BIPED LOCOMOTION CONTROL VIA HYBRID POSITION CONTROL AND GRAVITY COMPENSATION MODES

by

OZAN AYHAN

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of Master of Science

Sabancı University
June 2004

BIPED LOCOMOTION CONTROL VIA HYBRID POSITION CONTROL AND GRAVITY COMPENSATION MODES

APPROVED BY:

Assist. Prof. Dr. Kemalettin Erbatur …………………………
(Thesis Advisor)

Prof. Dr. Asif Şabanovic …………………………

Assist. Prof. Dr. Mahmut Akşit …………………………

Assist. Prof. Dr. Selim Balcısoy …………………………

Assist. Prof. Dr. Mustafa Ünel …………………………

DATE OF APPROVAL: …………………………

ABSTRACT

Past three decades witnessed a growing interest in biped walking robots because of their advantageous use in the human environment. However, their control is challenging because of their many degrees of freedom and non-linearities in their dynamics. Various trajectory generation and walking control approaches ranging from open loop walking to systems with many sensors and feedback loops have been reported in the literature. The tuning of the parameters of reference gait is a common complication encountered. Another important problem of the walking control is the contact transients at the instant of landing.

The method presented in this thesis generated position references for the upper body. Optimization techniques are employed to obtain suitable leg joint torques for the supporting leg to track body reference trajectories. Locomotion is achieved by a swinging leg control scheme, which is based on position control on certain directions, and artificial gravity compensation on certain directions. Soft landing and low impedance problem of the legs can easily be handled with this scheme. Another advantage is that limited number of parameters is required for gait generation.

3D dynamics and ground interaction simulation techniques are employed for a 12-DOF biped robot to test the proposed method. The simulations indicate the applicability of the method in real implementations.

ÖZET

İnsanların bulunduğu ortamlardaki olası faydaları dolayısıyla geçtiğimiz otuz yıl, insansı robotlara karşı giderek artan bir ilgi ve meraka şahitlik etti. Ne var ki, insansı robotların dinamiklerinin doğrusal olmaması ve birçok serbestlik derecesine sahip olmaları denetimlerini güçleştirmektedir. Açık devre yürüyüşten birçok algılayıcı ve geri besleme çemberine sahip sistemlere kadar sayısız güzergah belirleme ve yürüyüş denetim yaklaşımı önerilmiştir. Yürüyüş referansının parametrelerinin ayarlanması karşılaşılan genel bir problemdir. Bir diğer problem de ayağın yere konma anında yaşanan geçici durum davranışlarıdır.

Bu tezde önerilen metot üst vücut için çevrim-dışı güzergah referansları vermektedir. Bu referanslara karşı gelen destek ayağında oluşması gereken eklem momentlerinin elde edilmesi amacıyla eniyileme teknikleri kullanılmıştır. Hareket, belli yönlerde pozisyon denetimine, belli yönlerde ise sanal ağırlık beslemesine dayanan salınım ayağı denetim stratejisiyle sağlanır. Bu metotla ayağın yere yumuşak konması sağlanabilir ve düşük empedans probleminin kolaylıkla üstesinden gelinebilir. Bu metodun bir diğer avantajı da yürüyüşün sağlanması için sınırlı sayıda parametrenin gerekmesidir.

Önerilen metodun test edilmesi için 12 serbestlik derecesine sahip bir insansı robotun üç boyutlu dinamik ve yerle etkileşim benzeşim teknikleri kullanılmıştır. Benzeşimler, bu metodun gerçek uygulamalarda kullanılabilirliğini ortaya koymaktadır.

In the loving memory of my mother Fatma Ayhan,
whose loss diminished us all…

ACKNOWLEDGMENTS

I express my deepest regards and appreciation to Dr. Kemalettin Erbatur, whose invaluable support made realization of my thesis possible.

I extend my appreciation to graduate students in the Mechatronics Program, for their nice talk and company, and being tea mates to me.

I would like to thank to my mother, Fatma Ayhan for sharing her limitless love and grace with us, and to my father, Ahmet Ayhan, who patiently tried to answer my questions, even the most pathological ones, when I was a child.

Finally, to my older brother, Bora, for being at my side, when nobody was.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	:	Constraint matrix
${}^{x}\boldsymbol{A}_{y}$	:	Rotation matrix
$\boldsymbol{A}_{B}$	:	Base-link attitude matrix
$\boldsymbol{b}$	:	Bias vector consisting of gravitational, centrifugal and Corioli's forces
$\tilde{\boldsymbol{b}}$	:	Bias vector corresponding to body dynamics
$\hat{\boldsymbol{b}}$	:	Artificial bias vector for swing leg
$\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})$	:	Centrifugal and Corioli's force matrix
$\boldsymbol{c}$	:	Contact point position vector
D	:	Viscous damping coefficient
d	:	Scalar off-set for foot position reference in sagittal plane
e	:	Error vector position and orientation
$\boldsymbol{F}$	:	Total force
$\boldsymbol{f}_{E}$	:	External force vector
$\boldsymbol{g}(\boldsymbol{x})$	:	Gravity vector
g_{i}	:	Artificial gravity constant
$\boldsymbol{H}(\boldsymbol{x})$	:	Inertia matrix
$\boldsymbol{H}_{ij}$	:	Sub-matrices of inertia matrix
I	:	Index numbers
J	:	Index numbers
$\boldsymbol{K}$	:	External force to generalized force transformation matrix
$\boldsymbol{K}'$	:	Matrix relating external forces to body reaction force and torques
$\boldsymbol{K}''$	:	Matrix relating external forces to joint reaction force and torques

$\boldsymbol{K}_{fc}$	:	Matrix relating foot center velocities and body coordinate center velocities
K_0	:	Derivative gain matrix describing body dynamics
K_1	:	Proportional gain matrix describing body dynamics
K_d	:	PD controller derivative gain matrix for joint position control
K_p	:	PD controller proportional gain matrix for joint position control
M	:	Number of time-variant active contact points
M_A	:	Index numbers of active contact points
$\boldsymbol{M}$	:	Index numbers of external forces
N	:	Number of joints of the robot
$\boldsymbol{N}$	:	Total moment
$\boldsymbol{p}$	:	Axes origin position vector
$\dot{\boldsymbol{p}}$	:	Axes origin velocity vector
$\ddot{\boldsymbol{p}}$	:	Axes origin acceleration vector
$\boldsymbol{p}_B$	:	Base-link position vector
R	:	Rotation matrix
r	:	Body linear and angular velocity
$\boldsymbol{r}$	:	Center of mass position
$\dot{\boldsymbol{r}}$	:	Center of mass velocity
$\ddot{\boldsymbol{r}}$	:	Center of mass acceleration
S	:	Selection matrix for position controlled and gravity compensated modes
s_i	:	Entry of the selection matrix
$\boldsymbol{S}$	:	Center of mass position vector with respect to coordinate system
$\boldsymbol{u}$	:	Generalized force vector
$\boldsymbol{u}_E$	:	Generalized force vector generated by external forces
v	:	Reference acceleration for the body in inverse dynamics
$\boldsymbol{v}$	:	Generalized velocities vector
$\dot{\boldsymbol{v}}$	:	Generalized accelerations vector
$\boldsymbol{v}_B$	:	Base-link velocity vector
$\dot{\boldsymbol{v}}_B$	:	Base-link acceleration vector

$\boldsymbol{w}$	:	Joint angular velocity vector
$\dot{\boldsymbol{w}}$	:	Joint angular acceleration vector
$\boldsymbol{w}_B$	:	Base-link angular velocity vector
$\dot{\boldsymbol{w}}_B$	:	Base-link angular acceleration vector
$\boldsymbol{x}$	:	Generalized coordinate vector
x_i	:	i^{th} input variable
$\boldsymbol{Z}$	:	Direction vector of the joint
$\boldsymbol{\theta}$	:	Joint angle vector
$\boldsymbol{\tau}$	:	Actuator torque vector

LIST OF ABBREVIATIONS

c_θ	:	$\cos(\theta)$
DOF	:	Degree of freedom
D-H	:	Denavit – Hartenberg
Kg	:	Kilogram
LxWxH	:	Length - Width – Height
M	:	Meter
S	:	Seconds
s_θ	:	$\sin(\theta)$
ZMP	:	Zero moment point

BIPED LOCOMOTION CONTROL VIA HYBRID POSITION CONTROL AND GRAVITY COMPENSATION MODES

by

OZAN AYHAN

Submitted to the Graduate School of Engineering and Natural Sciences in partial fulfillment of the requirements for the degree of Master of Science

Sabancı University

June 2004

BIPED LOCOMOTION CONTROL VIA HYBRID POSITION CONTROL AND GRAVITY COMPENSATION MODES

APPROVED BY:

Assist. Prof. Dr. Kemalettin Erbatur …………………………
(Thesis Advisor)

Prof. Dr. Asif Şabanovic …………………………

Assist. Prof. Dr. Mahmut Akşit …………………………

Assist. Prof. Dr. Selim Balcısoy …………………………

Assist. Prof. Dr. Mustafa Ünel …………………………

DATE OF APPROVAL: …………………………

ABSTRACT

Past three decades witnessed a growing interest in biped walking robots because of their advantageous use in the human environment. However, their control is challenging because of their many degrees of freedom and non-linearities in their dynamics. Various trajectory generation and walking control approaches ranging from open loop walking to systems with many sensors and feedback loops have been reported in the literature. The tuning of the parameters of reference gait is a common complication encountered. Another important problem of the walking control is the contact transients at the instant of landing.

The method presented in this thesis generated position references for the upper body. Optimization techniques are employed to obtain suitable leg joint torques for the supporting leg to track body reference trajectories. Locomotion is achieved by a swinging leg control scheme, which is based on position control on certain directions, and artificial gravity compensation on certain directions. Soft landing and low impedance problem of the legs can easily be handled with this scheme. Another advantage is that limited number of parameters is required for gait generation.

3D dynamics and ground interaction simulation techniques are employed for a 12-DOF biped robot to test the proposed method. The simulations indicate the applicability of the method in real implementations.

ÖZET

İnsanların bulunduğu ortamlardaki olası faydaları dolayısıyla geçtiğimiz otuz yıl, insansı robotlara karşı giderek artan bir ilgi ve meraka şahitlik etti. Ne var ki, insansı robotların dinamiklerinin doğrusal olmaması ve birçok serbestlik derecesine sahip olmaları denetimlerini güçleştirmektedir. Açık devre yürüyüşten birçok algılayıcı ve geri besleme çemberine sahip sistemlere kadar sayısız güzergah belirleme ve yürüyüş denetim yaklaşımı önerilmiştir. Yürüyüş referansının parametrelerinin ayarlanması karşılaşılan genel bir problemdir. Bir diğer problem de ayağın yere konma anında yaşanan geçici durum davranışlarıdır.

Bu tezde önerilen metot üst vücut için çevrim-dışı güzergah referansları vermektedir. Bu referanslara karşı gelen destek ayağında oluşması gereken eklem momentlerinin elde edilmesi amacıyla eniyileme teknikleri kullanılmıştır. Hareket, belli yönlerde pozisyon denetimine, belli yönlerde ise sanal ağırlık beslemesine dayanan salınım ayağı denetim stratejisiyle sağlanır. Bu metotla ayağın yere yumuşak konması sağlanabilir ve düşük empedans probleminin kolaylıkla üstesinden gelinebilir. Bu metodun bir diğer avantajı da yürüyüşün sağlanması için sınırlı sayıda parametrenin gerekmesidir.

Önerilen metodun test edilmesi için 12 serbestlik derecesine sahip bir insansı robotun üç boyutlu dinamik ve yerle etkileşim benzeşim teknikleri kullanılmıştır. Benzeşimler, bu metodun gerçek uygulamalarda kullanılabilirliğini ortaya koymaktadır.

In the loving memory of my mother Fatma Ayhan,
whose loss diminished us all…

ACKNOWLEDGMENTS

I express my deepest regards and appreciation to Dr. Kemalettin Erbatur, whose invaluable support made realization of my thesis possible.

I extend my appreciation to graduate students in the Mechatronics Program, for their nice talk and company, and being tea mates to me.

I would like to thank to my mother, Fatma Ayhan for sharing her limitless love and grace with us, and to my father, Ahmet Ayhan, who patiently tried to answer my questions, even the most pathological ones, when I was a child.

Finally, to my older brother, Bora, for being at my side, when nobody was.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	:	Constraint matrix
${}^{x}\boldsymbol{A}_{y}$	:	Rotation matrix
$\boldsymbol{A}_{B}$	:	Base-link attitude matrix
$\boldsymbol{b}$	:	Bias vector consisting of gravitational, centrifugal and Corioli's forces
$\tilde{\boldsymbol{b}}$	:	Bias vector corresponding to body dynamics
$\hat{\boldsymbol{b}}$	:	Artificial bias vector for swing leg
$\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})$	:	Centrifugal and Corioli's force matrix
$\boldsymbol{c}$	:	Contact point position vector
D	:	Viscous damping coefficient
d	:	Scalar off-set for foot position reference in sagittal plane
e	:	Error vector position and orientation
$\boldsymbol{F}$	:	Total force
$\boldsymbol{f}_{E}$	:	External force vector
$\boldsymbol{g}(\boldsymbol{x})$	:	Gravity vector
g_{i}	:	Artificial gravity constant
$\boldsymbol{H}(\boldsymbol{x})$	:	Inertia matrix
$\boldsymbol{H}_{ij}$	:	Sub-matrices of inertia matrix
I	:	Index numbers
J	:	Index numbers
$\boldsymbol{K}$	:	External force to generalized force transformation matrix
$\boldsymbol{K}'$	:	Matrix relating external forces to body reaction force and torques
$\boldsymbol{K}''$	:	Matrix relating external forces to joint reaction force and torques

$\boldsymbol{K}_{fc}$	:	Matrix relating foot center velocities and body coordinate center velocities
K_0	:	Derivative gain matrix describing body dynamics
K_1	:	Proportional gain matrix describing body dynamics
K_d	:	PD controller derivative gain matrix for joint position control
K_p	:	PD controller proportional gain matrix for joint position control
M	:	Number of time-variant active contact points
M_A	:	Index numbers of active contact points
$\boldsymbol{M}$	:	Index numbers of external forces
N	:	Number of joints of the robot
$\boldsymbol{N}$	:	Total moment
$\boldsymbol{p}$	:	Axes origin position vector
$\dot{\boldsymbol{p}}$	:	Axes origin velocity vector
$\ddot{\boldsymbol{p}}$	:	Axes origin acceleration vector
$\boldsymbol{p}_B$	:	Base-link position vector
R	:	Rotation matrix
r	:	Body linear and angular velocity
$\boldsymbol{r}$	:	Center of mass position
$\dot{\boldsymbol{r}}$	:	Center of mass velocity
$\ddot{\boldsymbol{r}}$	:	Center of mass acceleration
S	:	Selection matrix for position controlled and gravity compensated modes
s_i	:	Entry of the selection matrix
$\boldsymbol{S}$	:	Center of mass position vector with respect to coordinate system
$\boldsymbol{u}$	:	Generalized force vector
$\boldsymbol{u}_E$	:	Generalized force vector generated by external forces
v	:	Reference acceleration for the body in inverse dynamics
$\boldsymbol{v}$	:	Generalized velocities vector
$\dot{\boldsymbol{v}}$	:	Generalized accelerations vector
$\boldsymbol{v}_B$	:	Base-link velocity vector
$\dot{\boldsymbol{v}}_B$	:	Base-link acceleration vector

$\boldsymbol{w}$	:	Joint angular velocity vector
$\dot{\boldsymbol{w}}$	:	Joint angular acceleration vector
$\boldsymbol{w}_B$	:	Base-link angular velocity vector
$\dot{\boldsymbol{w}}_B$	:	Base-link angular acceleration vector
$\boldsymbol{x}$	:	Generalized coordinate vector
x_i	:	i^{th} input variable
$\boldsymbol{Z}$	:	Direction vector of the joint
$\boldsymbol{\theta}$	:	Joint angle vector
$\boldsymbol{\tau}$	:	Actuator torque vector

LIST OF ABBREVIATIONS

c_θ	:	$\cos(\theta)$
DOF	:	Degree of freedom
D-H	:	Denavit – Hartenberg
Kg	:	Kilogram
LxWxH	:	Length - Width – Height
M	:	Meter
S	:	Seconds
s_θ	:	$\sin(\theta)$
ZMP	:	Zero moment point

1. INTRODUCTION

There is extensive research going on about humanoid robots and biped locomotion. The interest stems from the dream of humanoids substituting or accompanying human beings in certain tasks in the future. Yet, existing humanoids suffer from various problems, such as lack of mobility [1], efficiency and walking speed [2], to name a few.

Apart from the huge cost and man-hours of constructing one, the structure exploits very non-linear and complex dynamics [3]. A walking pattern resulting in a stable gait is difficult to find. Many methods ranging from off-line trajectory generation [4] to feedback systems based on multi-sensor-fusion are employed in the control of biped walking robots [5]. Trial and error based tuning of gait parameters are reported. In some works, tuning of the many parameters of the robot gait is carried out by evolutionary control [6] or via neuro-fuzzy systems [7]. Energy or joint torque minimization can be used as criteria for the tuning [8,9]. This thesis addresses the reference generation problem in such a way that a small number of parameters are used in the description of the desired trajectory. The references employed are in the form of desired trajectories for the trunk of the robot. Suitable torques for the supporting leg joints for the tracking of the trunk reference trajectory are obtained by constrained optimization techniques. Locomotion is achieved by a swinging leg control scheme, which works in parallel with the body reference position tracking. The swinging leg control scheme employs gravity compensation in the z -direction, and application of position control in x, y and rotational axes. This scheme of locomotion generation can be achieved with a small number of walking parameters, since it does not impose any fixed joint trajectories for the legs. The soft landing and low impedance problem of robot legs is considered by a number of researchers [10-14]. An additional advantage of the gravity compensation technique is that soft landing of the swinging leg can be achieved intrinsically.

The next chapter is a survey of the literature on the topics of the history of biped robots and biped locomotion techniques.

Chapter 3 describes the biped model used in this work explaining the dynamics equations of the test bed and the basis of the equations used for the simulation.

A reactive force control scheme, which is employed as a building block for control scheme in the subsequent chapters is introduced in Chapter 4.

Swing leg foot control based solely on artificial gravity compensation is outlined in Chapter 5.

Swing leg foot control based on selective gravity compensation and position controlled modes is outlined in Chapter 6.

Chapter 7 presents the simulation results of the work.

The last chapter summarizes the work carried out and results obtained. Comments on the results, conclusion remarks and directions for the future work are also given in this part.

2. A SURVEY ON BIPED LOCOMOTION CONTROL TECHNIQUES

2.1. History of Biped Robotics

Ever after the first humanoid robot, Wabot-1, [15] was developed at the Waseda University, biped robotics has drawn the attraction of many researchers. This robot's abilities were really impressive at the time. Although it took 45 seconds to take a step, this robot had gripping hands with tactile sensors, and a communication and vision system.

Many other walking robots were developed in the 1980s like Waseda's WL-10RD, WHL-11, and Batelle's Pacific Northwest Laboratories' Manny, some of which even realized dynamic walking [16]. Another research topic focused on efficiency and active balance in legged locomotion. M. H. Raibert's one-legged hopping machine [17] (Fig.2.1.) demonstrated that stability can be achieved by bouncing, pioneering the ballistic flight in biped locomotion. McGeer's passive walker [18] (Fig.2.2.) could walk down a slight incline without intervention or control action. However, it was not until Honda announced its long existing project to build a humanoid robot that a self-contained and free-roaming robot was to appear. The last generation Honda [19] robot, Asimo, has not only a very natural walking style, but a cute looking too. It has become a public figure in social events ever then, revealing a different aspect of developing a humanoid.

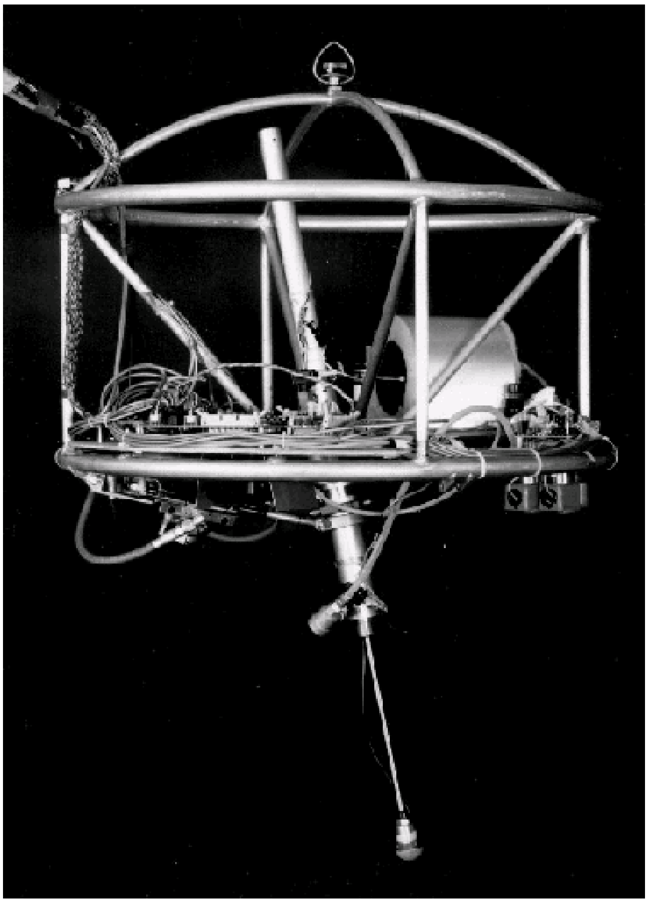


Figure 2.1. M.H. Raibert's 3D hopping machine.

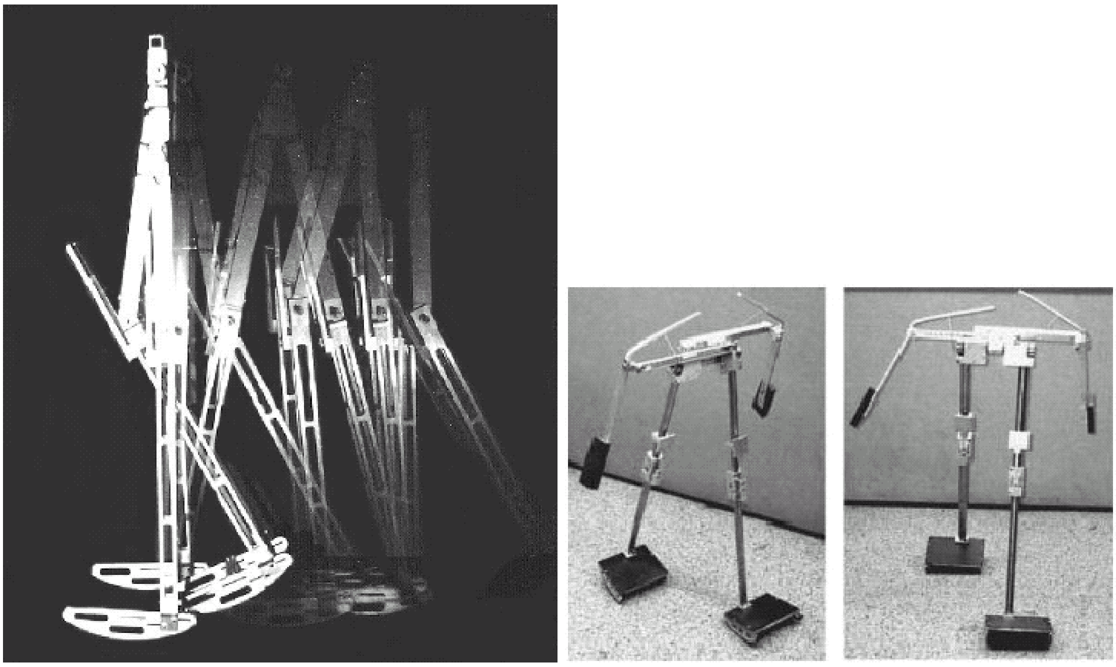


Figure 2.2. McGeer's passive walker.

Recently, Kawada Industries Inc. has also introduced the HRP-2P (Fig.2.3.). The structure is really impressive with 30 dofs and no backpack. Another on-going project is Technical University of Munich's Johnnie, which can walk on even and uneven ground, and around curves (Fig.2.4.).

2.2. Literature Survey on Biped Locomotion Control

Its superior characteristics in obstacle avoidance and dexterity make biped structure attractive to many researchers. Various locomotion strategies relying on intuition [20], optimization [8,9,21,22,23] or human motion capture data [24] have been proposed. Although some algorithms have already demonstrated successful and dynamic walking [25-27], there are still many rooms for increasing efficiency, walking speed and mobility.

The theoretical foundation of biped locomotion is often attributed to the work of Vukobratovic *et. al* [28-30]. They provided the concept of zero moment point (ZMP) and studied the stability of biped gaits.

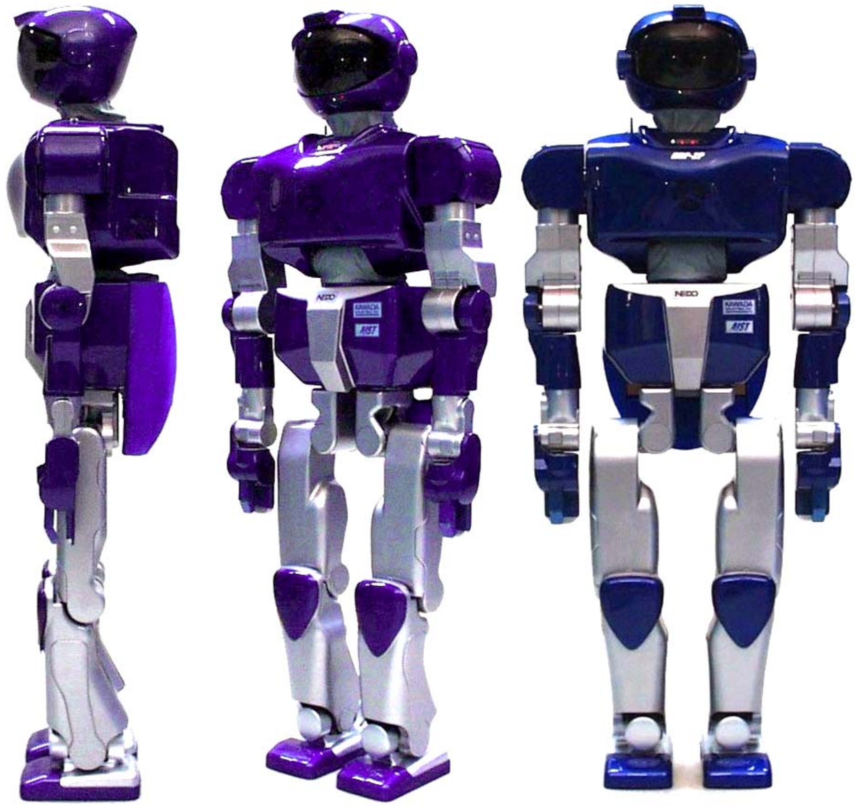


Figure 2.3. The last prototype of Humanoid Research Project: HRP-2.

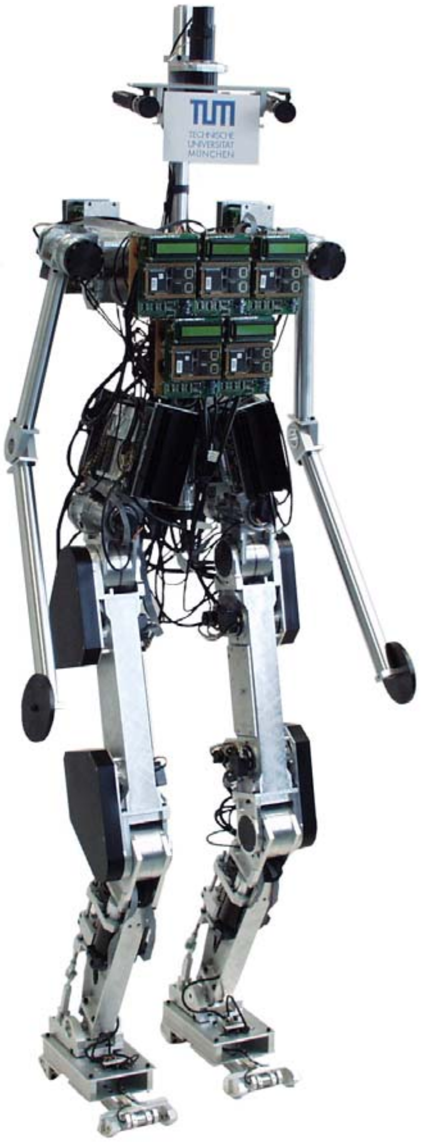


Figure 2.4. Johnnie, Technical University of Munich.

Biped dynamics was investigated in a control perspective by [31]. They divided biped gait into fast and slow modes.

Takanishi *et. al* [27] devised a control system which planned a ZMP trajectory and pre-defined motion trajectories to realize stable dynamic walking. Yamaguchi, Takanishi and Kato [32] used trunk moment compensation for canceling the external moment effects occurring on the body due to environmental interaction. Fujimoto and Kawamura proposed a similar scheme, which involved a feedback compensation of yaw axis rotation by arm motion. Huang and Kajita [33] gave trajectories for foot and waist separately and investigated the stability margin. Also, Park and Chung [34] devised an on-line algorithm modifying the desired acceleration of the body in its vertical direction to compensate for the external moment effects using the ZMP information. The trajectory is modified to increase the vertical acceleration of the body, in case the ZMP goes out of some specific bounds for the ZMP.

The aforementioned approaches are mostly model-based, requiring precise knowledge of dynamics of the robot. There are other approaches [5,17,20,35], where limited knowledge about the humanoid is utilized. These schemes generally disregard the knowledge about the overall center of mass of the humanoid, or mass, inertia and center of mass of each link, or total angular momentum.

Less modeling immediately brings the necessity of feedback control, since the controller knows little about the system. The authors of [5] employ a feedback control scheme based on multi-sensor fusion, for their hybrid position-force control based approach. In [20], virtual forces are acting on a biped to create a gait pattern, based on physical intuition. The virtual forces or torques are generated by mechanical elements like springs and dampers, or gravitational or potential field. The virtual force is projected to the joint torques, creating the illusion that the force was available in the system.

In [35], biped model is simplified to a 3D inverted pendulum model. The dynamics governing the simplified model is relatively simple, and includes less parameters of the humanoid.

All of the previous approaches lack the mobility and adaptability for real world conditions. They require fine tuning of various parameters for a successful walk. Another problem of the existing approaches is that they suffer the problems associated with position control as unnatural stiffness of joints.

In the proposed approach, however, limited number of parameters is involved and compliance of joints is achieved, which may improve the usability and safety of humanoids in human environment. The approach inherits ideas from virtual model control [20], reactive force control [5] and hybrid position/control of Raibert and Craig..

3. THE BIPED MODEL EMPLOYED AS THE SIMULATION TEST BED

The biped model used in this work as a simulation test bed is called “Mari-2”, one of the biped robots of Yokohama National University, Japan (Fig. 3.1., Fig. 3.2.).

This model is selected since it is an experimentally tested model and suitable for our simulations. The test bed consists of two 6-DOF legs and a trunk connecting them. Approximate link sizes and the masses of the biped are given in Table 3.1.

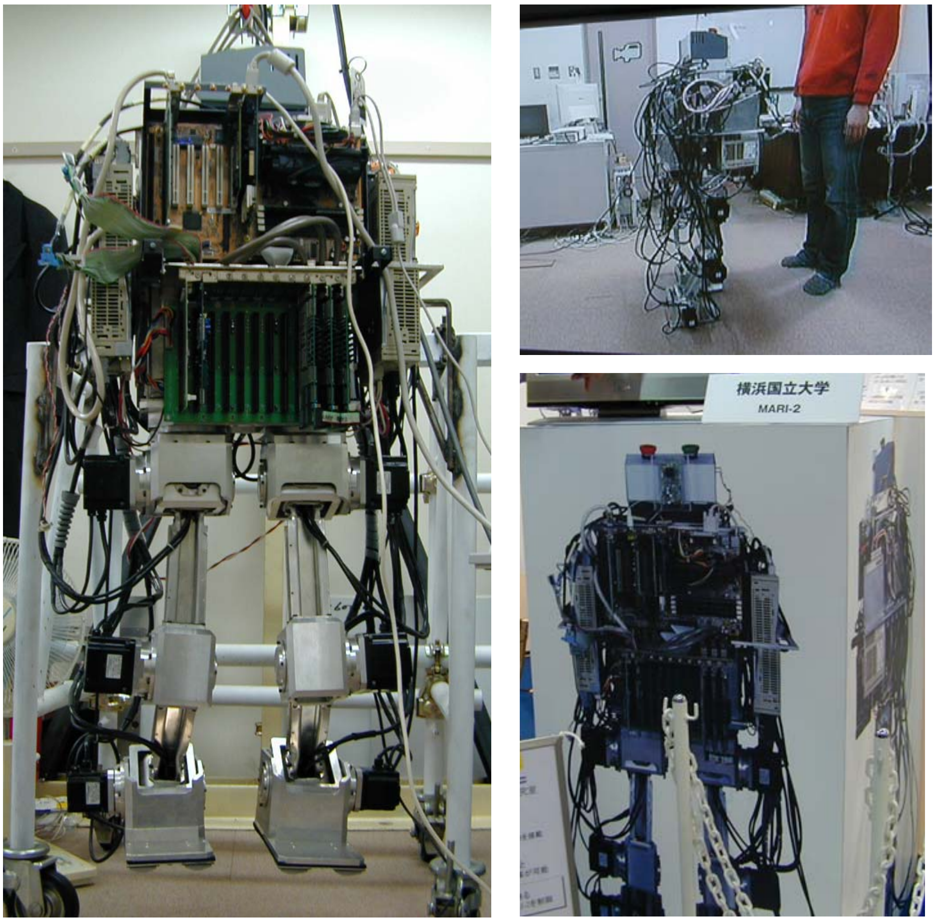


Figure 3.1. Some pictures of the used test bed robot, Mari-2.

Table 3.1. Masses and dimensions of the biped robot links.

Link	Dimensions (LxWxH) [m]	Mass [kg]
Trunk	0.2 x 0.4 x 0.5	50
Thigh	0.27 x 0.1 x 0.1	12
Calf	0.22 x 0.05 x 0.1	0.5
Foot	0.1 x 0.12 x 0.25	5.5

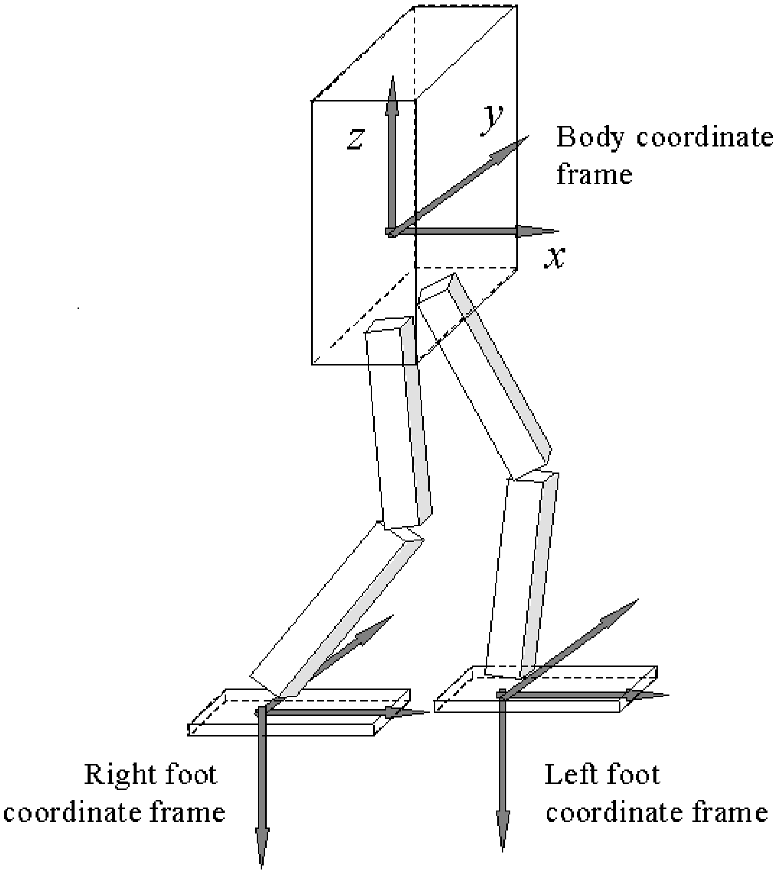


Figure 3.2. The body and the foot coordinate frames.

According to D-H convention, the joint axes can be shown in figures 3.3 and 3.4, and accordingly, D-H parameters are listed at Table 3.2.

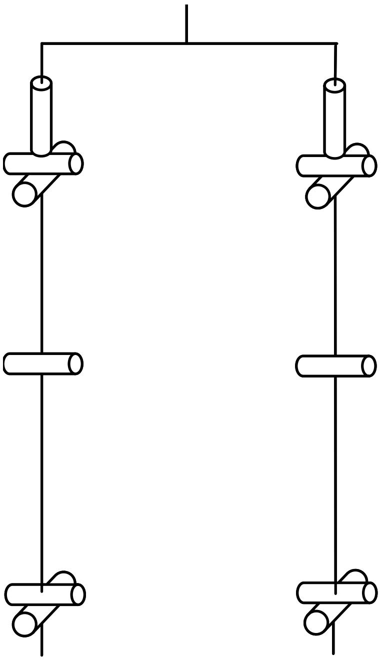


Figure 3.3. Joint placements of the biped robot.

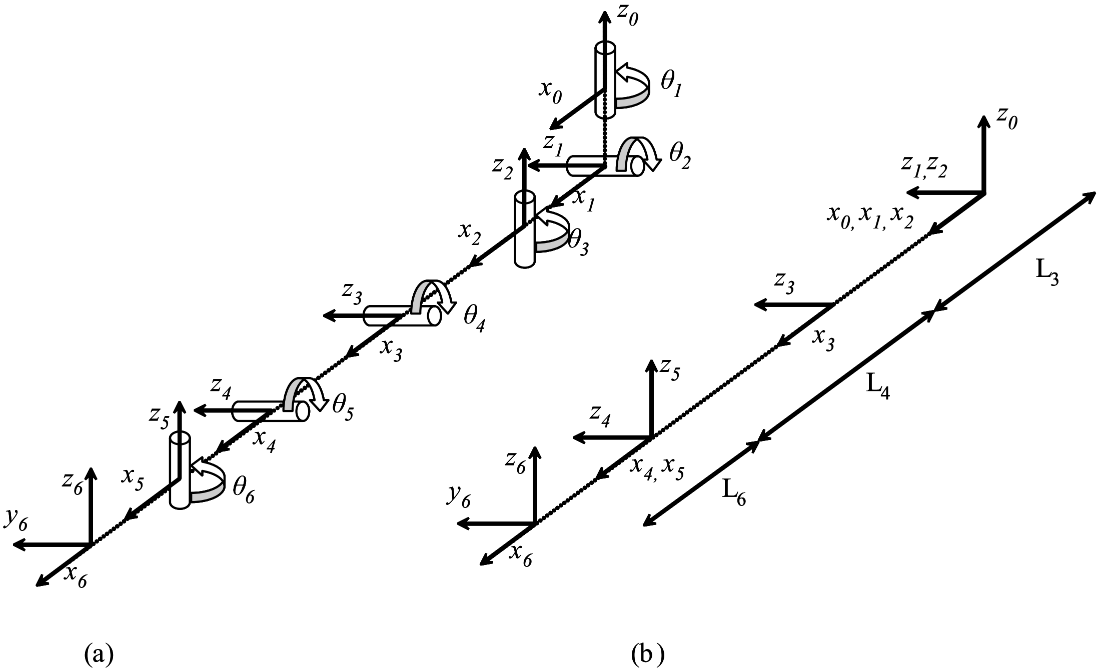


Figure 3.4. (a) Exploded view of the joints and their axes; (b) Joint axes and their placements.

Table 3.2. D-H Parameters of the biped leg

Link	a_i	$\boldsymbol{\alpha}_i$	d_i	$\boldsymbol{\theta}_i$
1	0	$\frac{\pi}{2}$	0	θ_1^*
2	0	$-\frac{\pi}{2}$	0	θ_2^*
3	L_3	$\frac{\pi}{2}$	0	θ_3^*
4	L_4	0	0	θ_4^*
5	0	$-\frac{\pi}{2}$	0	θ_5^*
6	L_6	0	0	θ_6^*

The biped robot is modeled as a free-fall manipulator, which is not fixed to the ground but has interaction with. In order to formulate the dynamics of a free-fall manipulator, position and attitude variables of the base-link should be introduced. Let generalized coordinates $\boldsymbol{x}$, generalized velocities $\boldsymbol{v}$, and generalized forces $\boldsymbol{u}$ be:

$$\boldsymbol{x}^T = [\boldsymbol{p}_B^T, \boldsymbol{A}_B^T, \boldsymbol{\theta}^T] \in R^3 \times SO(3) \times R^N \tag{3.1}$$

$$\boldsymbol{v}^T = [\boldsymbol{v}_B^T, \boldsymbol{w}_B^T, \boldsymbol{w}^T] \in R^3 \times R^3 \times R^N \tag{3.2}$$

$$\boldsymbol{u}^T = [\boldsymbol{f}_B^T, \boldsymbol{n}_B^T, \boldsymbol{\tau}^T] \in R^3 \times R^3 \times R^N \tag{3.3}$$

where

$\boldsymbol{p}_B$: 3×1 vector specifying base-link position

$\boldsymbol{A}_B$: 3×3 matrix specifying base-link attitude

$\boldsymbol{\theta}$: $N\times1$ vector specifying joint angle

$\boldsymbol{v}_B$: 3×1 vector specifying base-link velocity

$\boldsymbol{w}_B$: 3×1 vector specifying angular velocity of base-link

$\boldsymbol{w}$: $N\times1$ vector specifying joint angular velocity

$\boldsymbol{f}_B$: 3×1 force vector generated in base-link

$\boldsymbol{n}_B$: 3×1 torque vector generated in base-link

$\boldsymbol{\tau}$: $N\times1$ torque vector generated by actuator

N : Number of joints of the robot

$\boldsymbol{A}_B$ is the transformation matrix giving the position of link axes relative to the world axes. The equations of motion of the robot become:

$$\dot{\boldsymbol{p}}_B = \boldsymbol{v}_B \tag{3.4}$$

$$\dot{\boldsymbol{A}}_B = \boldsymbol{w}_B \times \boldsymbol{A}_B \tag{3.5}$$

$$\dot{\boldsymbol{\theta}}_B = \boldsymbol{w} \tag{3.6}$$

and

$$\boldsymbol{H}(\boldsymbol{x})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{u}_E = \boldsymbol{u} \tag{3.7}$$

where

$\boldsymbol{H}(\boldsymbol{x})$: $(N+6)\times(N+6)$ inertia matrix

$\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})$: $(N+6)\times(N+6)$ matrix specifying centrifugal and Corioli's effects

$\boldsymbol{g}(\boldsymbol{x})$: $(N+6)\times 1$ vector specifying gravity effect

$\boldsymbol{u}_E$: $(N+6)\times 1$ vector specifying generalized forces generated by external forces

Equation (3.7) represents a general form of the dynamic equation.

We have modeled the biped robot with the previous analysis. A simulation environment is devised to test the proposed method that will numerically calculate joint angles and body position at each step-time, given the actuator torque input. The following argumentation summarizes how the simulation of the biped system is implemented.

The generalized states $(\boldsymbol{x},\boldsymbol{v})$ are updated by Euler integration at each step time h.

$$\boldsymbol{p}_B(t+h) = \boldsymbol{p}_B(t) + h\boldsymbol{v}_B(t) \tag{3.8}$$

$$\boldsymbol{A}_B(t+h) = \boldsymbol{T}(h\boldsymbol{w}_B)\boldsymbol{A}_B(t) \tag{3.9}$$

$$\theta(t+h) = \theta(t) + h\boldsymbol{w} \tag{3.10}$$

$$\boldsymbol{v}(t+h) = \boldsymbol{v}(t) + h\dot{\boldsymbol{v}}(t) \tag{3.11}$$

$$\dot{\boldsymbol{v}}(t) = \boldsymbol{H}(\boldsymbol{x}(t))^{-1}\left[\boldsymbol{u}(t) - \boldsymbol{u}_E(\boldsymbol{x}(t),\boldsymbol{v}(t)) - \boldsymbol{b}(\boldsymbol{x}(t),\boldsymbol{v}(t)\right] \tag{3.12}$$

where the biasing vector $\boldsymbol{b}(\boldsymbol{x}(t),\boldsymbol{v}(t))$ is defined as

$$\boldsymbol{b}(\boldsymbol{x}(t),\boldsymbol{v}(t)) = \boldsymbol{C}(\boldsymbol{x}(t),\boldsymbol{v}(t))\boldsymbol{v}(t) + \boldsymbol{g}(\boldsymbol{x}(t)) \tag{3.13}$$

Equations (3.8)-(3.13) summarizes how the simulation is done with numerical integration. $\boldsymbol{T}(h\boldsymbol{w}_B)$ of equation (3.9) is a rotational transformer around $\boldsymbol{w}_B$ axis with an angle $h|\boldsymbol{w}_B|$. It is used for updating the base link's orientation matrix, given the angular velocity of the base link. It is obtained through

$$\boldsymbol{T}(h\boldsymbol{w}_B) = \left[(\cos\psi)\boldsymbol{I}_3 + (1-\cos\psi)\boldsymbol{r}\boldsymbol{r}^T + (\sin\psi)[\boldsymbol{r}\times]\right] \tag{3.14}$$

where $\psi = h|\boldsymbol{w}_B|$, $\boldsymbol{I}_3$ is 3×3 identity matrix, and $\boldsymbol{r} = \boldsymbol{w}_B / |\boldsymbol{w}_B|$.

$\boldsymbol{H}(\boldsymbol{x})$ and $\boldsymbol{b}(\boldsymbol{x},\boldsymbol{v})$ can be obtained by inverse dynamics calculation using the Newton-Euler iterative formulation as in [36]. Inertia matrix $\boldsymbol{H}(\boldsymbol{x})$ can be calculated numerically by solving inverse dynamics with setting $\boldsymbol{x}$ to the current state and $\dot{\boldsymbol{v}} = \boldsymbol{e}_j$ for $1 \le j \le N+6$, and ignoring centrifugal, Corioli's, gravitational and external forces. Here $\boldsymbol{e}_j$ means a unit vector with all elements being 0, other than the *j*th element which is 1. At each iteration, *j*th column of the inertia matrix is calculated. Also, symmetricity of inertia matrix can be made use for computational considerations. The biasing vector $\boldsymbol{b}(\boldsymbol{x},\boldsymbol{v}) = \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x})$ can be calculated numerically by solving inverse dynamics with setting ($\boldsymbol{x}$, $\boldsymbol{v}$) to the current state and $\dot{\boldsymbol{v}} = 0$, and ignoring external forces.

The abovementioned inverse dynamics calculation of the robot is done by the following recursive Newton-Euler formulation. The formulation is performed in the algorithm frame of the [36] with some difference. To compute inverse dynamics efficiently, the link-fixed coordinates shown in Fig. 3.6 are introduced.

The i^{th} coordinates are defined as

$${}^0\boldsymbol{A}_i = {}^0\boldsymbol{A}_1 {}^1\boldsymbol{A}_2 \ldots {}^{\underline{i}}\boldsymbol{A}_i \tag{3.15}$$

$${}^0\boldsymbol{A}_i = \boldsymbol{A}_B \tag{3.16}$$

where

$${}^{\underline{i}}\boldsymbol{A}_i = \left[{}^{\underline{i}}\boldsymbol{x}_i, {}^{\underline{i}}\boldsymbol{y}_i, {}^{\underline{i}}\boldsymbol{z}_i\right]. \tag{3.17}$$

${}^{\underline{i}}\boldsymbol{A}_i$ transforms any vector with reference to i^{th} coordinate system to a new coordinate system whose coordinates are parallel to $\underline{i}^{\text{th}}$ coordinate system. Note that ${}^{\underline{i}}\boldsymbol{A}_i^{-1} = {}^{\underline{i}}\boldsymbol{A}_i^{T} = {}^{i}\boldsymbol{A}_{\underline{i}}$ since coordinate system is orthogonal. The transformation can be parameterized by

$${}^{\underline{i}}\boldsymbol{A}_i = \begin{bmatrix} c_{\phi_{i-1}} c_{\theta_{i-1}} - s_{\phi_{i-1}} c_{\alpha_{i-1}} s_{\theta_{i-1}} & -c_{\phi_{i-1}} s_{\theta_{i-1}} - s_{\phi_{i-1}} c_{\alpha_{i-1}} c_{\theta_{i-1}} & s_{\phi_{i-1}} s_{\alpha_{i-1}} \\ s_{\phi_{i-1}} c_{\theta_{i-1}} + c_{\phi_{i-1}} c_{\alpha_{i-1}} s_{\theta_{i-1}} & -s_{\phi_{i-1}} s_{\theta_{i-1}} + c_{\phi_{i-1}} c_{\alpha_{i-1}} c_{\theta_{i-1}} & -c_{\phi_{i-1}} s_{\alpha_{i-1}} \\ s_{\alpha_{i-1}} s_{\theta_{i-1}} & s_{\alpha_{i-1}} c_{\theta_{i-1}} & c_{\alpha_{i-1}} \end{bmatrix}. \tag{3.18}$$

When $\dot{\boldsymbol{v}}_B, \dot{\boldsymbol{w}}_B, \boldsymbol{w}_B, \ddot{\boldsymbol{q}}_i, \dot{\boldsymbol{q}}_i$ and $\boldsymbol{q}_i, 1 \le i \le N,$ are given, for *i*=1, the angular velocity ${}^{i}\boldsymbol{w}_i$, the angular acceleration ${}^{i}\dot{\boldsymbol{w}}_i$ and the acceleration of the origin ${}^{i}\ddot{\boldsymbol{p}}_i$ of *i*th link referenced to its own link coordinates can be recurrently obtained as follows.

$${}^{1}\ddot{\boldsymbol{p}}_1 = {}^{1}\boldsymbol{A}_0(\dot{\boldsymbol{v}}_B + \boldsymbol{g}) \tag{3.19}$$

$${}^{1}\boldsymbol{w}_1 = {}^{1}\boldsymbol{A}_0\boldsymbol{w}_B \tag{3.20}$$

$${}^{1}\dot{\boldsymbol{w}}_1 = {}^{1}\boldsymbol{A}_0\dot{\boldsymbol{w}}_B \tag{3.21}$$

and for $2 \le i \le N$

$${}^{i}\boldsymbol{w}_i = {}^{i}\boldsymbol{A}_{\underline{i}}\,{}^{\underline{i}}\boldsymbol{w}_{\underline{i}} + \boldsymbol{z}_0\dot{\boldsymbol{q}}_{i-1} \tag{3.22}$$

$${}^{i}\dot{\boldsymbol{w}}_i = {}^{i}\boldsymbol{A}_{\underline{i}}\,{}^{\underline{i}}\dot{\boldsymbol{w}}_{\underline{i}} + \boldsymbol{z_0}\ddot{\boldsymbol{q}}_{i-1} + ({}^{i}\boldsymbol{A}_{\underline{i}}\,{}^{\underline{i}}\boldsymbol{w}_{\underline{i}}) \times \boldsymbol{z_0}\dot{\boldsymbol{q}}_{i-1} \tag{3.23}$$

$${}^{i}\ddot{\boldsymbol{p}}_i = {}^{i}\boldsymbol{A}_{\underline{i}}\left[{}^{\underline{i}}\dot{\boldsymbol{w}}_{\underline{i}} \times {}^{\underline{i}}\boldsymbol{p}_i^{*} + {}^{\underline{i}}\boldsymbol{w}_{\underline{i}} \times \left({}^{\underline{i}}\boldsymbol{w}_{\underline{i}} \times {}^{\underline{i}}\boldsymbol{p}_i^{*}\right) + {}^{\underline{i}}\ddot{\boldsymbol{p}}_{\underline{i}}^{*}\right]. \tag{3.24}$$

Thus for $1 \le i \le N$, the acceleration of the center of mass $\ddot{\boldsymbol{r}}_i$, the total force $\boldsymbol{F_i}$, and the total moment $\boldsymbol{N_i}$ of the *i*th link can be calculated as follows.

$${}^{i}\ddot{\boldsymbol{r}}_i = {}^{i}\dot{\boldsymbol{w}}_i \times {}^{i}\boldsymbol{s}_i + {}^{i}\boldsymbol{w}_i \times ({}^{i}\boldsymbol{w}_i \times {}^{i}\boldsymbol{s}_i) + {}^{i}\ddot{\boldsymbol{p}}_i \tag{3.25}$$

$${}^{i}\boldsymbol{F}_i = {}^{i}m_i\,{}^{i}\ddot{\boldsymbol{r}}_i \tag{3.26}$$

$${}^{i}\boldsymbol{N}_i = {}^{i}\boldsymbol{J}_i\,{}^{i}\dot{\boldsymbol{w}}_i + {}^{i}\boldsymbol{w}_i \times ({}^{i}\boldsymbol{J}_i \times {}^{i}\boldsymbol{w}_i), \tag{3.27}$$

where

$${}^{\underline{i}}\boldsymbol{p}_i^{*} = \boldsymbol{p}_i - \boldsymbol{p}_{\underline{i}} = \left[a_{i-1}\cos\phi_{i-1}, a_{i-1}\sin\phi_{i-1}, d_{i-1}\right]^T. \tag{3.28}$$

$\boldsymbol{z_i}$ denotes the direction of the joint *i*-1, and $\boldsymbol{s_i}$ denotes the center of mass with respect to the *i*th coordinate system origin (Fig. 3.5). The gravity effect can be considered by adding a gravitational acceleration on one of the base links' $\ddot{\boldsymbol{v}}_B$. Then, for $1 \le i \le N$ $\boldsymbol{f_i}$ and $\boldsymbol{n_i}$, the force and moment exerted on link *i* by inner link $\underline{i}$ is

$${}^{i}\boldsymbol{f}_i = {}^{i}\boldsymbol{F}_i + \sum_{j \in O_i} {}^{\underline{j}}\boldsymbol{A}_j\,{}^{j}\boldsymbol{f}_j + {}^{i}\boldsymbol{A}_0 \sum_{j \in M_i} {}^{0}\boldsymbol{f}_{E_j} \tag{3.29}$$

$${}^{i}\boldsymbol{n}_i = {}^{i}\boldsymbol{N}_i + \sum_{j \in O_i}\left\lfloor {}^{\underline{j}}\boldsymbol{A}_j\,{}^{j}\boldsymbol{n}_j + {}^{\underline{j}}p_j^{*} \times ({}^{\underline{j}}\boldsymbol{A}_j\,{}^{j}\boldsymbol{f}_j)\right\rfloor + {}^{i}\boldsymbol{s}_i \times {}^{i}\boldsymbol{F}_i - \sum_{j \in M_i}\left\lfloor {}^{j}\boldsymbol{c}_j \times ({}^{i}\boldsymbol{A}_0\,{}^{0}\boldsymbol{f}_{E_j})\right\rfloor \tag{3.30}$$

where

$\boldsymbol{f}_{E_j}$: 3×1 vector specifying *j*th external force

$\boldsymbol{M}_i$: A set of index numbers of external forces which are imposed on link *I*

$\boldsymbol{c}_j$: 3×1 position vector of *j*th contact point with respect to the origin of its own link-fixed coordinates as shown in Fig. 3.6.

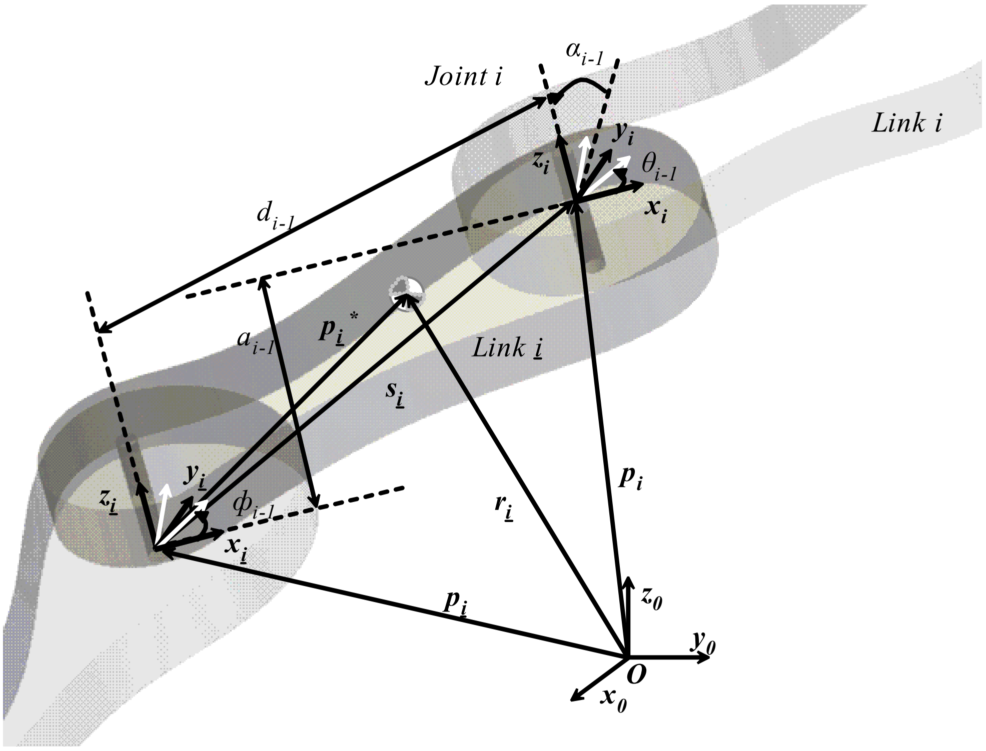


Figure 3.5. Coordinates and parameters of links.

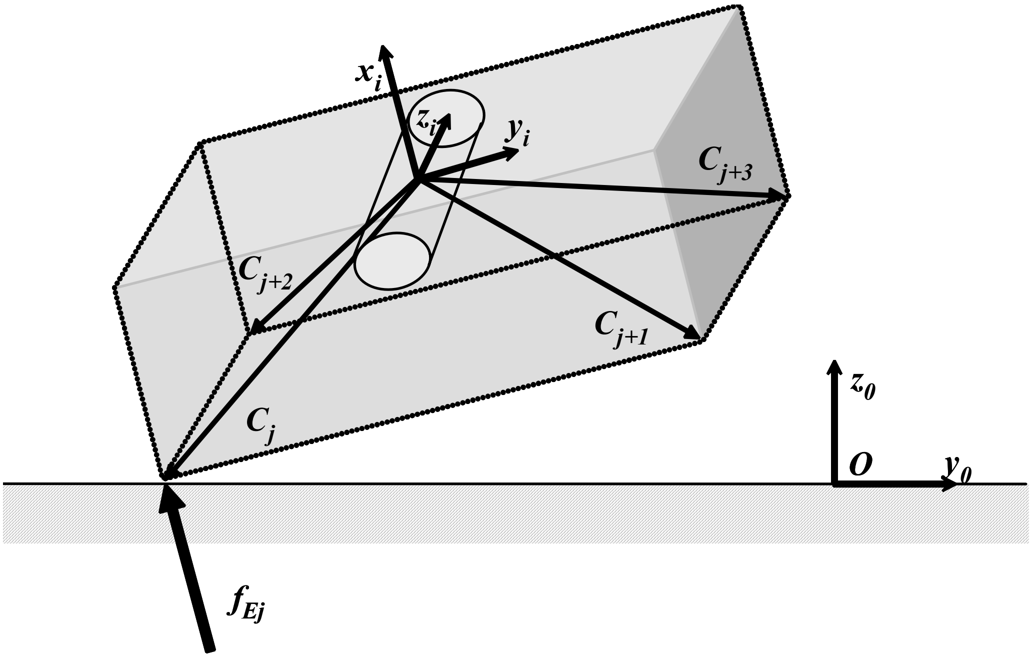


Figure 3.6. Contact points of the foot.

As a result, given $\boldsymbol{x}$, $\boldsymbol{v}$ and $\dot{\boldsymbol{v}}$, for $1 \le i \le N-1$ the solution of inverse dynamics $\boldsymbol{u}^T = \left[\boldsymbol{f}_B^T, \boldsymbol{n}_B^T, \boldsymbol{\tau}_B^T\right]$ is

$$\tau_i = {}^{i+1}\mathbf{n}_{i+1}^T \mathbf{z}_0 + D_i \dot{q}_i \,, \tag{3.31}$$

where τ_i denotes the torque on i^{th} joint, D_i denotes the viscous damping coefficient of joint *i*; and for *i*=1,

$$\boldsymbol{f}_B = {}^0\boldsymbol{A}_1 \, {}^1\boldsymbol{f}_1 \tag{3.32}$$

$$\boldsymbol{n}_B = {}^i\boldsymbol{A}_1 {}^1\boldsymbol{n}_1 \tag{3.33}$$

where $\boldsymbol{f_B}$ and $\boldsymbol{n_B}$ are the force and the moment exerted on the origin of the base link, respectively. They are equal to zero for a free-fall manipulator.

With this iterative scheme, joint angles and body position can be calculated at each step-time, given the actuator torque input. The question remains, however, how to model the environmental interaction forces, i.e. compute the external force vector $\boldsymbol{f}_E$. An adaptive penalty based method [37] is employed to calculate the interaction forces. The method is based on the minimization of the kinetic energy of the bodies in contact. After the external force vector $\boldsymbol{f}_E$ is obtained generalized external force $\boldsymbol{u}_E$ can be obtained by

$$\boldsymbol{u}_E = \sum_{j \in M_A} \boldsymbol{K}_j \boldsymbol{f}_E \tag{3.34}$$

where

$$M_A = \bigcup_{i=1}^{N} M_i \tag{3.35}$$

$\boldsymbol{K}_j$: $(N+6)\times 3$ matrix specifying transformer from *j*th external force to generalized forces

M_A : A set of index numbers of all active contact points

$\boldsymbol{f}_E$: $(3M)\times 1$ vector which consists of all components of all active contact points

$\boldsymbol{K}$: $(N+6)\times(3M)$ matrix specifying transformer from $\boldsymbol{f}_E$ to generalized forces

M : Number of active contact points which is time-variant

The transformer matrix $\boldsymbol{K}$ too, can be computed as a part of the Newton-Euler formulation by setting $\boldsymbol{x}$ to the current state, $f_E = \mathbf{e}_j$ for $1 \le j \le M$ and $\dot{\boldsymbol{v}} = 0$, and ignoring gravity, centrifugal, and Corioli's effects. With the above scheme, generalized external force $\boldsymbol{u}_E$ can be computed for use in the simulation.

The simulation scheme follows the procedure summarized below at each step time:

1. Set the initial state ($\boldsymbol{x}, \boldsymbol{v}$) to the current state.
2. Calculate the inertia matrix $\boldsymbol{H}(\boldsymbol{x})$ numerically by solving inverse dynamics (3.15)- (3.33) with setting $\boldsymbol{x}$ to the current state and $\dot{\boldsymbol{v}} = \boldsymbol{e}_j$ for $1 \le j \le N + 6$, and ignoring centrifugal, Corioli's, gravitational and external forces. The ignored forces can be set to zero can be done by setting $\boldsymbol{v} = 0$, gravitational acceleration $g_z = 0$ and $f_E = 0$ in the inverse dynamics. This step, however, is not implemented at each step time, but once every 100 steps, for computational considerations.
3. Calculate the biasing vector $\boldsymbol{b}(\boldsymbol{x},\boldsymbol{v}) = \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x})$ numerically by solving inverse dynamics (3.15)- (3.33) with setting ($\boldsymbol{x}, \boldsymbol{v}$) to the current state and $\dot{\boldsymbol{v}} = 0$, and ignoring external forces. This step too, is not implemented at each step time, but once every 100 steps, for computational considerations.
4. Calculate the transformation matrix $\boldsymbol{K}$ numerically by solving inverse dynamics equations (3.15)-(3.33) with setting $\boldsymbol{x}$ to the current state, $f_E = \mathbf{e}_j$ for $1 \le j \le M$ and $\dot{\boldsymbol{v}} = 0$, and ignoring gravity, centrifugal, and Corioli's effects.
5. Calculate the external force f_E by the adaptive penalty method of [37] with the current state ($\boldsymbol{x}, \boldsymbol{v}$). Calculate $\boldsymbol{u}_E$ by (3.34).
6. Calculate the generalized acceleration $\dot{\boldsymbol{v}}$ by (3.12).
7. Update the generalized states ($\boldsymbol{x}, \boldsymbol{v}$) by numerical integration described in (3.8)-(3.11) and (3.14).
8. Return to 1 for next simulation cycle.

Using this procedure, we can completely analyze joint torques of the biped and interaction forces from the ground when the trajectories of the position, the velocity and the acceleration of the joint angles, those of the body, and those of the body attitude are given.

The next chapter explains the reactive force control scheme to track off-line generated body reference trajectories.

4. A REACTIVE FORCE CONTROL SCHEME

The general biped dynamical equation is given in (3.7). By defining the biasing vector $\boldsymbol{b}(\boldsymbol{x},\boldsymbol{v}) = \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x})$ as in (3.13), and decomposing it for body and leg dynamics, one can obtain the following equation,

$$\begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} & \boldsymbol{H}_{13} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} & \boldsymbol{H}_{23} \\ \boldsymbol{H}_{31} & \boldsymbol{H}_{32} & \boldsymbol{H}_{33} \end{bmatrix} \begin{pmatrix} \dot{\boldsymbol{v}}_B \\ \dot{\boldsymbol{w}}_B \\ \ddot{\theta} \end{pmatrix} + \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \end{pmatrix} + \begin{pmatrix} \boldsymbol{u}_{E_1} \\ \boldsymbol{u}_{E_2} \\ \boldsymbol{u}_{E_3} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \tau \end{pmatrix} \tag{4.1}$$

where $\boldsymbol{H}_{ij}$ for $(i, j) \in \{1,2,3\}$ are sub-matrices of the robot inertia matrix $\boldsymbol{H}(\boldsymbol{x})$. $\boldsymbol{v}_B$ is the linear velocity of the robot body coordinate frame center with respect to a fixed world coordinate frame, $\boldsymbol{w}_B$ is the angular velocity of the robot body coordinate frame with respect to a fixed world coordinate frame, and θ is the vector of joint displacements of the biped. The vectors $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{b}_3$ are the sub-vectors of the bias vector $\boldsymbol{b}(\boldsymbol{x},\boldsymbol{v})$. $\boldsymbol{u}_{E_1}$ is the net force effect and $\boldsymbol{u}_{E_2}$ is the net torque effect of the reaction forces on the robot body. $\boldsymbol{u}_{E_3}$ stands for the effect of reaction forces on the robot joints. They are also sub-vectors of the generalized external force vector $\boldsymbol{u}_E$. τ is the generalized joint control vector, typically consisting of joint actuation torques for a robot with revolute joints. $\boldsymbol{H}_{11}, \boldsymbol{H}_{12}, \boldsymbol{H}_{21}$, and $\boldsymbol{H}_{22}$ are 3×3 matrices. $\boldsymbol{H}_{13}$ is $3 \times N$, $\boldsymbol{H}_{23}$ is $3 \times N$, $\boldsymbol{H}_{31}$ is $N \times 3$, $\boldsymbol{H}_{32}$ is $N \times 3$, and $\boldsymbol{H}_{33}$ is $N \times N$.

The expression in (4.1) reveals the importance of controlling the reactive force to control the body dynamics. It is the reaction force that governs the body dynamics. This is due to the fact that the body is not directly actuated. One can yet perceive reaction forces as the control effort for the body dynamics.

From (4.1), it can be noted that the body dynamics are given by

$$\begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} \end{bmatrix} \begin{pmatrix} \dot{\boldsymbol{v}}_B \\ \dot{\boldsymbol{w}}_B \end{pmatrix} + \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix} + \begin{bmatrix} \boldsymbol{H}_{13} \\ \boldsymbol{H}_{23} \end{bmatrix} \ddot{\theta} + \begin{pmatrix} \boldsymbol{u}_{E_1} \\ \boldsymbol{u}_{E_2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \tag{4.2}$$

(4.2) can be expressed in a more compact form as

$$\widetilde{\boldsymbol{H}}\dot{r} + \widetilde{\boldsymbol{b}} + \tau_{dis} = \tau_E \tag{4.3}$$

where

$$r = \begin{pmatrix} \boldsymbol{v}_B \\ \boldsymbol{w}_B \end{pmatrix}, \quad \widetilde{\boldsymbol{b}} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}, \quad \tau_{dis} = \begin{bmatrix} \boldsymbol{H}_{13} \\ \boldsymbol{H}_{23} \end{bmatrix} \ddot{\theta}, \quad \tau_E = -\begin{pmatrix} \boldsymbol{u}_{E_1} \\ \boldsymbol{u}_{E_2} \end{pmatrix} \tag{4.4}$$

In the proposed approach, off-line trajectories for the body coordinate frame center are generated. The reference curve in the x-direction has the shape of a ramp and the reference in the y-direction is a sinusoid. The z coordinate reference is a constant term, and the orientation reference for the body is fixed and parallel to the ground. Leg dynamics is neglected and reaction forces are considered as control input. Inverse dynamics is used to generate reference reaction forces, given the body trajectories.

First, tracking errors for the body are defined.

$$\begin{gathered} \begin{pmatrix} e_{pos\,body} \\ e_{rot\ body} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_{B\,ref} - \boldsymbol{p}_B \\ 0.5\left(n_{body} \times n_{body\,ref} + s_{body} \times s_{body\,ref} + a_{body} \times a_{body\,ref}\right) \end{pmatrix} \\ \begin{pmatrix} \dot{e}_{pos\,body} \\ \dot{e}_{rot\ body} \end{pmatrix} = \begin{pmatrix} \boldsymbol{v}_{B\,ref} - \boldsymbol{v}_B \\ \boldsymbol{w}_{B\,ref} - \boldsymbol{w}_B \end{pmatrix} \end{gathered} \tag{4.5}$$

where $e_{pos\,body}$ is the Cartesian position error for the body, and $e_{rot\ body}$ stands for the orientation error of the body. n_{body}, s_{body} and a_{body} are the column vectors of the body rotation matrix.

$$R_{body} = \begin{bmatrix} n_{body} & s_{body} & a_{body} \end{bmatrix} \tag{4.6}$$

A similar rotation matrix for the reference orientation of the foot is also defined

$$R_{body\,ref} = \begin{bmatrix} n_{body\,ref} & s_{body\,ref} & a_{body\,ref} \end{bmatrix} \tag{4.7}$$

The expression for computing orientation error for the body coordinate frame $0.5\left(n_{body} \times n_{body\,ref} + s_{body} \times s_{body\,ref} + a_{body} \times a_{body\,ref}\right)$ is, in fact, an approximate formula.

The cross products building up the orientation errors can be examined and seen to be approximating the orientation errors for small angles (Fig. 4.1.).

$$\begin{aligned}
n_{body} \times n_{body_{ref}} &\approx \sin\theta_s \, s_{body} + \sin\theta_a \, a_{body} \approx \theta_s \, s_{body} + \theta_a \, a_{body} \\
s_{body} \times s_{body_{ref}} &\approx \sin\theta_a \, a_{body} + \sin\theta_n \, n_{body} \approx \theta_a \, a_{body} + \theta_n \, n_{body} \\
a_{body} \times a_{body_{ref}} &\approx \sin\theta_s \, s_{body} + \sin\theta_n \, n_{body} \approx \theta_s \, s_{body} + \theta_n \, n_{body}
\end{aligned} \tag{4.8}$$

Therefore,

$$\begin{aligned}
&0.5\left(n_{body} \times n_{body_{ref}} + s_{body} \times s_{body_{ref}} + a_{body} \times a_{body_{ref}}\right) \\
&\qquad \approx \theta_n \, n_{body} + \theta_s \, s_{body} + \theta_a \, a_{body}
\end{aligned} \tag{4.9}$$

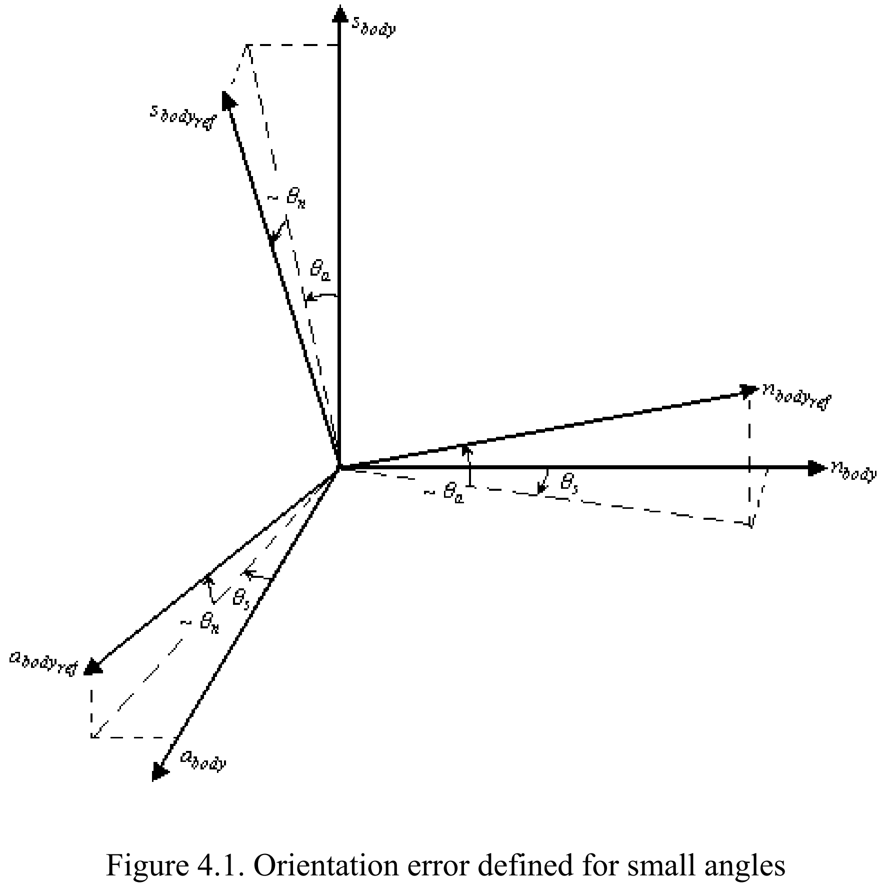


Figure 4.1. Orientation error defined for small angles

After properly defining tracking errors for the body, a control law of the following form reduces the non-linear plant into second order dynamics,

$$v = K_0 \begin{pmatrix} e_{pos_{body}} \\ e_{rot_{body}} \end{pmatrix} + K_1 \begin{pmatrix} \dot{e}_{pos_{body}} \\ \dot{e}_{rot_{body}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{v}_{B_{ref}} \\ \boldsymbol{w}_{B_{ref}} \end{pmatrix} \tag{4.10}$$

$$\begin{pmatrix} \boldsymbol{u}_{E_{1ref}} \\ \boldsymbol{u}_{E_{2ref}} \end{pmatrix} = \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} \end{bmatrix} v + \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix} \tag{4.11}$$

where K_0 and K_1 are parameters of the desired dynamics defined by the control designer.

Referring to (3.34), the reference body forces and torques found above can be related to references for the ground interaction forces by the equation (Fig. 4.2.)

$$\begin{pmatrix} \boldsymbol{u}_{E_{1ref}} \\ \boldsymbol{u}_{E_{2ref}} \end{pmatrix} = \boldsymbol{K}'\boldsymbol{f}_{E_{ref}} \tag{4.12}$$

where $\boldsymbol{K}'$ is the $6 \times (3M)$ sub-matrix of $\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}' \\ \boldsymbol{K}'' \end{bmatrix}$. The lower half of matrix $\boldsymbol{K}$ is named as $\boldsymbol{K}''$, which is $N \times (3M)$. Also note that $\boldsymbol{K}'$ is of the form

$$\boldsymbol{K}' = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{I}_3 & \boldsymbol{I}_3 & \boldsymbol{I}_3 \\ (\boldsymbol{d}+\boldsymbol{c}_1)\times & (\boldsymbol{d}+\boldsymbol{c}_2)\times & (\boldsymbol{d}+\boldsymbol{c}_3)\times & (\boldsymbol{d}+\boldsymbol{c}_4)\times \end{bmatrix} \tag{4.13}$$

where

$\boldsymbol{c}_j$: 3×1 position vector of j^{th} contact point with respect to the foot center as shown in Fig. 3.6.

$\boldsymbol{d}$: 3×1 position vector of foot center with respect to the base link coordinate frame

$\boldsymbol{I}_3$: 3×3 identity matrix

Therefore

$$\begin{pmatrix} \boldsymbol{u}_{E_1} \\ \boldsymbol{u}_{E_2} \end{pmatrix} = \boldsymbol{K}'\boldsymbol{f}_E = \begin{pmatrix} \sum\limits_{j \in M_A} \boldsymbol{f}_{E_j} \\ \boldsymbol{d} \times \sum\limits_{j \in M_A} \boldsymbol{f}_{E_j} + \sum\limits_{j \in M_A} \boldsymbol{c}_j \times \boldsymbol{f}_{E_j} \end{pmatrix} \tag{4.14}$$

(4.12) is solved to obtain the references for the forces at the foot. Note, however, that (4.12) poses an under-determined system of equations. Further, not every solution to the system is physically feasible due to the non-attractive nature of the contact, and no-slip condition to be satisfied for a successful walk.

Let $\boldsymbol{f}_{E_i} = \lfloor \boldsymbol{f}_{E_{ix}} \quad \boldsymbol{f}_{E_{iy}} \quad \boldsymbol{f}_{E_{iz}} \rfloor \quad i \in \{1,2,....,8\}$; with this definition, the constraints can be formulized as follows.

Non-attractive nature of the contact requires

$$\boldsymbol{f}_{E_{iz}} \geq 0, \quad \forall\, i \in \{1,2,...,8\} \tag{4.15}$$

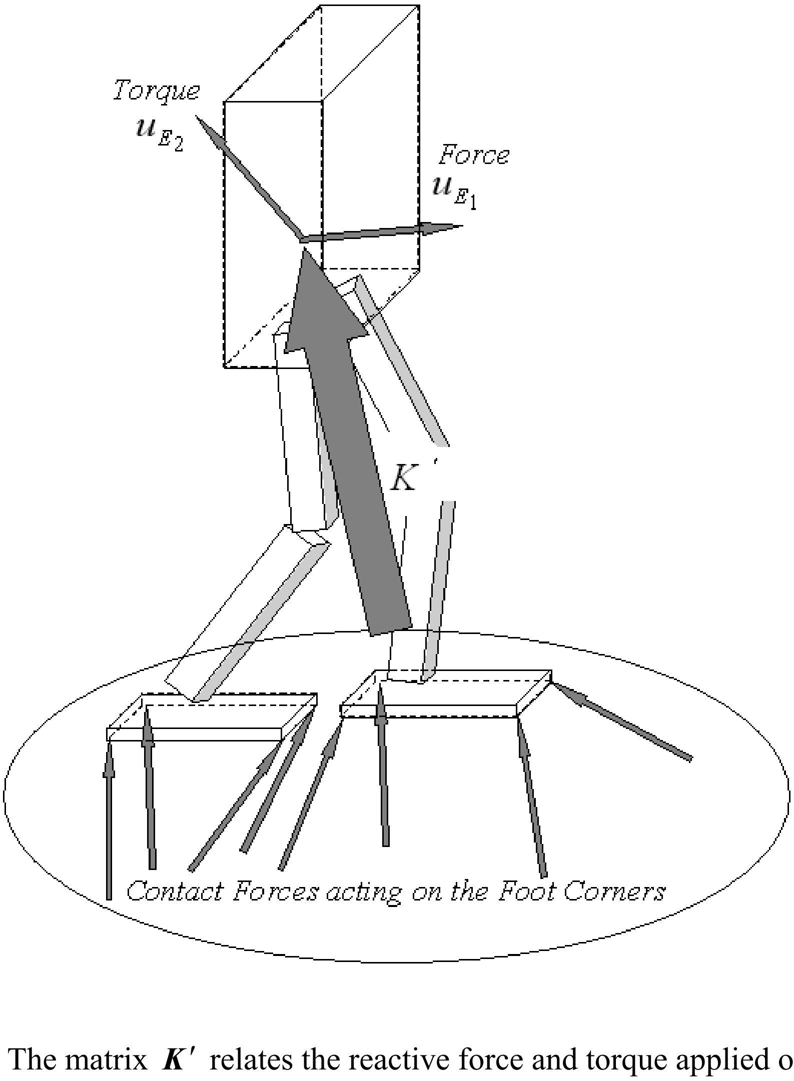


Figure 4.2. The matrix $\boldsymbol{K'}$ relates the reactive force and torque applied on the robot body to the ground interaction forces on the foot corners.

The constraint for no-slip is

$$-\mu \leq \frac{\sqrt{\boldsymbol{f}_{E_{ix}}{}^2 + \boldsymbol{f}_{E_{iy}}{}^2}}{\boldsymbol{f}_{E_{iz}}} \leq \mu, \quad \forall\, i \in \{1,2,...,8\} \tag{4.16}$$

The inequality in (4.16) is a nonlinear constraint. The process of computing a set of ground interaction forces for the locomotion of the biped is based on optimization with constraints. The optimization process can be simplified by approximating the nonlinear constraint in (4.16) by

$$\begin{gathered} -\frac{\sqrt{2}}{2}\mu \leq \frac{\boldsymbol{f}_{E_{ix}}}{\boldsymbol{f}_{E_{iz}}} \leq \frac{\sqrt{2}}{2}\mu \\ -\frac{\sqrt{2}}{2}\mu \leq \frac{\boldsymbol{f}_{E_{iy}}}{\boldsymbol{f}_{E_{iz}}} \leq \frac{\sqrt{2}}{2}\mu, \quad \forall\, i \in \{1,2,...,8\} \end{gathered} \tag{4.17}$$

(4.17) is more conservative than (4.16), however it is a set of linear constraints and this simplifies the problem.

(4.15) and (4.17) can be cast into the more compact form below

$$A\boldsymbol{f}_{Eref} \leq 0 \tag{4.18}$$

where A is a 24×24 matrix obtained from these constraint inequalities.

With this formulation, the problem becomes obtaining $\boldsymbol{f}_E$ which minimizes

$$\frac{1}{2}\left\| \boldsymbol{K}'\boldsymbol{f}_{Eref} - \begin{pmatrix} \boldsymbol{u}_{E_{1ref}} \\ \boldsymbol{u}_{E_{2ref}} \end{pmatrix} \right\|^2 \tag{4.19}$$

subject to the constraint $A\boldsymbol{f}_{Eref} \leq 0$.

This linear constrained least squares problem is solved by sequential quadratic programming. The details of this optimization are omitted in this thesis. However, information on this optimization technique can be found in [39].

The solution of the optimization algorithm is directly transformed to joint torques by:

$$\boldsymbol{u}_{E_{3ref}} = \boldsymbol{K}''\boldsymbol{f}_{Eref} \tag{4.20}$$

At single support phases, appropriate portions of $\boldsymbol{K}'$ and $\boldsymbol{K}''$ are considered when solving (4.19) and (4.20). Therefore, either right or left $6 \times (3M/2)$ portion of $\boldsymbol{K}'$, and either right or left $N \times (3M/2)$ portion of $\boldsymbol{K}''$ is considered, depending on which leg is the supporting leg. At double support phase, however, full matrices $\boldsymbol{K}'$ and $\boldsymbol{K}''$ are utilized for solving for reaction forces.

A question may arise here on the feasibility of realizing any reference contact forces via actuating joints only, i.e. on the controllability of contact forces. Obviously, with 12 input torques and 24 contact force variables, one cannot arbitrarily assign contact forces. However, this is not necessary. Consider the single support case only. With $\boldsymbol{K}'$ being $6\times(3M/2)=6\times12$ and $\begin{pmatrix}\boldsymbol{u}_{E_{1ref}}\\ \boldsymbol{u}_{E_{2ref}}\end{pmatrix}$ being 6×1 there are infinitely many solutions to $\begin{pmatrix}\boldsymbol{u}_{E_{1ref}}\\ \boldsymbol{u}_{E_{2ref}}\end{pmatrix}=\boldsymbol{K}'\boldsymbol{f}_{E_{ref}}$. Let $\boldsymbol{f}_{E_{ref}}{}^{(1)}$ and $\boldsymbol{f}_{E_{ref}}{}^{(2)}$ be two different solutions to (4.12). Therefore, from (4.14)

$$\begin{pmatrix}\boldsymbol{u}_{E_{1ref}}{}^{(1)}\\ \boldsymbol{u}_{E_{2ref}}{}^{(1)}\end{pmatrix}=\begin{pmatrix}\boldsymbol{u}_{E_{1ref}}{}^{(2)}\\ \boldsymbol{u}_{E_{2ref}}{}^{(2)}\end{pmatrix}$$

$$\begin{pmatrix}\sum\limits_{j\in M_A}\boldsymbol{f}_{E_j}{}^{(1)}\\ \boldsymbol{d}\times\sum\limits_{j\in M_A}\boldsymbol{f}_{E_j}{}^{(1)}+\sum\limits_{j\in M_A}\boldsymbol{c}_j\times\boldsymbol{f}_{E_j}{}^{(1)}\end{pmatrix}=\begin{pmatrix}\sum\limits_{j\in M_A}\boldsymbol{f}_{E_j}{}^{(2)}\\ \boldsymbol{d}\times\sum\limits_{j\in M_A}\boldsymbol{f}_{E_j}{}^{(2)}+\sum\limits_{j\in M_A}\boldsymbol{c}_j\times\boldsymbol{f}_{E_j}{}^{(2)}\end{pmatrix} \tag{4.21}$$

$$\Rightarrow\sum_{j\in M_A}\boldsymbol{c}_j\times\boldsymbol{f}_{E_j}{}^{(1)}=\sum_{j\in M_A}\boldsymbol{c}_j\times\boldsymbol{f}_{E_j}{}^{(2)}$$

This means that two different solutions will have the same resultant force and moment about the foot center. By defining the transfer matrix $\boldsymbol{K}_{fc}$, which relates foot center reaction forces to joint torques

$$\boldsymbol{u}_{E_{3ref}}=\boldsymbol{K}_{fc}\boldsymbol{u}_{E_{fc}} \tag{4.22}$$

where $\boldsymbol{K}_{fc}$ is 6×6. (4.22) is one-to-one and into, apart from singular configurations. Hence, all solutions to (4.12) correspond to a single foot center force and torque, and joint reference torques. Whichever solution is selected by the optimization scheme, by (4.20), we will end up with the same joint torque reference.

However, with the application of the reference joint torque, only one of the solutions set will be realized. At this point, unconsidered compatibility equations come into play. Compatibility equations ensure that a displacement field is single-valued and continuous. With the application of the joint torque reference, a completely different contact force set may arise. However, required reaction force and torque will be generated on the body. So, the objective is not tracking the reference contact forces, which is impossible, yet, the reference contact forces complete its mission by delivering certain information.

After obtaining the reference reaction forces on joints, $\tau_{ref} = \boldsymbol{u}_{E_{3ref}} + \boldsymbol{b}_3$ will track reference reaction forces at steady state, considering the leg dynamics:

$$\boldsymbol{H}_{33}\ddot{\theta} + \begin{bmatrix} \boldsymbol{H}_{31} & \boldsymbol{H}_{32} \end{bmatrix} \begin{pmatrix} \dot{\boldsymbol{v}}_B \\ \dot{\boldsymbol{w}}_B \end{pmatrix} + \boldsymbol{b}_3 + \boldsymbol{u}_{E_3} = \tau \tag{4.23}$$

This chapter summarized how reactive force control is done on the supporting leg or legs to track certain body trajectory reference. However, biped walking consists of both swing and support phases. The next two chapters introduce a novel scheme for controlling the swing phase.

5. BIPED LOCOMOTION CONTROL VIA VIRTUAL GRAVITY FIELDS

The previous analysis summarizes how body posture control is done with the control of the reaction forces on supporting leg or legs.

A novel scheme is used for controlling the swing leg. For the swing leg, the dynamics are given as:

$$\hat{\boldsymbol{H}}_{33}\ddot{\theta} + \begin{bmatrix}\hat{\boldsymbol{H}}_{31} & \hat{\boldsymbol{H}}_{32}\end{bmatrix}\begin{pmatrix}\dot{\boldsymbol{v}}_B \\ \dot{\boldsymbol{w}}_B\end{pmatrix} + \hat{\boldsymbol{b}}_3 = \hat{\tau} \tag{5.1}$$

where $\hat{\boldsymbol{H}}_{33}$, $\hat{\boldsymbol{H}}_{31}$, $\hat{\boldsymbol{H}}$, $\hat{\boldsymbol{b}}_3$, and $\hat{\tau}$ denote the inertia matrices, bias vectors and joints corresponding to the swing leg, either left or right.

In this work, an artificial bias vector is computed for with gravity acceleration components g_x, g_y, and g_z with values different than the actual ones. In other words, gravity compensation is done for virtual g_x, g_y, and g_z parameters. No control action other than gravity compensation is taken in this chapter.

The component g_x is used as a constant term, g_y is a function of time and phase, and g_z is a function of time starting with a value higher than the original gravitational acceleration, then decaying with time to lower values, until the contact is well established with the ground and the robot gets into double support phase. Usually, the transition between the swing and support phases is problematic due to the instantaneous high forces generated during the contact initiation. Decaying gravity compensation yields a smoother transition between phases. The general structure of the control algorithm is shown in Fig. 5.1.

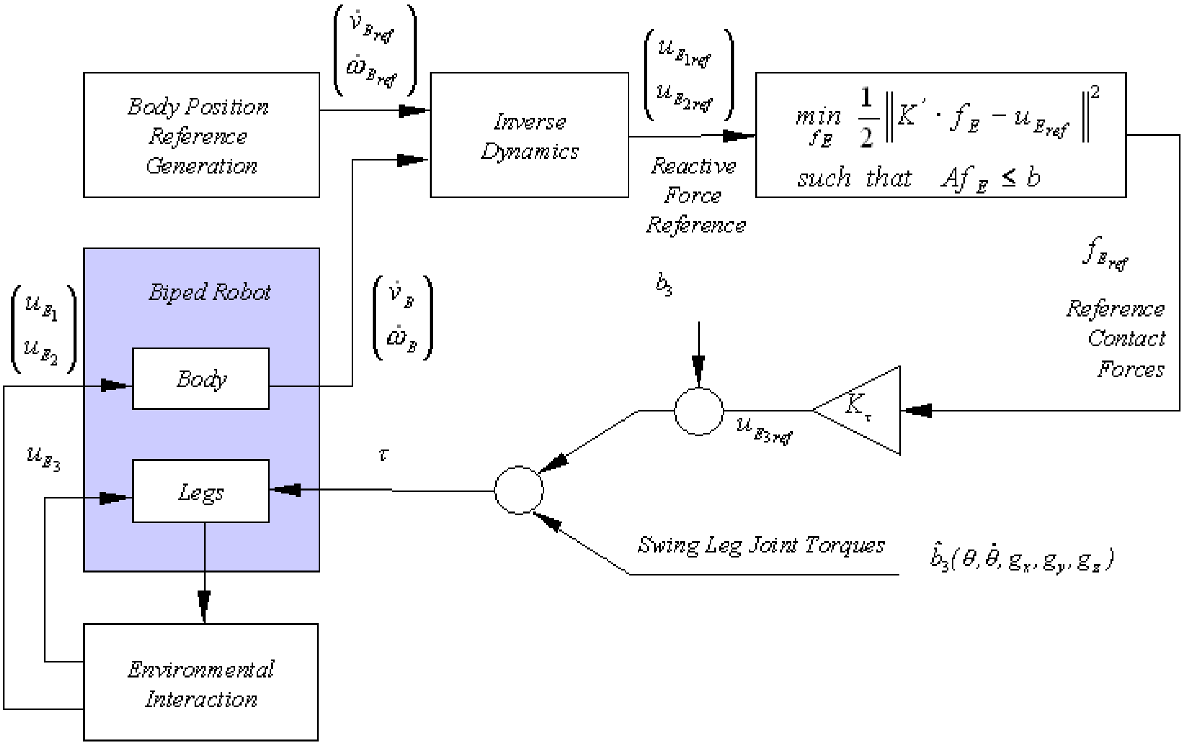

Body Position Reference Generation
$\begin{pmatrix}\dot{v}_{B_{ref}}\\ \dot{\omega}_{B_{ref}}\end{pmatrix}$
Inverse Dynamics
$\begin{pmatrix}u_{B_{1ref}}\\ u_{B_{2ref}}\end{pmatrix}$
$\min_{f_B} \frac{1}{2}\left\| K' \cdot f_B - u_{B_{ref}} \right\|^2$
such that $Af_B \le b$
Reactive Force Reference
$f_{B_{ref}}$
Reference Contact Forces
b_3
K_τ
$u_{B_{3ref}}$
Biped Robot
$\begin{pmatrix}u_{B_1}\\ u_{B_2}\end{pmatrix}$
Body
$\begin{pmatrix}\dot{v}_{B}\\ \dot{\omega}_{B}\end{pmatrix}$
u_{B_3}
Legs
τ
Swing Leg Joint Torques
$\hat{b}_3(\theta, \dot{\theta}, g_x, g_y, g_z)$
Environmental Interaction

Figure 5.1. The control algorithm based on the computation of the reactive force and virtual gravity compensation

6. BIPED LOCOMOTION CONTROL VIA HYBRID POSITION CONTROL AND GRAVITY COMPENSATION

The previous scheme of gravity compensation provides infinite compliance for the swing foot, and is promising in eliminating impact forces curing the contact transients. However, pure gravity compensation is quite open-loop, and it may be difficult or even impossible to tune the virtual gravitational acceleration parameters for a stable walk. Instead, applying position control strategies for certain directions is more natural.

In this work, the swing leg control consists of gravity compensated and position controlled modes, which are determined by a selection matrix, as in a hybrid force control scheme. First, reference transformation matrix for the swing foot is imposed. When the right foot is swung, we have

$$T_{R_{ref}} = \begin{bmatrix} 0 & 0 & 1 & -0.05 + x_{ref_{body}} + d \\ 0 & 1 & 0 & -0.085 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} R_{R\,ref} & & & p_{R\,ref} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

Similarly, for the left foot in the swing phase, reference transformation matrix is

$$T_{L_{ref}} = \begin{bmatrix} 0 & 0 & 1 & -0.05 + x_{ref_{body}} + d \\ 0 & 1 & 0 & 0.085 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} R_{L\,ref} & & & p_{L\,ref} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

Here, the 3×3 upper left sub-matrix of the transformation matrix is the reference rotation matrix ($R_{R\,ref}$ or $R_{L\,ref}$), and the first three elements of the fourth column ($p_{R\,ref}$ or $p_{L\,ref}$), are the reference positions for the swing foot.

The value -0.05 is a constant off-set between body and foot center frame centers in the sagittal plane , $x_{ref_{body}}$ is the body position reference in the sagittal plane, and d is a scalar accounting for the fact that the swing foot has to precede the body at the transition to the support phase. Therefore, it is a positive scalar when walking forward, and negative when walking backward. Also, it should increase with increasing walking speed.

After obtaining a reference transformation for the swing leg, position and orientation errors are obtained. By dropping the subscripts R and L (for the right and left feet, respectively) for simplicity and concentrating on the swing leg, the following expression can be used for the foot position and orientation errors.

$$\begin{pmatrix} e_{pos\,foot} \\ e_{rot\,foot} \end{pmatrix} = \begin{pmatrix} p_{foot_{ref}} - p_{foot} \\ 0.5\left(n_{foot} \times n_{foot_{ref}} + s_{foot} \times s_{foot_{ref}} + a_{foot} \times a_{foot_{ref}}\right) \end{pmatrix} \tag{6.3}$$

where $e_{pos_{foot}}$ is the Cartesian position error for the foot center, and $e_{rot\,foot}$ stands for the orientation error of the foot. n_{foot}, s_{foot} and a_{foot} are the column vectors of the swing foot rotation matrix.

$$R_{foot} = \lfloor n_{foot} \quad s_{foot} \quad a_{foot} \rfloor \tag{6.4}$$

A similar rotation matrix for the reference orientation of the foot is also defined

$$R_{foot_{ref}} = \lfloor n_{foot_{ref}} \quad s_{foot_{ref}} \quad a_{foot_{ref}} \rfloor \tag{6.5}$$

By multiplying these errors with selection matrix, we obtain the position errors in position-controlled directions. The selection matrix, which is denoted by S in this paper, is of the following form

$$S = diag\lfloor s_1, \mathrm{s}_2, \mathrm{s}_3, \mathrm{s}_4, \mathrm{s}_5, \mathrm{s}_6 \rfloor, \qquad s_1,\ldots s_6 \in [0,1] \tag{6.6}$$

However, unlike the conventional usage, the selection matrix is not comprised of zeros and ones only. There can be other values on its diagonal depending on the stiffness required in that direction. An element of the selection matrix being one means that the position error in that direction is directly reflected via the Jacobian relation to joint error. A value between zero and one means that the error is scaled, and the scaled error is reflected to joint space position errors. Zero elements mean that there is no position control in that direction.

In our proposed approach, s_3 is zero, which indicates that there is no swing foot position control in the z -direction. This direction is actuated with virtual gravity compensation only. Also, s_4 and s_5 are lower than other elements. They correspond to the rotational degrees of freedom with rotational axes in the x and y -directions, respectively. The reason for requiring less stiffness in these directions is that position errors in these directions are admissible, and even preferable. During the swing phase, some tracking error may occur as a part of the walking, and become undone naturally, after the leg gets into support phase.

After the position errors in the position controlled directions are found, corresponding joint position errors are obtained by

$$\begin{aligned} e_S &= S\begin{pmatrix} e_{pos} \\ e_{rot} \end{pmatrix} \\ \dot{e}_S &= S\begin{pmatrix} \dot{e}_{pos} \\ \dot{e}_{rot} \end{pmatrix} \end{aligned} \tag{6.7}$$

$$\begin{aligned} e_\theta &\approx K_{fc}^{T^{-1}} e_S \\ \dot{e}_\theta &= K_{fc}^{T^{-1}} \dot{e}_S \end{aligned} \tag{6.8}$$

To be noted here is that, the position errors are specified for the foot centers. Therefore, another Jacobian as in (4.22) has to be calculated that relates foot center velocities and joint velocities, unlike the reaction force control scheme, where we utilized a Jacobian relating foot corner velocities and joint velocities.

A PD based control is used to track the joint references.

$$\tau_{pc} = K_p e_\theta + K_d \dot{e}_\theta \tag{6.9}$$

This argumentation completes the position control scheme for the swing foot. Yet, another scheme is used for the z -axis position control. An artificial bias vector is computed for with gravity acceleration component g_z with value different than the actual one. In other words, gravity compensation is done for virtual g_z parameter. The joint torque for gravity compensation is simply

$$\tau_{gc} = \hat{\boldsymbol{b}}_3(\theta, \dot{\theta}, g_z) \tag{6.10}$$

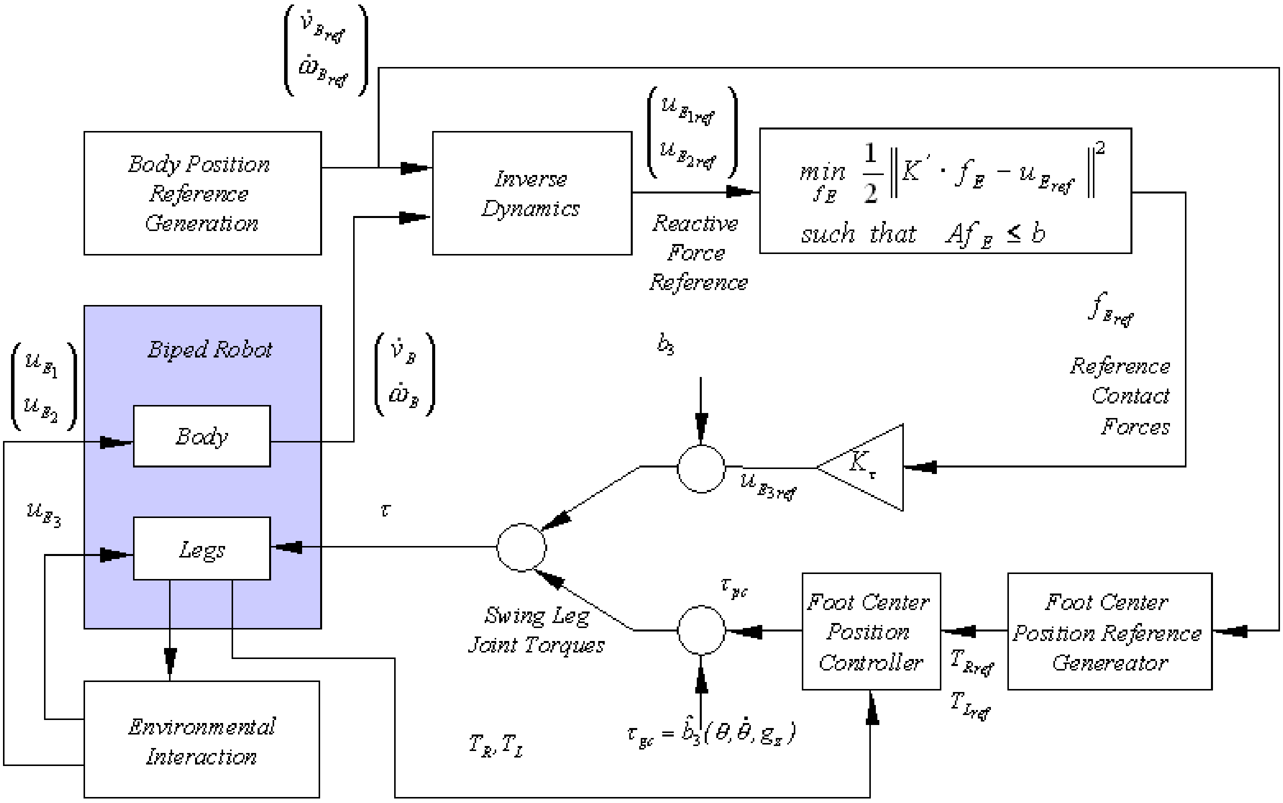
The component g_z is a function of time starting with a value higher than the original gravitational acceleration, then decaying with time to lower values, until the contact is well established with the ground and the robot gets into double support phase.

Usually, the transition between the swing and support phases is problematic due to the instantaneous high forces generated during the contact initiation. Decaying gravity compensation yields a smoother transition between phases.

Finally, the swing foot control torque is obtained by adding the torques found for position control and gravity compensation modes.

$$\tau_{swing\ leg} = \tau_{pc} + \tau_{gc} \tag{6.11}$$

The general structure of the control algorithm is shown in Fig. 6.1.



$\begin{pmatrix} \dot{v}_{B_{ref}} \\ \dot{\omega}_{B_{ref}} \end{pmatrix}$
Body Position Reference Generation
Inverse Dynamics
$\begin{pmatrix} u_{B_{1ref}} \\ u_{B_{2ref}} \end{pmatrix}$
Reactive Force Reference
$\min_{f_B} \frac{1}{2} \left\| K' \cdot f_B - u_{B_{ref}} \right\|^2$
such that $Af_B \le b$
$f_{B_{ref}}$
Reference Contact Forces
Biped Robot
Body
Legs
$\begin{pmatrix} u_{B_1} \\ u_{B_2} \end{pmatrix}$
u_{B_3}
$\begin{pmatrix} \dot{v}_B \\ \dot{\omega}_B \end{pmatrix}$
b_3
K_τ
$u_{B_{3ref}}$
τ
Swing Leg Joint Torques
τ_{pc}
Foot Center Position Controller
Foot Center Position Reference Genereator
$T_{B_{ref}}$
$T_{L_{ref}}$
Environmental Interaction
T_B, T_L
$\tau_{gc} = \hat{b}_3(\theta, \dot{\theta}, g_z)$

Figure 6.1. The control algorithm based on the computation of the reactive force and swing foot hybrid control

7. SIMULATION RESULTS

In this chapter, the simulation of the proposed control scheme with the 12 dof biped model described in Chapter 2 is discussed.

The simulations are carried implemented in Simulink with sampling time of 0.5 milliseconds with Euler integration. The simulation scheme is similar to the one in [38], which generalizes the recursive dynamic modelling method in [36,40] to the tree structure. The details of the simulation algorithm can be found in [37]. The simulation code for the hybrid position and gravity compensation described in Chapter 6 is given in Appendix A-E. In order to visualize the walking, simulation results are animated using an OpenGL based animation environment. A snapshot of the animation is shown in Fig. 7.1.

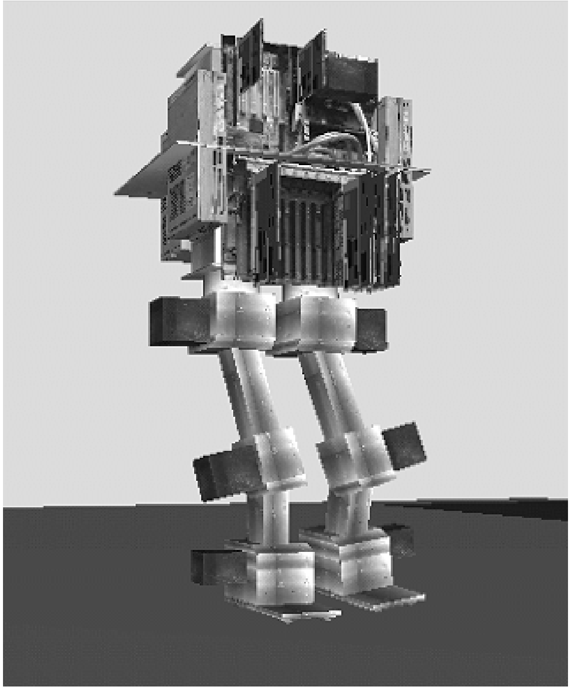


Figure 7.1. The biped robot animated in 3D.

7.1. Reactive Force Control Scheme

To test the efficiency of the reactive force control scheme, a sequence of step input position references for body position and orientation are entered. The robot is required to follow some reference input at the double-support stance (Fig. 7.2.). The step position inputs are such that the robot's center of gravity is contained within the supporting polygon.

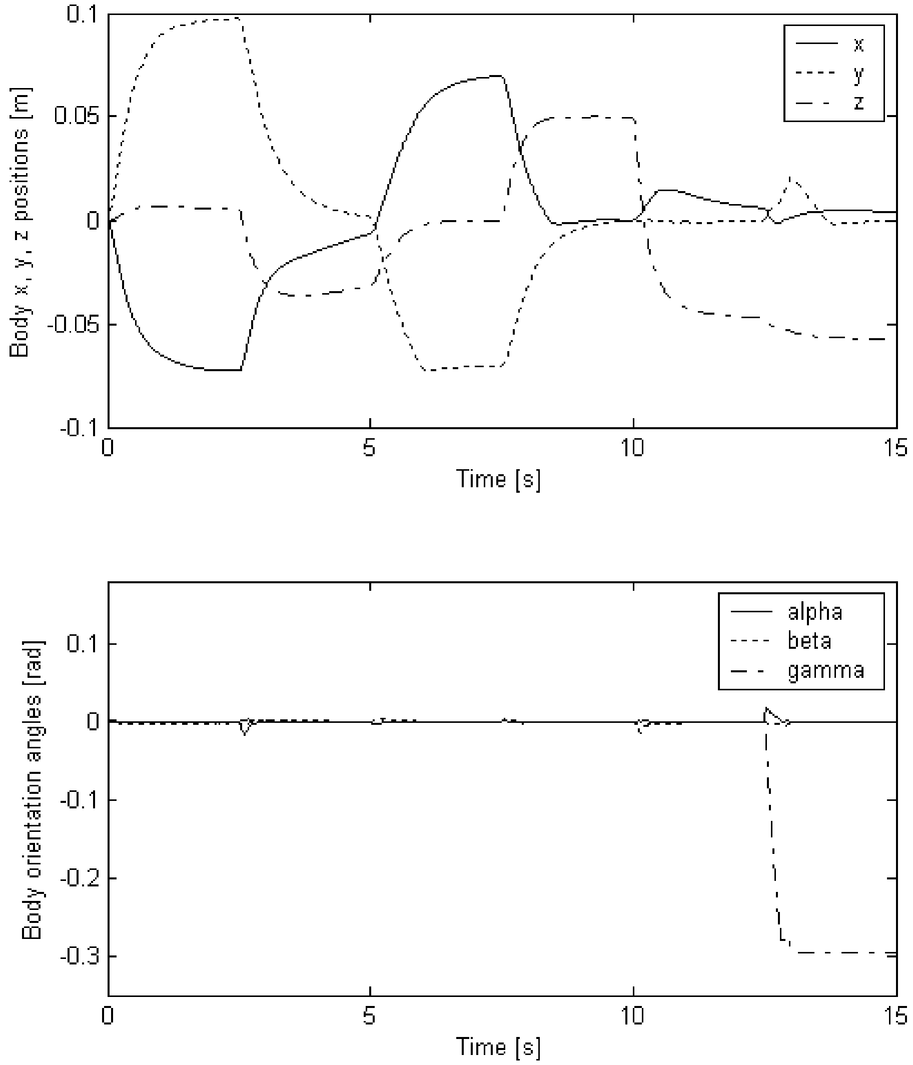


Figure 7.2. Body position and orientation trajectories at double support.

7.2. Biped Locomotion Control Via Virtual Gravity Fields

The matrices K_0 and K_1 describing the desired trunk dynamics are chosen as

$$\begin{aligned} K_1 &= diag[36, 36, 100, 400, 400, 1600] \\ K_0 &= diag[12, 12, 20, 40, 40, 80] \end{aligned} \tag{7.1}$$

Fig. 7.3. shows the off-line generated reference position curves for the trunk together with the actual trajectories obtained in the simulations. The position reference in the x-direction is in the form of a combination of a ramp with a step function. This is the walking direction of the robot. The y-direction reference trajectory is a sinusoid and it is synchronized with the gravity compensation phases of the swinging leg.

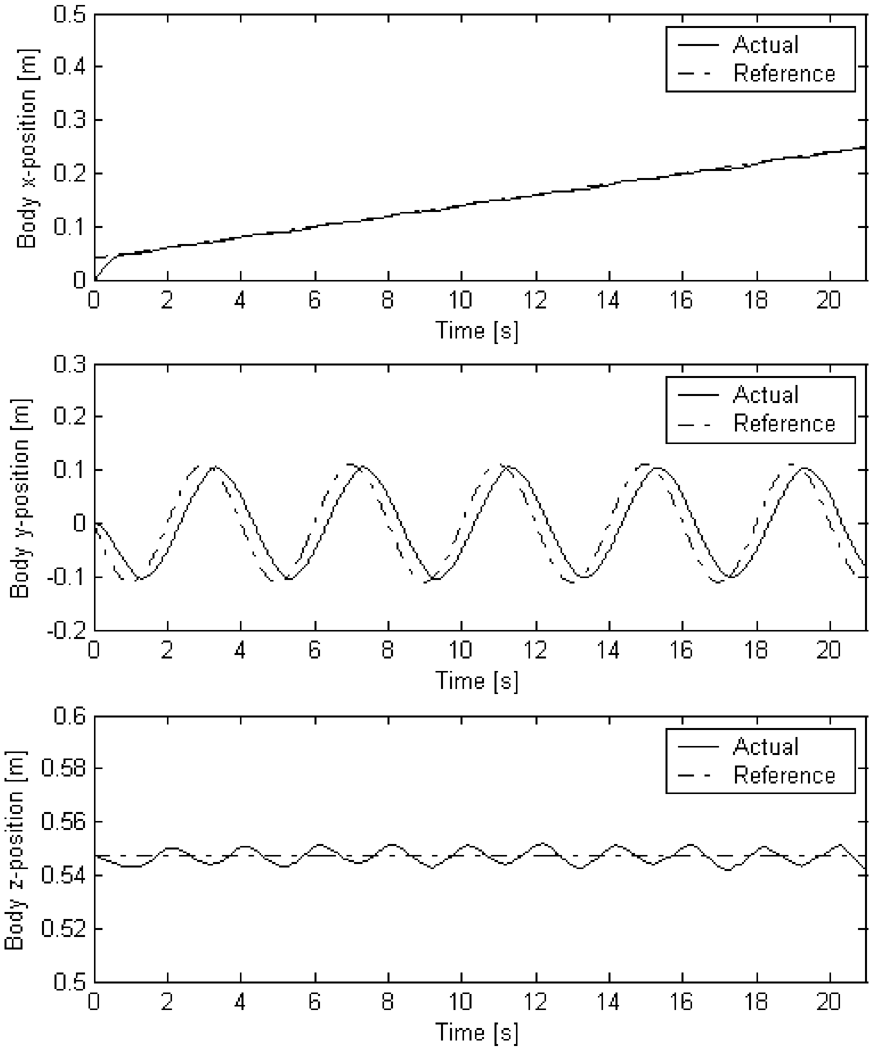


Figure 7.3. Body Cartesian trajectory components in the x, y and z directions.

The height of the body is aimed to be kept constant. The slope of the ramp, and the amplitude and frequency of the sinusoid are obtained by trail and error. It can be noted that these reference curves are tracked closely.

The components of the foot coordinate frame center trajectories are shown in Fig. 7.4., 7.5. and 7.6. Also, the artificial gravity acceleration curves in the x, y and z directions are displayed in these figures. The foot frame center coordinates indicate that the robot walks with a steady gait. In Fig. 7.4., a fixed artificial gravity at 0.4 m/s^2 is displayed after the first step. This step is generated by the function

$$\gamma_x = 0.2\rho_1 + 0.2 \tag{7.2}$$

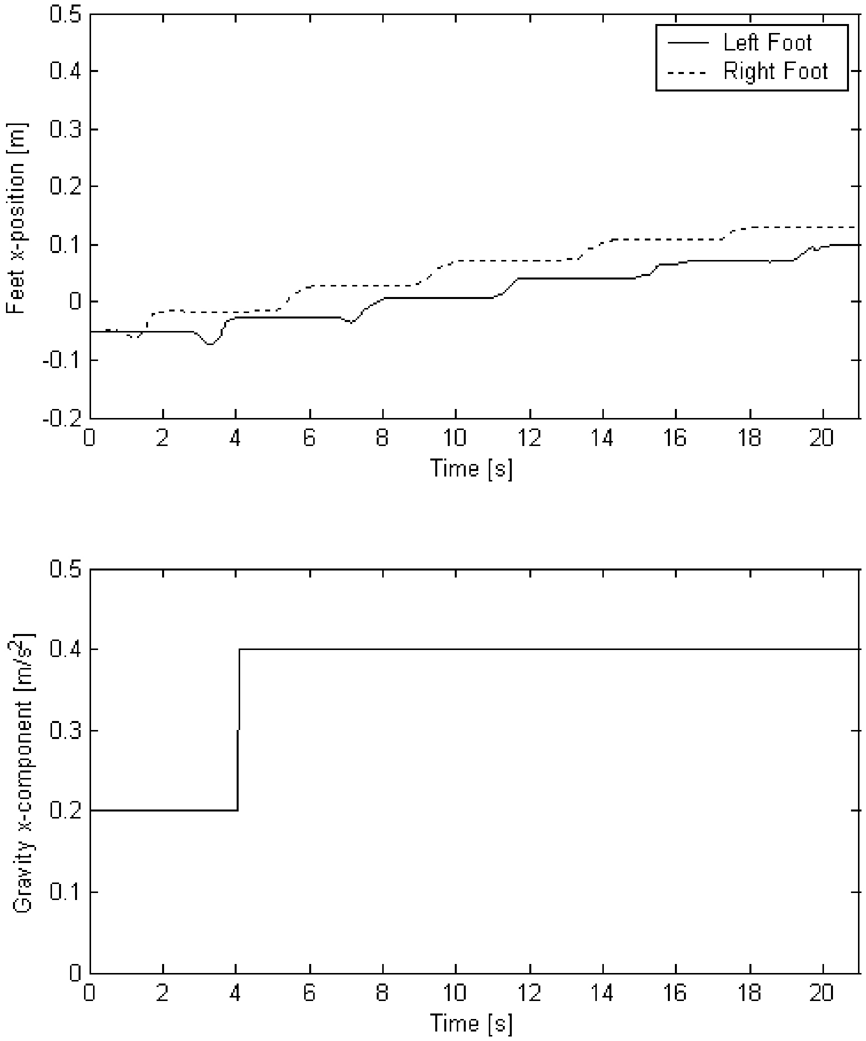


Figure 7.4. x-components of the foot trajectories and artificial gravity component.

where γ_x is the magnitude of the gravity-like effect created in the x-direction and ρ_1 is defined as

$$\rho_1 = \begin{cases} 0 & \text{before the end of the first step} \\ 1 & \textit{else} \end{cases} \tag{7.3}$$

Note also that, this effect only acts when a leg is in the swing phase, due to the architecture of the control algorithm shown in Fig. 5.1. and explained in Chapter 5. The artificial gravity effect in the y-direction is presented in Fig. 7.5. This effect is realized by the following function

$$\gamma_y = 0.08(\rho_l - \rho_r)\cos(0.3t)\exp(-0.15t + 0.02) \tag{7.4}$$

where γ_y is the generated gravity component in the y-direction and t is the time. ρ_r and ρ_l are defined as

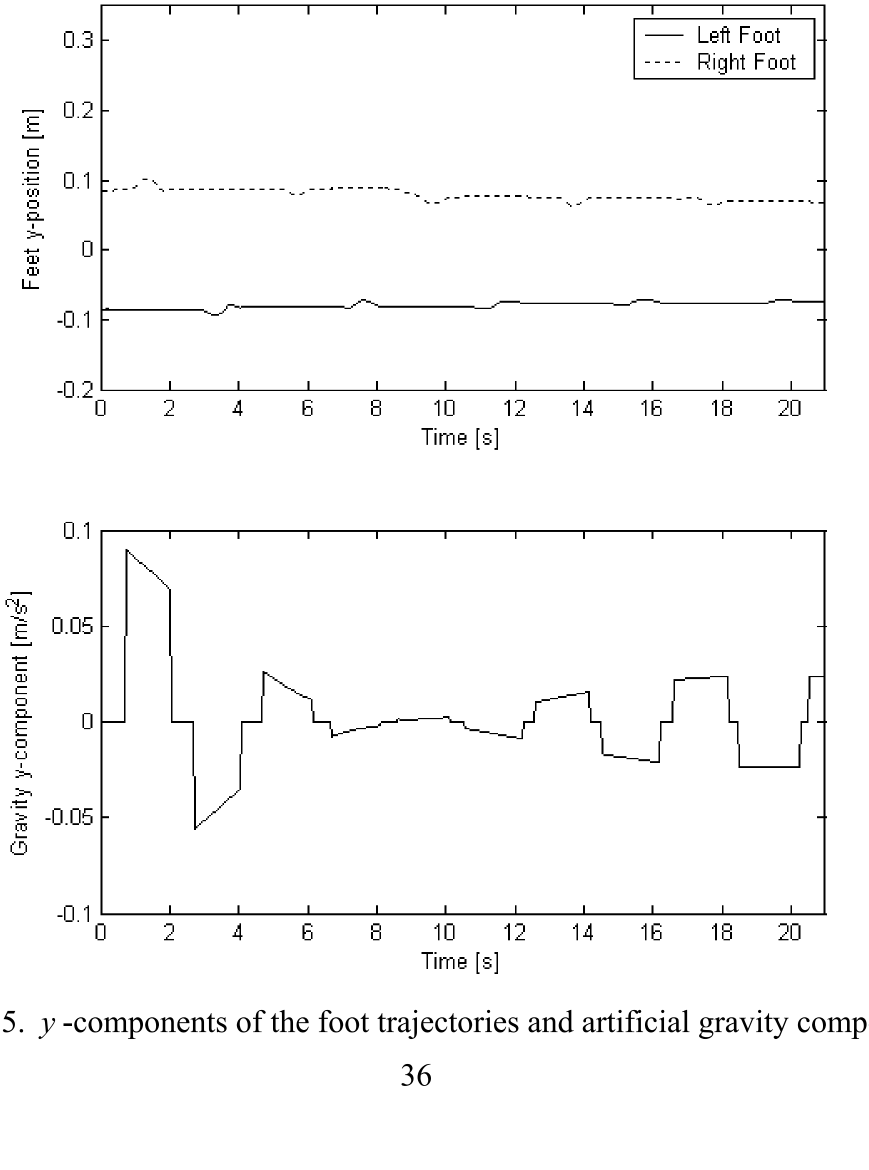


Figure 7.5. y-components of the foot trajectories and artificial gravity component.

$$\rho_r = \begin{cases} 0 & \text{in the right support phase} \\ 1 & \textit{else} \end{cases}$$

$$\rho_l = \begin{cases} 0 & \text{in the left support phase} \\ 1 & \textit{else} \end{cases} \qquad (7.5.)$$

With the introduction of the variables ρ_r and ρ_l the y-component of the added gravity effect acts on the right or the left foot selectively.

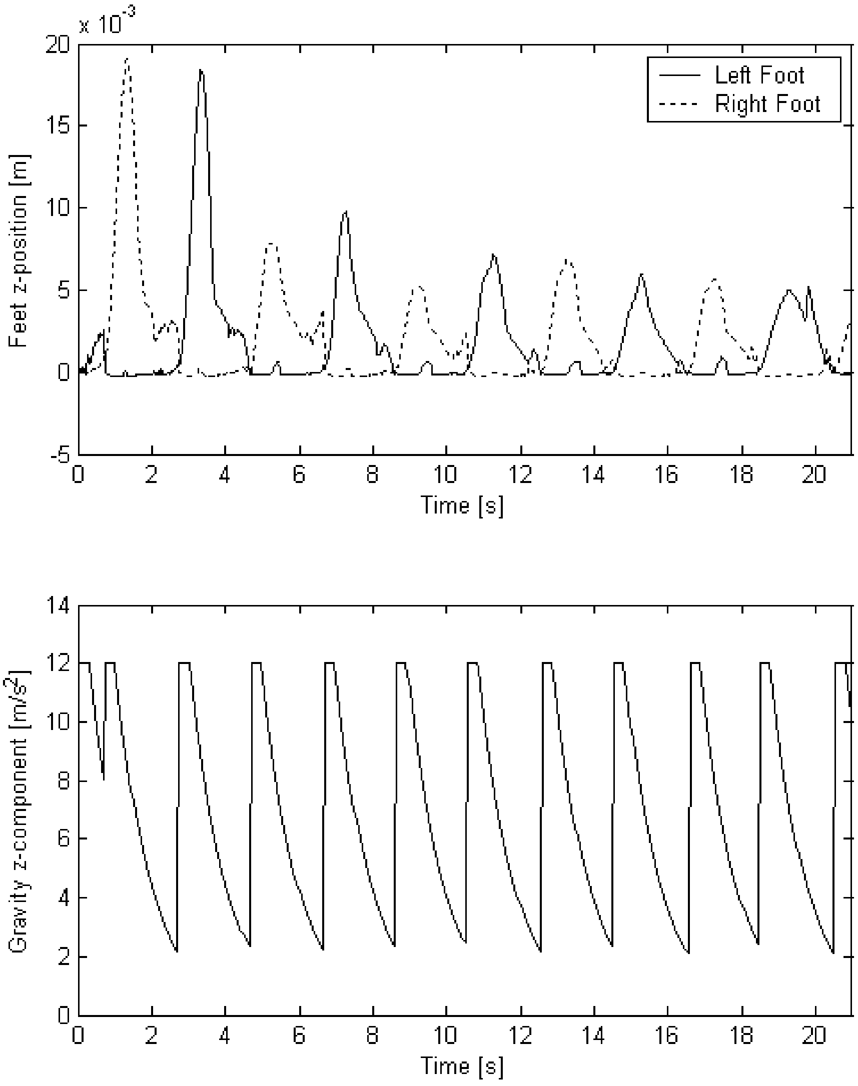


Figure 7.6. z-components of the foot trajectories and artificial gravity component.

Fig. 7.6. shows the z-component of the artificial gravity effect. The function, which generates the gravity curve, is

$$\begin{aligned}\gamma_z &= 12\left[1 - sign(t - \max(t_r, t_l) - 0.3)\right]/2 \\ &+ 12\left[1 + sign(t - \max(t_r, t_l) - 0.3)\right] \\ &\qquad \exp(-t + \max(t_r, t_l) + 0.3)/2\end{aligned} \tag{7.6}$$

where t_r is the time when the last right support phase to time ends. t_l is defined similarly. Again, the function is applied to the right or the left foot selectively. The functions in (7.2-7.6) are obtained iteratively by observing the robot walk in a number of simulations.

7.3. Biped Locomotion Control Via Hybrid Position Control And Gravity Compensation

The matrices K_0 and K_1 describing the desired trunk dynamics are chosen as

$$\begin{aligned}K_1 &= diag\left[36, 36, 100, 400, 400, 1600\right] \\ K_0 &= diag\left[12, 12, 20, 40, 40, 80\right]\end{aligned} \tag{7.7}$$

The variables in these equations are explained previously. The variable $d = \pm 0.03$ depending on the direction of walk.

The selection matrix identifies the position-controlled directions. In our case it is

$$S = diag\left[0.3, 1, 0 ,0.6, 0.3, 1\right], \tag{7.8}$$

Foot center position controller is simply a PD controller, which accepts joint position and tracking errors. The parameters of the controller are

$$\begin{aligned}K_p &= diag\left[100, 100, 100, 100, 100, 100\right] \\ K_d &= diag\left[20, 20, 20, 4, 4, 4\right]\end{aligned} \tag{7.9}$$

The function, which generates the gravity curve, is

$$\begin{aligned}\gamma_z &= 12\left[1 - sign(t - \max(t_r, t_l) - 0.4)\right]/2 \\ &+ 12\left[1 + sign(t - \max(t_r, t_l) - 0.4)\right]\exp(-t + \max(t_r, t_l) + 0.4)/2\end{aligned} \tag{7.10}$$

where t_r is the time when the last right support phase to time ends. t_l is defined similarly. Again, the function is applied to the right or the left foot selectively. The functions in (7.7-7.10) are obtained iteratively by observing the robot walk in a number of simulations.

Fig. 7.7. shows the off-line generated reference position curves for the trunk together with the actual trajectories obtained in the simulations for a forward walk.

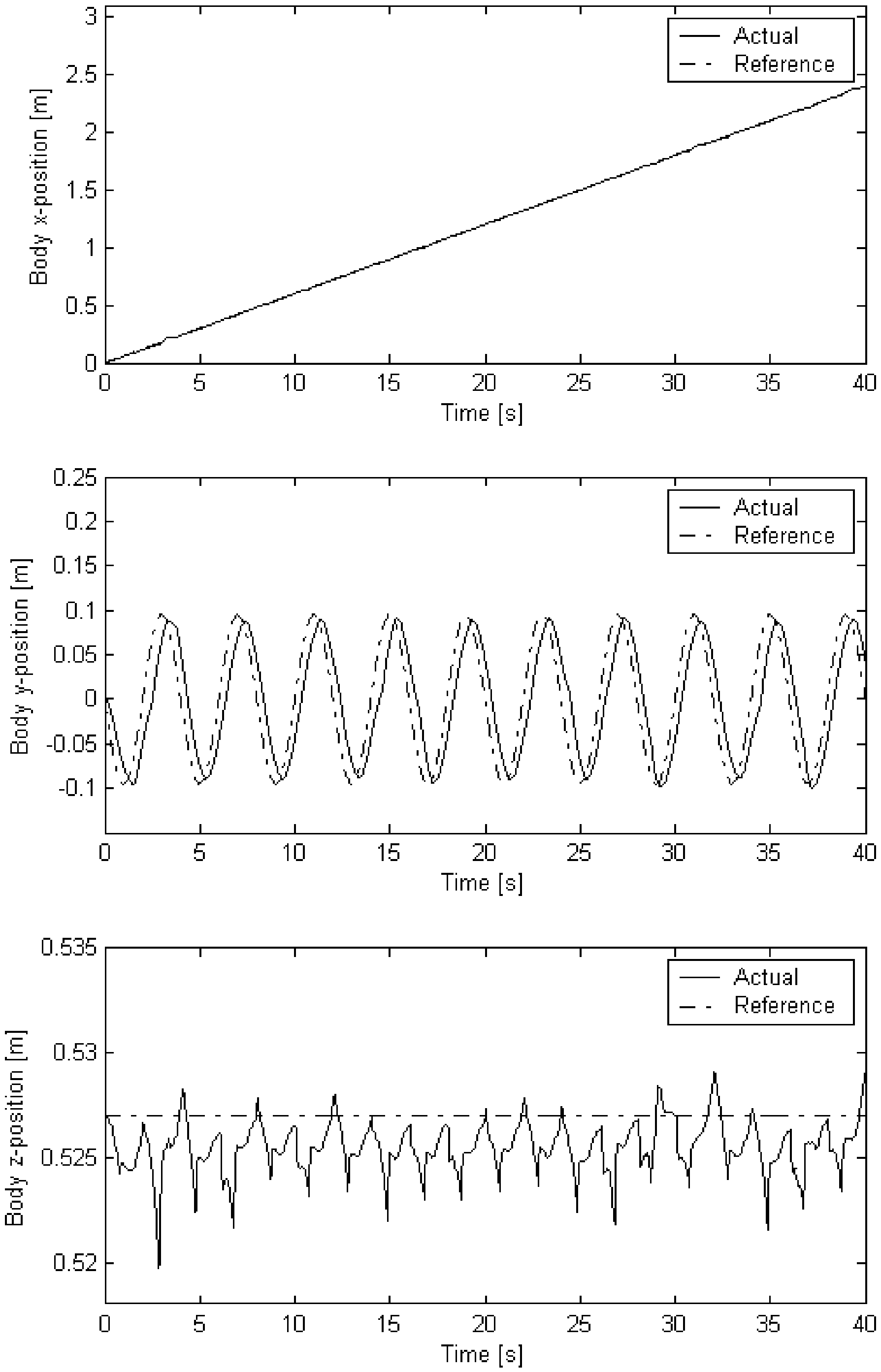


Figure 7.7. Body reference and actual positions in a forward walk.

The position reference in the x-direction is in the form of a ramp input. This is the walking direction of the robot. The y-direction reference trajectory is a sinusoid and it is synchronized with the leg swing. The height of the body is aimed to be kept constant. The amplitude and frequency of the sinusoid are obtained by trial and error. The components of the foot coordinate frame center trajectories are shown in Fig. 7.8. These trajectories indicate that the robot walks with a steady gait.

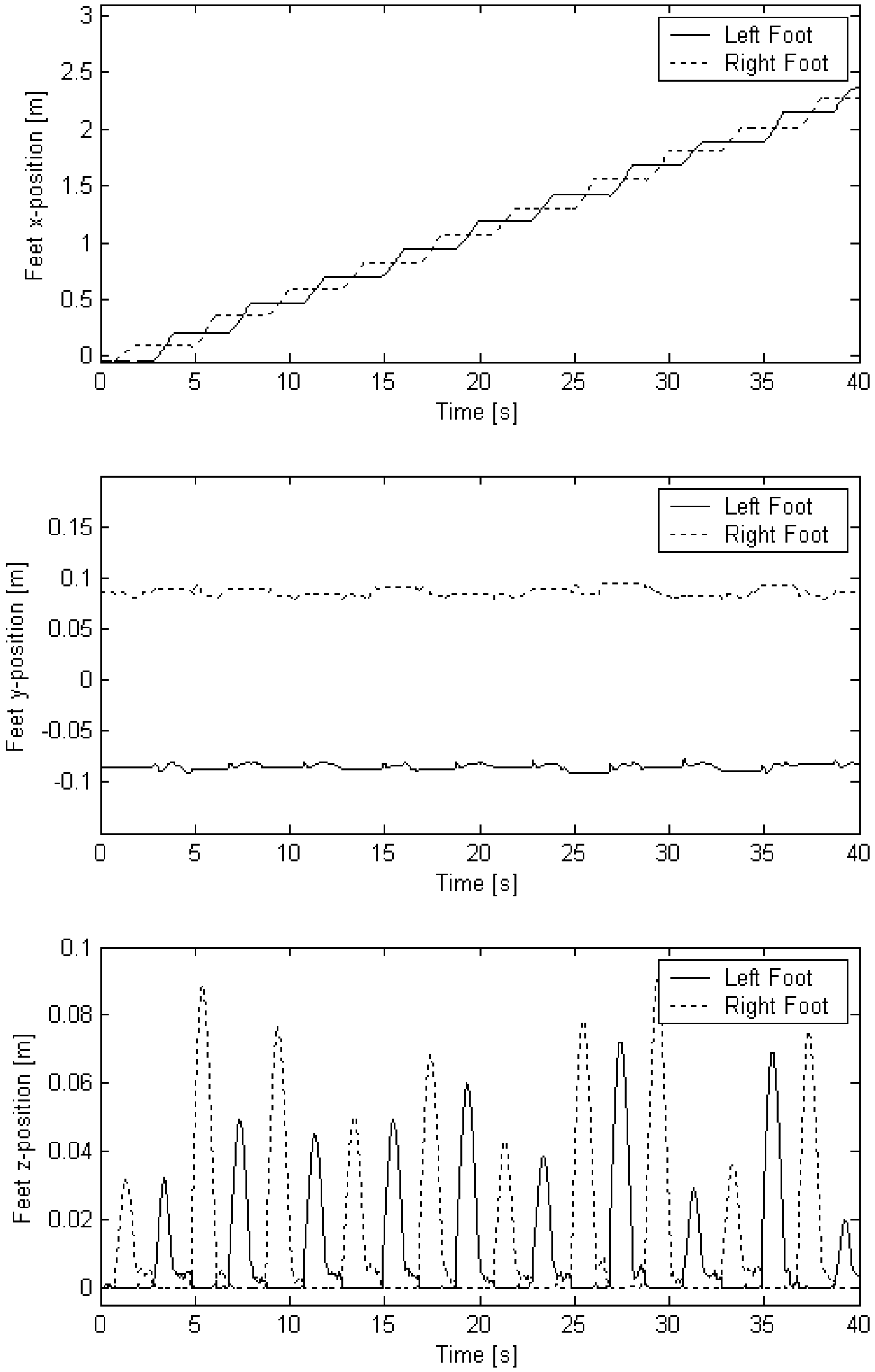


Figure 7.8. Position components of the foot trajectories in a forward walk.

Fig. 7.9. and 7.10. display similar results for a backward walk. The direction of the walk is simply adjusted by the body x-position reference and the sign of the variable d in (6.1.) and (6.2.).

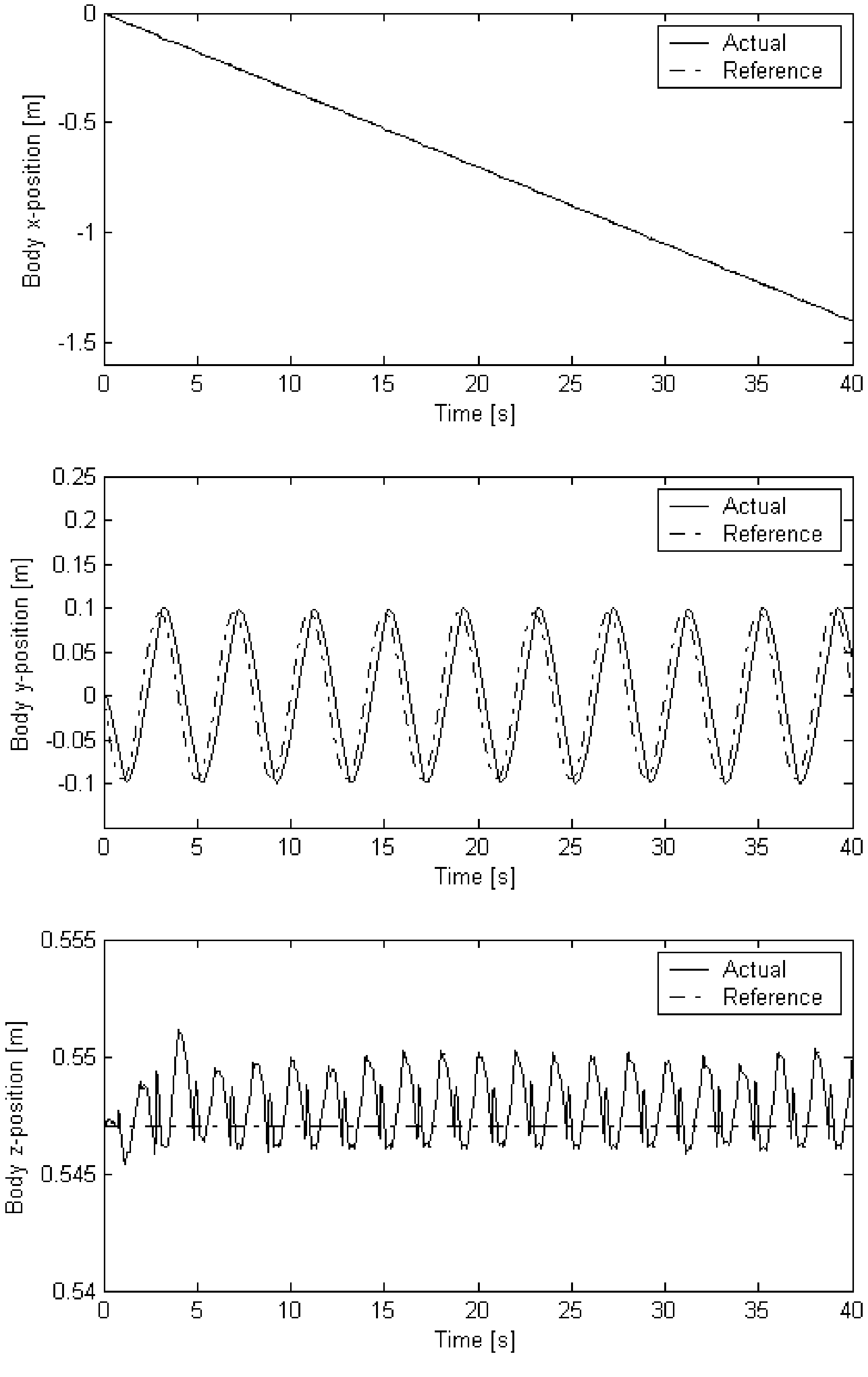


Figure 7.9. Body reference and actual positions in a backward walk.

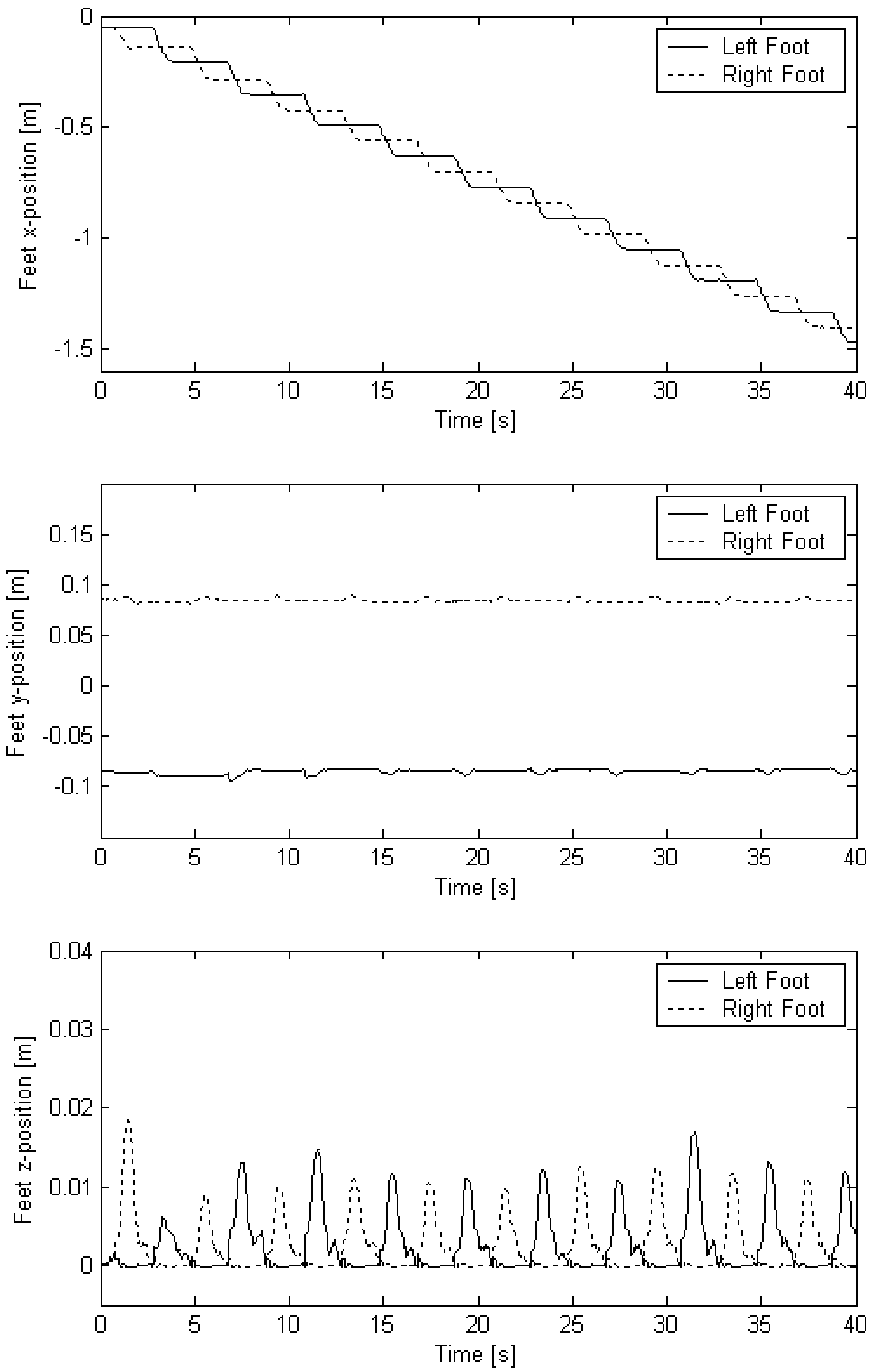


Figure 7.10. Position components of the foot trajectories in a backward walk.

By looking at the feet z -position, it is observed to be quite deviational since this direction is not controlled through position control. Also, impact forces during the contact phase become more pronounced by the hybrid position control and virtual gravity compensation, comparing body z -position tracking in Fig. 7.3. with Fig 7.7. and 7.9.

8. CONCLUSIONS

Biped robots attract the attention of researchers because of their possible uses in the human environment. Achieving complex tasks that human can do is a challenging work for the bipeds, and many problems regarding these are waiting to be solved. Many of the existing robots lack mobility, agility and adaptability. Although many robust techniques have been developed with on-line trajectory modification schemes, they generally require a very stiff position control scheme, which is not natural and typical of human beings and animals.

This work addresses the low impedance problem of the humanoid robots. The scheme generates off-line trajectories for the trunk of the robot. Suitable torques for the supporting leg joints for the tracking of the trunk reference trajectory are obtained by constrained optimization techniques and a reactive force control scheme. Step generation is achieved by a swinging leg control scheme, which is based on position control on certain directions, and artificial gravity compensation on certain directions. This method allows having infinite compliance at certain directions along which impact forces are expected to occur. The method also has the advantage of modifying the scheme easily to implement forward walking, backward walking, stair climbing, cornering, bending, kneeling down and standing-up.

Simulation results indicate that the methods are successful and a stable gait is obtained, although the trajectory for the trunk is off-line and do not include on-line modification terms.

The method does not require specific knowledge on total center of gravity, ZMP, angular momentum of the body, nor any force sensor feed-back. However, the inertia matrix, the Jacobian, and the bias vector, which contains gravitational, centrifugal, and Corioli's forces have to be known with high accuracy. As a future work, a disturbance observer can be employed and the method can be made robust to plant parameter uncertainties.

9. APPENDICES

Appendix A

Matlab script file for parameter initialization.

```
% parat.m.
% used for parameter initialization
% global parameters
global alpha a d
global n
global A0ipiSTAR
global A0isiHAT
global z0
global m
global I
global H
global bARo
global bALo
global A0nc1 A0nc2 A0nc3 A0nc4
global Ro_pSTAR_Ro Lo_pSTAR_Lo
global b_sHAT_b
global mb
global Ib
global vvector
global steptime
global xvector
global B_joint_friction_matrix
global b
global Kspring Bdamper Bhorizontalfriction
global K K_fc
global simulationcycle
global mu
global EnergyList
global KspringList
global minimumKspring fixedKspring maximumKspring
global elasticity_coefficient
global minimisation_on
global horizontal_on
global B_K_ratio
global ZLayerThickness
global groundforce
global integral_force
global impulse_container
global Kp Kd Ki
global qRinterror
global qLinterror
global avector
global jfvector
global uEvector
global ZMPxList ZMPyList
global qRref
global qRdotref
global qLref
global qLdotref
global qrefList
global time
global alphabetagamma
global xcontainer
global xyzanglescontainer
global qref
global initialflatangle
global initialbackangle
global stepdivider
global velslip
global checkenergycurve

global referencecycle
global zone1
global zone2
global forwardzmpsteptrigger
global gx
global gy
global gz

global ZMPx
global ZMPy

global dummyrefR_old
global dummyrefL_old

global body_height

global to_desktop_too

global uvector
global ulist

global damper_threshold

global time_long
global xcontainer_long
global xyzanglescontainer_long
global KspringList_long
global ZMPxList_long
global ZMPyList_long

global inverseH

global touch_on_off
global touch_on_off_OLD
global touch_xy_coordinates_array
global Khorizontalspring
global spring_threshold

global fE_list
fE_list=[];

global Alsqlin ublsqlin lblsqlin blsqlin beqlsqlin Aeqlsqlin
global lb_lsqnonneg  %lower bound on tangential contact force

global K0 K1
global TR TL  % Right and Left foot center transformation matrices
initialization

TR=[0    0.0000    1.0000   -0.0500;...
   -0.0000    1.0000   -0.0000   -0.0850;...
   -1.0000   -0.0000    0.0000   -0.0000;...
         0         0         0    1.0000];

TL=[0    0.0000    1.0000   -0.0500;...
   -0.0000    1.0000   -0.0000    0.0850;...
   -1.0000   -0.0000    0.0000   -0.0000;...
         0         0         0    1.0000];

global TR_fixed TL_fixed
global sfr_err_old sfl_err_old  % Swing foot position tracking error
initialization

sfr_err_old=zeros(6,1);
sfl_err_old=zeros(6,1);

TR_fixed=TR;  % Fixed orientation & position reference for the swing
foot
TL_fixed=TL;

global Kp_fc Kd_fc  % Swing foot joint PD controller parameters

Kp_fc=1*diag([100 100 100 100 100 100]);
Kd_fc=1*diag([20 20 20 4 4 4]);

global w_p_c  % Feet corner points position initialization

w_p_c=[-0.1003   0.1003   -0.1003    0.1003   -0.2203   -0.0197   -
0.2203  -0.0197;...
-0.0720   -0.0720   -0.1480   -0.1480    0.1480    0.1480    0.0720
0.0720;...
-0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000
-0.0000];

global trigger_1 trigger_2
trigger_1=0;
trigger_2=0;

global options  % options for optimization parameters
options=optimset('Display','off');

global g_trig  % Trigger for gravity compensation in sagittal plane
g_trig=1;

global ds_switch rs_switch ls_switch rs_time ls_time ds_time
% Parameters for identifying the walking phase
ds_switch=1;
rs_switch=1;
ls_switch=1;

rs_time=0;  % previous right support time
ls_time=0;  % previous left support time

%SIMULATION PARAMETERS
```

```
steptime=0.0005;
starttime=0;
stoptime=40.0;
simulationcycle=-1;
stepdivider=100;

%simulation time addition parameters
addtime=1;
xcontainer=[];
xyzanglescontainer=[];
groundforcecontainer=[];
groundforce=[];
impulse_container=zeros(10,8);
time=[];
alphabetagamma=zeros(3,1);
time_long=[];
xcontainer_long=[];
xyzanglescontainer_long=[];
KspringList_long=[];
ZMPxList_long=[];
ZMPyList_long=[];



%LINK PARAMETERS.
%NUMBER OF JOINTS.
n=6;
gx=0;
gy=0;
gz=9.80621;

%LINK LENGTS MASSES, GENERAL INFORMATION (BLOCK SHAPED LINKS ASSUMED).
Lb=0.2;
Wb=0.4;
Hb=0.5;
mb=50;


L1=0.1;
W1=0.1;
H1=0.1;
m1=5;

L2=0.1;
W2=0.1;
H2=0.1;
m2=3;

L3=0.27;
W3=0.05;
H3=0.1;
m3=4;

L4=0.22;
W4=0.05;
H4=0.1;
m4=0.5;

L5=0.1;
W5=0.1;
H5=0.1;
```

```
m5=0.5;

L6=0.087;
W6=0.12;
H6=0.25;
m6=5;

%LINK INERTIA MATRICES ABOUT CENTER OF MASS IN THE ASSOCIATED LINK
FRAME.
Ib=(mb/12)*[Wb^2+Hb^2 0 0;  0 Lb^2+Hb^2 0;  0 0 Lb^2+Wb^2];
I1=(m1/12)*[W1^2+H1^2 0 0;  0 L1^2+H1^2 0;  0 0 L1^2+W1^2];
I2=(m2/12)*[W2^2+H2^2 0 0;  0 L2^2+H2^2 0;  0 0 L2^2+W2^2];
I3=(m3/12)*[W3^2+H3^2 0 0;  0 L3^2+H3^2 0;  0 0 L3^2+W3^2];
I4=(m4/12)*[W4^2+H4^2 0 0;  0 L4^2+H4^2 0;  0 0 L4^2+W4^2];
I5=(m5/12)*[W5^2+H5^2 0 0;  0 L5^2+H5^2 0;  0 0 L5^2+W5^2];
I6=(m6/12)*[W6^2+H6^2 0 0;  0 L6^2+H6^2 0;  0 0 L6^2+W6^2];


%DENAVIT-HARTENBERG PARAMETERS (CONVENTIONAL NOTATION).
alpha=zeros(n,1);
a=zeros(n,1);
d=zeros(n,1);

alpha(1)=pi/2;
a(1)=0;
d(1)=0;

alpha(2)=-pi/2;
a(2)=0;
d(2)=0;

alpha(3)=pi/2;
a(3)=L3;
d(3)=0;

alpha(4)=0;
a(4)=L4;
d(4)=0;

alpha(5)=-pi/2;
a(5)=0;
d(5)=0;

alpha(6)=0;
a(6)=L6;
d(6)=0;



%LINK RELATED POSITION VECTOR COMPUTATIONS FOR THE LUH-WALKER-PAUL N-E
MECHANISM.
%NEXT COORDINATE FRAME ORIGIN LOCATION IN THE CURRENT FRAME.
A0ipiSTAR=zeros(3,n);
for i=1:n
   A0ipiSTAR(1:3,i)=[a(i); d(i)*sin(alpha(i));d(i)*cos(alpha(i))];
end

%CENTER OF MASS LOCATION OF THE CURRENT LINK IN THE CURRENT FRAME.
A0isHAT=zeros(3,n);

A0isiHAT(1:3,1)=[0;0;0];
```

```
A0isiHAT(1:3,2)=[0;0;0];
%A0isiHAT(1:3,3)=[-a(3)/2;0;0];
A0isiHAT(1:3,3)=[-a(3)/6;0;0];
%A0isiHAT(1:3,4)=[-a(4)/2;0;0];
A0isiHAT(1:3,4)=[-a(4)/6;0;0];
A0isiHAT(1:3,5)=[0;0;0];
%A0isiHAT(1:3,6)=[-a(6)/2;0;0];
A0isiHAT(1:3,6)=[-a(6)/6;0;0];

%LINK MASS VECTOR, DEFINED FOR PROGRAMMING CONVENIENCE.
m=[m1;m2;m3;m4;m5;m6];

%LINK INERTIA ARRAY, DEFINED FOR PROGRAMMING CONVENIENCE.
I=[I1 I2 I3 I4 I5 I6];

%GENERAL CONSTANTS
z0=[0;0;1];

%joint VISCOUS FRICTION COEFFICIENTS
Bfriction1=5;
Bfriction2=5;
Bfriction3=5;
Bfriction4=5;
Bfriction5=5;
Bfriction6=5;
B_joint_friction_matrix=diag([Bfriction1     Bfriction2     Bfriction3
Bfriction4 Bfriction5 Bfriction6 ...
                              Bfriction1     Bfriction2     Bfriction3
Bfriction4 Bfriction5 Bfriction6 ]);

%contact parameters
B_K_ratio=0;
Kspring=20000;
Bdamper=2000;
damper_threshold=0.005;
Bhorizontalfriction=0.1;
mu=0.7;
minimumKspring=1000000;
maximumKspring =10000000;
elasticity_coefficient=1;
fixedKspring=5000;
horizontal_on=1;
ZLayerThickness=0.5;
integral_force=zeros(8,1);
velslip=0.00001;
touch_on_off=zeros(1,8);
touch_on_off_OLD=zeros(1,8);
touch_xy_coordinates_array=zeros(2,8);
Khorizontalspring=1000;
spring_threshold=0.000;



%LEG ATTACHMENT FRAME TO BODY FRAME ROTATION MATRICES
bARo=[1  0  0;
      0  1  0;
      0  0  1];
bALo=bARo;

%CORNER COORDINATES IN FOOT LINK FRAME
A0nc1=[0;0.038;-0.0503];
```

```
A0nc2=[0;0.038;0.1503];
A0nc3=[0;-0.038;-0.0503];
A0nc4=[0;-0.038;0.1503];
%P_STARS FOR THE VIRTUAL LINKS BETWEEN BODY ORIGIN AND LEG ATTACMENT
POINTS
%Ro_pSTAR_Ro=[-0.1;0;Wb/2-H1/2-0.01];
%Lo_pSTAR_Lo=[-0.1;0;-Wb/2+H1/2+0.01];
Ro_pSTAR_Ro=[0;-0.085;0];
Lo_pSTAR_Lo=[0;0.085;0];

%THE BODY S_HAT
%b_sHAT_b=[0;0;Hb/2];
b_sHAT_b=[-0.05;0;Hb/2];

body_height = 0.27 + 0.22 + 0.087 - 0.05;   % Stand-up Posture

TRight=[0 0 1                -0.05; ...
        0 1 0                -0.00; ...
       -1 0 0                 -body_height-Ro_pSTAR_Ro(3); ...
        0 0 0 1];
dummyrefR_old=i_kine_analytic(TRight);
%dummyrefR_old=ikine(leg,TRight,[0   -pi/2    0    0    0   0]);

TLeft =[0 0 1                 -0.05; ...
        0 1 0                 0.00; ...
       -1 0 0                 -body_height-Lo_pSTAR_Lo(3); ...
        0 0 0 1];
dummyrefL_old=i_kine_analytic(TLeft);
%dummyrefL_old=ikine(leg,TLeft,[0   -pi/2   0   0    0    0]);

% initial conditions

avector=zeros(6+2*n,1);

vvector =    [[0 0 0] [0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0]]';

initialflatangle=pi*20/180;
initialbackangle=acos(2*cos(initialflatangle)-1);
%xvector = [[0 0 2*cos(initialflatangle)*0.4+0.02] ...
%          [1 0 0 0 1 0 0 0 1] ...
%          [0 -initialbackangle  initialbackangle 0 0 0] ...
%            [initialflatangle -2*initialflatangle initialflatangle 0 0
0] ]';

xvector =    [[0 0 body_height] [1 0 0 0 1 0 0 0 1] ...
             [dummyrefR_old] ...
             [dummyrefL_old]]';

uEvector=zeros(6+2*n,1);
jfvector=zeros(6+2*n,1);

ZMPxList=[];
ZMPyList=[];

%INERTIA MATRIX INITIALIZATION
H=treehinitializer(xvector);
inverseH=inv(H);

%external force transformation matrix (Jacobian Transpose)
K=treekinitializer(xvector);  % (for foot corners)
K_fc=treek_fc_initializer(xvector);  % (for foot center)
```

```
%bias vector
b=zeros(6+2*n,1);

%control initialization
uvector=zeros(6+2*n,1);
ulist=[];

%MINIMIZATION VARIABLES (ground force interaction)
EnergyList=[];
KspringList=[];
minimisation_on=1;
checkenergycurve =0;

%walking zones and reflex triggers
zone1=0;
zone2=1;
forwardzmpsteptrigger=0.05;
referencecycle=0;
ZMPx=0;
ZMPy=0;

% Constraint Matrix for optimization (A*fE<b)
Alsqlin=[1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0; ...
-1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; ...
0 1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; ...
0 -1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; ...
0 0 0 1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 -1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 -1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 -1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 -1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 -1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 -1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 -1 0 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -mu*sqrt(2)/2 0 0 0 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 -mu*sqrt(2)/2 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 -mu*sqrt(2)/2 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -mu*sqrt(2)/2 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -mu*sqrt(2)/2 0 0 0 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 -mu*sqrt(2)/2 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 -mu*sqrt(2)/2 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -mu*sqrt(2)/2 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -mu*sqrt(2)/2 0 0 0;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 -mu*sqrt(2)/2;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 -mu*sqrt(2)/2;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -mu*sqrt(2)/2;...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -mu*sqrt(2)/2];
blsqlin=zeros(32,1);

% Lower and upper bounds for fE (fE(z)>0)
lblsqlin=[-100 -100 0 -100 -100 0 -100 -100 0 -100 -100 0 -100 -100 0
-100 -100 0 -100 -100 0 -100 -100 0]';
ublsqlin=[];
```

```
% No equality type constraint
Aeqlsqlin=[];
beqlsqlin=[];


% Parameters for Inverse Dynamics in Body Posture Control
K0=diag([36 36 100 400 400 1600]);
K1=diag([12 12 20 40 40 80]);
```

Appendix B

Matlab M-file for control mode selection (double-support, right-support, left support).

```
function result=controller(inputs)

% controller.m
% accepts body position and orientation references, actual ones, feet
% corner and center positions, bias vector, contact forces, artificial
bias
% vector, inertia matrix, transfers to the appropriate control phase
% (double-support, right-support, left-support) accordingly.
global n
global uEvector avector jfvector
global b H xvector vvector
global simulationcycle
global steptime

global zone1
global zone2
global referencecycle

global ZMPx ZMPy
global uvector

global K
global avector

global Alsqlin ublsqlin lblsqlin blsqlin beqlsqlin Aeqlsqlin
% Parameters for lsq constrained optimization
global lb_lsqnonneg  %lower bound on tangential contact force

global TR TL  % Right foot and left foot center transformation
matrices.
global pseudo_b % Feed-forward term for swing leg
global touch_on_off % Touch detection
global w_p_c % Foot corners positions
global K0 K1

global pbref wAbref vbref wbref vbdotref wbdotref
global result

global fE

global trigger_1 trigger_2
global ds_switch rs_switch ls_switch rs_time ls_time ds_time

global options

global g_trig
global pbref

%INPUT VARIABLES
velocity_vector=inputs(1:6+2*n);
position_vector=inputs(7+2*n:18+4*n);
pb=position_vector(1:3);
wAblist=position_vector(4:12);
q=position_vector(13:12+2*n);
vb=velocity_vector(1:3);
wb=velocity_vector(4:6);
qdot=velocity_vector(7:6+2*n);
wAb=[wAblist(1:3) wAblist(4:6) wAblist(7:9)];


pbref=[0.06*simulationcycle*steptime; -
0.075*sin(pi/2*simulationcycle*steptime) ; 0.27 + 0.22 + 0.087 - 0.05];
wAbref=eye(3);
vbref=[0.06 -0.075*pi*cos(pi/2*simulationcycle*steptime)/2  0]';
wbref=[0 0 0]';
vbdotref=[0 0.075*pi^2*sin(pi/2*simulationcycle*steptime)/4  0]';
wbdotref=[0 0 0]';

if trigger_1*(-pb(2) + max(w_p_c(2,1:4)) -1) + pb(2) >
max(w_p_c(2,1:4)) -0.01 & trigger_2*(-pb(2) + min(w_p_c(2,5:8)) +1) +
pb(2) < min(w_p_c(2,5:8)) + 0.01
    % Double-support phase. Body coordinate center is not close enough
to the supporting
    % polygone.

        if ds_switch==1;
            % First time in double support, display time.
            ds_time=simulationcycle*steptime
            ds_switch=0;
            rs_switch=1;
            ls_switch=1;
        end

        ds_force_control;

%     end

elseif trigger_1*(-pb(2) + max(w_p_c(2,1:4)) -1) + pb(2) <
max(w_p_c(2,1:4)) - 0.01
    % Get into the single support phase. Right foot stance, left foot
swing

    if pb(2) > max(w_p_c(2,1:4))
        if sqrt(fE(15)^2 + fE(18)^2 + fE(21)^2 + fE(24)^2) > 10
            % Get into the double support phase. Norm of contact
forces is
            % greater than a threshold (10 N).
            trigger_1=0;
            ds_force_control;
        end

    else

        if rs_switch==1;
            % First time in right support, display time.
            rs_time=simulationcycle*steptime
            rs_switch=0;
            ds_switch=1;
            ls_switch=1;
        end

        trigger_1=1;
        rs_force_control;

    end

else
    % Get into the single support phase. Left foot stance, right foot
swing
    if pb(2) < min(w_p_c(2,5:8))
        if sqrt(fE(3)^2 + fE(6)^2 + fE(9)^2 + fE(12)^2) > 10
            % Get into the double support phase. Norm of contact
forces is
            % greater than a threshold (10 N).
            trigger_2=0;
            ds_force_control;
        end

    else

        if ls_switch==1;
            % First time in left support, display time.
            ls_time=simulationcycle*steptime
            ls_switch=0;
            ds_switch=1;
            rs_switch=1;
        end

        trigger_2=1;
        ls_force_control;

    end

end

uvector=result;
```

Appendix C

Matlab code for double support force control.

```
% ds_force_control.m
% Control scheme for double support phase
global n
global uEvector avector jfvector
global b H xvector vvector
global simulationcycle
global steptime

global ZMPx ZMPy
global uvector

global K
global avector

global  Alsqlin  ublsqlin  lblsqlin  blsqlin  beqlsqlin  Aeqlsqlin    %
Parameters for lsq constrained optimization
global lb_lsqnonneg  %lower bound on tangential contact force
global  TR  TL   %  Right  foot  and  left  foot  center  transformation
matrices.
global touch_on_off % Touch detection
global w_p_c % Foot corners positions
global K0 K1
global pbref wAbref vbref wbref vbdotref wbdotref
global result

global options

accn_ref=[vbdotref ; wbdotref];

pberror=pbref-pb;
wAberror=0.5*(cross(wAb(1:3,1),  wAbref(1:3,1))  +  cross(wAb(1:3,2),
wAbref(1:3,2)) + cross(wAb(1:3,3), wAbref(1:3,3)));

vberror=vbref-vb;
wberror=wbref-wb;

error_position_body=[pberror ; wAberror];
error_velocity_body=[vberror ; wberror];

v_inversedynamics=K0*error_position_body  +  K1*error_velocity_body  +
accn_ref;

uE12ref=H(1:6,1:6)*v_inversedynamics + b(1:6);

fEref=lsqlin(K(1:6,:),uE12ref,Alsqlin,blsqlin,Aeqlsqlin,beqlsqlin,
lblsqlin,ublsqlin,[],options);
TauRef=K(7:18,:)*fEref;

result=[zeros(6,1) ; -TauRef + b(7:18)];
```

Appendix D

Matlab code for right support force control.

```
% rs_force_control.m
% Control scheme for right support phase
% Control scheme for left support phase is not included in the thesis,
since it is similar to rs_force_control;
global n
global uEvector avector jfvector
global b H xvector vvector
global simulationcycle stepdivider
global steptime

global ZMPx ZMPy
global uvector

global K K_fc
global avector

global Alsqlin ublsqlin lblsqlin blsqlin beqlsqlin Aeqlsqlin
% Parameters for lsq constrained optimization

global lb_lsqnonneg  %lower bound on tangential contact force
global TR TL   % Right foot and left foot center transformation
matrices.

global TR_fixed TL_fixed
global sfl_err_old

global pseudo_b % Feed-forward term for swing leg
global touch_on_off % Touch detection
global w_p_c % Foot corners positions
global K0 K1
global pbref wAbref vbref wbref vbdotref wbdotref
global result

global options

global Kp_fc Kd_fc

global pbref

accn_ref=[vbdotref ; wbdotref];

pberror=pbref-pb;
wAberror=0.5*(cross(wAb(1:3,1), wAbref(1:3,1)) + cross(wAb(1:3,2),
wAbref(1:3,2)) + cross(wAb(1:3,3), wAbref(1:3,3)));

vberror=vbref-vb;
wberror=wbref-wb;

error_position_body=[pberror ; wAberror];
error_velocity_body=[vberror ; wberror];

v_inversedynamics=K0*error_position_body + K1*error_velocity_body +
accn_ref;

uE12ref=H(1:6,1:6)*v_inversedynamics + b(1:6) +
H(1:6,7:18)*zeros(12,1);

fEref =
lsqlin(K(1:6,1:12),uE12ref,Alsqlin,blsqlin,Aeqlsqlin,beqlsqlin,lblsqli
n,ublsqlin,[],options);

% !!! For the swinging foot (Left Foot), Position Control Scheme
TL_fixed(1,4)=pbref(1) - 0.05 + 0.03;  % d=0.03;

pfl_err=TL_fixed(1:3,4) - TL(1:3,4);
thetafl_err=0.5*(cross(TL(1:3,1), TL_fixed(1:3,1)) + cross(TL(1:3,2),
TL_fixed(1:3,2)) + cross(TL(1:3,3), TL_fixed(1:3,3)));

sfl_err=[pfl_err; thetafl_err];
vsfl_err= (sfl_err - sfl_err_old)/steptime;

sfl_err_old=sfl_err;

S=diag([0.4 1 0 0.6 0.3 1]);   % Selection matrix for position
controlled and gravity compensated directions

sfl_pc_err=S*sfl_err;
vsfl_pc_err=S*vsfl_err;

sfl_j_err=inv(K_fc(13:18,7:12)')*sfl_pc_err;   % Approximate joint
tracking error
```

```
vsfl_j_err=inv(K_fc(13:18,7:12)')*vsfl_pc_err;  % Joint velocity error

tau_pc=Kp_fc*sfl_j_err + Kd_fc*vsfl_j_err;

TauRefR=K(7:12,1:12)*fErefR;
result=[zeros(6,1) ; -TauRefR + b(7:12); pseudo_b(13:18) + tau_pc];
```

REFERENCES

[1] Sugihara, T., Nakamura, Y., Inoue, H., "Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control", *Proc. IEEE Int. Conf. on R&A, ICRA 2002,* Washington, DC, May 2002.

[2] Knight, R., and Nehmzow, U., "Walking Robots, A Survey and a Research Proposal", *Technical Report.*

[3] M. Vukobratovic, B. Borovac, D. Surla, and Stokic, *Biped Locomotion: Dynamics, Stability and Application.* Springer-Verlag, 1990.

[4] Yonemura, A., Y. Nakajima and A. Kawamura, "Experimental Approach for the Fast Walking by the Biped Robot," *Proc. IASTED Int. Conference Robotics and Applications*, Honolulu, Hawaii, USA, 2000.

[5] Y. Fujimoto, A. Kawamura, "Robust Biped Walking with Active Interaction Control between Foot and Ground", *Proc. IEEE Int. Conf. on R&A*, pp. 2030-2035, 1998.

[6] Fukuda T., Youichirou K., Takemasa A. "Stabilization Control of Biped Locomotion Robot based Learning with GAs having Self-adaptive Mutation and Recurrent Neural Networks", *Proc. IEEE Int. Conf. Robotics and Automation ICRA '97*; Albuquerque, New Mexico, April 1997.

[7] Bebek, O., Erbatur, K., "A Gait Adaptation Scheme for Biped Walking Robots," *8th IEEE International Workshop on Advanced Motion Control*, AMC '04, March 2004, pp. 409-141.

[8] Chevallereau, C., A. Formal'sky and B. Perrin, "Low Energy Cost Reference Trajectories for a Biped Robot," *Proc. IEEE Int. Conf. Robotics and Automation ICRA '98*; Leuven, Belgium, May 1998.

[9] Genk, J., Schmidt, G., "Synthesis of a Walking Primitive Database for a Humanoid Robot using Optimal Control Techniques", *Proc. IEEE-RAS Int. Conf. on Humanoid Robots HUMANOIDS 2001,* pp. 319-326, Tokyo, Japan, November 2001.

[10] Pratt G., "Low Impedance Walking Robots", *Journal of Integrative and Comparative Biology, Vol. 42, No. 1*, 2002, pp. 174-181.

[11] Bruneau, O., F. Ben Ouezdou, “Compliant Contact of Walking Robot Feet,” *Proc. of the third ECPD International Conference on Advanced Robotics*; Bremen, Germany, September 26-28, 1997.

[12] Linde, R. Q., “Active Leg Compliance for Passive Walking,” *Proc. IEEE Int. Conf. Robotics and Automation, ICRA ’98*; Leuven, Belgium, May 1998.

[13] Silva, F. M., J.A.Tenreiro Machado “Goal-Oriented Biped Walking Based on Force Interaction Control,” *Proc. IEEE Int. Conf. R&A., ICRA ’01*; Seoul, Korea, May 21-26, 2001.

[14] S. Kawaji, K. Ogasawara, J. Iimori and S. Yamada, “Compliance Control for Biped Locomotion Robot”, *Proc. IEEE Int. Conf. on R&A*, pp. 3801-3806, 1994.

[15] Hashimoto, S., *et al*, *“Humanoid Robots in Waseda University, Hadaly-2 and Wabian”,* Humanoid Robotics Institute, Waseda University.

[16] A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato, “The Realization of Dynamic Walking by the Biped Walking Robot WL-10RD,” in *ICAR’85*, pp. 459–466, 1985.

[17] Marc Raibert, *Legged Robots that Balance*, MIT Press, Cambridge, MA, 1986.

[18] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9:62-82,1990.

[19] The HONDA humanoid robot, http://world.honda.com/ASIMO/.

[20] Pratt, J., Chee-Meng Chew, Torres, A., Dilworth, P., Pratt, G., “Virtual Model Control: An Intuitive Approach for Bipedal Locomotion”, *The International Journal of Robotics Research, Vol. 20, No. 2,* February 2001, pp. 129-143.

[21] Nagasaka, K., Inaba, M., Inoue, H., “Walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method”, *Proc. IEEE Int. Conf. R&A, ICRA ’98*; Leuven, Belgium, May 1998.

[22] Pandy, M. G., and Anderson, F. C., “Dynamic Simulation of Human Movement using Large-Scale Models of the Body”, *Proc. IEEE Int. Conf. R&A, ICRA ’00*; San Fransico, California, 2000, pp. 676-681.

[23] Channon, P. H., Hopkins, S.H., Pham, D.T., “Derivation of Optimal Walking Motions for a Bipedal Walking Robot,” *Robotica*, Vol. 10, pp. 165-172, 1992.

[24] Dasgupta, A., Nakamura, Y., “Making Feasible Walking Motion of Humanoid Robots From Human Motion Capture Data,” *Proc. IEEE Int. Conf. R&A, ICRA ’99*; Detroit, Michigan, May 1999.

[25] Miura, H., and Shimoyama, I., “Dynamic Walk of a Biped”, *International Journal of Robotics Research, Vol. 3, No. 2*, pp. 60-72, 1984.

[26] Hirai, K., Hirose, M., Haikawa, Y. and Takenaka, T., “The Development of Honda Humanoid Robot” ”, *Proc. IEEE Int. Conf. R&A, ICRA ’98*; Leuven, Belgium, May 1998, pp. 1321-1326.

[27] Yamaguchi, J., Soga, E., Inoue, S. and Takanishi, A., “Development of a Bipedal Humanoid Robot – Control of Whole Body Cooperative Dynamic Biped Walking – ,“ *Proc. IEEE Int. Conf. R&A, ICRA ’99*; Detroit, Michigan, pp. 368-374, May 1999.

[28] Vukobratovic, M., *Legged Locomotion Robots and Anthropomorphic Mechanisms*, based on the Final Report of the project: Active Exoskeleton for Paraplegics, Institute “M. Pupin,” vol. 25, Belgrade, Yugoslavia, 1974.

[29] Juricic D., and Vukobratovic, M., “Mathematical modeling of a bipedal walking system,” *ASME Publ.* 72-WA/BHF-13, 1972.

[30] Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, Z., Valavanis, K. and Vukobratovic M., ” *IEEE Robotics and Automation Magazine*, March 2001, pp. 66-73.

[31] Miyazaki, F. and Arimoto, S., “A Control Theoretic Study on Dynamical Biped Locomotion” *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 102:233–239, 1980.

[32] Yamaguchi, J., Takanishi, A., and Kato, I., “Development of a Biped Walking Robot Compensating for Three-axis Moment by Trunk Motion,” *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS ‘93*, pp. 561–566, July 1993.

[33] Huang, Q., Kajita, S., *et. al*, “A High Stability, Smooth Walking Pattern for a Biped Robot,” *Proc. IEEE Int. Conf. R&A, ICRA ’99*; Detroit, Michigan, pp. 65-71, May 1999.

[34] Park, J. H. and Chung, H., “ZMP Compensation by On-Line Trajectory Generation for Biped Robots,” *Proc. Conf. on Systems, Man and Cybernetics,* 1999.

[35] Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K. and Hirukawa H., “The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation,” *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS ‘01*, pp. 239–246, November 2001.

[36] Luh, J.Y.S., M. W. Walker and R.P.C. Paul, “On-line Computational Scheme for Mechanical Manipulators,” *Jour. Dyn. Syst. Meas. Cont.*, Vol. 102, pp. 69, June 1980.

[37] Erbatur, K., A. Kawamura, “A New Penalty Based Contact Modeling and Dynamics Simulation Method as Applied to Biped Walking Robots," *Proc. 2003 FIRA World Congress*, October 1-3, 2003 Vienna, Austria.

[38] Fujimoto, Y. and A. Kawamura, “Simulation of an Autonomous Biped Walking Robot Including Environmental Force Interaction,” *IEEE Robotics and Automation Magazine*, June 1998, pp. 33-42.

[39] R. L. Rardin, “*Optimization in Operations Research,*” Prentice Hall , 2000.

[40] R. Featherstone, *Robot Dynamics Algorithms.* Kluwer Academic Publishers, 2003.

REFERENCES

[1] Sugihara, T., Nakamura, Y., Inoue, H., "Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control", *Proc. IEEE Int. Conf. on R&A, ICRA 2002,* Washington, DC, May 2002.

[2] Knight, R., and Nehmzow, U., "Walking Robots, A Survey and a Research Proposal", *Technical Report.*

[3] M. Vukobratovic, B. Borovac, D. Surla, and Stokic, *Biped Locomotion: Dynamics, Stability and Application.* Springer-Verlag, 1990.

[4] Yonemura, A., Y. Nakajima and A. Kawamura, "Experimental Approach for the Fast Walking by the Biped Robot," *Proc. IASTED Int. Conference Robotics and Applications*, Honolulu, Hawaii, USA, 2000.

[5] Y. Fujimoto, A. Kawamura, "Robust Biped Walking with Active Interaction Control between Foot and Ground", *Proc. IEEE Int. Conf. on R&A*, pp. 2030-2035, 1998.

[6] Fukuda T., Youichirou K., Takemasa A. "Stabilization Control of Biped Locomotion Robot based Learning with GAs having Self-adaptive Mutation and Recurrent Neural Networks", *Proc. IEEE Int. Conf. Robotics and Automation ICRA '97*; Albuquerque, New Mexico, April 1997.

[7] Bebek, O., Erbatur, K., "A Gait Adaptation Scheme for Biped Walking Robots," *8th IEEE International Workshop on Advanced Motion Control*, AMC '04, March 2004, pp. 409-141.

[8] Chevallereau, C., A. Formal'sky and B. Perrin, "Low Energy Cost Reference Trajectories for a Biped Robot," *Proc. IEEE Int. Conf. Robotics and Automation ICRA '98*; Leuven, Belgium, May 1998.

[9] Genk, J., Schmidt, G., "Synthesis of a Walking Primitive Database for a Humanoid Robot using Optimal Control Techniques", *Proc. IEEE-RAS Int. Conf. on Humanoid Robots HUMANOIDS 2001,* pp. 319-326, Tokyo, Japan, November 2001.

[10] Pratt G., "Low Impedance Walking Robots", *Journal of Integrative and Comparative Biology, Vol. 42, No. 1*, 2002, pp. 174-181.

[11] Bruneau, O., F. Ben Ouezdou, "Compliant Contact of Walking Robot Feet," *Proc. of the third ECPD International Conference on Advanced Robotics*; Bremen, Germany, September 26-28, 1997.

[12] Linde, R. Q., "Active Leg Compliance for Passive Walking," *Proc. IEEE Int. Conf. Robotics and Automation, ICRA '98*; Leuven, Belgium, May 1998.

[13] Silva, F. M., J.A.Tenreiro Machado "Goal-Oriented Biped Walking Based on Force Interaction Control," *Proc. IEEE Int. Conf. R&A., ICRA '01*; Seoul, Korea, May 21-26, 2001.

[14] S. Kawaji, K. Ogasawara, J. Iimori and S. Yamada, "Compliance Control for Biped Locomotion Robot", *Proc. IEEE Int. Conf. on R&A*, pp. 3801-3806, 1994.

[15] Hashimoto, S., *et al*, *"Humanoid Robots in Waseda University, Hadaly-2 and Wabian",* Humanoid Robotics Institute, Waseda University.

[16] A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato, "The Realization of Dynamic Walking by the Biped Walking Robot WL-10RD," in *ICAR'85*, pp. 459–466, 1985.

[17] Marc Raibert, *Legged Robots that Balance*, MIT Press, Cambridge, MA, 1986.

[18] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9:62-82,1990.

[19] The HONDA humanoid robot, http://world.honda.com/ASIMO/.

[20] Pratt, J., Chee-Meng Chew, Torres, A., Dilworth, P., Pratt, G., "Virtual Model Control: An Intuitive Approach for Bipedal Locomotion", *The International Journal of Robotics Research, Vol. 20, No. 2,* February 2001, pp. 129-143.

[21] Nagasaka, K., Inaba, M., Inoue, H., "Walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method", *Proc. IEEE Int. Conf. R&A, ICRA '98*; Leuven, Belgium, May 1998.

[22] Pandy, M. G., and Anderson, F. C., "Dynamic Simulation of Human Movement using Large-Scale Models of the Body", *Proc. IEEE Int. Conf. R&A, ICRA '00*; San Fransico, California, 2000, pp. 676-681.

[23] Channon, P. H., Hopkins, S.H., Pham, D.T., "Derivation of Optimal Walking Motions for a Bipedal Walking Robot," *Robotica*, Vol. 10, pp. 165-172, 1992.

[24] Dasgupta, A., Nakamura, Y., "Making Feasible Walking Motion of Humanoid Robots From Human Motion Capture Data," *Proc. IEEE Int. Conf. R&A, ICRA '99*; Detroit, Michigan, May 1999.

[25] Miura, H., and Shimoyama, I., "Dynamic Walk of a Biped", *International Journal of Robotics Research, Vol. 3, No. 2*, pp. 60-72, 1984.

[26] Hirai, K., Hirose, M., Haikawa, Y. and Takenaka, T., “The Development of Honda Humanoid Robot” ”, *Proc. IEEE Int. Conf. R&A, ICRA ’98*; Leuven, Belgium, May 1998, pp. 1321-1326.

[27] Yamaguchi, J., Soga, E., Inoue, S. and Takanishi, A., “Development of a Bipedal Humanoid Robot – Control of Whole Body Cooperative Dynamic Biped Walking – ,“ *Proc. IEEE Int. Conf. R&A, ICRA ’99*; Detroit, Michigan, pp. 368-374, May 1999.

[28] Vukobratovic, M., *Legged Locomotion Robots and Anthropomorphic Mechanisms*, based on the Final Report of the project: Active Exoskeleton for Paraplegics, Institute “M. Pupin,” vol. 25, Belgrade, Yugoslavia, 1974.

[29] Juricic D., and Vukobratovic, M., “Mathematical modeling of a bipedal walking system,” *ASME Publ.* 72-WA/BHF-13, 1972.

[30] Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, Z., Valavanis, K. and Vukobratovic M., ” *IEEE Robotics and Automation Magazine*, March 2001, pp. 66-73.

[31] Miyazaki, F. and Arimoto, S., “A Control Theoretic Study on Dynamical Biped Locomotion” *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 102:233–239, 1980.

[32] Yamaguchi, J., Takanishi, A., and Kato, I., “Development of a Biped Walking Robot Compensating for Three-axis Moment by Trunk Motion,” *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS ‘93*, pp. 561–566, July 1993.

[33] Huang, Q., Kajita, S., *et. al*, “A High Stability, Smooth Walking Pattern for a Biped Robot,” *Proc. IEEE Int. Conf. R&A, ICRA ’99*; Detroit, Michigan, pp. 65-71, May 1999.

[34] Park, J. H. and Chung, H., “ZMP Compensation by On-Line Trajectory Generation for Biped Robots,” *Proc. Conf. on Systems, Man and Cybernetics,* 1999.

[35] Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K. and Hirukawa H., “The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation,” *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS ‘01*, pp. 239–246, November 2001.

[36] Luh, J.Y.S., M. W. Walker and R.P.C. Paul, “On-line Computational Scheme for Mechanical Manipulators,” *Jour. Dyn. Syst. Meas. Cont.*, Vol. 102, pp. 69, June 1980.

[37] Erbatur, K., A. Kawamura, “A New Penalty Based Contact Modeling and Dynamics Simulation Method as Applied to Biped Walking Robots," *Proc. 2003 FIRA World Congress*, October 1-3, 2003 Vienna, Austria.

[38] Fujimoto, Y. and A. Kawamura, “Simulation of an Autonomous Biped Walking Robot Including Environmental Force Interaction,” *IEEE Robotics and Automation Magazine*, June 1998, pp. 33-42.

[39] R. L. Rardin, “*Optimization in Operations Research,*” Prentice Hall , 2000.

[40] R. Featherstone, *Robot Dynamics Algorithms.* Kluwer Academic Publishers, 2003.