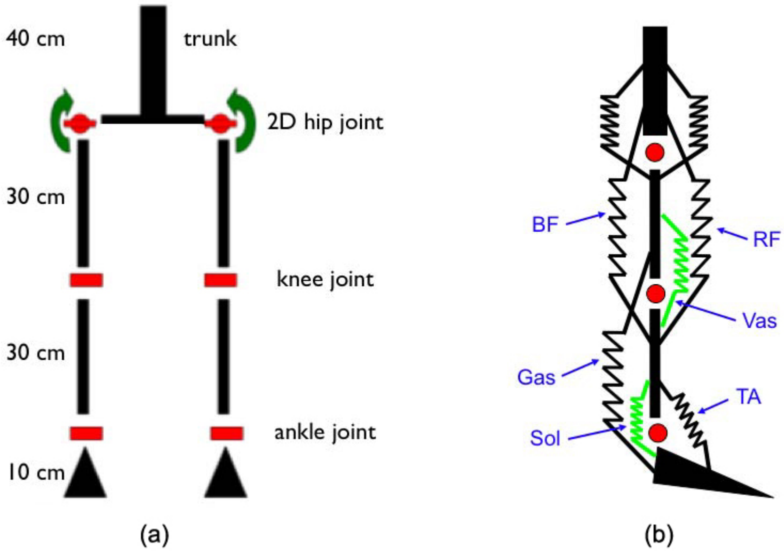


Figure 4.5: (a) Front view of the structure of the bipedal model. (b) Side view of the mechanical design of the robot including the elastic structures spanning the ankle, knee and hip joint. The tensions marked in green color are actuated. All others are integrated as passive structures [45].

give the opportunity to compare with experimentally observed data from human subjects resulting in the possibility to even better understand the principles underlying legged locomotion and tune the system such as the objective criterions can be achieved. In fact, the obtained sensory data are compared with human experimental data from walking and running gaits, in order to assess the performance of the used actuation modules including the underlying actuation principles and the integrated spring properties. Finally, an inertial measurement unit in the trunk keeps track of changes in the linear acceleration and angular velocity of the robot.

4.2 Structure Elements of the Robot

In this section we provide a description of the structure elements of the robot. There are three main part that characterize this flexible robot system: ***springs***, ***servo actuator*** and ***contactless magnetic rotary encoder***. For how regard springs we describe only some basic concept because it is still an open topic into the design of the robot. On the other hand, for how regard servo actuator and encoder we report some characteristics information obtained from the respectively producers. At this stage of development, we cannot be more detailed about the structure elements of the robot because only many future experiments will bring to a complete characterization of the structures during the construction stage. At the time of these thesis, an experimental platform for testing the abilities of elastic joint which simply contains a servo motor, a joint, springs and other corresponding mechanical parts has started at Dalian University under the direction of Fuben He. Encoders and control board are also applied to this platform. This job will provide a series of more detailed data of elasticity before manufacturing the bipedal robot.

4.2.1 Springs

A spring is an elastic object used to store mechanical energy. Most springs obey Hook's law, which states that the force with which the spring pushes back is linearly proportional to the distance from its equilibrium length:

$$F = -kx \tag{4.1}$$

where x is the displacement vector, F is the resulting force vector, and k is the spring constant or force constant.

There are several uses in manipulator: used as the best passive elastic elements. The joint becomes stiffer when accelerating and decelerating by using spring characteristic, having a positive effect on the performance. ***Spring rate*** is used to measure the reaction forces and moments or the position difference, so the motion of the manipulator can be adjusted to limit the possible force-elongation in changing situations which is similar to the variability of the human muscle. In biped robot three main uses are suggested: pogo stick principle in order to bounce along on springs: save energy and reducing unwanted heat production; return springs to halt the legs at the end of each forward or backward swing and start them swinging the other way: save further energy; compliant foot pads to moderate forces at impact of feet with the ground: improve road holding by preventing vibrations.

The decision about what type of spring to put in our robot has not been taken completely, yet. For the moment, into the design phase has been considered the use of ***extension springs*** principaly for energy-efficiency reason. Consequently, in this section we provide a brief overview about the springs' theory based on `http://springipedia.com/` with particularly attention to the extension spring type and its uses. In the future, after more accurate studies about the behavior of different types of springs more specific extension springs will be place in the real robot prototype. Nowadays, several type of springs exist and many studies have been conducted during the years about their mechanics, shape, behavior, materials and possible applications but there are three major typology of springs commonly used, that we discuss here: *Compression Springs*, *Extension Springs* and *Torsion Springs.*

Compression Springs

This is the most common metal spring configuration more than other because in this configuration they are one of the most efficient energy storage devices available. A compression spring is an open-coil helical spring that offers resistance to a compressive force applied axially. This device can be use in many shapes as cylindrical, conical, barrel and hourglass; it is commonly placed over a rod or fitted inside a hole. For these reasons they are found in a wide variety of applications ranging from automotive engines and large stamping presses to major appliances and lawn-mowers to medical devices, cell phones, electronics and sensitive instrumentation devices. When you put a load on a such type of spring, making it shorter, it pushes back against the load and tries to get back to its original length. Total number of coils is counted from tip to tip.

The active coils are what make a spring a spring. In the case of compression spring, the active portion will expand as the spring is compressed. This term can be applied to any portion of a spring that stores and releases energy. So, when designing a spring and

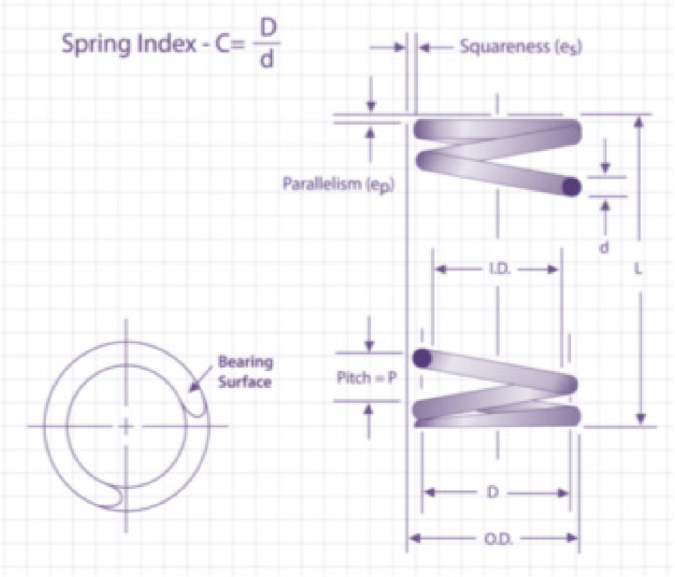


Figure 4.6: A basic characterization.

specifying its dimensions, it is critical that the number of coils is counted correctly, as this can have a huge effect on the strength of the spring.

There are four basic types of compression spring ends: ***open end*** (the coils are consistent with no pitch change through the end of the spring), ***closed ends - not ground*** (the end coils' pitch is reduced so the end coils touch), ***open & ground*** (last coil ground "flat" in appearance and has a less parallel end), ***closed - squared & ground*** (last coil not "flat" in appearance and has a less parallel end). There is also a variety of shapes, however, custom designs may have number of shapes depending on the application. Some common custom shapes include the ***cone*** shape where the spring radius decreases, a common shape is a ***battery spring***; an ***hour glass*** shape tapers tighter towards the center and the outer coils have a larger diameter; the ***barrel shape*** is reduced at the ends and wider in the center; the ***reduced ends*** spring is straight across the center coils and tapers only towards the end coils.

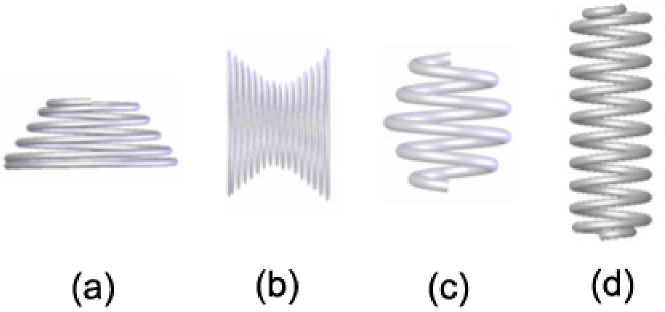


Figure 4.7: A variety of shapes. (a)Conical; (b)Hourglass (Convex); (c)Barrel (concave); (d)Reduced Ends.

When a compression spring is loaded, the coiled wire is stressed in torsion. The stresses in the spring is determined from the dimensions along with the load and deflection requirements. ***Stress*** and ***stress range*** govern the life of the spring: the higher the stress range, the lower the maximum stress must be to attain comparable life. Generally, the stress is greatest at the surface of the wire; as the spring is deflected, the load varies, producing a range of operating stress. On the other hand, in designing compression springs the space alloted governs the dimensional limits of a spring with regard to allowable solid height, and outside and inside diameters. These dimensional limits, together with the load and deflection requirements, determine the stress level.

Extension Springs

Helical extension springs are similar to helical compression springs, but are loaded in tension. Usually, extension springs are attached at both ends to other components. When these components move apart, the spring tries to bring them together again. Extension springs absorb and store energy as well as create a resistance to a pulling force. It is initial tension that determines how tightly together an extension spring is coiled. This initial tension can be manipulated to achieve the load requirements of a particular application. Extension Springs are wound to oppose extension. They are often tightly wound in the no-load position and have hooks, eyes, or other interface geometry at the ends to attach to the components they connect. They are frequently used to provide return force to components that extend in the actuated position.
Applications for extension springs, further than robotics, include automotive interiors and exteriors, garage door assemblies, vise-grip pliers, carburetors, trampolines, washing devices, farm machinery, toys as well as thousands of other uses. Extension springs come in a wide array of sizes, from small medical devices to off-road machinery brake springs.

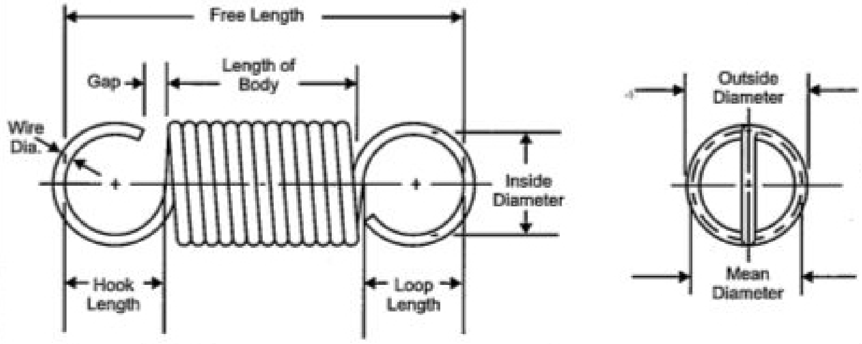


Figure 4.8: Typical Extension Spring dimensions.

With reference to the Figure 1.8, ***Free Length*** is the overall dimension of an extension spring in the unloaded position. The free length is measured from the inside of one end loop (or hook) to the other and can be varied by changing the end configuration without changing the number of coils. ***Body Length*** is the measure of the spring length, excluding the loops. ***Extended Length*** is the length at full rated extension. ***Maximum Load*** is the load at full extension. ***Loop Length*** measures the length of a loop from the end of a body to the inside diameter of the loop. ***Hook Length*** measures the length of a hook from the end of a body to the inside diameter of the hook. ***Gap*** is the dimension that controls the loop or hook opening. ***Initial Tension*** is the force that keeps the coils of an extension spring closed and which must be overcome before coils start to open. ***Active Coils***, all coils in the body are considered active coils, but one coil is typically added to the number of active coils to obtain the body length.

Extension-spring coils are wound tightly together, and the wire is twisted as if is wound, creating a preload in the coils that must be overcome to separate them. With reference to a typical ***load-deflection*** (ld) curve for an helical extension spring, the spring rate is linear except for the initial portion. The preload is measured by extrapolating the linear portion of the curve back to the force axis. The *spring rate* formula for an extension spring is the same as that of a compression spring:

$$R = \frac{Gd^4}{8N_a D_m^3} \tag{4.2}$$

where G is the *gap*, d is the *wire size*, N_a is the *number of active coils*, D_m is the *outside diameter*.

Most extension springs are wound with initial tension. The measure of the initial tension is the loadnecessary to overcome the internal force and just start coil separation. Unlike a compression spring, which has zero load at zero deflection, an extension spring can have a preload at zero deflection. This is graphically illustrated in Figure 4.9.

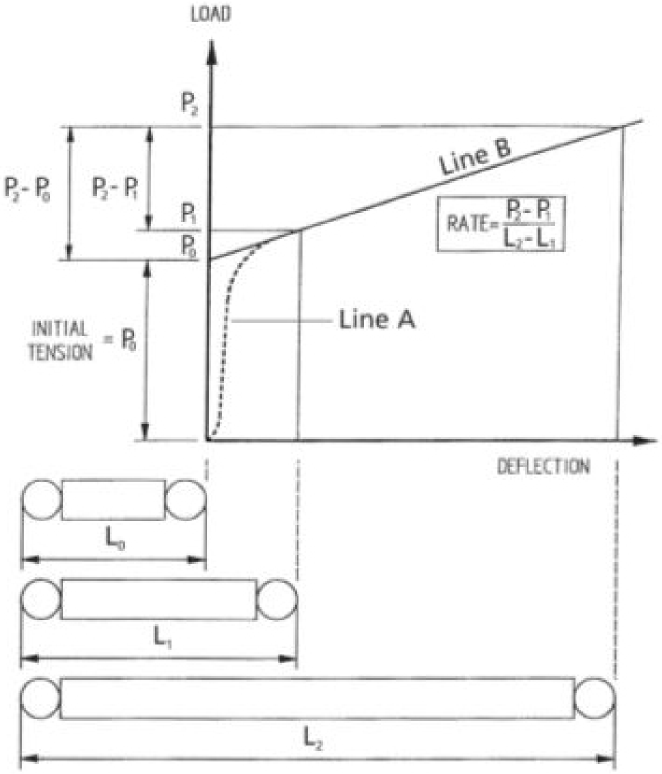


Figure 4.9: An illustration of a preload at zero deflection for an extension spring.

This built-in load, called initial tension, can be varied within limits, decreasing as the spring index increases. If the designer needs an extension spring with no initial tension, the spring should be designed with space between the coils. An accurate method for testing initial tension is as follows:

1. Extend the spring any convenient amount (such as 1 cm) and measure the load;
2. Extend the spring exactly twice that amount (in this case, 2 cm) and again measure the load;
3. Subtract the first load from the second;
4. Subtract this difference from the first load and the remainder obtained is the exact amount of initial tension.

Various types of ends are used to attach the extension spring to the source of the force. They include hooks or eyes at varying positions or distances from the body of the spring, reduced and expanded eyes on the side or in the center of the spring, extended loops, threaded inserts and even rectangular or teardrop-shaped ends. By far the most common, are the machine loop and crossover loop. These ends are made with standard tools in one operation and should be specified when possible to minimize cost. Unlike compression springs, extension springs do not have a solid stop to prevent overloading. Because of this design stress levels are lower for extension springs than for compression. A special type of extension spring called a drawbar spring, has a solid stop and is a type of compression spring with special hooks.

In designing extension springs, it is important to be aware that as the space occupied by the machine loop is shortened, the transition radius is reduced and a substantial stress concentration occurs. This contributes to shortened spring life and premature failure. Most extension spring failures occur in the area of the end. To maximize the life of the spring, the path of the wire should be smooth and gradual as it flows into the end.

Torsion Springs

Helical springs used to apply a torque or store rotational energy are commonly referred to as torsion springs. The ends of torsion springs are attached to other components, and when those components rotate around the center of the spring, the spring tries to push them back to their original position. Although the name implies otherwise, torsion springs are subjected to bending stress rather than torsional stress. They can store and release angular energy or statically hold a mechanism in place by deflecting the legs about the body centerline axis.

This type of spring is normally close wound but can have pitch to reduce friction between the coils. They offer resistance to twist or rotationally applied force. Depending on the application, torsion springs can be designed to work in a clockwise or counter-clockwise rotation, thus determining the direction of the wind.

Common torsion springs are used in clothes pins, clipboards, swing-down tailgates, garage doors, window shades, counterbalance mechanisms, ratchets and various types of machine components. Torsion springs are used for hinges, counterbalances and lever return applications. They are also used as couplings between concentric shafts, such as in a motor and pump assembly. Torsion springs are generally mounted around a shaft or arbor, and must be supported at three or more points. Various kinds of ends are available to facilitate mounting.

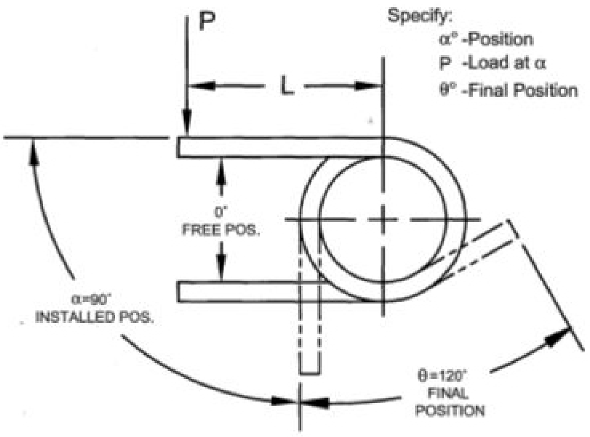


Figure 4.10: Torsion Spring representation and terminology.

With reference to Figure 4.10, a basic terminology can be introduce as follows: ***Angular Deflection*** is the angle of rotation as measured from the free position to the installed, intermediate or final positions; ***Free Angle*** is the angle between the arms of a torsion spring when the spring is in the unloaded position; ***Leg Length*** is the length of legs as defined from the axis of the spring body to the outermost point; ***Mandrel*** is a rod or shaft over which a torsion spring operates; ***Radius*** is the bend radius at which a load is applied to a leg. The radius is usually assumed to be equal to $\frac{1}{2}$ the leg length; ***Torque*** is a twisting action in torsion springs which produces rotation, equal to the load multiplied by the distance from the load to the axis of the spring body; ***Maximum Deflection*** is the maximum rated angular deflection of spring before damage.

Torsion springs are stressed in bending. Rectangular wire is more efficient in bending than round wire, but due to the premium cost of rectangular wire, round wire is preferred. Torsion springs, whose ends are rotated in angular deflection, offer resistance to externally applied torque. The wire itself is subjected to bending stresses rather than torsional stresses, as might be expected from the name. Springs of this type are usually close wound. The coil diameter reduces and body length increases as they are deflected. The designer must consider the effects of friction and arm deflection on the torque.

The number of active turns in a helical torsion spring is equal to the number of body turns, plus a contribution from the ends. For straight torsion ends, this contribution is equal to one-third of the moment arms and is usually expressed as an equivalent number of turns:

$$N_e = \frac{L_1 + L_2}{3\pi D_m} \tag{4.3}$$

with L_1 is the *length of moment arm 1*, L_2 is the *length of moment arm 2*, $N_a = N_b + N_e$ with N_b the *number of body turns.*

Many common end configurations are available on the market and special configurations are available generally on request. In designing ends, it is important to recall that bends, loaded to decrease their radius of curvature, have favorable residual stresses. They can operate at higher applied stress levels than bends that increase the radius by loading. Frequently, spring performance is limited because the sharply bent ends have greater stress than the body. Special types of torsion springs include double torsion springs and springs having a space between the coils to minimize friction. Double torsion springs consist of one set of coils coiled right hand and one set of coils coiled left hand. These coils are connected, usually with an unwound section between the winds and work in parallel. The sections are designed separately with the total torque being sum of the two.

4.2.2 Series Elastic Actuator (SEA)

In the real world, all force-controllable actuators have limitations that result in deviations from a perfect force source. These limitations include impedance, stiction, and bandwidth. An actuator's impedance is the additional force created at the output by load motion. Impedance is a function of the frequency of the load motion, typically increasing with frequency of load motion. Stiction describes the phenomenon of stick-slip friction, which is present in most devices where mechanical components are in sliding contact. Stiction must be overcome by a breakaway force, which limits the smallest force the actuator can output. The bandwidth of an actuator is the frequency up to which forces can be accurately commanded. Bandwidth is affected by saturation of power elements, mechanical stiffness and control system gain among other things. In an ideal force source, impedance is zero, stiction is zero and bandwidth is infinite. Muscle is the current best known actuation technology that approaches a perfect force source because it has extremely low impedance and stiction, and moderate bandwidth. An interesting review of the state-of-the-art force control technologies can be found at [39] in which are discussed traditional technologies for force control that include current control with direct drive actuation, current control with a geared actuator, current control with low-friction cable drive transmissions, load cells with force feedback and fluid pressure control. In the same work are briefly presented also the Series Elastic Actuators.

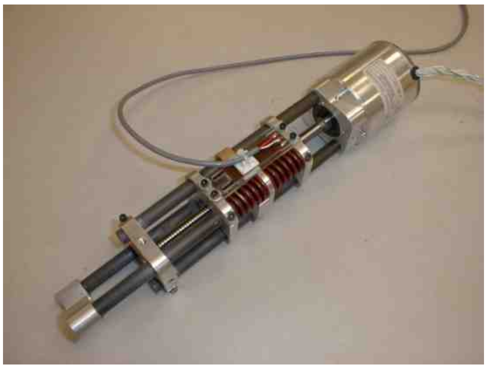


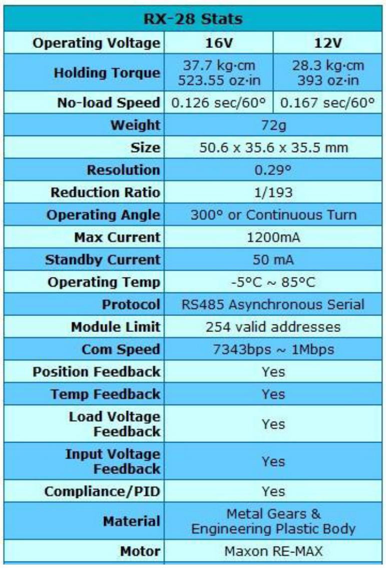
Figure 4.11: A SEA picture.

Series Elastic Actuators employ a novel mechanical design architecture which goes against the common machine design principal of "*stiffer is better*". A compliant element is placed between the gear train and driven load to intentionally reduce the stiffness of the actuator. A position sensor measures the deflection and the force output is accurately calculated using Hooke's Law ($F = Kx$). A control loop then servos the actuator

to the desired output force. The resulting actuator has inherent shock tolerance, high force fidelity and extremely low impedance. These characteristics are desirable in many applications including legged robots, exoskeletons for human performance amplification, robotic arms, haptic interfaces and adaptive suspensions.
More detailed descriptions about the design, construction, control and evaluation of a SEA are in [54], some results about the torque control of high compliant SEA can be found in [68], and then one of the most recent application in robotics research area is into BioBiped R1 project [48]. We discuss again about this type of actuator in chapter 5, with regard to its application into our bipedal robot.

4.2.3 Servo

The ***RX-28 Dynamixel Robot Servo Actuator*** is more than just a digital servo, it is a highly sophisticated robotic component. Each servo has the ability to track its speed, temperature, shaft position, voltage, and load. As if this weren't enough, the control algorithm used to maintain shaft position can be adjusted individually for each servo, allowing you to control the speed and strength of the motor's response. All of the sensor management and position control is handled by the servo's built-in microcontroller. This distributed approach leaves your main controller free to perform other functions.



RX-28 Stats		
Operating Voltage	16V	12V
Holding Torque	37.7 kg·cm 523.55 oz·in	28.3 kg·cm 393 oz·in
No-load Speed	0.126 sec/60°	0.167 sec/60°
Weight	72g	
Size	50.6 x 35.6 x 35.5 mm	
Resolution	0.29°	
Reduction Ratio	1/193	
Operating Angle	300° or Continuous Turn	
Max Current	1200mA	
Standby Current	50 mA	
Operating Temp	-5°C ~ 85°C	
Protocol	RS485 Asynchronous Serial	
Module Limit	254 valid addresses	
Com Speed	7343bps ~ 1Mbps	
Position Feedback	Yes	
Temp Feedback	Yes	
Load Voltage Feedback	Yes	
Input Voltage Feedback	Yes	
Compliance/PID	Yes	
Material	Metal Gears & Engineering Plastic Body	
Motor	Maxon RE-MAX	

Figure 4.12: RX-28 Dynamixel Robot Servo Actuator.

These datas and many others can be found at `http://www.trossenrobotics.com/dynamixel-rx-28-robot-actuator.aspx`.

4.2.4 Encoder

The ***AS5040*** is a ***contactless magnetic rotary encoder*** for accurate angular measurement over a full turn of 360°. It is a system-on-chip, combining integrated Hall elements, analog front end and digital signal processing in a single device. To measure the angle, only a simple two-pole magnet, rotating over the center of the chip, is required. The magnet may be placed above or below the IC. The absolute angle measurement provides instant indication of the magnets angular position with a resolution of 0.35° = 1024 positions per revolution. This digital data is available as a serial bit stream and as a PWM signal.

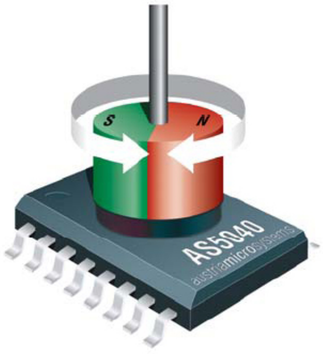


Figure 4.13: AS5040 contactless magnetic rotary encoder.

Furthermore, a user-programmable incremental output is available, making the chip suitable for replacement of various optical encoders. An internal voltage regulator allows the AS5040 to operate at either 3.3 V or 5 V supplies.

Benefits

- Complete system-on-chip;
- Ideal for applications in harsh environments due to contactless position sensing;
- Tolerant to magnet misalignment and airgap variations;
- No temperature compensation necessary;
- No calibration required.

Key Features

- Contactless high resolution rotational position encoding over a full turn of 360 degrees;
- Two digital 10bit absolute outputs: serial interface and pulse width modulated (PWM) output;
- User programmable zero / index position;
- Failure detection mode for magnet placement monitoring and loss of power supply;
- Rotational speeds up to 30,000 rpm;
- Wide temperature range: - 40°C to + 125°C;

Applications

- Industrial applications: contactless rotary position sensing, robotics, brushless DC motor commutation, power tools;
- Automotive applications: steering wheel position sensing, gas pedal position sensing, transmission gearbox encoder, headlight position control, power seat position indicator;

- Office equipment: printers, scanners, copiers;
- Replacement of optical encoders;
- Replacement of potentiometers.

All these data and many others are available at `http://www.austriamicrosystems.com/eng/Products/Magnetic-Encoders/Rotary-Encoders/AS5040`.

4.3 Musculoskeletal Robot Modeling

This bipedal robot is a novel elastic and biologically inspired four-segmented robot. It is principally based on two previous project: *Jena Walker II* and *BioBiped R1*, as previously presented. In fact, its ***total mass is about 2 Kg*** and its ***height is about 40 cm*** exactly as Jena Walker II. On the other hand, the mechanical design including the elastic structures spanning the ankle, knee and hip joint are inspired to BioBiped R1. From these, a small but significant improvement introduced by this robot, compared to existing models, is given without doubt from the presence of a fourth segment for each leg: the toes. This allows to model in much more detail the joints that involve the toes, foot and ankle. Consequently, the goal is to generate movements such as walking or running as close as possible to those made by human beings. It has also been radically rethought in the skeletal structure, while remaining of course linked to that of a human being; different technologies are used in terms of actuators and there are not pulleys or bionic drivers as those BioRob arm or BioBiped R1.

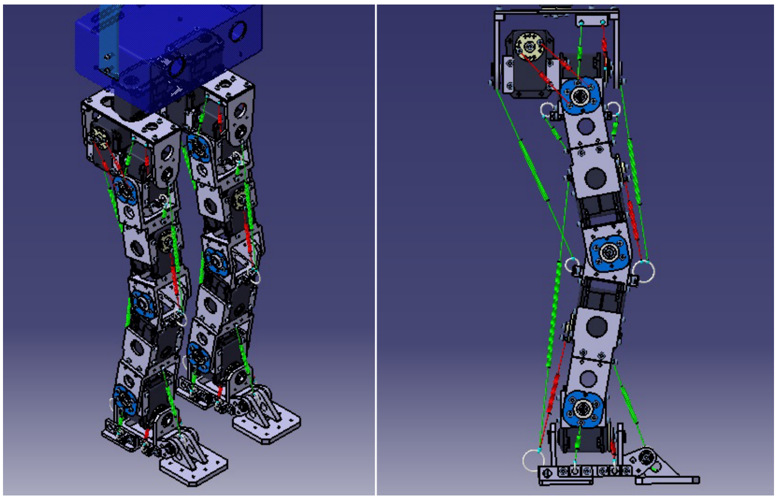


Figure 4.14: A novel elastic bipedal robot.

The bipedal locomotion model and the respective future real-world robotic platform consist of eight limb segments (***four segments in each leg***), plus one body segment called waist, which technically represents with its center of mass and dimensions the upper part of the skeleton. This model is equiped with ***12 RX-28 Dynamixel Robot Servo Actuators*** and relatives AS5040 contactless magnetic rotary encoder. In particular, as depicted in Figure 4.14, for each leg, there are:

- 1 servo between waist and hip bracket;
- 2 servo actuators into the hip joint, between the hip bracket and thigh body parts;

- 2 servo actuators, one into the thigh and one into the shank to actuate the knee joint;
- 1 servo actuators into the ankle joint, between the shank and foot body parts.

Complexively, it shows ***7 DoFs for each leg***, as depicted in Figure 4.15(a) in which we have:

- The first servo actuator provides 1 DoF between waist and hip bracket. It actuates the *internal/external rotation* of the lower limb.
- The second and third servo actuators provide 2 DoF between hip bracket and thigh. They actuate *flexion/extension* and *adduction/abduction movements*, respectively. That together the previous one correspond to the *hip ball and socket joint* with 3 DoFs in a human;
- The fourth servo actuator provide 1 DoF and it is placed into the thigh. It actuates the single movement of *knee flexion/extension* exactly as in a human;
- The fifth servo actuator provide 1 DoF and it is placed into the shank. It actuates the single movement of *foot dorsiflexion/plantarflexion*. That corresponds to the same *ankle revolute joint* between the tibia and talus in a human;
- The sixth servo actuator provide 1 DoF between shank and foot. It actuates the single movement of *adduction/abduction* of the foot. It corresponds to the *subtalar revolute joint* between the talus and subtalar in a human;
- The last DoF is obtained without servo actuator and it is placed into the toe articulation. That is similar to the so-called *mtp revolute joint* between calcaneus and toes defined by 1 DoF in a human.

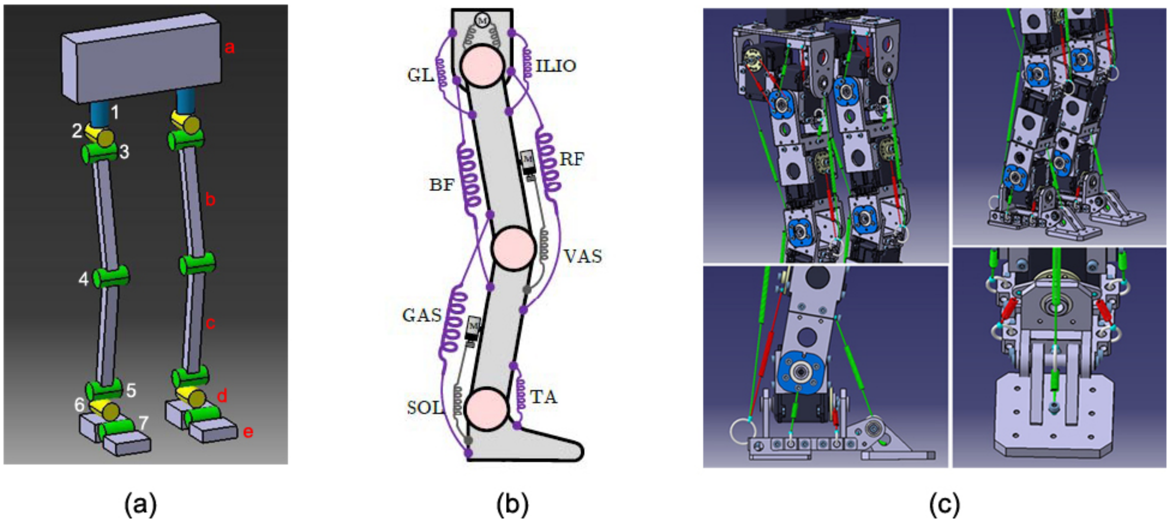


Figure 4.15: (a)The structure of the robot, with numbers (1-7) the Degrees of Freedom and with letters (a-e) the major parts of body. (b)A side view of the mechanical design of the robot including the elastic structures spanning the ankle, knee and hip joint. (c)Some snapshots of the elastic structures applied to the robot; Red is active and green is passive.

The robot systems consists also of elastic structures that represent the major muscle groups as depicted in Figure 4.15(b): *iliopsoas* (ILIO), *gluteus* (GL), *rectus femoris* (RF), *biceps*

femoris(BF), *vastus* (VAS), *gastrocnemius* (GAS), *tibialis anterior* (TA) and *soleus* (SOL). These are the main muscle-tendon groups of the three segmented human leg that span one or two leg joints. Their functionality is mimicked by ***38 elastic cable spring structures*** in our robot design. Similarly to Jena Walker II, except for TA and SOL, all muscle groups span two joints leading to an inter-joint coupling within the leg. Moreover, the monoarticular extensor muscle groups (VAS and SOL) are implemented as ***Series Elastic Actuators*** (SEAs), powered by electric DC motors. In this case, similarly to BioBiped R1, in order to feed the ankle and the knee joint with energy, and so to support them during bending with motor power.
In our robot the spring structures can also be divided in two subset: ***passive springs*** and ***active springs***. The first one refers to a spring connected between two body parts, on the other hand, an active spring refers to a spring that is connected on one side to a servo motor and on the other side to a load represented from a body part. A such type of spring is directly extended from a servo motor activation. This division can also be referred to joints as ***monoarticular joint*** and ***biarticular joint*** as shown in Figure 4.16. If we look at the case (a), when you extend the spring, it can create the force and restore the energy to prevent the elongation. If the actuation is over or the motor stops, the spring will release the energy and drive the joint load. It seems that the spring only transmit the energy from the motor to the mechanism, and also consumes a little, but it will make joint move more smoothly and sufficiently. On the other case (b), we have a servo motor connects with a joint by two groups of springs which reproduce an antagonist behavior where A1 and A2 are active springs while B1 and B2 are passive ones. With this configuration each motion involves one or more active spring and on or more respective passive spring that aim to obtain a more compliant behavior during complex movement such as jumping, walking or running gaits.

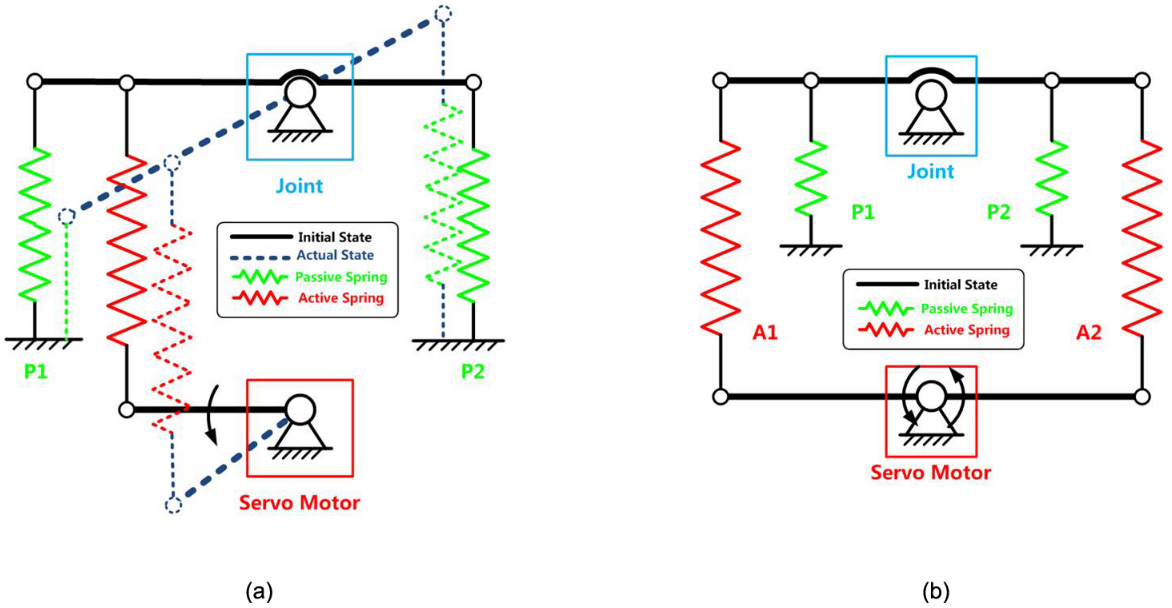


Figure 4.16: (a)Representation of a mono-articular joint; Red is active and green is passive. (b)Representation of a bi-articular joint; while A1, A2 are active, B1 and B2 are passive.

Two passive springs are applied to keep the equilibrium of the joint before motions which is called the initial state. At this time, both springs are stretched. When the servo motor rotates, the active spring and the passive spring in the opposite (P2) are elongated. Note that spring (P1) is negative, for we choose the tension spring as the elasticity. The difference of bio-articular joint from mono-articular is that there is one more active spring in the structure, and the motor can rotate on both sides which means that the it can create motions in two different directions. Hence we can say that the bi-articular joint is the extended mono-articular joint in some way.

Chapter 5

Expected Control System Design

After all that we have discussed in the previous chapters about movement, locomotion, elastic bipedal locomotion and elastic bipedal robot design, in this chapter we provide a description of the expected control system design for our bipedal robot. All themes treated in this chapter comes from the original project realized by Fuben He for his PhD thesis. As author of this work, I provide a briefly overview of different elastic actuation models and describe the expected control structure with reference to Fuben work.
We see the *Original Series Elastic Actuators* (SEA) Model introduced by Prof. Gill A. Pratt and Matthew M. Williamson for his Master of Science at the MIT. Then we see the *Extended SEA Model* based on works by Prof. Oskar von Stryk and Andre Seyfarth. Last but not least comes the *Musculotendon Model*, for which we refer to several works by Prof. Oussama Khatib, Scott L. Delp and many others. In particular, we mean this last section as a sequel to how discussed in Chapter 2 for human and robot movement, so to close the loop with the control system design.

5.1 Comparison of Different Elastic Actuation Models

Citing M. M. Williamson: "*The traditional premise for good robot design is 'Stiffer is better'* "...[but]..."*Stiffness isn't everything*" [54].
As we have seen in Chapter 2 there are at least two fundamental problems in legged locomotion. The way they move is not at all energy efficient. A lot of effort is put in the swinging of the leg without reusing the negative work for the deceleration of leg. Then, at the end of each swinging movement, a certain amount of energy gets also lost in the form of collision energy. Another big disadvantage of legged robots is the need of much more control effort. In a well known environment, controlling the position of the actuated robot joints might be sufficient, but not all environment are known a priori [68].
As the robots are stiff, their links tend to be heavy, so large forces are needed to accelerate them. Electric motors, which are the most common actuator type cannot generate large forces at low speeds, so gear reductions need to be used. Consequently, the power density increases, introducing friction, noise, backlash and torque ripple to the system. These effects will be transmitted to the endpoint of the robot, giving poor performance. In the last years, many researchers have investigated in a possible alternative and one of the most advanced solution comes from human studies. In fact, human are good at force control because the humans have a low stiffness, low bandwidth system, compared

to the high bandwidth, high stiffness robot. The solutions proposed in literature can be implemented by placing an elastic element into the actuator.
While stiff actuator with torque sensor uses a torque sensor that measures the load acting on the joint directly at the output of the actuator [17, 18, 2], in SEA the actuator is extended by a spring in series to the output of the transmission decoupling the link from the actuator. Different groups [39, 28] came to the conclusion that this last actuation method is ideal for a lot of application, especially for walking robots.

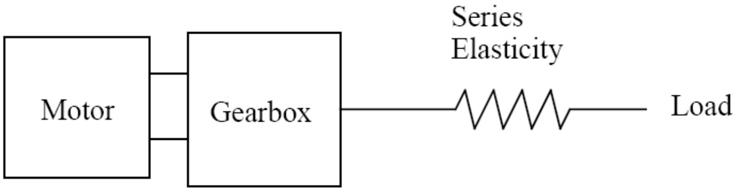


Figure 5.1: Schematic of Series Elastic Actuator. [54]

The force control become easier with elasticity, as larger deformations of the robot structure are needed to exert the same forces as a stiff robot. The force control problem turns into a position control one, which greatly improves force accuracy. The motor position determines the force, because the output force is proportional to the twist in the spring. The control action reduces the reflected inertia, protecting the gearbox from damage. Introducing series elasticity also makes stable force control easier to achieve. The performace of robots can be improved by using SEA, to reach a human-like behavior.

5.1.1 Original SEA Model

A simple model for an actuator is a spring in series with a stiff actuator. The compliance of this actuator is fixed and determined by the selection of the spring, so the physical compliance cannot be changed during operation. A schematic of this model is shown in Figure 5.2.

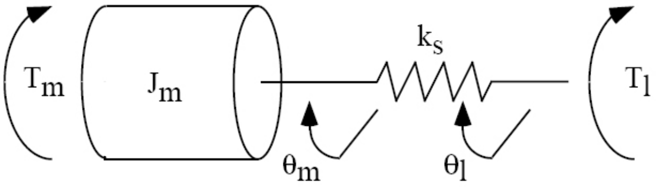


Figure 5.2: Model of actuator. [54]

In this picture we can see the model of the motor mass J_m, the spring stiffness k_s, the force on the motor T_m, the output force T_l, the movement of motor shaft θ_m and the movement of load θ_l. By applying Newton's Laws, we can exctract immediately some relations from the diagram.

$$T_m + k_s(\theta_l - \theta_m) = J_m\ddot{\theta}_m \tag{5.1}$$

$$-k_s(\theta_l - \theta_m) = T_l \tag{5.2}$$

Then, an expression relating T_m and T_l can be found by taking Laplace transforms:

$$T_m(s) = \left(1 + \frac{J_m}{k_s}s^2\right) T_l(s) + J_m s^2 \theta_l(s) \tag{5.3}$$

This equation is important because it shows what motor torques are needed to give an output torque of T_l, when the output of the actuator is moving. It also shows what the componenets of T_m are. In fact, if the output of the actuator is assumed clamped ($\ddot{\theta}_l = 0$), then the transfer function between output torque and motor torque is:

$$\frac{T_l}{T_m}(s) = \frac{1}{1 + s^2 J_m/k_s} \tag{5.4}$$

The trasfer function between the actual output force T_l and the motor force T_m has no zeros, and two poles on the imaginary axis, at a frequency $w = \sqrt{k_s/J_m}$ which corresponds to the natural frequency of the motor mass and the spring.
Again, by taking Laplace transform, also the transfer function between the motion of the output shaft θ_l and the output force T_l can be written.

$$Z(s) = \frac{T_l}{\theta_l}(s) = \frac{-s^2 J_m}{1 + s^2 J_m/k_s} \tag{5.5}$$

Generally, the ratio T_l/θ_l is defined as the impedance Z of the system, looking from the output. This is an important parameter when the stability of the control system is analysed [54]. This transfer function has the same poles as the T_l/T_m, but it also has two zeros at the origin. The negative sign comes from the definition of the directions of T_l and θ_l.
The equations 5.4 and 5.5 define the model of the system to be controlled as it shown in the Figure 5.3.

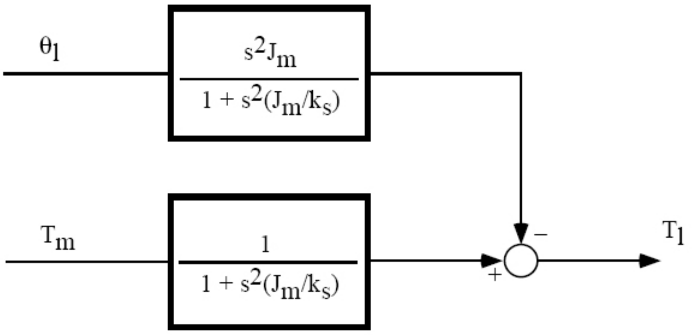


Figure 5.3: Plant model. [54]

With reference to Eq. 5.3, this is a feedforward model, but to increase the performance of this system, to compensate for errors and to reject unmodeled disturbances, a feedback control is required. It is important that the system is stable, and especially, the system must be stable when in contact with all environments. And so, with the introduction of a PID controller, the plant model shown in Figure 5.3 becomes a closed loop system. In particular, the form of PID assumed in [54] is $K(1 + 1/sT_i + sT_d)$.

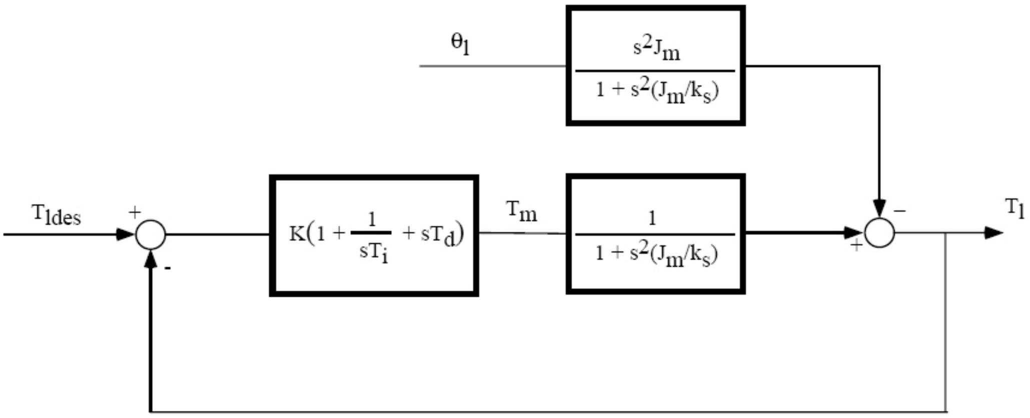


Figure 5.4: Closed loop system. [54]

5.1.2 Extended SEA Model

This model represents a bionic drives consisting of a DC motor that is elastically coupled to the joint with antagonistic, elastic pulleys with progressive angle-torque characteristics, as illustrated in Figure 5.5. It is originally inspired by the functional principles of the elastic and antagonistic muscle and tendon apparatus of the human arm [7] and subsequently extensively tested in a real manipulator, the BioRob arm [72] and simulated in animal-like four-legged robot [46].

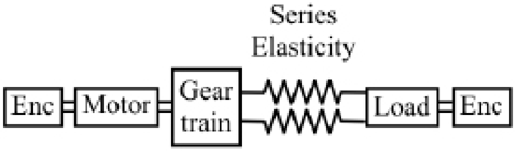


Figure 5.5: Schematic of the Extended Series Elastic Actuator with only one motor. [48]

It is considered as an extended series elastic actuator with only one motor, offering, however, compared to the original SEAs, different possibilities of feedback and feedforward control. In contrast to SEAs, play and backlash can be reduced by pretension in the equilibrium position and angular sensors in the joints enable higher positioning accuracy. Another advantage is the reduction of damages of the motors as the elasticity low-pass filters shock loads, protecting the gearbox from damage. In conventional robots additional masses are introduced by the motors that are directly coupled to the joints. These masses potentially damage the structures at higher speeds because they increase the effective mass during landing impact which lead to higher loading rates. For this reason, such actuators are not used for fast locomotion. A potential solution, presented in literature, to this problem is to shift the motors proximally and to decouple the motors mechanically from the rigid segments by tendons and springs. In such way, the joint becomes stiffer when accelerating and decelerating, having a positive effect on the performance, because it uses a progressive spring characteristic [41, 65]. However, the joint torque measurement is less accurate than in SEAs, because an inverse model of the spring characteristic is needed. So, for example, with reference to the BioRob Robot arm, thanks to Extended

SEA, a model such that in Figure 5.6 can be transformed from A (left side of picture) to B (right side of picture). This transformation can be performed if the mass of the cables and elastic parts is so small that the kinetic energy of these elements can be neglected compared to the kinetic energy of the other mechanical robot arm/leg parts.

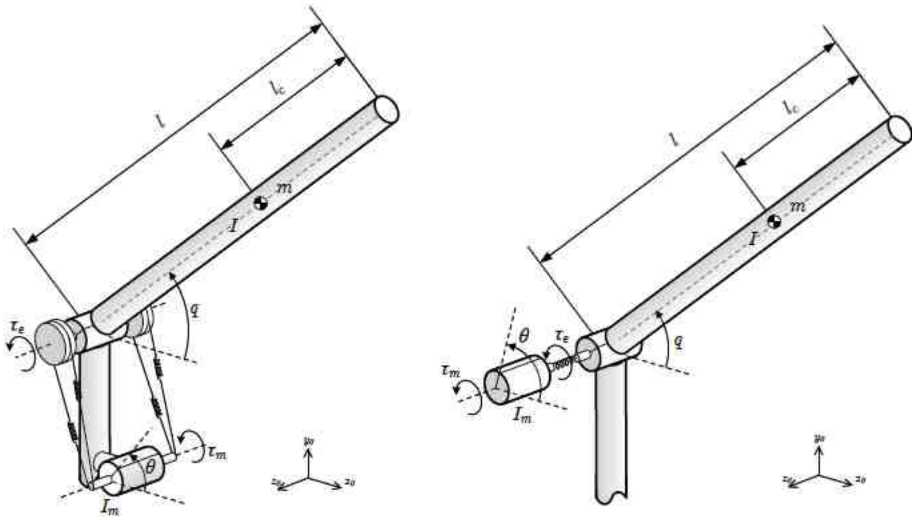


Figure 5.6: Schematic transformation. [79]

It is possible to carry out an analytical inverse dynamics approach using numerical computing environments in order to determine the desired torques τ_m that are necessary to produce the specified motions, that is the system input. In this model, the joint elasticity equation (nonlinear joint spring characteristics curve) can be represented by a function of the deviation of the joint position q_i of its equilibrium position $\widehat{q_i}$, normally called motor position θ_i, but can also be dependent of the position of previous joints:

$$\tau_e = k_e(\widehat{q} - q) \tag{5.6}$$

At this point, by using the reduced model of elastic joint robots, the multibody dynamics of the rigid structures and the motors can be described:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + g(q) = \tau_e \tag{5.7}$$

$$I_m\ddot{\theta} + D_m\dot{\theta} + \tau_e = \tau_m \tag{5.8}$$

where, respectively, for the Eq. 5.7 M is the mass matrix, C is the Coriolis matrix, g is the gravity torque vector and D is the diagonal friction matrix; and for the Eq. 5.8 I_m is the diagonal motor rotor inertia matrix, D_m is the diagonal friction matrix, and τ_m is the motor torque.
From the joint elasticity equation 5.6 it is possible to calculate the link equilibrium positions with the desired link trajectory:

$$\hat{q}_d = k_e^{-1}(\tau_{e,d}) + q_d \tag{5.9}$$

and then, by applying the rigid link dynamics equation 5.7 and transforming the equilibrium positions in motor positions, to obtain the desired motor trajectory that produces the given desired joint trajectory $q_d(t)$.

$$\theta_d = k_e^{-1}\left(M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + D\dot{q}_d + g(q_d)\right) + q_d + \alpha_c(q_d) \tag{5.10}$$

The desired motor torques $\tau_{m,d}$ can then be calculated through Eq. 5.8:

$$\tau_{m,d} = I_m\ddot{\theta}_d + D_m\dot{\theta}_d + \tau_{e,d} \tag{5.11}$$

This controller structures uses a global nonlinear calculation of the motor setpoint, which linearizes each joint around the current desired position for all states to receive damped and exact steady state behavior. So, the desired link trajectory q_d and the desired motor trajectory θ_d can be used for a controller as shown in Figure 5.7.

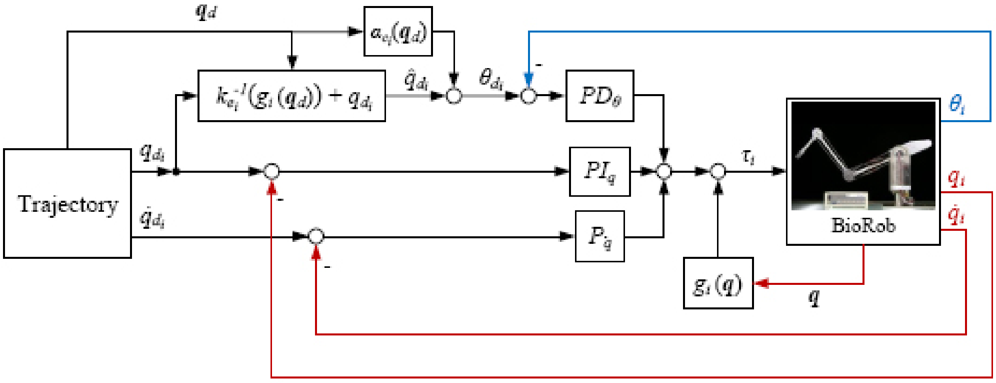


Figure 5.7: Control structure for joint i. [79]

A more detailed description of control approach and simulation experiments can be found at [79].

5.1.3 Musculotendon Model

With these type of models we look at the robotic research areas in which approaches from the biomechanics and robotics communities are integrated to address the challenge of synthesizing low-level human motion control from high-level commands. Nowadays, muscle strength limitations, activation delays and overall muscle contraction dynamics can be described thanks to the characterization obtained from the biomechanics community, by using computational muscle models. On the other hand, the robotics community has investigated the task-level feedback control of robots using the operational space approach. It recasts the dynamics of the robotic system into a relevant task space description [60, 22, 21]. This is interesting for us because the integration of these approaches into a unified framework offers a promising methodology for synthesizing goal-directed human-like motion control.
Now, with reference to the musculoskeletal model presented in chapter 2 and commonly used in literature we can briefly describe a unique integration of these model into a task-level control framework, based on [81, 52, 61]. By considering a set of generalized

coordinates q, usually taken to be the joint angles between limb segments, the configuration of the skeletal system modeled as system of constrained rigid bodies can be described. The n equations of motion can be represented in standard form as:

$$\Gamma = A(q)\ddot{q} + b(q, \dot{q}) + g(q) \tag{5.12}$$

where Γ is the set of *joint torques*, $A(q)$ is the system *mass matrix*, $b(q, \dot{q})$ is the vector of *centrifugal* and *Coriolis terms* and $g(q)$ is the vector of *gravity terms.*
The next step is to assign a system of musculotendon actuators to the skeletal system described by the previous dynamical model. By assuming that all musculotendon lengths l can be uniquely determined from the system configuration q, that is $l = l(q)$, differential variations in l are given by:

$$\delta l = L(q)\delta q \tag{5.13}$$

where $L(q)$ is the *muscle Jacobian.* Then, by applying the Principle of Virtual Work:

$$\Gamma = -L^T f_T \tag{5.14}$$

where f_T is the vector of *net muscle forces*, with active and passive components. The negative sign is due to the convention of taking contractile muscle forces as positive.

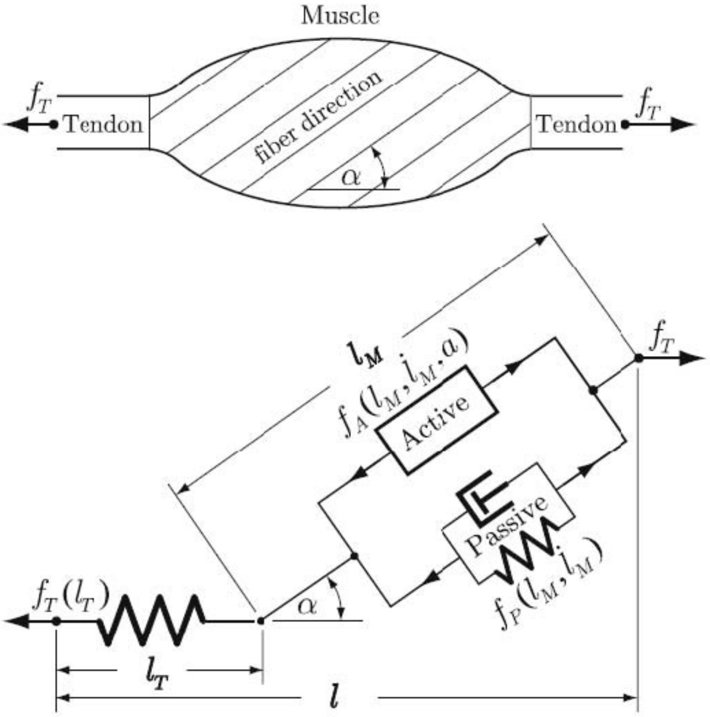


Figure 5.8: Active state musculotendon model. The active contractile element and passive viscoelastic element are in parallel. The passive elastic tendon element is in series [81].

Considering an Hill-type active state model the described net dynamic effect can be modeled. They can be divided into active dynamics and musculotendon contraction dynamics. Generally, neuromuscular dynamics refer to the time-evolution of muscle states (length and force) in response to neural excitation (control input).

Activation Dynamics

It refers to the process of muscle activation in response to neural excitation which can be modeled by the terms of the activation, a:

$$\dot{a} = \frac{u - a}{\tau(u, a)} \tag{5.15}$$

where $u \in [0, 1]$ is the *neural input.* The term $\tau(u, a)$ is a *time constant* given by:

$$\tau(u, a) = \begin{cases} (\tau_a - \tau_d)u + \tau_d & , u \geq a \\ \tau_d & , u < a \end{cases} \tag{5.16}$$

where τ_a, τ_d are the *activation* and *deactivation* constants, respectively.

Contraction Dynamics

It refers to the process of force generation in the muscle based on muscle contraction, rate of contraction and activation. Based on the geometry of Figure 5.8, we have the following relationships:

$$l(q) = l_M \cos\alpha + l_T \tag{5.17}$$

$$\dot{l}(q, \dot{q}) = \dot{l}_M \cos\alpha + \dot{l}_T \tag{5.18}$$

where l_M is the *muscle length*, l_T is the *tendon length* and α is the *pennation angle.* We have also the following force equilibrium equation:

$$f_T = (f_A + f_P) \cos\alpha \tag{5.19}$$

Consequently, an equation of state can be expressed in the following functional form:

$$\dot{l}_M = \dot{l}_M(l(q), \dot{l}(q, \dot{q}, l_M, a) \tag{5.20}$$

Combining neuromuscular dynamics with multibody skeletal dynamics a model of the overall musculoskeletal system can be obtained. This process is described by the feed-forward path of Figure 5.9. In which a given set of neural inputs u is the starting point for an activation dynamics block in which a set of muscle activations a arises. This set of muscle activations, as well as the skeletal configuration q and $\dot{q}$ is the input for a next block, called contraction dynamics from which a set of muscles forces arises. These forces, related to the joint torques Γ through the muscle Jacobian L drive the skeletal dynamics resulting in motion of the system.

A useful simplification to the full musculoskeletal dynamics described is to make the assumption that the tendon is infinitely stiff. Then l_M is not an independent state, but it is related to the overall musculotendon length, $l(q)$. In this way the muscle forces can now be expressed in terms of this gain:

$$f_T(q, \dot{q}, a) = f_P(q, \dot{q}) + K_f(q, \dot{q})a \tag{5.21}$$

where $f_P = (f_{P_1} \cos\alpha_1, ..., f_{P_r} \cos\alpha_r)^T$ and $K_f = \frac{\partial f_T}{\partial a} = f_S \cos\alpha$ is the *force-activation gain* defined as the magnitude of force generation in the muscle per change in unit activation, where $f_S(q, \dot{q}) = f_A(q, \dot{q}, 1)$ is the *muscle saturation force.*

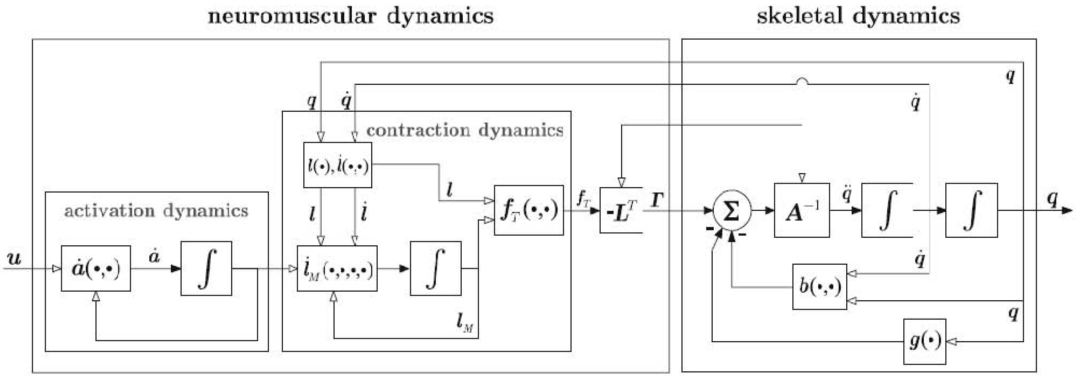


Figure 5.9: Musculoskeletal system(feed-forward path). Neural excitations provide input to the activation dynamics. Output of the activation dynamics provides input to the contraction dynamics. Output of the contraction dynamics provides input to the skeletal dynamics through the joint torques [81].

Task-level Control Framework

A major limitation comes from employing inverse kinematic methods, which are unable to control task impedances, a key element for whole-body contact interactions. Another limitation comes from the lack of techniques that can monitor behavior feasibility and solve scenarios where the global behavior is infeasible under the acting constraints.
In response to these limitations, O. Khatib, L. Sentis et al. proposed a prioritized control approach based on operational tasks [52]. This hierarchical approach prevents lower priority tasks from interfering with higher priority tasks, and provides the means to monitor behavior feasibility at runtime. In this framework, a task is any formal description of desired activity that can explicitly be represented as a function of the generalized coordinates. This could be as simple as specifying the position of a limb to be in a certain location. Multiple tasks can be combined into a single task definition, as long as they are kinematically consistent with each other. A much more detailed mathematical description of this approach can be found at [81, 59, 52] where starting from the previous overall musculoskeletal dynamics modeling, based on generalized coordinates, a task coordinates description is obtained.

5.2 Expected Control Structure

In summary, if we consider this briefly overview of different elastic actuation model is note worth that in our robot model:

- We apply the extension springs to the robot, also if in the musculotendon model the elasticity is similar to the compression spring;
- There is no fiber either;
- The actuation is built up based on the extended SEA principle with encoders to confirm the desired position;

- Both the controllers in the original and the extended SEAs can be recommended for our control system;

- A basic PID controller is also necessary as it can be implemented easily and rapidly in recent.

The expected control structure basically obeys the original SEA feedforward control theory and a PID controller is also introduced, the sensor will provide data for the controller as feedback compensate.

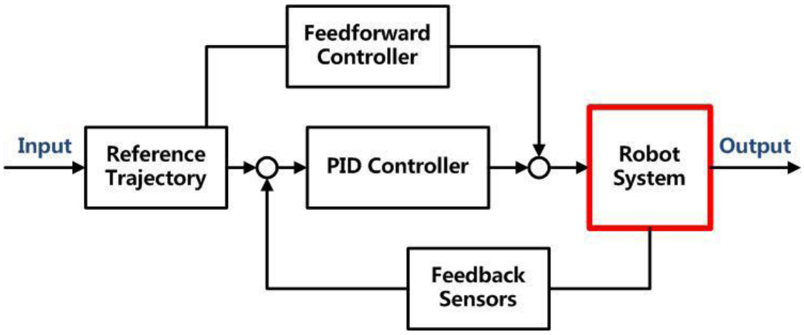


Figure 5.10: Expected Control Structure.

Chapter 6

OpenSim Modeling and Simulation

This chapter ends the part of thesis concerning the development of robots designed by Fuben He with the modeling and simulation of movement realized by the author of this thesis through the OpenSim platform, based on the human musculoskeletal models.
We started to study a generic Human Musculoskeletal Model provided with OpenSim platform. This model has been simulated with some basic movement and with a more complex movement like the walking gait. Starting from it we mapped its 12 compartments of muscles in 7 compartments of springs in the robot, based on the movements in which they are involved. Then we simulated the robot with some basic movements and static poses. In the next step the kinematic of human walking movement was mapped into the kinematic of the robot to obtain a walking movement for our elastic bipedal robot. We provide below a detailed description of our workflow and some results obtained from it. There are also four appendicies in which we have illustrated each single parts, the assembly phases of the model with their specific physical properties, and some source code examples.

6.1 Human Musculoskeletal Model

We used a computer model of the musculoskeletal system that represents the geometry of the bones, the kinematics of joints, and the lines of action and force-generating properties of lower limb muscles. Given patterns of muscle activations and joint angles, the musculoskeletal model calculates the muscle-tendon lengths, muscle forces, tendon strains and muscle fiber lengths in a dynamic simulation.
The model includes the geometry of the bones of the lower limb and pelvis, created by digitizing the bones of a male subject by Delp S.L., Loan J.P., Hoy M.G., Zajac F.E., Topp E.L., Rosen J.M., Thelen D.G., Anderson F.C., Seth A., and then provided with the OpenSim software. In particular, the lower extremity joint definition is based on Delp et al. (1990), the low back joint and anthropometry is based on Anderson and Pandy (1999, 2001), the planar knee model of Yamaguchi and Zajac (1989) and then Seth removed the patella to avoid kinematic constraints; insertions of the quadrucepts are handled with moving points in the tibia frame.
The original unscaled model shows bone dimensions of a 175 cm tall and 75.1646 Kg weight male. The model also includes representations of the mtp, subtalar, ankle, knee and hip joints that define motions between the bones.

- The mtp is a revolute joint between calcaneus and toes defined by 1 DoF.
- The subtalar is a revolute joint between the talus and subtalar defined by 1 DoF.
- The ankle is a revolute joint between the tibia and talus defined by 1 DoF (*dorsiflexion/plantarflexion*).
- The knee has a single DoF (*flexion/extension*) and uses the equations reported by Yamaguchi G.T., Zajac F.E. to define the translations and rotations between the femur, tibia and patella as functions of knee flexion angle.
- The hip is a ball and socket joint with 3 DoF (*flexion/extension*, *adduction/abduction* and *internal/external rotation*).

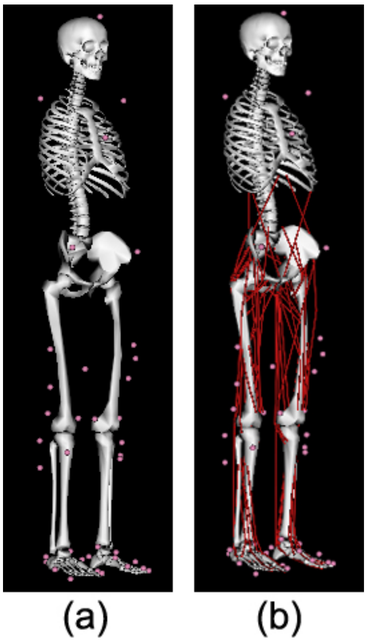


Figure 6.1: Human musculoskeletal model used. (a) Torque-driven model of a human subject; (b) Muscle-driven model of a human subject.

Thus, each leg in the model has ***7 DoF***. The model includes ***86 muscles of the lower limb***, 43 for each leg plus 6 muscles between pelvis and torso. Line segments approximate the muscle-tendon path from the origin to insertion and they are grouped in 12 compartments for each leg, based on their functions (i.e. R_hip_flex, L_knee_ext).
We use a lumped parameter model to characterize muscle-tendon contraction dynamics. This model includes four parameters (*optimal fiber length, maximum isometric force, pennation angle and tendon slack length*) that scale generic properties of muscle and tendon to represent the architecture of each muscle-tendon unit. The model of tendon representes the nonlinear elastic properties of tendon. The tendon force-strain relationship is scaled to represent a specific muscle-tendon complex by *tendon slack length* (L_S^T) and *peak muscle force* (F_0^M). *Tendon strain* is assumed to be 0.033 when muscle generated its *peak isometric force*. The model of muscle includes the active and passive *force-length relationships*, which were scaled by each muscle's optimal fiber length (L_0^M) and peak

isometric force (F_0^M). The force-velocity relationship is also included: the *maximum shortening velocity* of each muscle is assumed to be 10 optimal fiber length per second. The parameters used to scale properties of each muscle-tendon unit, optimal fiber length and pennation angles, maximum isometric forces and the anthropometry of the subject, but also marker trajectories and model are taken from OpenSim platform that provides, among others, some models of the lower limb with the software itself.

6.1.1 Scaling

We scaled the model to the anthopometry of the subject based on marker locations. Optimal fiber length and tendon slack length are scaled with muscle-tendon length so that they maintain the same ratio. This process starts with the unscaled OpenSim musculoskeletal model and places a set of virtual markers on the model to match the locations of the experimental markers. Marker trajectories collected during the static trials are used in the scaling process to adjust a number of variables of the OpenSim musculoskeltal model including:

- length of each bone and muscle;
- position of the centre of mass of each bone;
- mass of each segment.

The unscaled model has predefined weight, height, inertia and position of centre of mass for each body segment. Scaling factor is obtained by computing the ratio between the subject's segments mass and dimension and the generic OpenSim model segment mass and dimension. The dimension of the subject's segments is estimated by computing the distance between the centres of the joints the segment is connected to [55]. The mass of each segment is derived from the subject total mass using anthropometric tables. In this way, starting from the original unscaled model we obtained a new one referred to a human subject height about 180 cm and 72.6 Kg of weight. The scaling process is an OpenSim built-in tool.

6.1.2 Inverse Kinematics

Marker trajectories recorded during dynamic movements and saved in a .trc file are used by an Inverse Kinematics (IK) model to calculate 3D joint angles. Similarly to scaling process, also this step is done using the OpenSim Inverse Kinematics tool in which an IK algorithm solves for the joint angles that minimized the difference between the experimentally measured marker positions and the virtual markers on the model. A more detailed description of this tool is provided in chapter 3.

6.1.3 Residual Reduction Analysis

Ground reaction forces (GRF) recorded during motor tasks are used in conjunction with the three-dimensional joint angles computed through IK to derive the experimental joint moments. This is usually done by performing standard Inverse Dynamics (ID). However, this method does not provide optimal solutions. Due to limitations in marker trajectory

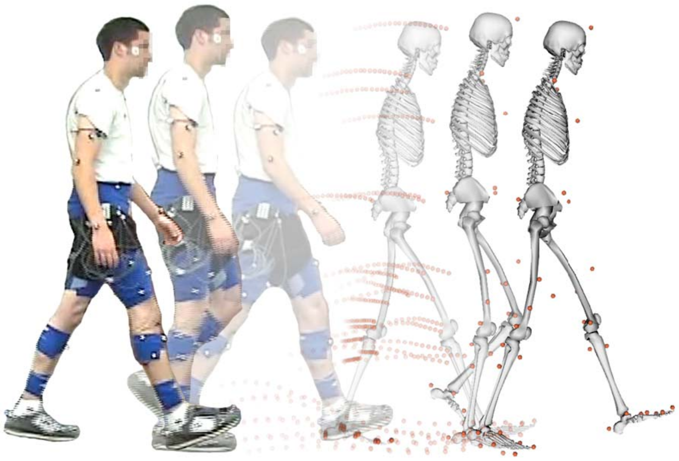


Figure 6.2: Inverse Kinematics Tool Overview. Experimental markers are matched by model markers throughout the motion by varying the generalized coordinates (e.g., joint angles) through time. [This image is taken from OpenSim Handout SIMPAR 2010]

acquisition and processing, there is an inherent mismatch between the recorded trajectories and the recorded GRF. As a results, in traditional ID algorithms, a non-physical external force and moment (residuals) are applied to a body in the model to resolve dynamic inconsistency between the measured kinematics and GRF. This implies that, the moments computed via ID do not match the motion computed via IK. In this work Residual Reduction Analysis (RRA) is used to minimize the mismatch between trajectories and GRF and to compute the joint moments needed to track the subject's motion. RRA, as described in Chapter 3, is an optimization procedure that slightly adjusts the joint kinematics and model mass properties until an inverse dynamic solution is found that minimizes the magnitude of the residuals.

6.1.4 Computed Muscle Control

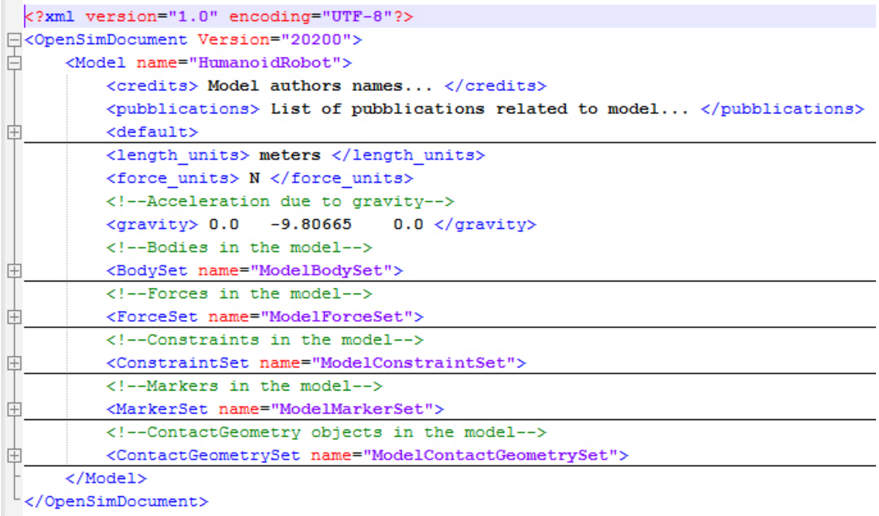
In this step we produce a dynamic simulation of muscle-tendon dynamics during walking. We prescribe muscle-activation patterns and joint kinematics, and calculate the muscle forces and fiber lengths that satisfy these constraints.

To study muscle behavior we produce simulations with three different activation cases: maximum activation, minimum activation and typical activation during gait. In the muscle model used, activation is a value between 0.0 and 1.0. In the maximum activation case, activation is 1.0 in all muscles. For the minimum activation case, it is not possible to prescribe 0.0 activation for the simulation of walking because the fibres must maintain tension while the muscle-tendon complex is shortening. Minimum activation are typically 0.05, however muscles that reach very high shortening velocities, like soleus, gastrocnemius or rectus femoris for example, demand higher values (i.e. 0.1, 0.15). In the typical activation case, we prescribe activations of muscle based on electromyography (EMG) data reported in literature.

Considering the walking motion the most interesting purposes of our research and the ultimate goal, we decided to subdivide such a movement in elementary ones on which to repeat the studies and analysis described above. This choice is also justified especially by the fact that our goal is to compare and validate the results obtained from humanoid robot with those obtained in human, which is a model already thoroughly validated and recognized in literature. Simple movements allow a more detailed study of the capacity expressed by the robot, at least at this early stage of development; on the other hand, a first analysis of a complex movement gives us useful information on the potential of the model itself.

6.2 Elastic Bipedal Robot Model

An OpenSim model represents the dynamics of a system of rigid bodies and joints that are acted upon by forces to produce motion. The OpenSim model is made up of components corresponding to parts of the physical system. These parts are: bodies, joints, forces, markers, constraints, contact geometry, and controllers. In formulating the equations-of-motion (the system dynamics), OpenSim employs Simbody via the SimbodyEngine, where the body is the primary building block of the model. Each body in turn owns a joint and that joint defines the coordinates and kinematic transforms that govern the motion of that body. Within the model all bodies are contained in a BodySet. The ConstraintsSet contains all the kinematic constraints that act on bodies (and/or their coordinates) in the model. User forces acting on the model are all included in an ActuatorSet.
An OpenSim model is described by a file that utilizes the XML code structure to organize its contents. XML uses tags to identify and manage information, such as <body> hip_bracket_r </body> where <body> signifies the opening of the tag, hip_bracket_r is the name of a body in the model, and </body> signifies the end of the tag. The name of the tag identifies the type of information between described.

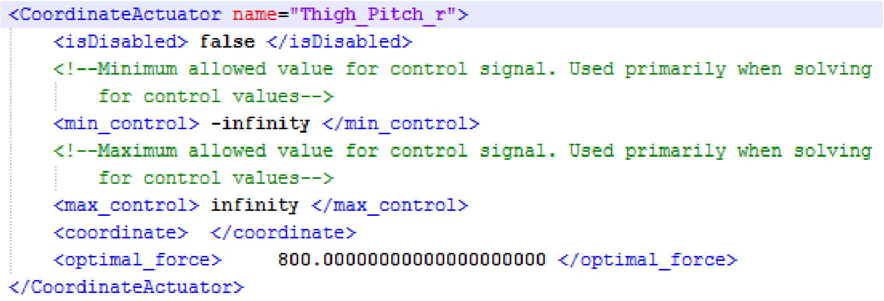


```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSimDocument Version="20200">
    <Model name="HumanoidRobot">
        <credits> Model authors names... </credits>
        <pubblications> List of pubblications related to model... </pubblications>
        <default>
        <length_units> meters </length_units>
        <force_units> N </force_units>
        <!--Acceleration due to gravity-->
        <gravity> 0.0    -9.80665     0.0 </gravity>
        <!--Bodies in the model-->
        <BodySet name="ModelBodySet">
        <!--Forces in the model-->
        <ForceSet name="ModelForceSet">
        <!--Constraints in the model-->
        <ConstraintSet name="ModelConstraintSet">
        <!--Markers in the model-->
        <MarkerSet name="ModelMarkerSet">
        <!--ContactGeometry objects in the model-->
        <ContactGeometrySet name="ModelContactGeometrySet">
    </Model>
</OpenSimDocument>
```

Figure 6.3: Typical structure of an OpenSim model.

To start our model, we need to define a set of *rigid bodies* that represent our system. In the <BodySet> section, we define this group of bodies, with the name and mass properties. Then we set the visible objects and geometry file associated with each body.
In addition to the set of rigid bodies, we also need to define the relationship between

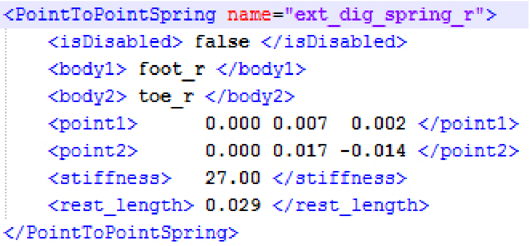
them. *Joints* are defined within each body and it defines how a body can move with respect to its parent body. All bodies have a parent and are connected to it via a joint, except for ground. A body has a moving reference frame (Bo) in which its center-of-mass and inertia are defined, and the location of a joint fame (B) fixed to the body can be specified. Similarly, the joint frame (P) in the parent body frame (Po) can also be specified. Additional flexibility in defining the joint is achieved by permitting joint frames that are not coincident with the body frame. Constraints limit the motion of bodies. For example, in our model the body Waist contains a joint called Free Joint which is caracterized by 6-degrees-of-freedom. Where <CoordinateSet> tag contains all property of each degree-of-freedom. There are also other built-in type of joint into OpenSim: Ball Joint, Ellipsoid Joint, Pin Joint, Slider Joint, Weld Joint and Custom Joint. In particular, the *Custom Joint* is the most used in our model because it allows us to define and customize the relationship between each bodies. In a same manner we have done for all the other parts of the model following the steps that you can find in Appendix B. Finally, we used also a so called *Weld Constraint* between waist and ground. Binding the model, we simplify the study that will not have to take into account the ground reaction forces due to interaction of the robot with the ground.



```
<CoordinateActuator name="Thigh_Pitch_r">
    <isDisabled> false </isDisabled>
    <!--Minimum allowed value for control signal. Used primarily when solving
        for control values-->
    <min_control> -infinity </min_control>
    <!--Maximum allowed value for control signal. Used primarily when solving
        for control values-->
    <max_control> infinity </max_control>
    <coordinate>  </coordinate>
    <optimal_force>     800.00000000000000000000 </optimal_force>
</CoordinateActuator>
```

Figure 6.4: Typical structure of CoordinateActuator section definition.

The next step was to define the actuators and elastic structures of our bipedal robot. For how regard the actuators we chose the *CoordinateActuator*, implemented in a built-in class of OpenSim that supports the application of a coordinate actuator to a model. This actuator has no states; the control is simply the force to be applied to the model (author Frank C. Anderson). You simply specify a minimum and maximum allowed values for control signal, used primarily when solving for control value, an associated coordinate and an optimal force value as illustrated in the code above.



```
<PointToPointSpring name="ext_dig_spring_r">
    <isDisabled> false </isDisabled>
    <body1> foot_r </body1>
    <body2> toe_r </body2>
    <point1>      0.000 0.007  0.002 </point1>
    <point2>      0.000 0.017 -0.014 </point2>
    <stiffness>   27.00 </stiffness>
    <rest_length> 0.029 </rest_length>
</PointToPointSpring>
```

Figure 6.5: Typical structure of PointToPointSpring section definition.

With regard to the elastic structure, in this fist stage we chose the *PointToPoint spring*.

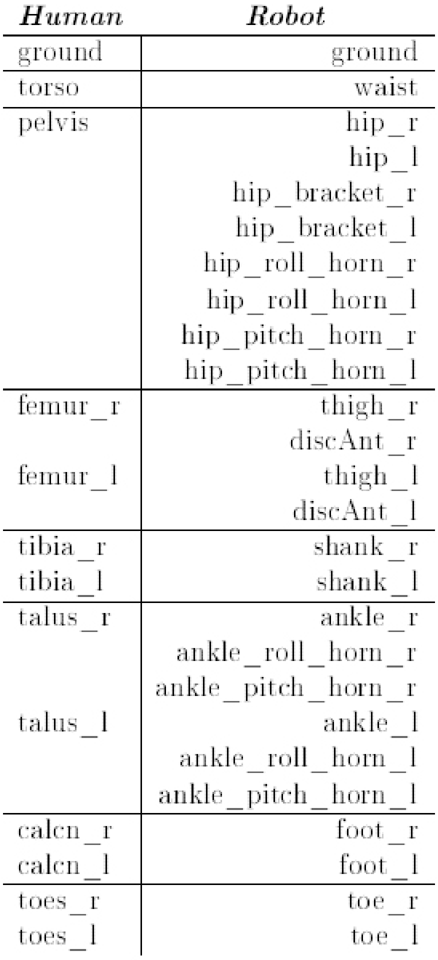
A simple point to point spring with a resting length and stiffness. Points are connected to bodies and are defined in the body frame. Also this is a built-in class of OpenSim (author Ajay Seth). The resting lengths come from the CAD model of the robot, while the stiffness coefficients was prescribed into a starting experimental range of 15-27 N/m. Some XML code example referred to the definition of a visible object for a bodies, to typical custom actuator and to the weld constraint can be found in Appendix C.

There is also another type of spring that we implemented following the documentation and tutorial provided within OpenSim, it is called ControllableSpring class. In OpenSim, when defining a new actuator, you can either start from scratch by deriving from the base class, CustomActuators, or if your actuator builds on an existing class, you can derive from that class. As presented in the documentation, the ControllableSpring class is derived from the Piston Actuator one. This is an exact copy of the class library called Line Actuator, in turn derived from Custom Actuator. Piston Actuator has been specially created for a tutorial, so to simplify this thing I directly derived Controllable Spring from Custom Actuator, implementing it as a plug-in. In this way we can load it into OpenSim and then use it through the GUI as well as in model simulations. In addition, you can view its properties from the GUI, as well as to consider it in motion analysis. All the documentation about the original version of ControllableSpring class can be found within the OpenSim package; here we briefly provide only an overview of the idea that is behind this custom actuator.

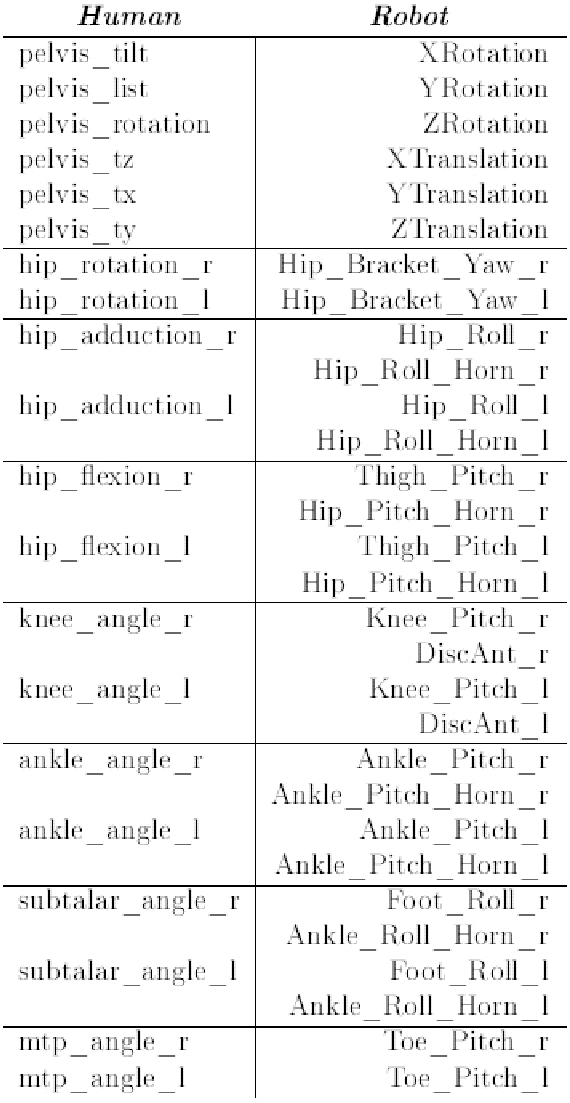
We define an actuator as something that produces controllable loads between two bodies. These could be torques applied between two bodies along a common axis, forces applied between two points defined on two different bodies, or some combination of loads applied according to some geometry and state parameters. The key function of any actuator class is to calculate and apply loads to its associated bodies based on its control value and the state variables any time step. The ControllableSpring actuator applies a force between two points fixed on two bodies. These bodies do not need to be consecutive bodies in a kinematic chain. This class calculates the magnitude of its stiffness coefficient k as the product (OptimalForce x ControlValue). Then it calculates the magnitude of its force F as the product k * (restLength - currentLength) and uses the convention that a positive force magnitude acts to increase the distance between points PA and PB fixed on two bodies. This actuator is not used in this work, but it will be introduce in the next version of the model for the characterization of some parts of our elastic structures, as a starting point for the development of a more accurate actuation of our bipedal robot. On this way, also a more detailed torque actuator will be introduce, inspired to the real robot servo actuators.

Mapping of body parts and joints from human to robot

As previously described, we can divide the musculoskeletal human model considered into five major parts (right and left): femur, tibia, talus, calcaneus, and toes, plus a torso and a pelvis. Similarly we can map the joint modeled in the human and the joint modeled in the robot. The major difference is on lumbar extension, bending, and rotation. Which are in the musculoskeletal human model, but not in the robot model because it doesn't still have an upper body part. It has only a waist that represents a torso with its physical properties, but that we have mapped into the pelvis for simplicity and clearness. The tables below illustrate this mapping; bodies on the left and joints on the right.



Human	*Robot*
ground	ground
torso	waist
pelvis	hip_r
	hip_l
	hip_bracket_r
	hip_bracket_l
	hip_roll_horn_r
	hip_roll_horn_l
	hip_pitch_horn_r
	hip_pitch_horn_l
femur_r	thigh_r
	discAnt_r
femur_l	thigh_l
	discAnt_l
tibia_r	shank_r
tibia_l	shank_l
talus_r	ankle_r
	ankle_roll_horn_r
	ankle_pitch_horn_r
talus_l	ankle_l
	ankle_roll_horn_l
	ankle_pitch_horn_l
calcn_r	foot_r
calcn_l	foot_l
toes_r	toe_r
toes_l	toe_l



Human	*Robot*
pelvis_tilt	XRotation
pelvis_list	YRotation
pelvis_rotation	ZRotation
pelvis_tz	XTranslation
pelvis_tx	YTranslation
pelvis_ty	ZTranslation
hip_rotation_r	Hip_Bracket_Yaw_r
hip_rotation_l	Hip_Bracket_Yaw_l
hip_adduction_r	Hip_Roll_r
	Hip_Roll_Horn_r
hip_adduction_l	Hip_Roll_l
	Hip_Roll_Horn_l
hip_flexion_r	Thigh_Pitch_r
	Hip_Pitch_Horn_r
hip_flexion_l	Thigh_Pitch_l
	Hip_Pitch_Horn_l
knee_angle_r	Knee_Pitch_r
	DiscAnt_r
knee_angle_l	Knee_Pitch_l
	DiscAnt_l
ankle_angle_r	Ankle_Pitch_r
	Ankle_Pitch_Horn_r
ankle_angle_l	Ankle_Pitch_l
	Ankle_Pitch_Horn_l
subtalar_angle_r	Foot_Roll_r
	Ankle_Roll_Horn_r
subtalar_angle_l	Foot_Roll_l
	Ankle_Roll_Horn_l
mtp_angle_r	Toe_Pitch_r
mtp_angle_l	Toe_Pitch_l

Mapping of human muscles into robot springs

Each leg in both model has ***7 DoF***. The musculoskeletal human model includes ***86 muscles of the lower limb***, 43 for each leg plus 6 muscles between pelvis and torso. Line segments approximate the muscle-tendon path from the origin to insertion and they are grouped in 12 compartments for each leg, based on their functions. For completeness the tables in Appendix D show the subdivision of human muscle which we have referred and the resultant robot spring groups (the last two table). We mapped these muscles into the springs in the robot. This mapping is principally based on the movement in which each muscle is involved.

- **ext_dig_spring** that corresponds to *flex_dig*, *flex_hal*, *ext_dig*, and *ext_hal* muscles. It is involved into the movement of flexion and extension of toes;

- **per_long_spring** that corresponds to *per_long* and *per_brev* human muscles. It is involved into the movement of flexion/extension, and inversion/eversion of the foot;

- **tib_post_A_spring**. It is a symmetric spring to *per_long_spring*;

- **per_tert_spring** maps the *per_tert* muscle;

- **tib_post_B_spring**. Similarly to *tib_post_A_spring*, this is symmetric to *per_tert_spring*;

- **tib_ant_spring** that take the function of *tib_ant* human muscle in the dorsiflexion/plantarflexion movement of the ankle/foot;

- **soleus_spring** related to the homonym human muscle;

- **gastrocnemius_spring**. It is primarily involved into the bending movement of the knee and maps two human muscles: *med_gas* and *lat_gas*;

- **bifem_spring**. Is is used into the movement of hip and knee. It also maps two muscles: *bifemlh* and *bifemsh*;

- **vas_med_spring**. It maps three human muscles: *vas_med*, *vas_int* and *vas_lat*;

- **rect_fem_spring** related to the homonym human muscle;

- **glut_max_spring**. Similarly to *vas_med_spring*, also in this case we have three corresponding human muscles: *glut_max1*, *glut_max2* and *glut_max3*;

- **psoas_spring** involved on the hip flexion/extension movement;

- **glut_med_A_spring** and **glut_med_B_spring**. They are the robotics counterparts of *glut_med1*, *glut_med2* and *glut_med3*;

- **glut_min_A_spring** and **glut_min_B_spring**. That stay for *glut_min1*, *glut_min2* and *glut_min3*;

- **add_long_spring** involved on the abduction/adduction movement of the lower limb;

- **add_brev_spring** that works coupled with the *add_long_spring.*

This is only a conceptual mapping, for many reasons: first of all, we apply extension springs to the robot, but in the musculotendon model the elasticity is similar to the compression spring and there is no fiber either. Moreover, for a movement in which the robot involves one or two springs, many times the human involves three, four or more muscles. The ankle dorsiflexion/plantarflexion movements, in which a human musculoskeletal model like that described here uses four and eight muscles respectively, the robotics counterpart uses five springs involved in both movements. Human uses eight muscle during knee bending where robot involves only four springs. The same is for the all other major articulation in the musculoskeletal model, as described in Appendix D.

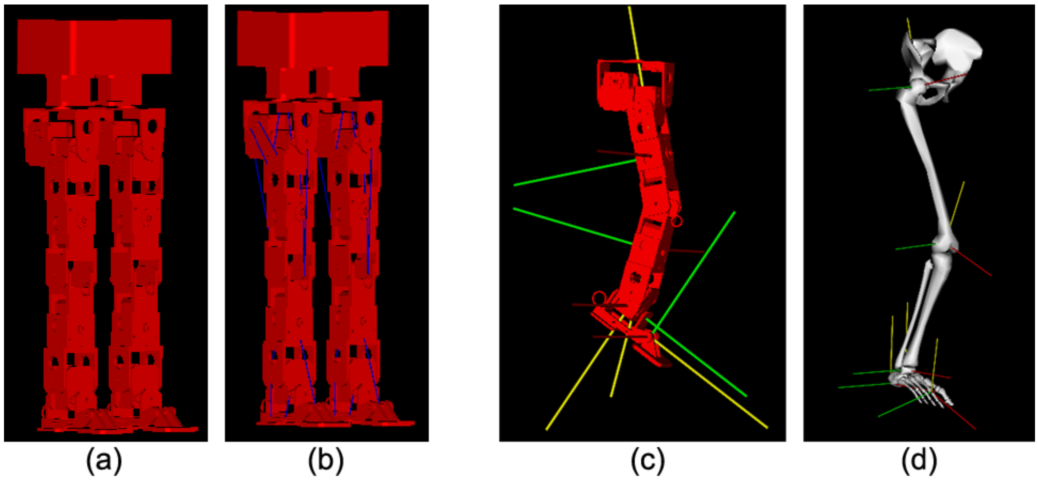
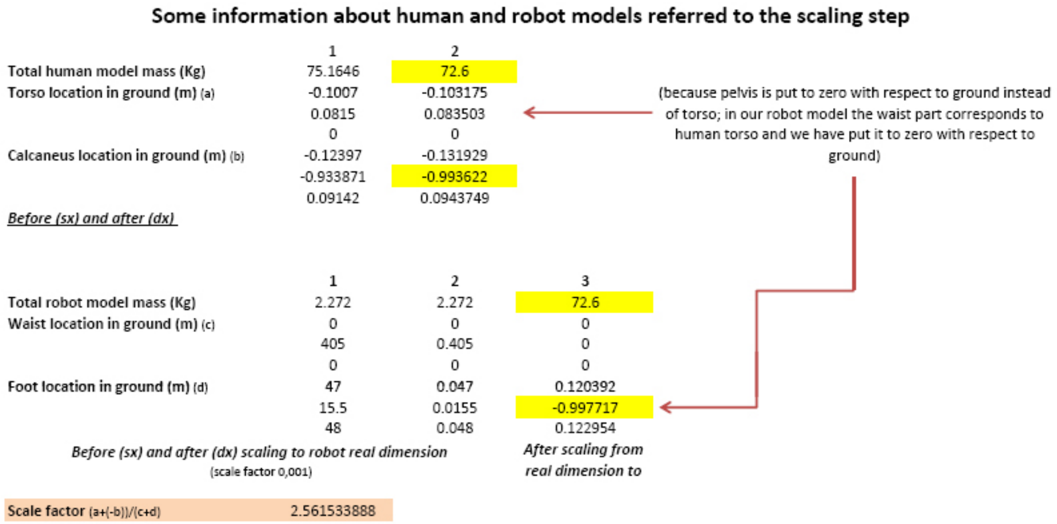


Figure 6.6: Robot models and system reference comparison. (a) Torque-driven model of robot; (b) Torque+Spring-driven model of the robot. (c)-(d) shows the system reference of the major bodies for robot and human, respectively.

6.2.1 Scaling

We started with a generic model of the robot where we have specified all the physical properties such as mass, inertia matrix and positions/relationships between the bodies through the definition of each joint. The scaling was then performed with the aim to obtain a model of the robot in real size, requiring the maintenance of specified mass distribution.

This tool was also used in the opposite direction: from the real robot, we got a version of the same at "*human dimension*". The scale factor was calculated by dividing the distance of the human pelvis from the ground with the distance of the waist of the robot from the ground, as shown in Fig. 6.7. The factor obtained is equal to: **2.561533888**.



Some information about human and robot models referred to the scaling step

	1	2
Total human model mass (Kg)	75.1646	72.6
Torso location in ground (m) (a)	-0.1007	-0.103175
	0.0815	0.083503
	0	0
Calcaneus location in ground (m) (b)	-0.12397	-0.131929
	-0.933871	-0.993622
	0.09142	0.0943749
Before (sx) and after (dx)		

(because pelvis is put to zero with respect to ground instead of torso; in our robot model the waist part corresponds to human torso and we have put it to zero with respect to ground)

	1	2	3
Total robot model mass (Kg)	2.272	2.272	72.6
Waist location in ground (m) (c)	0	0	0
	405	0.405	0
	0	0	0
Foot location in ground (m) (d)	47	0.047	0.120392
	15.5	0.0155	-0.997717
	48	0.048	0.122954
	Before (sx) and after (dx) scaling to robot real dimension (scale factor 0,001)		*After scaling from real dimension to*

Scale factor (a+(-b))/(c+d)	2.561533888

Figure 6.7: Scheme adopted for the calculation of the scaling factor.

In a similar way also another additional scale factor related to the length of the femur was calculated, comparing the length of the human femur with that mounted on the robot. In this case, the scale factor obtained is equal to: **1.3**. A model of this type makes the arrangement of the joints in the robot much closer to that in humans, however, would require a re-design of the original robot. For this reason and for our practical purposes we have not used this model in our simulations, but it may still provide a platform on which to make, at least for the moment, other theoretical studies.

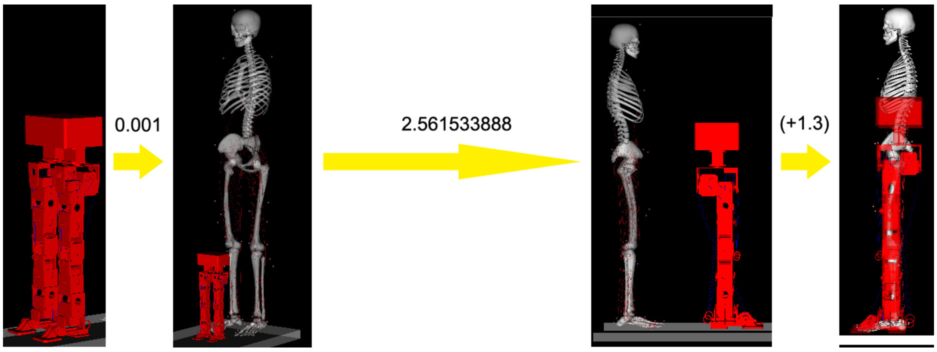


Figure 6.8: Snapshot of the scaling procedure.

6.2.2 Inverse Dynamics

As Described in Chapter 3, the Inverse Dynamics tool determines the generalized forces (eg, net forces and torques) at joint responsible for each given movement. Given the kinematics (eg, states or motion) describing the movement of a model and perhaps a portion of the kinetics (eg, external loads) applied to the model, these tools uses the data to perform an inverse dynamics analysis. This tool was initially used for dynamic simulations of movement (torque-driven simulation), from the files that drive a kinematic motion simulation of the robot. In a second step, in order to avoid the application of traditional methods of direct and inverse dynamics, we used another tool made available by OpenSim called *Computed Muscle Control*. We have already discussed the operation and application of this tool in the previous chapters. Simply, in this context was repeated the procedure previously described for the human being, but in the model of the robot.

6.3 Results and Comparisons

We conclude this chapter by showing the results of the first tests and doing some considerations on them. The results presented here refer in particular to four models: two models of the human and two of the robot. We have validated our data especially simulating and considering, among others, two movements in both human and robot, in their torque-driven version; a subsequent validation was obtained from the simulation of the human model Muscle-driven and robot Torque-driven with springs. Here, the musculoskeletal human model is assumed with a full activation of each muscle. Our goal was to understand if our robots would actually be able to support a complex movement such as **squat** or **walk**, and if it were possible establish some terms of comparison with the human being. In particular, while squat movement has been initially created by prescribing the kinematic and then deriveing the forces and moments that drive the dynamic simulation by CMC; in the case of walking gait we directly mapped the human kinematic for this movement into the robot kinematic associateing each human joint into the robotic counterpart and then executing CMC to drive the dynamic simulation. This phase was guided from the previously described mapping. On this way, we decided to analyze the simulations on several fronts, using the OpenSim built-in tools:

- analysis of the **Body Kinematics** that make up the models in order to obtain *positions* (center of mass position and orientation), *velocities* (linear and angular) and *accelerations* (linear and angular) of the centers of mass of each segment. Additionally, it records the overall center of mass of the model, as well as the velocity and acceleration of this center of mass. We shall see some of these data in detail referring to the femur, tibia and foot;

- analysis of **Joint Reactions** through which we can evaluate the *forces* and *moments* applied to a pair of segments connected by a coupling. It reports the joint reaction loads from a model. For a given joint, the reaction load is calculated as the forces and moments required to constrain the body motions to satisfy the joint as if the joint did not exist. The reaction load acts at the joint center (mobilizer frame) of both the parent and child bodies and either force can be reported and expressed in either the child, parent or ground frames. The default behavior is to express the force on the child in the ground frame;

- a comparison in terms of **Actuation**. Another built-in tool of OpenSim that records the *generalized force*, *speed*, and *power* developed by each actuator of the model. The generalized force can either be a force (with units N) or a torque (with units Nm). The actuator speed is the rate at which the actuator shortens. Depending on the actuator, a speed can be either a translational speed (m/s) or an angular speed (deg/s). An actuator power (Watts) is the rate at which an actuator does work. Positive work means that the actuator is delivering energy to the model; negative power means that the actuator is absorbing energy from the model;

- finally, some data relating to *forces developed by the springs* (**Force Reporter** analysis). The tests carried out have involved all the springs introduced in the model, but below we report only the results referring to the major ones: Bicep Femoris, Rectus Femoris, Vastus Medialis, Gastrocnemius, Soleus and Tibialis Anterior. This not only a matter of brevity, but especially they are the most commonly used because each of these crosses two joints and are involved in 3 types of motion as illustrated previously, except for Soleus and Vastus.

The Body Kinematics analysis allows us to make some assumptions about the results that will be explained below. In particular, the Fig. 6.6 (c - d) shows the reference systems of the main components of the lower limb. The different arrangement of these, together with the position and orientation of the centers of mass, the resulting distribution of mass and anthropometric characteristics of the human subject and robot do not allow us to directly compare the results except in terms of waveforms. For this reason, all torques have been scaled by the weight of the subject and the robot respectively. In this way we show a torque expressed in Nm/Kg, obtaining a more usefull comparison.

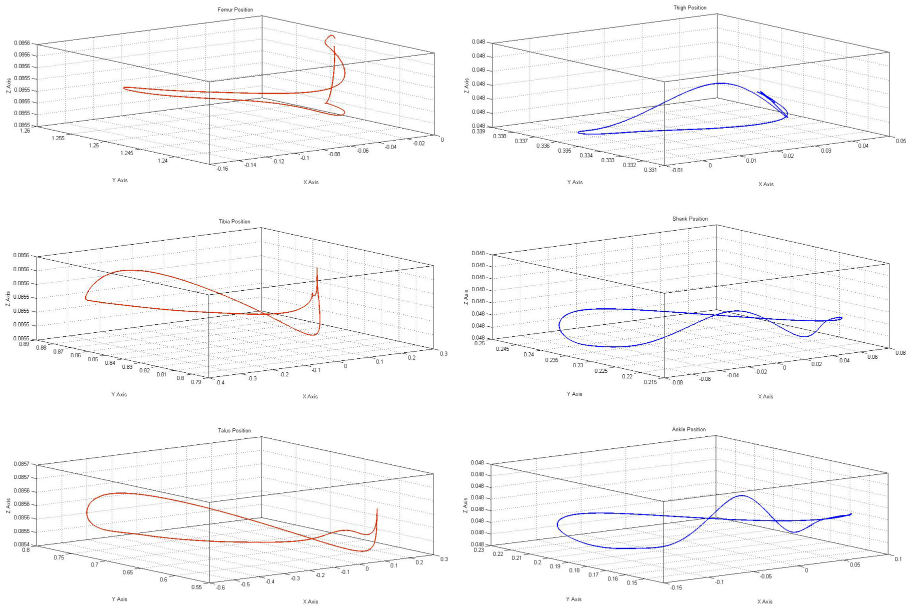
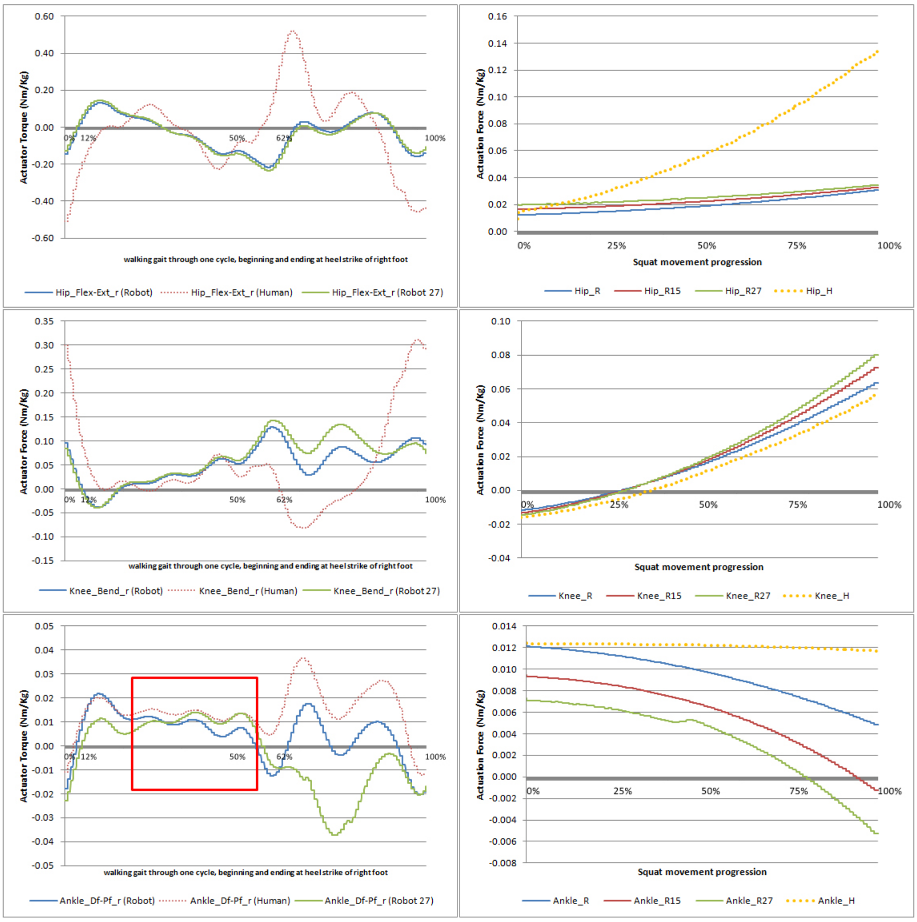


Figure 6.9: Positions of the centers of mass of the femur, tibia and ankle (top to bottom) in human (left) and robot (right).

As an illustration we show in this figure some graphs obtained from the analysis of the kinematics of the segments constituting the model. The figure shows the position of the center of mass of the femur, tibia and ankle (top to bottom, respectively) in human and robot (left to right, respectively) considered during the phase from heel-strike to next heel-strike of the right foot during a walking gait. To confirm the above discussion, we observe significant differences between the human model and the robots one, but in general there is also a consistency between the displacements of the centers of mass of the three segments within the same model.

It should be noted that all results presented are based on models constrained by a so called Weld Constraint placed between pelvis and ground. Accordingly, the effect induced by the ground reaction forces on the model is not considered at this stage of study, what constitutes a strong limitation on the one hand and a simplification on the other hand, because it does not allows to fully appreciate the compliant effects due to springs, but allows us to set an initial feasibility study of our project and its possible problems of modeling. Based on the studies made on human beings, we expect, on average, much higher values than those reported here, when we also consider the interaction of the robot with a flat or rough floor. Accordingly with what is shown about the kinematics, also the analysis of reaction forces at the joints indicate strong differences between different models. All the same, it is interesting to note that the forces and moments that bind a joint change with the introduction of elastic structures making the curves more regular. More detailed analysis on this front will be provided in the future.

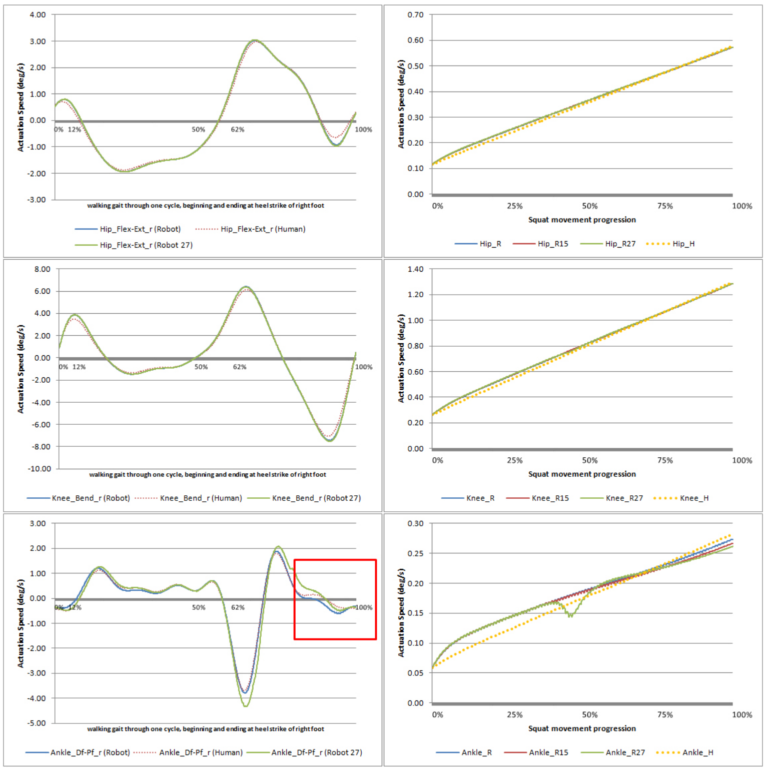
Regarding the analysis of the actuation, it is based on three parameters: force/strength, speed and power, which are shown below on the basis of the premises previously exposed. The forces developed by actuators placed in the hip and knee joints still show substantial differences from human model, particularly if we look at the squat movement. It is interesting to look at the graph of forces generated in the ankle joint. Especially during the stance phase (from heel strike (0%) to toe-off (62%) of the right foot) we can see (a - green line) the effect induced by the use of springs, which allow us to get a more similar behavior to a human being as highlighted with a red box in the picture below.



(a) (b)

Figure 6.10: Actuation Force of Hip, Knee and Ankle CoordinateActuator (top to bottom). (a) Referred to walking gait in human (red dotted line) and robot (blue and green line). The blue line is for the robot without springs, while the green line is for the robot with springs having a stiffness coefficient of 27 N/m; (b) Referred to squat movement in robot. The blue line is for the robot without springs, while both the red and green lines are referred to the robot with springs having a stiffness coefficient of 15N/m and 27N/m, respectively. The orange dotted line is referred to the human subject.

The same conclusion is valid for the squat movement (b-c), in which a preliminary analysis of the graphs shows immediately the utility in the use of elastic structures such as springs, in the knee joint design (green line) but only about 30% of the movement analyzed. This can be justified in part on the assumptions made in modeling, first of all to consider all the springs with the same stiffness coefficient and, secondly, the fact of having a weld constraint that bind the movement of the pelvis. In the future, a more detailed modeling of this part will give us more information about it. In general, ignoring the orders of magnitude of the plotted values, trend profiles and waveforms enable us to state that the robot as it is modeled is able to support the movement exhibiting a behavior comparable to that shown by the human model.



(a) (b)

Figure 6.11: Actuation Speed of Hip, Knee and Ankle CoordinateActuator (top to bottom). (a) Referred to walking gait in human (red dotted line) and robot (blue and green line). The blue line is for the robot without springs, while the green line is for the robot with springs having a stiffness coefficient of 27 N/m; (b) Referred to squat movement in robot and human. The blue line is for the robot without springs, while both the red and green lines are referred to the robot with springs having a stiffness coefficient of 15N/m and 27N/m, respectively; The orange dotted line is for the human.

In terms of speed, despite the differences observed previously generally we obtain good results. While hip and knee curves are almost identical it is important to note again the graph refers to the ankle. Throughout the stance phase the use of spring allows us to bridge the small gap that exists between humans and robots, as highlighted in the Fig. 6.11. The major differences persist during the transition from stance phase to swing one. The next analysis concerns the developed powers, consequently to the forces and speeds described above. In both knee and ankle joints, we obtain appreciable values during the stance phase; on the other hand the more differences remain on the hip and on the transition from one phase to another as highlighted in the graphs below. The most negative aspect on this front is given by the higher power consumption which is evident in the transition.

These tests allow us to conclude that already from this first studies of modeling, positive elements arising from the use of elastic structures in the design of humanoids robots. The main problems encountered in the design, implementation and control are on the hip joint, given also the complexity of the same in the human counterpart. In this

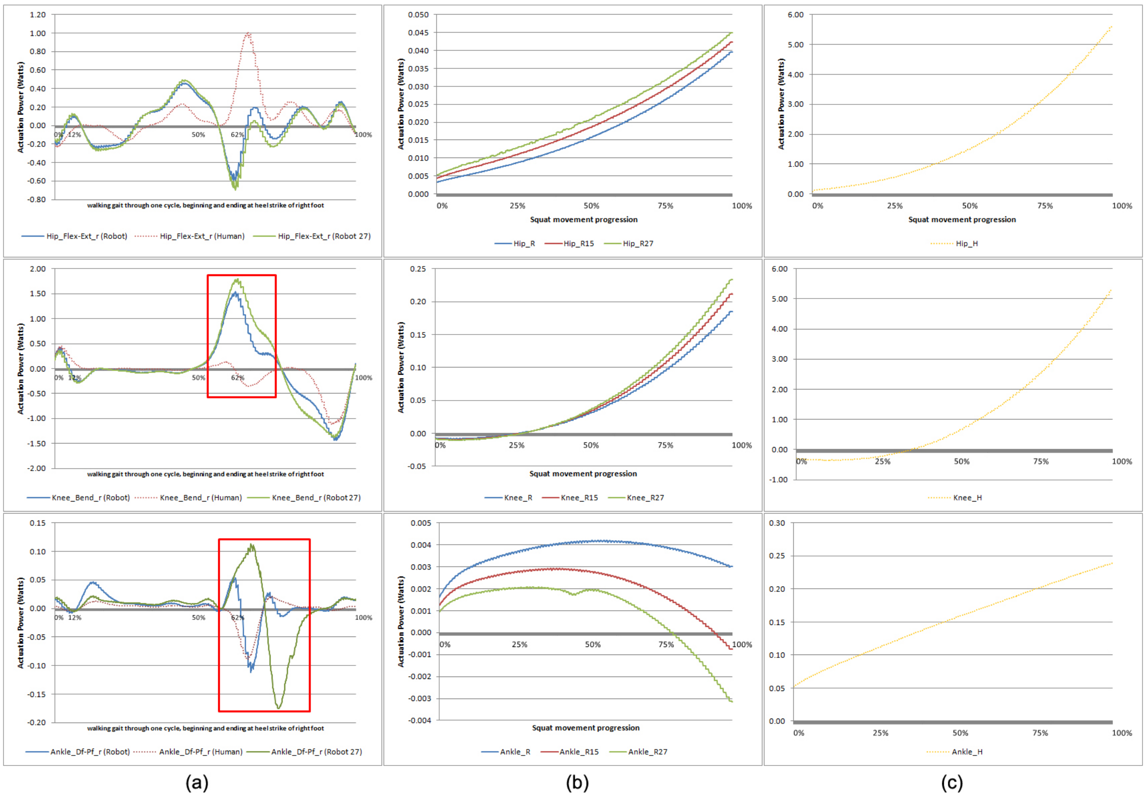


Figure 6.12: Actuation Power of Hip, Knee and Ankle CoordinateActuator (top to bottom). (a) Referred to walking gait in human (red dotted line) and robot (blue and green line). The blue line is for the robot without springs, while the green line is for the robot with springs having a stiffness coefficient of 27 N/m; (b) Referred to squat movement in robot. The blue line is for the robot without springs, while both the red and green lines are referred to the robot with springs having a stiffness coefficient of 15N/m and 27N/m, respectively; (c) Referred to squat movement in human.

sense, it should be noted again that all simulations were performed involving a robot configuration in which all the springs have the same constant stiffness: this is a strong idealization of the real robot. For this reason, we also implemented a version of the model in which the stiffness coefficients have been set on the basis of the ratio between Tendon Slack Length and Optimal Fiber Force, that characterizes the compliant behavior of each human muscle. The validation will be based on the results of the experiments that Fuben is leading on mechanical components making up the real platform. These results are not available at the time of this thesis.

Finally, we report some results about springs and forces that they develope. Such analysis involving, of course, all the springs introduced into the robot, but for simplicity and importance we choose to report here only the results refer to the 6 major springs of interest. Three are referred to the upper part of the lower limb: Bicep Femoris, Vastus and Rectus Femoris; the others are referred to the lower part of the lower limb: Gastrocnemius, Soleus and Tibialis Anterior. In light of the discussion so far, this analysis is based essentially only on the waveforms. In particular with regard to springs, the graphs plotted at the end of chapter provide the forces generated along different axes, considering the Z component as negligible compared to X and Y. With regard to the muscles, we considered only the passive force because the spring play a passive role in the implementation of this model. Future versions of the same, will also provide the use of springs based on the idea of active controllable spring, thus directly connected to a motor at one end. This will allow us to make comparisons on the level of active component of the force generated by human muscle.

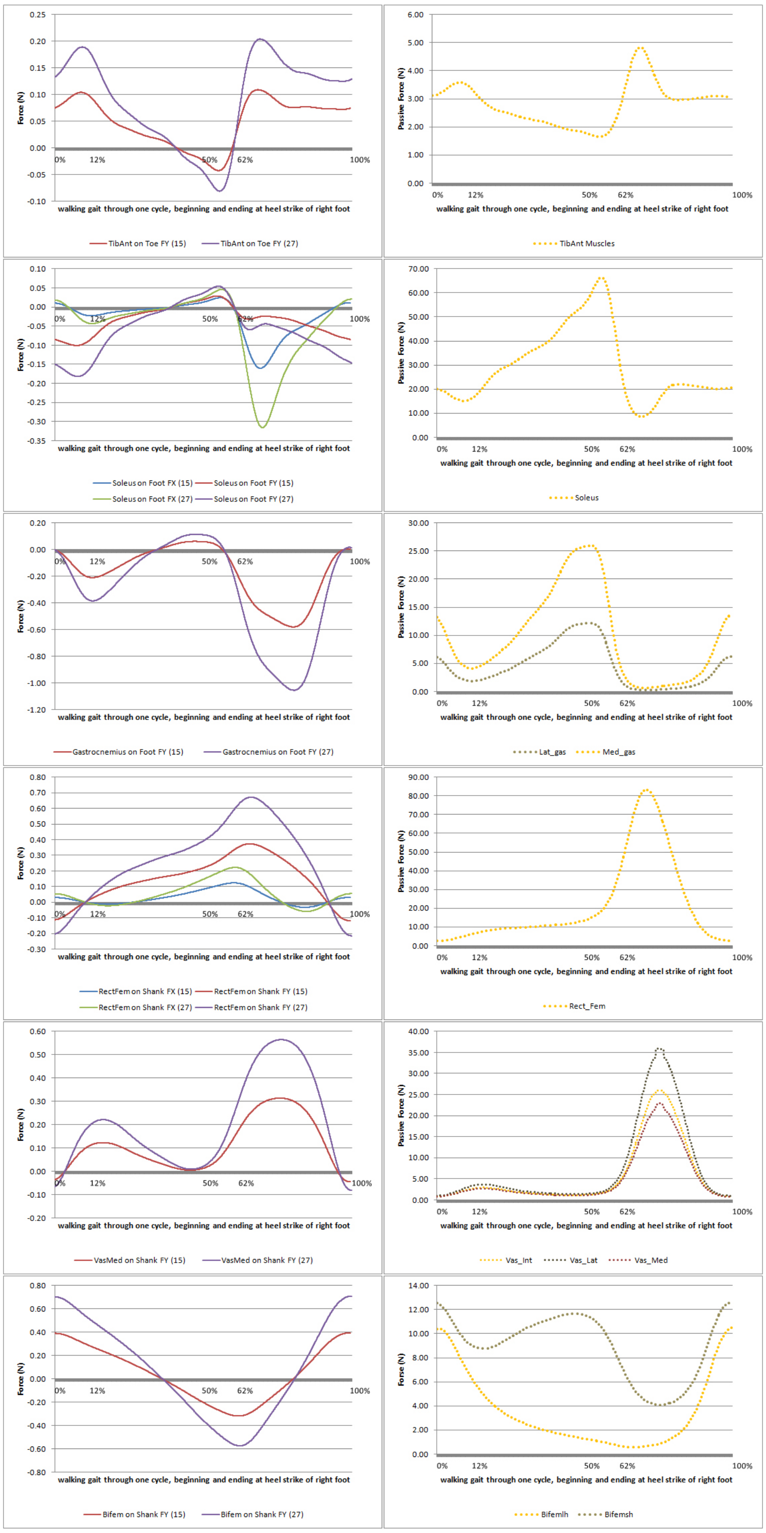
If we consider the fact that the springs used in the robot are only extensible springs, therefore, based on a completely opposite to that which governs the muscles that contract during their activation, in general we can observe waveforms that approximate well the curves that describe the passive force of human muscle during a cyclic movement such as walk. The analysis that was done in this sense does not consider the orders of magnitude of values for the same reasons already given about the study of the kinematics. However, we can state that the values referred to the force expressed in Newtons, if transformed into the equivalent ones of weight force are consistent with the physical properties of the model both in the case of the robot and human. However, we reiterate the fact that with the introduction of the contact we expect significant increases in this direction, precisely due to the action of ground reaction forces. Future developments of this work will provide answers on this.

6.4 Conclusion and Future Work

For what is in our knowledge this is the first elastic robot to be modeled and simulated OpenSim: a software platform originally designed for the study of human movement. As described, these first results demonstrate the potential brought by such a tool also used to study the movement of a robot, especially a biologically-inspired robots. Starting from neuromusculoskeletal models in the literature we were able to carry out this study with results that make us hope for the continuation of our research, convinced that the introduction of elastic-components is undoubtedly the future for the development and design of humanoid robot like this.

Our goal is to improve the current model, refining the essential components such as actuators and springs with the intent to obtain more information and new solutions applicable in the robotic field but also in other areas such as human rehabilitation with the study of what is the ultimate expression of the interaction between man and machine: the exoskeletons. A further validation of the results we have presented will be possible with the advent of the real robot and therefore their use in control. Moreover, in the light of studies that the scientific community is making about the creation of so-called "artificial muscles" this work lays the foundation for the development of another theoretical model of the same robot in which the actuators are muscles with properties quite comparable to those of human muscles. We are already working on this front.

For these reasons we believe that it is important to understand more in detail what governs human movement, and especially how to apply these laws to design and construction of new generation of humanoid robots able to move and interact with human beings in a reliable, safe and efficient way independently from the environment in which they are.



(a) (b)

Figure 6.13: A comparison between forces generated from springs with passive forces referred to muscles. (a) Humaoids Passive Spring Forces components; (b) Human Passive Muscle Forces.

Chapter 7

Conclusions

This work presents the design and development of a novel elastic bipedal robot. The objective was to advance the research on modeling and simulation of humanoid robots based on musculoskeletal human model. This because the new-generation of humanoids robots will have highly complex skeletal structures and actuator systems that increasingly resemble the human musculoskeletal system. So, if on one side the neuromusculoskeletal (NMS) models provide a tool for studying the human muscular activation, and making EMG-driven Human-Machine Interface (HMI) to control active orthosis; on the other side, the understanding of how muscles are activated to actuate the human body will directly allow designing motion and balance controller to move humanoids in a more sophisticated way. Potentially they provide also a tool to create humanoid robot with artificial muscles, which reproduces the same activation dynamics of a human.

The obtained results show an improvement in humanoid robots design with the introduction of elastic cable and springs into the actuation structure, but also show the possibility to map a real human movement into a human-inspired robot. The introduction of monoarticular and biarticular joints appropriately based on bionics and ergonomics shows benefits on making the motions of robot more flexible and more human like. This model also aims to obtain solutions that can be applied to the exoskeleton development to improve their capabilities and the way in which they interface with the human in a more flexible way. Among others, for these reasons NMS modeling offers great solutions for exoskeletons control and humanoids actuation, but can also improve the realization of more realistic virtual humans by providing a more accurate estimation of human and robot internal state.

This research aims to put the basis for the development of a real-time control structure that include human and robot in a same loop in a very similar way to how described in "*Man in the loop*" [55] where a human subject is placed in a loop within which the NMS model evaluates motion data and biological signals generated by the subject as they move. Evalutations are used by the NMS model to predict what force the subject's muscles are able to generate. These are then further evaluated to calculate the amount of support the assistive device will have to provide the user with, in order to allow for the proper execution of the desired movement. The assistive device will then actively support the subject by providing a force feedback. In our case we will have a bipedal robot instead of an assistive device like a powered orthosis and the NMS model evaluations will be used to calculate what force the robot will have to generate, in order to allow the execution of a desired movement. The robot device will then actively interact with the subject by providing a force feedback. A such technology can be applied for example in extremely dangerous environment avoiding to expose the human to risks; it can be exploited to deepen our knowledge about the interaction between robot and human and so for the development and use of exoskeletons and active orthosis.

Humanoid robots can already autonomously perform task decomposition necessary to carry out high-level, complex commands given through gesture and speech. Humanoids can adapt and orchestrate existing capabilities as well as create new behaviors using a variety of machine learning techniques. Humanoids may prove to be the ideal robot design to interact with people. After all, humans tend to naturally interact with other human-like entities; the interface is hardwire in our brains. Furthermore, while the industrial manipulator and mobile robots needs to adapt and drastically change their environment, the humanoid robots can work directly in the same human environment without any modification. Historically, we humans have adapted to the highly constrained modality of monitor and keyboard. In the future, technology will adapt to us. Humanoids probably will change the way we interact with machines and will impact how we interact with and understand each other.

7.1 Summary

Chapter 1 defines the research problem investigated within this Thesis. It gives an overview of the research conducted in the field of Humanoid Robots, it highlights the novelty and the significance of the proposed research, and the main limitations of the study conducted.

Chapter 2 presents a survey about the research conducted in the field of Human Movement Modeling with particular attention to the major NMS model and to the interface between the Central Nervous System (CNS) and the articulated body segments. From the biomechanics of walking and the locomotor apparatus to the functional tasks and phases of gait through the temporal parameters and determinants of gait in a human subject. On the other hand, it provides an overview of the most important approach to Robot Movement Modeling. Starting from some basic concept about legged and bipedal locomotion through the Inverted Pendulum Model and Dynamical Walking Model it provides the state-of-the-art about Elastic Bipedal Locomotion, which is the approach that has been driven the development of this bipedal robot.

Chapter 3 presents an introduction to the OpenSim: the open-source platform for modeling, simulating, and analyzing the neuro-musculoskeletal system that we have used in our research work. It gives a detailed description of the functioning of the major tools that we used in our simulations. The Scale Tool that alters the anthropometry of a model so that it matches a particular subject as closely as possible. The Inverse Dynamics toot that determines the generalized forces at each joint responsible for a given movement. The Residual Reduction Analysis Tool that minimizes the effects of modeling and marker data processing errors that aggregate and lead to large nonphysical compensatory forces called residuals. The Computed Muscle Control (CMC) Tool that computes a set of muscle excitations (or more generally actuator controls) that will drive a dynamic musculoskeletal model to track a set of desired kinematics in the presence of applied external forces. The Forward Dynamics Tool that drives a forward dynamic simulation based on the controls computed by the CMC. And last but not least the Analyze Toot, with particular attention to actuation and joint reaction analysis, that enables us to analyze a model or simulation based on a number of inputs that can include time histories of model states, controls, and external loads applied to the model.

Chapter 4 and 5 presents a novel elastic bipedal robot design and an expected control system design based on the studies conducted by Fuben He. Starting from the previously developed robots *BioRob arm*, *Jena Walker* in his versions I and II, and finally *BioBiped R1*, a structure element detailed description is provided. It gives an overview about three different types of springs: compression springs, extension springs, and torsion springs. An introduction to Series Elastic Actuator that are considered in our robot design for the soleus and vastus muscle groups in order to feed the ankle and the knee joint with energy, and so to support them during bending with motor power. A briefly description of the RX-28 Dynamixel Robot Servo Actuator and the AS5040 contactless magnetic rotary encoder, which are the servo motors and the encoders chosen for this bipedal robot. After a detailed description of each degrees of freedom in relation to its human counterpart, chapter 4 finishes with a comparison between monoarticular and biarticular joints which are the basic idea on making the motions of robot more flexible and more human-like. Chapter 5 introduces the Expected Control Structure, starting from a comparison of different elastic actuation models like Original SEA Model end Extended SEA Model. It ends with the proposed control structure that basically obey the original SEA feedforward control theory and in which a PID controller is also introduced, the sensor will provide data for the controller as feedback compensate.

Chapter 6 describe the modeling and simulation phases into the OpenSim platform. From the studies conducted on the neuromusculoskeletal human models to the implementation and dynamic simulation of our robot into OpenSim. A detailed description of the mapping between human and robot is provided, with assuption done for the modeling phase. A discussion of some first results is made at the end of the chapter by showing several graphs of forces, speeds and power developed by the actuators and springs considered in our studies.

7.2 Limitations and Future Work

This work was conducted only on theoretical model, without a real robot platform. We will validate our results only in the future when we will have also the real bipedal robot. We apply the extension springs to the robot, but in the musculotendon model the elasticity is similar to the compression spring. There is no fiber either. Moreover, it should be noted again that all simulations were performed involving a robot configuration in which all the springs have the same constant stiffness: this is a strong idealization of the real robot. For this reason, we also implemented a version of the model in which the stiffness coefficients have been set on the basis of the ratio between Tendon Slack Length and Optimal Fiber Force, that characterizes the compliant behavior of each human muscle. The validation will be based on the results of the experiments that Fuben is leading on mechanical components making up the real platform. These results are not available at the time of this thesis. On the other hand, we have used only theoretical coordinate actuator without model a real motor actuator. This gives us the possibility to study the robot behavior during a prescribed movement, but it doesn't take in account specific properties of any real motors. Another limitation gives from the weld constraint introduced between waist and ground and the consequent lack of a contact model between foot/toe and ground. This is a condition very similar to Jena Walker II, that is attached at the trunk to a lateral guide. Accordingly, the effect induced by the ground reaction forces on the model is not considered at this stage of study, what constitutes a strong limitation on the one hand and a simplification on the other hand, because it does not allows to fully appreciate the compliant effects due to springs, but allows us to set an initial feasibility study of our project and its possible problems of modeling.

The human musculoskeletal model used in this research is the same model formed the starting point for the researches led by Massimo Sartori [55]. This is important for us because our next goal is to introduce a contact model between foot/toe and ground based on the results obtained in that research. In this way, we can remove the weld constraint between waist and ground and so introduce also the ground reaction forces in our simulations. We aim to improve the movement of the toe with respect to foot so that to characterize with more precision the behavior of toe-foot-ankle parts during the walking gait. A such types of improvements in our robot modeling give us the possibility to effectively compare our results with those referred to a human subject or to the BioBiped R1 former robot. We want also to introduce the controllable spring actuators described previously for modeling the SEAs and then modified them with reference to the evolution of the real mechanical experiments that Fuben is conducting at Dalian University. The use of springs based on the idea of active controllable spring, thus directly connected to a motor at one end, will allow us to make comparisons on the level of active component of the force generated by human muscle.

This thesis is only the starting point of a wide range of other possible future works: from the control structure completion and whole-body control application, to imitation learning and reinforcement learning for human locomotion, from motion test on flat ground to motion test on rough ground, and obviously the transition from simulation to practice with a real elastic bipedal robot biologically-inspired that can move like a human being.

Appendix A - Model Parts

Part: Toe

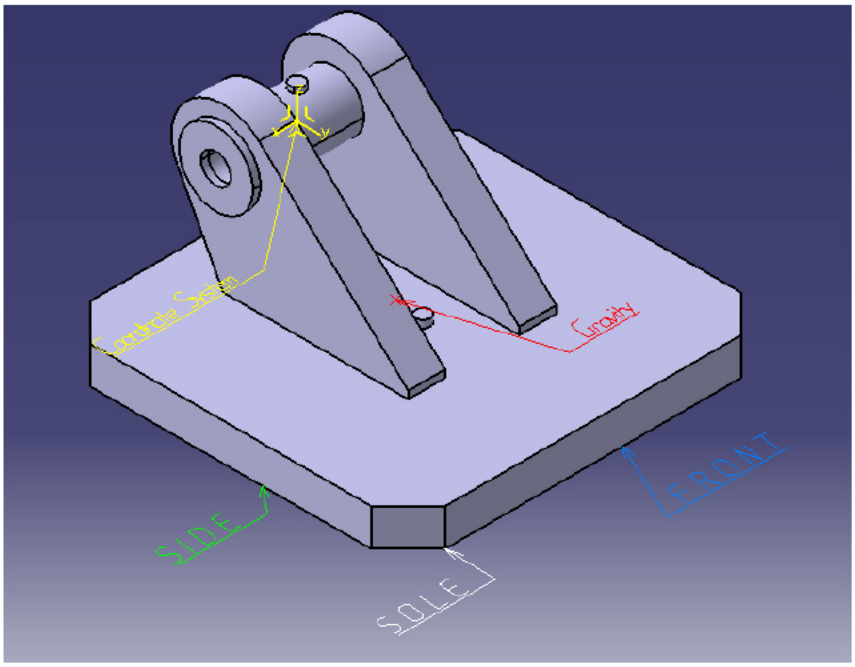
Volume: 1.626e-005 m3
Weight: 0.044 kg

Gravity:
X = -2.349e-015 mm ≈ 0 mm
Y = 13.102 mm
Z = -14.113 mm

Inertial Matrix:

Ixx = 9.308e-006 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 8.891e-006 kgxm2;	Iyz = 1.424e-006 kgxm2;
Izx = 0 kgxm2;	Izy = 1.424e-006 kgxm2;	Izz = 1.459e-005 kgxm2;

Coordinate System:
Px = 25 mm (from x-axis to side)
Py = 40 mm (from y-axis to front)
Pz = 20 mm (from z-axis to sole)



Part: Foot

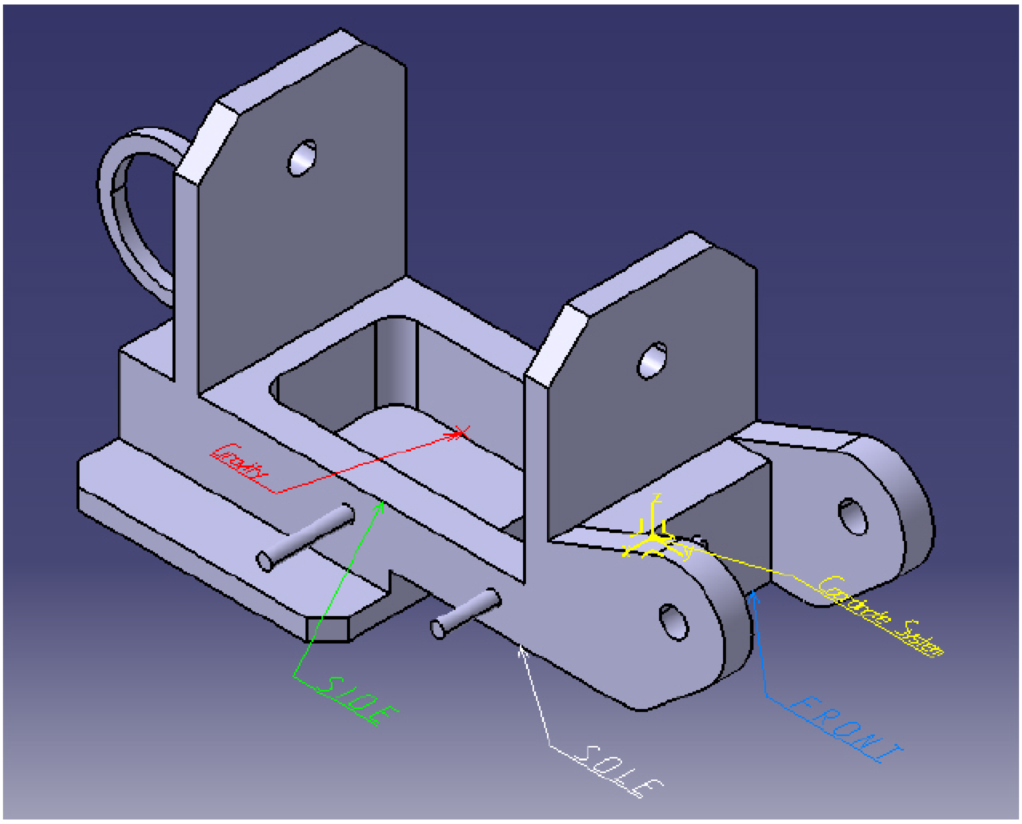
Volume: 2.811e-005 m3
Weight: 0.076 kg

Gravity:
X = -1.359e-015 mm ≈ 0 mm
Y = -27.474 mm
Z = -0.641 mm

Inertial Matrix:

Ixx = 5.967e-005 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 1.827e-005 kgxm2;	Iyz = -5.37e-006 kgxm2;
Izx = 0 kgxm2;	Izy = -5.37e-006 kgxm2;	Izz = 5.937e-005 kgxm2;

Coordinate System:
Px = 15 mm (from x-axis to side)
Py = 6 mm (from y-axis to front)
Pz = 8 mm (from z-axis to sole)



Part: Ankle

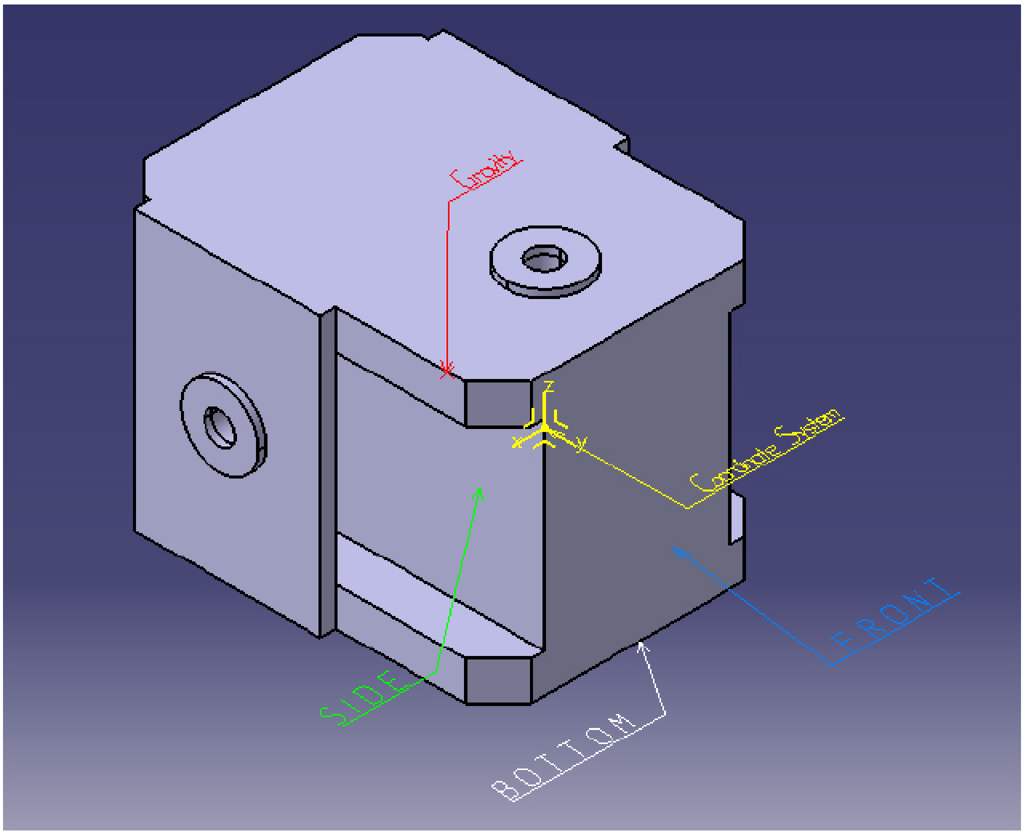
Volume: 3.279e-005 m3
Weight: 0.089 kg

Gravity:
X = -3.021e-015 mm ≈ 0 mm
Y = -12.669 mm
Z = -0.102 mm

Inertial Matrix:

Ixx = 3.245e-005 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 2.575e-005 kgxm2;	Iyz = 1.147e-007 kgxm2;
Izx = 0 kgxm2;	Izy = 1.147e-007 kgxm2;	Izz = 2.906e-005 kgxm2;

Coordinate System:
Px = 12 mm (from x-axis to side)
Py = 12 mm (from y-axis to front)
Pz = 18 mm (from z-axis to bottom)



Part: Ankle_Roll_Horn

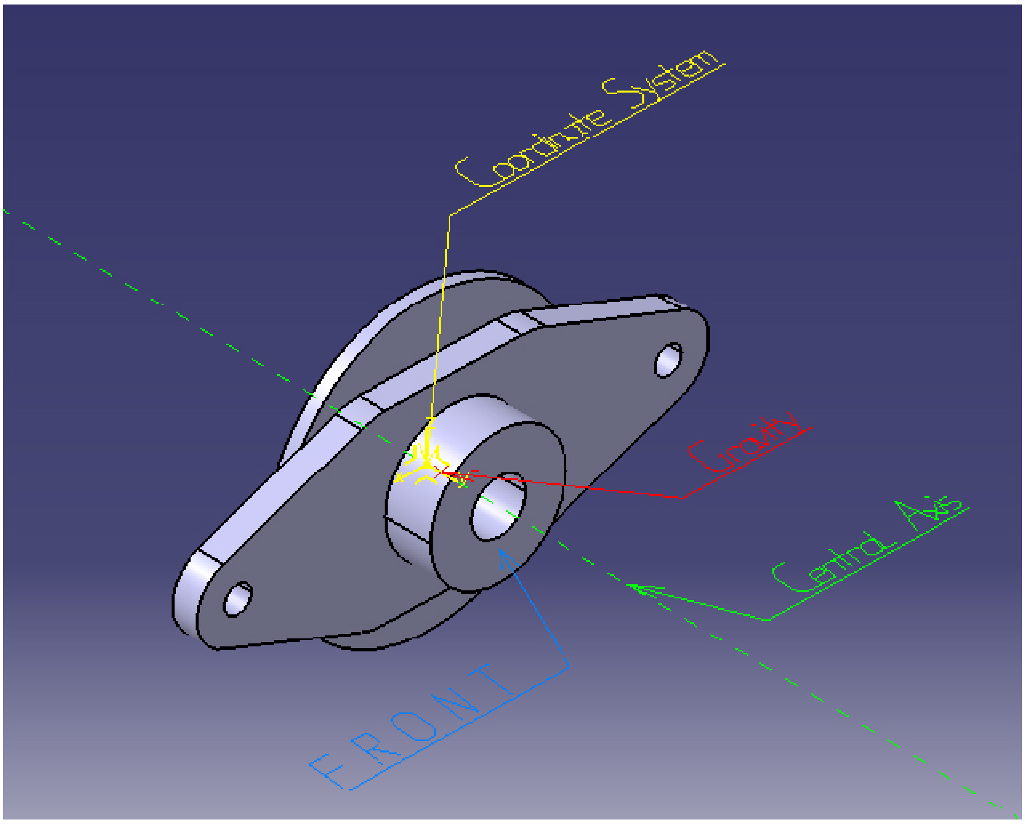
Volume: 1.362e-006 m3
Weight: 0.004 kg

Gravity:
X = -5.51e-015 mm ≈ 0 mm
Y = 1.061 mm
Z = 1.059e-015 mm ≈ 0 mm

Inertial Matrix:

Ixx = 7.008e-008 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 2.738e-007 kgxm2;	Iyz = 0 kgxm2;
Izx = 0 kgxm2;	Izy = 0 kgxm2;	Izz = 2.204e-007 kgxm2;

Coordinate System:
Px = 0 mm (in the central axis)
Py = 5.5 mm (from y-axis to front)
Pz = 0 mm (in the central axis)



Part: Ankle_Pitch_Horn

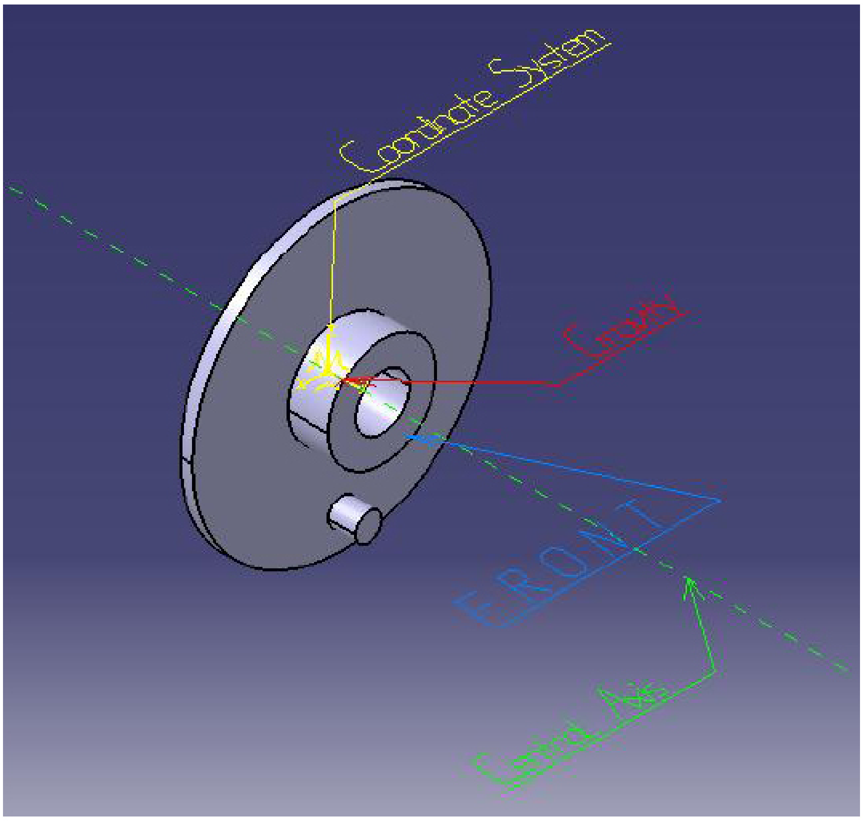
Volume: 4.869e-007 m3
Weight: 0.001 kg

Gravity:
X = 8.584e-016 mm ≈ 0 mm
Y = 0.984 mm
Z = -0.11 mm

Inertial Matrix:

Ixx = 3.515e-008 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 6.654e-008 kgxm2;	Iyz = 1.471e-010 kgxm2;
Izx = 0 kgxm2;	Izy = 1.471e-010 kgxm2;	Izz = 3.394e-008 kgxm2;

Coordinate System:
Px = 0 mm (in the central axis)
Py = 4 mm (from y-axis to front)
Pz = 0 mm (in the central axis)



Part: Shank

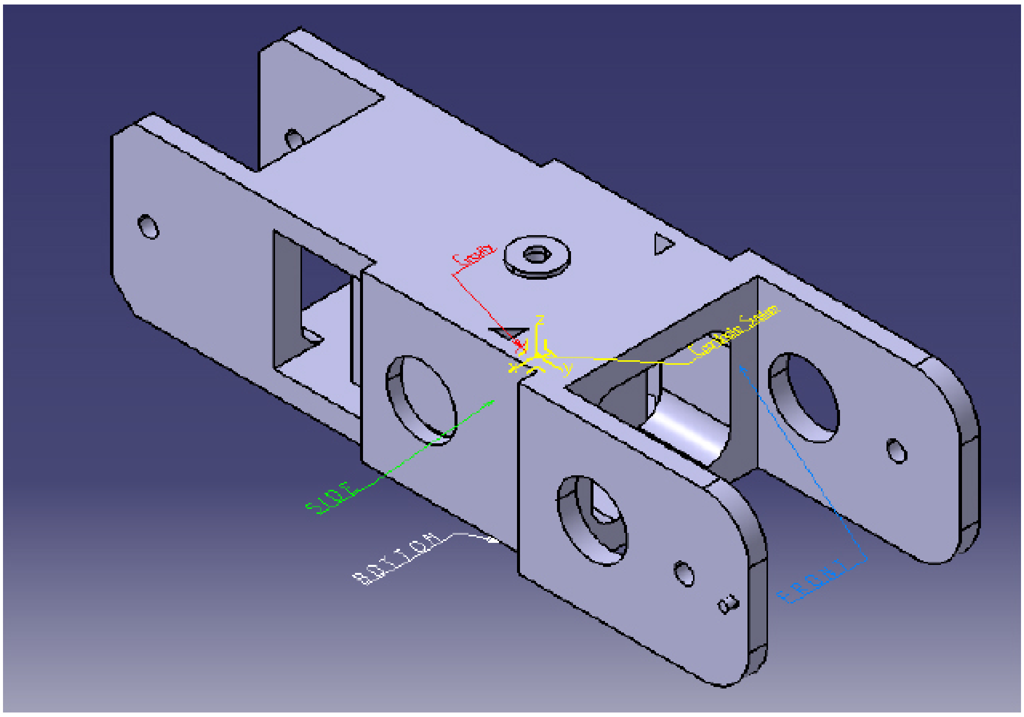
Volume: 7.081e-005 m3
Weight: 0.192 kg

Gravity:
X = 1.503e-015mm ≈ 0 mm
Y = -1.5 mm
Z = -0.112 mm

Inertial Matrix:

Ixx = 2.411e-004 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 7.452e-005 kgxm2;	Iyz = -5.067e-007 kgxm2;
Izx = 0 kgxm2;	Izy = -5.067e-007 kgxm2;	Izz = 2.571e-004 kgxm2;

Coordinate System:
Px = 21 mm (from x-axis to side)
Py = 27 mm (from y-axis to front)
Pz = 18 mm (from z-axis to bottom)



Part: Thigh

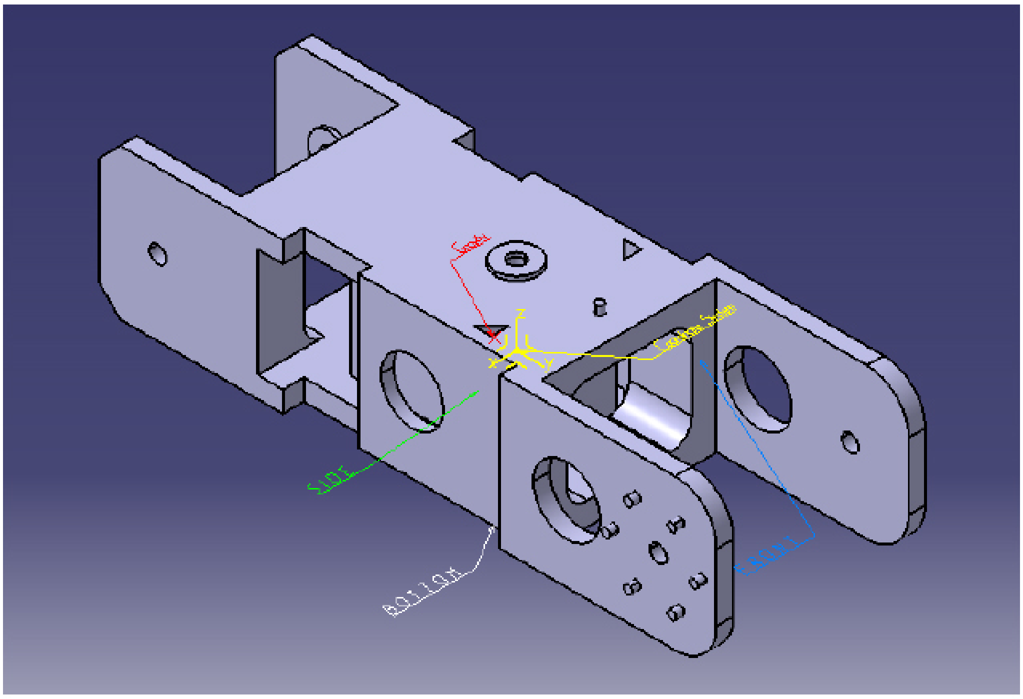
Volume: 7.661e-005 m3
Weight: 0.208 kg

Gravity:
X = -1.227e-015mm ≈ 0 mm
Y = -5.368 mm
Z = -0.044 mm

Inertial Matrix:

Ixx = 2.88e-004 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 8.612e-005 kgxm2;	Iyz = 4.858e-008 kgxm2;
Izx = 0 kgxm2;	Izy = 4.858e-008 kgxm2;	Izz = 3.124e-004 kgxm2;

Coordinate System:
Px = 21 mm (from x-axis to side)
Py = 27 mm (from y-axis to front)
Pz = 18 mm (from z-axis to bottom)



Part: Hip

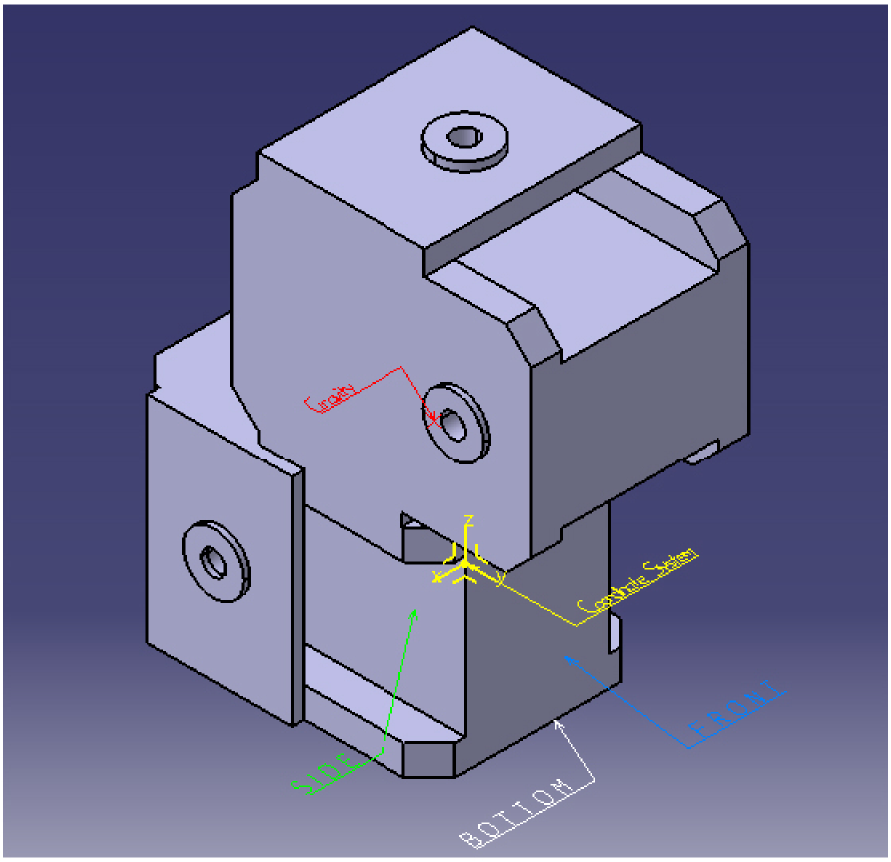
Volume: 5.89e-005 m3
Weight: 0.16 kg

Gravity:
X = 4.942e-016 mm ≈ 0 mm
Y = -5.003 mm
Z = 18.208 mm

Inertial Matrix:

Ixx = 1.361e-004 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 1.146e-004 kgxm2;	Iyz = -2.648e-005 kgxm2;
Izx = 0 kgxm2;	Izy = -2.648e-005 kgxm2;	Izz = 6.896e-005 kgxm2;

Coordinate System:
Px = 12 mm (from x-axis to side)
Py = 12 mm (from y-axis to front)
Pz = 18 mm (from z-axis to bottom)



Part: Hip_Roll_Horn

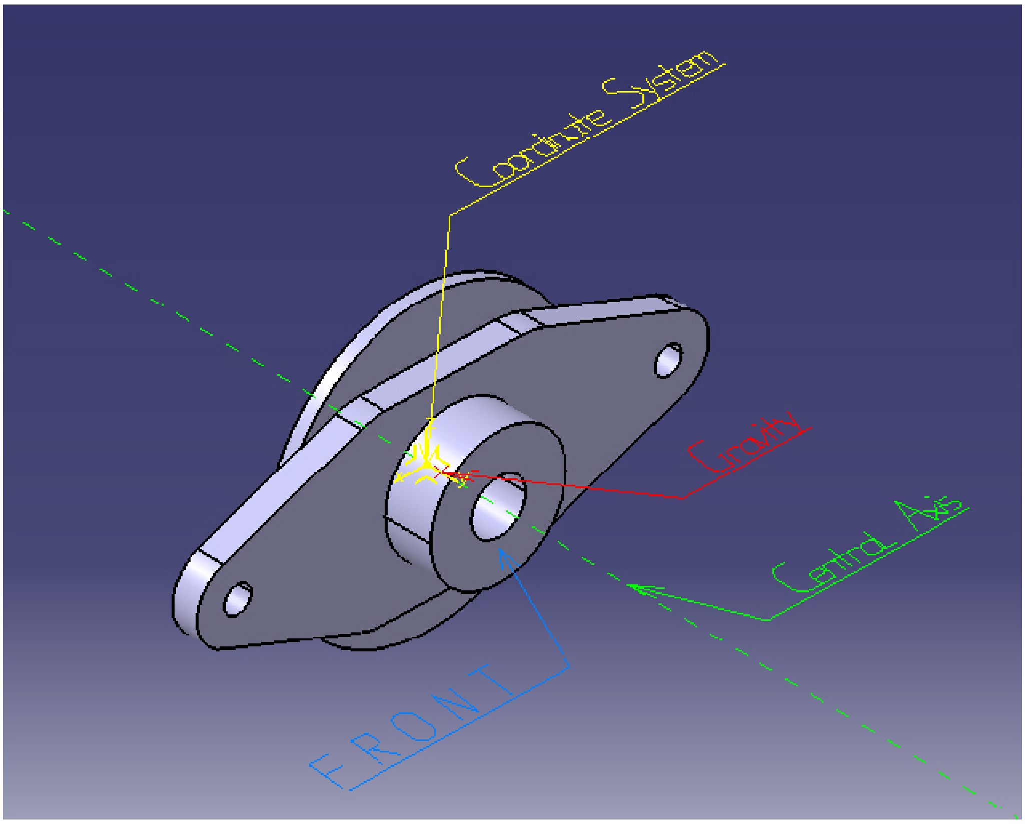
Volume: 1.362e-006 m3
Weight: 0.004 kg

Gravity:
X = -5.51e-015 mm ≈ 0 mm
Y = 1.061 mm
Z = 1.059e-015 mm ≈ 0 mm

Inertial Matrix:

Ixx = 7.008e-008 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 2.738e-007 kgxm2;	Iyz = 0 kgxm2;
Izx = 0 kgxm2;	Izy = 0 kgxm2;	Izz = 2.204e-007 kgxm2;

Coordinate System:
Px = 0 mm (in the central axis)
Py = 5.5 mm (from y-axis to front)
Pz = 0 mm (in the central axis)



Part: Hip_Pitch_Horn

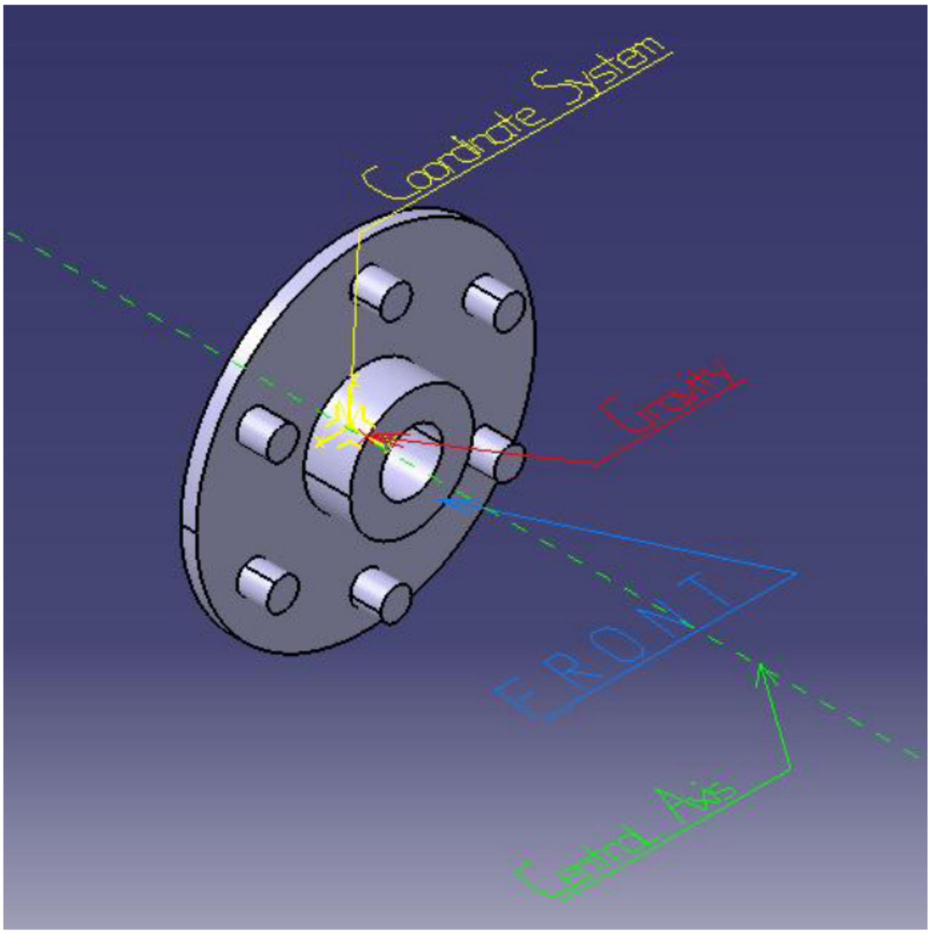
Volume: 5.184e-007 m3
Weight: 0.001 kg

Gravity:
X = 7.745e-016 mm ≈ 0 mm
Y = 1.045 mm
Z = 7.95e-016 mm ≈ 0 mm

Inertial Matrix:

Ixx = 3.776e-008 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 7.275e-008 kgxm2;	Iyz = 0 kgxm2;
Izx = 0 kgxm2;	Izy = 0 kgxm2;	Izz = 3.776e-008 kgxm2;

Coordinate System:
Px = 0 mm (in the central axis)
Py = 4 mm (from y-axis to front)
Pz = 0 mm (in the central axis)



Part: Hip_Bracket

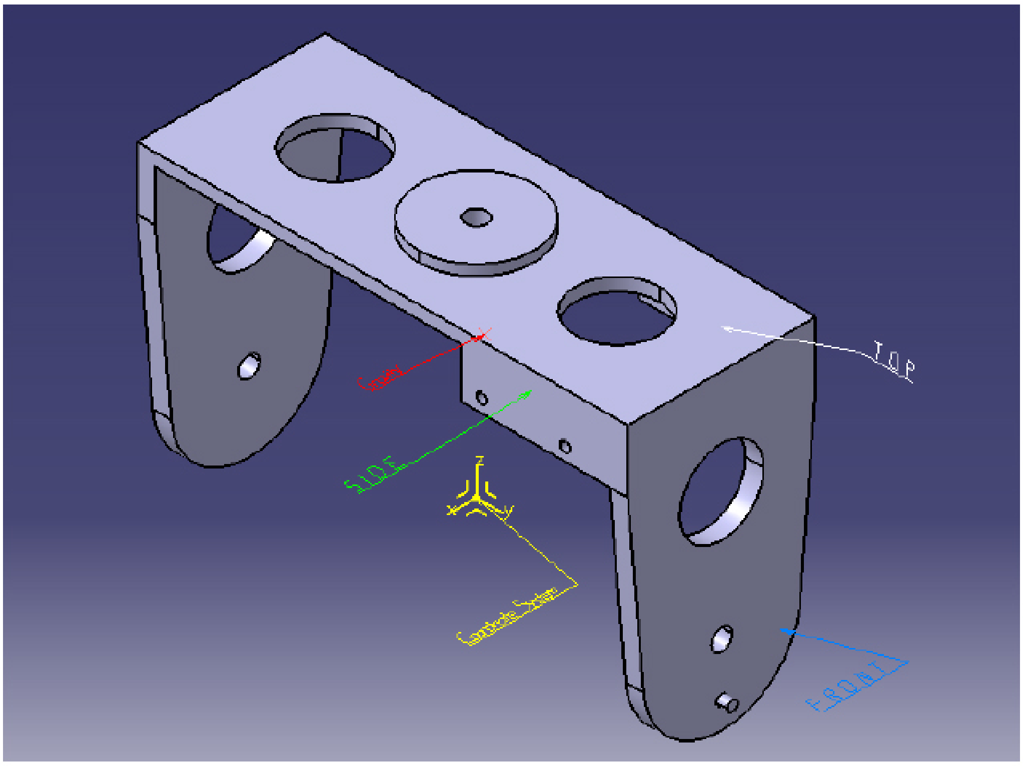
Volume: 2.012e-005 m3
Weight: 0.055 kg

Gravity:
X = -3.604e-016 mm ≈ 0 mm
Y = 1.54 mm
Z = 27.76 mm

Inertial Matrix:

Ixx = 9.633e-005 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 2.454e-005 kgxm2;	Iyz = -8.013e-007 kgxm2;
Izx = 0 kgxm2;	Izy = -8.013e-007 kgxm2;	Izz = 8.421e-005 kgxm2;

Coordinate System:
Px = 18 mm (from x-axis to side)
Py = 47 mm (from y-axis to front)
Pz = 44.5 mm (from z-axis to top)



Part: Waist

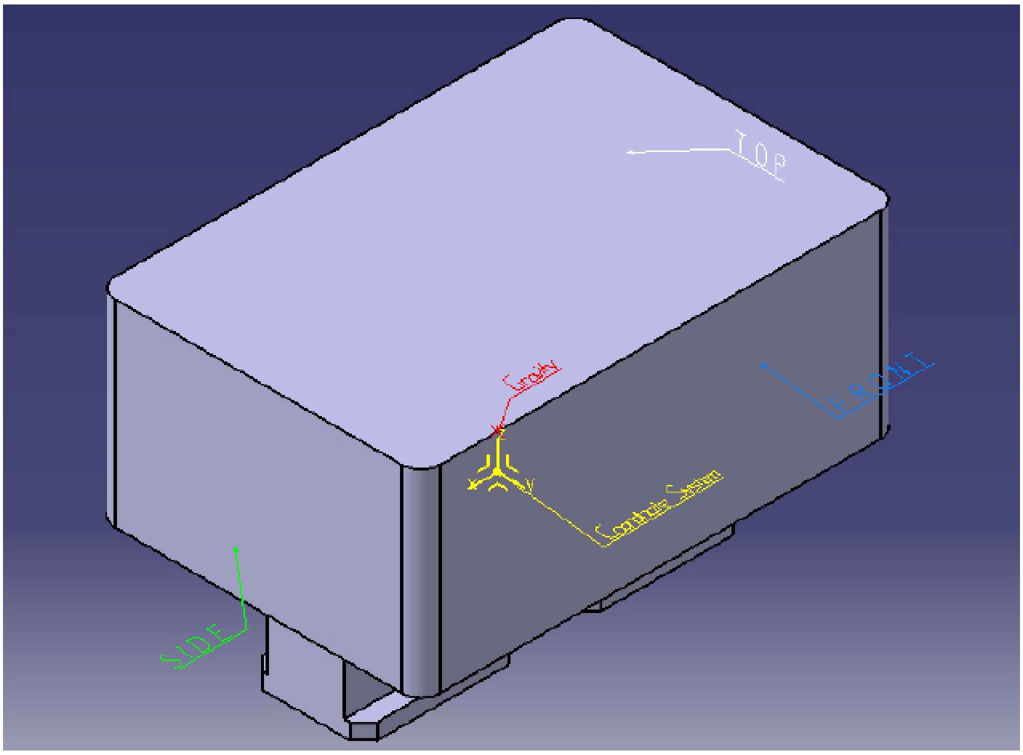
Volume: 2.222e-004 m3
Weight: 0.602 kg

Gravity:
X = -5.632e-015 mm ≈ 0 mm
Y = 1.144e-014 mm ≈ 0 mm
Z = 10.994 mm

Inertial Matrix:

Ixx = 0.001 kgxm2;	Ixy = 0 kgxm2;	Ixz = 0 kgxm2;
Iyx = 0 kgxm2;	Iyy = 0.002 kgxm2;	Iyz = 0 kgxm2;
Izx = 0 kgxm2;	Izy = 0 kgxm2;	Izz = 0.002 kgxm2;

Coordinate System:
Px = 75 mm (from x-axis to side)
Py = 51 mm (from y-axis to front)
Pz = 62 mm (from z-axis to top)

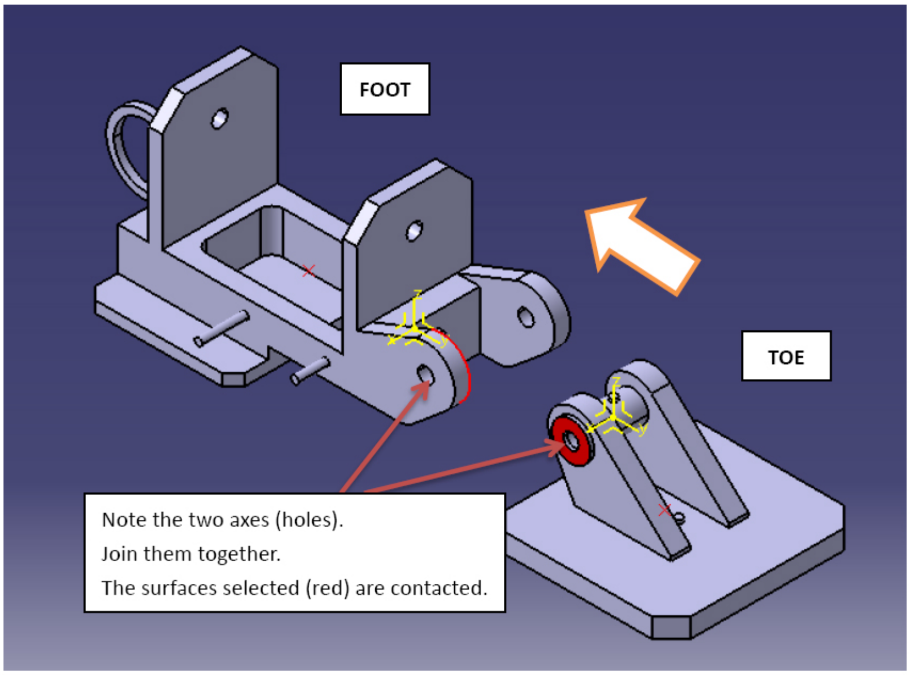


Appendix B - Assembly Phases

Step 1

Assemble **Toe** and **Foot**.

Before:

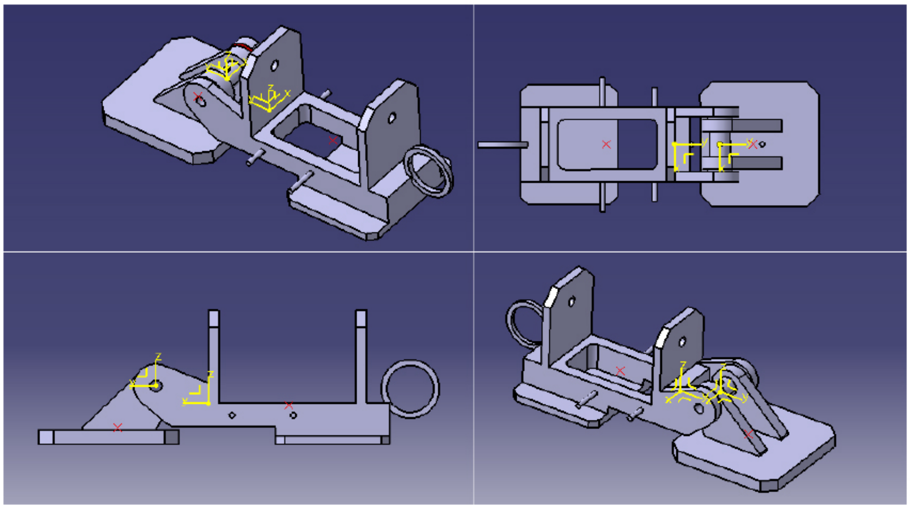


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

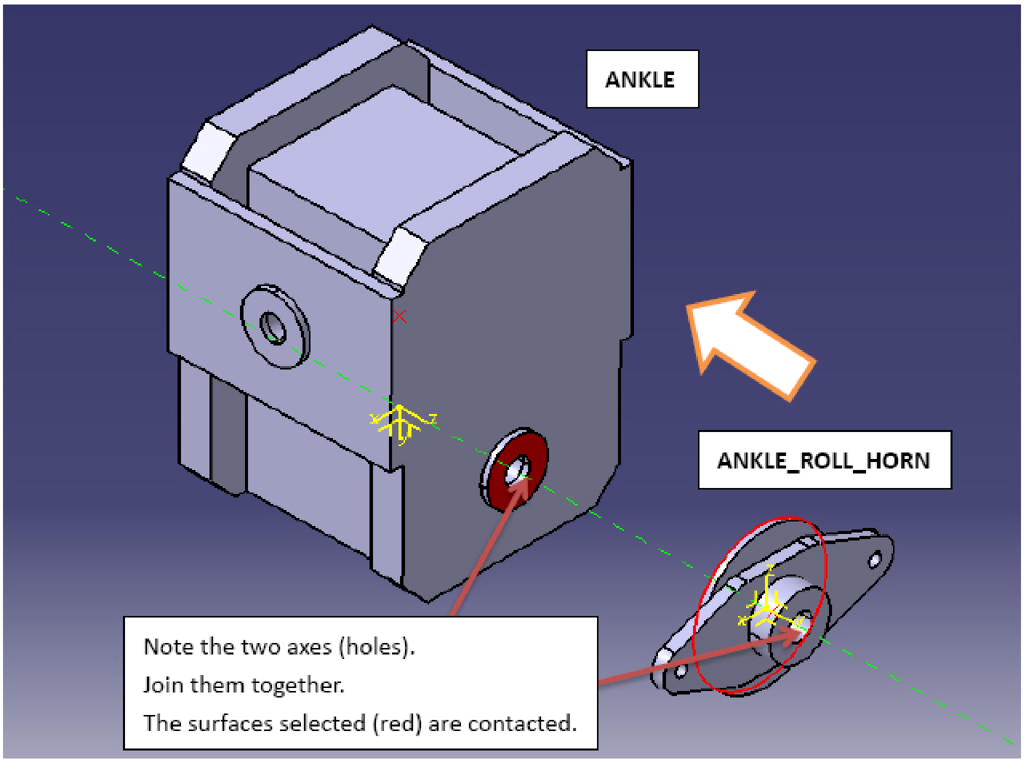
After:



Step 2

Assemble **Ankle** and **Ankle_Roll_Horn**.

Before:

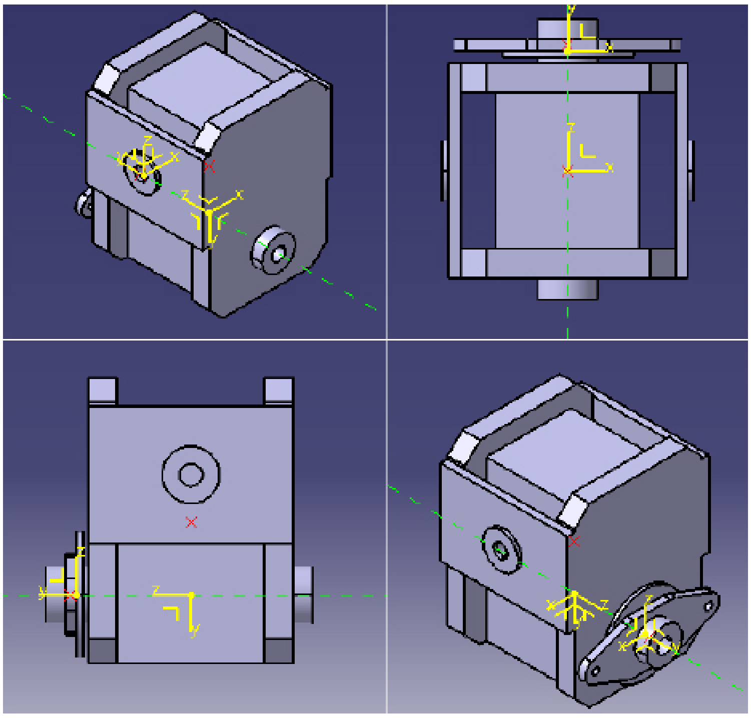


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

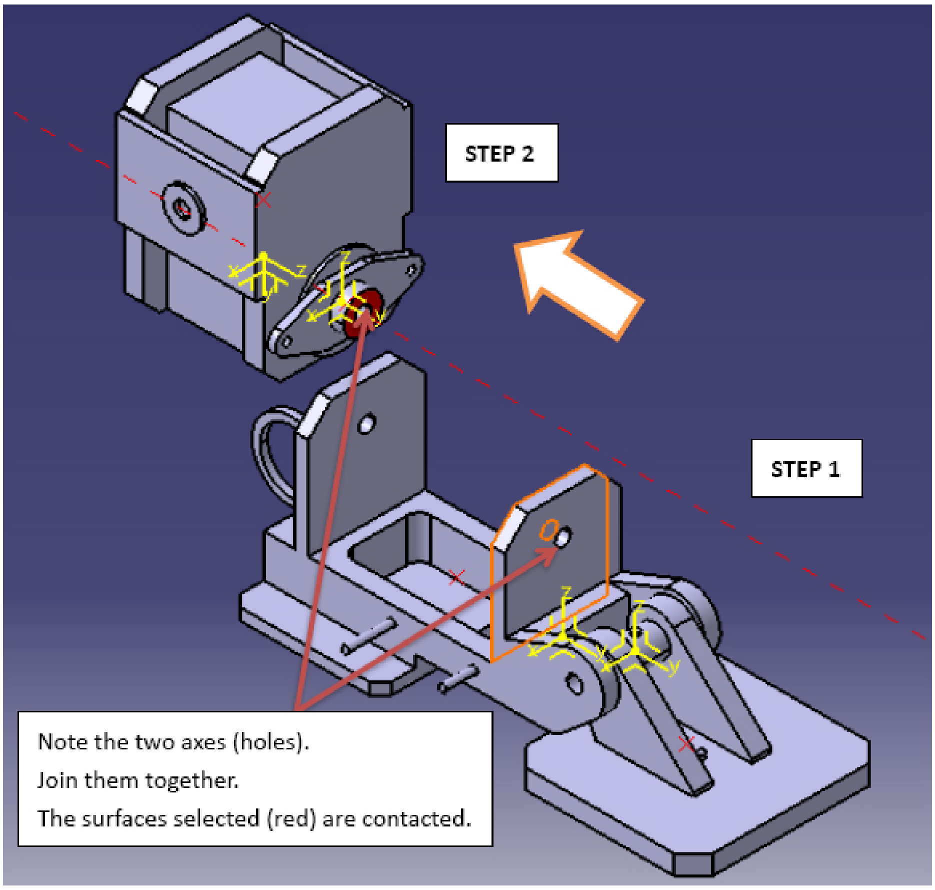
After:



Step 3

Assemble **Step 1** and **Step 2** parts.

Before:

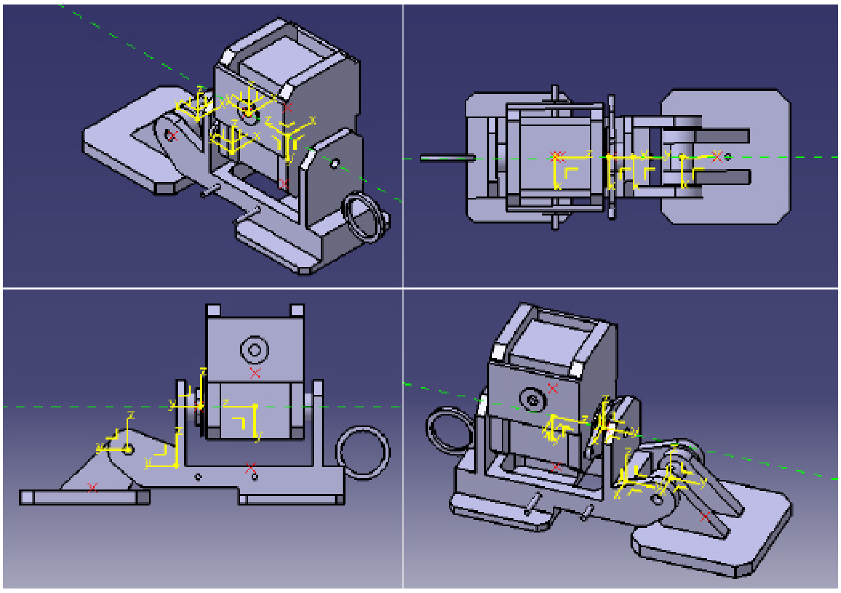


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

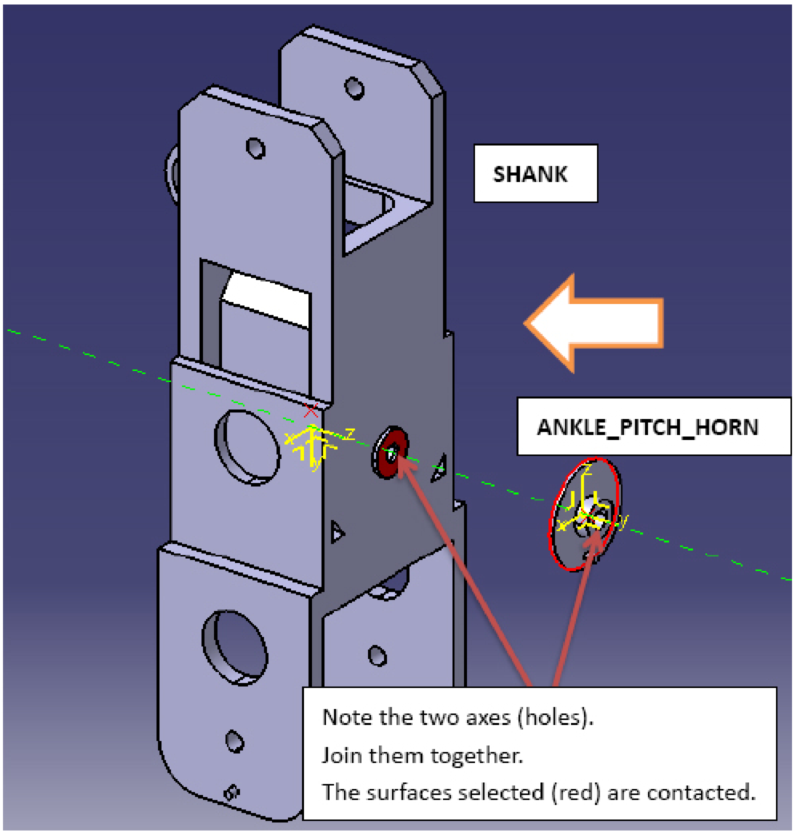
After:



Step 4

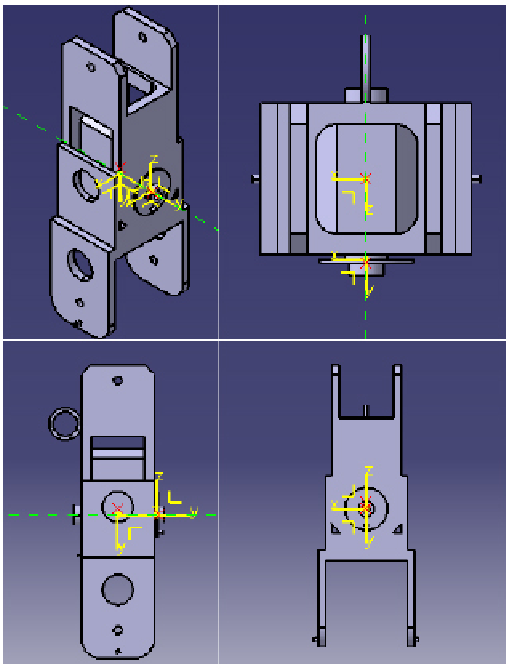
Assemble **Shank** and **Ankle_Pitch_Horn**.

Before:



Note the two axes (holes).
Join them together.
The surfaces selected (red) are contacted.

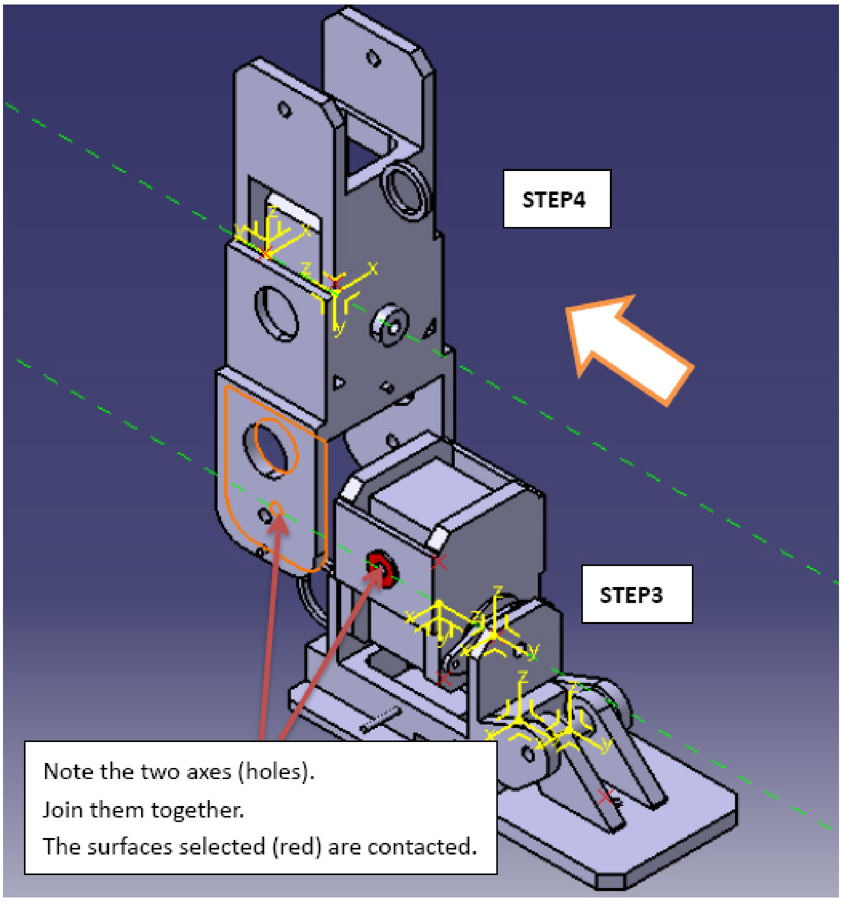
After:



Step 5

Assemble **Step3** and **Step4** parts.

Before:

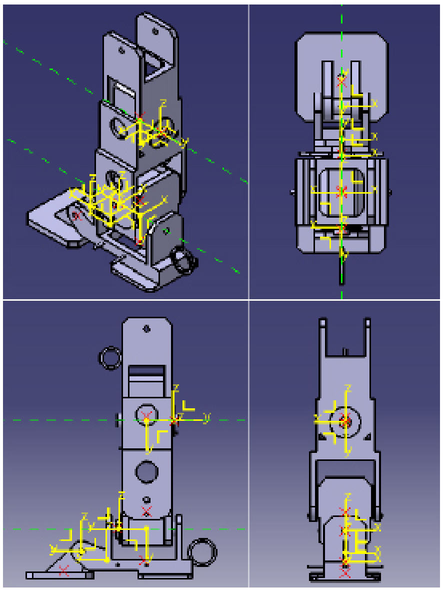


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

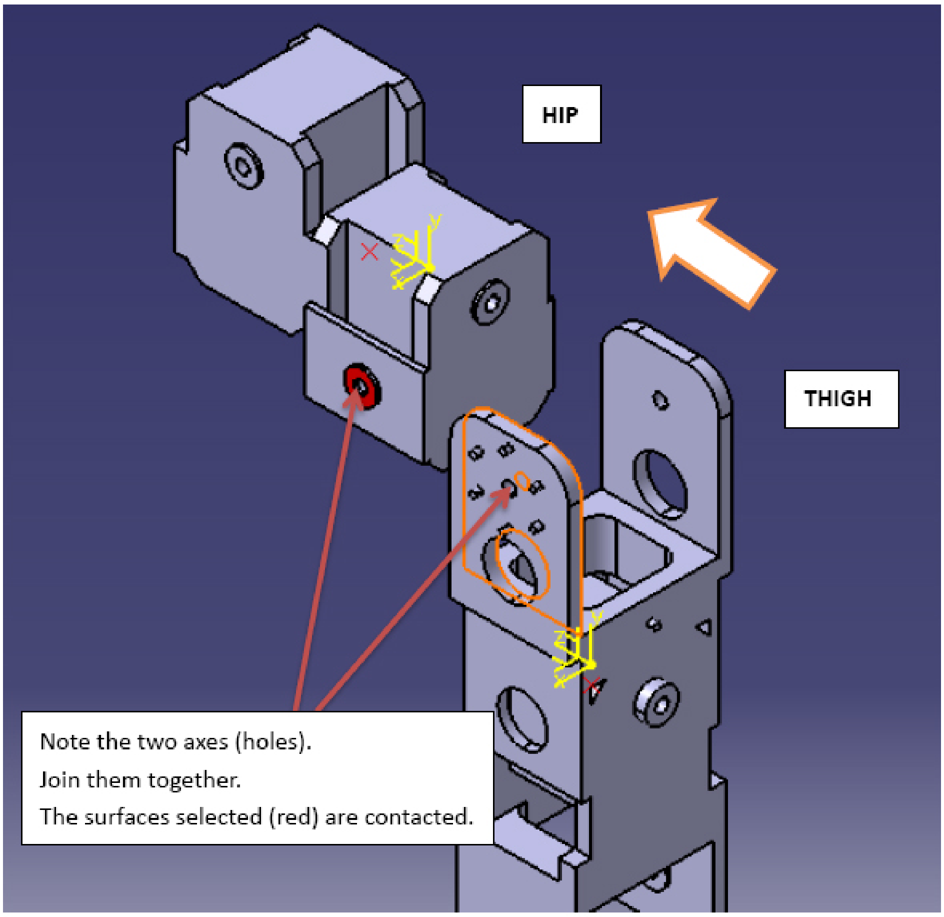
After:



Step 6

Assemble **Thigh** and **Hip** parts.

Before:

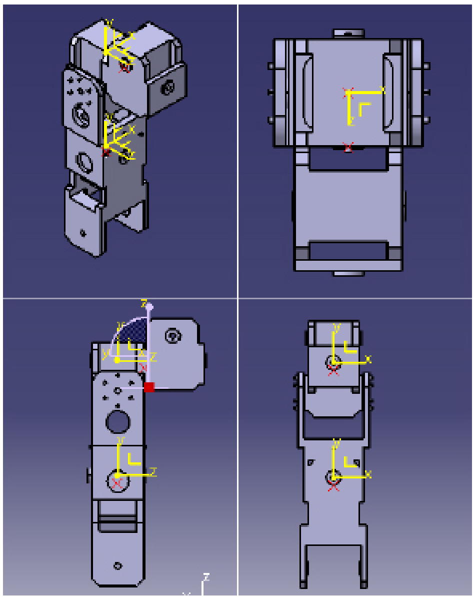


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

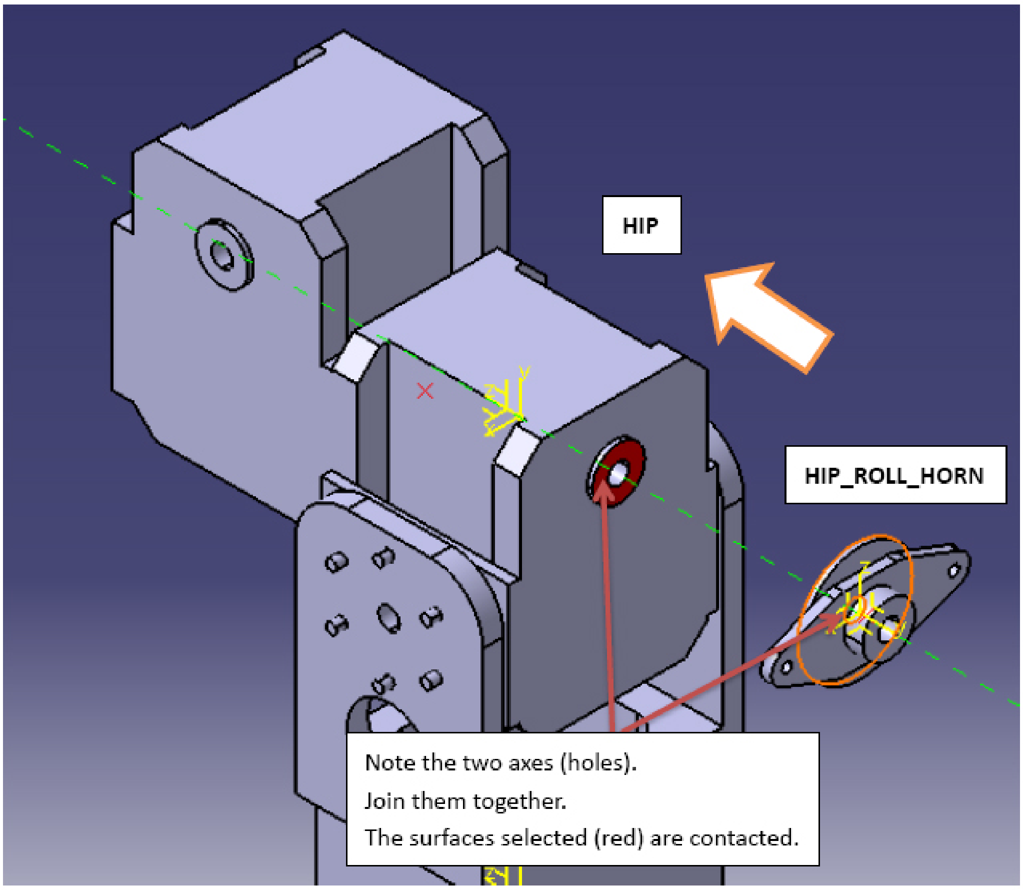
After:



Step 7

Assemble **Hip** and **Hip_Roll_Horn** parts.

Before:

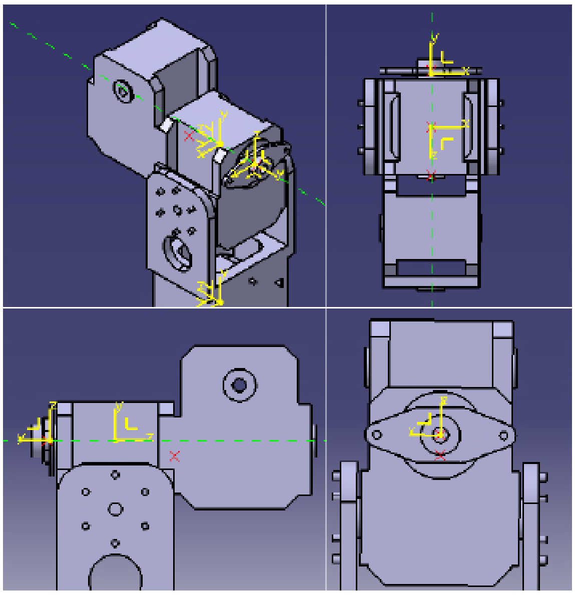


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

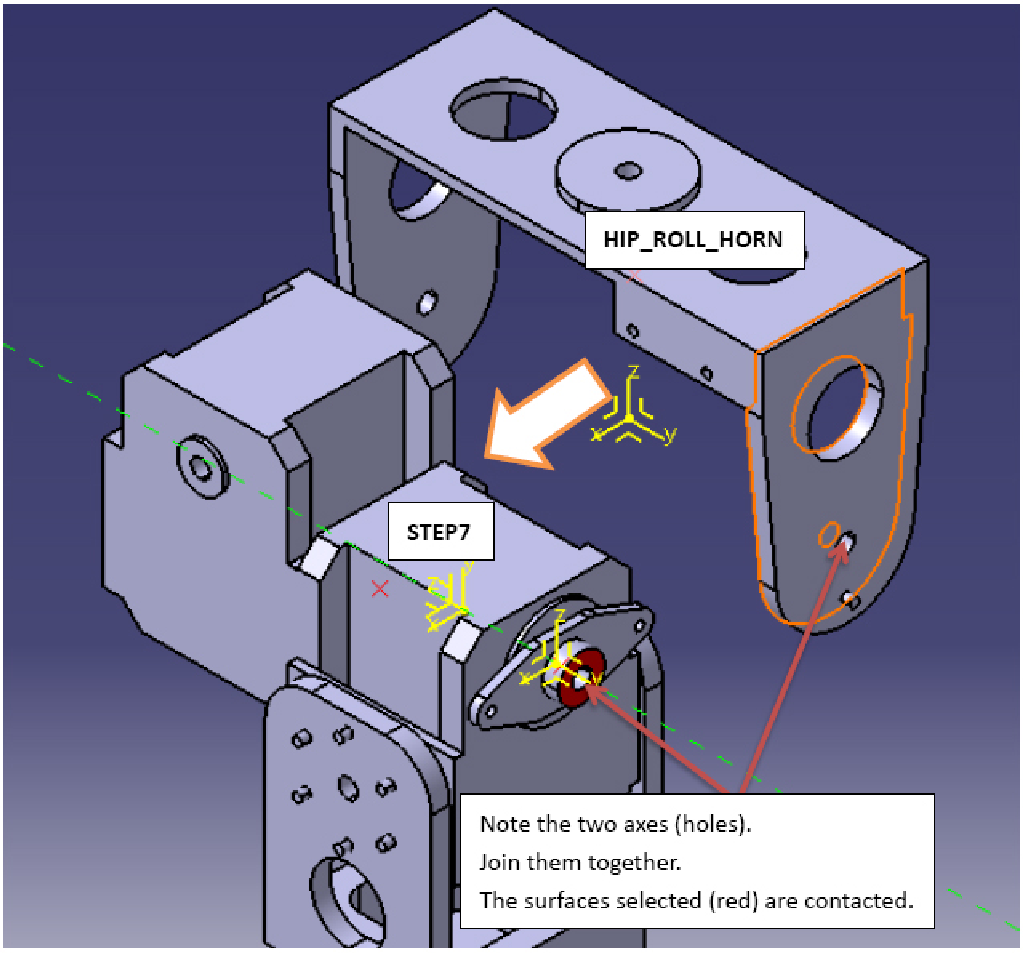
After:



Step 8

Assemble **Step7** and **Hip_Bracket** parts.

Before:

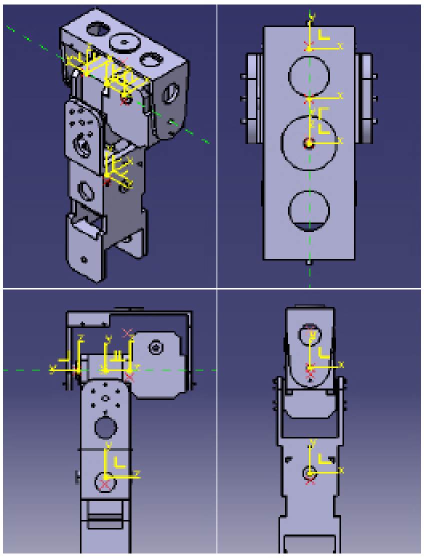


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

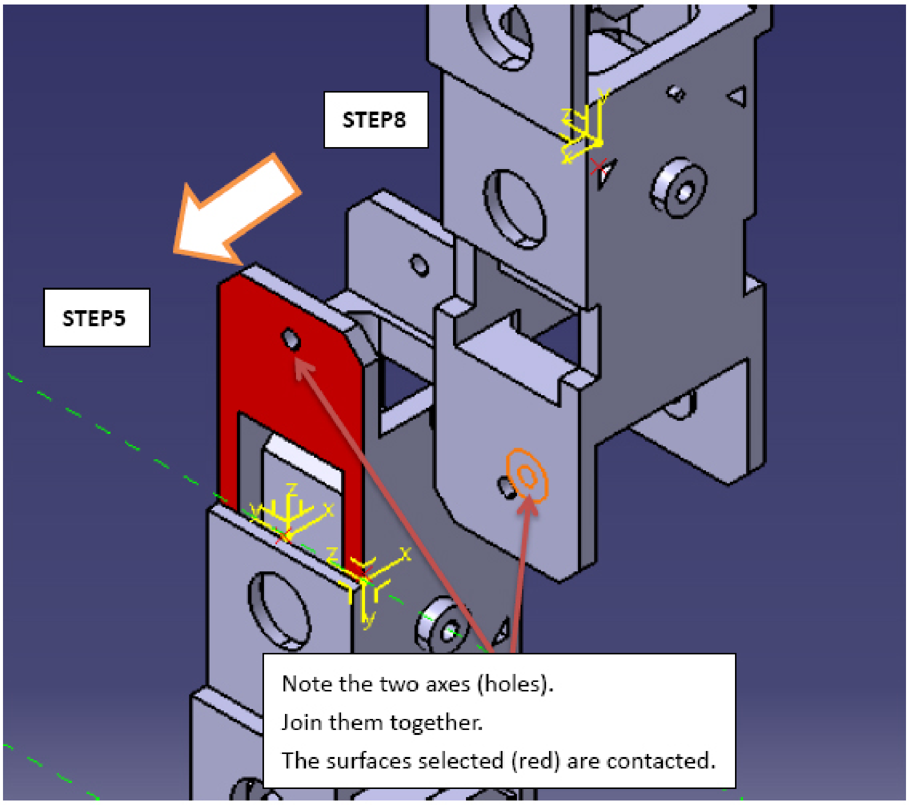
After:



Step 9

Assemble **Step5** and **Step8** parts.

Before:

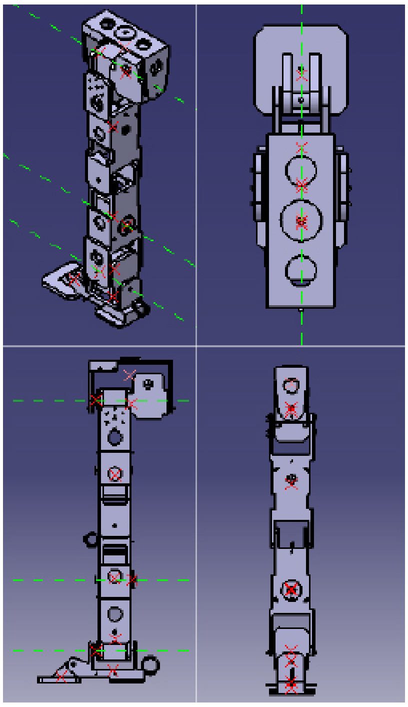


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

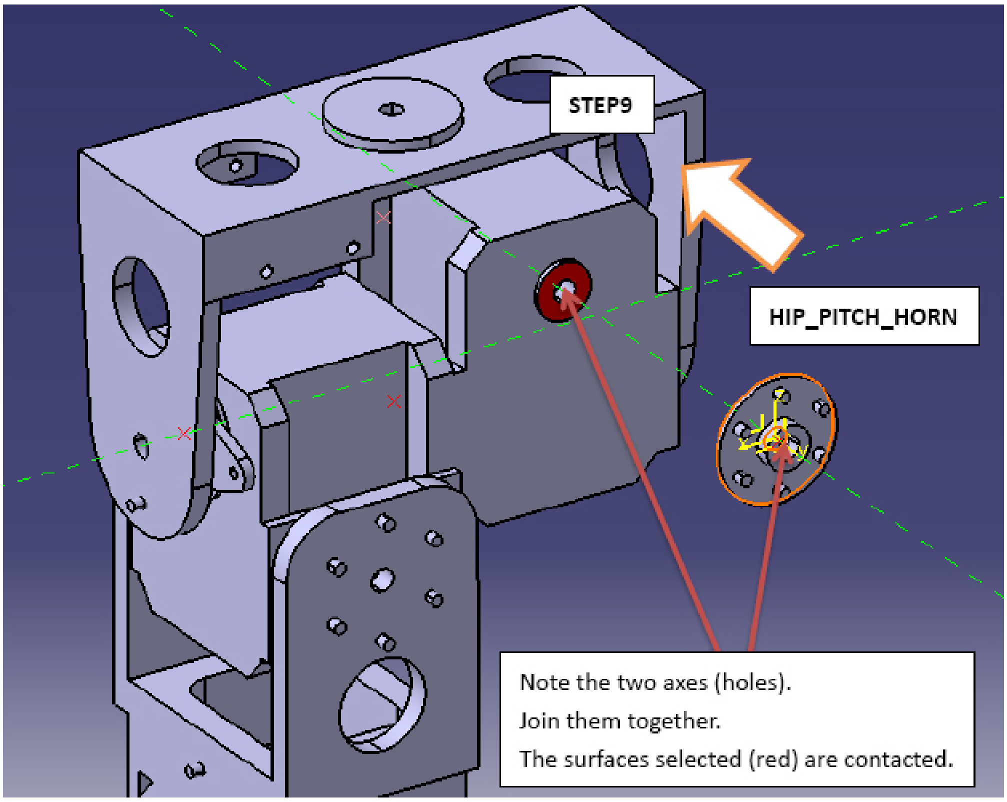
After:



Step 10 Left Leg

Assemble **Step9** and **Hip_Pitch_Horn** parts.

Before:

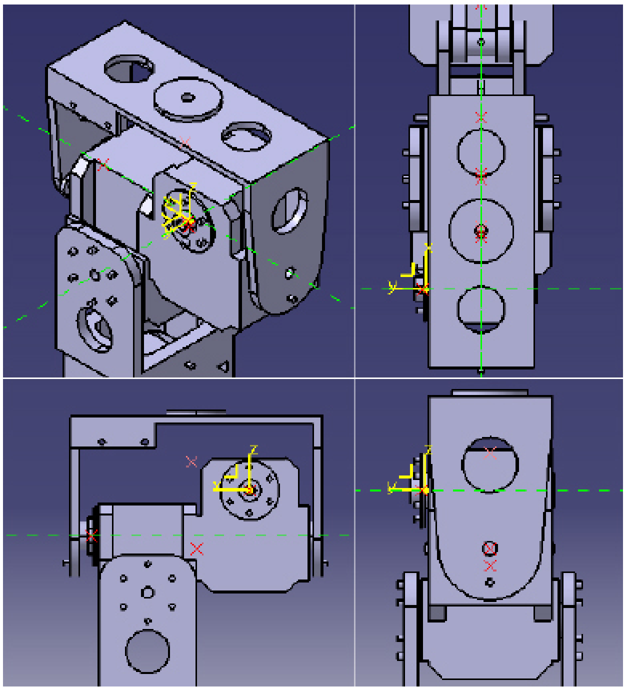


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

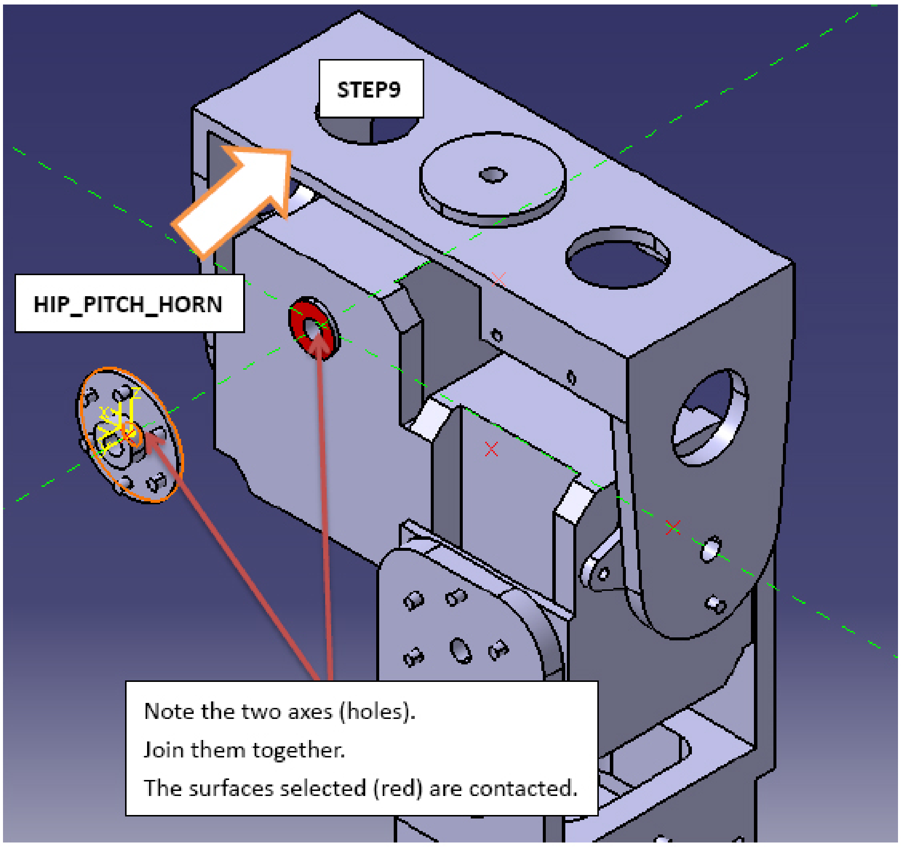
After:



Step 11 Right Leg

Assemble **Step9** and **Hip_Pitch_Horn** parts.

Before:

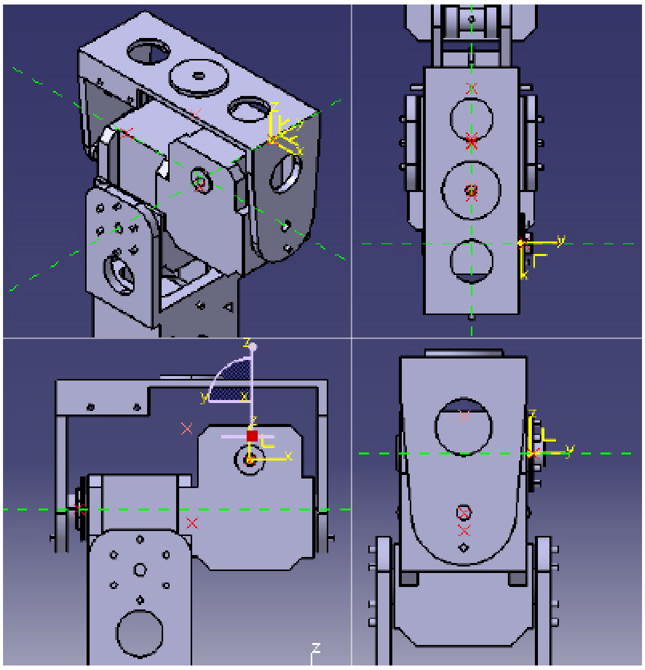


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

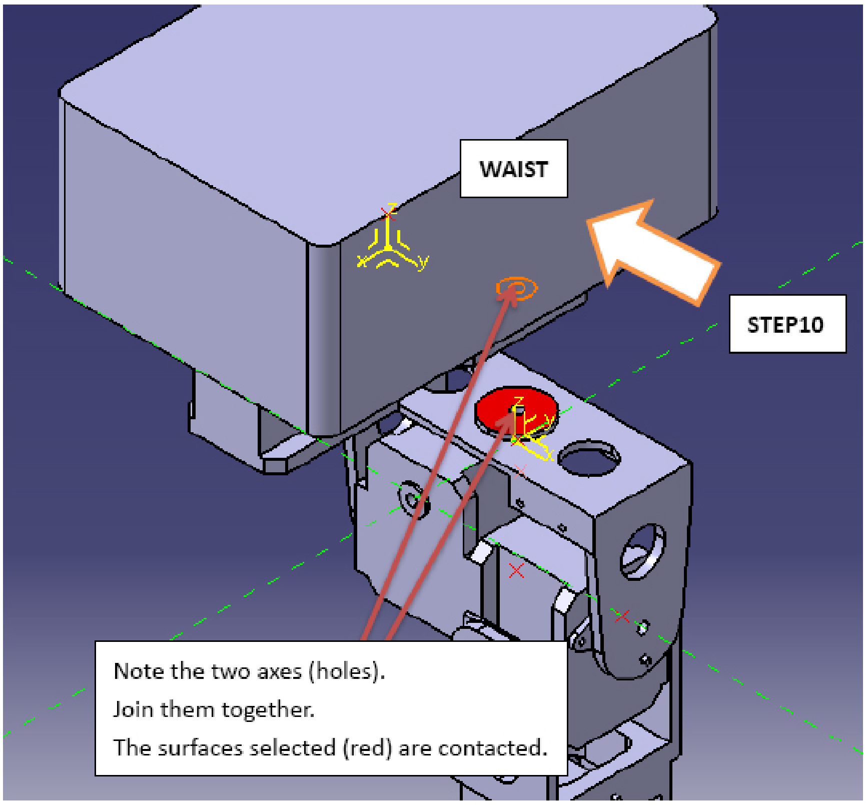
After:



Step 12

Assemble **Step10** and **Waist** parts.

Before:

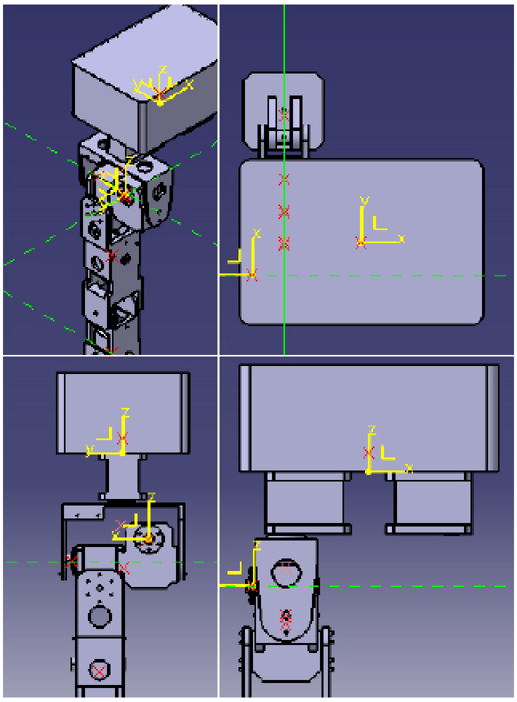


Note the two axes (holes).

Join them together.

The surfaces selected (red) are contacted.

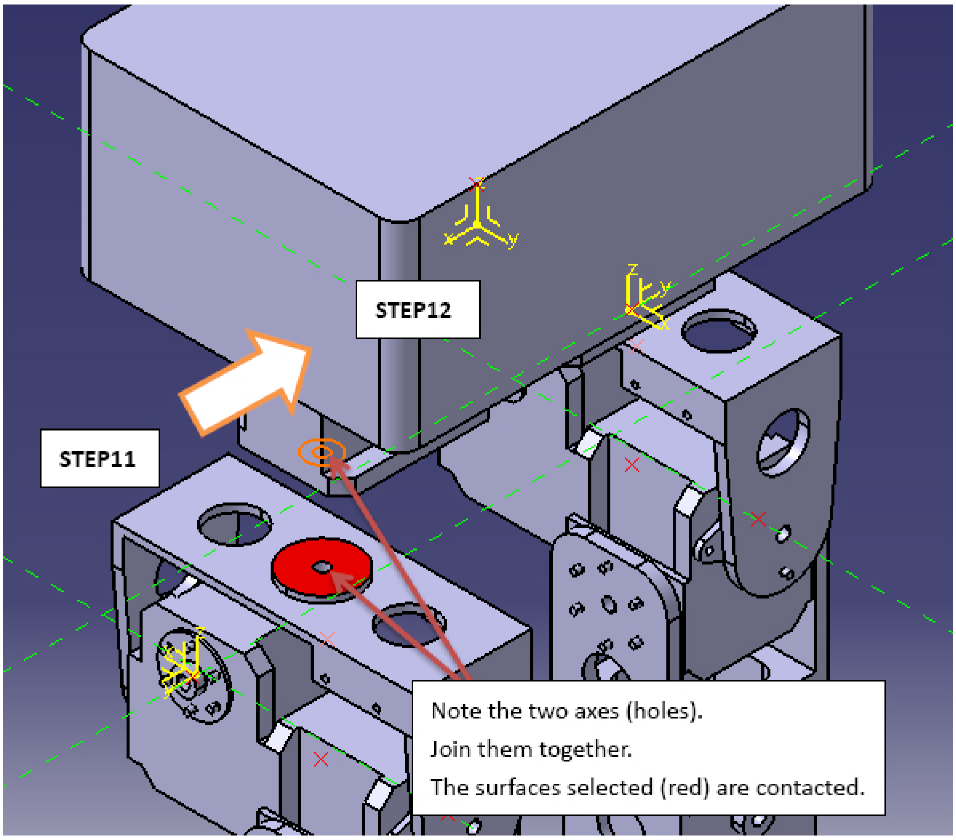
After:



Step 13

Assemble **Step11** and **Step12** parts.

Before:

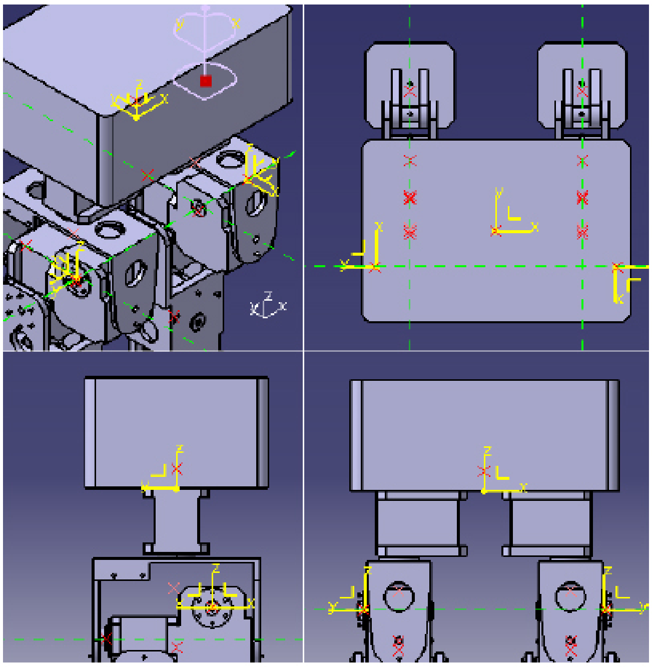


Note the two axes (holes).

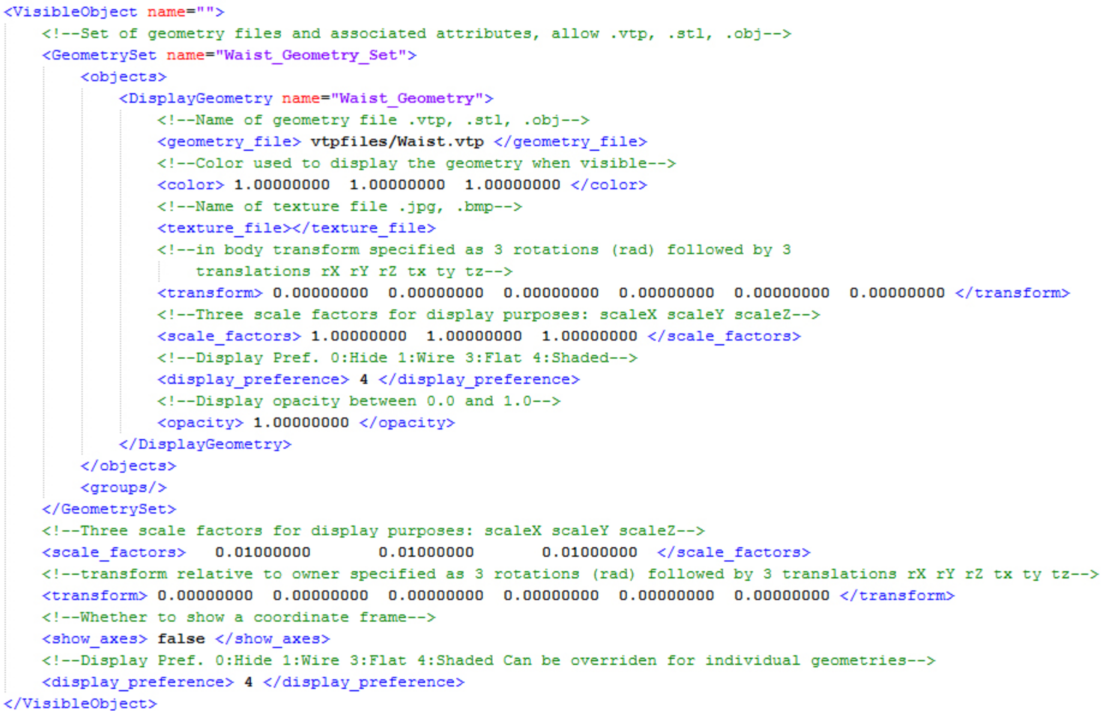
Join them together.

The surfaces selected (red) are contacted.

After:

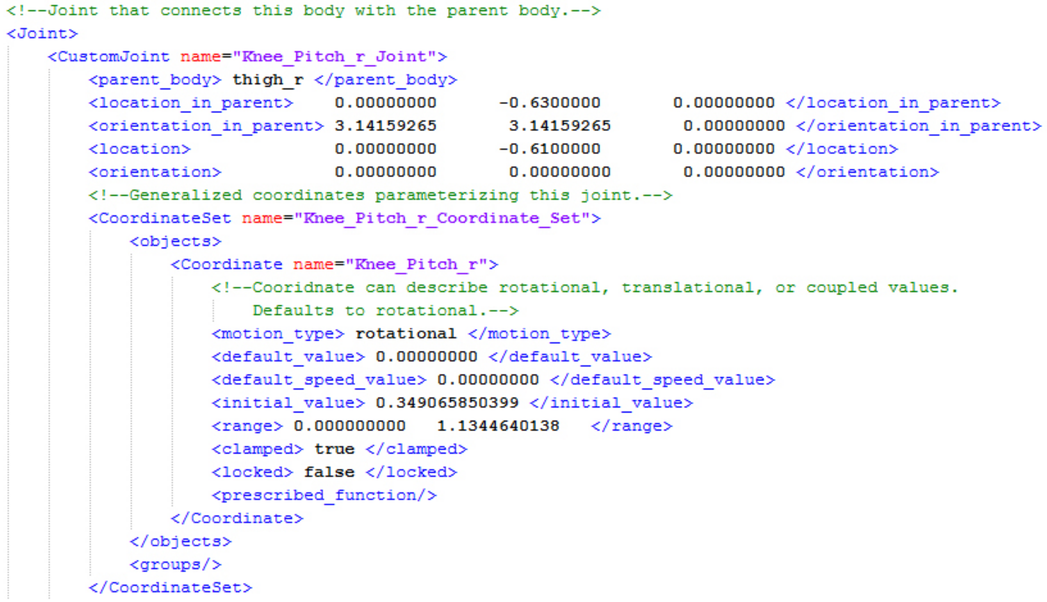


Appendix C - Extract from .osim Model



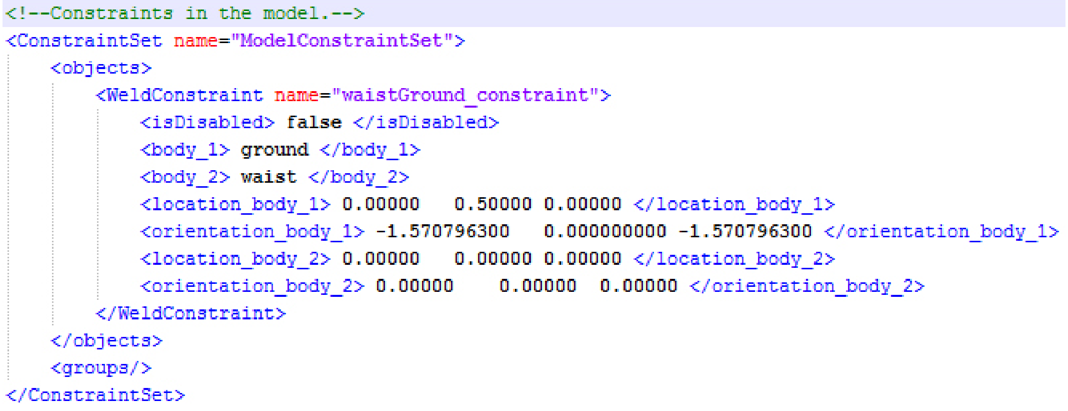
```
<VisibleObject name="">
    <!--Set of geometry files and associated attributes, allow .vtp, .stl, .obj-->
    <GeometrySet name="Waist_Geometry_Set">
        <objects>
            <DisplayGeometry name="Waist_Geometry">
                <!--Name of geometry file .vtp, .stl, .obj-->
                <geometry_file> vtpfiles/Waist.vtp </geometry_file>
                <!--Color used to display the geometry when visible-->
                <color> 1.00000000  1.00000000  1.00000000 </color>
                <!--Name of texture file .jpg, .bmp-->
                <texture_file></texture_file>
                <!--in body transform specified as 3 rotations (rad) followed by 3
                    translations rX rY rZ tx ty tz-->
                <transform> 0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000 </transform>
                <!--Three scale factors for display purposes: scaleX scaleY scaleZ-->
                <scale_factors> 1.00000000  1.00000000  1.00000000 </scale_factors>
                <!--Display Pref. 0:Hide 1:Wire 3:Flat 4:Shaded-->
                <display_preference> 4 </display_preference>
                <!--Display opacity between 0.0 and 1.0-->
                <opacity> 1.00000000 </opacity>
            </DisplayGeometry>
        </objects>
        <groups/>
    </GeometrySet>
    <!--Three scale factors for display purposes: scaleX scaleY scaleZ-->
    <scale_factors>   0.01000000        0.01000000        0.01000000  </scale_factors>
    <!--transform relative to owner specified as 3 rotations (rad) followed by 3 translations rX rY rZ tx ty tz-->
    <transform> 0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000 </transform>
    <!--Whether to show a coordinate frame-->
    <show_axes> false </show_axes>
    <!--Display Pref. 0:Hide 1:Wire 3:Flat 4:Shaded Can be overriden for individual geometries-->
    <display_preference> 4 </display_preference>
</VisibleObject>
```

Figure 7.1: Typical structure of Visible Object section definition.



```
<!--Joint that connects this body with the parent body.-->
<Joint>
    <CustomJoint name="Knee_Pitch_r_Joint">
        <parent_body> thigh_r </parent_body>
        <location_in_parent>    0.00000000      -0.6300000       0.00000000 </location_in_parent>
        <orientation_in_parent> 3.14159265       3.14159265       0.00000000 </orientation_in_parent>
        <location>              0.00000000      -0.6100000       0.00000000 </location>
        <orientation>           0.00000000       0.00000000       0.00000000 </orientation>
        <!--Generalized coordinates parameterizing this joint.-->
        <CoordinateSet name="Knee_Pitch_r_Coordinate_Set">
            <objects>
                <Coordinate name="Knee_Pitch_r">
                    <!--Cooridnate can describe rotational, translational, or coupled values.
                        Defaults to rotational.-->
                    <motion_type> rotational </motion_type>
                    <default_value> 0.00000000 </default_value>
                    <default_speed_value> 0.00000000 </default_speed_value>
                    <initial_value> 0.349065850399 </initial_value>
                    <range> 0.000000000   1.1344640138   </range>
                    <clamped> true </clamped>
                    <locked> false </locked>
                    <prescribed_function/>
                </Coordinate>
            </objects>
            <groups/>
        </CoordinateSet>
```

Figure 7.2: Typical structure of Custom Joint section definition.



```
<!--Constraints in the model.-->
<ConstraintSet name="ModelConstraintSet">
    <objects>
        <WeldConstraint name="waistGround_constraint">
            <isDisabled> false </isDisabled>
            <body_1> ground </body_1>
            <body_2> waist </body_2>
            <location_body_1> 0.00000   0.50000 0.00000 </location_body_1>
            <orientation_body_1> -1.570796300   0.000000000 -1.570796300 </orientation_body_1>
            <location_body_2> 0.00000   0.00000 0.00000 </location_body_2>
            <orientation_body_2> 0.00000    0.00000  0.00000 </orientation_body_2>
        </WeldConstraint>
    </objects>
    <groups/>
</ConstraintSet>
```

Figure 7.3: Typical structure of Weld Constraint section definition.

Appendix D - Muscles vs. Springs

Subdivision of human muscles based on movements in which they are involved.

Hip abduct.	***Hip adduct.***	***Hip flex.***	***Hip exten.***	***Hip inrot***	***Hip exrot***
glut_max1	add_brev	add_brev	add_long	glut_med1	gem
glut_med1	add_long	add_long	add_mag1	glut_min1	glut_med3
glut_med2	add_mag1	glut_med1	add_mag2	iliacus	glut_min3
glut_med3	add_mag2	glut_min1	add_mag3	psoas	peri
glut_min1	add_mag3	grac	bifemlh	tfl	quad_fem
glut_min2	bifemlh	iliacus	glut_max1		
glut_min3	grac	pect	glut_max2		
peri	pect	psoas	glut_max3		
sar	semimem	rect_fem	glut_med3		
tfl	semiten	sar	glut_min3		
		tfl	semimem		
			semiten		

Knee bend.	***Knee ext.***	***Ankle pf.***	***Ankle df.***	***Inverter***	***Everter***
bifemlh	rect_fem	flex_dig	ext_dig	ext_hal	ext_dig
bifemsh	vas_int	flex_hal	ext_hal	flex_dig	per_brev
grac	vas_lat	lat_gas	per_tert	flex_hal	per_long
lat_gas	vas_med	med_gas	tib_ant	tib_ant	per_tert
med_gas		per_brev		tib_post	
sar		per_long			
semimem		soleus			
semiten		tib_post			

back ext.	***back flex.***	***back rlb***	***back llb***	***back introt***	***back extrot***
ercspn_r	extobl_r	ercspn_r	ercspn_l	ercspn_r	ercspn_l
ercspn_l	extobl_l	extobl_r	extobl_l	extobl_l	extobl_r
	intobl_r	intobl_r	intobl_l	intobl_r	intobl_l
	intobl_l				

Subdivision of robot springs based on corresponding muscles and movements in which they are involved.

Hip add/abd	*Hip flex/exten*	*Hip inrot/exrot*
bifem	bifem	motors
rect_fem	rect_fem	
add_long	glut_max	
add_brev	psoas	
glut_min_A	add_long	
glut_min_B	glut_min_A	
	glut_med_A	
	glut_med_B	

Knee bend/ext	*Ankle pf/df*	*Inv/evert*	*Toes flex/ext*
gastrocnemius	gastrocnemius	gastrocnemius	ext_dig
bifem	soleus	soleus	
vas_med	tib_post_A	tib_post_A	
rect_fem	per_long	tib_post_B	
	tib_ant	per_long	
		per_tert	
		tib_ant	

Bibliography

[1] A. Albu-Schaffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimbock, S. Wolf, G. Hirzinger. **Soft Robotics - Adaptable Compliance -**. *IEEE Robotics & Automation Magazine*, September 2008.

[2] A. Albu-Schaffer, S. Haddadin, Ch. Ott, A. Stemmer, T. Wimbock, G. Hirzinger. **The DLR lightweight robot: design and control concepts for robots in human environments.** *Industrial Robot: An International Journal, Vol.34,No.5*, 2007.

[3] A. M. Dollar, H. Herr. **Lower Extremity Exoskeletons and Active Orthoses: Challenges and State-of-the-Art**. *IEEE Transactions on Robotics, Vol.24, No.1*, February 2008.

[4] A. M. M. Omer, R. Ghorbani, H. Lim, A. Takanishi. **Semi-Passive Dynamic Walking for Humanoid Robot Using Controllable Spring Stiffness on the Ankle Joint**. *Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zeland*, February 2009.

[5] A. Nagano, B. R. Umberger, M. W. Marzke, K. G. M. Gerritsen. **Neuromusculoskeletal Computer Modeling and Simulation of Upright, Straight-Legged, Bipedal Locomotion of *Australopithecus afarensis* (A.L. 288-1)**. *American Journal of Physical Anthropology*, 2004.

[6] A. Seth, M. Sherman, J. A. Reinbolt, S. L. Delp. **OpenSim: a musculoskeletal modeling and simulation framework for *in silico* investigations and exchange**. *ELSEVIER, 2011 Symposium on Human Body Dynamics, [Online: www.sciencedirect.com]*, 2011.

[7] B. Mohl. **Bionic robot arm with compliant actuators**. *Proc. of SPIE - Sensor Fusion and Decentralized Control in Robotic Systems III, Vol.4196*, October 2000.

[8] C. Canton-Ferrer, J. R. Casas, M. Pardas. **Towards a low cost multi-camera marker based human motion capture system**. *IEEE*, 2009.

[9] C. L. Vaughan. **Theories of bipedal walking: an odyssey**. *ELSEVIER, Journal of Biomechanics, 36*, 2003.

[10] C. Ott, D. Lee, Y. Nakamura. **Motion Capture based Human Motion Recognition and Imitation by Direct Marker Control**. *8th IEEE-RAS International Conference on Humanoid Robots - Daejeon, Korea*, December 2008.

[11] C. Winby, D. G. Lloyd, T. Kirk. **Evaluation of different analytical methods for subject-specific scaling of musculotendon parameters**. *Journal of Biomech., Vol.41, No.8*, 1992.

[12] D. Forg, H. Ulbirch, A. Seyfarth. **Study of a Bipedal Robot with Elastic Elements**. *VDE VERLAG GMBH - Berlin, Offenbach. ISR / ROBOTIK 2010*, 2010.

[13] D. G. Lloyd, T. F. Besier. **An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo**. *Journal of Biomech., Vol.36, No.6*, 2003.

[14] D. G. Lloyd, T. S. Buchanan. **A model of load sharing between muscles and soft tissues at the human knee during static tasks**. *Journal of Biomechanical Engineering, Vol.118, No.3*, 1996.

[15] D. G. Thelen, F. C. Anderson. **Using computed muscle control to generate forward dynamic simulations of human walking from experimental data**. *ELSEVIER, Journal of Biomechanics, 39*, 2006.

[16] D. Shin, I. Sardellitti, Y. Park, O. Khatib, M. Cutkosky. **Design and Control of a Bio-Inspired Human-Friendly Robot**. , .

[17] D. Vischer, O. Khatib. **Design and Development of Torque-Controlled Joints**. *Experimental Robotics, Vol.1*, 1990.

[18] D. Vischer, O. Khatib. **Design and Development of High-Performance Torque-Controlled Joints**. *IEEE Trans. Robot Automat., Vol.11*, August 1995.

[19] E. Ayyappa. **Normal Human Locomotion, Part 1: Basic Concepts and Terminology**. *American Academy of Orthotists & Prosthetists [Online: www.oandp.org]*, 1997.

[20] E. Ayyappa. **Normal Human Locomotion, Part 2: Motion, Ground Reaction Force and Muscle Activity**. *American Academy of Orthotists & Prosthetists [Online: www.oandp.org]*, 1997.

[21] E. Demircan, L. Sentis, V. De Sapio, O. Khatib. **Human Motion Reconstruction by Direct Control of Marker Trajectories**. , 2008.

[22] E. Demircan, T. Besier, S. Menon, O. Khatib. **Human Motion Reconstruction and Synthesis of Human Skills**. , .

[23] E. Kljuno, R. L. Williams II. **Humanoid Walking Robot: Modeling, Inverse Dynamics, and Gain Scheduling Control**. *Journal of Robotics*, 2010.

[24] E. M. Arnold, S. L. Delp. **Fibre operating lengths of human lower limb muscles during walking**. *Philosophical Transactions of The Royal Society - Biological Sciences*, 2011.

[25] F. E. Zajac. **Muscle and Tendon: Properties, Models, Scaling, and Application to Biomechanics and Motor Control**. *Critical Reviews in Biomecanical Engineering, Vol.17, Issue 4*, 1989.

[26] F. Iida, J. Rummel, A. Seyfarth. **Bipedal walking and running with spring-like biarticular muscles**. *ELSEVIER, Journal of Biomechanics, 41*, 2008.

[27] F. Iida, Y. Minekawa, J. Rummel, A. Seyfarth. **Toward a human-like biped robot with compliant legs**. *ELSEVIER, Robotics and Autonomous Systems [Online: www.elsevier.com/locate/robot]*, 2009.

[28] G. A. Pratt. **Low Impedance Walking Robots.** *Integrative and Comparative Biology, Vol.42*, 2002.

[29] G. van Oort, S. Stramigioli. **Geometric interpretation of the Zero-Moment Point**. *Work funded by the European Commission's Seventh Framework Programme as part of the project VIACTORS under grant no.231554*, .

[30] G. Venture, K. Ayusawa, D. Kulic, Y. Nakamura. **Monitoring The Segment Parameters During Long Term Physical Training From Motion Capture Data**. *31st Annual International Conference of the IEEE EMBS - Minneapolis, Minnesota, USA*, September 2009.

[31] G. Venture, K. Ayusawa, Y. Nakamura. **Identification of Human Mass Properties From Motion**. *Preprints of the 15the IFAC Symposium on System Identification - Saint-Malo, France*, July 2009.

[32] G. Venture, K. Ayusawa, Y. Nakamura. **Optimal Estimation of Human Body Segments Dynamics Using Realtime Visual Feedback**. *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems - St. Louis, USA*, October 2009.

[33] G. Venture, K. Ayusawa, Y. Nakamura. **Real-time Identification and Visualization of Human Segment Parameters**. *31st Annual International Conference of the IEEE EMBS - Minneapolis, Minnesota, USA*, September 2009.

[34] H. Kaminaga, J. Ono, Y. Shimoyama, T. Amari, Y. Katayama, Y. Nakamura. **Anthropomorphic Robot Hand with Hydrostatic Cluster Actuator and Detachable Passive Wire Mechanism**. *9th IEEE-RAS International Conference on Humanoid Robots*, December 2009.

[35] H. Kawamoto, S. Lee, S. Kanbe, Y. Sankai. **Power Assist Method for HAL-3 using EMG-based Feedback Controller**. *IEEE*, 2003.

[36] H. Kawamoto, Y. Sankai. **Comfortable Power Assist Control Method for Walking Aid by HAL-3**. *IEEE*, 2002.

[37] H. Kawamoto, Y. Sankai. **Power Assist System HAL-3 for Gait Disorder Person**. *Springer-Verlag Berlin Heidelberg*, 2002.

[38] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, M. Inaba. **An Advanced Musculoskeletal Humanoid Kojiro**. *IEEE*, 2007.

[39] J. E. Pratt, B. T. Krupp. **Series Elastic Actuators for legged robots.** *Proceedings of SPIE - The international society for optical engineering*, 2004.

[40] J. Hitt, T. Sugar, M. Holgate, R. Bellman, K. Hollander. **Robotic transtibial prosthesis with biomechanical energy regeneration**. *Emerald, Industrial Robot: An International Journal. [Online: www.emeraldinsight.com/0143-991X.htm]*, 2009.

[41] J. L. V. Leeuwen. **Muscle function in locomotion.** *Advances in Comparative and Environmental Physiology 11: Mechanics of Animal Locomotion. Berlin Heidelberg: Springer*, 1992.

[42] K. Hase, K. Miyashita, S. Ok, Y. Arakawa. **Human gait simulation with a neuromusculoskeletal model and evolutionary computation**. *The Journal of Visualization and Computer Animation*, 2003.

[43] K. Kurihara, S. Hoshino, K. Yamane, Y. Nakamura. **Optical Motion Capture System with Pan-Tilt Camera Tracking and Realtime Data Processing**. *Proceedings of the 2002 IEEE International Conference on Robotics & Automation - Washington, DC*, May 2002.

[44] K. Pearson, O. Ekeberg, A. Buschges. **Assessing sensory function in locomotor systems using neuro-mechanical simulations**. *ELSEVIER, Trends in Neurosciences, Vol.29, No.11 [Online: www.sciencedirect.com]*, September 2006.

[45] K. Radkhah, D. Scholz, O. von Stryk. **Toward Human-Like Bipedal Locomotion with Three-Segmented Elastic Legs**. *VDE VERLAG GMBH - Berlin, Offenbach. ISR / ROBOTIK 2010*, 2010.

[46] K. Radkhah, S. Kurowski, O. von Stryk. **Design considerations for a biologically inspired compliant four-legged robot.** *Proc. of the IEEE International Conference on Robotics and Biomimetics*, December 2009.

[47] K. Radkhah, S. Kurowski, T. Lens, O. von Stryk. **Compliant Robot Actuation by Feedforward Controlled Emulated Spring Stiffness**. *Proceedings of the Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR, 2010)*, 2010.

[48] K. Radkhah, T. Lens, A. Seyfarth, O. von Stryk. **On the influence of elastic actuation and monoarticular structures in biologically inspired bipedal robots.** *Proceedings of the 2010 IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2010.

[49] K. W. Hollander, R. Ilg, T. G. Sugar, D. Herring. **An Efficient Robotic Tendon for Gait Assistance**. *Transactions of the ASME, Vol.128*, October 2006.

[50] K. Yamane, Y. Nakamura. **Robot Kinematics and Dynamics for Modeling the Human Body**. , .

[51] Katsu Yamane and Yoshihiko Nakamura. **Comparative Stydy on Serial and Parallel Forward Dynamics Algorithms for Kinematic Chains**. *The International Journal of Robotics Research*, 2009.

[52] L. Sentis, O. Khatib. **A Whole-Body Control Framework for Humanoids Operating in Human Environments**. , .

[53] M. Field, Z. Pan, D. Stirling, F. Naghdy. **Human motion capture sensors and analysis in robotics**. *Emerald, Industrial Robot: An International Journal. [Online: www.emeraldinsight.com/0143-991X.htm]*, 2011.

[54] M. M. Williamson. **Series Elastic Actuators**. *Massachusetts Institute of Technology, Artificial Intelligence Laboratory*, January 1995.

[55] M. Sartori. **A Neuromuscular Human-Machine Interface for Applications in Rehabilitation Robotics**. *PhD Thesis - Department of Information Engineering, Padova*, January 2011.

[56] M. Viceconti, G. Clapworthy, S. Van Sint Jan. **The Virtual Physiological Human - A European Initiative for *in silico* Human Modeling -**. *The Journal of Physiological Sciences, Vol.58, No.7*, December 2008.

[57] M. Vukobratovic, B. Borovac. **Zero-Moment Point - Thirty Five Years of its Life**. *International Journal of Humanoid Robotics*, 2004.

[58] M. Wisse. **Essentials of dynamic walking - Analysis and design of two-legged robots**. *PhD Thesis - ISBN 90-77595-82-1*, September 2004.

[59] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, S. Delp. **Robotics-based synthesis of human motion**. *ELSEVIER, Journal of Physiology - Paris*, 2009.

[60] O. Khatib, J. Warren, V. De Sapio, L. Sentis. **Human-like motion from physiologically-based potential energies**. , 2004.

[61] O. Khatib, L. Sentis, J. Park. **A Unified Framework for Whole-Body Humanoid Robot Control with Multiple Constraints and Contacts**. , .

[62] O. Khatib, O. Brock, K. Chang, F. Conti, D. Ruspini, L. Sentis. **Robotics and Interactive Simulation**. *Communications of the ACM*, March 2002.

[63] R. Boulic, N. Magnenat-Thalmann, D. Thalmann. **A Global Human Walking Model with Real-Time Kinematic Personification**. , .

[64] R. Davoodi, I. E. Brown, N. Lan, M. Mileusnic, G. E. Loeb. **An Integrated Package of Neuromusculoskeletal Modeling Tools in Simulink**. *Proceedings of the 23rd Annual EMBS International Conference - Istanbul, Turkey*, 2001.

[65] R. F. Ker. **Dynamic tensile properties of the plantaris tendon of sheep.** *Journal of Experimental Biology, Vol.93*, 1981.

[66] R. Siegwart, I. R. Nourbakhsh. **Introduction to Autonomous Mobile Robots**. *2004 - Massachusetts Institute of Technology*, 2004.

[67] R. Slyper, J. K. Hodgins. **Action Capture with Accelerometers**. *Eurographics / ACM SIGGRAPH APH Symposium on Computer Animation*, 2008.

[68] S. Altwegg. **Torque Control of High Compliant Series Elastic Actuator**. *ETH - Autonomous Systems Lab, Prof. Roland Siegwart*, 2010.

[69] S. Attygalle, M. Duff, T. Rikakis, J. He. **Low-cost, at-home assessment system with Wii Remote based motion capture**. *IEEE*, 2008.

[70] S. Corazza, L. Mundermann, A. M. Chaudhari, T. Demattio, C. Cobelli, T. P. Andriacchi. **A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach**. *Annals of Biomedical Engineering, Vol.34, No.6*, June 2006.

[71] S. H. Collins, A. D. Kuo. **Recycling Energy to Restore Impaired Ankle Function during Human Walking**. *PLOS ONE, Vol.5, Issue 2 [Online: www.plosone.org]*, February 2010.

[72] S. Klug, T. Lens, O. von Stryk, B. Mohl, A. Karguth. **Biologically inspired robot manipulator for new applications in automation engineering.** *Proc. of Robotik, ser. VDI-Berichte, no. 2012. VDI Wissensforum GmbH. [Online available: www.biorob.de]*, June 2008.

[73] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, D. G. Thelen. **OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement**. *IEEE Transactions on Biomedical Engineering, Vol.54, No.11*, November 2007.

[74] S. L. Delp, J. P. Loan. **A Computational Framework for Simulating and Analyzing Human and Animal Movement**. *IEEE Computer Society, Computing in Science & Engineering*, September/October 2000.

[75] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, J. M. Rosen. **An Interactive Graphics-Based Model of the Lower Extremity to Study Orthopaedic Surgical Procedures**. *IEEE Transactions on Biomedical Engineering, Vol.37, No.8*, August 1990.

[76] T. B. Moeslund, A. Hilton, V. Kruger. **A survey of advances in vision-based human motion capture and analysis**. *ELSEVIER, Computer Vision and Image Understanding [Online: www.sciencedirect.com]*, October 2006.

[77] T. Buchanan, D. G. Lloyd, K. Manal, T. F. Besier. **Neuromusculoskeletal modeling: Estimation of muscle forces and joint moments and movements from measurements from neural command**. *Journal of Applied Biomechanics, Vol.20, No.4*, 2004.

[78] T. G. Sugar, K. W. Hollander, J. K. Hitt. **Walking with Springs**. *Human Machine Integration Laboratory*, .

[79] T. Lens, J. Kunz, O. von Stryk. **Dynamic Modeling of the 4 DoF BioRob Series Elastic Robot Arm for Simulation and Control.** *Proceedings of the Intl. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2010)*, 2010.

[80] T. Lens, K. Radkhah, O. von Stryk. **Simulation of Dynamics and Realistic Contact Forces for Manipulators and Legged Robots with High Joint Elasticity**. *Proceedings of the 2011 International Conference on Advanced Robotics*, 2011.

[81] V. De Sapio, J. Warren, O. Khatib, S. Delp. **Simulating the task-level control of human motion: a methodology and framework for implementation**. *Springer-Verlag*, May 2005.

[82] Y. Hurmuzlu, F. Génot, B. Brogliato. **Modeling, stability and control of biped robots - a general framework**. *ELSEVIER, Automatica [Online: www.sciencedirect.com]*, 2004.

[83] Y. Nakamura, K. Yamane, Y. Fujita, I. Suzuki. **Somatosensory Computation for Man-Machine Interface From Motion-Capture Data and Musculoskeletal Human Model**. *IEEE Transactions on Robotics, Vol.21, No.1*, February 2005.

[84] Yoshihiko Nakamura, Katsu Yamane. **Efficient parallel dynamics computation of human figures**. *Proc. IEEE Conf. Robotics, Automation, Vol.1*, 2002.

[85] Z. Zhang, H. S. Seah, C. K. Quah, A. Ong, K. Jabbar. **A Multiple Camera System with Real-Time Volume Reconstruction for Articulated Skeleton Pose Tracking**. *Springer-Verlag Berlin Heidelberg*, 2011.