Bipedal Walking Trajectory Energy Minimization Through a Learned Hip Height Trajectory

A Thesis

Submitted to the Faculty

of

Drexel University

by

Sean Mason

in partial fulfillment of the

requirements for the degree

of

Master of Science in Mechanical Engineering and Mechanics

June 2012

Dedications

I dedicate this thesis to my mother, father, and sister. You have always provided me with unwavering love and support. I would also like to dedicate this work to my amazing girlfriend Nathalie Capati. Thank you so much for being by my side and supporting me through this entire journey. It is easy to take great leaps when you have a support group that will never let you fall.

Acknowledgements

First and foremost I wold like to thank Dr. Oh. You have been both academic advisor and a personal mentor to me. The experiences and guidance you have provided me with at Drexel have undoubtedly shaped the researcher I have become and my aspirations for the future. You have provided me with a once in a lifetime opportunity to both pursue the field I love and see the world. You have my highest respect and I cannot thank you enough.

To all my friends at DASL, I must say that it has been an incredible journey. Thank you all for the countless hours of help on my course work, lab work, and for being there till the very end of every project. You are all the hardest working and most down to earth lab members I have ever had the pleasure of meeting. Specifically, I would like to acknowledge Robert Ellenberg, Youngbum Jun, and Kiwon Sohn for their significant contribution made in helping me conduct this research.

Table of Contents

List of Tables

List of Figures

Abstract

Bipedal Walking Trajectory Energy Minimization Through a Learned Hip Height Trajectory
Sean Mason
Paul Oh, Ph.D.

This thesis describes methods used to optimize energy consumption of an offline bipedal walking trajectories through hip height control. The experiments were carried out on a miniature humanoid robot within the simulation environment Webots. Zero Moment Point (ZMP) preview control methods were implemented in Matlab to produce a stable walking trajectory for the robot with a fixed hip height. The hip height trajectory was then developed using an observation based Q-learning method that consider both stability and energy consumption. Through the Q-learning methods there was approximately a 9% decrease in the average energy consumption. Additionally, an increase in stability was observed.

Chapter 1: Introduction

1.1 Introduction

As the name implies, a humanoid robot is designed to mimic both the form and function of humans. Most robots are specially adapted by their designers to suit a particular application within a particular environment. Humans, on the other hand, have adapted their environment to suit their own form. As humans continue to expand, we have converted many existing environments into ones that are more favorable for our own function and comfort. Hallways are narrow and tall to suit the upright human locomotion, table tops are built to a height where visual referencing and manipulation is convenient, and stairs are designed to be easily traversed by humans connecting different floors. Each of these specialized environments is pre-adapted for the human form. A humanoid robot therefore can utilize these adaptations without needing tailored designs. A humanoid shape allows it to manipulate and interact with the countless tools and objects that humans use. Rather than equipping the robot with specialized tools for the task, robots can make use of tools already available on the work site and used by humans. Beyond the shape of the humanoid, another fundamentally distinguishing characteristic of the humanoid robot is bipedal locomotion. Bipedal locomotion is advantageous over many other forms of locomotion in that once stability is achieved it requires a relatively low amount of input energy, enables the robot to traverse rough and discontinuous terrain, and has a small footprint when compared to other forms of locomotion.

In the history of bipedal research, most researchers have spent their efforts on making bipedal robots walk faster or more stably. One area that has been less addressed is the area of energy efficiency. If humanoids are truly designed to work in human environments, this likely means that they will work indoors. When in confined spaces, locomotion speed becomes less important as higher speeds tends towards to greater changes in acceleration. This start and stop motion is not consistent with the locomotion displayed by humans indoors. Rather than looking into methods that optimize speed, energy optimization becomes much more a pertinent topic. Currently, humanoid robots have a typically charge life of less than 2 hours [6]. This is considerably low given that a typical human is expected to work an eight hour work shift. In order for a humanoid to remain untethered for a full work day, the batteries powering the robots must improve and measures must be taken to operate more efficiently.

1.2 Review of Literature

To address the topic of energy minimization, researches have taken a variety of approaches. These efforts can be broken down into a few fundamentally different categories. Section 1.2.1 discusses methods of optimizing walking gaits through exploiting natural dynamics, Section 1.2.2 introduces search methods that optimize energy using evolutionary and genetic search algorithms, and Section 1.2.3 discusses work done to optimize energy using reinforcement learning methods.Within these different approaches, there is substantial cross-over in the methodology for formulation and evaluation. The literature chosen for discussion touches on the most seminal and relevant publications addressing the biped energy minimization problem.

1.2.1 Natural Dynamics

Bipedal research that has exploited natural dynamics began with passive walkers. The original passive walkers were developed with the capability to walk down gentle slopes with a constant walking cycle and no additional input energy. Because passive walking gaits are "naturally" formed , meaning that walking gaits are tuned by changing physical characteristics of the robot, and require no input energy they are likely to either achieve or approach energy optimality.

Passive dynamic walking was pioneered by McGeer's experimental research conducted on 2D bipeds and their walking gaits [1, 7]. A 2D robot is mechanically constrained so that there is no movement in the sagittal plane. To ensure that the robot would not collapse in the sagittal plane, the robot was designed to have a redundant set of legs that allowed the robot to move much like a person on crutches. To ensure smooth ground contact, curved feet were used as shown in Figure 1.1.

McGeer's work paved the road for passive walker research by providing the step-to-step analysis of a 2D biped achieving steady walking. In addition, his research explored the effects of parameter variation for the robot design, including: scale, foot radius, leg inertia, hip mass, Center of Mass (COM) height, hip damping, hip mass offset, and leg mismatch.

Following McGeer, other researchers began to develop control laws for stability and energy associated with the system. Zhenze et. al [8], worked to develop a control law for passive walkers that tracked passive energy levels. In their efforts,

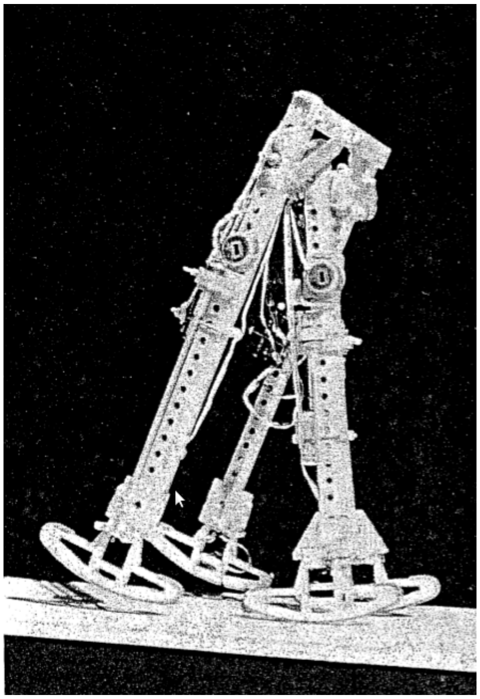


Figure 1.1: Realization of the 2D compass gait walker used by McGeer [1].

they defined the robot's "reference energy" as the characteristic energy of the passive limit cycle and that when the robot is driven towards the reference energy mobile balance is achieved. Furthermore, they developed two different control laws that allow them to more quickly converge to the reference energy: control with hip torque and control with ankle torque. As robots have evolved, metaphorically, they have become more complex and have incorporated more Degrees of Freedom (DOFs). While the fundamental concepts behind passive walkers are still relevant, the humanoids of higher DOF require a more extensive analysis on stability than the early 3 or 5 DOF robots.

While much research has moved away from the methods developed for the passive walker robots, Vanderborght et. al [9], was interested in retaining the energy efficient properties of natural dynamics. In 2007, his research aimed to preserve the versatility of an actively controlled humanoid while exploiting the known benefits of natural dynamics to reduce energy consumption. The proposed method to accomplish this on the pneumatically controlled biped known as "Lucy" is to fit the controllable stiffness of the actuators to the natural stiffness of the desired trajectory.

Diverging from the ideas previously expressed about "natural" walking gaits, other researchers have explored the straight, or stretched, legged walking gait observed by humans. Intuitively, walking with stretched legs, apposed to crouched, is more energy efficient because there is less torque on the knee and ankle joints. The cost of this improvement in energy conservation is a sacrifice in stability. As the distance from the ground to the COM of the robot increases the Zero Moment Point (ZMP) becomes more difficult to control. The reason for this is that when the legs become more stretched out some DOFs of motion are degenerated [2]. Researchers as Kyushu University proposed a straight leg walking controller that allowed for variable COM height that would help compensate for the decrease in stability.

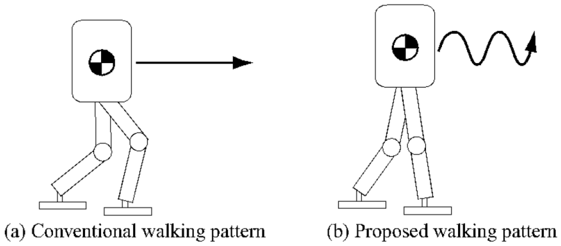

(a) Conventional walking pattern
(b) Proposed walking pattern

Figure 1.2: Proposed stretched legged walking gait by Kurazume [2].

1.2.2 Genetic Algorithm

Genetic algorithms (GA) consist of a global search procedure composed of reproduction, crossover, and mutation of chromosomes that aims to increase average fitness values that are defined by desirable traits. A chromosome contains all of the necessary information to define each of the desired parameters. After these three processes, the fitness function is applied to the current generation of chromosomes and the candidates for reproduction are selected. One major advantage of the genetic algorithm is that it has very little mathematical restrictions. Genetic algorithms have been used in robotics to accomplish a wide ranges of tasks that include optimizing stability, speed, similarities to human-like walking, and energy [10, 11, 12].

In 1999, researchers at Inha University used a GA to generate a leg trajectory that aimed to reduce peak velocity and accelerations [3]. In doing so, optimal via-points were found that when interpolated resulted in the leg trajectory. In this study, the fitness function, Equation 1.1, was defined to decrease the peak values of velocity and acceleration over the interval of one step.

$$f = 1/\sum((v_i + 1 - v_i)^2 + (a_i + 1 - a_i)^2) \tag{1.1}$$

where v_i and a_i represent the velocity and acceleration at a given interval. The method for this GA approach is shown in Figure 1.3.

Through numerical simulation and experimentation on the IWR-III system, it was observed that the GA had a damping effect on the robot allowed the robot to track the calculated ZMP more accurately.

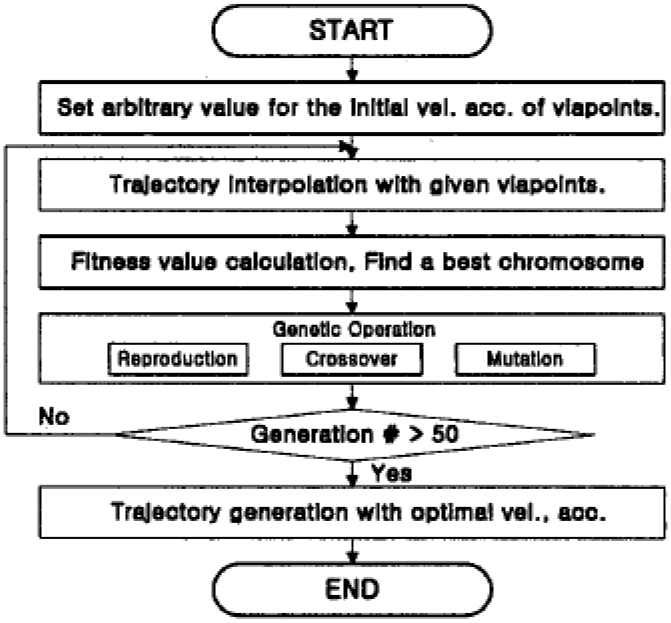

START
Set arbitrary value for the initial vel. acc. of viapoints.
Trajectory interpolation with given viapoints.
Fitness value calculation, Find a best chromosome
Genetic Operation
Reproduction
Crossover
Mutation
No
Generation # > 50
Yes
Trajectory generation with optimal vel., acc.
END

Figure 1.3: Flowchart for trajectory generation using via-points[3].

In 2004, researchers at Hanyang University used a GA to generate trajectories for each joint angle. In this case, each chromosome represents a coefficient of a 4th order polynomial that defines a trajectory for a given joint [13]. Rather than minimizing the differences in velocity and acceleration, the performance index to be minimized is a function of the power applied at each joint given by:

$$J = \frac{1}{2} * \int_{0}^{t_f} p^T Q p dt \tag{1.2}$$

where p is the power applied at each joint, t_f is the time for a step, and Q is a matrix comprised of weighting factors on control torque for the actuators. Using this criteria, walking gaits were optimized for walking on the ground, ascending stairs, and descending stairs using the computed torque controller.

Lastly, in 2008 researchers at the National University of Singapore used a GA to minimize torque within the robots joints while using the calculated ZMP as the stability criteria to see if the trajectory is physically realizable [14]. In total, seven key parameters were used that represent a set of redundant coefficients that are used to interpolate the 4^{th} order foot trajectory, and 7^{th} order hip trajectory. To simplify the problem, the hip height is constrained to be constant and the trunk is constrained to be upright. In this study, the cost function is given by:

$$P = \frac{1}{n} \int t_{s0} \tau^T \tau dt \tag{1.3}$$

where τ is the matrix of all the joint-torques, t_s it the time for one step, and n is the number of integration steps. From here, the fitness function evaluates stability, which will be discussed later, and is given by:

$$F = \begin{cases} \frac{1}{P}, & \text{if ZMP stays inside the stable region} \\ 0, & \text{else} \end{cases} \tag{1.4}$$

1.2.3 Reinforcement Learning

Compared to the other methods discussed previously, the use of machine learning methods is a relatively newer research area. Machine learning covers a wide range of approaches and algorithms that are typically formulated to work within Markov Decision Process (MDP) environments.

In 2004, researchers at the Robotics Institute of Carnegie Mellon University used model based reinforcement learning to learn the appropriate foot placement and the walking cycle timing for a 5 DOF walking robot [15]. The learned model consists of a Poincare map that directs the control actions based on a computed value function. The trajectories were represented by interpolating four via-points such that there was zero velocity and acceleration at each via point. The robot was rewarded for continuous walking and punished if the height of the robot drops below a set threshold, representing instability. The experiment was first simulated on a 3 and 5 link robot, and then finally on a 5 link bipedal robot that was fixed to a boom to constrict sagittal movement. After 80 trials, averaged over 5 experiments, a stable walking controller was acquired.

In 2007, the reinforcement method known as Q-learning was used by the Korea Advanced Institute of Science and Technology to develop a stable walking trajectory [16]. In this study, ZMP position was modeled after the inverted pendulum and a third order polynomial for the ankle and hip joint pattern was learned. The boundary conditions chosen for the third order walking pattern we as follows:

1. beginning angle of ankle joint
2. beginning velocity of the ankle joint
3. final position of foot (step length)
4. final velocity of the robot (hip position)

By learning the final velocity of the robot after a step, the walking pattern shape can be changed without changing the step size. Rewards were given based on the torso rotation angle with respect to the ground andif the robot did not fall. Stable walking was realized after 19 trials.

Recently (2011), researchers at National Cheng Kung University used reinforcement learning methods to generate bipedal walking trajectories [17]. In their research, they implemented policy gradient reinforcement learning to learn walking parameters to develop the fastest walking speed. In early tests, stability was not considered and the reward was only a function of the walking velocity. As the walking speed increased, so did the number of falls. To address this problem, a reward that considered the desired ZMP trajectory was introduced. This reward, as well as the velocity reward, were normalized and summed to create the total reward. Lagrange polynomial interpolation was used to generate the new motion trajectories.

1.3 Proposed Solution

As humanoid motion has evolved from the simple 2 or 4 DOF walkers into the advanced humanoids that we know today, over 30 DOF [18, 19, 20], it is essential to adapt energy minimization methods to more modern walking gaits. In order for the research conducted now to be relevant to future walking gaits, it must remain both general and adaptable. Machine learning lends itself well to this idea because it is not attached to mathematical constraints that analytical solutions posses and the methods can be integrated across different robotic platforms. Rather than tedious methods of tuning gains on a system, machine learning takes the system as it truly is and learns a desired trait based on observation or prediction. As mentioned in many of the previous works discussed, there are often simplifying assumptions or constraints imposed on the system to either reduce computational costs or allow a more simplistic model to be used. Many of these constraints may improve desired attributes, such as stability, but usually do not consider energy consumption. A common example of such a constraint is the fixed hip height constraint for many walking gaits. This constraint does give stability in the upper body, which can prove useful, but as seen in the motion capture data of a human waling gait, Figure 1.4, it is unnatural.

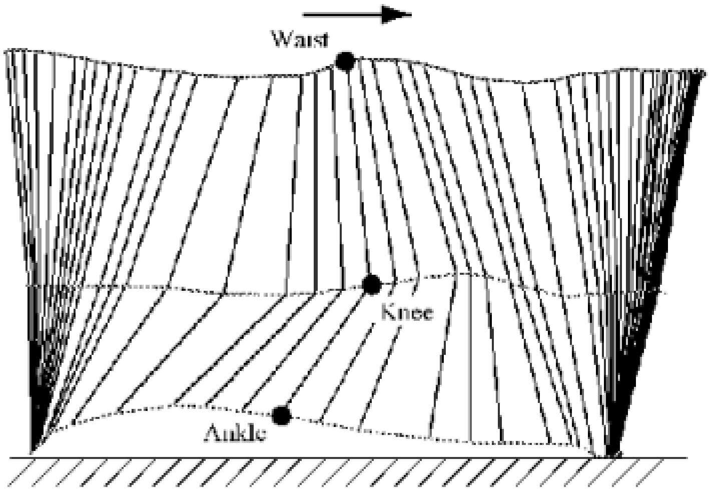


Figure 1.4: Up and down motion of the hip while walking [2].

In order to decrease energy consumption, I propose that Q-learning be applied to learn the hip height trajectory that optimizes energy consumption. In doing so, the stability provided by the fixed hip height trajectory generation will be considered. Because Q -learning is a observation based reinforcement learning algorithm, meaning that many tests must be conducted in order to converge to the optimal trajectory. In order to more rapidly test the learning method, the evaluation will be done on a 21 DOF miniature humanoid in a simulation environment. It is the author's hypothesis that through the proposed method, the optimal hip trajectory can be found that will further minimized energy consumption without destroying the stability provided by ZMP preview control.

Chapter 2: Simulation

2.1 Humanoid Robot Model

The robot used in these experiments was the miniature humanoid named miniHUBO designed by Dennis Hong at Virginia Polytechnic Institute and State University. It was designed as a scalable testing platform for the adult sized humanoid, HUBO [20]. This particular model was chosen because it reduced the complexity of the problem, with fewer DOFs, and has proven advantageous in other studies before testing on an adult sized humanoid [5, 21, 22]. The high level specifications of the robot are as follows:

Table 2.1: MiniHUBO Specifications

Height	46 cm
Weight	2.9 kg
DOF	22
Motors	Robotis Dynamixel (RX-10, RX-28, RX-64)

To create the model used in simulation, the robot was modeled in Autodesk Inventor. Each specific part was assigned mass values based off measurements taken from the actual robot. From the 3D CAD, properties such as mass, shape, size and moment of inertia were able to be exported for each moving part.

These properties were imported into the robotic simulator Webots. The model used in Webots consisted of prism shaped body parts that were bounded by the

Table 2.2: MiniHUBO Mass Properties

	H (m	W (m)	D (m)	Mass (Kg)
CHEST	0.10635	0.1297	0.056	0.366
SHOULDER - LF	0.025	0.034	0.044	0.009
SHOULDER - RT	0.025	0.034	0.044	0.009
BICEP	0.09965	0.0356	0.06125	0.140
ELBOW	0.034	0.044	0.025	0.009
FOREARM - LF	0.0506	0.0485	0.0356	0.065
FOREARM - RT	0.0506	0.0485	0.0356	0.065
WAIST	0.05585	0.1114	0.0631	0.285
HIP YAW	0.039	0.025	0.0865	0.015
HIP PITCH ROLL - LF	0.0506	0.0485	0.0905	0.158
HIP PITCH ROLL - RT	0.0506	0.0485	0.0905	0.158
THIGH	0.102	0.044	0.048	0.019
SHIN - LF	0.113837	0.0485	0.053538	0.100
SHIN - RT	0.113837	0.0485	0.053538	0.100
ANKLE PITCH ROLL - LF	0.0506	0.0485	0.0905	0.158
ANKLE PITCH ROLL - RT	0.0506	0.0485	0.0905	0.158
FOOT	0.041	0.064	0.11	0.048

Table 2.3: MiniHUBO Center of Mass Locations

	X (m)	Y (m)	Z (m)
CHEST	0.000000	0.007496	0.000613
SHOULDER - LF	0.008434	0.000000	0.000000
SHOULDER - RT	-0.008434	0.000000	0.000000
BICEP	0.000000	0.000767	0.002706
ELBOW	0.000000	0.008434	0.000000
FOREARM - LF	-0.001666	0.002207	0.000000
FOREARM - RT	0.001666	0.002207	0.000000
WAIST	0.000000	-0.000704	0.003727
HIP YAW	0.011966	0.000000	0.000113
HIP PITCH ROLL - LF	-0.001630	0.002940	0.001054
HIP PITCH ROLL - RT	-0.001630	0.002940	0.001054
THIGH	0.000000	-0.005221	0.006995
SHIN - LF	-0.001648	0.031707	0.000977
SHIN - RT	0.001648	0.031726	0.000894
ANKLE PITCH ROLL - LF	-0.001630	-0.002940	0.001054
ANKLE PITCH ROLL - RT	0.001630	-0.002940	0.001054
FOOT	-0.000170	-0.016098	-0.003432

Table 2.4: MiniHUBO Inertia Properties

	Ixx $(kgmm^2)$	Iyy $(kgmm^2)$	Izz $(kgmm^2)$	Ixy $(kgmm^2)$	Iyz $(kgmm^2)$	Izx $(kgmm^2)$
CHEST	0.377268	0.571481	0.761391	0.000000	-0.013663	0.000000
SHOULDER - LF	0.003225	0.004119	0.002129	0.000000	0.000000	0.000000
SHOULDER - RT	0.003225	0.004119	0.002129	0.000000	0.000000	0.000000
BICEP	0.128338	0.034871	0.112464	0.000000	-0.018017	0.000000
ELBOW	0.002129	0.003225	0.004119	0.000000	0.000000	0.000000
FOREARM - LF	0.015576	0.011951	0.019327	0.000000	0.000000	0.000000
FOREARM - RT	0.015576	0.011951	0.019327	0.000000	0.000000	0.000000
WAIST	0.106898	0.299527	0.273885	0.000001	0.000634	0.000001
HIP YAW	0.018457	0.015826	0.004771	0.000000	-0.000032	0.000000
HIP PITCH ROLL - LF	0.103932	0.091297	0.043876	0.000487	-0.003423	0.000349
HIP PITCH ROLL - RT	0.103932	0.091297	0.043876	-0.000487	-0.003423	-0.000349
THIGH	0.016916	0.011540	0.020715	0.000000	-0.000768	0.000000
SHIN - LF	0.160967	0.039660	0.159313	0.005246	0.002326	0.000414
SHIN - RT	0.160798	0.039507	0.159297	-0.005229	0.002243	-0.000381
ANKLE PITCH ROLL - LF	0.103932	0.091297	0.043876	-0.000487	0.003423	0.000349
ANKLE PITCH ROLL - RT	0.103932	0.091297	0.043876	0.000487	0.003423	-0.000349
FOOT	0.068493	0.072762	0.033967	-0.000161	-0.001749	-0.000009

maximum extents of each moving part in the real robot. This simplification was made to speed up the simulation process. Figure 2.1 shows the physical miniHUBO robot and the resulting model used in simulation.

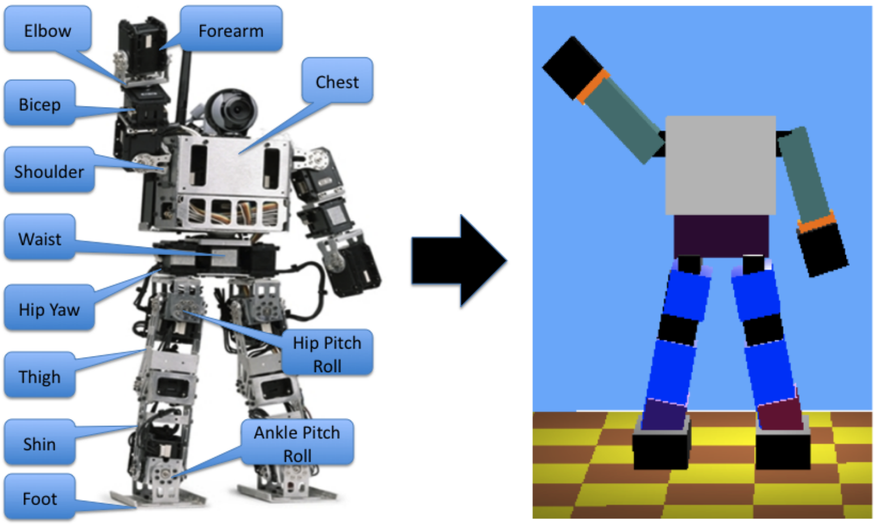
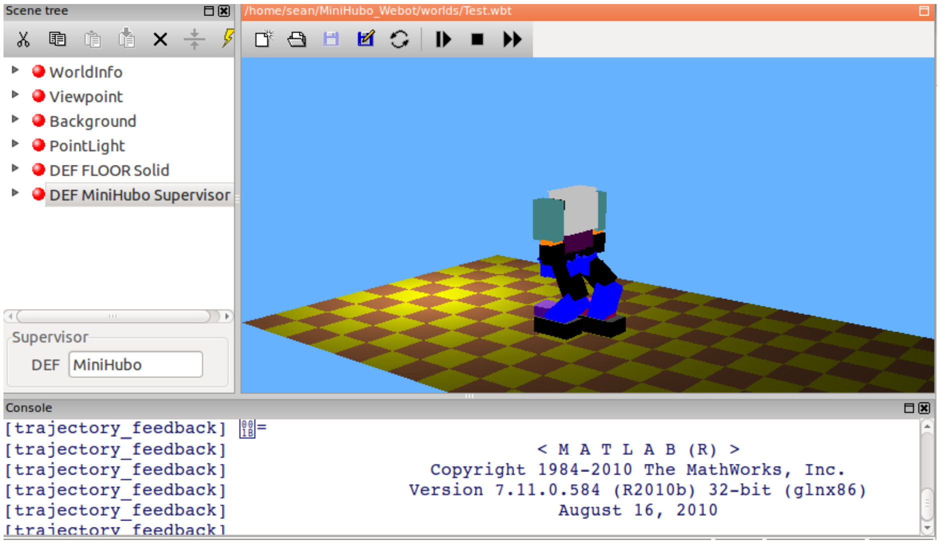


Figure 2.1: The physical miniHubo (right half body parts labeled) and the virtual model created in Webots that was used for used for simulation.

2.2 Simulation Environment and Physics Engine

Webots relies on Open Dynamics Engine (ODE) to perform the physics simulation of the experiment. Within Webots, robots and environments are modeled to interact with each other through the command code known as the controller. The controller specifies all the information about how the simulation will be executed. Using Webots, it is possible to obtain joint positions, velocities, and torques for rapid testing and evaluation. In this testing setup, the controller for the robot was written in MATLAB and was given supervisor permissions. Supervisor permissions allow the controller to access global environment information, which is useful for resetting the robot, scene, and physics engine in between tests.



Supervisor
DEF MiniHubo

```
[trajectory_feedback]  =
[trajectory_feedback]                  < M A T L A B (R) >
[trajectory_feedback]           Copyright 1984-2010 The MathWorks, Inc.
[trajectory_feedback]         Version 7.11.0.584 (R2010b) 32-bit (glnx86)
[trajectory_feedback]                       August 16, 2010
[trajectory_feedback]
```

Figure 2.2: Webots Guided User Interface (GUI) with modeled robot.

Chapter 3: Walking Trajectory Generation

3.1 Control Method

In the past, walking pattern generation fell into two fundamentally different categories: ZMP based pattern generation and the inverted pendulum approach [23].The ZMP approach is efficacious when an accurate model of the robot, including location of center-of-masses (COMs) and inertia for each link, is provided. The inverted pendulum approach requires much less information about the system, such as center of mass and total angular momentum. This system is simply treated as an inverted pendulum, and with a fast enough control loop the system can remain stable. In this paper, the author uses a method that falls in between these two categories known as a ZMP preview controller.This method of walking trajectory generation has been used successfully my many researchers to generate stable walking gaits. Specifically, the ZMP preview controller proposed has previously been successfully implemented on the miniHUBO robot [21, 5]

3.2 Derivation of ZMP Equation from 3D-LIPM

The Three-Dimensional Linear Inverted Pendulum Mode (3D-LIMP) [24] describes the dynamics of a inverted pendulum where the mass is constrained to move along an arbitrary defined plane. To fit this model, the robot is modeled as a point mass m located at length l at the humanoids COM. The dynamics given under the constraint

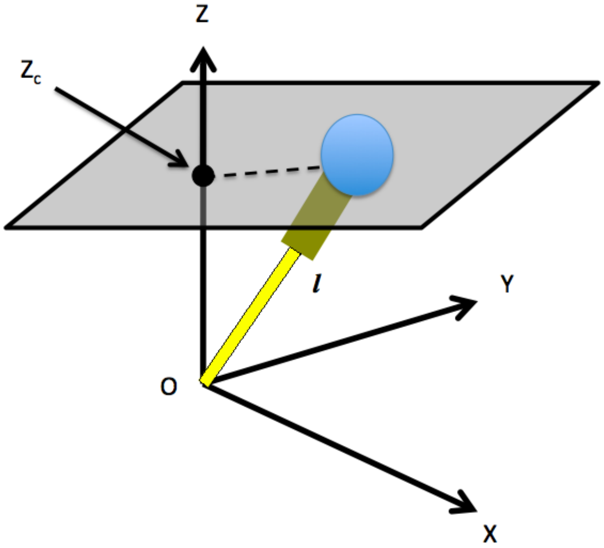


Figure 3.1: 3D inverted pendulum under constraint.

control is given by

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x \tag{3.1}$$

$$\ddot{x} = \frac{g}{z_c} x - \frac{1}{m z_c} \tau_y \tag{3.2}$$

and the constraint equation is:

$$\tau_x x + \tau_y y = 0 \tag{3.3}$$

where m is the mass of the pendulum, g is gravity acceleration and τ_x, τ_y are the torques around z-axis and y-axis respectively. The constraint plane, as shown in Figure 3.1 is given by the normal vector $(k_x, k_y, -1)$ where the intersection of z and

z_c is

$$z = k_x x + k_y y + z_c$$

The horizontal constraint imposed on the 3D-LIPM allows the sagittal and lateral motions to controlled separately and overall greatly simplifies the walking pattern generation. For a horizontal constraint "(kx = ky = 0)" the zero-moment point is easily calculated to be :

$$p_x = -\frac{\tau_y}{mg} \tag{3.4}$$

$$p_y = -\frac{\tau_x}{mg} \tag{3.5}$$

where point (p_x, p_y) is the projection of the ZMP on the ground. This point represents the position where the sum of the moments is equal to zero. For very slow statically stable systems, the ZMP is the projection of the COM. As the system moves faster, the dynamics of the bodies contribute to the ZMP and distort the shape.

Opposite to calculating the resulting ZMP from the system, walking pattern generation requires the systems motion to be calculated by a given ZMP trajectory. To generate the off-line reference trajectories, ZMP-Preview control is used [25]. Defining the control input u as the time derivative of the horizontal acceleration of the

COM, the ZMP equation is translated into state space as:

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{3.6}$$

$$y_{zmp} = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} \tag{3.7}$$

Using 3.6, it is possible to construct a control system that outputs the robots COM walking pattern based off of ZMP tracking control using preview control. To do this, first the system described by 3.6 is represented as a discrete system with sampling time T as:

$$Y(k+1) = AY(k) + Bu(k) \tag{3.8}$$

$$Y_{zmp}(k) = CY(k) \tag{3.9}$$

where

$$Y(k) = \begin{bmatrix} y(kT) & \dot{y}(kT) & \ddot{y}(kT) \end{bmatrix}^T$$

$$u(k) = u(kT)$$

$$Y_{zmp}(k) = Y_{zmp}(kT)$$

$$A = \begin{bmatrix} 1 & T & \frac{T^2}{T} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix}$$

The controller, simulated in MATLAB, takes a defined ZMP input and outputs the appropriate COM trajectory that a 3D-LIPM would need to follow to obtain the prescribed ZMP trajectory. To define the ZMP trajectory, the walking pattern cycle is broken down into discrete phases as shown in Figure 3.2 . If the robot were to start with a right step, the Single-Support Phase (SSP) occurs when the right leg is planted on the ground and the left leg is in transition between foot placements. When the left foot lands, the robot is in its Double-Support Phase (DSP) where the COM is shifted so that it can be supported by the initial foot. A robot is considered stable if the ZMP is located with the the stability polygon defined by the edges of the robots feet that are actively in contact with the ground. The progressive stability region for different foot placements is shown in Figure 3.3. Knowing this, the desired

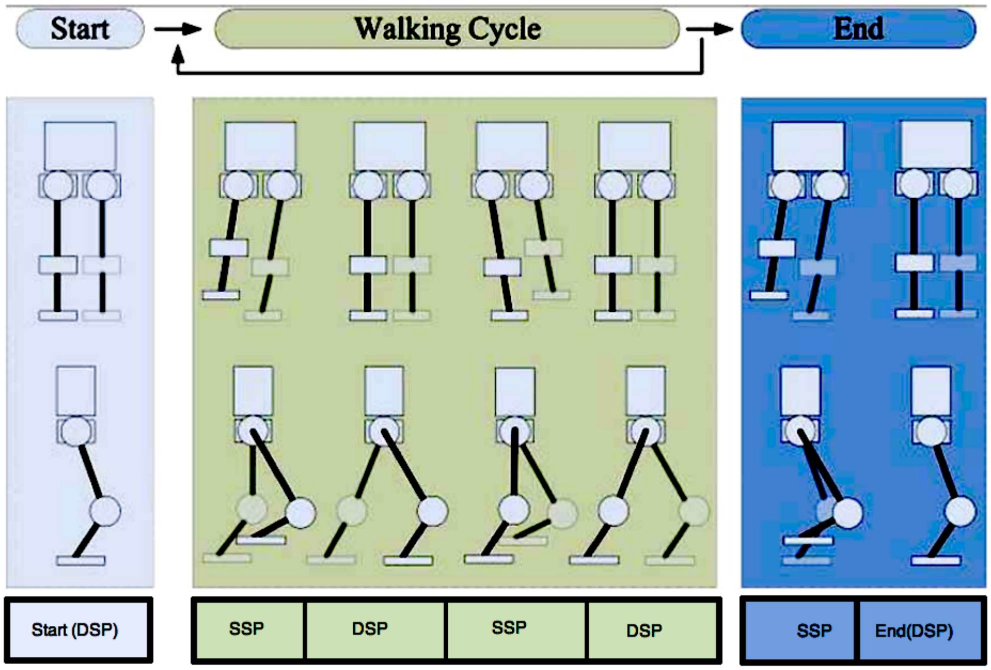


Figure 3.2: Phases of robot's walking cycle [4].

ZMP can easily be defined to traverse back and forth to the centroids of the planned footsteps during the DSP and dwell at the foot centroid during the SSP.

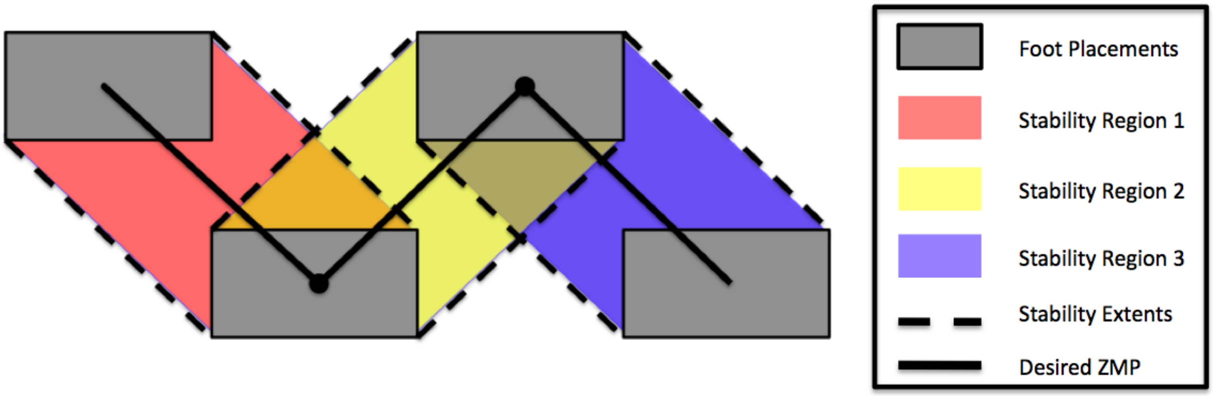


Figure 3.3: Stability region for given foot placements and the desired ZMP trajectory.

3.3 Foot Trajectory

Once, the COM trajectory is generated a complete walking pattern can be generated once the foot trajectory is defined. When defining a foot trajectory, a key component to energy loss is the impact force caused by the foot landing. To avoid this problem, may researchers have used cycloids to define the foot trajectory. A cycloid is generated by following the x-y position of a point on a rolling disk, Figure 3.4. Using this trajectory, the point on the disk, representing the foot trajectory, has an instantaneous velocity of zero when it contacts the ground.

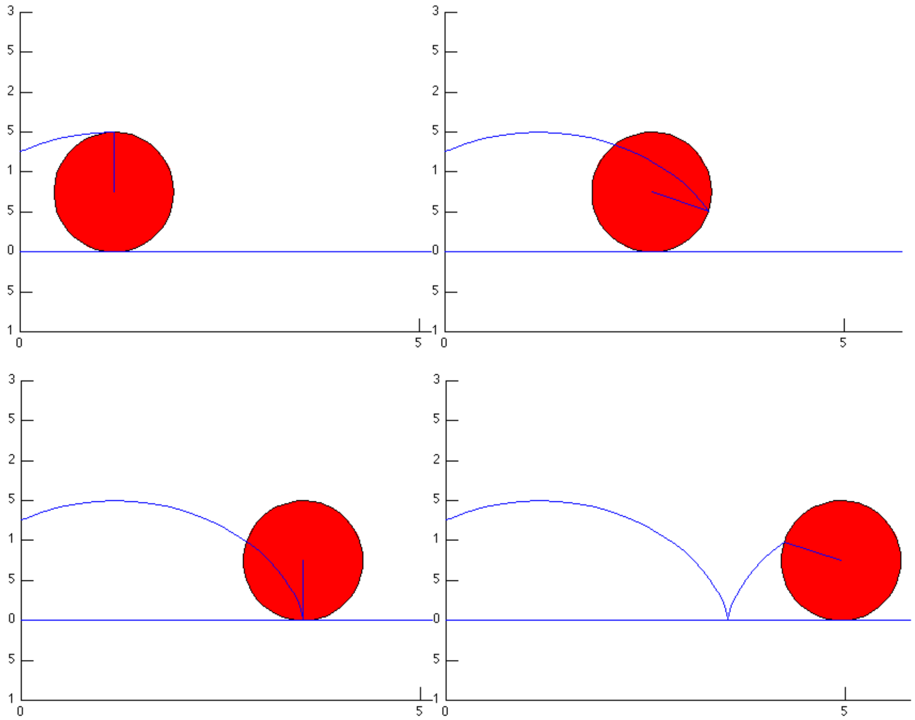


Figure 3.4: Generation of a cycloid trajectory using a rolling disk

To completely define the foot trajectory, the SSP, step distance, and max foot height are needed. Of these parameters, the first two are provided by the ZMP equations and the third is defined manually. In the case of this study, the value chosen for the max foot height is negligible because it will not significantly impact the robots stability. In addition, it is the main purpose to find an optimal trajectory for the hip motion, not for the foot trajectory. Using the three parameters, the equations for the foot trajectory for walking straight forwards are:

$$Foot_x = \frac{S_d}{2\pi}(2wt_{SSP} - sin(2wt_{SSP})) \tag{3.10}$$

$$\begin{aligned} Foot_z &= \tfrac{H}{2\pi}(2wt - sin(2wt)), && \text{for } 0 \leq t < \tfrac{1}{2}SSP \\ Foot_z &= 2H + \tfrac{H}{2\pi}(sin(2wt) - 2wt), && \text{for } \tfrac{1}{2}SSP \leq t < SSP \end{aligned} \tag{3.11}$$

$$Foot_y = constant \tag{3.12}$$

where w is defined by $2\pi f$, f is $\frac{1}{t_{SSP}}$, t_{SSP} is the duration of the SSP, S_d is the step distance, and H is the maximum foot height.

3.4 Inverse Kinematics

With the COM and feet trajectories generated, Inverse Kinematics (IK) can be used to solve for each individual joint angle of the robot. In this situation, the foot is treated as the origin, because it is grounded, and the hip is treated as the end effector, because we wish to control it's relative position. By constraining the yaw of the robot to be zero, [5] was able to constrain the system so that the remaining angles in each individual leg, three pitch angles and two roll angles, were able to be analytically calculated in vector space. The constraints on the system are:

1. Angular momentum of hip yaw = 0

2. Upper body remains perpendicular to the ground

3. Foot plane remains parallel to the ground

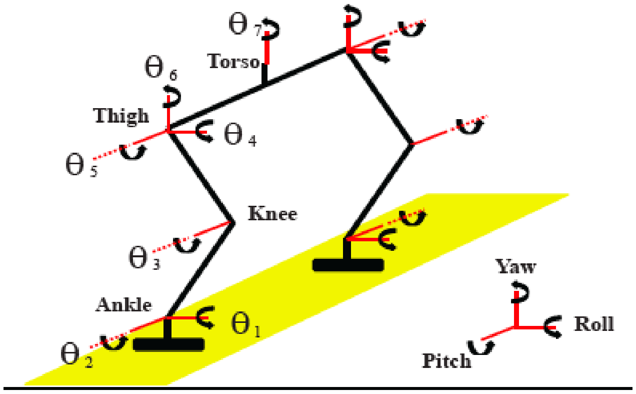


Figure 3.5: Joint configuration of the robot from an isometric view [5].

Using trigonometry, θ_1 and θ_4 are determined to be:

$$\theta_1 = 90 - cos^{-1}((Y_t - Y_a)/|\vec{l_{leg}}|) \tag{3.13}$$

$$\theta_4 = -\theta_1 \tag{3.14}$$

where

$$|\vec{l_{leg}}| = \sqrt{(Z_c - Z_a)^2 + (Y_t - Y_a)^2}$$

Then by using the law of cosines:

$$\theta_3 = 180 - cos^{-1}((l_{thigh}^2 + l_{shin}^2 - \vec{l_{leg}}^2)/(2l_{thigh}l_{shin})) \tag{3.15}$$

$$\theta_5 = 90 - \theta - cos^{-1}((l_{thigh}^2 + \vec{l_{leg}}^2 - l_{shin}^2)/(2l_{thigh}|\vec{l_{leg}}|)) \tag{3.16}$$

$$\theta_2 = 90 - \theta + cos^{-1}((l_{shin}^2 + \vec{l_{leg}}^2 - l_{thigh}^2)/(2l_{shin}|\vec{l_{leg}}|)) \tag{3.17}$$

where $|\vec{l_{leg}}| = \sqrt{(Z_t - Z_a)^2 + (Y_t - Y_a)^2 + (X_t - X_a)^2}$

$$\theta = cos^{-1}((X_t - X_a)/|\vec{l_{leg}}|)$$

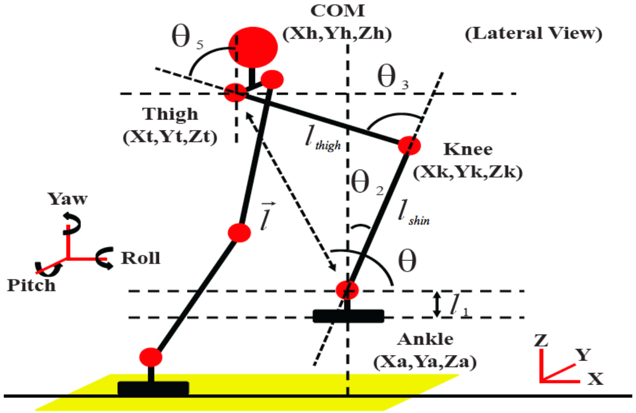


Figure 3.6: Joint configuration of the robot from a lateral view [5].

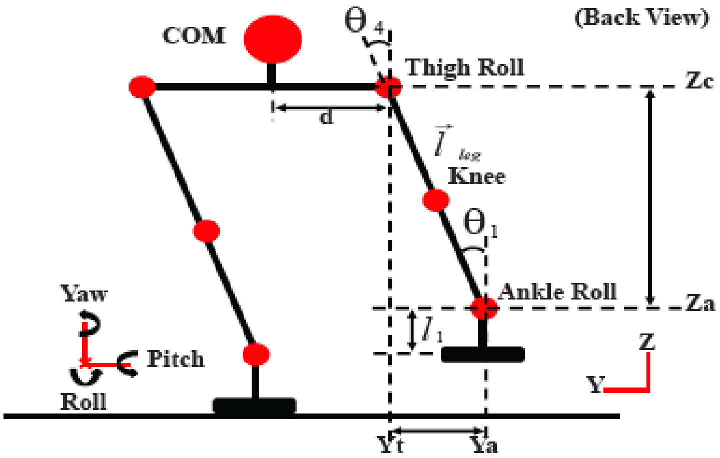


Figure 3.7: Joint configuration of the robot from a back view [5].

Chapter 4: Machine Learning

4.1 Q-Learning

To develop the optimal hip trajectory that will minimize energy comsumption, a Q-learning algorithm was used as the reinforcement agent. Figure 4.2, shows the architecture of the learning system. As shown, random trajectories for the hip z motion are input into the system. The random trajectories are created by defining the number of available neighbors N, the discrete height increment Δh, and the time step Δt. Because of this discretization, states and actions can be broken down into a directed node tree that represents the space for hip heights as shown in Figure 4.1 .

These trajectories are then used to create the full walking trajectory for the robot using the methods described in Chapter 3. Next, trajectories are executed in simulation and the values for torque and velocity are recorded for each time step. Energy

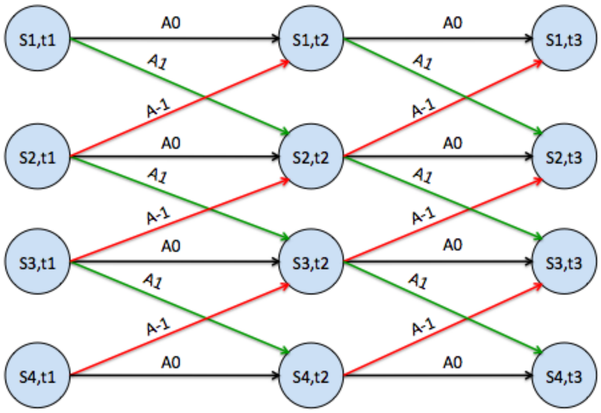


Figure 4.1: Example of the time dependent node tree for N = 1.

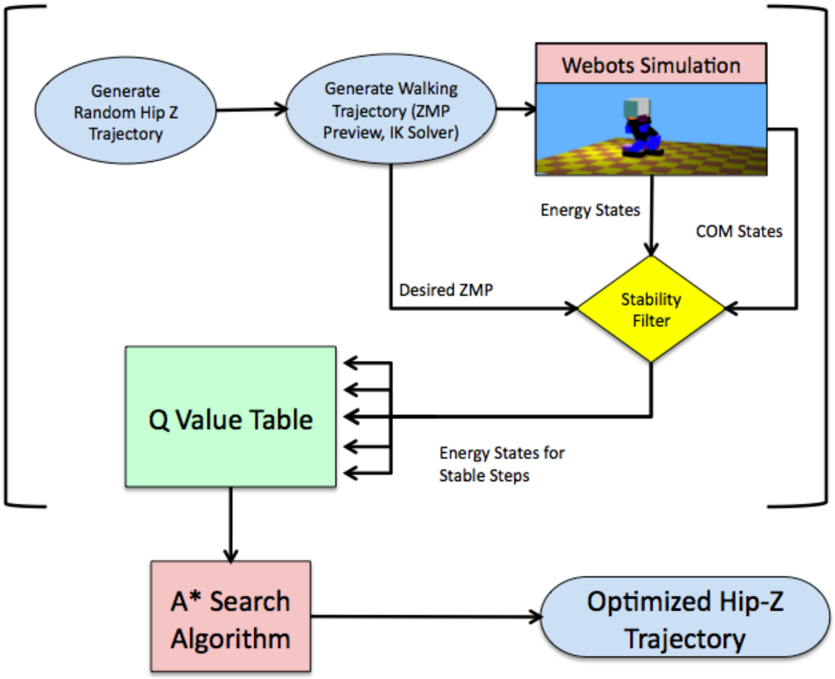

Generate Random Hip Z Trajectory
Generate Walking Trajectory (ZMP Preview, IK Solver)
Webots Simulation
Energy States
COM States
Desired ZMP
Stability Filter
Q Value Table
Energy States for Stable Steps
A* Search Algorithm
Optimized Hip-Z Trajectory

Figure 4.2: The proposed machine learning process.

consumption at each joint is approximated by the equation:

$$E_i = T_i v_i \Delta t \tag{4.1}$$

Where i is the joint, T is the torque measured, and v is the current velocity. The Q -table is organized to represent the "value" of all possible state-action pairs for each time step in the simulation. The break down of state-action pairs into the Q-table is shown in Table 4.1.

	Time Step 1	Time Step 2	$\cdots$	Time Step N
State 1	$Q(s_1, a_0)$	$Q(s_1, a_0)$	$\cdots$	$Q(s_1, a_0)$
	$Q(s_1, a_1)$	$Q(s_1, a_1)$	$\cdots$	$Q(s_1, a_1)$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$Q(s_1, a_N)$	$Q(s_1, a_N)$	$\cdots$	$Q(s_1, a_N)$
State 2	$Q(s_2, a_{-N})$	$Q(s_2, a_{-N})$	$\cdots$	$Q(s_2, a_{-N})$
	$Q(s_2, a_{-N+1})$	$Q(s_2, a_{-N+1})$	$\cdots$	$Q(s_2, a_{-N+1})$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$Q(s_2, a_0)$	$Q(s_2, a_0)$	$\cdots$	$Q(s_2, a_0)$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$Q(s_2, a_{N-1})$	$Q(s_2, a_{N-1})$	$\cdots$	$Q(s_2, a_{N-1})$
	$Q(s_2, a_N)$	$Q(s_2, a_N)$	$\cdots$	$Q(s_2, a_N)$
State N	$Q(s_N, a_{-N})$	$Q(s_N, a_{-N})$	$\cdots$	$Q(s_N, a_{-N})$
	$Q(s_N, a_{-N+1})$	$Q(s_N, a_{-N+1})$	$\cdots$	$Q(s_N, a_{-N+1})$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$Q(s_N, a_0)$	$Q(s_N, a_0)$	$\cdots$	$Q(s_N, a_0)$

To reduce the computation cost of learning, the space was discretized and the amount of neighbors for each state was fixed. The values for the number of neighbors and the interval between neighbors was determined based on the max velocity of the motor and the value for the time step was determined experimentally. Equation 4.2, shows how values in the Q-table were updated.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(s_t, a_t) * [P_{t+1} + \gamma * argmaxQ(s_{t+1}, a_{t+1})] \tag{4.2}$$

Where at each time step t, there are multiple states, s_t, which have a set of actions, a_t, available for each state. α is the learning rate, γ is the discount factor for the maximum future Q value, and P is the penalty value. The penalty value was formed by a weighted summation among the energy consumed at each joint and the 2-norm of the planned COM trajectory and the observed trajectory. To make the values of the two penalties comparable, both were normalized by a maximum acceptable value that was determined experimentally. For the energy penalty this was chosen to be the max value observed during an initial sweep of the hip height space where hip height was kept constant. For the ZMP penalty, this was chosen to be a value where instability that lead to falling was observed. While the energy penalty was left continuous, the ZMP penalty was converted to a binary value so the penalty would only be applied if the robot fell.

$$P = w1 * \|COM_{planned} - COM_{observed}\| + w2 * \sum_{i=1}^{n} E_i \tag{4.3}$$

where n is the number of joints in the robot.

4.2 Optimal Path Search - A*

Because walking trajectories are periodic, data from each individual step was used to update the Q-table. Once the Q-table has reached convergence, the best path is extracted from the Q table using the search algorithm A*. A* is a widely used "best first" search algorithm that combines knowledge of a current distance, where in this case the distance represents the summation of the chosen values in the Q table, and a heuristic that estimates future choices. Figure 4.3 shows an example surface that represents a Q matrix passed to A*. With a specific height chosen as a start and end point, A* finds the path that minimizes the total travel cost. To ensure a periodic walking trajectory, A* starts at each discrete start height and finds the penalty minimizing path that ends at the same height. Each trajectory from the available start heights are compared and the best global path is outputted. The resulting solution, for the given starting and end points, is shown in Figure 4.3.

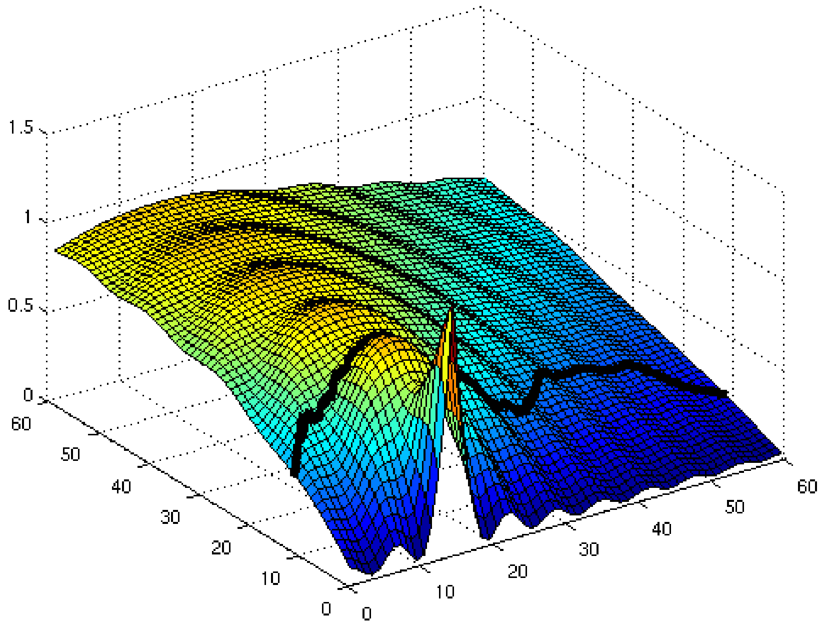


Figure 4.3: The local optimal path found by A* for the given example surface.

Chapter 5: Results and Discussion

5.1 Preliminary Testing

Using the ZMP preview and the proposed IK solver, a stable walking trajectory was able to be generated in MATLAB and verified in Webots. The walking parameters, Table 5.1, were manually tuned for stability within the webots environment.

Table 5.1: Walking Parameters

SSP	1.8 (s)
DSP	0.3 (s)
Lateral Distance	65 (mm)
Step Distance	80 (mm)
Step Height	50 (mm)

Using the methods outlined in Chapter 3 resulting stick figure diagram, Figure 5.1 and profiler of the generated walking pattern, Figure 5.2.

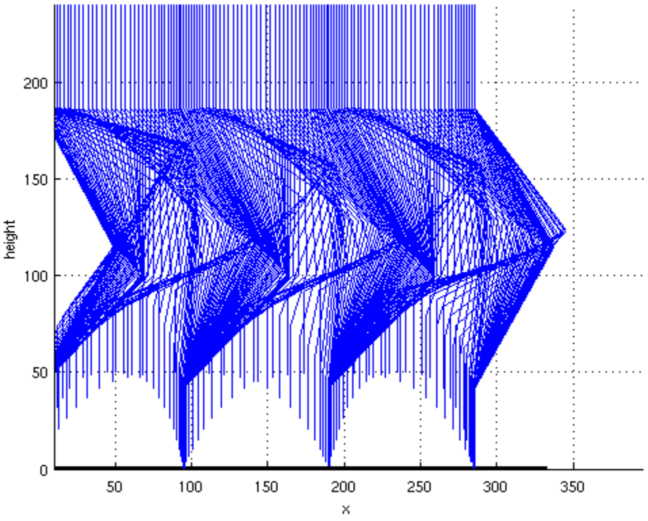


Figure 5.1: Stick figure profile of walking trajectory generated in Matlab.

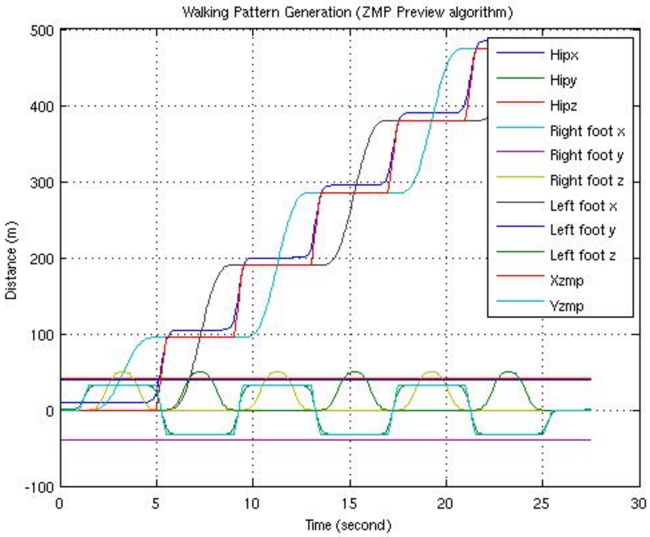


Figure 5.2: Profiler of all trajectories generated to create the full walking trajectory in Matlab.

As a baseline test, the energy and stability for each walking trajectory was evaluated at constant hip height that spanned the space being considered. The minimum and maximum hip height values used are 240 mm and 270 mm respectively. Figures 5.3 shows the normalized resulting values.

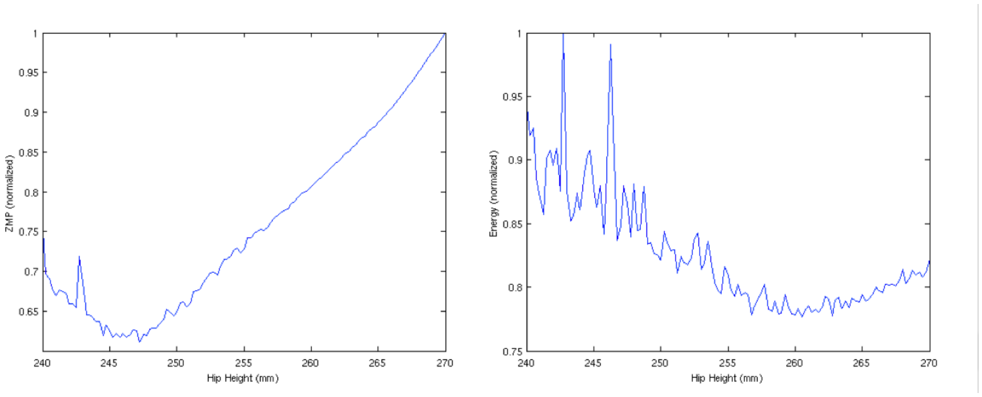


Figure 5.3: Normalized data for stability and energy consumption for walking trajectories with fixed hip height.

From the preliminary tests, the data shows that there are two general trends.

First, the energy consumed decreases as the hip height increases until 262 mm where energy consumption begins to increase. Second, the stability of the walking pattern is maximized early at 247 mm and then becomes increasingly more unstable.

5.2 Q-learning Results

The following gains were chosen for the Q learning process:

Table 5.2: Q-Learning Parameters

α (learning rate)	0.7
γ (discount rate)	0.3
$p1$	1.0
$p2$	0.2

The resulting Q table evolution is shown in Figure 5.4.

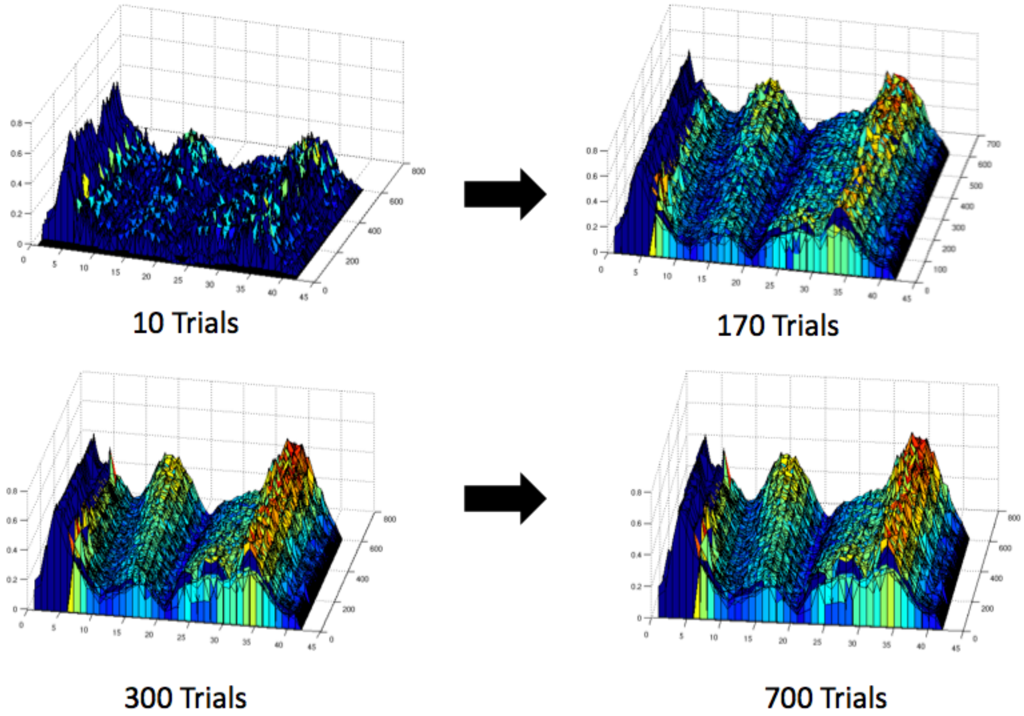


Figure 5.4: Evolution of the Q-table throughout the learning process.

On average the the Q table converged after approximately 600-700 tests where the robot was required to take 10 steps per test. The convergence is shown in Figure 5.5, where the error displayed is the error between consecutive Q tables.

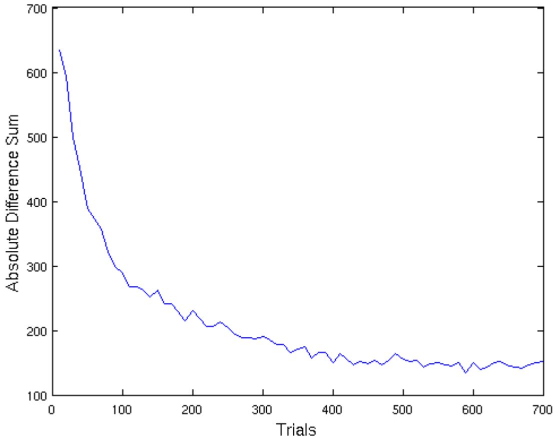


Figure 5.5: Convergence of the Q-table.

A more detailed version of the final Q -table is shown in Figure 5.6. After the Q table reached convergence, the following trajectories were extracted by using A*, where the best is shown in bold, Figure 5.7.

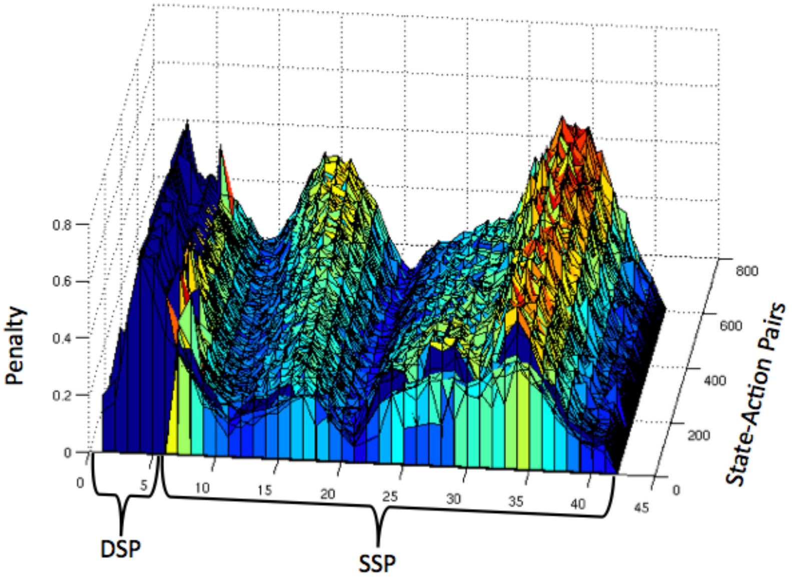


Figure 5.6: Labeled surface plot of the final Q-table after convergence.

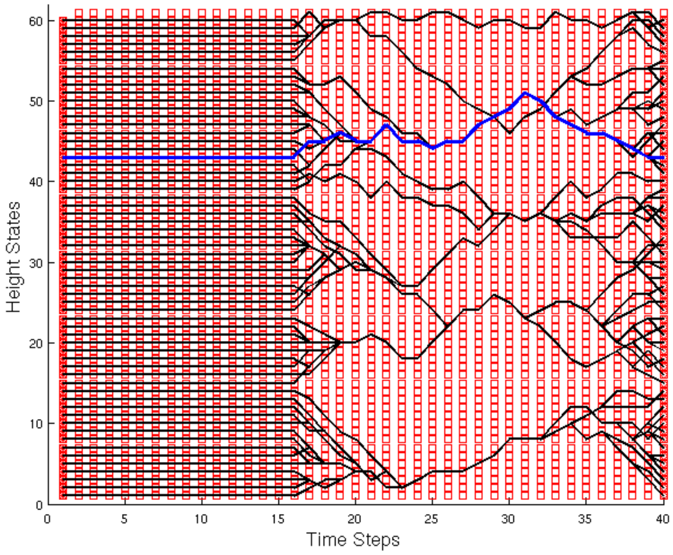


Figure 5.7: 2D plot of the extracted trajectories for each starting hip height.

This trajectory was then used in Webots and the energy consumption was then compared to the preliminary testing data. The resulting comparisions for energy consumption and stability are shown in Figure 5.8.

Overall, the final changes in energy that the learned hip height trajectory contributed are:

Table 5.3: Final Energy Results

	percent increase
min	3.106%
max	24.750%
average	8.953%

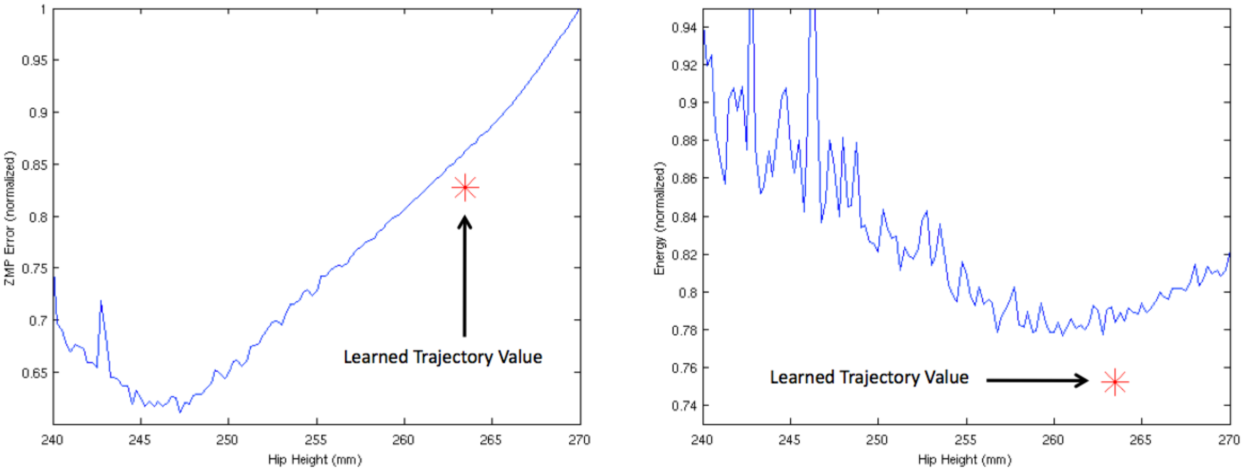


Figure 5.8: Values for stability and energy from the learned hip height trajectory.

Chapter 6: Conclusion

From this study, the goal was to develop the optimal hip height trajectory that when combined with the walking trajectories provided by ZMP preview control would reduce the total energy consumption. To accomplish this Q-learning methods were applied where input trajectories were randomly generated and observations were made in the simulation environment Webots. The proposed learned hip height trajectory should increase energy efficiency without destroying the stability provided by ZMP preview control. This resulting trajectory both increased energy efficiency of the average walking trajectory by approximately 9% and also provided a more stable walk at the given hip height than if it were fixed. Beyond this success, this study provided insight on how machine learning methods can be used as a post processing tool to further optimize given traits that had been previously constrained.

Bibliography

[1] Tad McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[2] R. Kurazume, S. Tanaka, M. Yamashita, T. Hasegawa, and K. Yoneda. Straight legged walking of a biped robot. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 337 – 343, aug. 2005.

[3] Sang-Ho Choi, Young-Ha Choi, and Jin-Geol Kim. Optimal walking trajectory generation for a biped robot using genetic algorithm. In *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 3, pages 1456 –1461 vol.3, 1999.

[4] T.S. Li, Yu-Te Su, Shao-Wei Lai, and Jhen-Jia Hu. Walking motion generation, synthesis, and control for biped robot by using pgrl, lpi, and fuzzy logic. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(3):736 – 748, june 2011.

[5] Youngbum Jun and Paul Oh. Humanoid stair climbing with direct hip control. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IASTED2011)*, October 2011.

[6] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi. Humanoid robot hrp-3. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2471 –2478, sept. 2008.

[7] T. McGeer. Passive walking with knees. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1640 –1645 vol.3, may 1990.

[8] Liu Zhenze, Zhang Peijie, Tian Yantao, and Zhou Changjiu. The passive energy tracking control law of the compass bipedal robot. In *Control Conference, 2007. CCC 2007. Chinese*, pages 457 –462, 26 2007-june 31 2007.

[9] B. Vanderborght, M. Van Damme, R. Van Ham, P. Beyl, and D. Lefeber. A strategy to combine active trajectory control with the exploitation of the natural dynamics to reduce energy consumption for bipedal robots. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 7 –12, 29 2007-dec. 1 2007.

[10] F. Yamasaki, K. Endo, H. Kitano, and M. Asada. Acquisition of humanoid walking motion using genetic algorithm-considering characteristics of servo modules. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 3123 –3128, 2002.

[11] T. Arakawa and T. Fukuda. Natural motion trajectory generation of biped locomotion robot using genetic algorithm through energy optimization. In *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, volume 2, pages 1495 –1500 vol.2, oct 1996.

[12] Y. Ogura, K. Shimomura, A. Kondo, A. Morishima, T. Okubo, S. Momoki, Hun ok Lim, and A. Takanishi. Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3976 –3981, oct. 2006.

[13] Kweon Soo Jeon, Ohung Kwon, and Jong Hyeon Park. Optimal trajectory generation for a biped robot walking a staircase based on genetic algorithms. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2837 – 2842 vol.3, sept.-2 oct. 2004.

[14] D. Van-Huan, Chee-Meng Chew, and A.-N. Poo. Optimized joint-torques trajectory planning for bipedal walking robots. In *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, pages 1142 –1147, sept. 2008.

[15] J. Morimoto, G. Cheng, C.G. Atkeson, and G. Zeglin. A simple reinforcement learning algorithm for biped walking. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, pages 3030 – 3035 Vol.3, april-1 may 2004.

[16] Jungho Lee and Jun Ho Oh. Biped walking pattern generation using reinforcement learning. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 416 –421, 29 2007-dec. 1 2007.

[17] P. Kormushev, B. Ugurlu, S. Calinon, N.G. Tsagarakis, and D.G. Caldwell. Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 318 –324, sept. 2011.

[18] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira. Humanoid robot hrp-4 - humanoid robotics platform with lightweight and slim body. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4400 –4407, sept. 2011.

[19] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent asimo: system overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478 – 2483 vol.3, 2002.

[20] Ill-Woo Park, Jung-Yup Kim, Jungho Lee, and Jun-Ho Oh. Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 321 –326, dec. 2005.

[21] Youngbum Jun, R. Ellenberg, and P. Oh. Realization of miniature humanoid for obstacle avoidance with real-time zmp preview control used for full-sized humanoid. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 46 –51, dec. 2010.

[22] R. Ellenberg, D. Grunberg, P.Y. Oh, and Youngmoo Kim. Using miniature humanoids as surrogate research platforms. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 175 –180, dec. 2009.

[23] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1620 – 1626 vol.2, sept. 2003.

[24] S. Kajita, O. Matsumoto, and M. Saigo. Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2299 – 2306 vol.3, 2001.

[25] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239 –246 vol.1, 2001.