

Modelling, Simulation, and Control for a Bipedal Walking Robot

by

Henry Fung

A Thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario, Canada
January 17, 2011

Copyright ©
2011 - Henry Fung

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the Thesis

**Modelling, Simulation, and Control for a Bipedal Walking
Robot**

Submitted by **Henry Fung**
in partial fulfilment of the requirements for the degree of
Master of Applied Science

M. Ahmadi, Supervisor

A. Chan

R. Doraiswami

L. Liu

M. Yaras, Department Chair
Carleton University

2011

Abstract

The goal of this thesis is to design a controller that enables the *Advanced Biomechatronics and Locomotion Laboratory- Biped I (ABL-BI)*, a biped with 13 active degrees of freedom, to walk continuously in a stable manner in the presence of small external disturbances. With this goal in mind, two walking controllers are designed. The first controller (the ZMP-based controller) is a modification of an existing real-time controller that is based on the Linear Inverted Pendulum Model (LIPM) and the Zero Moment Point (ZMP). Two major changes are made to this controller: 1) the addition of a Center of Gravity (COG) position control loop, and 2) the application of a task prioritization method in the inverse differential kinematics algorithm of the biped. These changes are made to simplify reference motion planning and postural control. The second controller (the ZRAM-based controller) is a new bio-inspired control method that regulates the centroidal moment of the biped by minimizing the distance between two ground reference points: the reference Centroidal Moment Pivot (CMP) and the Center of Pressure (COP). The main advantage of this method is its ability to modify the reference feedforward motion in real-time to attenuate external disturbances. It is also a relatively simple control strategy that focuses on the fundamental relationships and state variables that are key to the dynamic stability and high level motion of the biped. In addition, a new approach in kinematic modelling and solution of the equations has been proposed and implemented. The full kinematics of the biped is treated as separate open kinematic chains in order to

delegate task space objectives to specific segments of the biped and simplify postural control.

A 3-D simulated biped that is based on the kinematic and mass specification of ABL-BI is developed in a simulation environment called *Webots*. Walking tests are performed on the simulated biped to validate the two proposed walking controllers and the results show that both the ZMP-based controller and the ZRAM-based controller can generate stable walking gaits. In addition, a series of perturbation tests are performed on the ZRAM-based controller to evaluate its robustness to external disturbances. It is shown that the level of disturbance that is attenuated by the ZRAM-based controller in the perturbation tests is comparable to a similar walking controller in the literature.

To my Lord and Savior Jesus Christ.

Acknowledgments

I would first like to thank my mentor, friend and supervisor Prof. Mojtaba Ahmadi for his guidance and patience, and for giving me an opportunity to pursue my graduate studies in a truly interesting research topic. I am proud to be a part of the effort that is working towards making a new form of intelligence on earth. I would also like to thank Alexis Guigue and Majid Moghaddam for their help in the early parts of this project. I am in debt to my friends in the ABL-Lab: Ali, for his friendship and insightful advice on work and life, Richard, my good friend and main collaborator of this project, Adam, Kyle C., Kyle K., Owain, and Mahmoud for making the long hours in the lab enjoyable.

Besides the people in the ABL-Lab, there are also many others who have helped me in some way or another for the past few years. I would like to thank Dave Miyata and Tim, who were always there to help whenever I need it, and for their unique sense of humor, Shashank, to whom I can always go to for advice, Dave Miskowicz, Jess, Jonathan, Hanjie, Doris and Karen, for their friendship. I also want to express my gratitude to Kin, Sarah, and Monica for teaching me the most important and relevant things in the world.

Lastly, I want to thank my parents, my brother, and my family for their loving support and for making all this possible for me.

Table of Contents

Abstract	iii
Acknowledgments	vi
Table of Contents	vii
List of Tables	xii
List of Figures	xiv
Nomenclature	xxvi
1 Introduction	1
1.1 Motivation and Challenges	3
1.1.1 Why Study Bipedal Locomotion?	3
1.1.2 Challenges	4
1.2 Common Definitions and Stability Criteria in Bipedal Locomotion . .	7
1.2.1 The Center of Gravity (COG)	7
1.2.2 The Center of Pressure (COP)	7
1.2.3 The Zero Moment Point (ZMP)	9
1.2.4 The Centrioidal Moment Pivot (CMP)	10
1.2.5 Static vs. Dynamic Stability	11
1.3 Research on Bipedal Walking	12

1.3.1	Offline Trajectory Generation	12
1.3.2	Model-based Online Trajectory Generation	14
1.3.3	Virtual Model Control (VMC)	15
1.3.4	Central Pattern Generators (CPG)	15
1.3.5	Passive Dynamic Walkers	16
1.4	Objectives	17
1.5	Contributions	19
1.6	Outline	20
2	ABL-BI: The Experimental Platform and the Simulation Model	23
2.1	The ABL-BI Platform	24
2.2	Webots and the ABL-BI Simulation Model	26
2.2.1	Overview of Webots	26
2.2.2	The Features of the Simulation Model in Webots	31
2.2.3	Sensory Information in Webots	40
2.2.4	The Control Levels in Webots	42
2.2.5	Summary	47
3	The Essential Elements of the Robot Controllers	48
3.1	The High Level Tasks of Bipedal Walking	49
3.2	The Locomotion States	52
3.2.1	Locomotion States in a Walking Cycle	52
3.2.2	Intermediate Objectives and Workspace Constraints	55
3.2.3	State Transitions	58
3.3	Biped Kinematic Arrangements	61
3.3.1	Kinematic Task Formulation	63
3.3.2	Definition of the Kinematic Chains	75
3.4	The Forward Kinematics of ABL-BI	78

3.4.1	Overview	78
3.4.2	The Output Variables of the Simulated Biped	78
3.4.3	The Output Variables in the Actual Biped	82
3.4.4	The Local Inertial Frame	104
3.5	The Inverse Kinematics of ABL-BI	107
3.5.1	Overview	107
3.5.2	The Jacobian Pseudoinverse	108
3.5.3	The Kinematic Control of ABL-BI	112
3.5.4	Initial Simulation Test Results	124
3.6	The Simplifying Models in Bipedal Walking	138
3.6.1	Overview	138
3.6.2	Dynamic Relationships in Bipedal Walking	141
3.6.3	The 3-D Linear Inverted Pendulum Model (3-D LIPM) . . .	143
3.6.4	The ZRAM Model	150
3.7	The Control Architecture	154
4	A Walking Controller Based on the LIPM and the ZMP Stability Criterion	158
4.1	The Control Algorithm	159
4.2	Initial Walking Tests in Webots	164
4.2.1	A Summary of the User-specified Parameters for the ZMP-based Controller	164
4.2.2	Results and Discussions	166
4.3	Limitations on the Walking Speed of ABL-BI	185
4.3.1	Limiting Factors in Double Stance	186
4.3.2	Limiting Factors in Single Stance	197
4.4	Summary	208

5 A Walking Controller Based on CMP Planning and COG Position Control	211
5.1 Motivations and Theoretical Background	212
5.1.1 The ZRAM Stability Criterion	212
5.1.2 Related Control Methods	213
5.1.3 Angular Momentum in Bipedal Locomotion	215
5.2 The Control Algorithm	217
5.2.1 The Control Architecture of the ZRAM-based Controller	217
5.2.2 Dynamically Stable COG Generation based on ZRAM	218
5.3 Simulation Tests in the 2-D Simulation	221
5.3.1 Planar Model	221
5.3.2 Simulation Results and Discussions	225
5.4 Implementation on the 3-D Model	231
5.4.1 Walking Tests	232
5.4.2 Perturbation Tests	252
5.5 Summary	264
5.5.1 Comparing the ZRAM-based Controller to a Similar Control Strategy	267
6 Conclusions	270
6.1 Summary	270
6.2 Conclusions	271
6.3 Suggestions for Future Work	276
List of References	279
Appendix A Parameters for the ABL-BI Structural Model	287
Appendix B The Transformation Matrices of the Biped Model	289

Appendix C Additional Test Results for the Forward Kinematics of the Biped	294
Appendix D The Derivation of the COG Jacobian	306
Appendix E The Differential Kinematics of the Kinematic Chains	315

List of Tables

2.1	ABL-BI specifications.	25
2.2	Actuators specifications.	25
2.3	The definable physical properties of a Servo node that include physical properties of the actual motor and link.	36
2.4	The name and number of the frames in the Simulation Model.	37
2.5	The name, number, and mass of the links in the Simulation Model. .	38
2.6	Summary of accessible and inaccessible states from <i>Webots</i>	42
3.1	State transition requirements.	61
3.2	The definition of the kinematic chains in each locomotion state. . .	76
3.3	The high level task assignment for the kinematic chains.	76
3.4	The damping factor and priority levels of the high level tasks in the kinematic chains.	123
4.1	The control gains for the ZMP-based controller in the walking test. .	165
4.2	The key reference trajectory planning parameters for the ZMP-based controller.	166
4.3	The mass distribution of ABL-BI, various biped models, a humanoid robot NUSBIP II, and a 10-year old child.	187
5.1	The kinematic and mass properties of the planar biped.	225
5.2	The control gains for the ZRAM-based controller in the walking test.	234

5.3	The key reference trajectory planning parameters for the ZRAM-based controller.	235
5.4	The control gains for the ZRAM-based controller in the perturbation tests.	254
A.1	Relative position between each successive frames in the Structural Model.	287
A.2	Link COG position with respect to nearest frame.	288

List of Figures

1.1	The foot contact forces (F_1 to F_4) can be represented by a resultant GRF acting at the COP.	8
1.2	The overall COP and GRF of both feet and the local COP and GRF of the left foot (COP_L and F_L) and the right foot (COP_R and F_R). . .	8
1.3	The Zero Moment Point.	10
1.4	The Centroidal Moment Pivot.	11
2.1	The ABL-BI Platform.	24
2.2	The world editor and the hierachal modelling structure in Webots . .	27
2.3	ODE objects: Frames, Shapes, and Joints	29
2.4	A comparison between the ABL-BI platform and the ABL-BI model .	31
2.5	A joint connecting two rigid links	32
2.6	The Simulation Model of ABL-BI.	34
2.7	The Servo Node in Webots	35
2.8	A schematic diagram of the control architecture of the ABL-BI simu- lation.	46
3.1	The importance of maintaining an upright pelvis orientation in bipedal walking	51
3.2	The Locomotion states in the walking cycle	54
3.3	The intermediate objectives of the locomotion states in the walking cycle.	59

3.4	The locomotion state and state transitions of ABL-BI.	60
3.5	The Segments and kinematic chains of the biped	62
3.6	Two ways to model the the biped in SSL.	65
3.7	Reference swing foot position tracking using the inverse kinematics of a single leg chain.	66
3.8	Swing leg motion produced by the joints in the swing leg chain. . . .	71
3.9	Comparison between the actual and reference swing foot X position. .	71
3.10	Comparison between the actual and reference swing foot Y position. .	72
3.11	Comparison between the actual and reference swing foot Z position. .	72
3.12	Pitch joint angle position comparisons between the stance and swing leg (HPL vs. HPR).	73
3.13	Pitch joint angle position comparisons between the stance and swing leg (KPL vs. KPR).	73
3.14	Pitch joint angle position comparisons between the stance and swing leg (FPL vs. FPR).	74
3.15	The kinematic chain definitions for each locomotion state.	77
3.16	The local inertial frame	79
3.17	The two step cycle reference motion for the swing foot position test in Webots.	84
3.18	The validation test results for the swing foot X position with respect to the stance foot for SSL and SSR.	84
3.19	The validation test results for the swing foot Y position with respect to the stance foot for SSL and SSR.	85
3.20	The validation test results for the swing foot Z position with respect to the stance foot for SSL and SSR.	85
3.21	The COG X position forward kinematics test	88
3.22	The COG Y position forward kinematics test	88

3.23	The COG Z position forward kinematics test	89
3.24	The COG X velocity test	89
3.25	The COG Y velocity test	90
3.26	The COG Z velocity test	90
3.27	The reference robot motions for the pelvis orientation test in SSL. . .	94
3.28	The validation test results for the pelvis orientation (about the X axis) in SSL.	95
3.29	The validation test results for the pelvis orientation (about the Y axis) in SSL.	95
3.30	The validation test results for the pelvis orientation (about the Z axis) in SSL.	96
3.31	The reference robot motions for the pelvis orientation test in SSR. . .	96
3.32	The validation test results for the pelvis orientation (about the X axis) in SSR.	97
3.33	The validation test results for the pelvis orientation (about the Y axis) in SSR.	97
3.34	The validation test results for the pelvis orientation (about the Z axis) in SSR.	98
3.35	The reference robot motions for the swing foot orientation test in SSL.	98
3.36	The swing foot orientation about the X axis in SSL.	99
3.37	The swing foot orientation about the Y axis in SSL.	99
3.38	The swing foot orientation about the Z axis in SSL.	100
3.39	The reference robot motions for the swing foot orientation test in SSR.	100
3.40	The swing foot orientation about the X axis in SSR.	101
3.41	The swing foot orientation about the Y axis in SSR.	101
3.42	The swing foot orientation about the Z axis in SSR.	102
3.43	The local inertial frame in SSR	105

3.44 The local inertial frame in SSR	106
3.45 The control architecture of the kinematic postural controller.	114
3.46 Initial inverse kinematics test: oscillatory COG motion in the lateral plane.	125
3.47 The comparison between the actual and reference COG velocity X. .	126
3.48 The actual COG velocity Y.	126
3.49 The comparison between the actual and reference COG velocity Z. .	127
3.50 Initial inverse kinematics test: oscillatory COG motion in the sagittal plane.	128
3.51 The comparison between the actual and reference COG velocity X. .	129
3.52 The actual COG velocity Y.	129
3.53 The comparison between the actual and reference COG velocity Z. .	130
3.54 Initial inverse kinematics test: the step cycle test.	131
3.55 The comparison between the actual and reference COG position X. .	131
3.56 The actual COG position Y.	132
3.57 The comparison between the actual and reference COG position Z. .	132
3.58 Pelvis Orientation of the Step Cycle Test	133
3.59 The actual and reference swing foot position X.	134
3.60 The actual and reference swing foot position Y.	134
3.61 The actual and reference swing foot position Z.	135
3.62 Swing foot orientation of the Step Cycle Test	136
3.63 A planar biped model in a 2-D simulation environment.	140
3.64 Modelling the dynamics of walking using an inverted pendulum model	142
3.65 The 3-D LIPM projected in the lateral plane (left) and the sagittal plane (right).	144
3.66 Dynamic behaviour of the LIPM	149
3.67 The external forces and moments that act on the biped.	151

3.68 A schematic diagram of the control architecture of the robot controllers	155
4.1 A schematic diagram of the control architecture of the ZMP-based controller.	162
4.2 A schematic diagram of the Stable Reference Trajectory Generation block in the ZMP-based controller.	163
4.3 Walking test of the ZMP-based controller: Time-elapsed freeze frames for the walking test in Webots.	167
4.4 Walking test of the ZMP-based controller: COG position (X) tracking	168
4.5 Walking test of the ZMP-based controller: COG position (Y)	168
4.6 Walking test of the ZMP-based controller: COG position (Z) tracking	169
4.7 Walking test of the ZMP-based controller: reference and modified reference ZMP position(X)	171
4.8 Walking test of the ZMP-based controller: reference and modified reference ZMP position(Z)	172
4.9 Walking test of the ZMP-based controller: absolute reference and modified COG velocity (X)	173
4.10 Walking test of the ZMP-based controller: absolute reference and modified COG velocity (Z)	174
4.11 Walking test of the ZMP-based controller: pelvis orientation tracking about the X axis	175
4.12 Walking test of the ZMP-based controller: pelvis orientation tracking about the Y axis	176
4.13 Walking test of the ZMP-based controller: pelvis orientation tracking about the Z axis	176
4.14 Walking test of the ZMP-based controller: swing foot position tracking in the X direction (SSL)	178

4.15 Walking test of the ZMP-based controller: swing foot position tracking in the Y direction (SSL)	178
4.16 Walking test of the ZMP-based controller: swing foot position tracking in the Z direction (SSL)	179
4.17 Walking test of the ZMP-based controller: swing foot position tracking in the X direction (SSR)	179
4.18 Walking test of the ZMP-based controller: swing foot position tracking in the Y direction (SSR)	180
4.19 Walking test of the ZMP-based controller: swing foot position tracking in the Z direction (SSR)	180
4.20 Walking test of the ZMP-based controller: swing foot orientation track- ing about the X axis (SSL)	181
4.21 Walking test of the ZMP-based controller: swing foot orientation track- ing about the Y axis (SSL)	182
4.22 Walking test of the ZMP-based controller: swing foot orientation track- ing about the Z axis (SSL)	182
4.23 Walking test of the ZMP-based controller: swing foot orientation track- ing about the X axis (SSR)	183
4.24 Walking test of the ZMP-based controller: swing foot orientation track- ing about the Y axis (SSR)	183
4.25 Walking test of the ZMP-based controller: swing foot orientation track- ing about the Z axis (SSR)	184
4.26 The pelvis and stance foot dimensions of ABL-BI.	188
4.27 Joint angles about the roll axis for the left leg in DB	190
4.28 Joint angles about the roll axis for the right leg in DB	191
4.29 Types of foot contact	193
4.30 Time elapsed frames of the torso roll at the start of DB in Webots. . .	195

4.31 COG position (X) at the start of DB, showing the effects of the torso roll to the COG in the lateral (X) direction.	196
4.32 Testing the inertial effect of the swing foot: the swing foot position (Z) in SSR	198
4.33 Testing the inertial effect of the swing foot: the ZMP position (Z) in SSR	199
4.34 Swing leg inertial effect test 1: the reference swing foot trajectory in the Z direction in SSL.	200
4.35 Swing leg inertial effect test 1: the reference and actual COG position in the Y direction.	201
4.36 Swing leg inertial effect test 1: the reference and actual COG position in the Z direction.	201
4.37 Swing leg inertial effect test 2: the reference swing foot trajectory in the Z direction in SSL.	202
4.38 Swing leg inertial effect test 2: Swing foot orientation about the Z axis.	203
4.39 Swing leg inertial effect test 2: the reference and actual COG position in the Y direction.	203
4.40 Swing leg inertial effect test 2: the reference and actual COG position in the Z direction.	204
4.41 Swing leg inertial effect test 3: the reference swing foot trajectory in the Z direction in SSL.	204
4.42 Swing leg inertial effect test 3: the reference and actual COG position in the Y direction.	205
4.43 Swing leg inertial effect test 3: the reference and actual COG position in the Z direction.	205

4.44 A comparison of the lateral ZMP position between two ABL-BI models, one with the original mass distribution and the other with a modified mass distribution.	208
5.1 A schematic diagram of the control architecture of the ZRAM-based controller.	218
5.2 The Dynamic Stable Gait block in the ZRAM-based controller.	219
5.3 The relationship between the reference CMP, the actual CMP and the COP	220
5.4 The Planar Model of ABL-BI.	222
5.5 The control architecture of the 2-D simulation.	224
5.6 Elapsed time snapshot of the planar biped walking test.	226
5.7 Planar test: The reference CMP, the reference COG, and the actual COG position (in the horizontal axis) with respect to the base frame.	227
5.8 Planar test: The horizontal and vertical component of the GRF.	227
5.9 Planar test: The error between the reference CMP and the COP and the actual CMP and the COP.	230
5.10 Planar test: The spin angular momentum of the planar biped.	230
5.11 Walking test of the ZRAM-based controller: the time-elapsed snapshots of the six step walking test in Webots.	236
5.12 Walking test of the ZRAM-based controller: the reference CMP, reference COG and actual COG position with respect to the world frame (X).	237
5.13 Walking test of the ZRAM-based controller: the reference COG and actual COG position with respect to the world frame (Y).	237
5.14 Walking test of the ZRAM-based controller: the reference CMP, reference COG and actual COG position with respect to the world frame (Z).	238

5.15 Walking test of the ZRAM-based controller: pelvis orientation tracking about the X axis	240
5.16 Walking test of the ZRAM-based controller: pelvis orientation tracking about the Y axis	240
5.17 Walking test of the ZRAM-based controller: pelvis orientation tracking about the Z axis	241
5.18 Walking test of the ZRAM-based controller: swing foot position tracking in the X direction (SSL)	243
5.19 Walking test of the ZRAM-based controller: swing foot position tracking in the Y direction (SSL)	243
5.20 Walking test of the ZRAM-based controller: swing foot position tracking in the Z direction (SSL)	244
5.21 Walking test of the ZRAM-based controller: swing foot position tracking in the X direction (SSR)	244
5.22 Walking test of the ZRAM-based controller: swing foot position tracking in the Y direction (SSR)	245
5.23 Walking test of the ZRAM-based controller: swing foot position tracking in the Z direction (SSR)	245
5.24 Walking test of the ZRAM-based controller: swing foot orientation tracking about the X axis (SSL)	246
5.25 Walking test of the ZRAM-based controller: swing foot orientation tracking about the Y axis (SSL)	246
5.26 Walking test of the ZRAM-based controller: swing foot orientation tracking about the Z axis (SSL)	247
5.27 Walking test of the ZRAM-based controller: swing foot orientation tracking about the X axis (SSR)	247

5.28 Walking test of the ZRAM-based controller: swing foot orientation tracking about the Y axis (SSR)	248
5.29 Walking test of the ZRAM-based controller: swing foot orientation tracking about the Z axis (SSR)	248
5.30 Walking test of the ZRAM-based controller: the reference CMP, COG and the actual COG for 18 step cycles (X)	250
5.31 Walking test of the ZRAM-based controller: the reference COG and the actual COG for 18 step cycles (Y)	251
5.32 Walking test of the ZRAM-based controller: the reference CMP, COG and the actual COG for 18 step cycles (Z)	251
5.33 Disturbance forces for the three perturbation tests in Webots.	253
5.34 Perturbation test I: The time-elapsed snapshots of the perturbation test.	255
5.35 Perturbation Test I: plot of the position (Z) of the ground reference points (Z) vs. time	256
5.36 Perturbation Test I: GRF (Z) vs. time plot	256
5.37 Perturbation Test I: The error between the reference CMP and the COP and the actual CMP and the COP.	257
5.38 Perturbation Test I: The pelvis orientation about the X axis.	257
5.39 Perturbation test II: The time-elapsed snapshots of the perturbation test.	259
5.40 Perturbation Test II: plot of the positions (Z) of the ground reference point vs. time	259
5.41 Perturbation Test II: GRF (Z) vs. time plot	260
5.42 Perturbation test III: The time-elapsed snapshots of the perturbation test.	261
5.43 Perturbation Test III: plot of the positions (X) of the ground reference point vs. time	261

5.44 Perturbation Test III: GRF (X) vs. time plot	262
5.45 Perturbation tests: The time-elapsed snapshots of the perturbation test for the ZMP-based controller.	263
5.46 Perturbation Test I: Reference COG (Z) and actual COG (Z) position of the ZMP-based controller.	264
C.1 The reference motion for the additional swing foot X position test (SSL)	294
C.2 The additional validation test results for the swing foot X position with respect to the stance foot for SSL.	295
C.3 The reference motion for the additional swing foot X position test (SSR).	296
C.4 The additional validation test results for the swing foot X position with respect to the stance foot for SSR.	297
C.5 The reference motion for the additional pelvis orientation test (SSL).	298
C.6 The additional validation test results for the pelvis orientation in SSL.	299
C.7 The reference motion for the additional pelvis orientation test (SSR).	300
C.8 The additional validation test results for the pelvis orientation in SSR.	301
C.9 The reference motion for the additional swing foot orientation test (SSL).	302
C.10 The additional validation test results for the swing foot orientation in SSL.	303
C.11 The reference motion for the additional swing foot orientation test (SSR).	304
C.12 The additional validation test results for the swing foot orientation in SSR.	305
D.1 The velocity of fixed frame with respect to the world frame	310
D.2 The COG velocity with respect to the world frame	312

Nomenclature

DOF	Degree of Freedom
ZMP	The Zero Moment Point
CMP	The Centroidal Moment Pivot
COG	The Center of Gravity
COP	The Center of Pressure
GRF	The Ground Reaction Force
LIPM	The Linear Inverted Pendulum Model
ZRAM	Zero Rate of Change of Angular Momentum
DB	Double Stance state
DB-Safe	Double Stance Safe state
SS	Single Stance state
SSL	Single Stance Left
SSR	Single Stance Right

DBL Double Stance Left

DBR Double Stance Right

HRL Hip Roll Left joint

HYL Hip Yaw Left joint

HPL Hip Pitch Left joint

KPL Knee Pitch Left joint

FPL Foot Pitch Left joint

FRL Foot Roll Left joint

HRR Hip Roll Right joint

HYR Hip Yaw Right joint

HPR Hip Pitch Right joint

KPR Knee Pitch Right joint

FPR Foot Pitch Right joint

FRR Foot Roll Right joint

Chapter 1

Introduction

In Ray Kurzweil's *The Age of Spiritual Machines*, the author raised an interesting question: *Can an intelligence create another intelligence more intelligent than itself?* It is a debatable question with different technological and philosophical aspects, but according to the author, artificial intelligence today exceeds humans in a variety of domain such as playing chess, selling/buying stocks and guiding cruise missiles [1]. However, some might argue that humans possess a far boarder range of capabilities and flexibility.

Incidentally, this thesis deals with a form of intelligence called bipedal locomotion. On the surface, there seem to be little in common between a robotic biped taking a step, and say, a man making his next move in a game of chess. But upon closer examination, a walking robot demonstrates a form of intelligence that is not very different from a human chess player:

- The human and the biped are both given a goal that they must complete under a set of rules (or constraints). The human player must win the game without moving his rooks diagonally and the biped must walk in a stable manner without moving its joints past its range of motion.

- The human and the biped both simplify problems by forming abstractions and deal with essential information that are relevant to the problem at hand. The human player will only consider chess pieces that are still on the board and have influence on the game. In a similar way, the biped forms simplified models that focuses on key state variables that affect its overall motion and stability.
- The human and the biped both obtain information about its current situation (or state) and respond by taking the optimal course of action. Some of these actions involve tradeoff and prioritization. For example, after surveying his situation on the chess board, the human player might sacrifice his Queen in order to save his King. Similarly, after acquiring sensory information, the biped might swing its foot backwards (instead of forward to take a step) in order maintain balance and prevent itself from falling.

The point of this comparison is to: 1) preview some of the characteristics of a biped walking controller (such as problem abstraction using simplifying models) that will appear again in this thesis, and 2) demonstrate that bipedal walking might involve some features of human intelligence.

Although walking bipeds can only perform a very specific task, they involve relatively primitive strategies that resembles some elements of human intelligence. The capability of current robots and artificial intelligences are still not comparable to the human brain. However, the pace of technology is accelerating exponentially. For example, as cited in [1], the exponential growth of technology in the first two decades of the twentieth century exceed that of the entire nineteenth century. Therefore, it is possible that walking bipeds are part of a “technological evolution” that will lead to a new form of intelligence on earth in our life-times!

This will be a discussion for another day; for now, let's return to the problem of bipedal locomotion and walking controllers.

1.1 Motivation and Challenges

1.1.1 Why Study Bipedal Locomotion?

The study of bipedal locomotion is motivated by applications in the service industry, and in the fields of biomechanics and robotics. These motivations are summarized below:

Service robots: Due to the rapid growth of retired population in the world (especially in Japan), it is predicted that in the near future, there will be a demand for service robots that can assist the elderly and the disabled in daily activities [1]. Service robots are expected to operate in enclosed human living environments with limited ground support surface. Since bipeds have small support bases and anthropomorphic (human-like) mechanical structures, they are more suited to operate in human environments than mobile robots.

Understanding how humans walk: The question of “how humans walk?” and the problem of controlling bipedal robots are closely related to one another. Contrary to intuition, it is actually difficult to understand human walking by conducting human walking trials and studying biomechanical data. First, it is difficult to derive the principles of human walking from biomechanical data alone. For this reason, it is also difficult to formulate the research objectives for these studies. Second, internal biomechanical information of the human body such as joint torques and command signals might be valuable to the study of human walking, but extremely difficult to measure in walking experiments.

Therefore, developing control strategies for bipeds might be an alternative approach to understanding human walking dynamics. Although the mechanical structure of the biped significantly differs from that of humans, it is believed that the basic principles of human and biped walking are essentially the same. Successful strategies that are developed for the biped suggest key characteristics and quantities that can be further studied in human walking trials. Hypotheses that are verified in human walking tests can in turn be applied to biped walking controllers. Therefore, the development of biped walking controllers and the biomechanical study of human walking mutually benefits one another. For safety and practical reasons, a biped can also be used to validate new rehabilitation devices before performing experiments on humans. Therefore, there is a demand in the biomechanical field for the development of biped robots.

Locomotion in rugged and discontinuous terrains: Although mobile robots are suitable for flat terrains due to the high rolling efficiency of wheels, it is more advantageous to use legs for rugged or discontinuous terrains such as hills and stairs. The reason is that the biped can step over obstacles and the foot can orient itself to attain foothold on discontinuous support surfaces. Although the locomotion control problem is a lot more complex for bipeds than for robots with wheels or tracks, bipeds are more flexible since they can *theoretically* traverse in all types of terrains. Therefore, they are more suited for applications such as search and rescue, mining, surveillance, and fire-fighting.

1.1.2 Challenges

Since there are many useful applications for bipeds, naturally, the following questions arise: Why do we not see bipeds in everyday life as predicted by science fiction writer Issac Asimov half a century ago? Why are bipeds still confined within research

laboratories? We believe that it is not because the necessary technology (motors, sensors, computers, battery) is lacking, although these elements do place limitations on the controller and performance of bipeds; rather, the main reason is that current control strategies do not have the flexibility, mobility and robustness to deal with everyday life situations.

Designing a control strategy that can match the capability of humans is an extremely difficult task. The following lists some of the common challenges that are pertinent to all bipeds, regardless of their size, weight, and the number of links and degrees of freedom (DOFs) that it has:

Stability: A robot that walks on two legs can tip over very easily. The reason is that bipedal robots are *underactuated* systems where the total number of controlled joints is less than the number of actual DOFs in the system. The contact between the stance feet and the ground can be considered as two 6-DOF *passive joints* that cannot be controlled by actuators. Instead, these passive joints can only be controlled indirectly by transforming the internal forces of the robot (joints torques) to Ground Reaction Forces (GRF) that are exerted on the contact points of the feet. Under large external disturbances, the robot might not be able to produce sufficient compensatory moment quickly enough and the robot would tip over. Therefore, the main goal of many robot controllers is to maintain balance while walking.

Energy Efficiency: As with other types of machines, it is desirable to maximize the energy efficiency of the design (ex: use actuators with high torque-weight ratio and light structural material) in order to reduce the power requirement and increase the battery life of the biped. In addition, many investigations in bipedal locomotion show that a major characteristic of natural human walking

is low energy consumption. Thus, an energy efficient controller can yield anthropomorphic walking gaits. Often, the requirement of energy efficiency come into conflict with stability. For example, a biped with large feet is more stable; however, it takes more energy for the biped to take a step during walking. Thus, having large feet can affect the walking speed and energy efficiency of the walking gait. Also, it is important to note that there is a coupling between the design of the walking controller and the mechanical structure of the biped. Therefore, in order to produce stable and energy efficient gaits, the controller and biped structure must *both* be optimized at the same time.

Computational Requirement: Bipeds often have a large number of links, DOFs and highly nonlinear and coupled dynamics. Therefore, it is impossible to consider the full body dynamics and generate online walking gaits at the same time. Currently, there are two approaches to this problem: 1) prepare stable trajectories offline and apply it to the biped (this approach sacrifices real-time gait generation), and 2) reduce the complexity of the dynamics by using a simplifying model (this approach sacrifices accurate modelling of the biped). For this reason, a compromised solution must be found so that the controller has the optimal performance.

Structural Interchangeability: Unlike a fixed mechanical structure, the structure of the biped changes throughout the walking cycle. Therefore, it is difficult to find a set of control objectives and parameters to satisfy all walking phases of the biped.

Ground Impact: Large impacts between the landing foot and the ground as the biped takes a step is detrimental to both the mechanical structure and the stability of the walking gait. Therefore, 1) the controller must be able to attenuate the disturbances that are caused by the ground impact, and 2) the sole of the

feet must have some compliance that absorbs the ground impact force.

1.2 Common Definitions and Stability Criteria in Bipedal Locomotion

1.2.1 The Center of Gravity (COG)

The Center of Gravity (COG) is generally used to represent the overall motion of the biped and the location of its ground projection determines its static stability. The COG (r_G) is calculated from the mass and COG location of the individual links of the biped (denoted by m_i and r_{Gi} respectively) and the total mass of the biped (M) in the following equation:

$$r_G = \frac{\sum m_i \cdot r_{Gi}}{M}. \quad (1.1)$$

1.2.2 The Center of Pressure (COP)

As shown in Figure 1.1, the foot contact forces (pressure field) of the human/biped can be represented by a resultant Ground Reaction Force (GRF) that is acting at the Center of Pressure (COP). It is important to note that the COP is an abstract quantity and does not reflect the actual contact between the foot and the ground. For example, if the biped is standing on the ground with the feet slightly apart (Figure 1.2), then the *overall COP* is located roughly between the support surface of the two feet. At the same time, each of the stance feet has a *local COP* that is located in their respective support surfaces.

The resultant force and moment that is induced by the GRF acting at the COP is equivalent to the force and moment produced by the pressure field. Therefore, the

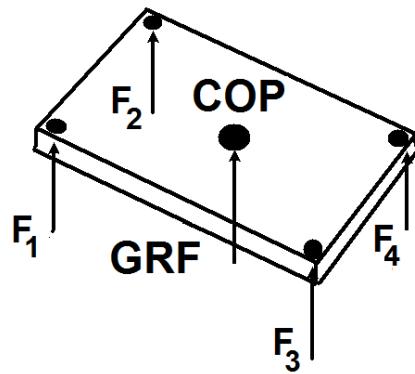


Figure 1.1: The foot contact forces (F_1 to F_4) can be represented by a resultant GRF acting at the COP.

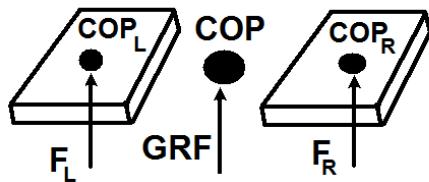


Figure 1.2: The overall COP and GRF of both feet and the local COP and GRF of the left foot (COP_L and F_L) and the right foot (COP_R and F_R).

COP can be calculated by the following equation:

$$r_P = \frac{\sum F_i \cdot r_{Pi}}{\sum F_i}, \quad (1.2)$$

where F_i and r_{Pi} represents the local foot contact force i and its point of action respectively and r_P is the COP.

1.2.3 The Zero Moment Point (ZMP)

The ZMP is a ground reference point that is commonly used as a stability criterion for walking controllers. The position of the ZMP with respect to the *support polygon* (an area on the ground that is spanned by the convex hull of the foot contact points) provides information on the dynamic stability of the biped: if the ZMP is within the support polygon, then the biped is dynamically stable. The ZMP is defined as a point on the ground at which the net moment of the inertial and gravitational forces has no component along the horizontal axes [2]. In Figure 1.3a, F_A represents the total forces that are acting on the links of the biped *above the ankle joint*. Likewise, M_A represents the resultant moment of the links above the ankle joint. In order to avoid foot rotation about the horizontal axes, the GRF (R) must act at point P within the support polygon to balance out the moment that are induced by F_A and M_A . This point is called the Zero Moment Point (ZMP). Note that we are only concerned with the rotation about the horizontal axes since we assume the ground friction force is sufficient to prevent any rotational movement about the vertical axis.

Theoretically, the ZMP cannot leave the support polygon. In the case where the support polygon is not large enough to encompass the ZMP (Figure 1.3b), the GRF will act on the edge of the foot and an uncompensated moment will cause the foot to rotate about its edge. The magnitude of the uncompensated moment is

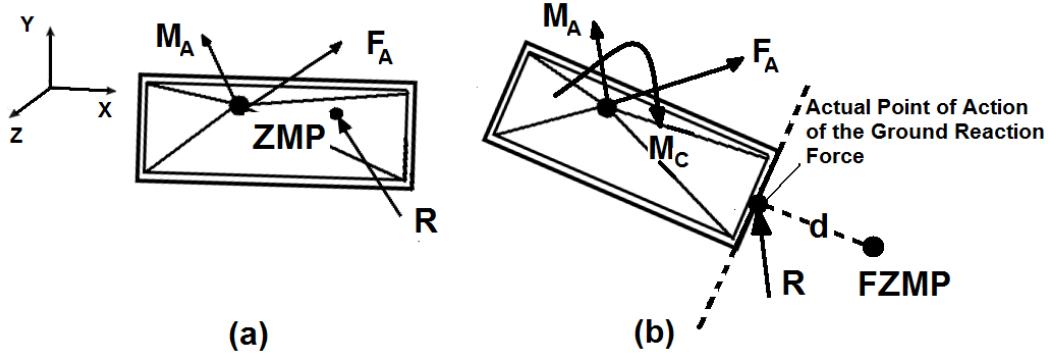


Figure 1.3: The Zero Moment Point.

proportional to the distance (“ d ” in Figure 1.3b) between the edge of foot and the “Fictitious ZMP” (FZMP), which is a point where the GRF will have to act in order to generate a sufficient compensatory moment. The FZMP can exist outside of the support polygon. In order for a robot to be dynamically stable at all times, the ZMP must lie within the support polygon throughout its walking cycle.

1.2.4 The Centrioidal Moment Pivot (CMP)

Biomechanical studies in human walking show that the spin angular momentum (the angular momentum about the COG) is a highly regulated quantity that is conserved throughout the walking cycle [3]. Inspired by these studies, the CMP was developed as a measure of the rotational stability of the biped. It is defined as a point at which the GRF must act in order to ensure zero moment about the COG (zero centroidal moment) [3]. This condition is also known as the zero rate of change of angular momentum (ZRAM) condition, where $\dot{H}_G = 0$, and H_G is the spin angular momentum of the biped. The CMP position relative to the COP is an important indication of the biped’s rotational stability, stability margin, and magnitude of the uncompensated moment. If there is zero centroidal moment, then the COP coincides with the CMP (Figure 1.4a) and the line of action of the GRF is parallel to the vector

that connects the COP and the COG. The location of the CMP can be obtained by moving the line of action of the resultant GRF until it intersects with the COG. The intersection point between the “shifted line of action” and the ground is the location of the CMP. However, as shown in Figure 1.4b, if the line of action of the GRF does not pass through the COG, then a centroidal moment that is proportional to the distance between the CMP and the COP will be created.

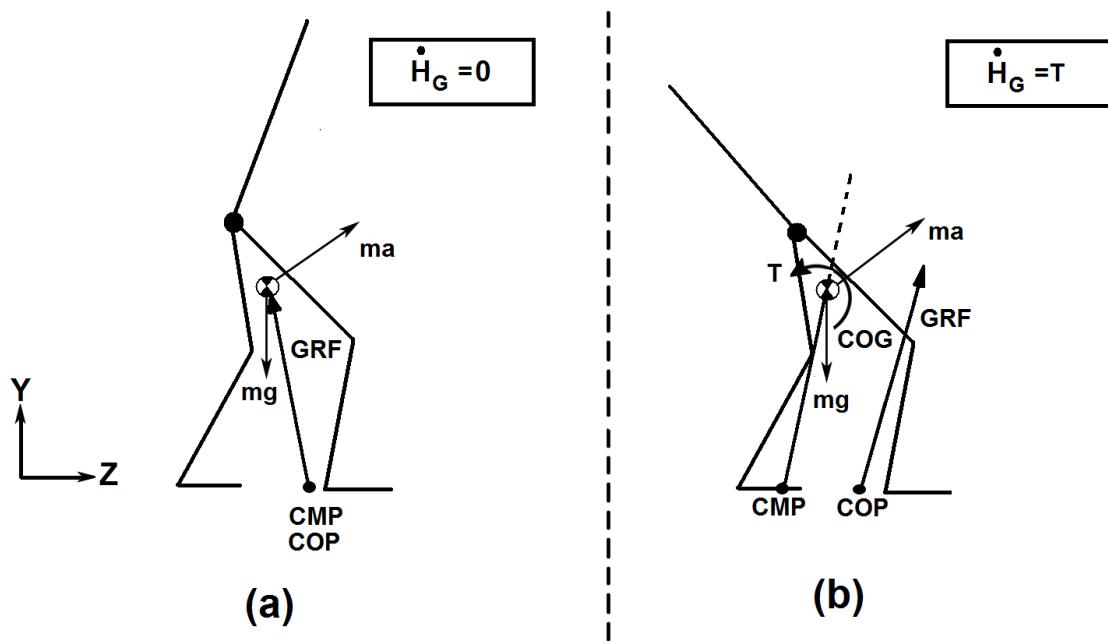


Figure 1.4: The Centroidal Moment Pivot.

1.2.5 Static vs. Dynamic Stability

There are two types of stability for bipedal locomotion. The first type is called *static stability* and it applies to slow moving bipeds. The ground projection of the COG is kept within the support polygon and the dynamic effects of the biped is minimized. In other words, the posture of the biped can be treated as a “static structure” at any time instance and the biped can stop without falling.

In contrast, the second type of stability is called *dynamic stability*. The COG is not restricted to within the support polygon and the biped can move at a faster speed. In order for the biped to be dynamically stable, the ZMP must be kept within the support polygon (this is also known as the ZMP stability criterion). Dynamically stable gaits are faster and more efficient than statically stable gait since there are less restrictions on the kinematic posture and acceleration of the biped. However, the biped might not be able stop at any given time since it is in a “constant falling state”. In order to stop in a stable manner, the biped must reduce its speed to regain static stability.

1.3 Research on Bipedal Walking

As with most engineering problems, there are many different solution approaches to the problem of bipedal locomotion. This section briefly reviews some of the control algorithms that were developed in previous works to generate dynamically stable walking gaits.

1.3.1 Offline Trajectory Generation

A common approach to the development of stable walking controllers is trajectory tracking. This method involves tracking a set of joint trajectories that are generated based on an a priori stability criterion (ex: ZMP criterion for dynamically stable gaits). For example, the controller for WABIAN at Waseda University [4] planned a reference motion for the lower body while an iterative algorithm was used to compute the corresponding torso reference trajectory that maintains the ZMP within the support polygon. Since the preplanned trajectories are not modified online, precise knowledge of the biped dynamics (ex: mass, moment of inertia and link COG position) is required to minimize the effects of modelling errors in the calculated

compensatory torso motion. Takanishi et al. [5], Hirai et al. [6], and Konno et al. [7] also proposed similar methods to generate offline dynamically stable gaits based on the ZMP criterion.

Huang et al. [8] proposed a more sophisticated method to generate the swing leg motion. Based on known obstacles and ground conditions, kinematic constraints on the foot motion were specified (ex: foot orientation at foot landing and maximum vertical height of the swing foot) and the foot trajectories were generated using the cubic spline interpolation method. The torso trajectory was calculated as the output to the ZMP position equation [2] to maximize the stability margin of the biped (ie: the objective of the torso motion is to manipulate the ZMP as close to the center of the support polygon as possible).

Various optimization methods were also used to generate offline trajectories. For example, in [9], Arakawa et al. applied hierarchical methods (includes genetic algorithm and evolutionary programming) to generate trajectories that minimizes the actuator torque and maximize the stability margin of the biped.

In summary, most offline trajectory generation methods derive upper body motions (based on the equations of the ZMP position) to maintain the ZMP within the support polygon and compensate for the effects of external forces (including the GRF) and the inertial effects of the lower body motion. The advantage of this method is that a large stability margin can be achieved if the ZMP is kept in the middle of the support polygon. The disadvantage is that ZMP manipulation is limited by the mass of the torso. Since the torso is relatively heavy, the hip actuator might not be able to generate the required torso motion to manipulate the ZMP. Thus, there are limitations to the level of external disturbances that the controller can handle.

1.3.2 Model-based Online Trajectory Generation

It is difficult for a biped to interact with the environment if it only relies on tracking a set of preplanned trajectories. The reason is that it is impractical to plan a set of trajectories for every conceivable situation for the biped. To address this issue, the reference trajectories must be generated in real-time with minimal motion planning. To facilitate this, the complex dynamics of the biped is first simplified by a *reduced dynamics model* that has similar dynamic behaviours as the biped. Kajita et al. [10] proposed a 3-D Linear Inverted Pendulum Model (3D-LIPM) that approximate the dynamics of the biped and establishes a relationship between the COG and the ZMP positions. In an extended work, Park et al. [11] developed the Gravity Compensated Inverted Pendulum Model (GCIPM) to account for the inertial effects of the swing leg motion and generated stable walking gaits using the computed torque control method. Komura et al. [12] developed an Angular Momentum Pendulum Model (AMPM) that focuses on controlling the rate of change of angular momentum of the biped. For these methods, dynamically stable reference COG trajectories are generated from the dynamic equations of the reduced dynamics model. The reference COG is then decomposed into reference joint velocities by applying kinematic resolution methods that are commonly used in redundant manipulators. Thus, the controllers perform motion planning and control *only* using several key state variables (ex: the COG and the ZMP) of the biped and the computational requirement of the controller is reduced. The drawback of this approach is that the reduced dynamics model might not fully capture the actual biped dynamics. Therefore, feedback control is required to account for modelling errors as well as external disturbances.

Different controllers that are based on the above models were developed. In [13], the controller uses the LIPM to generate reference COG trajectories and include a

term in the control input that is a function of the future ZMP reference trajectory. This enables the controller to perform “preview control” by taking into account the time evolution of the ZMP trajectory and significant changes in the states of the biped. In [14], Nishiwaki proposed a fast online gait update method that uses the LIPM to generate modified reference trajectories in real-time and connects them smoothly to the existing reference trajectories. Thus, reduced dynamic models are used in different ways to generate stable reference gaits in real-time controllers.

1.3.3 Virtual Model Control (VMC)

The VMC is a control framework that was first proposed in [15]. The main idea is to attach “virtual” mechanical components such as springs and dampers to specific points on the biped (such as the local COG position of the torso) and facilitate control using virtual forces that are generated from the virtual components. The virtual forces can be decomposed into actual torques of the biped using the inverse Jacobian matrix that relates the attached frame to the reference frame. Thus, the biped generates the equivalent forces/torques that would have been generated by the virtual components had they existed. The VMC was verified in several robots including the Spring Turkey in [16] and the Spring Flamingo in [17].

1.3.4 Central Pattern Generators (CPG)

In biological systems, rhythmic motions such as walking are controlled by motion generators called CPG that can generate trajectories without the need of any rhythmic input. Adopted to walking controllers for bipedal robots, CPGs are often modelled as a network of differential equations called neurons that are used to plan oscillatory reference joint trajectories without any sensory feedback [18]. However, deviations from the reference motion due to external disturbances are usually attenuated by feedback

control. By coupling the signals that are generated by the different neurons (ex: the neurons that control the left and right hip joints), the motion of two different limbs can be synchronized. The CPG plays an important role in some controllers that are based on evolutionary programming and adaptive learning. For example, in [19] the CPG generated a nominal periodic gait and “peripheral” controllers intervened only when the motion from the CPG contradicted their individual control policies. Each peripheral controller then corrected the errors in the CPG and issued an evaluation of the general behaviour. Gradually, using reinforcement learning, the CPG was trained to adopt a control policy. However, the training time was reported in [19] to be long (about 5 hours in simulation). In summary, CPGs generates reference joint trajectories based on user-specified objectives such as a target walking velocity. They are mostly used within a reinforcement learning framework and in conjunction with other optimization methods.

1.3.5 Passive Dynamic Walkers

In contrast to powered bipeds, *fully* passive walkers such as the McGeer walker [20] do not have any actuated joints and rely solely on an impulse to initiate the walking motion and gravity to walk down a slope. Since the joints are not controlled, the main drawback of fully passive walkers is that it is impossible for the walker to achieve user-specified objectives such as changing the walking direction or stopping. Regardless, the main conclusion that can be drawn from the performance of passive walkers is this: instead of strictly controlling every joint in the biped to track a preplanned gait, it is possible to rely on the unforced response of the biped to perform walking and only apply joint actuation when it is necessary. Active control might be required to change the biped walking speed or attenuate external disturbances. For example, in [21], the author proposed a control algorithm that enabled a one-legged hopping robot ARL-Monopod II to achieve highly efficient running while maintaining

balance with a stabilizing controller. However, the strategy also relied on a complex leg design with compliant elements such as a hip spring and a leg with variable length. The same requirement also applied to the Spring Flamingo in [17] where a compliant ankle and a kneecap was required to passively manipulate the COP and increase the energy efficiency of the swing leg motion. Other robots with passive elements were also proposed in [22–24]. It is important to note that regardless of whether the biped is passive or actively controlled, energetics an important aspect in bipedal walking. For example, the controllers for powered bipeds in [25–27] produced efficient walking gaits by minimizing the joint torque output in various space-time optimization techniques.

Since the mechanical structure ABL-BI does not have any compliant elements, we opt for a more pragmatic trajectory tracking approach. In the ZMP-based controller and the ZRAM-based controller, we simplified the full dynamics of the biped using different reduced dynamic models and generate stable trajectories in realtime with feedback control.

1.4 Objectives

ABL-BI was the first biped developed at the Advanced Biomechatronics and Locomotion Laboratory (ABL-Lab) at Carleton University. The design, manufacturing and assembly of the mechanical structure of ABL-BI was done by Richard Beranek and the realtime control software and hardware architecture was developed by Owain Barker. The development began in August 2008 and a large portion of the work was completed by April 2010. Concurrently, initial walking controllers (which are the focus of this thesis) were designed, implemented and verified on a 3-D simulated biped. Aside from the obvious goal of enabling the biped to walk in a stable manner,

the walking controller was developed to: 1) test the electro-mechanical components of ABL-BI, and 2) evaluate the performance of the real-time software architecture in an operational test. Currently, the controllers have not been implemented on the actual platform; however, simple walking based on preplanned joint trajectories had been implemented on the biped to test the actuators and sensors. In addition, simulation results show that the controllers can generate stable walking gaits and are ready to be implemented on the biped.

The main research objectives of this thesis are listed as follows:

1. Develop a simple walking controller that can generate stable walking gait in real-time and reject external disturbances by modifying preplanned high level trajectories online. In addition, the controller must be based on the underlying principles of bipedal walking and dynamic stability criteria so that it can be applied to bipeds with arbitrary kinematic specifications (ex: number of links and DOFs). Finally, the controller must only rely on sensory information that are available in real-life in order to be transferred directly onto the actual platform in the future. This requires: 1) an accurate kinematic model of the actual biped, and 2) the forward kinematics of the biped in order to compute physical states that are not directly measurable from real-life sensors.
2. From simulation tests, evaluate how well the mechanical design of ABL-BI lends itself to dynamically stable walking and recovery from external disturbances. Certain aspects of the design such as the dimension of the feet, the mass distribution of the links, and the stiffness of the joints can, along with the design of the walking controllers, affect the walking performance of ABL-BI.
3. Verify through simulation that the controllers are able to generate stable walking gaits and attenuate external disturbances.

1.5 Contributions

The following are the research contributions of this thesis:

1. The main contribution is the development of a new control strategy (henceforth referred to as **the ZRAM-based controller**) that is based on regulating the rate of change of angular momentum. Although many angular momentum-based controllers had been proposed in the past, this is the first real-time walking controller that exploits the relationship between the COG, CMP and the GRF to maintain the ZRAM condition. By focusing on key state variables that represents the overall motion stability of the biped, the ZRAM-based controller is relatively simple and requires less computation than other existing controllers. Unlike most ZMP-based controllers, the ZRAM-based controller can modify the reference COG trajectory online in the presence of disturbances to increase the flexibility and robustness of the biped. Since the controller is based on the fundamental principles of bipedal walking (the ZRAM criterion), it can be applied to bipeds with an arbitrary number of DOFs. The controller was implemented and validated on a 3-D simulated model of ABL-BI to satisfy the research objectives. The underlying principles and simulation results of this controller is provided in Chapter 3 and Chapter 5.
2. A model of the biped was developed based on the specification of the actual ABL-BI platform in a 3-D simulation environment (Chapter 2). A new kinematic modelling approach of decomposing the full kinematics of the biped into open kinematic chains was proposed and implemented (Chapter 3). The forward kinematics equation of the simulated biped was derived to calculate important state variables such as the swing foot and COG position. Thus, the walking controllers only relied on “real-life” sensors (the joint encoders) to obtain the output states of the biped. In addition, a Closed-loop Kinematic Controller

was developed and implemented on the simulated model for postural control. The controller was validated in a series of tests in simulation and the results are provided in Chapter 3.

3. A controller based on the LIPM and the ZMP criterion (henceforth referred to as **The ZMP-based controller**) was developed in Chapter 4. The ZMP-based controller is largely based on an existing realtime walking controller; however, several modifications were made to simplify the motion planning and postural control of the biped. Simulation results were used to validate the ZMP-based controller and identified some issues in the simulated ground model, the control architecture of CI, and the mechanical design of ABL-BI that affected the walking speed of the biped. Additional simulation trials were conducted to examine the effects of these issues. Several suggestions were made to improve the walking performance of ABL-BI.

1.6 Outline

This remainder of this thesis is organized as follows:

Chapter 2: ABL-BI: The Experimental Platform and the Simulation Model

This chapter begins with a brief description of the mechanical, electrical, and software design of the actual ABL-BI platform. Next, a simulated biped model based on the mass and kinematic specification of the actual platform is developed in a 3-D simulation environment called *Webots*. Several important features of *Webots* such as the simulated sensors, the control levels of the simulation program, and the ground model are also described in this chapter.

Chapter 3: The Essential Elements of the Robot Controllers

This chapter provides an in-depth description on the essential components that are common to the two walking controllers in this thesis. First, the locomotion state machine is introduced to separate the walking cycle into several sequential walking phases. Second, the kinematic arrangements of the biped model is described and the solution approaches to the forward and inverse kinematics problem are developed. Simulation test results are provided to verify these solution approaches. Third, reduced dynamic models that are used in the proposed walking controllers are presented. Finally, this chapter concludes with a description on the control architecture of the walking controllers.

Chapter 4: A Walking Controller Based on the LIPM and the ZMP Stability Criterion

This chapters describes the initial controller implementation of ABL-BI. The walking controller is based largely on an existing work that uses the linear inverted pendulum model to simplify the dynamics of the biped and the ZMP stability criterion. A walking test is conducted in 3-D simulation to validate the controller and a discussion on the results is provided. This chapter also discusses several possible factors that limit the walking performance of ABL-BI.

Chapter 5: A Walking Controller Based on CMP Planning and COG Position Control

This chapter presents a walking control strategy that is based on regulating the rate of change of angular momentum. First a discussion on the underlying principles of human walking and the related biomechanical studies is provided. Second, a new feedback control strategy that exploits the relationship between the COG, COP and

the GRF is developed. Third, simulation results are provided to evaluate the effectiveness of the controller in generating stable walking gaits in the presence of external disturbances.

Chapter 6: Conclusions

This chapter summarizes the important results and conclusions of the research and provide suggestions for the future development for this project.

Chapter 2

ABL-BI: The Experimental Platform and the Simulation Model

One of the main objectives of this project is to develop walking control algorithms for ABL-BI. Although the end goal is to implement a walking controller on the actual biped, computer simulation and a simplified model of the biped is used extensively for development and validation. Therefore, before presenting the proposed control strategies and evaluate their performances, it is necessary to first provide a description of the ABL-BI platform, its simplified model, and the simulation environment.

This chapter is organized as follows: Section 2.1 briefly outlines the design objectives and key areas in the mechanical design and software architecture of the ABL-BI platform [28]. Section 2.2 presents the kinematic description and robot parameters of the simplified simulation model and an overview of the simulation development environment *Webots*. Finally, a brief summary of this Chapter is provided in Section 2.3.

2.1 The ABL-BI Platform

This section presents a brief overview on the design of the ABL-BI platform. A description of the platform can be found in [28].

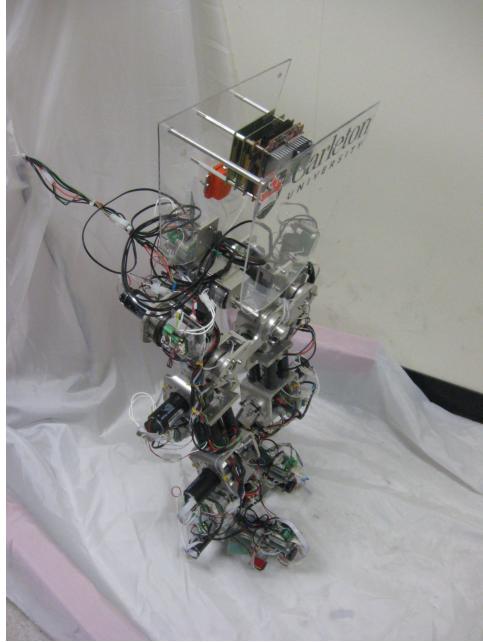


Figure 2.1: The ABL-BI Platform.

ABL-BI is a bipedal robot designed for studies in bipedal locomotion and research in robotic rehabilitation devices (Figure 2.1). The kinematics arrangements, link dimensions, and the number of DOFs of the biped is mostly adopted from anthropomorphic data [29] and bipeds with similar specifications [30]. The overall mechanical specifications for ABL-BI is given in Table 2.1. The biped has 13 internal (actuated) DOFs with 6 DOFs for each leg (3 DOFs for the hips, 1 DOF for the knee, and 2 DOFs for the ankles) and 1 DOF in the torso. A major challenge for the biped design is the actuator selection. Since the mass of the motors composes up to 60 percent of the overall mass of the biped, they contribute significantly to the torque requirements of the biped. An iterative selection process was performed in computer simulations

Table 2.1: ABL-BI specifications.

Total Height	1.0 <i>m</i>
Height from Leg to Hip	0.6 <i>m</i>
Height of the Torso	0.4 <i>m</i>
Total Mass (including batteries)	21.9 <i>kg</i>
Internal Degree of Freedom	13

to select actuators with the appropriate mass and torque output. In addition, the required maximum angular velocity of actuators is estimated to be 5 *rads/s* based on the target walking speed of 0.7 *m/s*. Based on these requirements, two different motor-gear head couples are selected. For the hip pitch, hip roll and knee pitch joints, the Maxon RE40 brushed DC motor coupled with CSG171202UH harmonic drives are used and for the remaining joints, the Maxon RE30 coupled with CSF141002XHF harmonic drives are used. The actuator specifications are summarized in Table 2.2.

Table 2.2: Actuators specifications.

Motor Model	RE40:148867	RE30:310007
Gearhead Model	CSG171202UH	CSF141002XHF
Gear Ratio	120	100
Mass (<i>g</i>)	1160	533
Nominal Torque (<i>Nm</i>)	14.28	5.95
Nominal Speed (<i>Rpm</i>)	58.0	80.5
Torque at 50 <i>Rpm</i> (<i>Nm</i>)	40.2	29.4

The distributed hardware architecture of ABL-BI consists of the following major components:

1. An onboard embedded computer (model: VersaLogic EMP-32 Cheetah) that operates on a realtime operation system called the Xenomai Linux.

2. A Control Area Network (CAN) bus that provides the communication channel for all the actuators, sensors, and the controller.
3. A digital servo drive called the ELMO-Whistle that provides local control and power amplification for the biped.

An important feature of this control architecture is the use of the CAN bus. The advantage of the CAN bus is that it provides flexible digital communication and reduces the required number of wires for communication, compared to an analogue system. Due to the limitation imposed by the Elmo drives, a maximum bandwidth of $250Hz$ is achieved on ABL-BI.

2.2 Webots and the ABL-BI Simulation Model

2.2.1 Overview of Webots

In order to validate and demonstrate the effectiveness of the proposed controllers, a simplified biped model is created in a simulation software called *Webots*. The main advantage of *Webots* is that it allows the user to prototype biped models and test walking controllers quickly without having to deal with the mechanical, electrical and safety issues that are associated with a real biped. As with all simulations, there is a tradeoff between the accuracy and simplicity of the biped model. On one hand, the model must capture the essential kinematic and dynamic properties of the ABL-BI platform; on the other hand, the model must simplify the complex geometries and dynamic properties of the ABL-BI platform in order to ease the computational requirements of the simulation. In the view of this, the structure of the actual biped is reduced to simple geometric shapes in the biped model; but at the same time, the link dimensions, link mass, and the link moment of inertia of

the ABL-BI model in *Webots* is defined to match those of the ABL-BI platform as closely as possible. Note that the link mass and link moment of inertia of the ABL-BI platform were obtained from the Pro-Engineer (Pro-E) CAD biped model.

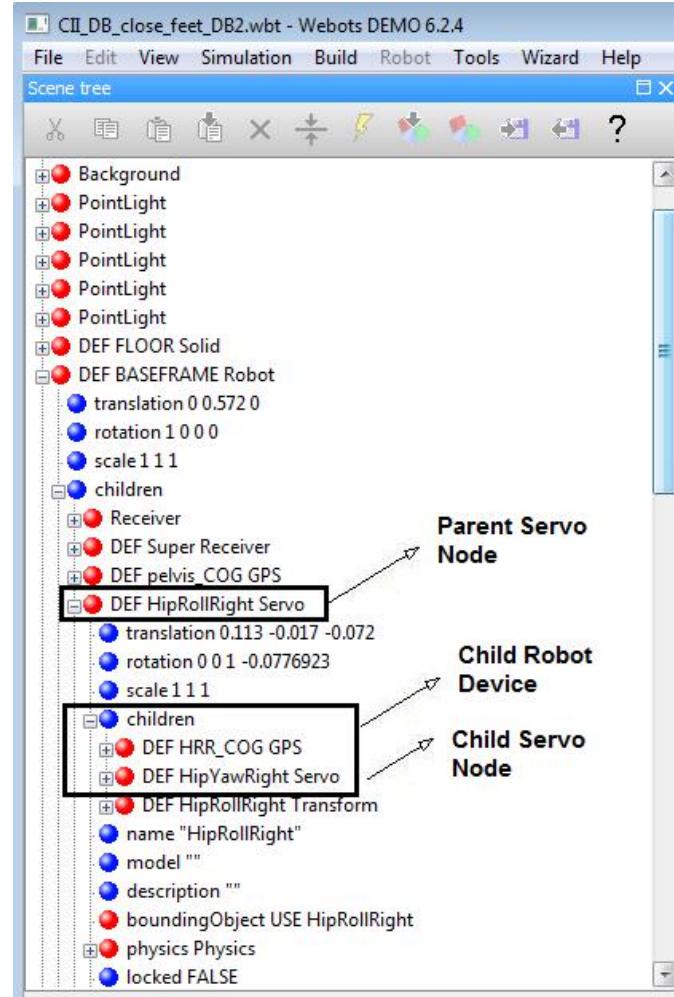


Figure 2.2: The world editor and the hierachal modelling structure in Webots. The child Servo node is defined and moves with respect with the parent Servo node. Similarly, the child robot device (GPS sensor node) is defined with respect to the parent Servo node.

Webots has two main features: a user interface called the *world editor*, and the robot controller.

1. The world editor allows the user to create a virtual environment called the

World, construct the structure of the biped model, and define the physical properties of the *World* and the biped model. The simulated objects in *Webots* are called *nodes*. The biped model has a hierarchical structure that is composed of simulated links and joints nodes (Figure 2.2). There is also has a special group nodes called “robot devices” that can be implemented on the biped model. Robot devices are simulated onboard sensors that provide information on the physical states of the biped to the robot controller. Examples include GPS sensors, force sensors, and distance sensors.

2. The robot controller program is written in C++ by the user to control the behaviour of the biped. The robot controller can interact with the simulated biped through a C++ API (Application Programming Interface) that contains a library of functions, data structures and object classes that are used in the C++ controller code.

The Physics Engine of Webots

Webots uses a physics engine called the Open Dynamics Engine (ODE) to simulate the dynamic behaviour of a system of rigid bodies that are connected together by joints (a system of articulated rigid bodies). The following are some of the features of ODE that are pertinent to the robot simulation in *Webots*:

1. *Rigid Bodies*— Rigid bodies in the ODE are composed of two components: frame and shape (Figure 2.3). The frame contains information about the kinematic properties (ex: position and orientation) and mass properties (ex: mass and the moment of inertia) of the rigid body. The origin of the frame is located at the center of mass of the body. The shape contains the geometrical properties of the rigid body (ex: size and shape) and is used for collision detection in the simulation.

2. *Joints*— In ODE, joints are modelled as constraints that enforce two rigid bodies to have a certain linear and angular motion relative to each other. The constraint forces are calculated and are applied to the rigid bodies at every time step.
3. *Collision Handling*— At every time step, the ODE detects the intersections between the shapes of two rigid bodies. Special temporary joints called contact joints are created at each of the contact points and constraint forces are applied at the contact joints. The constraint joints are removed at the end of each time step.
4. *Integrator*— ODE uses a first order fixed step integrator that emphasizes on speed and stability over accuracy. However, higher accuracy can be achieved with smaller time steps. The simulations of ABL-BI use a time step of 1 millisecond.

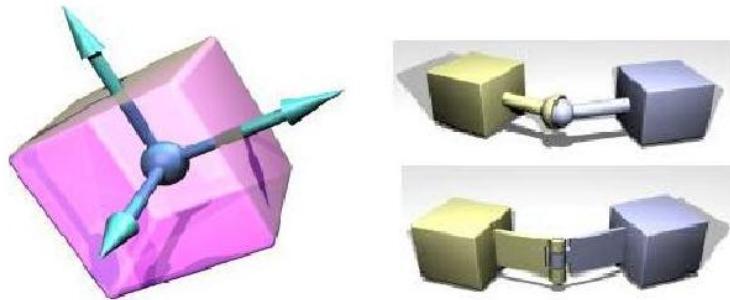


Figure 2.3: The left figure shows an ODE frame and space of a rigid body. The right figure shows ODE joints that constrain the relative motion between two rigid bodies (from the ODE manual: <http://www.ode.org>).

Simulation Time vs. Actual Time

In *Webots*, the simulation time and the actual time are two different notions. The simulation time can run at a speed that is faster or slower than the actual time

depending on the complexity of the robot controller, the computational power of the computer that is running *Webots*, and the number of programs and background processes that are running at the same time. The simulation time of the proposed controllers is about ten times slower than the actual time. However, since the computer operating system (OS) that is used for the ABL-BI platform is designed to perform a small number of dedicated functions and has a higher performance than the simulation computer, the robot controller can meet the real-time constraints of 250 Hz on the ABL-BI platform.

The robot simulation and the controller run at a fixed time step of 1 millisecond (of the simulation time). The simulation step is the rate at which the dynamic behaviour and the sensor values of the simulated biped is updated and the controller step is the rate at which the instructions of the robot controller are executed. Note that the robot controller runs at the same rate (in simulation time) as the simulated biped. This implies that the physical behaviours and sensor information update of the simulated biped is synchronized with the actuation commands generated by the robot controller.

Two versions of *Webots* are used to test the proposed controllers. For the ZMP-based controller, Webots 5.9.2 Educational Version is used and for the ZRAM-based controller (the ZRAM-based controller), Webots 6.2.4 Professional Version is used. Note that the functionalities that are provided by the Supervisory Controller (Section 2.2.4) and the Physics Plugin (Section 2.2.4) are only available in the Professional Version and are used by the ZRAM-based controller only.

2.2.2 The Features of the Simulation Model in Webots

This section describes the modelling of the actual biped through the use of different simulated components in *Webots*. In addition, the frame assignments, mass properties, units, and rotational matrix conventions of the biped kinematic model is also presented.

Nodes

The complex geometries of the structural components of ABL-BI is represented by a model (Figure 2.4) composed of simple geometric shapes (henceforth referred to as the *Simulation Model*). The Simulation Model is to be distinguished from the *Dynamic Model* introduced in Chapter 3). The Simulation Model in *Webots* has a hierachial structure: the position and motion of a “child” node is defined with respect to its “parent” node. A node can be a structural component such as a *Solid*, a user defined local coordinate frame such as a *Transform*, an actuated DOF such as a *Servo*, and a robot device such as a GPS sensor (Figure 2.2).

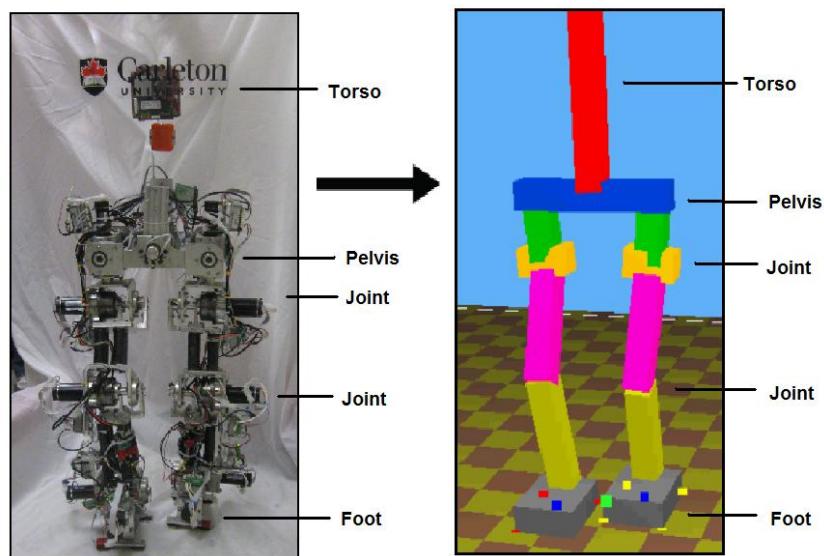


Figure 2.4: The left figure is the ABL-BI platform. The right figure is the ABL-BI model that is composed of simple geometric shapes.

Links and Frames

The Simulation Model consists of a series of rigid bodies called *links*. A link has the following attributes: 1) external shape, 2) mass, 3) the local Center of Gravity position (link COG), 4) the mass distribution of the link that is defined an inertia tensor, and 5) a boundary enclosure that is used for collision detection (equivalent to the *shape* object in ODE). The links and motors of the ABL-BI platform are represented by rectangular prisms with defined physical properties in the Simulation Model. Each pair of neighboring link is connected by a 1-DOF revolute *joint*. In order to deal with the kinematics of a complex mechanism such as ABL-BI, a common approach is to attach frames to different parts of the mechanism and then describe the kinematic relationship between the frames. This approach is used to describe the kinematics of the Simulation Model. The relative position and orientation of link i and link $i - 1$ is characterized by the relative position and orientation of Frame i and Frame $i - 1$ that is attached to joint i and $i - 1$ respectively (Figure 2.5).

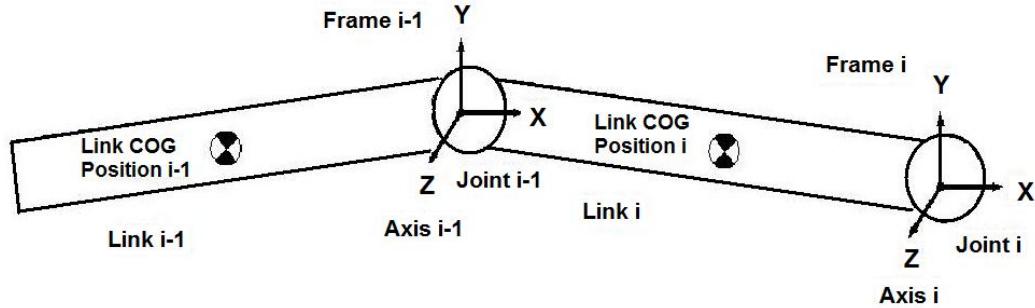


Figure 2.5: A pair of links that are connected by a joint. Frames are attached to the joints and are used to describe the relative position and orientation of two rigid links.

Note that since there is no relative motion between the joint and the link that is attached to it (ie: the link is rigid and it rotates with the joint), the relative motion between links can be equivalently described by the relative motion between their corresponding frames. In this thesis, the term *joint* mostly refers to the mechanical

connection between two rigid bodies and *frame* refers to a local coordinate system that is used to describe the position and orientation of an object. A schematic diagram of the ABL-BI Simulation Model is presented in (Figure 2.6).

Actuated and Passive DOFs

In the ABL-BI platform, the rotation of link i about the joint axis of joint i is actuated by a motor. Equivalently, a joint in the Simulation Model is actuated by a simulated motor called the *Servo*. This type of joint constitutes an *actuated* DOF which differs from a *passive* DOF. An actuated DOF is an internal joint that can be actuated and controlled by a motor whereas a passive DOF cannot be controlled. In total, ABL-BI (platform and model) has 25 DOFs with 13 actuated DOFs and 12 passive DOFs (Figure 2.6). The relative motion of each pair of neighboring links of ABL-BI (and hence its overall gait) is controlled by 13 actuated DOFs and the contacts between the ground and the feet of the biped can be treated as two 6-DOF passive joints. As mentioned in Chapter 1, since a restorative moment cannot be generated at the 6-DOF ground-foot joint directly, the biped is an underactuated system and has a tendency to tip over under large disturbances.

Link Properties and Frame Assignments in the Simulation Model

In *Webots*, the position and orientation of a frame and the physical properties of its corresponding link can be defined by a single node called the *Servo* (Figure 2.7). In the Simulation Model, the *Servo* is also a simulated motor that provide joint actuation. Followed from the hierachal structure in *Webots*, the location and motion of a child Servo is defined with respect to its parent Servo (ie: the location and orientation of Frame i is defined with respect to Frame $i - 1$). The physical properties of the links are defined in *physics nodes* that are associated with each Servo node. The definable physical properties for a single Servo in the Simulation

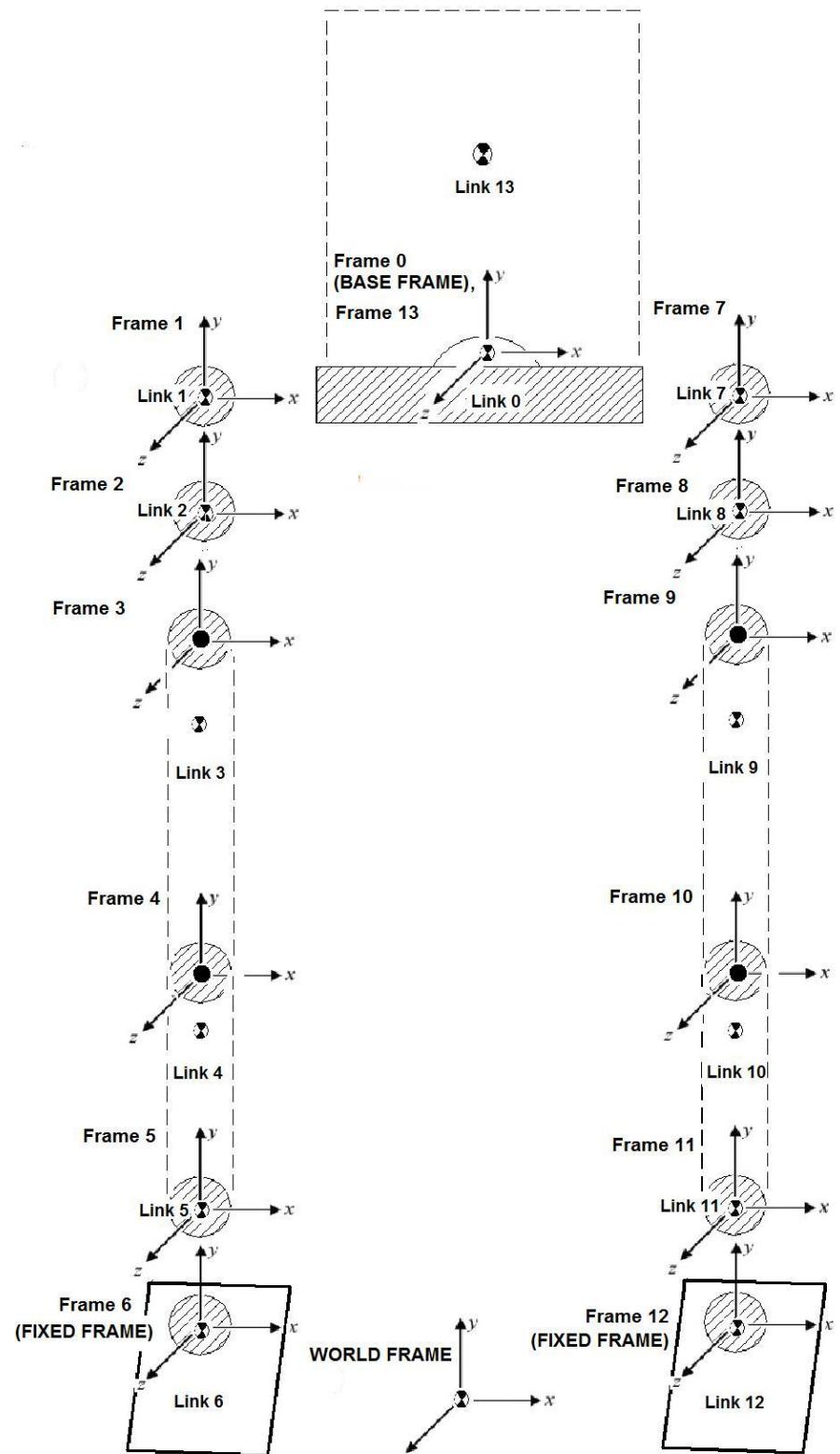


Figure 2.6: The Simulation Model of ABL-BI.

Model is summarized in Table 2.3.

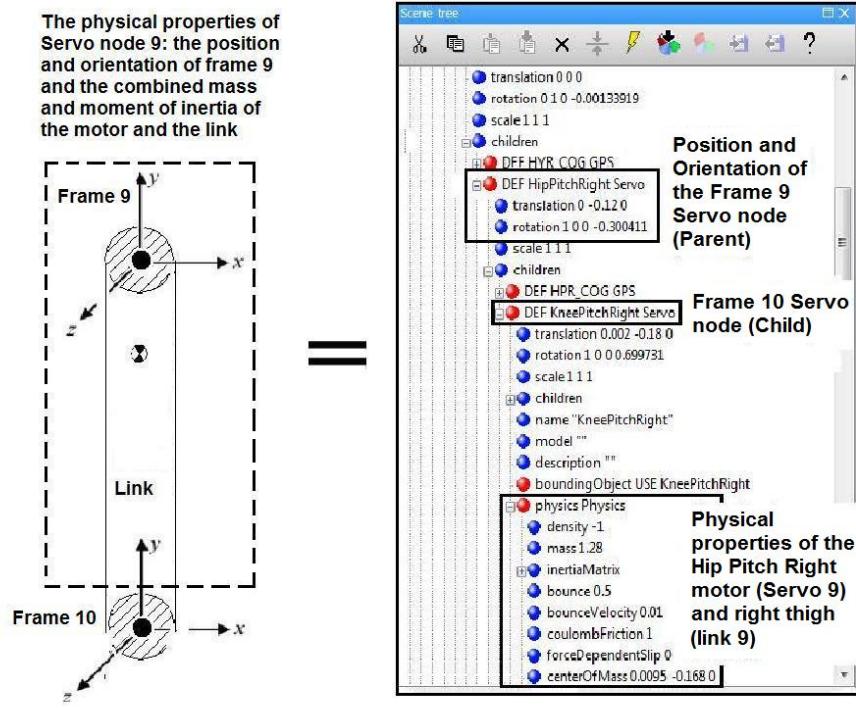


Figure 2.7: The Servo Node in Webots contains information on the frame position and orientation and the physical properties of the links.

Note that since most of the mass of the biped is concentrated in the joints (contributed by the mass of the motors) rather than in the structural linkages, a joint in the Simulation Model is treated as a “link” with physical parameters and a link COG that is located near the origin of the assigned frame. The mass and moment of inertia of the actual structural linkages (such as the thigh, lower leg and foot) are “lumped” with the mass and moment inertia of its nearest joint (Figure 2.7). Therefore, even though the biped has seven actual structural linkages, the motors associated with each joint also count as “links” and the total number of links in the model is thirteen. The link COG locations, frame locations and frame assignments is presented in Figure 2.6. The names and numbers of the joint frames and links that are used for

Table 2.3: The definable physical properties of a Servo node that include physical properties of the actual motor and link.

Fields (Tunable Parameters)	Description	Values
Type	Specify the actuator type (linear or revolute).	Revolute
Maximum Velocity [rads/s]	Set the maximum velocity of the motor, values are taken from the specifications of the the actual motors.	5 rad/s
Position [rads]	The current angular position of the motor.	Varies
Mass [kg]	Mass of the Servo node (can include both the mass of the motor and the link that is attached to it).	Varies
Inertia Matrix	The elements of the inertia tensor of the Servo node (can include both the mass of the motor and the link that is attached to it).	Varies
Center of Mass location (cm)	The location of the COG position of the link with respect to the parent Servo node.	Varies
Orientation	The orientation of the Servo frame with respect its Parent frame (frame of the previous joint), given in axis-angle representation.	Varies

the remainder of this thesis is summarized in Table 2.4 and Table 2.5, respectively. The combined mass of the links and motors (assumed to be concentrated at the link COG positions defined under each frame) are also presented in Table 2.5. Aside from the frames that are associated with the actuated DOFs, three additional frames are defined in the model:

1. *World (Inertial) Frame*— A reference frame with a fixed origin on the ground and a constant position and orientation that is independent in time.
2. *Local Base Frame*— A frame that is attached to the pelvis of the biped. It can

translate or rotate with respect to the World Frame but does not translate or rotate with respect to the pelvis (Local Base link).

3. *Fixed Frame*— A frame that is attached to the *Stance Foot*. The stance foot is the foot that is in contact with the ground. Assuming that there is sufficient friction force to keep the foot from slipping, the fixed frame does not translate or rotate with respect to the world frame (refer to Chapter 3 for a detailed discussion).

Table 2.4: The name and number of the frames in the Simulation Model.

Frame Name	Number
Pelvis Frame (pelvis)	0
Hip Roll Left(HRL)	1
Hip Yaw Left (HYL)	2
Hip Pitch Left (HPL)	3
Knee Pitch Left (KPL)	4
Foot Pitch Left (FPL)	5
Foot Roll Left (FRL)	6
Hip Roll Right (HRR)	7
Hip Yaw Right (HYR)	8
Hip Pitch Right (HPR)	9
Knee Pitch Right (KPR)	10
Foot Pitch Right (FPR)	11
Foot Roll Right (FRR)	12
Torso Roll (TorsoRoll)	13

Forward kinematics is a problem that involves computing the relative position and orientation between two frames. In the Simulation Model, a frame is either attached to a joint (joint frames) or to a link COG/overall COG of the biped (Chapter 3). Two

Table 2.5: The name, number, and mass of the links in the Simulation Model.

Link COG Name	Number	Mass [kg]
Pelvis	0	3.73
Hip Roll Left Motor (HRL_COG)	1	0.933
Hip Yaw Left Motor (HYL_COG)	2	1.44
Hip Pitch Left Motor and Left Thigh (Thigh_L)	3	2.00
Knee Pitch Left Motor and Left Shank (Lowerleg_L)	4	1.28
Foot Pitch Left Motor (FPL_COG)	5	0.133
Foot Roll Left Motor and Left Foot (Foot_L)	6	0.765
Hip Roll Right Motor (HRR_COG)	7	0.933
Hip Yaw Right Motor (HYR_COG)	8	1.44
Hip Pitch Right Motor and Right Thigh (Thigh_R)	9	2.00
Knee Pitch Right Motor and Right Shank (Lowerleg_R)	10	1.28
Foot Pitch Right Motor (FPR_COG)	11	0.133
Foot Roll Right Motor and Right Foot (Foot_R)	12	0.765
Torso and the Torso Roll Motor (Torso)	13	5.12
Total Mass	-	21.95

important *constant parameters* from the Simulation Model (Figure 2.6) are needed for this computation: 1) The relative position between the origin of two successive joint frames (summarized in Table A.1), and 2) The position of link COG i with respect to its nearest frame i (summarized in Table A.2). These parameters are obtained from the Pro-Engineer CAD model.

Conventions, Notations and Units in the Simulation

This section establishes consistent standards of units, notation, and conventions that are used in the ABL-BI Simulation Model.

1. *Orientation and Position Description*– Standard approaches in manipulator

kinematics can be applied to the kinematic description of ABL-BI. This involves attaching frames to different parts of the biped (such as link COG and joints) and describing the relative positions and orientation between these frames. Adopting the same conventions as [31], the position of a point in space or an origin of a frame is given by a 3×1 vector ${}^A P$, where the post-superscript indicates that point P is described with respect to frame A. The orientation of frame i with respect to frame $i - 1$ is represented by a 3×3 rotation matrix ${}^{i-1}{}_i R$. A common approach in describing the relative position and orientation of two frames is to use the Denavit-Hartenburg (DH) parameters [32]. However, this notation requires the frame axes to be defined in a specific way (the Z-axis of the frame is always defined to be coincident with the joint axis). *Webots* on the other hand uses Cartesian X-Y-Z coordinate systems. Therefore, it is more convenient to describe the orientation of a frame using Y-X-Z (yaw-pitch-roll) fixed angles representation. The resulting 3×3 rotation matrix of frame i with respect to a fixed frame $i - 1$ is as follows:

$${}^{i-1}{}_i R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} = \begin{bmatrix} c\gamma c\alpha - s\gamma s\beta s\alpha & -s\gamma c\beta & c\gamma s\alpha + s\gamma s\beta c\alpha \\ s\gamma c\alpha + c\gamma s\beta s\alpha & c\gamma c\beta & s\gamma s\alpha - c\gamma c\alpha s\beta \\ -c\beta s\alpha & s\beta & c\beta c\alpha \end{bmatrix}, \quad (2.1)$$

where γ , β , and α are the roll (about Z axis), pitch (about X axis), and yaw (about Y axis) angles respectively, cX is $\cos(X)$ and sX is $\sin(X)$.

The combination of orientation and position of frame i with respect to frame $i - 1$ is written in a 4×4 transformation matrix denoted by the notation ${}^{i-1}{}_i T$.

2. *Coordinate Axes and Planes*– The coordinate axes of the frames in the Simulation Model (ex: joint frames, World Frame) are defined in Figure 2.6. A Cartesian X,Y,Z coordinate system is used. The plane spanned by the Z and Y axis of the Local Base Frame is referred to as the *sagittal* plane, the plane spanned by the X and Y axis is the *medio-lateral* plane, and the plane spanned by the X and Z axis is the *transverse* plane.
3. *Units*– The units for angular positions and its higher derivatives is in radians (rads), the units for force is Newtons (*N*), torque and moment is in Newton meters (*N.m*), and time is in milliseconds (ms). Two different units are used for linear positions and its higher derivatives. In the *Webots* world editor, the Simulation Model is defined in meters (*m*); in the robot controller and the output plots of the simulation results, centimeters is used (*cm*).

2.2.3 Sensory Information in Webots

Since the proposed controllers rely on state feedback to generate and control biped motions, the controller needs access to the physical states of the biped.

These states can either be measured directly from the biped or calculated from sensor information. The physical states are categorized into two groups:

1. *Accessible States*– This refers to physical states that are supplied by onboard sensors and are *observable* in real life. In *Webots*, there are virtual sensors called robot devices that simulate the behaviors and outputs of actual sensors. The robot controller can make the appropriate API function calls to obtain the accessible states from the virtual sensors. Through the use of virtual sensors, the simulated biped only have access to states that are available in real life. As the result, the robot controller is designed within the limits of accessible sensor

information. The joint positions and torques are obtained from the *Servo node*, which functions as an actuator, encoder and a torque sensor. In addition, the position and orientation of an arbitrary link on the biped (with respect to the world frame) can be obtained from a *GPS sensor* and an *INS* sensor respectively. However, these states are technically not observable in real-life since it is not feasible to attach GPS/INS sensors to every link in the actual biped. Finally, the vertical component of the GRF can be measured by *touch sensors*. A summary on the accessible states provided by the robot devices in *Webots* is presented in Table 2.6.

2. *Inaccessible States*— These are physical states that are *unobservable* in real life but can be calculated from sensor information. In *Webots*, the robot controller can directly obtain inaccessible states in two ways: 1) Through a special controller called the *Supervisor* (for more details, refer to the Section 2.2.4). The Supervisor uses special API functions that allow direct access to the kinematic and dynamic properties of the biped, and 2) Implement and run a shared library called the *Physics Plugin* at runtime (for more details, refer to the Section 2.2.4). The Physics Plugin allow the user to use low level ODE functions to directly obtain dynamic properties of the biped and customize the physics in the simulation (ex: change the ground model and apply forces or torques directly to the biped). The kinematic and dynamics properties of the biped can be obtained entirely through the Physics Plugin and the Supervisory Controller. However, if the robot controller relies too heavily on inaccessible states and ignore real-life limitations, the controller might need to undergo major modifications before it can be used on the actual platform. For this reason, only a few selected inaccessible states , namely the link COG position, the foot contact points, the GRF (X,Y, and Z direction), and the pelvis and foot orientation (given in axis-angle

representation), are obtained from the Supervisory Controller/Physics Plugin in the simulated biped (only in the *Webots* 6.24 Professional Version). In the actual biped, the inaccessible states can be calculated (Chapter 4 and 5) from the forward kinematics of the biped using the current joint positions. The inaccessible states are summarized in Table 2.6.

Table 2.6: Summary of accessible and inaccessible states from *Webots*.

States	Type
Servo Position	Accessible
Servo Torque	Accessible
Position of a point (wrt to the world frame)	Accessible to Webots 5.92
Orientation of a Frame (wrt to the world frame)	Accessible to Webots 5.92
Contact Force (horizontal X,Z)	Inaccessible
Contact Force (vertical Y)	Accessible
Contact Points	Inaccessible

2.2.4 The Control Levels in Webots

This section focuses on the control architecture of the *Webots* simulation program and present a high-level overview on the roles and dataflow of each of its main components: the Robot Controller, the Supervisory Controller, and the Physics Plugin. The discussion on the theoretical background and control algorithm of the robot controller, in particular, will be further expanded in Chapters 3, 4, and 5.

The Robot Controller

The robot controller is the main controller that generates stable walking gait and use state feedback to cope with external disturbance in real-time. The input to the controller is high level walking task goals such as a reference COG trajectory

for the ZMP-based controller and a reference CMP trajectory for the ZRAM-based controller. The outputs of the robot controller are actuator commands for the 13 actuated DOFs of ABL-BI. Note that based on the state feedback, the controller can modify the reference trajectories of the ZMP-based controller and the ZRAM-based controller based on the stability criteria outlined in Chapter 1. For more details on the ZMP-based controller and the ZRAM-based controller, please refer to Chapter 4 and 5.

The Supervisory Controller

The Supervisory Controller is a special high-level controller with extended capabilities. In the ABL-BI simulation, the Supervisory Controller is used to directly obtain the orientation of the pelvis frame and foot frame with respect to the world frame. The frame orientation are specified by the equivalent angle-axis representation in the Supervisory Controller. Note that the orientation information from the Supervisory Controller is converted to Euler Angles before they are sent to the robot controller at every simulation step.

The Physics Plugin

The Physics Plugin is a dynamics library that contains user defined functions that are called by *Webots* during the simulation. For the proposed controller, the Physics Plugin is used to apply external disturbance forces on the torso of the biped and obtain the ground contact points and GRFs directly using ODE functions. In the ABL-BI simulation, the Physics Plugin detects the ground contact points position and the GRFs, calculate the COP with respect to the world frame, and send the GRF and COP information to the robot controller at every time step. Below is a summary of the Physics Plugin implementation:

1. *Initialization*— The Initialization function is called at the beginning of the simulation. This function creates pointers to the world, geometric shapes and rigid bodies with dynamic properties (all ODE objects) that will be used by the functions Physics Plugin later in the simulation. In the ABL-BI simulation, ODE variables that are associated with the torso, ground and foot contact objects are initialized.
2. *Collision Detection*— The Collision Detection function is called every time there is an intersection between a pair of geometric shapes. In our case, we are interested in defining the contact behaviour between the foot and the ground. During collision, ODE objects called *dContact* are attached at the contact points. These ODE joint structures allow the user to tune different ground contact parameters. The two main parameters are the Error Reduction Parameter (ERP) and the Constraint Force Mixing (CFM) parameters. The ERF is defined as the magnitude of the constraint force that corrects joint errors and realigns the constrained position of two rigid bodies that are connected by a joint. The CFM determines the allowable error in the constraint of two rigid bodies. These parameters are used to compute the stiffness k_p and damping k_d terms of an equivalent spring-damper ground model:

$$k_p = ERP / (T_s \cdot CFM), \quad (2.2)$$

$$k_d = (1 - ERP) / CFM, \quad (2.3)$$

where T_s is the time step of the simulation in seconds (set to 0.001 s). Two sets of ground model gains were used for the ZMP-based controller and the ZRAM-based controller. For the ZMP-based controller, the default values ($k_p =$

2.5×10^8 and $kd = 8.0 \times 10^4$) were used. For the ZRAM-based controller, the gains were based on the ground model in [33] and were set to $k_p=30000$ and $kd=2000$. Using Equation 2.2 and 2.3, the parameters ERP and CFM are calculated.

3. *Customized Physics*— The Customized Physics function is called at every simulation step. It is used to apply an external force to the torso (as a disturbance) at a predefined time during the simulation.

In the robot simulation for the ZRAM-based controller, the Physics Plugin is used to detect the contact force and points at each time step. This information is then used to calculate the location of the the COP with respect to the world frame. The communication between the Physic Plugin, the Supervisory Controller and the robot controller is modelled as radio transmissions in *Webots* (the “radio signals” are assumed to have infinite penetration and range). By using robot devices called *Emitters* and *Receivers*, the inaccessible states obtained from the Supervisory Controller and the Physics Plugin are transferred to the robot controller via these communication endpoints.

The dataflow of the *Webots* simulation program is summarized in Figure 2.8. Similar to the actual biped, the robot controller generates stable walking gaits by using physical states that are measured from robot devices in the simulated biped. However, in order to simplify the robot controller, physical states that are not observable in real-life can also be obtained directly from special API functions in the Supervisory Controller (hence the “key” symbol in Figure 2.8) or ODE functions from the Physics Plugin. The Physics Plugin also has an additional functionality of generating customized physics for the simulated world in *Webots*.

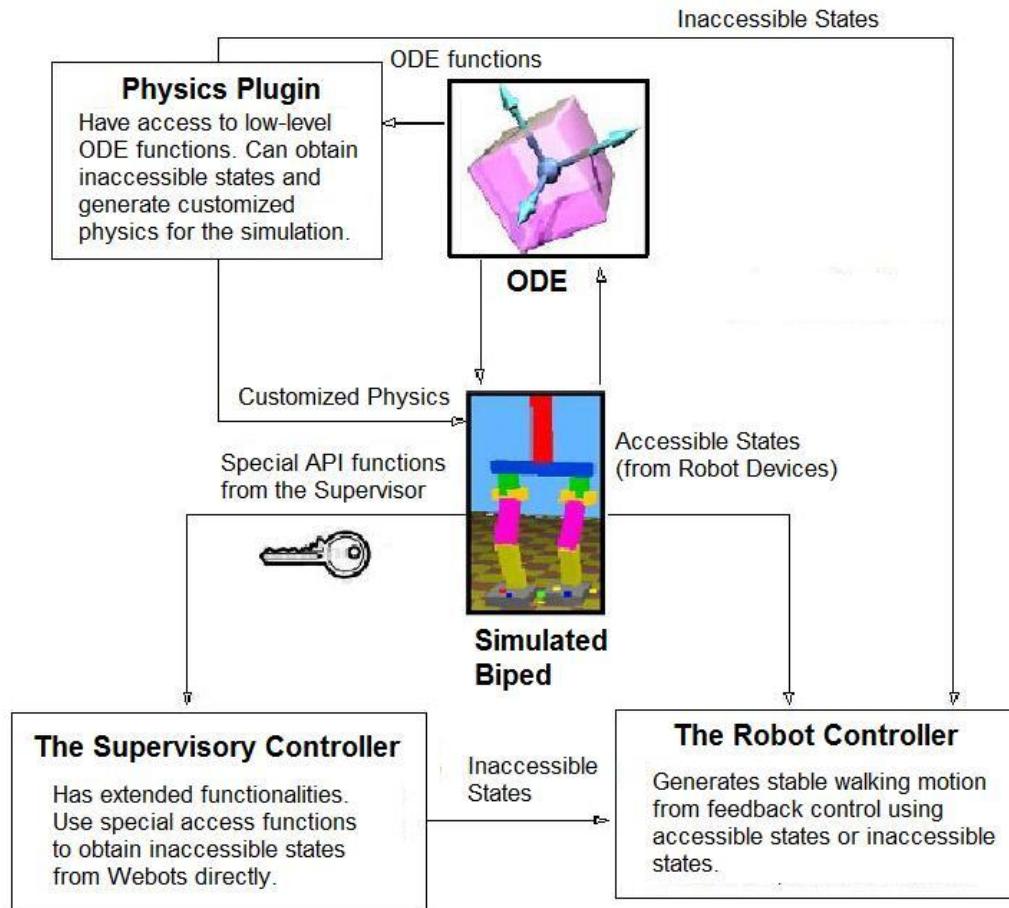


Figure 2.8: A schematic diagram of the control architecture of the ABL-BI simulation.

2.2.5 Summary

In this chapter, we presented the design of the ABL-BI platform, and the key features of the ABL-BI Simulation Model and the simulation environment *Webots*. The Simulation Model is composed of links that are connected and actuated by joints. It has 13 actuated DOFs and 12 uncontrolled DOFs. The links of the model are rigid bodies with kinematic and dynamic properties that are obtained from the 3-D CAD model of the ABL-BI platform. The kinematic description of the Simulation Model is facilitated by frames that are attached to the joints. The relative position and orientation between a given pair of links can be equivalently described by the relative position and orientation of their corresponding frames. There are three control levels in the simulation. The robot controller generates stable walking trajectories and uses state feedback from simulated sensors (robot devices) in *Webots*. The Supervisory Controller has privileged access to inaccessible states and the Physics Plugin is used to customized the ground model and apply external forces on the biped.

Chapter 3

The Essential Elements of the Robot Controllers

This chapter focuses on the technical issues and solution approaches that are found in both the ZMP-based controller and the ZRAM-based controller. Due to the varying kinematic structure of the biped, it is useful to divide the walking cycle into different locomotion phases in order to reduce the complexity of motion planning and control. In addition, in order to facilitate kinematic control of the biped, the full kinematics is treated as several open kinematic chains. The high level motions of the biped are then assigned to each kinematic chain as task space objectives. Applying local optimization techniques and redundancy resolution methods that are used in redundant manipulators, the task space objectives of each kinematic chain are reduced to joint velocities. In order to ease computational requirements and enable realtime control, the complex dynamics of the biped is approximated by simplified models. This Chapter is organized as follows: Section 3.1 provides a brief overview on the high level objectives of the biped and the characteristics of bipedal kinematics. In Section 3.2, the bipedal walking cycle is divided into locomotion states and spatial and workspace constraints are specified for each state. Section 3.3 describes how the kinematics of the biped is divided into multiple open kinematic chains and

presents a summary of the high level task assignment for each chain. Sections 3.4 and 3.5 introduce the forward and inverse kinematics problems of the kinematic chains and present a *Closed Loop Kinematic Controller* that is used in ABL-BI. Section 3.6 formulates the dynamics of the simplifying models and discusses their applications in robot controllers. Finally, this chapter is concluded by a top-level overview of the biped walking controllers in Section 3.7.

3.1 The High Level Tasks of Bipedal Walking

In order to achieve stable walking, the biped must meet multiple high level goals under different kinematic and dynamic constraints that are imposed on its motions. There are two types of constraints: external and internal. External constraints are defined by high level goals such as moving the biped to a target location in a set period of time or preserving the balance of the biped while it is performing a task. Internal constraints refer to the physical limitations of the robot. For example, the biped falls under a large disturbance as it may not be able to perform the required compensatory motions (such as providing a large acceleration to its torso or limbs) due to actuator limitations. Internal constraints must be taken into account during high-level motion planning (Chapter 4). For example, the maximum angular velocity of the motors, and the moment of inertia and dimension of the links place an upper-bound on the step size and step frequency of the biped. Therefore, the swing leg trajectory (high level goal) should be planned within a reasonable range. For stable walking, ABL-BI needs to achieve the following high level goals simultaneously:

- *COG/CMP tracking*— The overall motion of the biped is expressed by the COG (for the ZMP-based controller) and by the CMP (for the ZRAM-based controller) with respect to the world frame. In order to move the biped to a desired position, the robot controller must plan and track a COG or CMP trajectory

from its initial position to the desired position. To maintain static stability, the COG must be planned such that its initial and final position (when the biped is standing still) lies within the support polygon. In the ZMP-based controller, the COG has an additional function of maintaining dynamic balance by indirectly manipulating the ZMP. For dynamic stability, the COG is allowed to move outside the support polygon while the ZMP must remain within the support polygon. In the ZRAM-based controller, the reference CMP trajectory is planned within the support polygon to ensure dynamic stability. Note that CMP tracking can be achieved by tracking a reference COG trajectory that is generated from the reference CMP trajectory. In summary, the COG and the CMP define two high level objectives: 1) track a user-defined motion, and 2) maintain static/dynamic balance while the biped is walking.

- *Swing foot position and orientation tracking*– To facilitate locomotion, the swing foot must track a reference position trajectory to swing forward in order to take a step. The reference position trajectory is generated based on walking parameters such as the step period, the step length, and the clearance length (maximum height of the foot in the swing motion). Aside from position, the orientation of the swing foot is also important. At lift-off (the start of the swing phase), the foot must remain at a certain orientation in order to avoid interference between the foot and the ground. Similarly, during touch-down (end of the swing phase), the landing foot must have the same orientation as the ground surface so that it can regain sufficient ground contact upon landing.
- *Pelvis orientation tracking*– In bipedal walking, it is desirable for the pelvis to remain at an upright orientation to 1) maintain an anthropomorphic posture, and 2) maximize the workspace for the swing leg (see Figure 3.1). Therefore, one of the objectives in normal walking is to tracking a zero reference pelvis

orientation with respect to the ground.

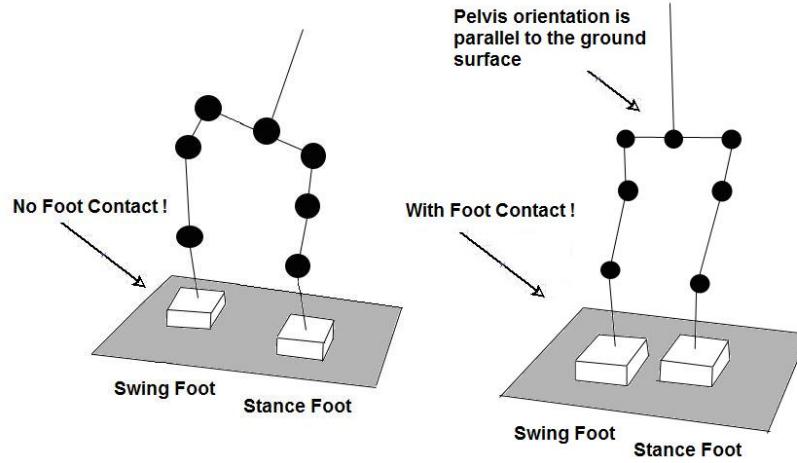


Figure 3.1: In the left figure, the pelvis has a large roll angle with respect to the ground. As the result, the workspace of the swing leg is reduced and the swing foot cannot reach the ground even if it is fully extended. This problem can be avoided if the pelvis remain flat with respect to the ground in the right figure.

To handle multiple high level tasks in stable walking, the biped can be treated as a redundant articulated structure with multiple end-effectors and a large number of joints. However, there are two important differences between manipulator kinematics and biped kinematics:

1. *Moveable Base*— Recall from Chapter 1 that a biped is an *underactuated* system with ground contacts that are modelled by passive 6-DOF joints. Unlike most robot manipulators, the biped does not have a fixed base. This is due to the fact that the stance foot, under a large disturbance, can rotate or translate with respect to the ground. This can pose a problem when estimating the linear and angular velocities of the end-effectors (ex: the swing foot) with respect to the world frame using the stance foot as the base frame (Section 3.4). The stance foot frame and the world frame can be treated equivalently only if there are

no relative motions between the stance foot and the ground. Therefore, unless the relative motions is taken into account in the kinematic calculations, the stance foot is assumed to be a fixed base. This assumption is valid only if the following conditions are met: 1) the COP (or equivalently, the ZMP) lies within the support polygon and the feet do not rotate about the horizontal axis, and 2) there is sufficient ground friction to keep the feet from slipping.

2. *Varying Structure*– The structure of the biped changes in a sequential manner during the course of a walking cycle. As a consequence, the biped kinematics vary between locomotion states. This problem can be simplified by defining different kinematic arrangements and control objectives for each locomotion state. The locomotion states will be discussed further in Section 3.2.

3.2 The Locomotion States

3.2.1 Locomotion States in a Walking Cycle

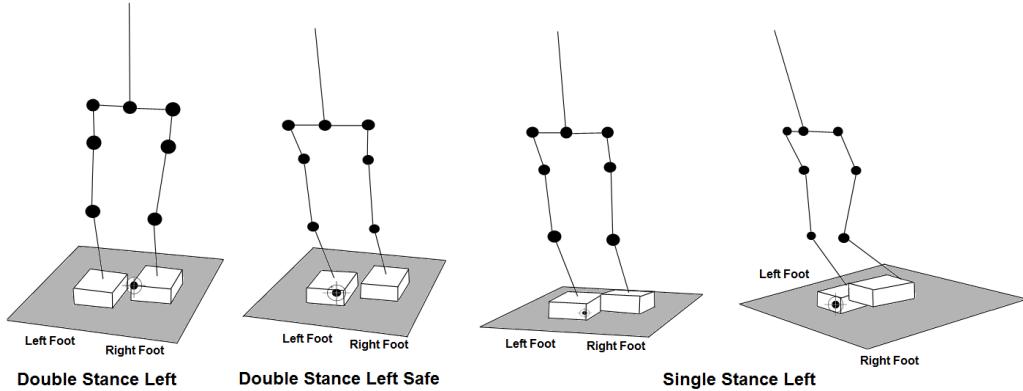
Bipedal walking is a periodic cycle that can be broken down into sequential walking phases. The requirements and tasks for each walking phase are defined by *locomotion states*. A summary of the locomotion states is presented below. In addition, graphical representations of the locomotion states are given in Figure 3.2.

- *Double stance left (DBL)*– The biped begins in the DBL state, with both feet in contact with the ground and the left foot in front of the right foot. During DBL, the COG/CMP position in the horizontal plane is manipulated by the joint angles to a target position that lies within the support surface of the left foot (weight shift).
- *Double stance left safe (DBL-Safe)*– Once the COG/CMP is within the support

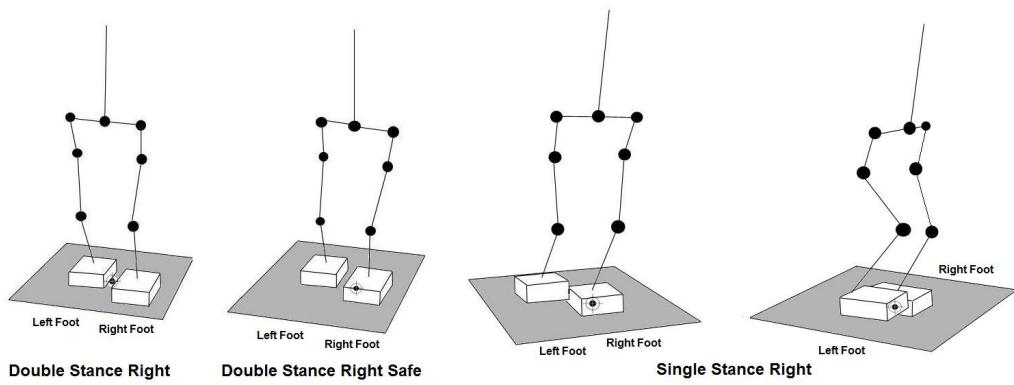
surface of the left foot, the biped is in the DBL_Safe state and it can safety lift its right foot (swing foot) off the ground (lift-off).

- *Single Stance left (SSL)*— The biped enters SSL after the right foot lifts off. In SSL, the left leg (stance leg) supports the biped and manipulates the ZMP/CMP to maintain balance while the right leg (swing leg) swings forward to take a step. Note that in single stance, the COG of the biped does not necessarily need to remain within the support polygon; as long as the ZMP is within the support polygon, the robot is dynamically stable.
- *Double stance right (DBR)*— The biped enters DBR after the right foot lands and regains contact with the ground (touch down). In DBR, both feet are in contact with the ground with the right foot in front of the left foot. During DBR, the biped performs weight shift to the support surface of the right foot.
- *Double stance right safe (DBR_Safe)*— Once the COG/CMP is within the support surface of the right foot, the robot is in the DBR_Safe state and it can safety lift its left foot (swing foot) off the ground.
- *Single stance right (SSR)*— The biped enters SSR after the left foot lifts off. In SSR, the right leg (stance leg) supports the biped and manipulates the ZMP/CMP to maintain balance while the left leg (swing leg) swings forward to take a step.
- *Flight (F)*— The biped is in *Flight* if there are no foot contacts. Typically in stable walking, the biped will not voluntarily enter into the Flight phase unless it has fallen. However, the flight phase is part of the stable locomotion cycle of other types of locomotion such as running and jumping.

The locomotion states provide a qualitative description for different parts of the walking cycle. As shown in Figure 3.2, each locomotion state is defined by ground



(a) The biped in DBL, DBL-Safe, and SSL.



(b) The biped in DBR, DBR-Safe, and SSR.

Figure 3.2: The Locomotion states in the walking cycle, the COG/CMP is represented by a point on the support surfaces of the feet.

contacts of the feet, the relative position between the left and right foot, and the position of the COG/CMP relative to the support surface of the robot. Notice that biped has to meet certain conditions (ex: swing foot lift-off) before it can proceed to the next locomotion state. Transitions between locomotion states will be discussed further in Section 3.2.3. In addition, the locomotion states also specify high level goals and assign different roles to the legs of the biped. A detailed discussion of this will be presented in Section 3.2.2.

3.2.2 Intermediate Objectives and Workspace Constraints

The locomotion states are used to define foot contact requirements and high level goals for different parts of the walking cycle. Each locomotion state specifies goals for state variables such as the swing foot position and the COG/CMP position. These goals can be viewed as “check-points” called *Intermediate Objectives* that the robot states must attain before reaching the *final* high-level goal. The intermediate objectives must be completed within an allotted period of time and the robot states must remain in operating regions that are defined by the *workspace constraints*. Let’s clarify this with an example. Consider the case where we command the biped to move from point A to point B. The biped cannot directly move its COG along an arbitrary path to reach point B; instead, it must alternate its legs between two discrete roles (swing and support) and propel itself by moving its swing leg to specific positions while being supported by the stance leg. Due to the discrete foot contact conditions of the feet, the COG/CMP trajectory must pass through a series of “via points” on the foot support surfaces before it can reach point B in a stable manner. The intermediate objectives are expressed as external spatial constraints that are defined for each locomotion state. The spatial constraints are imposed on the following state variables in the above example: 1) the COG/CMP position, and 2) The position and orientation of the stance foot and swing foot.

Aside from spatial constraints, the locomotion states also specify time constraints which require the biped to complete the intermediate objectives within an acceptable period of time and make the transition to the next locomotion state. For example, the time period for the double stance and single stance state in the ZRAM-based controller is around 5500 *ms* (5.5 *s*). Note that these time constraints are based on the target walking velocity specified by the user and the internal constraints of the biped. Finally, workspace constraints are imposed on the COG, ZMP, CMP and the stance foot and swing foot positions. To maintain balance, the ZMP/CMP trajectory is restricted within the support polygon of the biped throughout the walking cycle. This is especially important during single stance where the support polygon is limited to the support surface of the stance foot. Inappropriate motions in this situation can cause the biped to fall even in the absence of external disturbance. In addition, workspace constraints enforce the stance foot to be fixed to the ground and the swing foot to remain within its kinematic workspace.

The intermediate objectives and workspace constraints of each locomotion state are summarized below. A graphical representation of the walking cycle with the intermediate objectives is also presented in Figure 3.3 .

In **double stance (DB)**, there are two intermediate objectives:

1. The robot must shift its COG/CMP to the support surface of the forward stance foot before the back stance foot can lift off in a stable manner. Therefore, the intermediate objective for the COG/CMP is a target position that lies within the forward foot.
2. The feet of the robot must remain stationary and flat on the ground.

The workspace constraints for **DB** are:

1. The allowed region for the COG/CMP trajectory is defined by the support polygon spanned by the left and right foot contact points.
2. The feet must remain fixed throughout double stance; therefore, the workspace of the stance feet is constrained to their initial positions and orientations at the start of the *double stance* state.

In **double stance safe (DB-Safe)**, there are two intermediate objectives:

1. The COG/CMP must remain within the support surface of the stance foot to maintain balance. Therefore, the intermediate goal for the COG/CMP is a target position that lies well within the boundary of the stance foot.
2. The stance foot of the robot must remain stationary and flat on the ground while the swing foot lifts off. The intermediate objective for the position of the swing foot is a target vertical (Y) position that results in a safe clearance between the swing foot and the ground. In order to avoid interference with the ground, the orientation of the swing foot is constrained to be parallel with the ground surface.

The workspace constraints for **DB-Safe** are:

1. The workspace for the COG/CMP trajectory is the support surface of the stance foot.

2. The workspace for the stance foot is its initial position and orientation at the start of the *double stance safe* state. The workspace of the swing foot position and orientation is the reachable workspace of the swing leg.

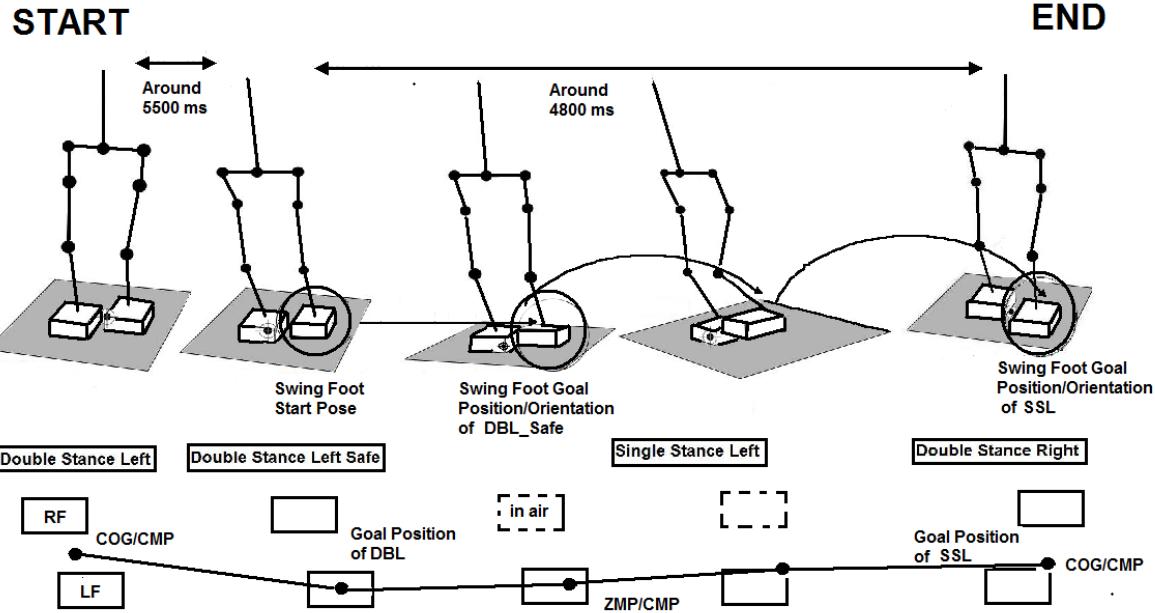
In **single stance (SS)**, there are two intermediate objectives:

1. For the ZMP-based controller, the target COG position can lie outside of the support polygon as long as the ZMP position remain within the support polygon. However, due to possible external disturbances, the target COG position is usually placed near the edge of the stance foot support surface to maintain a high margin of stability (Figure 3.3). For the ZRAM-based controller, the intermediate objective for the CMP is a target position that lies within the support surface of the stance foot (usually near the edge of the stance foot, towards the direction of the foot placement of the swing foot) (Figure 3.3).
2. The intermediate objective of the swing foot is to land at a certain distance (defined by the step length parameter) ahead of the stance foot. The objective of the stance foot is to remain stationary and flat with respect to ground.

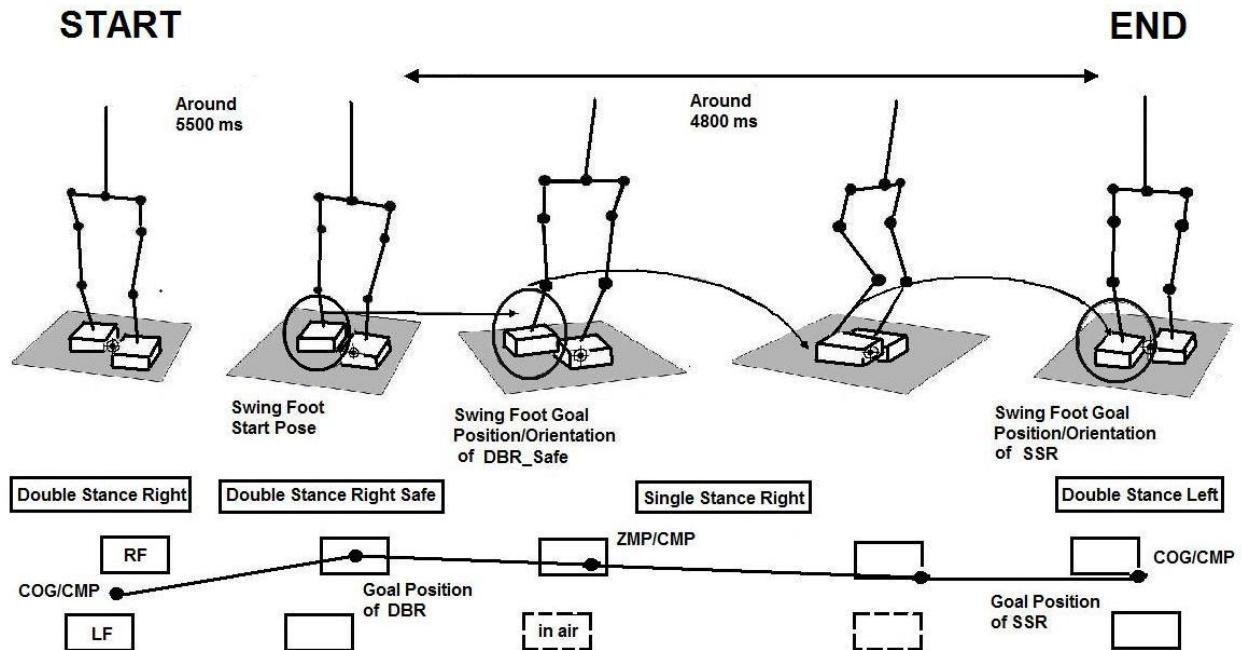
The workspace constraints of the **SS** state is the same as the *double stance safe* state. The only exception is that the workspace of the COG (in the ZMP-based controller) can extend beyond the support surface of the stance foot.

3.2.3 State Transitions

State transitions from one locomotion state to another is triggered by the completion of the intermediate objectives of the current locomotion state. For example, one of the intermediate objectives in the SS state is that the swing foot must attain a firm



(a) The intermediate objectives and time constraints for the locomotion states: DBL, DBL_Safe, and SSL.



(b) The intermediate objectives and time constraints for the locomotion states: DBR, DBR_Safe, and SSR.

Figure 3.3: The intermediate objectives of the locomotion states in the walking cycle.

ground contact. This occurs when the swing foot is at a certain vertical position with respect to the ground and the foot has the same orientation as the ground surface. Note that unless this objective is completed, the biped cannot proceed to the DB state. Therefore, the intermediate objectives of the current state is also the prerequisite for the next. For state transitions, the external constraints on the foot position and orientation are more stringent than the ZMP/CMP positions. Recall that the biped can maintain dynamic balance as long as the ZMP/CMP is within the support polygon. Ideally, the ZMP/CMP should reach its target value at the end of a locomotion state. However, as long as the ZMP/CMP position is within an **acceptable region** around the target position, it is feasible for the biped to proceed to the next state without falling. The same cannot be said about the foot position and orientations due to the discrete nature of the foot contact condition. Figure 3.4 summarizes the main locomotion states and state transitions of ABL-BI. In addition, a summary of the conditions for state transitions is provided in Table 3.1.

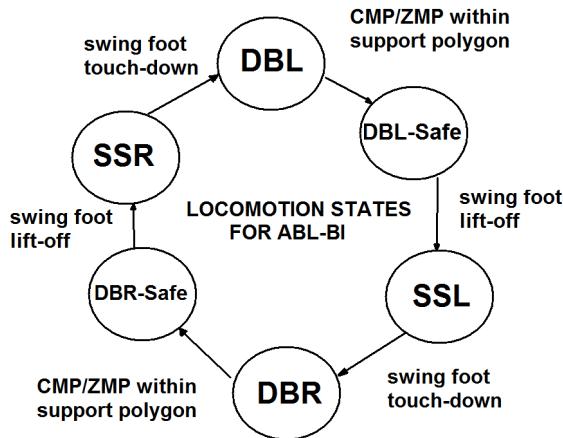


Figure 3.4: The locomotion state and state transitions of ABL-BI.

In summary, the intermediate objectives are defined for each locomotion state to guide the biped to its final high-level goal in a safe and feasible manner. The spatial

Table 3.1: State transition requirements.

State Transitions	Conditions
DB to DBS	The COG/CMP is within the support surface of the stance foot. The error between the actual and reference COG/CMP is within 1 cm.
DBS to SS	Clearance between the swing foot and the ground. The COG/CMP is within the support surface of the stance foot.
SS to DB	The swing foot touches down and regains ground contact. The COG/CMP is within the support surface of the stance foot.

and time constraints defined in this section will be used to generate state trajectories using the cubic spline interpolation method [8]. Besides intermediate objectives, the kinematic structure of the biped is also dependent on the locomotion state. The biped model can be divided into a number of open kinematic chains whose role and behaviour interchange throughout the walking cycle. The definition and application of these kinematic chains will be discussed in Section 3.3.

3.3 Biped Kinematic Arrangements

To start, let's consider the two main problems in manipulator kinematics. The first is to find the position and orientation of the end-effector as a function of the joint variables. This problem, called forward kinematics, is formulated to calculate the *inaccessible states* of the biped. The second problem, called inverse kinematics, involves computing the joint variables from a given end-effector position and orientation. In previous bipedal walking controllers [12, 34], the biped is usually regarded as a single open kinematic chain in SS and as a parallel manipulator in DB. Here, we will propose a new and intuitive method in dealing with the full kinematics of the biped in order to simplify motion planning and postural control.

In dealing with the forward and inverse kinematics of the biped, it is useful to separate the full kinematics into several open kinematic chains and apply control and

analytical methods that are used for robot manipulators. The overall structure of the biped can be divided into different segments and each segment has a specific role: the pelvis connects the upper body to the lower limbs and provide structural support for the torso, the legs enable bipedal locomotion, and the torso carries important components of the robot and can be used to manipulate the COG. The segments of the robot are modelled as a series of joints and links called *kinematic chains*. Like the robot manipulator, a kinematic chain has a base frame that is assumed to be fixed with respect to the world frame, and an end-effector frame that is attached to the end of the chain (Figure 3.5). The end-effector frames are associated with the high level tasks of the robot that were outlined in Section 3.1. The remainder of this section is organized as follows, the concept of task delegation and joint space constraints is presented in Section 3.3.1, and the formal definition of the kinematic chains is presented in Section 3.3.2.

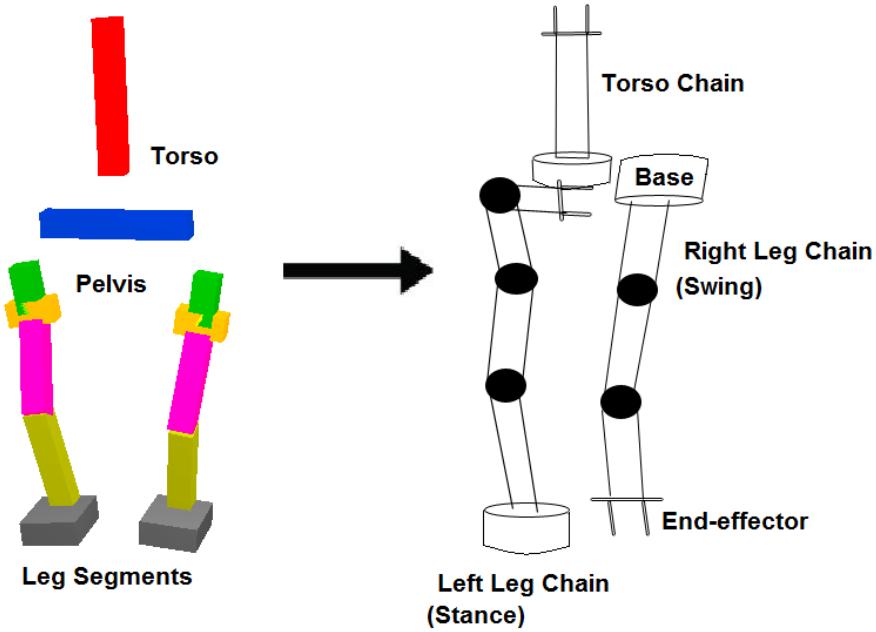


Figure 3.5: The segments of the biped are modelled as separate kinematic chains. Each chain has a base frame and an end-effector frame.

3.3.1 Kinematic Task Formulation

Problem Statement

The inverse kinematics problem involves finding a joint configuration of the biped (expressed as a joint angle vector $\vec{\theta}$), such that the resulting end-effectors position, given by the task space vector \vec{X} matches the target position \vec{X}_d . The joint angle vector is related to the end-effector position by the forward kinematics equation:

$$\vec{X} = \vec{f}(\theta), \quad (3.1)$$

where $\vec{f}(\cdot)$ is a set of nonlinear equations that relates the joint angles to the task space variable. Due to the complex kinematics of the biped, there are generally no exact close form solutions to Equation 3.1. However, an approximate solution can be obtained by first finding the linear approximation of $\vec{f}(\cdot)$. This is done by differentiating Equation 3.1 with respect to time:

$$\vec{\dot{X}} = J(\theta) \cdot \vec{\dot{\theta}}, \quad (3.2)$$

where $\vec{\dot{X}}$ is the task velocity vector, $\vec{\dot{\theta}}$ is the joint velocity vector, and J is the *Jacobian matrix* that defines the linear mapping between the joint velocity and the task velocity. The Jacobian matrix is a function of the joint angles and it can be obtained from the partial differentiation of the kinematic equation $\vec{f}(\cdot)$ with respect to the joint angles $\vec{\theta}$ (Equation 3.3).

$$J = \begin{bmatrix} \frac{\partial X_1}{\partial \theta_1} & \dots & \frac{\partial X_1}{\partial \theta_{13}} \\ \frac{\partial X_2}{\partial \theta_1} & \dots & \frac{\partial X_2}{\partial \theta_{13}} \\ \frac{\partial X_3}{\partial \theta_1} & \dots & \frac{\partial X_3}{\partial \theta_{13}} \end{bmatrix} \quad (3.3)$$

Assuming that the task velocity vector is 3×1 and the joint velocity vector is 13×1 (ie: the 3-DOF end-effector task is controlled by 13 joints), the Jacobian matrix will have the dimension of 3×13 . With the Jacobian matrix calculated, it can then be used to compute the proper joint velocities that change the end-effector position towards the target position. The end-effector \vec{X} has to move at a certain velocity $\vec{\dot{X}}_d$ at every time step until the error between the actual \vec{X} and the target \vec{X}_d becomes sufficiently small. The desired $\vec{\dot{X}}_d$ is in turn generated by $\vec{\theta}_d$ that is computed by the Jacobian matrix from Equation 3.2. Note that the Jacobain Matrix J is the linearization of the vector of nonlinear equations $\vec{f}(\cdot)$; therefore, $\vec{\theta}_d$ is a local solution that is evaluated at every instant of time about the neighbourhood of $\vec{\theta}$.

The solution to Equation 3.2 is usually calculated by premultiplying the task velocity vector by the least squares (LS) pseudoinverse of the Jacobian matrix J^\dagger (Equation 3.4).

$$\vec{\theta} = J^\dagger \cdot \vec{\dot{X}}, \quad (3.4)$$

The LS pseudoinverse solution $\vec{\theta}_d$ (Equation 3.5) satisfies two conditions in the **following order**: 1) The norm of the error between the desired task vector \vec{X}_d and the actual task vector \vec{X} is minimized, 2) The 2-norm of the joint velocities $\vec{\theta}$ is minimized.

$$\begin{aligned} & \min \| \vec{\theta} \|, \\ & \text{subjected to } \min \| \vec{X} - J \vec{\theta} \|, \end{aligned} \quad (3.5)$$

However, depending on how we model the kinematics of the biped, the solutions that

are based on this method might not be desirable for biped walking. Let's illustrate this problem with an example. In SSL, one of the intermediate objectives is to move the swing foot position to a target position with respect to the stance foot frame. As shown in the left image of Figure 3.6, the biped is modelled as a single open kinematic chain (*leg chain*) that includes all the joints of the biped (with the exception of the *TorsoRoll* joint). Here, the objective is to move the right foot along a planned trajectory; hence, since the torso motion has no effect on the swing foot position, it is excluded from the kinematics in this phase of the problem. The base of the kinematic chain is the stance foot frame (Frame 6) and the end-effector is the swing foot frame (Frame 12).

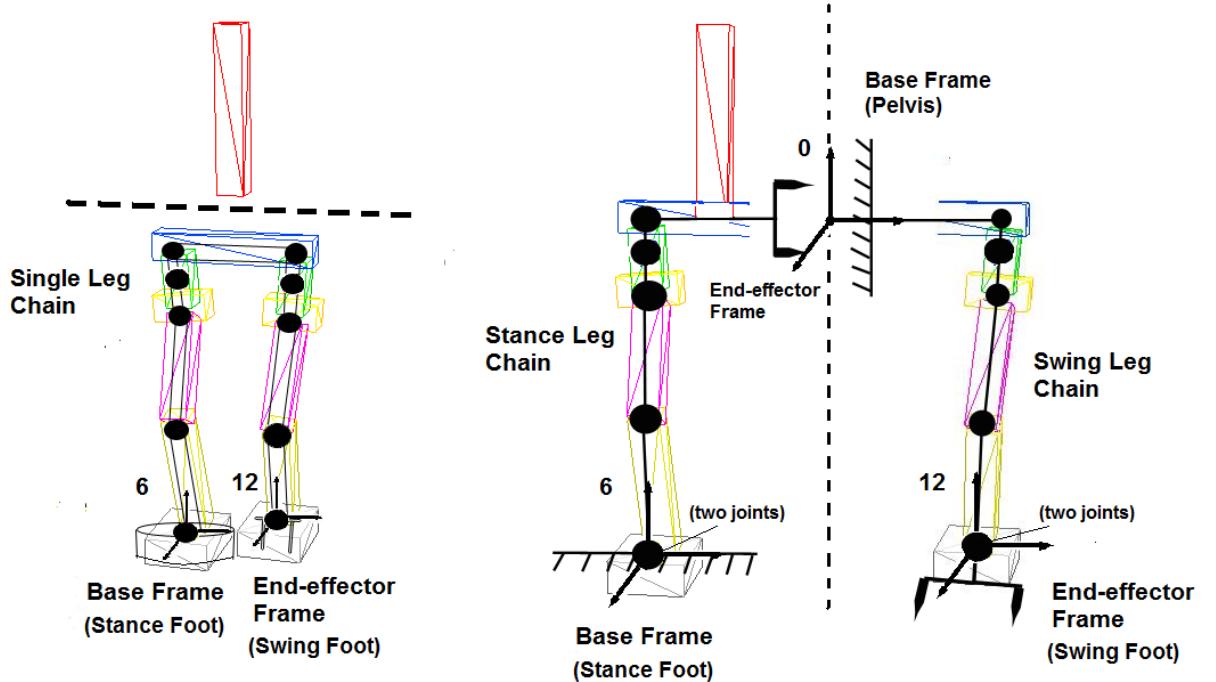


Figure 3.6: Two ways to model the biped in SSL. In the left figure, the biped is modelled as a single open kinematic chain. In the right figure, the biped is modelled as two separate chains, each of which has a base and an end-effector.

In order to track a desired swing foot position, a desired swing foot velocity ${}_{12}^6\dot{X}_d$ is usually calculated from a simple feedback controller that accounts for the error

between the end-effector position ${}_{12}^6X$ and its target position ${}_{12}^6X_d$ (more details will follow in Section 3.4). From ${}_{12}^6\dot{X}_d$, we can calculate the proper joint motions $\overrightarrow{\theta}_d$ from the LS pseudoinverse using Equation 3.4. However, due to the minimum joint velocities condition on the LS solution— $\min \|\overrightarrow{\theta}\|$, it is possible for the solution to be one where most of the joint motions take place in the joints further up in the stance leg chain. This results in a gait where the swing foot motion is largely contributed by the pitching of the stance leg and the swing leg motion that is found in normal walking gaits is absent (Figure 3.7). A more anthropomorphic solution with swing leg motion can be obtained by imposing constraints such as joint torque minimization (as done in [35]) or kinematic constraints that require the corresponding links (such as the thigh) of the two legs to be separated by a certain distance. However, due to the varying structure and complex kinematics of the biped, it is not feasible to select ad hoc constraints to produce swing leg motions.

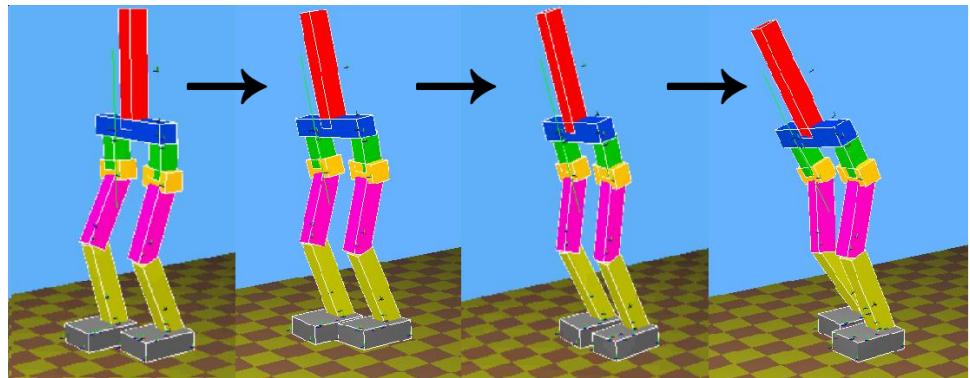


Figure 3.7: Swing foot motion produced using the single leg chain kinematic arrangement. Note that the configuration of the swing leg is fairly constant throughout the motion since most of the swing foot motion is contributed by the stance leg.

The Solution Approach

A more intuitive approach is to separate the *single leg chain* into two kinematic chains: the swing leg chain and the stance leg chain as illustrated in the right

image of Figure 3.6. Each kinematic chain is treated as a “robot manipulator” that has a base frame and an end-effector frame. The task assignment for each chain is as follows: the swing foot position and orientation is manipulated by the joints in the *swing foot chain*, which models the swing leg, and the biped COG position and pelvis orientation are manipulated by the joints in the *stance foot chain*, which the stance leg and the torso. Therefore, the joint angles of the robot will be calculated from separate Jacobian matrices. The main advantage of this kinematic setup is that we can specify motions and assign tasks to specific segments of the robot.

Let’s illustrate this approach by examining the forward and inverse kinematics of the stance leg and swing leg chain in the right figure of Figure 3.6. Note that we will only be dealing with the swing foot position in this example. The base of the swing leg chain is the Pelvis frame (Frame 0) and its end-effector is the swing foot frame (Frame 12). The position of Frame 12 with respect to the Frame 0 is denoted by the $4 \times 4 {}_{12}^0T$ (for more details, refer to Section 3.4). The position vector ${}_{12}^0P$ is then obtained from the first three elements of the last column of ${}_{12}^0T$ (Equation 3.6).

$${}_{12}^0T = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & {}_{12}^0P_X \\ R_{2,1} & R_{2,2} & R_{2,3} & {}_{12}^0P_Y \\ R_{3,1} & R_{3,2} & R_{3,3} & {}_{12}^0P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.6)$$

where $R_{i,j}$ is the element of the i th row and j th column of the 3×3 rotation matrix ${}_{12}^0R$. Note that the elements of ${}_{12}^0R$ and ${}_{12}^0P$ are functions of the swing leg joints $\vec{\theta}_{swing}$ (a 6×1 vector).

Unlike a robot manipulator, the base of the swing foot chain is a *floating base* that can translate and rotate with respect to the ground. In order to describe the motion of the swing foot, we need to express the swing foot position with respect to a non-moving reference frame. Assuming that the biped is dynamically balanced throughout SS (ie: the stance foot does not rotate with respect to the ground) and that there are sufficient ground friction to prevent the stance foot from translating, the stance foot frame (Frame 6) can be assumed to be fixed. The relative translation and rotation between the *moving base* of the swing leg chain (Frame 0) and the fixed frame (Frame 6) is described by ${}_0^6T$, and the swing foot position with respect to the fixed frame is given by:

$${}_{12}^6P = {}_0^6T {}_{12}^0P. \quad (3.7)$$

The motion of the floating base with respect to the fixed frame ${}_0^6T$ is a function of the stance leg joint angles θ_{stance} (a 7×1 vector, with an additional *TorsoRoll* joint that has no effect on the floating base motion). This follows from the fact that the base of the swing leg chain is also an end-effector of the stance leg chain (Figure 3.6). Since we are interested in describing the position of the swing foot with respect to the fixed frame (stance foot frame), the forward kinematics of the swing foot frame (Equation 3.7) can only be calculated by effectively treating the biped as a single kinematic *leg chain*. In other words, even though the kinematics of the biped is divided into two separate chains, the combined kinematic equations of the swing foot position with respect to the stance foot is equivalent to the kinematics of a single leg chain. Note that ${}_{12}^6P$ is a function of both $\vec{\theta}_{stance}$ and $\vec{\theta}_{swing}$. Taking the partial differentiation of ${}_{12}^6P$ with respect to the joint angles of both legs $\vec{\theta}_{leg}$ (a 13×1 vector), we obtain:

$${}_{12}^6\vec{X} = {}_{12}^6J(\theta_{leg}) \cdot \vec{\dot{\theta}}_{leg}, \quad (3.8)$$

where ${}_{12}^6J$ is a 3×13 matrix. The positions and velocities of the swing foot with respect to the stance foot ${}_{12}^6X$ and ${}_{12}^6\dot{X}$ (inaccessible states) can be calculated using Equation (3.7) and (3.8).

The kinematic chains and the task assignment approach is more relevant in the inverse kinematics of the swing foot position. The objective of this problem is to track a desired ${}_{12}^6\dot{X}$ using *only* the joint motions of the swing leg $\vec{\theta}_{swing}$. However, since the base of the swing leg chain (Frame 0) with respect to the stance foot (Frame 6) is dependent on $\vec{\theta}_{stance}$, the solution must also take into account the joint configuration of the stance leg. In other words, the desired swing foot position must be tracked using the swing leg joints despite the motions in the stance leg.

First, $\vec{\theta}_{leg}$ can be divided into $\vec{\theta}_{stance} \epsilon R^7$ and $\vec{\theta}_{swing} \epsilon R^6$:

$$\vec{\theta}_{leg} = \begin{bmatrix} \vec{\theta}_{stance} \\ \vec{\theta}_{swing} \end{bmatrix}. \quad (3.9)$$

Similarly, the elements of ${}_{12}^6J(\vec{\theta}_{leg})$ can be divided into the Jacobian of the stance leg $J_{stance}(\vec{\theta}_{leg})$ (a 3×7 matrix), and the Jacobian of the swing leg $J_{swing}(\vec{\theta}_{leg})$ (a 3×6 matrix):

$${}_{12}^6J(\vec{\theta}_{leg}) = \begin{bmatrix} J_{stance} & J_{swing} \end{bmatrix} \quad (3.10)$$

Substituting Equations 3.9 and 3.10 into Equation 3.8:

$${}_{12}^6\vec{X} = \begin{bmatrix} J_{stance} & J_{swing} \end{bmatrix} \cdot \begin{bmatrix} \vec{\theta}_{stance} \\ \vec{\theta}_{swing} \end{bmatrix}. \quad (3.11)$$

By decomposing the Jacobian matrix ${}_{12}^6J(\theta_{leg})$ and the joint velocity vector $\vec{\theta}_{leg}$ of the *single leg chain* (Equation 3.8) into two separate chains in Equation (3.11), we can assign the high level task to a specific group of joint. In this example, the task ${}_{12}^6\vec{X}$ is manipulated by the swing leg joints only while the stance leg joints are specified by some other high level task. Therefore, Equation (3.11) is simplified as:

$${}_{12}^6\vec{X} = J_{swing} \cdot \vec{\theta}_{swing} \quad (3.12)$$

The solution to Equation (3.12) ${}^{ref}\vec{\theta}_{swing}$ is obtained from the LS pseudoinverse method:

$${}^{ref}\vec{\theta}_{swing} = J_{swing}^\dagger(\vec{\theta}_{leg}) \cdot {}_{12}^6\vec{X} \quad (3.13)$$

Note that the solution in Equation (3.13) is dependent on the joint configuration of both stance and swing legs. Figures 3.8 to 3.11 show the results of a swing foot position tracking test that is used to validate this approach. The kinematics of the biped is divided into two kinematic chains in the same manner that is described in this section. As oppose to the unnatural gait in Figure 3.7, the solution produced by this kinematic setup (Equation 3.13) manipulates the swing foot using only the joints in the swing leg (Figure 3.8). This is further verified by comparing the hip, knee and foot pitch ankles of the swing leg to the stance leg. Notice that most of the joint motions take place in the swing leg (Figures 3.12 to 3.14). This shows that Equation

3.13 assumes the stance leg joints to be “fixed” and only find LS solutions for the swing leg joints. At the same time, however, the pseudoinverse matrix in Equation 3.13 takes into account the instantaneous joint configuration in the stance leg. Thus, using this method, we can delegate high-level task (in this case, tracking a reference swing foot motion) to a specific chain in the biped.

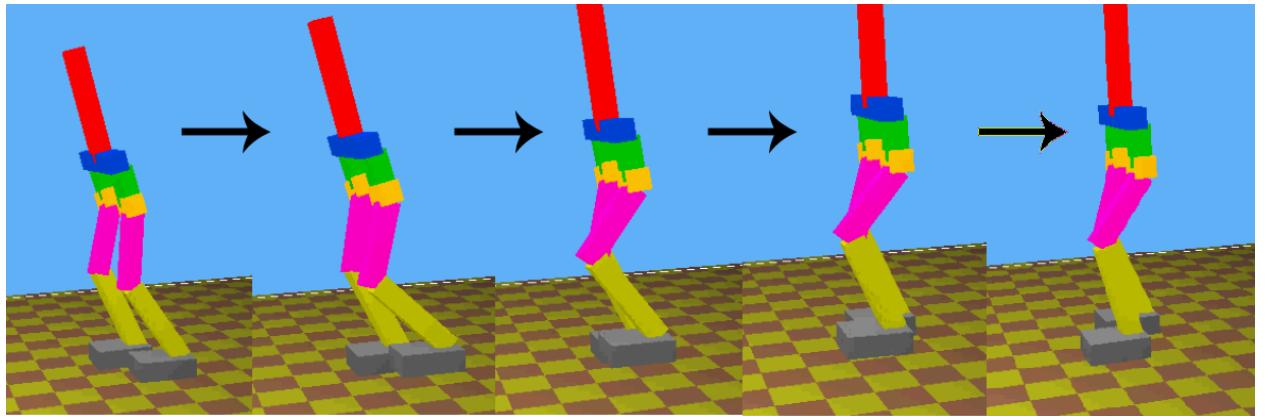


Figure 3.8: Swing leg motion produced by the joints in the swing leg chain.

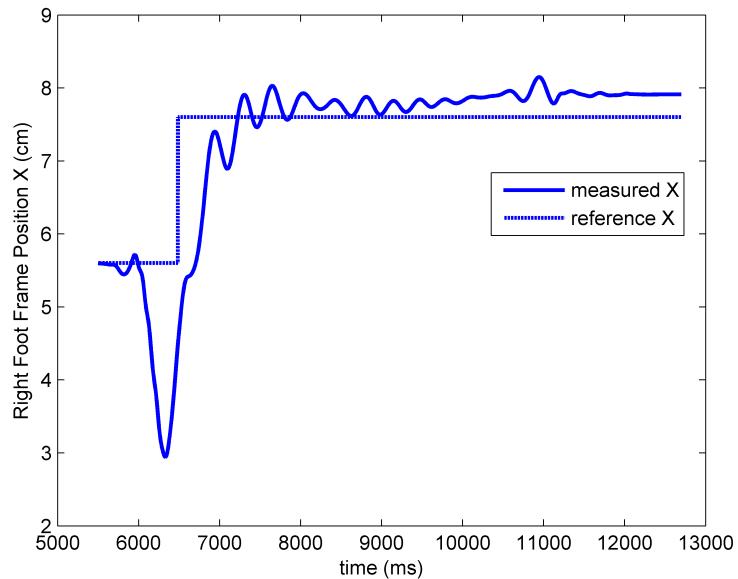


Figure 3.9: Comparison between the actual and reference swing foot X position.

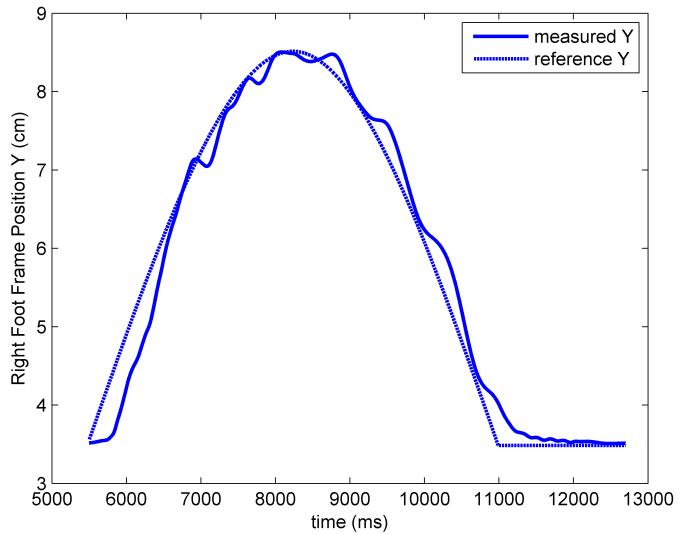


Figure 3.10: Comparison between the actual and reference swing foot Y position.

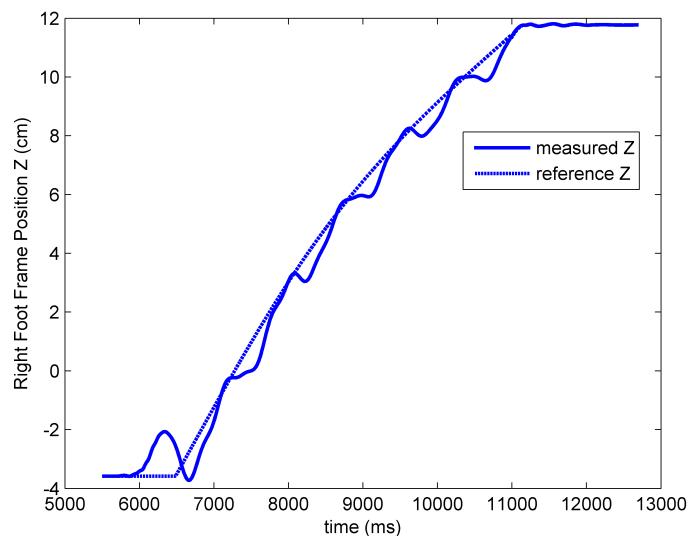


Figure 3.11: Comparison between the actual and reference swing foot Z position.

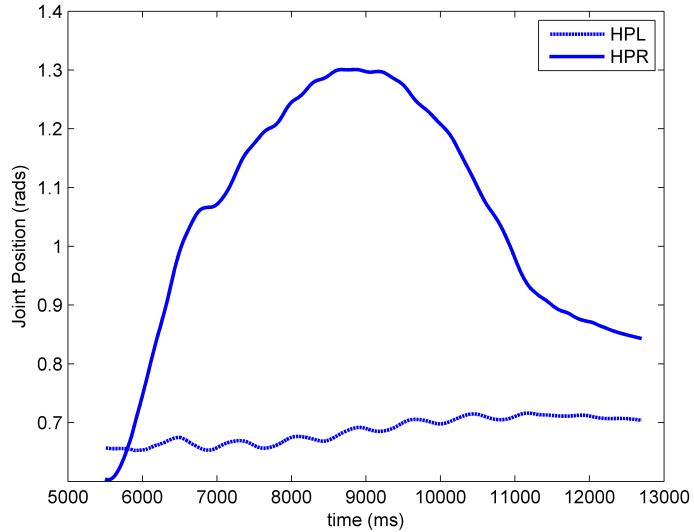


Figure 3.12: Pitch joint angle position comparisons between the stance and swing leg (HPL vs. HPR).

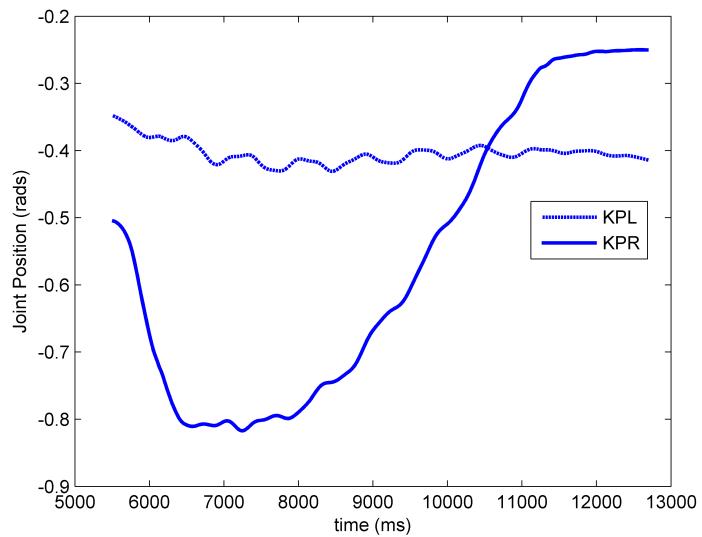


Figure 3.13: Pitch joint angle position comparisons between the stance and swing leg (KPL vs. KPR).

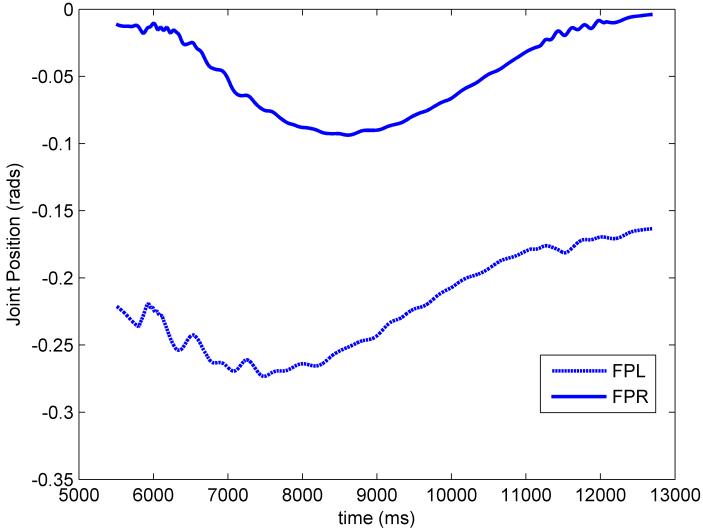


Figure 3.14: Pitch joint angle position comparisons between the stance and swing leg (FPL vs. FPR).

To summarize, the objective in this example is to control the swing foot position by using the joints in the swing leg chain. In the forward kinematics problem, the swing foot position with respect to a reference frame (stance foot frame) is calculated by effectively treating the biped as a single leg chain. For the inverse kinematics problem, the joint space solution is constrained to use only the swing leg joints to produce the desired swing foot position. The results show that the desired swing foot position can be tracked using the swing leg joints while adjusting to the motion in the stance leg.

Although we can simplify the kinematics and control of the biped by assigning high level tasks to separate kinematic chains, the tradeoff is the reduced flexibility due to the lower number of joints that are available for each task. For example, under a large disturbance, the biped might require the coordinated movement of all its joints (not just the joints in the stance leg) to maintain balance. One possible solution to this problem is to define a “normal walking mode” that models the biped

using multiple kinematic chains and a “disturbance compensation mode” that uses a different kinematic setup to handle cases where “whole body cooperation” (WBC) is required. This would, however, require a high level decision-making control layer called the *executive layer* [36] that is beyond the scope of this thesis. Aside from reduced flexibility, another issue is that the coupled motion between the stance and swing leg might result in conflicts between the high level tasks of the chains. However, in normal walking, these conflicts can be avoided by planning the task motions reasonably within the operational workspace specified by the locomotion states.

The next section will provide formal definitions for the kinematics chains in the double stance and single stance state.

3.3.2 Definition of the Kinematic Chains

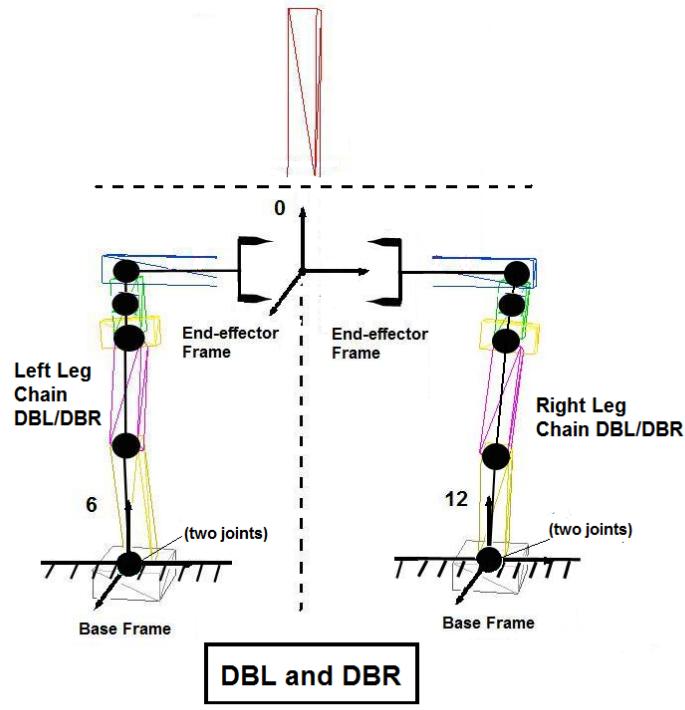
Due to the varying kinematic structure of the biped, different kinematic chains must be defined for each locomotion state. This section will define the kinematic chains by specifying the sequential order of the frames in the chain starting from the base frame to the end-effector frame. In addition, we will also specify the high level tasks that are assigned for each kinematic chain. The definition of the kinematic chains is organized in Table 3.2, a graphical representation of the kinematic chains are provided in Figure 3.15, and the high level task assignment for each kinematic chain is presented in Table 3.3. Please refer back to Table 2.4 and Figure 2.6 for the frame number and location, and Section 3.1 for detail descriptions of the high level tasks. The kinematic chains that are defined in this section will be used extensively in the forward and inverse kinematics problem that will be presented in Sections 3.4 and 3.5.

Table 3.2: The definition of the kinematic chains in each locomotion state.

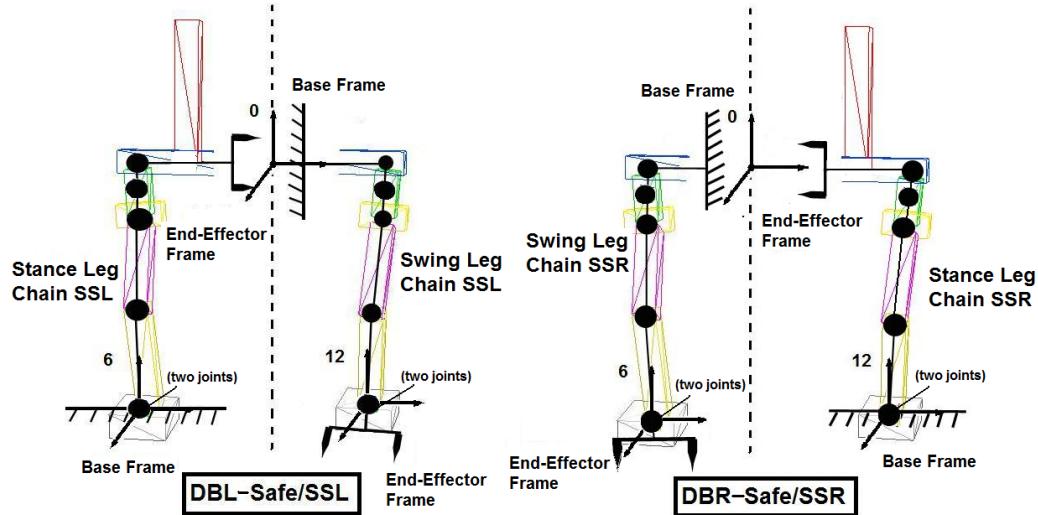
Locomotion State	Kinematic Chain	Sequential order of the frames
DBL	Left Leg Chain DBL	Frame 6,5,4,3,2,1,0,13
	Right Leg Chain DBL	Frame 12,11,10,9,8,7,0
DBL_Safe, SSL	Stance Leg Chain SSL	Frame 6,5,4,3,2,1,0,13
	Swing Leg Chain SSL	Frame 0,7,8,9,10,11,12
DBR	Left Leg Chain DBR	Frame 6,5,4,3,2,1,0
	Right Leg Chain DBR	Frame 12,11,10,9,8,7,0,13
DBR_Safe, SSR	Stance Leg Chain SSR	Frame 12,11,10,9,8,7,0,13
	Swing Leg Chain SSR	Frame 0,1,2,3,4,5,6

Table 3.3: The high level task assignment for the kinematic chains.

Kinematic Chain	High-level Tasks
Left Leg Chain DBL	COG Position Tracking, Pelvis Orientation
Right Leg Chain DBL	COG Position Tracking, Pelvis Orientation
Stance Leg Chain SSL	COG Position Tracking, Pelvis Orientation
Swing Leg Chain SSL	Swing Foot Position and Orientation Tracking
Left Leg Chain DBR	COG Position Tracking, Pelvis Orientation
Right Leg Chain DBR	COG Position Tracking, Pelvis Orientation
Stance Leg Chain SSR	COG Position Tracking, Pelvis Orientation
Swing Leg Chain SSR	Swing Foot Position and Orientation Tracking



(a) Kinematic chains in DBL and DBR.



(b) Kinematic chains in DBL-Safe, SSL, DBR-Safe, and SSR.

Figure 3.15: The kinematic chain definitions for each locomotion state.

3.4 The Forward Kinematics of ABL-BI

3.4.1 Overview

In order for the biped to achieve the high level objectives that are described in Section 3.1, it must obtain the associated output variables for feedback control. The outputs of the biped that are of interest are: 1) the COG position and velocity, 2) the angular position and velocity of the pelvis, 3) the linear position and velocity of the swing foot, and 4) the angular position and velocity of the swing foot. This section is divided into three parts. In 3.4.2, we will discuss how the output variables are measured from simulated sensors or calculated from the kinematic equations in the *simulated biped*. In Section 3.4.3, we will discuss how the output variables are obtained in the *actual* ABL-BI platform. Unlike simulation, the joint position is the only output variable that is measured directly from sensors (encoders); all other output variables are calculated from the forward kinematic equations of the actual biped. Therefore, the focus of the discussion will be on the formulation and validation of the forward kinematics equations. Finally, in Section 3.4.4, we will introduce a *local inertial frame* that has the same orientation as the world frame but is attached to the stance foot of the biped (Figure 3.16). The local inertial frame is used to define the heading (angular position about the vertical axis) of the pelvis and the swing foot in the actual biped.

3.4.2 The Output Variables of the Simulated Biped

In the simulated biped, most of the output variables are measured directly from the simulated sensors in *Webots*. There are two reasons for this: 1) the complexity of the robot controller in the simulation is reduced, and 2) the “measured” output variables in the simulation can be used to validate the forward kinematics calculations in the actual biped. From the point of view of reference motion planning, it is more

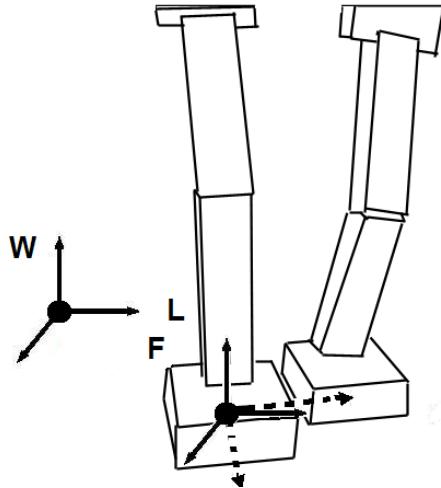


Figure 3.16: The local inertial frame (Frame L, solid line) is attached to the stance foot. It has the same orientation as the world frame (Frame W) and does not rotate with the stance foot frame (Frame F, dotted line).

convenient to specify the reference swing foot position with respect to the stance foot. The reason is that one of the intermediate objective in SS is to place the swing foot at a certain step length distance ahead of the stance foot (Section 3.2.2). Therefore, to facilitate tracking control of the swing foot, the actual swing foot position is measured with respect to the stance foot frame (assumed to be fixed to the ground). All other output variables, which includes the swing foot orientation, the pelvis orientation, and the COG position and velocity, use the world frame as the reference. The following sections will summarize how output variables are obtained from the simulated biped.

The Orientation of the Pelvis and the Swing Foot

The orientation of the pelvis (described by Frame 0) with respect to the world frame is expressed as a vector ${}^W_0\Omega = [\beta, \alpha, \gamma]^T$, where β , α , and γ are the Fixed Euler Angles about the X, Y, and Z axis of the world frame respectively. In *Webots*, ${}^W_0\Omega$, is measured from a *GPS sensor* (also functions as an IMU) that is attached to the local COG position of the pelvis. Similarly, the angular position of the swing foot in SSR ${}^W_6\Omega$ and the swing foot in SSL ${}^W_{12}\Omega$ are measured from GPS sensors in *Webots*.

The angular velocity of the pelvis and the swing foot can be approximated from the measured Euler Angles. First, the velocity of the Euler Angles $\dot{\Omega}$ is calculated by the following equation:

$$\dot{\Omega} = \frac{\Omega(t) - \Omega(t-1)}{Ts} \quad (3.14)$$

where $\Omega(t)$ and $\Omega(t-1)$ are the Euler angles of the current and previous time step respectively and Ts is the constant simulation and controller step period (1 ms in *Webots*). Note that in general, angular velocity ω is different from the velocity of the Euler Angles $\dot{\Omega}$. Therefore, in order to obtain the angular velocity ω , $\dot{\Omega}$ must be pre-multiplied by a 3×3 matrix $R_{\omega dot}$ that defines the mapping between $\dot{\Omega}$ and ω . $R_{\omega dot}$ is given by the following equation [37]:

$$R_{\omega dot} = \begin{bmatrix} c(\Omega_X)c(\Omega_Y) & -s(\Omega_Y) & 0 \\ c\Omega_X s\Omega_Y & c(\Omega_Y) & 0 \\ -s(\Omega_X) & 0 & 1 \end{bmatrix}, \quad (3.15)$$

and the angular velocity of the pelvis or the swing foot frame (represented by a general variable ω) is obtained from the following equation:

$$\omega = R_{\omega dot}\dot{\Omega}. \quad (3.16)$$

The COG Position and Velocity of the Simulated Biped

The COG position with respect to the world frame ${}^W\vec{P}_G$ is calculated from the mass of each individual link m_i and the locations of their local COG ${}^W\vec{P}_{Gi}$ (measured from GPS sensors) as follows:

$${}^w\vec{P}_G = \frac{\sum_{i=1}^{13} m_i \cdot {}^w\vec{P}_{Gi}}{\sum_{i=1}^{13} m_i}, \quad (3.17)$$

The COG velocity with respect to the world frame \vec{X}_G is calculated from the joint velocities $\vec{\theta}$ and the *COG Jacobian* WJ_G that is derived in Appendix D. Note that the COG Jacobian is derived under the assumption that there are no relative motion between the stance foot frame and the world frame. Unlike other output variables, the COG velocities that are calculated from forward kinematics are preferred over the measured COG velocities due to the large amount of noise in the COG position data (contributed by the noise in the local COG position of each link).

$$\vec{X}_G = {}^WJ_G(\vec{\theta}) \cdot \vec{\dot{\theta}}. \quad (3.18)$$

The Swing Foot Position of the Simulated Biped

The swing foot position with respect to the stance foot frame (for SSL and SSR) is calculated from measured values of the swing foot position and the stance foot position:

$${}^{12}\vec{P}_{ORG} = {}^W\vec{P}_{ORG} - {}^W\vec{P}_{6ORG}, \quad (3.19a)$$

$${}^6\vec{P}_{ORG} = {}^W\vec{P}_{ORG} - {}^{12}\vec{P}_{ORG}, \quad (3.19b)$$

where ${}_j\vec{P}_{ORG}$ is the frame origin position of Frame j with respect to Frame i . The swing foot velocity can be obtained from the Jacobian matrices ${}^{12}J_6$ (for SSL) and ${}^{12}J_6$ (for SSR) that relates the swing foot velocity to the joint space velocities (Equations E.23 and E.27 in Appendix E).

3.4.3 The Output Variables in the Actual Biped

The following section will focus on the output variables of the actual biped and the forward kinematic equations of the kinematic chains. In addition, the kinematic equations will be validated by a series of tests in *Webots*.

The Swing Foot Position

As noted in the beginning of this section, the biped is treated as separate kinematic chains and the *inaccessible states* of the actual robot can be equivalently obtained from the position/orientation of the end-effector frames. Let's start with the calculation of the swing foot position with respect to the stance foot. In SS, the swing leg of the biped is modelled as an open kinematic chain with a floating base frame 0 (Figure 3.15). The position of the floating base frame with respect to the stance foot frame is calculated from the kinematics of the stance leg chain. Therefore, the swing foot frame with respect to the stance foot frame can be calculated by effectively treating the biped as a *Single Leg Chain* (Figure 3.6). Assuming the biped is in SSL, the origin of Frame 12 (end-effector) with respect to Frame 6 (base) is obtained by first multiplying the individual link transformation matrices ${}^{i-1}_iT$ (as defined in Appendix B for ABL-BI):

$${}_{12}^6T = {}_5^6T {}_4^5T {}_3^4T \dots {}_0^1T {}_7^0T {}_8^7T \dots {}_{12}^{11}T. \quad (3.20)$$

The position of the swing foot frame ${}_{12}^6\vec{P}_{ORG}$ is then obtained from the elements in the last column of the transformation matrix ${}_{12}^6T$ (refer to Equation 3.6). Similarly, ${}_{12}^6\vec{P}_{ORG}$ is obtained from the transformation matrix ${}_{12}^6T$ when the biped is in SSR. Note that the kinematic equations in ${}_{12}^6\vec{P}_{ORG}$ and ${}_{12}^6\vec{P}_{ORG}$ and the transformation matrices ${}^{i-1}_iT$ are dependent on the joint variables θ_1 to θ_{13} and constant parameters ${}^{i-1}\vec{FD}$ that specify the relative position between the origins of each successive

frames in the kinematic model (Table A.1). Therefore, given the measured $\theta_{1..13}$ (from encoders in the actual robot or *Servos* in Webots) and a set of pre-defined ${}^{i-1}\overrightarrow{FD}$ from the biped model, the swing foot position is calculated from the forward kinematics of the biped.

The kinematic equations for the swing foot position is validated by the following test in *Webots*:

1. A two-step reference gait with a duration of 23000 *ms* (23 seconds) is generated for the biped (Figure 3.17).
2. At every time step in the two-step cycle, the swing foot and stance foot positions with respect to the world frame are measured from GPS sensors. The position of the swing foot with respect to the stance foot is then calculated by Equations 3.19a or 3.19b, depending on the locomotion state of the biped. These results are referred to as the *measured* positions of the swing foot.
3. The joint positions (measured from the *Servos* at every time step), and the robot parameters ${}^{i-1}\overrightarrow{FD}$ are used to calculate the swing foot position with respect to the stance foot from the kinematic equations in ${}^{12}\overrightarrow{P}_{ORG}$ (for the SSL portion of the two-step cycle) or ${}^{12}\overrightarrow{P}_{ORG}$ (for the SSR portion of the two-step cycle). These results are the *calculated* positions of the swing foot.
4. The kinematic equations ${}^{12}\overrightarrow{P}_{ORG}$ or ${}^{12}\overrightarrow{P}_{ORG}$ are validated by comparing the *measured* and the *calculated* swing foot positions (Figures 3.18 and 3.19, and 3.20). Note that the results for an additional swing foot position test (with a large swing foot motion in the lateral direction) is presented in Appendix C.

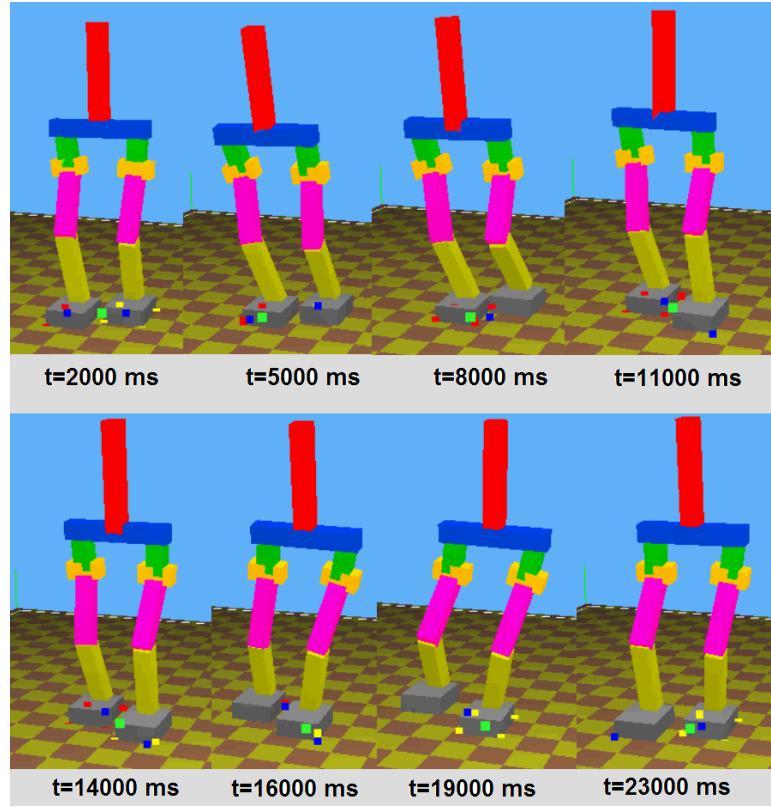


Figure 3.17: The two step cycle reference motion for the swing foot position test in Webots.

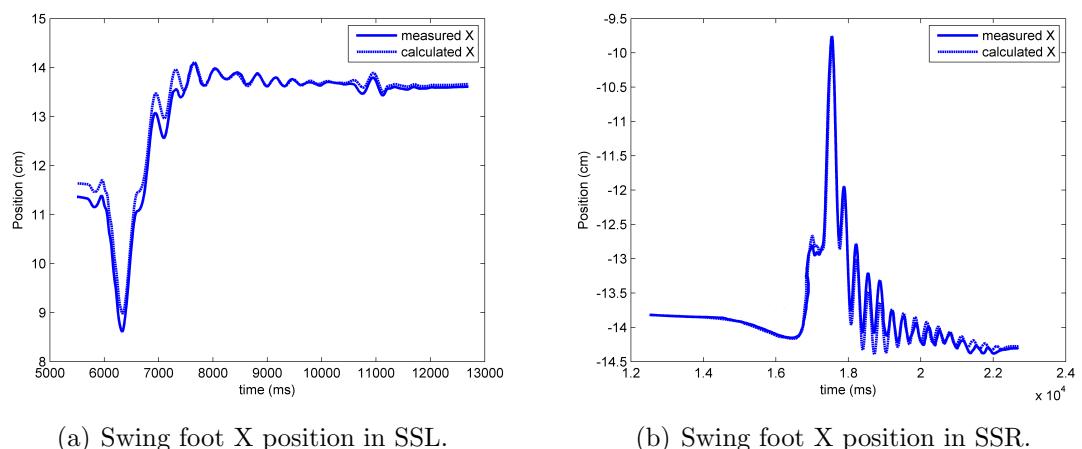


Figure 3.18: The validation test results for the swing foot X position with respect to the stance foot for SSL and SSR.

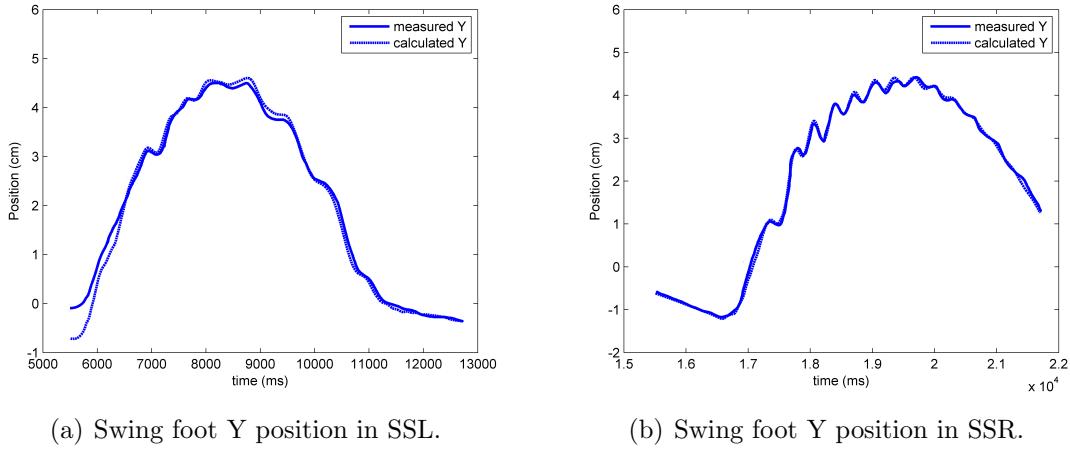


Figure 3.19: The validation test results for the swing foot Y position with respect to the stance foot for SSL and SSR.

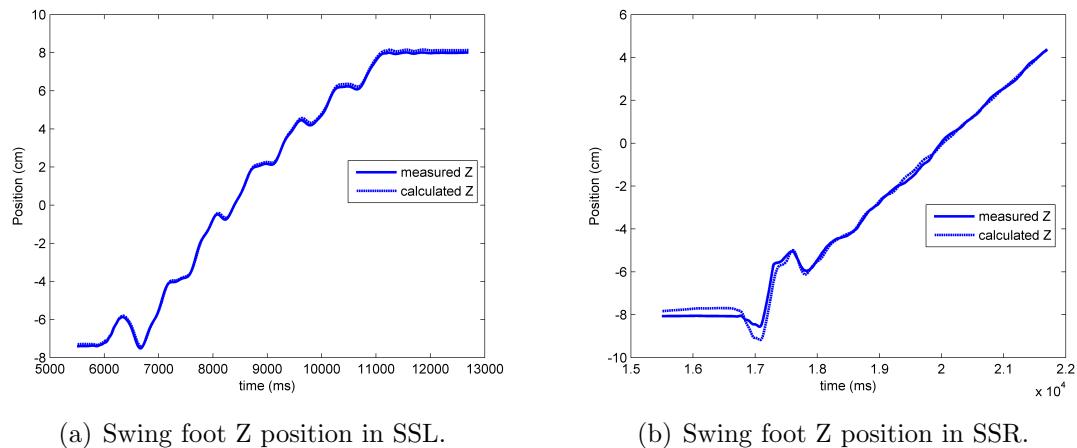


Figure 3.20: The validation test results for the swing foot Z position with respect to the stance foot for SSL and SSR.

The calculated results closely tracks measured results with a maximum error of less than 0.5 cm, which are very good approximations of the physical states (not directly measurable from sensors) from forward kinematics. The small discrepancy is likely due to 1) to small delay (3-4 ms) in the filtered joint positions, and 2) slight relative movements between the stance foot (base of the kinematic chain, assumed to be fixed on the ground) and the world. But for all practical purposes, the swing foot positions can be approximated from the biped kinematics.

The COG Position and Velocity

Unlike the swing foot position, the COG position is dependent on the mass and link COG positions of every link in the biped. The calculation for the COG position with respect to the stance foot frame is a two step process: 1) Calculate the COG position with respect to the pelvis frame (Frame 0) ${}^0\vec{P}_G$, and 2) Express the COG position with respect to the stance foot frame.

The COG position with respect to Frame 0 ${}^0\vec{P}_G$ is given by the weighted average of the relative position between the local COG and the origin of Frame 0:

$${}^0\vec{P}_G = \frac{\sum_{i=1}^{13} m_i \cdot {}^0\vec{P}_{Gi}}{\sum_{i=1}^{13} m_i}, \quad (3.21)$$

where ${}^0\vec{P}_{Gi} = {}^0\vec{T} {}^i\vec{P}_{Gi}$.

The COG position of link i with respect to its link frame is denoted by ${}^i\vec{P}_{Gi}$, which are constant vectors defined by the mass properties of ABL-BI. The matrix ${}^0\vec{T}$ is the transformation of the COG of link i with respect to Frame 0, and ${}^0\vec{P}_{Gi}$ is the COG position of link i with respect to Frame 0. Once ${}^0\vec{P}_G$ is calculated, it can be expressed with respect to the stance foot frame (Frame 6 for SSL or Frame 12 for

SSR):

$${}^6\vec{P}_G = {}_0^6T^0\vec{P}_G, \quad (3.22a)$$

$${}^{12}\vec{P}_G = {}_0^{12}T^0\vec{P}_G. \quad (3.22b)$$

The kinematic equations of ${}^6\vec{P}_G$ and ${}^{12}\vec{P}_G$ are dependent on 1) the joint variables, 2) the relative position between each successive frames, and 3) the relative position of the link COG with respect to its link frame.

From the kinematic model, the COG velocity of the biped is calculated using the COG Jacobian and Equation 3.18. For the purpose of validating the COG Jacobian, the COG velocity vector obtained from Equation 3.18 is compared to the COG velocity which is calculated from *Webots* via:

$$\vec{\dot{X}}_G = \frac{\vec{X}_G(t) - \vec{X}_G(t-1)}{T_s}, \quad (3.23)$$

where $\vec{X}_G(t)$ and $\vec{X}_G(t-1)$ are the COG position of the current and previous time step obtained from *Webots* respectively.

Using the same two-step test in Figure 3.17, the COG position that is obtained from the calculated link COG positions from Equation 3.22a (for SSL) and 3.22b (for SSR) are compared with the measured results in Equation 3.17. In addition, in order to validate the COG Jacobian WJ_G derived in Appendix D, the results calculated from Equation 3.18 are compare with the measured results from (Equation 3.23).

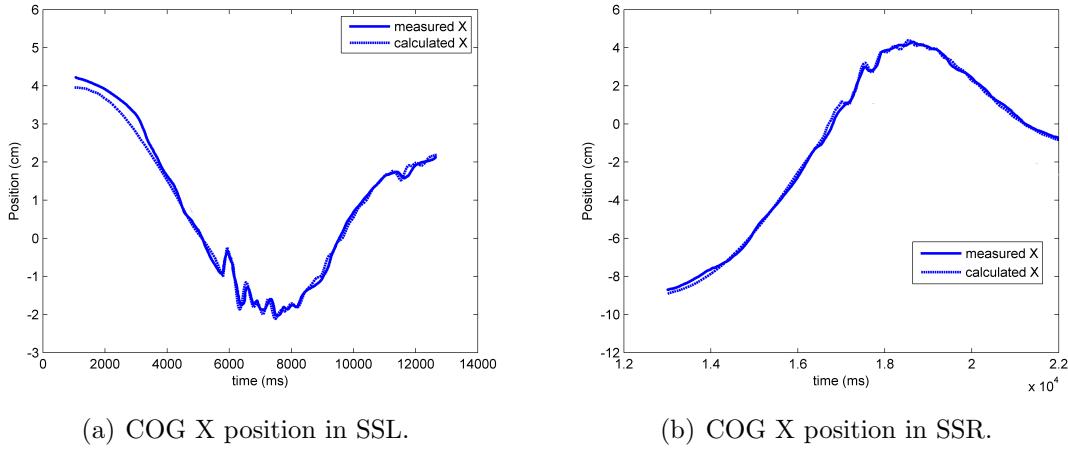


Figure 3.21: The validation test results for the COG X position with respect to the stance foot for SSL and SSR.

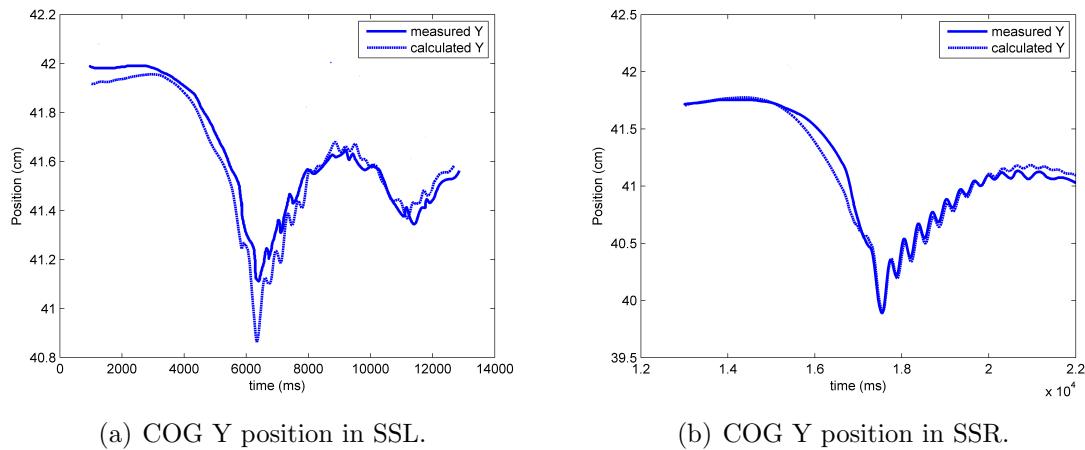


Figure 3.22: The validation test results for the COG Y position with respect to the stance foot for SSL and SSR.

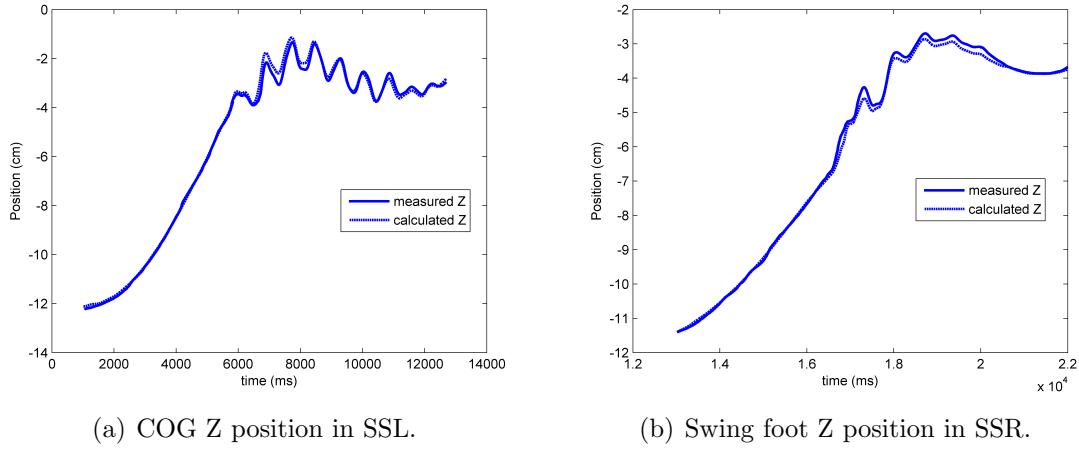


Figure 3.23: The validation test results for the COG Z position with respect to the stance foot for SSL and SSR.

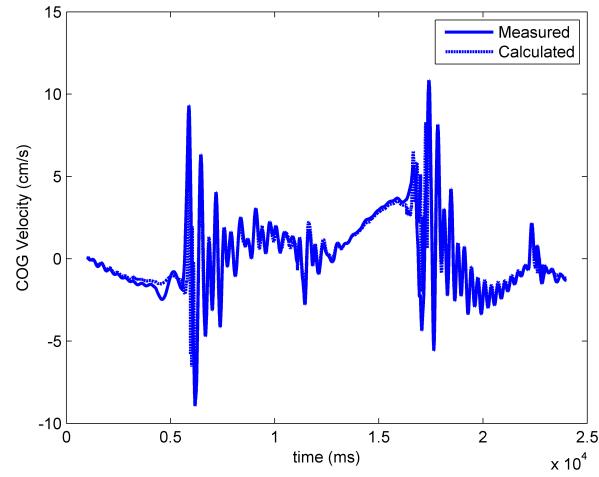


Figure 3.24: The validation test results for the COG X Velocity with respect to the stance foot.

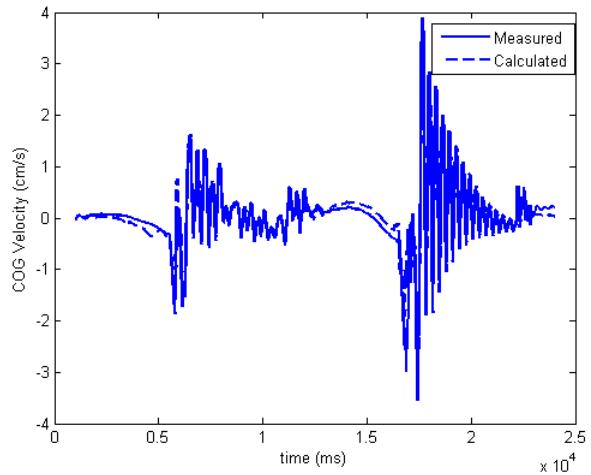


Figure 3.25: The validation test results for the COG Y Velocity with respect to the stance foot.

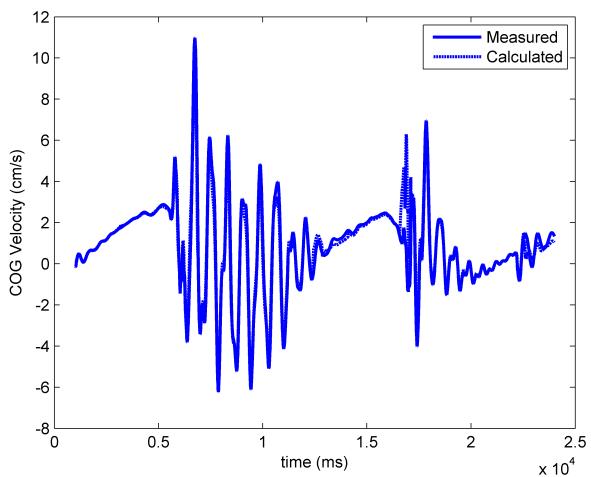


Figure 3.26: The validation test results for the COG Z Velocity with respect to the stance foot.

As shown in the above figures, the calculated COG position and velocities results matches the measured results very closely (within 0.3 cm). The small error is likely caused by slight motion in the stance foot (base of the kinematic chain).

The Pelvis and Swing Foot Orientations

The orientation of the pelvis with respect to the stance foot frame: ${}^6\Omega$ (for SSL), ${}^{12}\Omega$ (for SSR) can be calculated from the forward kinematics. According to the convention stated in Section 2.2.2, the rotation of Frame i is performed about a fixed reference frame (Frame $i - 1$) in the following order: 1) rotate about the Y axis by α (yaw angle), 2) rotate about the X axis by β (pitch angle), and 3) rotate about the Z axis by γ (roll angle). Based on this convention, the rotation matrix ${}^{i-1}_i R$ is derived in Equation 2.1. The elements of the rotation matrix ${}^{i-1}_i R$ are nine equations that are in terms of the Euler angles β , α , and γ . Out of the nine equations, only three are independent. Therefore, the Euler angles can be extracted from the elements of ${}^{i-1}_i R$ by solving the three unknowns β , α , and γ from three equations. First, β is computed from the elements $R_{3,1}$, $R_{3,2}$, and $R_{3,3}$:

$$c(\beta) = \sqrt{R_{3,1}^2 + R_{3,2}^2} = \sqrt{c^2\beta(s^2\alpha + c^2\alpha)}, \quad (3.24a)$$

$$\beta = \arctan \left(\frac{R_{3,2}}{\sqrt{R_{3,1}^2 + R_{3,2}^2}} \right) = \arctan \left(\frac{s\beta}{c\beta} \right). \quad (3.24b)$$

Then, using β , $R_{3,1}$, and $R_{3,3}$, we can compute α :

$$s\alpha = \frac{-R_{3,1}}{c\beta} = \frac{c\beta s\alpha}{c\beta}, \quad (3.25a)$$

$$c\alpha = \frac{R_{3,3}}{c\beta} = \frac{c\beta c\alpha}{c\beta}, \quad (3.25b)$$

$$\alpha = \arctan \left(\frac{\frac{-R_{3,1}}{c\beta}}{\frac{R_{3,3}}{c\beta}} \right) = \arctan \left(\frac{s\alpha}{c\alpha} \right). \quad (3.25c)$$

Similarly, using β , $R_{1,2}$ and $R_{2,2}$, we compute γ :

$$s\gamma = \frac{-R_{1,2}}{c\beta} = \frac{c\beta s\gamma}{c\beta}, \quad (3.26a)$$

$$c\gamma = \frac{R_{2,2}}{c\beta} = \frac{c\beta c\gamma}{c\beta}, \quad (3.26b)$$

$$\gamma = \arctan \left(\frac{\frac{-R_{1,2}}{c\beta}}{\frac{R_{2,2}}{c\beta}} \right) = \arctan \left(\frac{s\gamma}{c\gamma} \right). \quad (3.26c)$$

Note that if $\beta = \pm 90^\circ$, $c\beta = 0$ and the solutions to Equations 3.25 and 3.26 will be undefined. However, this situation requires the pelvis frame or the swing foot frame to pitch 90° with respect to the stance foot frame. In typical walking, the biped would not adopt this configuration unless it has fallen; therefore, Equations 3.24, 3.25, and 3.26 can be used to extract the Euler angles from the rotation matrix even though they may have degenerated cases.

Going back to our initial problem of finding ${}_0^6\Omega$ and ${}^1{}_0^{12}\Omega$, the first step is to calculate the rotation matrix of the pelvis frame with respect to the stance foot frame. Assuming SSL, ${}_0^6R$ is calculated by multiplying the link rotation matrices in the *Left Leg Chain DBL* (Figure 3.15):

$${}_0^6R = {}_5^6R {}_4^5R {}_3^4R \dots {}_0^1R \quad (3.27)$$

Similarly in SSR, ${}^1{}_0^{12}R$ can be calculated from the link rotation matrices in the *Swing Leg Chain SSR*. For DBL and DBR, the rotation matrices are calculated from the *Left Leg Chain DBL* and *Right Leg Chain DBR* in Figure 3.3.2 respectively. Once the rotation matrices are obtained, the Euler angles ${}_0^6\Omega$ and ${}^1{}_0^{12}\Omega$ can be extracted

from Equations 3.24, 3.25, and 3.26.

Using the same equations, the Euler angles of the swing foot with respect to the stance foot (${}_{12}^6\Omega$ for SSL and ${}_{6}^{12}\Omega$ for SSR) are extracted from the rotation matrices ${}_{12}^6R$ and ${}_{6}^{12}R$. The rotation matrices are calculated from the link rotation matrices in the *single leg chain* (Figure 3.6). Assuming SSL, ${}_{12}^6R$ is calculated from the following equation:

$${}_{12}^6R = {}_5^6R_4^5R_3^4R \dots {}_0^1R_7^0R_8^7R \dots {}_{12}^{11}R. \quad (3.28)$$

Similarly, ${}_{6}^{12}R$ is calculated by treating Frame 12 as the base and Frame 6 as the end-effector in the *single leg chain*.

$${}_{6}^{12}R = {}_6^5R_5^4R_4^3R \dots {}_1^0R_7^7R_7^8R \dots {}_{11}^{12}R. \quad (3.29)$$

The orientation of the pelvis and swing foot calculated from the biped kinematics is validated by the following test in *Webots*:

1. Set the gravity of the simulation world to zero and translate the biped vertically above the ground.
2. In order to ensure that the stance foot does not move, a very large mass (represented by the black block in Figure 3.27) is attached to the bottom of the stance foot. This attachment produces the same effect as fixing the stance foot to the ground.
3. The biped is given a set of reference joint positions that will produce robot configurations with distinctive relative orientations between the swing foot/pelvis and the stance foot (Figures 3.27, 3.31, 3.35, and 3.39).

4. To validate the calculations of the pelvis/swing foot frame orientation with respect to the stance foot frame (in SSL and SSR) from the biped kinematics, the calculated results are compared with the measured orientations with respect to the world frame. Note that additional tests on the pelvis and swing foot orientation is presented in Appendix C.

The following plots are the pelvis orientation test results for the biped in SSL and SSR.

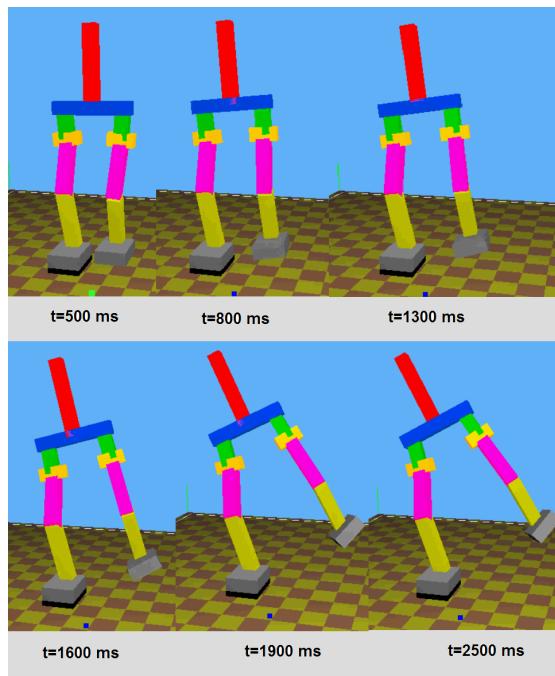


Figure 3.27: The reference robot motions for the pelvis orientation test in SSL.

The following plots are the swing foot orientation test results for the biped in SSL. The following plots are the swing foot orientation test results for the biped in SSR.

The above figures show that the error between the calculated and measured orientation is very small (the maximum error is less than 0.5 degree). Therefore, the pelvis and swing foot orientation can be calculated from the biped kinematics with good accuracy.

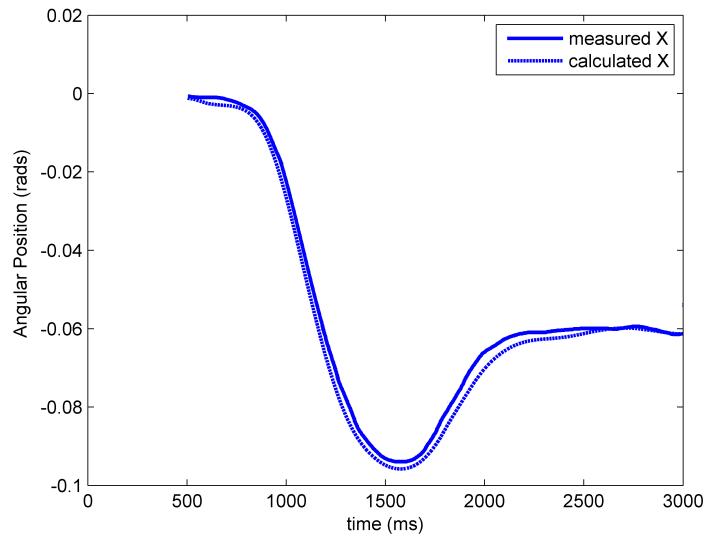


Figure 3.28: The validation test results for the pelvis orientation (about the X axis) in SSL.

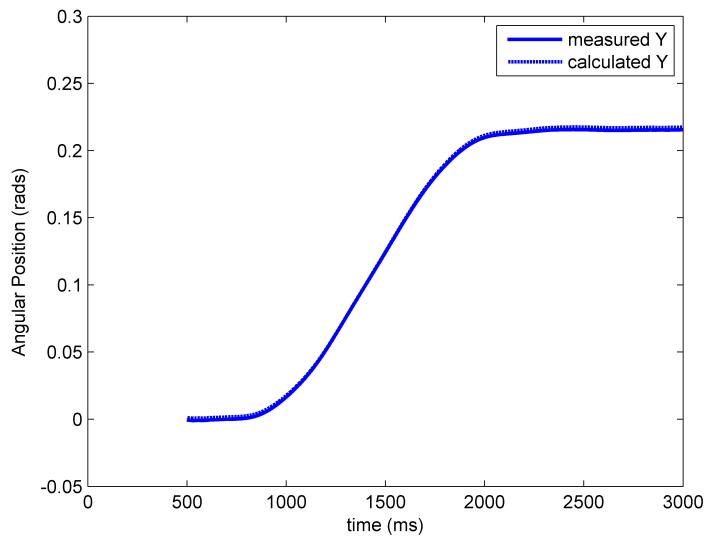


Figure 3.29: The validation test results for the pelvis orientation (about the Y axis) in SSL.

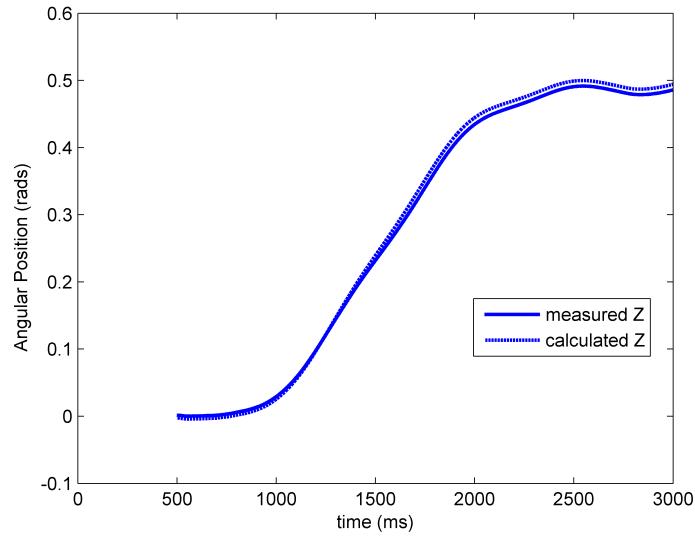


Figure 3.30: The validation test results for the pelvis orientation (about the Z axis) in SSL.

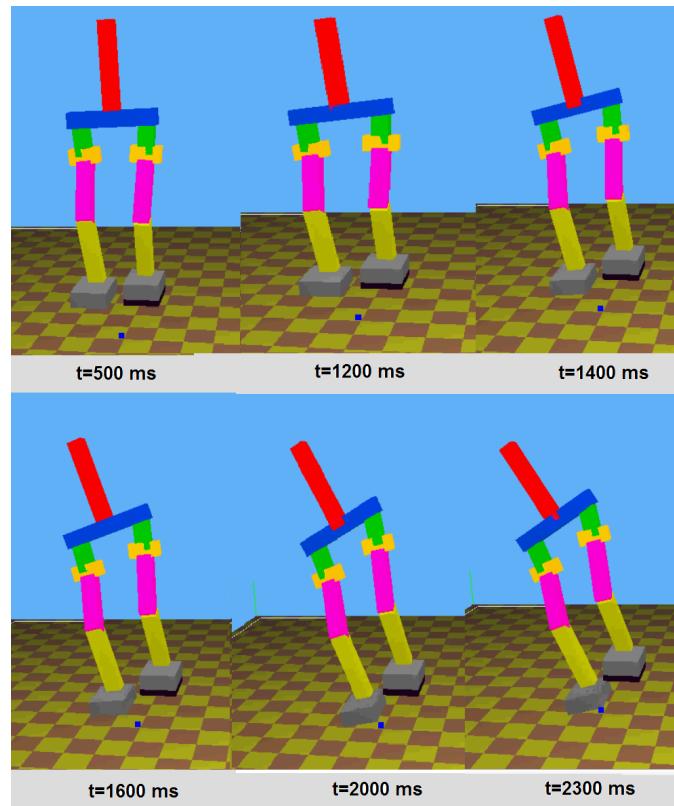


Figure 3.31: The reference robot motions for the pelvis orientation test in SSR.

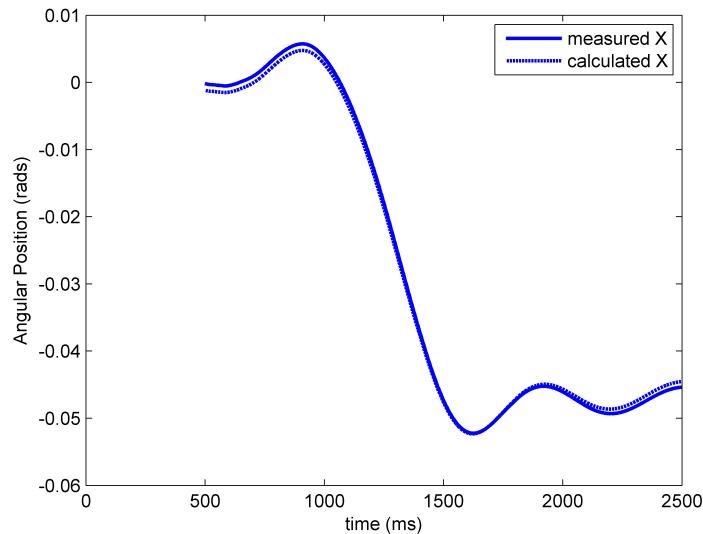


Figure 3.32: The validation test results for the pelvis orientation (about the X axis) in SSR.

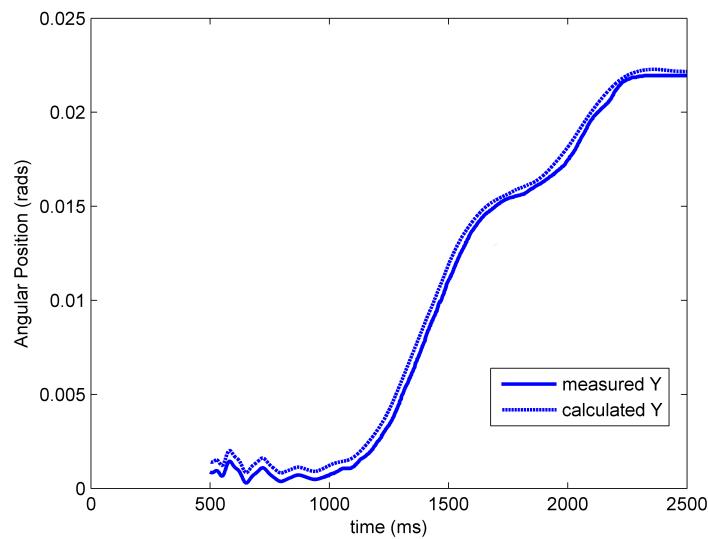


Figure 3.33: The validation test results for the pelvis orientation (about the Y axis) in SSR.

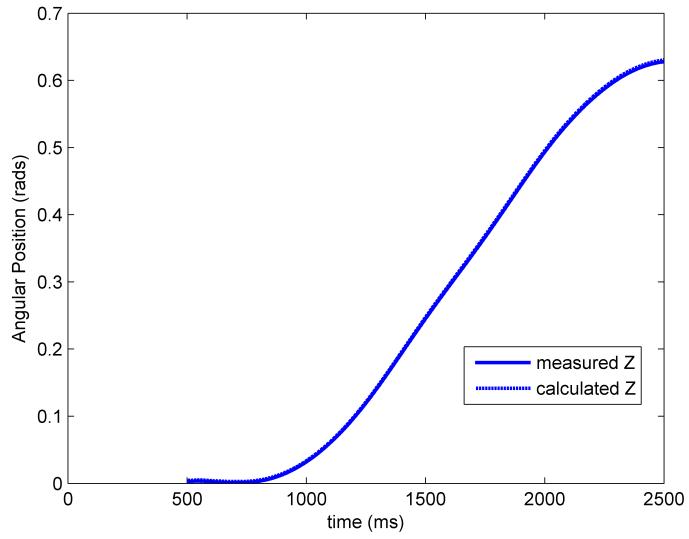


Figure 3.34: The validation test results for the pelvis orientation (about the Z axis) in SSR.

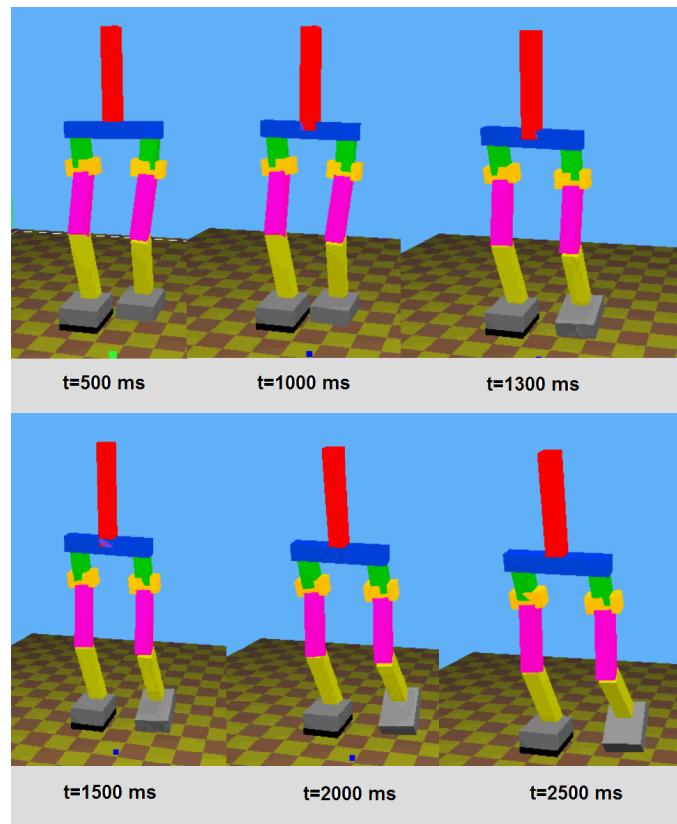


Figure 3.35: The reference robot motions for the swing foot orientation test in SSL.

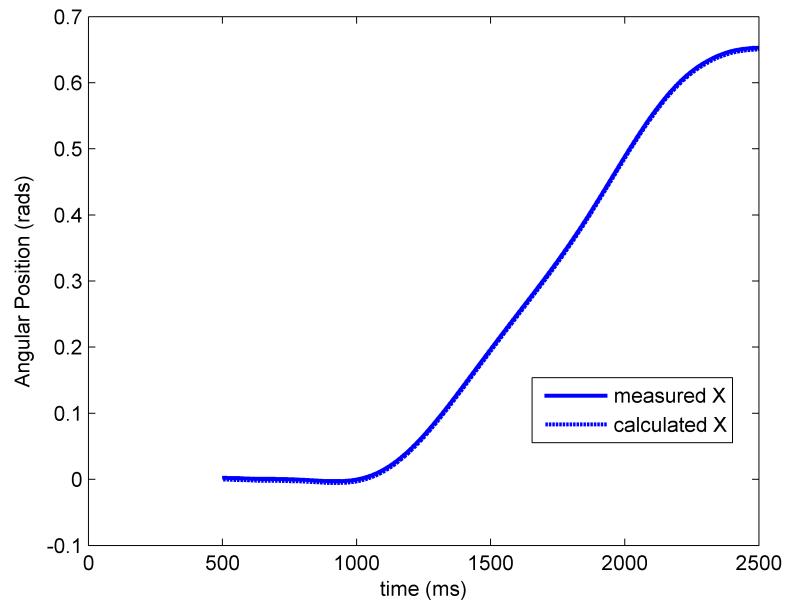


Figure 3.36: The swing foot orientation about the X axis in SSL.

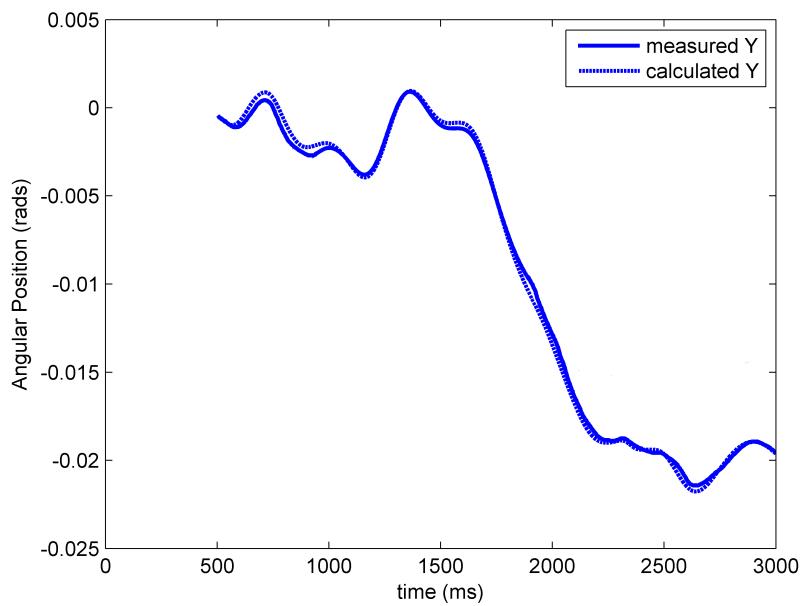


Figure 3.37: The swing foot orientation about the Y axis in SSL.

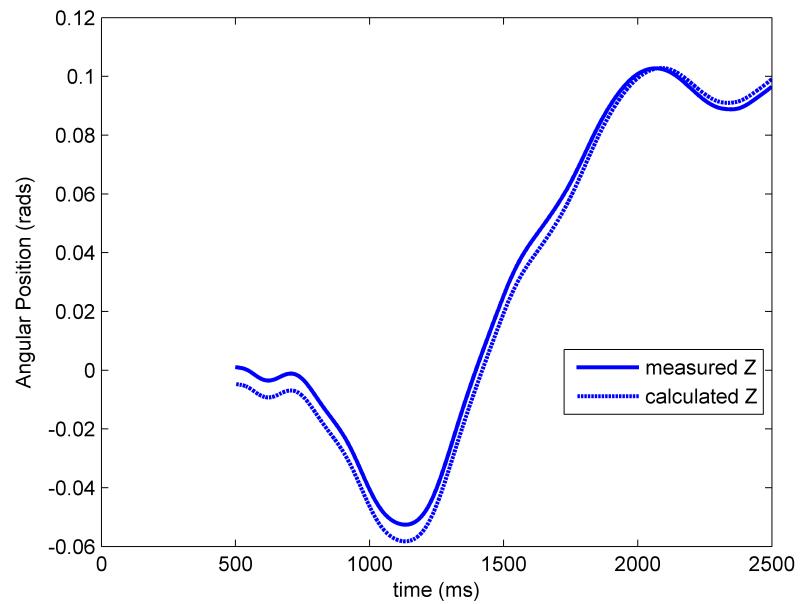


Figure 3.38: The swing foot orientation about the Z axis in SSL.

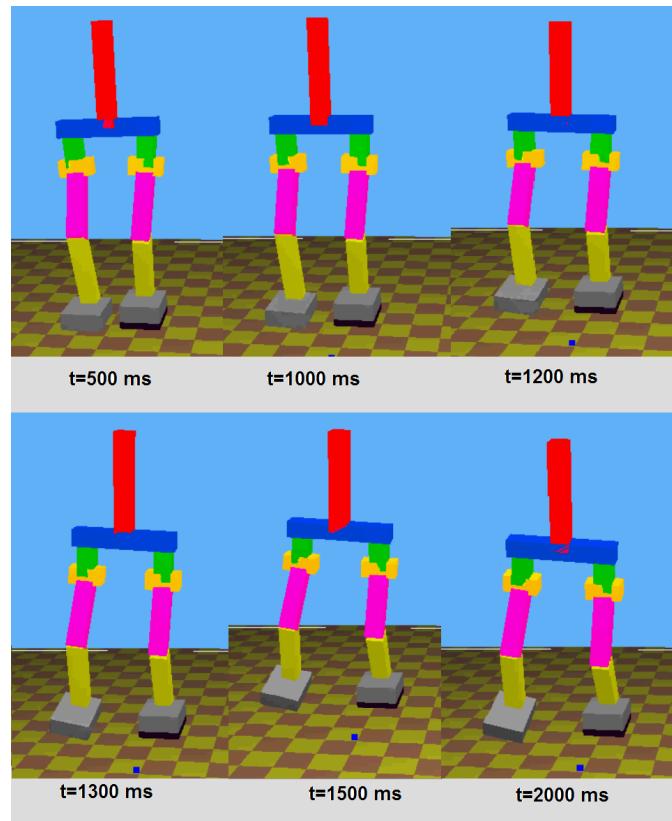


Figure 3.39: The reference robot motions for the swing foot orientation test in SSR.

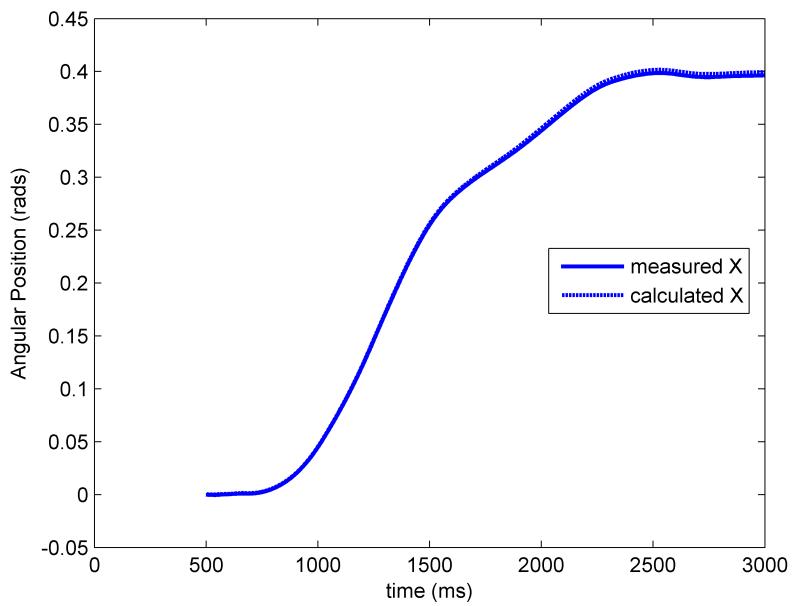


Figure 3.40: The swing foot orientation about the X axis in SSR.

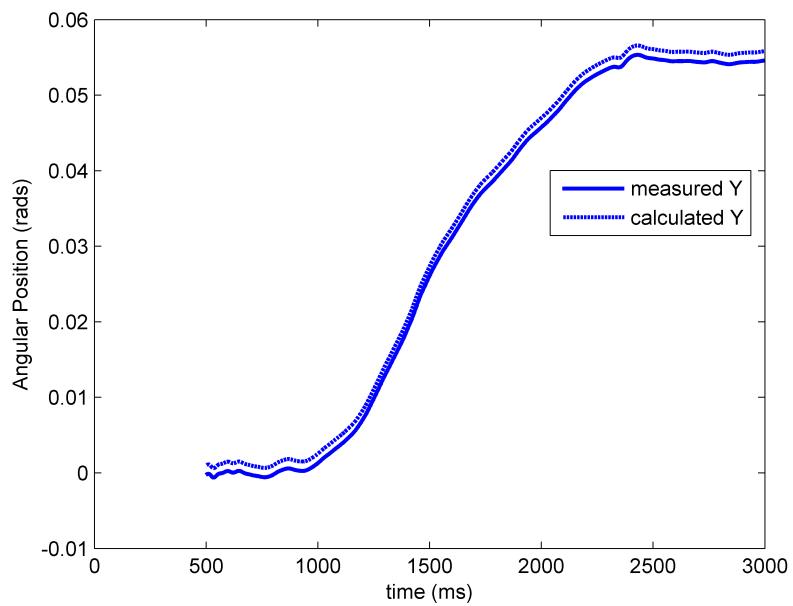


Figure 3.41: The swing foot orientation about the Y axis in SSR.

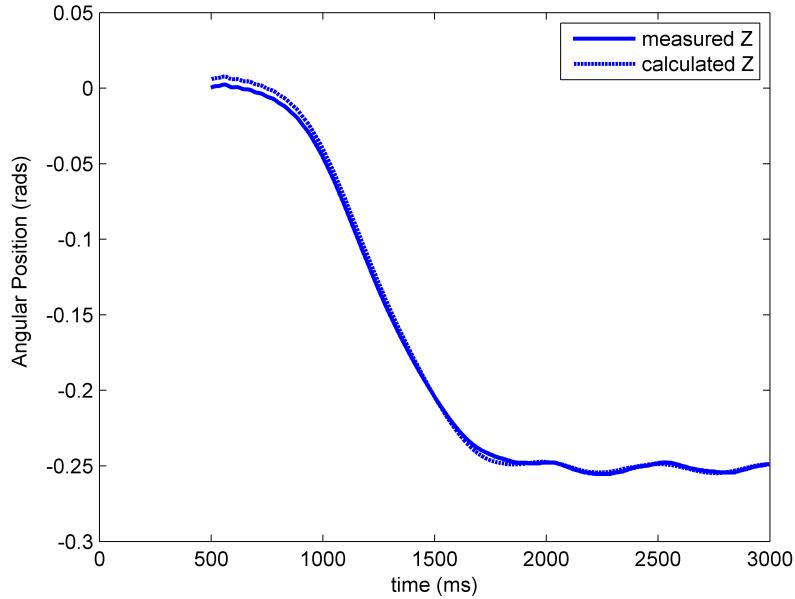


Figure 3.42: The swing foot orientation about the Z axis in SSR.

The Assumption of the Reference Stance Foot Frame

For the above calculations, the stance foot frame is used as the reference frame under the assumption that it does not rotate in the vertical and horizontal plane. However, the violation of this assumption will lead to inaccurate calculations of the pelvis and swing foot orientation. At certain phases of the walking cycle, this can cause the biped to fall. Let's clarify this with an example. At the end of SSL, the Euler angles of the swing foot frame with respect to the *world frame* (and **not** the stance foot frame) must be to zero (ie: ${}^W\Omega = [0, 0, 0]$) in order for the swing foot to land flat on the ground for proper ground contact. Therefore, it is obvious that calculating and tracking a reference ${}^{12}\Omega = [0, 0, 0]$ does not guarantee that the swing foot will be level with the ground. In fact, if the stance foot rotates about the X or Z axis, tracking ${}^{12}\Omega = [0, 0, 0]$ will cause the swing foot to land on its edge and the biped to fall. However, there are two justifications for using the stance foot as the reference frame:

1. The rigid foot considered for ABL-BI does not provide the same flexibility as a human foot or the “two-link compliant foot” that was used in the WL-12RVI and WL-12RVII robots [4]. As the result of foot elasticity, the ZMP stability boundary of a flexible foot is defined by a “critical region”. This region provides the biped a “leeway” to employ different control strategies to regain foot contact even after the foot starts to rotate and the biped momentarily loses balance. In contrast, the ZMP stability boundary of the rigid foot is an abrupt edge. If the rigid foot starts to rotate about its edge (ie: the ZMP crosses the boundary of the support polygon), the biped will not be able to generate the required GRF to regain balance due to the limited ground contact (an edge). Therefore, it is safe to assume that unless the biped is falling, the stance foot will not have any rotation in the horizontal plane (especially in single stance) and it can be used as the reference frame.
2. Unlike simulation, the pelvis and the swing foot orientation of the actual robot are unobservable states that cannot be measured directly with respect to the world frame. Instead, we can only approximate the orientations with respect to the stance foot frame from the kinematics of the biped. Unless we can directly measure the link orientation with respect to a fixed reference frame (using an inertial measurement unit), this approximation is the only feasible option available to the actual robot.

In addition to rotations about the X and Z axis, it is also possible for the stance foot to rotate about the Y (vertical) axis due to insufficient friction between the foot and the ground. However, since the weight of the biped is mostly supported by the stance foot throughout the SS state and at the end of the DB state (after the biped performs weight-shift), the traction force (maximum static friction force) is usually sufficient to keep the stance foot from slipping on dry ground surfaces. In fact, in

the actual biped walking tests (note: not presented in this thesis), foot slippage is only observed in the back stance foot during DB state.

Note that the above discussion also applies to the orientation of the pelvis frame with respect to the stance foot. Although the planning and calculation of the pelvis frame orientation can be affected by the motion of the stance foot, small errors are acceptable since the orientation of the pelvis is not as critical to the balance and walking gait of the biped as the swing foot is.

3.4.4 The Local Inertial Frame

When the biped is walking in a straight line, the swing foot usually lands with the same heading as the stance foot. In turning however, the biped can generate the motion by placing the swing foot at a different heading as the stance foot [38]. Therefore, in the general case, it is desirable to calculate and control the orientation of the swing foot with respect to the *local inertial frame* that was described in Section 3.4.1. The accumulated changes in the swing foot orientation with respect to the local inertial frame is approximated by dead reckoning. In order to conveniently compute the changes in the stance foot orientation between the current step cycle (from the start of DB to the end of SS) and the previous step cycle, we calculate the orientation between the stance foot and the swing foot at the end of the SS phase. In other words, the dead reckoning process only accounts for the discrete changes in the stance foot orientation between each step cycle due to the placement of the swing foot.

Let's illustrate this process with an example. As shown in Figure 3.43, the biped starts in DBL with the fixed frame (Frame 6) coincident with the *local inertial frame* (Frame L). Assuming that the stance foot does not rotate about the vertical axis

throughout the step cycle, the swing foot (Frame 12) with respect to Frame L in DBL and SSL can be calculated by the following equation:

$${}_{12}^L R = {}_6^L R {}_{12}^6 R, \quad (3.30)$$

where ${}_6^L R = {}_F^L R = I_3$.

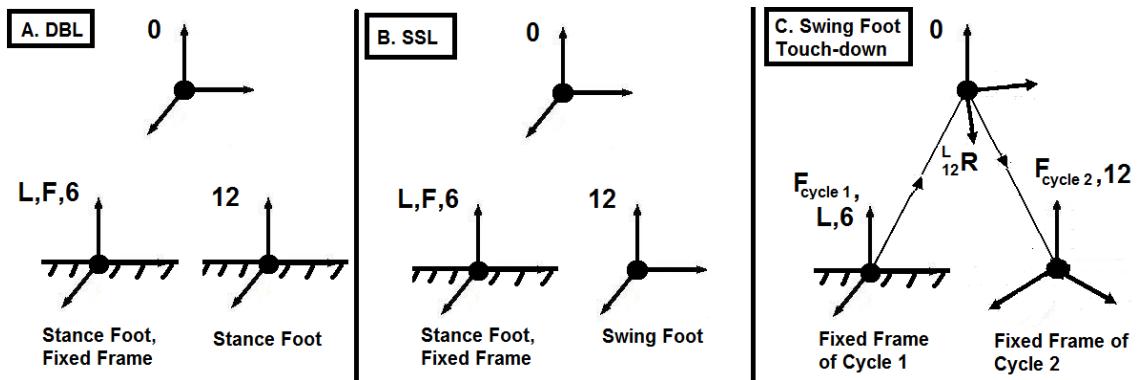


Figure 3.43: The swing foot frame, stance foot frame (fixed frame), the pelvis frame, and the local inertial frame (dotted line) in a SSL step cycle. The right-most figure is the graphical representation of the relative orientation between the swing foot frame (fixed frame of the next cycle) and the local inertial frame.

At the end of SSL, the swing foot (Frame 12) lands and becomes the fixed frame of the next step cycle (C. in Figure 3.43). Assuming that there is a change in the pelvis and swing foot orientation, ${}_F^L R$ is updated by the calculation of the orientation of the swing foot frame (the fixed frame of the next step cycle) with respect to Frame L . Note that ${}_F^L R$ is only updated at touch-down.

$${}_F^L R = {}_{12}^L R = {}_0^L R {}_{12}^0 R, \quad (3.31)$$

where ${}_0^L R = {}_6^L R {}_0^6 R$. (3.32)

In DBR and SSR (Figure 3.44), the swing foot (Frame 6) with respect to Frame L is calculated using the *updated* ${}_F^L R$:

$${}_6^L R = {}_{12}^L R {}_6^R, \quad (3.33)$$

where ${}_{12}^L R = {}_F^L R$.

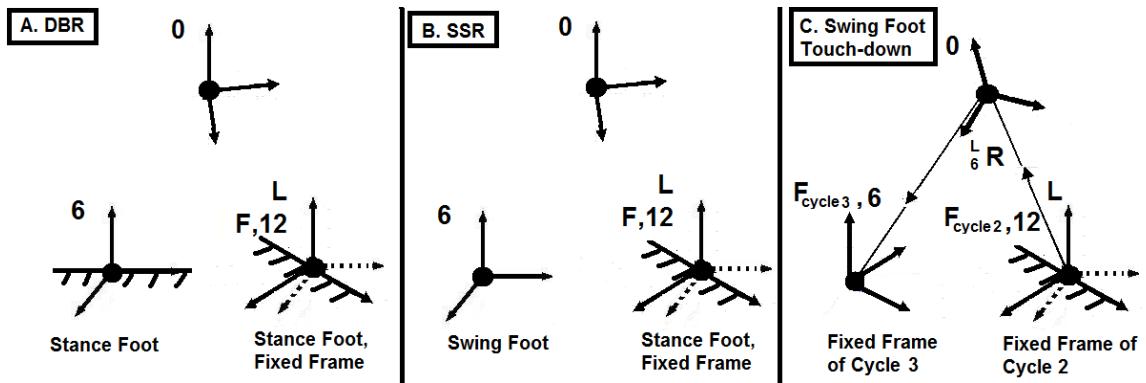


Figure 3.44: The swing foot frame, stance foot frame (fixed frame), the pelvis frame, and the local inertial frame (dotted line) in a SSR step cycle. The right-most figure is the graphical representation of the relative orientation between the swing foot frame (fixed frame of the next cycle) and the local inertial frame.

At the end of the step cycle (C. in Figure 3.44), ${}_F^L R$ is updated by the following equation:

$${}_F^L R = {}_6^L R = {}_0^L R {}_6^0 R, \quad (3.34)$$

where ${}_0^L R = {}_{12}^L R {}_0^R$ (3.35)

Note that ${}_{12}^L R$ is the ${}_F^L R$ of the current step cycle and it is used to calculate ${}_F^L R$ of the next step cycle.

From the above example, the orientation of the swing foot frame is calculated with respect to a fixed local inertia frame that is updated at the end of each step cycle, under the assumption that the stance foot is fixed to the ground throughout the walking cycle. Aside from the swing foot, the orientation of the pelvis can also be calculated with respect to the local inertia frame ${}_0^L R$. In the above example, ${}_0^L R$ is calculated in Equations 3.32 and 3.35.

In summary, this section presents the formulations of the biped kinematic equations for calculating the inaccessible states. The validation test results show that the kinematic equations can be reliably used to approximate the inaccessible states under the assumption that the stance foot remain fixed on the ground throughout the walking cycle. In the next section, we will examine the inverse problem of computing the joint positions given a reference high-level task that is associated with an end-effector position/orientation of a kinematic chain. In addition, the inaccessible states that are calculated in this section will be used in a closed-loop control scheme to control the position/orientation of the end-effectors.

3.5 The Inverse Kinematics of ABL-BI

3.5.1 Overview

The focus of this section is the kinematics control of the biped and the inverse kinematic of the kinematics chains. The kinematic control problem is simplified by treating the biped as separate kinematic chains, each with a base and an end-effector. From this kinematic setup, the kinematics control objectives of the biped can then be expressed as reference end-effector positions and orientations. Thus, the kinematic control of the biped is reduced to a series of inverse kinematic problems, which in

turn can be handled by local optimization techniques and redundancy resolution methods that are usually used for robot manipulators. This type of control approach is called the *whole body cooperative control* (WBC). The main advantage of WBC is that the biped achieves multiple high level tasks by coordinating the movement of its limbs in a cooperative manner. One of the early examples of WBC can be found in [39], where the joint motions of the arms and legs of the biped are preplanned to achieve high level walking and grasping tasks, and the pelvis and torso provide compensatory motions to maintain dynamic balance. In [40], the authors proposed a walking method that resolving the overall linear and angular momentum of the biped to joint velocities. This method, called the *Resolved Momentum Control* is based on the concept of WBC and is further applied in controllers proposed in [34,41].

A kinematic chain (or a manipulator) is considered to be redundant only if the number of actuated (controllable) joints is higher than the total number of DOFs required to perform the high-level tasks. Therefore, kinematic redundancy is determined by the assigned task rather than the robot manipulator itself. For example, a six DOF manipulator can be considered redundant if the assigned task does not involve all 6-DOFs in the operational space. Redundancy is usually used towards optimizing certain criteria. In ABL-BI, the kinematic chains are considered to be redundant and they are assigned specific roles that contribute to the overall gait of the biped. The following section will introduce methods that exploit the redundancy of the kinematic chains in different ways to fulfilled the high level tasks simultaneously.

3.5.2 The Jacobian Pseudoinverse

Recall from Section 3.3.1 that task velocities $\vec{X} \in \Re^m$ is resolved to joint space velocities $\vec{\theta} \in \Re^n$ by inverting the differential kinematic equation (Equation 3.4).

If ($m < n$), the joint space has more DOFs than the task space and the kinematic chain becomes redundant. This also implies that Equation 3.4 has an infinite number of solutions. In this case, the least square (LS) pseudoinverse J^\dagger ($n \times m$) can be used to provide a criterion (Equation 3.5) for choosing an optimal solution. For the non-redundant case where $m = n$ and the rank of J is m , the pseudoinverse J^\dagger is equivalent to the matrix inverse J^{-1} and Equation 3.4 has a unique solution.

Applying the Singular Value Decomposition (SVD) to J , we obtain:

$$J = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (3.36)$$

From Equation 3.36, J is decomposed into the following matrices:

1. U is a $m \times m$ matrix that is composed of the output singular vectors u_i . If J has a rank of r , then its range space $R(J)$ is spanned by u_i ($i = 1 \dots r$).
2. V is a $n \times n$ matrix that is composed of the input singular vectors v_i . The null-space of the Jacobian matrix $\eta(J)$ is spanned by v_i ($i = r + 1 \dots n$).
3. Σ is a $m \times n$ matrix that specifies the singular values σ of J by its first m diagonal elements. Note that ($\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$), and ($\sigma_{r+1} \dots \sigma_n = 0$).

At kinematic singularity, the rank of J is $r < m$ and $R(J)$ reduces from m to r ; at the same time, $\eta(J)$ increases from $n - m$ to $n - r$. This implies that the number of feasible DOFs in the task space is reduced. Since $\dim(R(J)) + \dim(\eta(J)) = n$, these DOFs are considered to be “transferred” to the null-space of J and are referred to as the “degenerated directions” of the end-effector.

The LS-pseudoinverse of J can be obtained by rearranging Equation 3.36. Substituting J^\dagger (in SVD form) into Equation 3.4, the joint space solution becomes:

$$\vec{\theta} = \sum_{i=1}^r \frac{1}{\sigma_i} u_i v_i^T \vec{X} \quad (3.37)$$

In the neighborhood of singularity, σ_m goes to zero and $\frac{1}{\sigma_m}$ approaches infinity. This indicates that a very large change in $\vec{\theta}$ is required to achieve a small change in \dot{X} in the u_m^T direction (Equation 3.37). Note that the joint space solution is discontinuous in the *transition* between the non-singular region and the singular region of J . Therefore, if there is a reference velocity in the degenerate direction (\dot{X}_m), then the joint space solution $\vec{\theta}$ becomes progressively large and infeasible as the kinematic chain approaches to its singular configuration.

As stated in Section 3.3.1, the LS-pseudoinverse gives preference to the accuracy of the joint space solution (ie: minimization of the error in the task space velocity) over feasibility (ie: the minimization of the joint velocities). Thus, the LS-pseudoinverse cannot be used to attain feasible solutions near singular configuration.

To overcome for this shortcoming, the damped least square (DLS) pseudoinverse is used to produce well-conditioned solutions that are based on the tradeoff between accuracy and feasibility [42–44]. The joint space solution from the DLS-pseudoinverse is given by the following equation [42]:

$$\vec{\theta} = J^\lambda \vec{X} = [(J^T W_1 J + W_2)^{-1} J^T W_1] \vec{X}, \quad (3.38)$$

which fulfills the following condition [45]:

$$\text{minimize } \{ \| \vec{X} - J \vec{\theta} \|_{W_1} + \| \vec{\theta} \|_{W_2} \}, \quad (3.39)$$

where J^λ is the DLS-pseudoinverse matrix, W_1 is a positive definite matrix that

defines the prioritization of the components in the task space vector, and $W_2 = \lambda I$ defines the tradeoff between accuracy and feasibility in the joint-space solution. A high damping in W_2 (determined by the damping factor λ) will result in smaller joint velocities near singular configurations; however, the accuracy of the end-effector velocity tracking will be compromised. Note that in our formulation of J^λ , W_1 is set to be I and the prioritization of the task space velocity components will instead be handled by a separate task priority redundancy resolution method [46] that will be presented in Section 3.5.3.

The DLS-pseudoinverse matrix can be written in SVD form and the joint space solution in Equation 3.40 becomes:

$$\vec{\theta} = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda_i^2} u_i v_i^T \vec{X} \quad (3.40)$$

The effects of λ in the DLS-pseudoinverse is analyzed from Equation 3.40:

1. Near singular configurations, the σ^2 term will go to zero and the joint velocities will increase to a maximum value at $\sigma = \lambda^2$. After that, the joint velocities will be reduced to zero by the term $\frac{\sigma}{\lambda^2}$. Therefore, high values of $\vec{\theta}$ are “damped out” by λ and the feasibility of the joint space solution is maintained near kinematic singularities. In addition, the solution from Equation 3.40 will remain continuous across the non-singular and singular regions of the kinematic chain.
2. If λ is set to zero, then the DLS-pseudoinverse reverts to the LS-psuedoinverse and the solution given by Equation 3.40 is equivalent to the solution in Equation 3.37.

The only issue that remains is the selection of the damping factor λ . In the literature, there are different methods for determining a variable λ that applies

damping only near singular configurations [45]. However, these methods often require a threshold condition and a maximum damping factor λ_{max} that are determined empirically from the actual robot or simulation.

In order to simplify the control algorithm of ABL-BI, a constant λ is selected to ensure that the task velocity tracking error remains within acceptable limits at all configurations. Note that a different λ is selected for each kinematic chain and locomotion state based on simulation tests in *Webots* (Table 3.4).

In summary, the Jacobian pseudoinverse J^\dagger and J^λ are used in the inverse kinematics algorithms that are described in Section 3.5.3. In particular, J^λ is used to compute the joint velocities for the non-redundant and redundant kinematic chains near singular configurations. Finally, it is important to note that the solution to Equation 3.37 and 3.40 is not the final solution to the inverse kinematics problem. Rather, it is used as a means to determine $\vec{\theta}_d$ for a reference \vec{X}_d (ie: determine a solution for Equation 3.1). However, the damping factor in J^λ and the numerical integration from \vec{X}_d to \vec{X}_d can introduce *operational space tracking error* between the reference and the actual task space vectors (ie: the end-effector position/orientation of the kinematic chains). Therefore, a feedback control scheme is required to reduce the operational space tracking error to zero. The full inverse kinematics algorithm of ABL-BI will be presented in Section 3.5.3.

3.5.3 The Kinematic Control of ABL-BI

This section presents the the inverse kinematics of the kinematic chains that is used in the kinematic postural control of ABL-BI. The focus of this section is a closed loop control algorithm that is used to track the high level objectives of the kinematic chains. This control algorithm utilizes the kinematic chains that are defined in Section 3.3.2,

the Jacobian pseudoinverse in Section 3.5.2, and the kinematic redundancy resolution methods that are introduced in this section. Note that the derivation of the Jacobian matrices for the high level tasks of each kinematic chain is presented in Appendix E. Also, refer back to Section 3.3.2 for the high level tasks that are assigned to each kinematic chain.

The Control Architecture of the Kinematic Controller

The purpose of the kinematic postural controller is to track a set of reference trajectories that defines the high-level motion of the biped (ex: reference COG velocity and pelvis angular velocity). The task space velocities of each kinematic chain can be resolved into joint velocities by the inverse differential kinematic equations in Appendix E. However, we cannot use the computed joint space velocities to obtain an exact joint space solution to Equation 3.1 due to the following sources of error:

- *Numerical error:* If the damping factor λ in the DLS-pseudoinverse Jacobian J^λ is non-zero, it will introduce inaccuracies in the task velocity tracking. This error, combined with the numerical errors that are contributed by the integration of $\vec{\dot{\theta}}$ to $\vec{\theta}$ and numerical filtering will generate operational space errors in the task space positions.
- *External Disturbances and Modelling Error:* External disturbances on the task space variables such as the COG position (ex: a sudden impact on the biped), and modelling errors (ex: the estimation of the local COG positions of the actual robot that are used in the *Webots* model) will cause deviations between the reference and the actual task space position.

As shown in Figure 3.45, we can implement a closed-loop version of the inverse differential kinematics in order to reduce the operational space error and track the

reference task space positions ${}^{ref}\vec{X}_i$ (note: assuming that $i = 1..p$, and p is the number of high-level tasks that are described in Section 3.3.2). The reference task space velocities ${}^{ref}\dot{\vec{X}}_i$ are calculated from $\bar{K}_i \cdot e_i$, where \bar{K}_i are diagonal proportional gain matrices with diagonal elements K_X , K_Y , and K_Z (proportional gains in the X, Y, and Z direction) and $e_i = {}^{ref}\vec{X}_i - {}^{Act}\vec{X}_i$ are the operational space errors. Note that the actual task space positions ${}^{Act}\vec{X}_i$ are either directly measured in the *Webots* simulation or calculated from the forward kinematics equations (Section 3.4) in the actual robot.

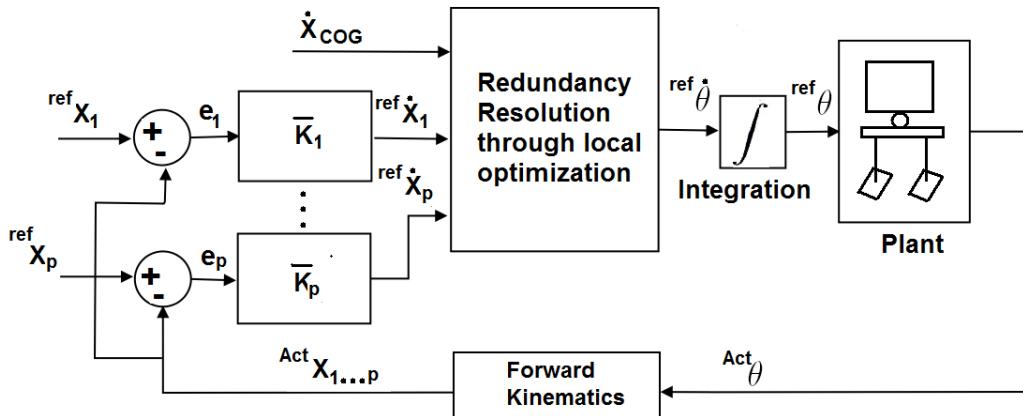


Figure 3.45: The control architecture of the kinematic postural controller.

In order to perform stable walking, the biped is required to achieve multiple tasks by the coordinated motion of its links. For example, in order for the swing foot to perform touch-down successfully, its position and orientation must be controlled simultaneously by the joints in the swing leg. Therefore, rather than finding an optimal solution for a single high level task that is based on local optimization techniques such as the DLS-pseudoinverse, the problem now extends to finding a joint space solution that satisfies all the high level tasks of a given kinematic chain. As shown in Figure 3.45, the high level tasks (1 to p) are resolved by the “Redundancy Resolution block”

to joint velocities. The “Redundancy Resolution block” represents two approaches that are commonly used in the kinematic control of redundant manipulators: 1) Task Space Augmentation, and 2) Task Prioritization.

The Redundancy Resolution of the Kinematic Chains

In ABL-BI, the kinematic chains are considered to be redundant with respect to the given high level tasks. As a result, the kinematic chain has an infinite number of solutions for a given reference end-effector position/orientation. In addition, the joints can generate internal motions within the kinematic chain without affecting the position/orientation of the end-effector. These joint motions belong in the null space of J that satisfy $J \cdot \theta = 0$. One of the main advantages of internal motions is that the kinematic chain can be reconfigured to avoid singularities and achieve additional task space objectives while tracking the reference end-effector of the primary task. In ABL-BI, the kinematic singularity problem is resolved by DLS-pseudoinverse; therefore, the redundancies in ABL-BI is primarily used to achieve multiple high level tasks in a systematic way through the application of different redundancy resolution methods.

In **DB**, we neglect the control of the vertical component of the COG (YG_{DB}) and instead focus on controlling the more crucial horizontal components of the COG within the support polygon of the biped. It is observed from the simulation results that although YG_{DB} is not controlled, it only decreases from its initial position by 3 to 4 cm due to the pitching motion of the legs and the pelvis during weight-shift. Furthermore, since YG_{DB} is controlled in the subsequent SS phase, the changes in the YG_{DB} position in DB is corrected and YG_{DB} moves back to its “nominal” position during SS. Therefore, in normal walking, we can safely ignore the control of YG_{DB} to 1) simplify the motion planning of the reference COG position, and 2) introduce more flexibility to the joint space solution for the horizontal

components of the COG position by relaxing the constraint in the vertical component.

For the first task in **DB**, the COG velocity (a 2-DOF task that only includes the X and Z direction) is controlled by six joints in each of the Left Leg Chain and Right Leg Chain and an additional *TorsoRoll* joint in the pelvis. Combining the two leg chains together (Equation E.2, E.4, and E.7), the stance legs have a total of 4 DOFs in the task space and 13 DOFs in the joint space. Since the vertical component of the COG is neglected, the Jacobian matrix JG_{DB} in Equation E.7 is modified to only account for $\dot{X}G_{DB}$ and $\dot{Z}G_{DB}$ in the following Jacobian matrix JG_{DB_XZ} :

$$JG_{DB_XZ} = \begin{bmatrix} JGL_{1,1} & \dots & JGL_{1,6} & 0 & \dots & 0 & JGL_{1,13} \\ JGL_{3,1} & \dots & JGL_{3,6} & 0 & \dots & 0 & JGL_{3,13} \\ 0 & \dots & 0 & JGR_{1,7} & \dots & JGR_{1,12} & 0 \\ 0 & \dots & 0 & JGR_{3,7} & \dots & JGR_{3,12} & 0 \end{bmatrix}_{2 \times 13} \quad (3.41)$$

The reference COG trajectory is calculated by a *dynamic stable gait generator* that will be described in Chapters 4 and 5.

For the second task in DB, the angular velocity of the pelvis (3 DOFs) is controlled by six joints in the Left Leg and Right Leg Chains. Combining the two chains together (Equations E.19, E.20, and E.21), the stance legs have a total of 6 DOFs in the task space and 13 DOFs in the joint space. Note that the reference pelvis orientation is set to zero and the reference angular velocity of the pelvis is

calculated from the closed loop kinematic controller (Figure 3.45).

In **SS**, the Stance Leg Chain is a redundant kinematic chain with seven joints that controls two 3-DOF tasks: the COG velocity and the pelvis angular velocity. Again, the reference COG trajectory is generated by a high level motion planner and the reference pelvis orientation is set to zero. On the other hand, the Swing Leg Chain has six joints that are used to control the linear and angular velocities of the swing foot (both 3-DOF tasks). Although the Swing Leg Chain is considered to be redundant with respect to each individual swing foot tasks (with three redundant DOFs), a unique joint space solution is reached if we combine the Jacobian matrices of the swing foot tasks into a single augmented matrix (a 6×6 matrix). This approach, called the *Task Space Augmentation*, fully uses the redundant joints to achieve additional functional tasks.

Assuming SSL, the swing foot linear velocity can be manipulated along with the swing foot angular velocity by the joints in the Swing Leg Chain. The Jacobian matrices for each task Jv_{SSL} and $J\omega_{SSL}$ are combined to form the augmented Jacobian $J_{SwingSSL}$:

$$J_{SwingSSL} = \begin{bmatrix} Jv_{SSL} \\ J\omega_{SSL} \end{bmatrix}_{6 \times 6} \quad (3.42)$$

In addition, the augmented task space vector is obtained from the concatenation of ${}_{12}^6\dot{X}$ and ${}_{12}^6\omega$:

$$\vec{\dot{X}}_{SwingSSL} = \begin{bmatrix} {}^6\dot{X} \\ {}^{12}\dot{X} \\ {}^{12}\omega \end{bmatrix}_{6 \times 1}. \quad (3.43)$$

The joint space solution of the swing foot is given by the following equation:

$$\vec{\dot{\theta}}_{7\dots12} = J_{SwingSSL}^\lambda \cdot \vec{\dot{X}}_{SwingSSL} \quad (3.44)$$

In Equation 3.44, it is possible for $J_{SwingSSL}$ to become ill-conditioned and the joint space solution to be very large. In addition to the kinematic singularities that are found in the individual components (Jv_{SSL} and $J\omega_{SSL}$) of the augmented Jacobian matrix, $J_{SwingSSL}$ can become singular if the task space objectives ${}^6\dot{X}$ and ${}^{12}\omega$ are in conflict with one another. This additional type of singularity is called *algorithmic singularity* and it can occur even if the end-effector of the kinematic chain is not in a singular position/orientation (ie: Jv_{SSL} and $J\omega_{SSL}$ is full rank). For this reason, DLS-pseudoinverse is used in Equation 3.44 to provide singularity robustness to the joint space solution. By combining Jv_{SSL} and $J\omega_{SSL}$ the into a single matrix, the task augmentation method effectively places the two corresponding high level tasks in the same priority level. In other words, if conflict occurs between ${}^6\dot{X}$ and ${}^{12}\omega$, the DLS-pseduoinverse will compute a compromised solution, but at the cost of lower tracking accuracy of *both tasks*.

In the current case however, it is appropriate to apply task space augmentation since it is equally crucial to control both ${}^6\dot{X}$ and ${}^{12}\omega$ in order to ensure successful lift-off/touch-down and accurate swing foot placement. The reference swing foot orientation is set to zero in order to keep the swing foot flat with respect to the ground. On the other hand, the reference swing foot position (ie: the swing leg

motion) is generated from a trajectory planner that will be described in Chapter 4. As shown in the two-step initial simulation test in Figure 3.54, it is difficult to plan a reference swing foot trajectory such the pelvis orientation can be kept at zero throughout SS. Generally, the swing foot orientation might deviate slightly from zero at certain parts of the swing phase. However, this error is acceptable as long as that the swing foot orientation can avoid ground interference at lift-off and regain ground contact at touch-down.

For the simultaneous control of the COG velocity and pelvis orientation in 1) the *Combined Stance Leg Chain* in **DB**, and 2) the *Stance Leg Chain* in **SS**, a *task prioritization method* is used to assign priority to each task. In this hierachal framework, the primary task will always be fulfilled by the joint space solution and the secondary task is only achieved if it does not interfere with the primary task. If the secondary task cannot be achieved, then its operational space error will be minimized.

In ABL-BI, the most crucial control objective is to maintain dynamic balance and foot contacts with the ground. One of the key state variables that determine the stability and overall motion of the biped is the COG position; therefore, COG tracking is assigned as the primary task. If the primary task is satisfied, then the next concern would be to maintain an anthropomorphic posture and maximize the kinematic workspace of the swing leg (please refer to Section 3.1). Therefore, the orientation of the pelvis is assigned as the secondary task. The priority assignment for each high level task in the kinematic chains is summarized in Table 3.4

The main idea of the task prioritization method is that the joint space solution of the secondary task is projected to the nullspace of the Jacobian matrix that corresponds to the primary task. Thus, the secondary task is achieved by the internal

motion of the kinematic chain that does not affect the primary task. At the same time, the optimal joint space solution for the primary task is calculated independently from the secondary task. Although there are different formulations of the task prioritization method, we will use the formulation proposed by Chiaverini et al. [46] in the kinematic controller of ABL-BI. The key equations of the formulation will be presented below. For detailed derivations, please refer to [46]. The Jacobian matrix that corresponds to the primary task of the stance leg chain in DB and SS (COG velocity) is denoted as J_1 and \dot{X}_1 respectively. Likewise, the Jacobian matrix and task velocity vector of the secondary task (angular velocity of the pelvis) is J_2 and \dot{X}_2 .

The differential kinematic equations of the primary and secondary task are given by the following equations:

$$\dot{X}_1 = J_1 \cdot \dot{\theta} \quad (3.45a)$$

$$\dot{X}_2 = J_2 \cdot \dot{\theta} \quad (3.45b)$$

The minimum norm joint space solution of the primary task is obtained from the LS-pseudoinverse of J_1 :

$$\dot{\theta} = J_1^\dagger \cdot \dot{X}_1 \quad (3.46)$$

Note that (Equation 3.46) is a specific joint space solution that is constrained by the minimum norm condition. A general form of the joint space solution is computed as [46]:

$$\dot{\theta} = J_1^\dagger \dot{X}_1 + \left(I - J_1^\dagger J_1 \right) \dot{\theta}_0 \quad (3.47)$$

The general solution $\dot{\theta}$ has two portions: 1) the minimum norm solution $J_1^\dagger \dot{X}_1$, and 2) additional nullspace joint velocities that do not affect the end-effector of the

kinematic chain. The nullspace joint velocity is obtained by projecting an arbitrary joint velocity vector $\dot{\theta}_0$ to the nullspace of J_1 by a “projector term” $(I - J_1^\dagger J_1)$. Note that the internal motion of the kinematic chain, represented by $\dot{\theta}_0$, is used to fulfill a secondary high level task.

The nullspace velocity vector $\dot{\theta}_0$ is calculated by first substituting Equation 3.47 to the differential kinematics of the secondary task in Equation 3.45b:

$$\dot{X}_2 = J_2 J_1^\dagger \dot{X}_1 + J_2 (I - J_1^\dagger J_1) \dot{\theta}_0 \quad (3.48)$$

Multiplying the LHS and RHS of Equation 3.48 by $J_2 J_2^\dagger$ and simplifying, we get:

$$J_2^\dagger \dot{X}_2 = J_1^\dagger \dot{X}_1 + (I - J_1^\dagger J_1) \dot{\theta}_0 \quad (3.49)$$

From Equation 3.49, we can solve for $\dot{\theta}_0$:

$$\dot{\theta}_0 = (I - J_1^\dagger J_1)^\dagger (J_2^\dagger \dot{X}_2 - J_1^\dagger \dot{X}_1) \quad (3.50)$$

From the matrix properties of the LS-pseudoinverse: 1) $J^\dagger = J^\dagger J J^\dagger$, and 2) $(I - J_1^\dagger J_1)^\dagger = (I - J_1^\dagger J_1)$, Equation 3.50 is simplified to the following equation:

$$\dot{\theta}_0 = (I - J_1^\dagger J_1) J_2^\dagger \cdot \dot{X}_2 \quad (3.51)$$

From Equation 3.51, we determine the nullspace solution $\dot{\theta}_0$ that satisfies the secondary task \dot{X}_2 . Substituting $\dot{\theta}_0$ to the general solution of the primary task in Equation 3.47, and noting the idempotence property of the projector term $N_1 = I_n - J_1 J_1^\dagger$, we get:

$$\dot{\theta} = J_1^\lambda \cdot \dot{X}_1 + N_1 \left(J_2^\lambda \cdot \dot{X}_2 \right) \quad (3.52)$$

$$\text{where } N_1 = I_n - J_1 J_1^\dagger \quad (3.53)$$

The task prioritization can be summarized in Equation 3.52:

1. The optimal DLS joint space solution for each of the high level tasks are calculated independently from all other tasks.
2. The joint space solution of a given task is projected to the nullspace of the higher priority tasks.

As an example, we apply the task prioritization method to the *Stance Leg Chain* in SSL and obtain the following equation:

$$\dot{\theta}_{1\dots 6,13} = JGlc_{DBL}^\lambda \cdot \dot{X}G + N_1 \left({}^6J_{\omega 0}^\lambda \cdot {}^6\omega_0 \right) \quad (3.54a)$$

$$N_1 = I_7 - JGlc_{DBL} \cdot JGlc_{DBL}^\dagger \quad (3.54b)$$

Notice that in Equation 3.54a, DLS-pseudoinverse is used to invert $JGlc_{DBL}$ and ${}^6J_{\omega 0}$ to ensure that the solution is robust to kinematic singularity. However, applying the DLS-pseudoinverse in N_1 can compromise the accuracy of the nullspace projection to $JGlc_{DBL}$. Therefore, if $JGlc_{DBL}$ is near a kinematic singularity, the joint solution of the secondary task might interfere with the tracking of the primary task.

For future developments of the robot controller, the task prioritization framework can be extended to account for a third high level task:

$$\dot{\theta} = J_1^\lambda \cdot \dot{X}_1 + N_1 \left(J_2^\lambda \cdot \dot{X}_2 \right) + N_1 N_{12} \left(J_3^\lambda \cdot \dot{X}_3 \right) \quad (3.55a)$$

$$N_{12} = I_n - J_{12} J_{12}^\lambda \quad (3.55b)$$

Note that in this case, the third task is projected to the nullspace intersection of J_1 and J_2 by N_{12} . The projector N_{12} is a function of J_{12} , the augmented Jacobian matrix of J_1 and J_2 . Therefore, if algorithmic singularity occur between the primary and secondary task, then the third task will affect the primary task. For this reason, the joint space solution of the third task is “filtered” by an additional projector term N_1 so that it will not affect the primary task.

From the initial simulation tests 3.5.4, it is observed that kinematic and algorithmic singularity can largely be avoided if the reference high-level task trajectories are planned according to the workspace constraints that are described in Section 3.2.2. Therefore, in normal walking, the damping factor for each pseudoinverse Jacobian can be set to relatively low values (Table 3.4).

Table 3.4: The damping factor and priority levels of the high level tasks in the kinematic chains.

Kinematic Chain	High-level tasks	Damping factor (λ)	Priority level
Combined Leg Chain DBL/DBR	COG position (X,Z direction)	1	primary
	Pelvis orientation	5	secondary
Stance Leg Chain SSL/SSR	COG position	1	primary
	Pelvis orientation	5	secondary
Swing Leg Chain SSL/SSR	Swing Foot position	5 (augmented matrix)	primary
	Swing Foot Orientation	5 (augmented matrix)	primary

In summary, this section presented the closed loop kinematic controller of ABL-BI. The objective of this controller is to enable high level postural and motion control of the biped. The kinematics of the biped is first divided into separate kinematic chains.

Next, manipulator redundancy resolution methods are applied to each kinematic chain in order to compute a feasible and accurate joint space solution for the given high level tasks.

3.5.4 Initial Simulation Test Results

In order to validate the performance of the closed loop kinematic controller and redundancy resolution methods that are presented in this section, a series of initial test were performed in *Webots*. Additionally, these tests are used to:

1. Check whether or not the secondary task (zero pelvis orientation in DB and SS) can be fulfilled along with a reference COG trajectory by the internal motions (nullspace velocities) of the kinematic chains. If not, we can then determine the effects of the pelvis orientation tracking errors on the overall walking gait.
2. Check whether the swing foot position and orientation tasks (both first priority level tasks) are fulfilled simultaneously in SS. If there are slight errors in position and orientation, we then focus on whether or not they can detrimentally affect the lift-off/touch-down of the swing foot. As mention in previous sections, the position and orientation of the swing foot is important in ensuring ground clearance at lift-off and regaining foot contact (and hence maintaining dynamic balance) at touch-down.
3. Plan the reference COG and swing leg trajectories such that the *Stance Leg Chain* and the *Swing Leg Chain SS* do not come into conflict with one another. Since the joint velocities of each chain are calculated independently from each other, only the conflicts *within the task space of each kinematic chain* are resolved by the techniques mentioned in this Section. Thus, we must rely on the careful planning of the COG and swing leg positions in order to avoid conflicts between the chains.

4. Manually tune the closed loop gains (\bar{K}_i) in Figure 3.45) for each high level tasks. In addition, the tests are used to select a constant damping factor λ_i for each task. The damping factor is selected to 1) damp out the end-effector oscillations that occur near singular configurations (only observed for certain reference COG trajectories in DB), and 2) ensure that the operational space error falls within acceptable bounds. Note that the constant λ values are organized in Table 3.4 and the diagonal elements of \bar{K}_i (which are the proportional gains K_X , K_Y , and K_Z) are presented in Tables 4.1 and 4.1.

COG Oscillatory Motion Test

The purpose of the following tests is to validate the joint space solution computed by the task periodization method. Given a reference cyclic COG motion in **DB** (Figure 3.46) and setting the pelvis orientation to zero, the actual COG velocity is reconstructed from the actual joint velocities and the COG Jacobian matrix. The quality of the joint space solution is then evaluated from the tracking accuracy of the reference COG velocity (Figure 3.47 and 3.49). Note that in DB, the vertical component of the COG is *not* controlled by the kinematic controller.

The results of the COG oscillatory test in the lateral plane are given by the following plots:

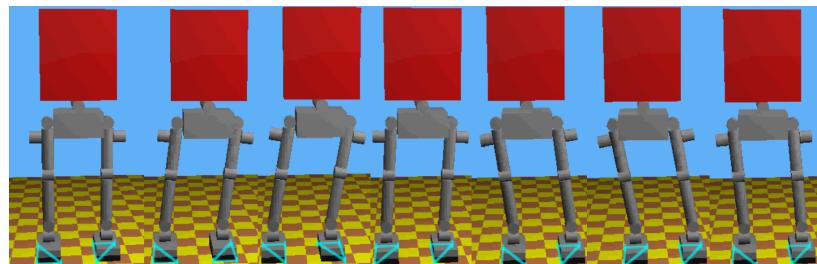


Figure 3.46: Initial inverse kinematics test: oscillatory COG motion in the lateral plane.

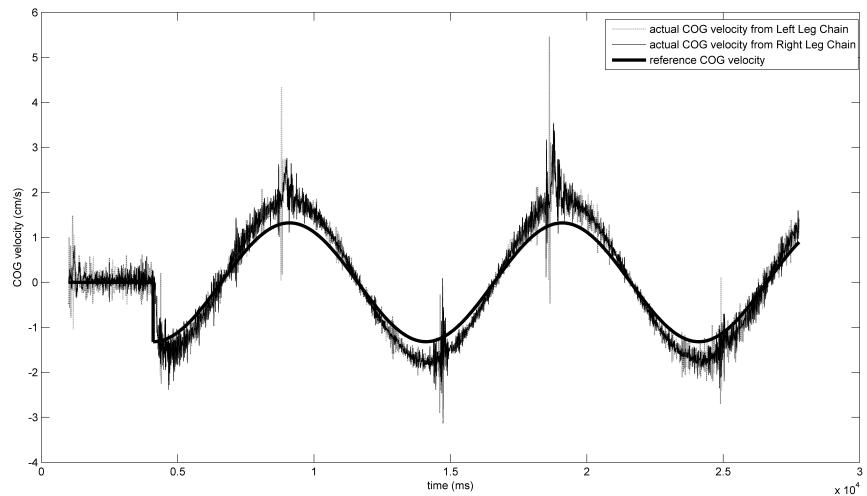


Figure 3.47: The comparison between the actual and reference COG velocity X.

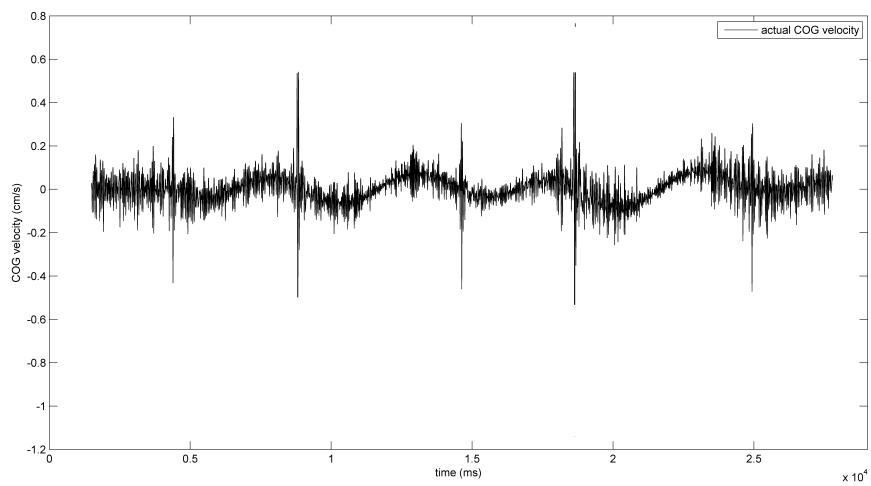


Figure 3.48: The actual COG velocity Y.

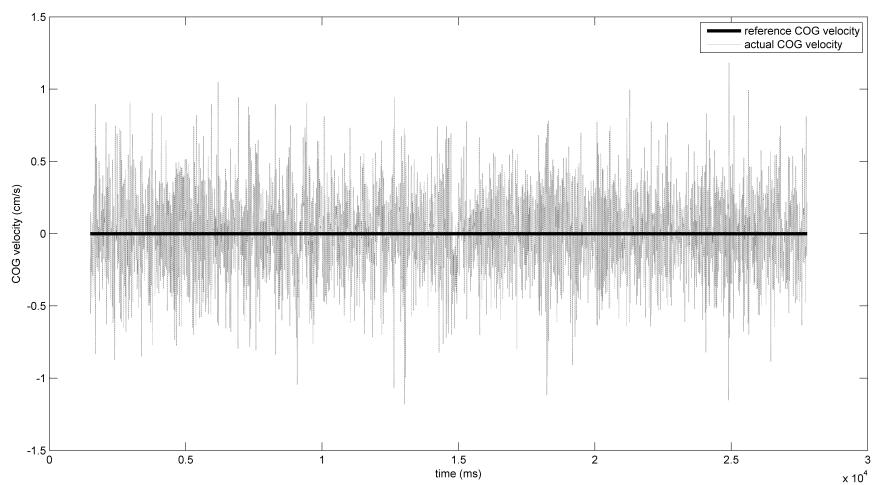


Figure 3.49: The comparison between the actual and reference COG velocity Z.

From the above results, we can observe the following:

1. The actual COG velocity (generated from the joint space solution) tracks the reference COG velocity with high accuracy in the X and Z direction.
2. The COG velocity in the Y direction is bounded even though it is not controlled by the kinematic controller.
3. The pelvis orientation remains at zero throughout the cyclic motion of the biped. Therefore, the secondary task of the *combined stance leg chain* is satisfied.
4. The COG velocity generated by the *Left Leg Chain* is similar to the velocity that is generated by the *Right Leg Chain* (Figure 3.51). Therefore, the stance legs chains in DB manipulates the COG in a cooperative manner.

The results of the COG oscillatory test in the sagittal plane are given by the following plots:

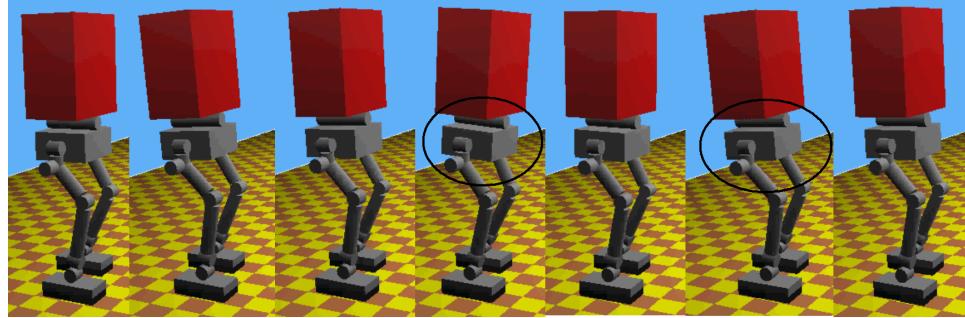


Figure 3.50: Initial inverse kinematics test: oscillatory COG motion in the sagittal plane. As shown in the fourth and the sixth frame, the pelvis orientation with respect to the stance foot changes about the pitch (X) axis.

From the COG test results in the sagittal plane, we can observe the following:

1. The actual COG velocity generated by the stance leg chains tracks the reference COG velocity cooperatively.

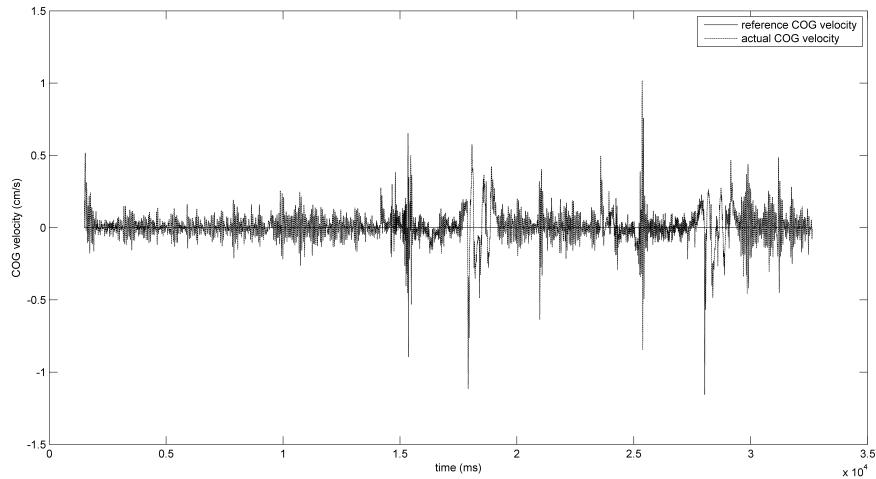


Figure 3.51: The comparison between the actual and reference COG velocity X.

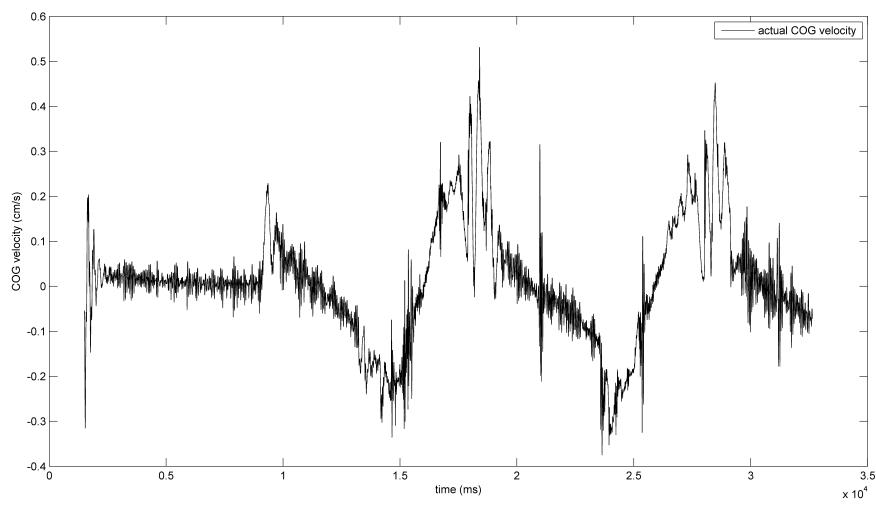


Figure 3.52: The actual COG velocity Y.

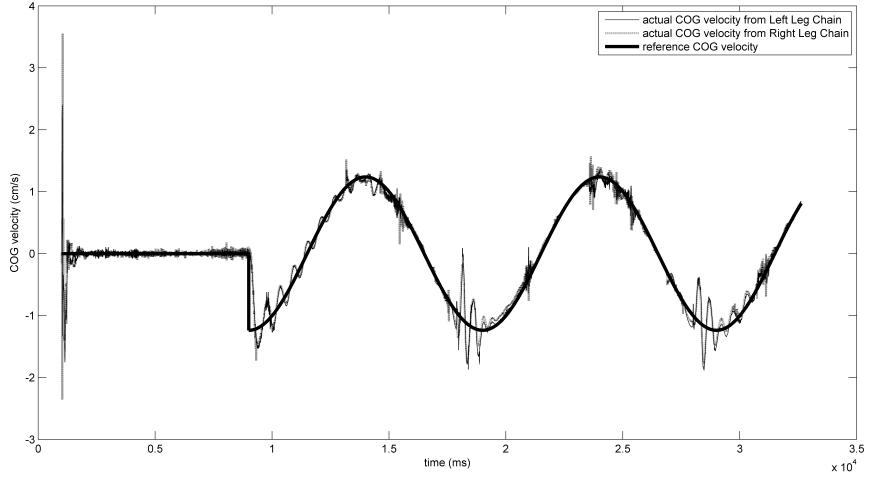


Figure 3.53: The comparison between the actual and reference COG velocity Z.

2. The pelvis orientation does not remain at zero about the X axis at certain phases of the cyclic motion. Therefore, the secondary task of the *combined stance leg chain* is not satisfied by the nullspace velocities.

The Step Cycle Test

In order to test the effectiveness of the kinematic controller and redundancy resolution methods in a walking biped, the COG and the swing foot position are given reference trajectories that enable the biped to complete a step cycle (starting at DBL and ending at DBR) in a stable manner (Figure 3.54). Note that the closed loop control gains \bar{K}_i are manually tuned from the Step Cycle Test and are presented in Tables 4.1 and 5.2. The results of the step cycle test are presented in the following plots:

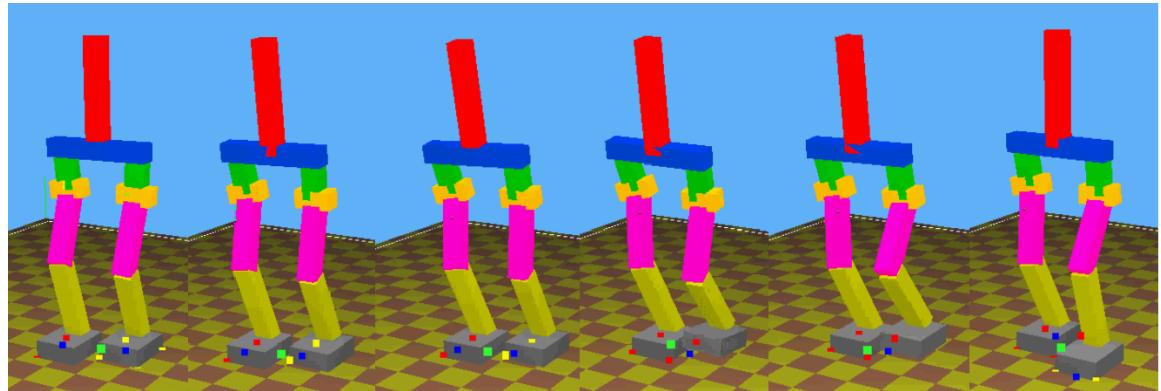


Figure 3.54: Initial inverse kinematics test: the step cycle test.

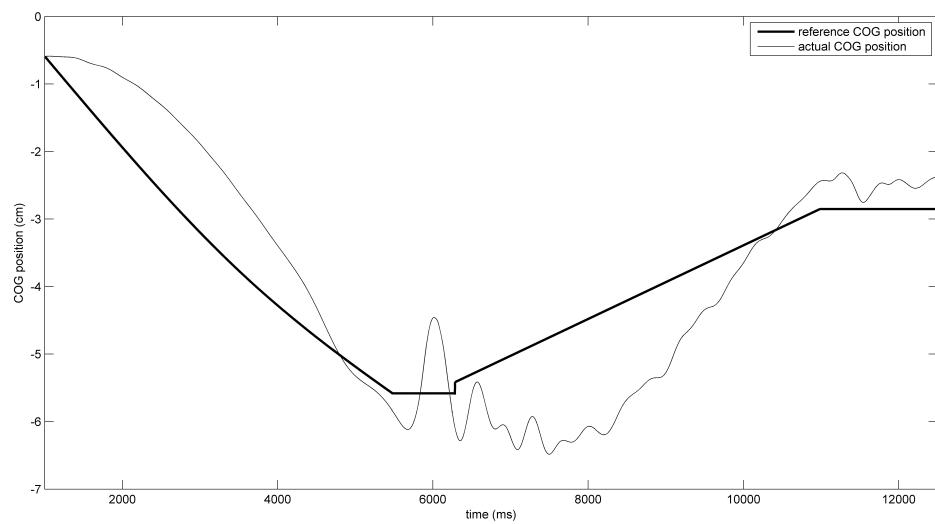


Figure 3.55: The comparison between the actual and reference COG position X.

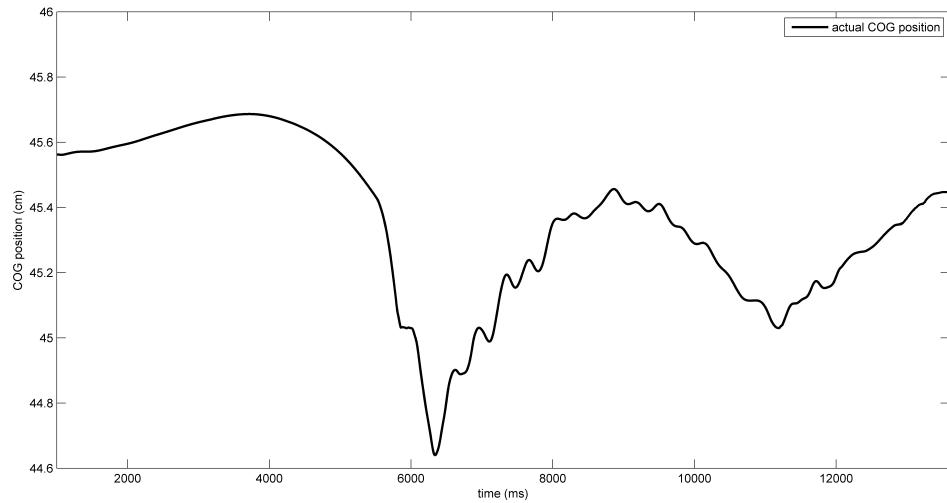


Figure 3.56: The actual COG position Y.

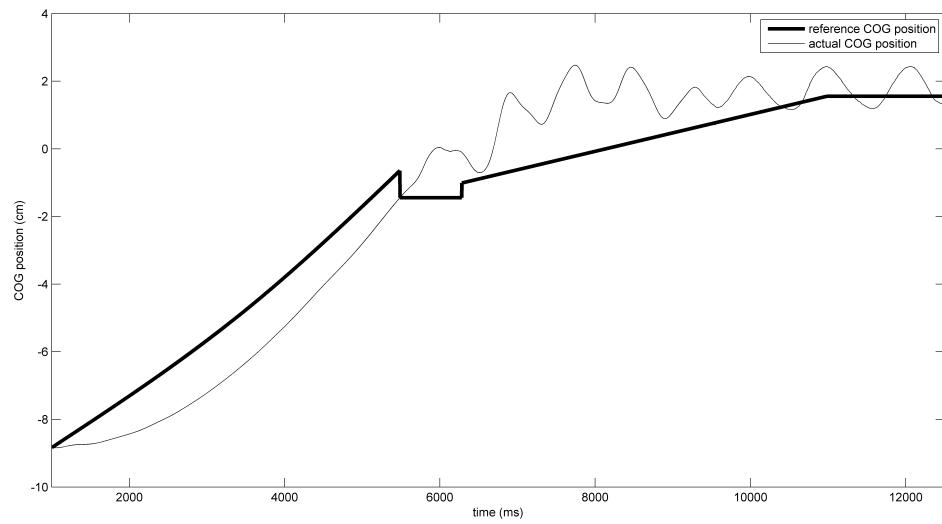
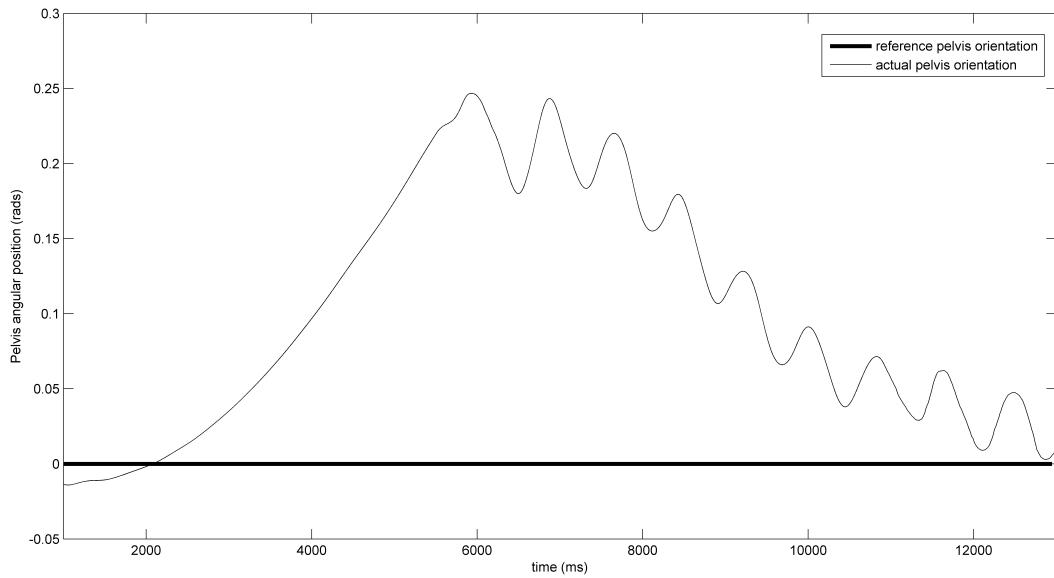
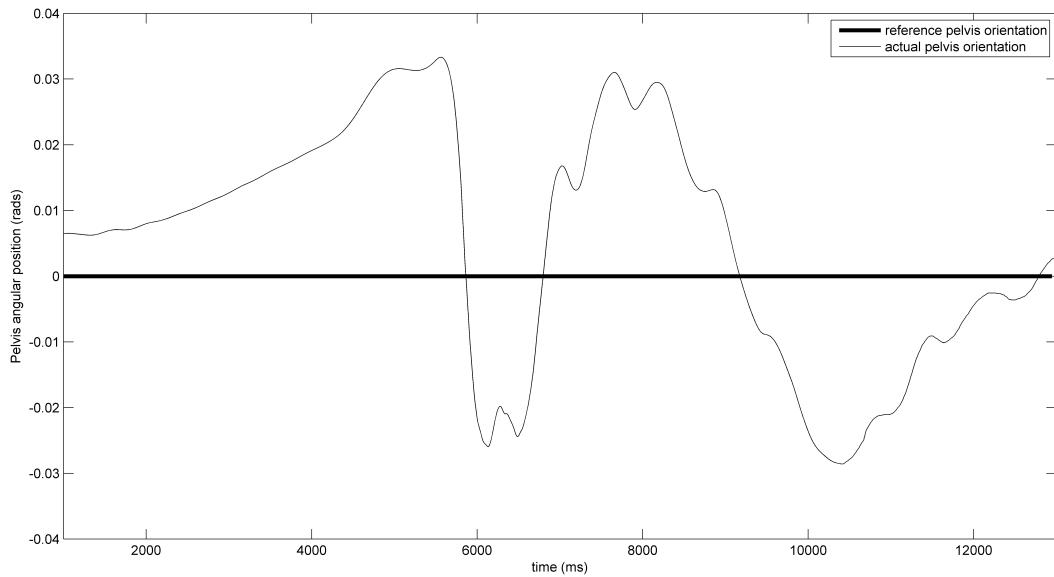


Figure 3.57: The comparison between the actual and reference COG position Z.



(a) Pelvis orientation about the X axis.



(b) Pelvis orientation about the Z axis.

Figure 3.58: Pelvis Orientation of the Step Cycle Test

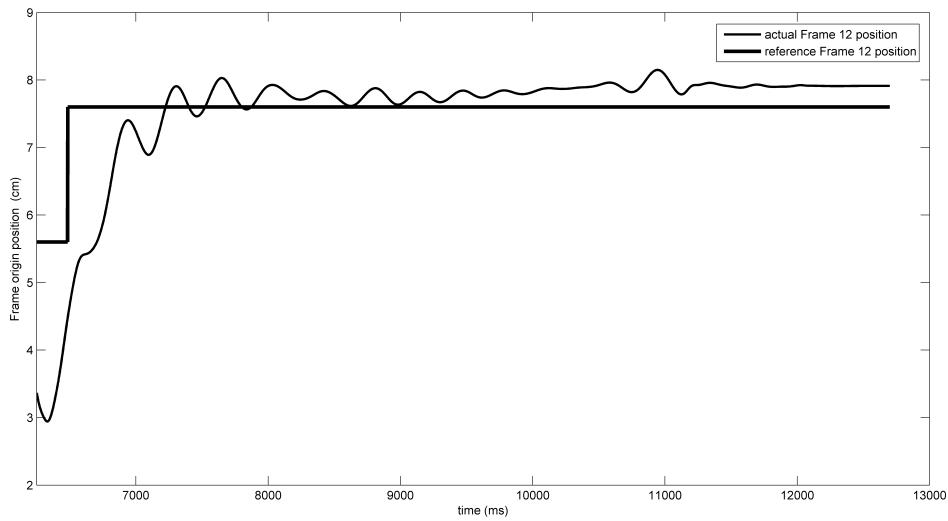


Figure 3.59: The actual and reference swing foot position X.

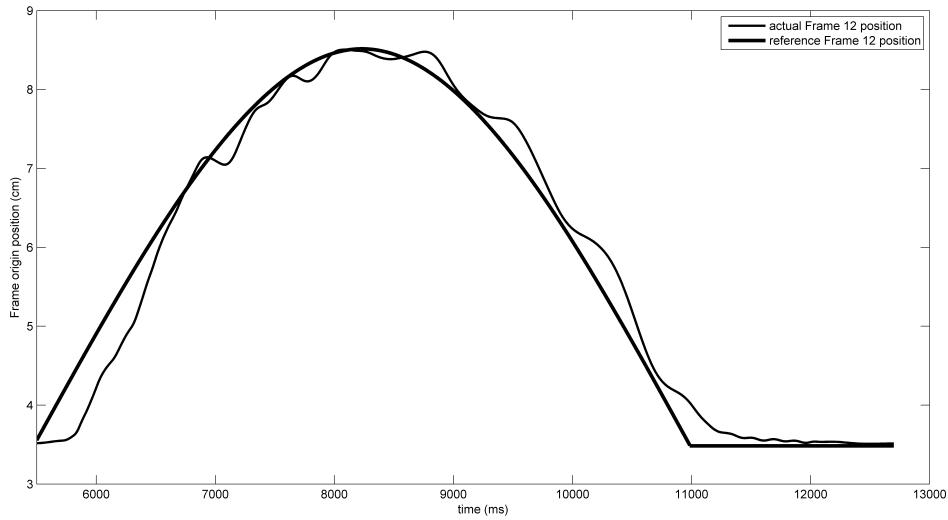


Figure 3.60: The actual and reference swing foot position Y.

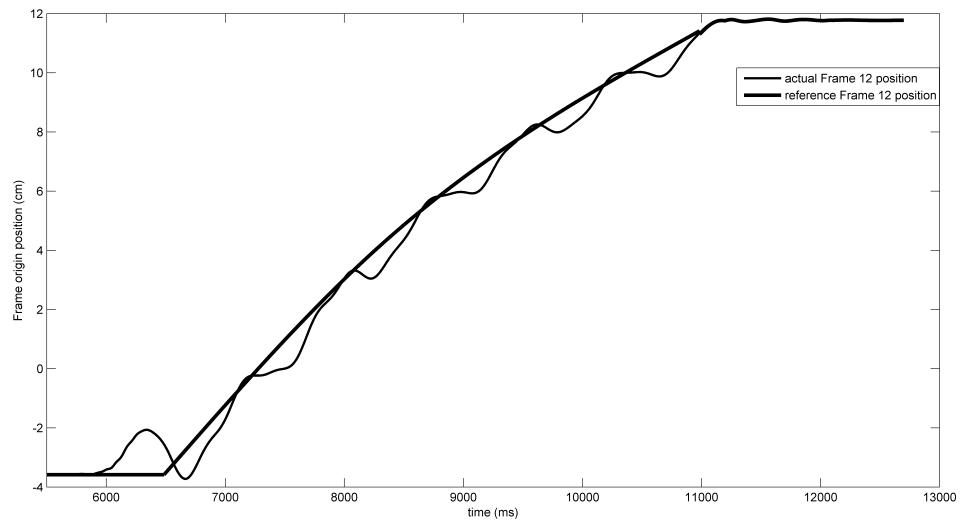
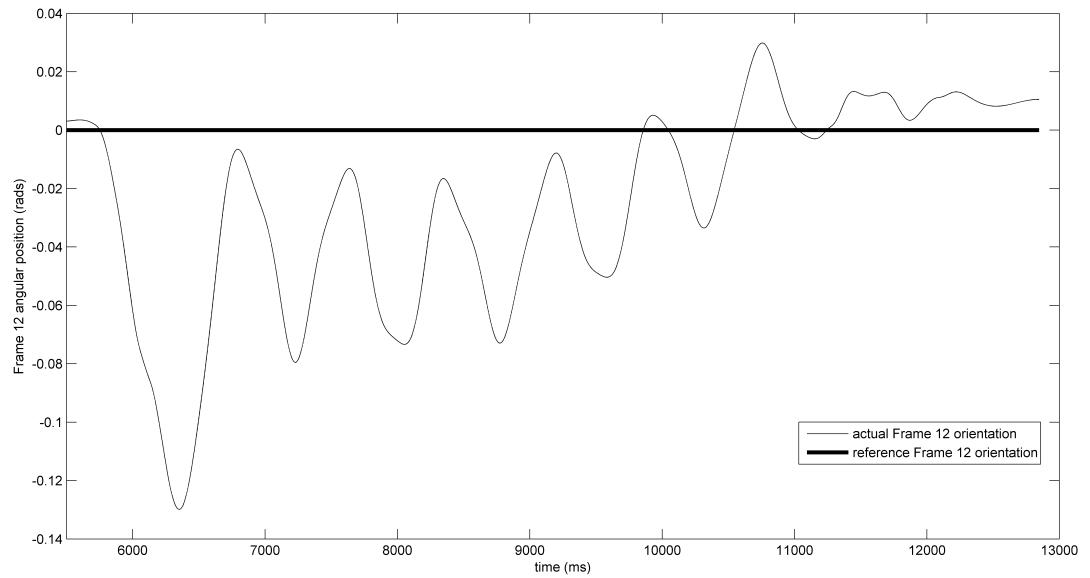
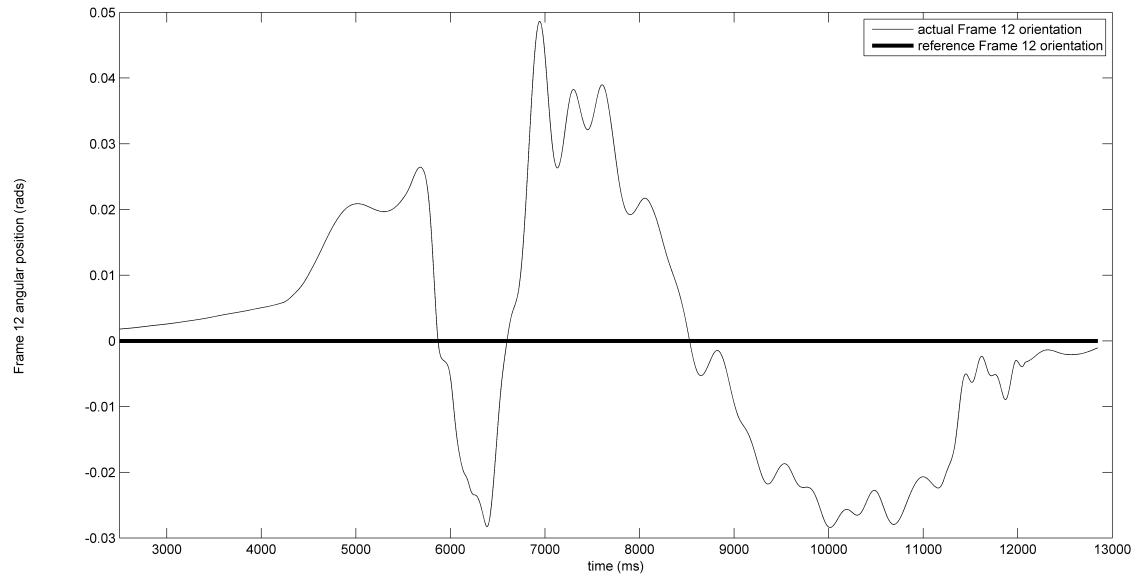


Figure 3.61: The actual and reference swing foot position Z.



(a) Swing foot orientation about the X axis.



(b) Swing foot orientation about the Z axis.

Figure 3.62: Swing foot orientation of the Step Cycle Test

From the above simulation results of the step cycle test, we observe the following:

1. The actual COG position does not track the reference position exactly in DB and SS. The tracking error is contributed partly by the top layer controller that applies dynamic stability constraints on the reference COG velocity. In addition, low controller gains and delays introduced by numerical filtering also contribute to the COG tracking error (to be discussed in further details in Chapter 4 and 5). However, as noted in previous sections, COG tracking errors are acceptable as long as they do not cause the COG to drift outside of the support polygon. At state transition, the COG position is perturbed by the angular momentum about the COG that is generated by the swing leg during lift-off (at about 6000 ms). The effects of this disturbance is shown by the deviations in the COG position immediately after the state transition. In addition, the oscillations in the COG motion is caused by the GRF from the ground contacts (modelled as spring-dampers in *Webots*). However, these errors are gradually reduced by the kinematic controller in SS.
2. In DB, the pelvis orientation is unable remain at zero about the pitch axis as the biped performs weight shift in the Z direction. In other words, the kinematic controller cannot achieve the secondary task of the *combined leg chain* and the error grows to a maximum value of 14° at the end of DB. However, the error in the pelvis orientation do not have significant effects on the walking gait and the dynamic stability of the biped.
3. As previously shown in Figure 3.5.4, the task space error in the pelvis orientation about the X axis is corrected in SS. Unlike DB, the *Stance Leg Chain* in SS can fulfill both the primary and secondary high level tasks. Therefore, the kinematic controller can generate angular velocity to reduce the pelvis orientation error from 14° at the start of SS to 1.4° towards the end of SS. In addition, the error

in the vertical component of the COG (the uncontrolled COG component in DB) is also reduced to within 0.5 *cm* of its initial value at the end of SS (Figure 3.56).

4. As shown in Figure 3.59, 3.60 and 3.61, the actual swing foot position tracks the reference position very accurately (maximum error is less than 1 *cm* in all three directions). This enables the biped to perform lift-off and touch-down successfully and allows the swing foot to reach its target position. The results of the swing foot orientation (Figure 3.62) indicate that there is a large deviation (a maximum error of about 6° about the X axis) between the actual and reference swing foot orientation at lift-off (around 6000 ms). However, once the swing foot clears from the ground and the disturbance at the state transition between DB and SS settles, the controller gradually reduces the error in the swing foot orientation. At the end of SS (about 12500 *ms*), the relative orientation between the swing foot and the ground is close to zero (less than 1° after 12500 *ms*). Therefore, the biped can regain ground contact upon foot landing.

5. In SS, the high level tasks in the swing leg chain do not come into conflict with the tasks in the stance leg chain. This is due to the fact that the reference trajectories of the high level tasks are planned within the operational workspace that is described in Section 3.2.2.

3.6 The Simplifying Models in Bipedal Walking

3.6.1 Overview

Bipeds are complex varying structure systems with a large number of DOFs. For example, ABL-BI is composed of 8 rigid links (excluding the links that are used to model the motors) that are connected by 13 actuated joints. Therefore, the dynamic

equation of the biped is highly nonlinear and coupled; it is difficult to determine the causal relationship between the control input (ex: joint torques) and high level motions (ex: COG motion). For this reason, in some previous walking controllers, the dynamics of the robot is represented by simplified models. For example, offline trajectory generation methods in [39, 47] use models that neglect the dynamics of small masses and DOFs that have insignificant roles in the overall motion. The major links of the robot (ex: torso, thigh, foot) are usually modelled either as concentrated point masses or rigid bodies with distributed mass, and the joints are modelled as rigid 1-DOF revolute joints. From the dynamic properties of the model, dynamically stable walking trajectories are generated based on the ZMP criterion. However, since the model and the controller consider the *full dynamics* (ie: the dynamic properties of each individual link) of the robot, it is too computationally expensive to be implemented in real-time.

At the other end of the spectrum, a simple planar biped model is commonly used [12, 14, 48, 49] to develop controllers in a 2-D simulation environment (Figure 3.63). This model makes two major simplifications:

1. The robot is represented by a system of point masses that are connected by massless links via 1-DOF revolute joints.
2. All motions are assumed to take place in the sagittal plane; therefore, the model only includes DOFs that are associated with motions in the sagittal plane.

Due to these simplifications, it is feasible to calculate the *exact dynamics* of the model using the Lagrange equation. Unfortunately, since the model neglects the lateral component of the robot dynamics, additional development and testing is required before the robot controller can be implemented on the actual system. That being said, it is much easier to implement and test a robot controller in the 2-D simulation; the

results can be used to demonstrate the effectiveness and feasibility of a new robot controller. As a first proof-of-concept, the ZRAM-based controller was implemented on a planar robot model (Chapter 5) before it was further developed and tested on *Webots*.

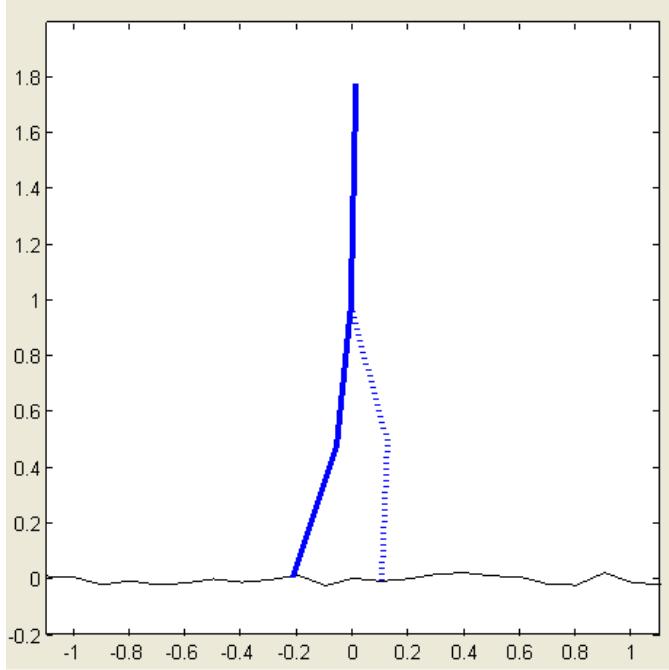


Figure 3.63: A planar biped model in a 2-D simulation environment.

From the above descriptions of the simplified models, it is clear that the main desirable qualities: 1) accurate dynamic modelling, and 2) simplicity, are in conflict with one another. However, a compromise can be made between these qualities because they are not stringent requirements for the robot controller. For example, the robot controller does not require a perfect representation of the biped dynamics since it can rely on feedback control to compensate for any modelling errors. A model that offers a good balance between model accuracy and model simplicity is called the *Linear Inverted Pendulum Model* (LIPM). The LIPM is commonly used in realtime walking controllers [11, 14, 50] and it focuses on the relationship between the COG, the COP, and the GRF. This relationship, called the *reduced biped dynamics* (Section

3.6.2) is simple enough to be used in a real-time controller; at the same time, it can capture the full dynamics of the biped with reasonable accuracy [10]. The ZMP-based controller and the ZRAM-based controller uses two different types of LIPM respectively to simplify the biped dynamics. The ZMP-based controller is a simple but pragmatic approach that uses the basic 3-D LIPM model. The description and the equations of motion of the model will be presented in Section 3.6.3. The ZRAM-based controller is based on a variation of the LIPM called the *Zero Rate of Change of Angular Momentum* (ZRAM) model that focuses on the rotational dynamics of the biped. The ZRAM model will be described in further details in Section 3.6.4.

3.6.2 Dynamic Relationships in Bipedal Walking

A main characteristic of bipedal walking is that it can only generate motion by converting internal joint torques to contact forces through interacting with the environment. This implies that 1) the loss of foot contact will likely cause the biped to fall because it cannot generate the necessary GRF to regain balance, and 2) there is a relationship between the joint torques and the GRF. Although this relationship is important in bipedal walking, it cannot be used directly in a robot controller. Rather, we must establish an additional relationship between the GRF, the COP and the COG (represents the motion of the biped) in order to facilitate high level motion planning and stability control. During walking, the biped relies solely on the GRF to generate the COG motion; therefore, the GRF, COP, and COG are the key input and state variables that defines the walking dynamics of the biped.

The relationship between the GRF, COP and COG can be approximated by the dynamics of the LIPM (Figure 3.64). There are two main dynamic similarities between the biped and the inverted pendulum: 1) both systems are inherently unstable and has a tendency to fall under a large perturbation, and 2) the motion of

the biped (represented of the COG) and the pendulum (represented by the head) is largely dependent on the line of action of the GRF. The line of action in turn, can be controlled by the relative position between the COG and the COP (Section 3.6.3).

It is important to note that the cart (base) of the inverted pendulum can only be manipulated by the acceleration of the head. Similarly, in the biped, we cannot control the COP (ZMP) position by directly apply a force to it; instead, we can only use the COG acceleration to manipulate the COP. For this reason, the goal of many realtime walking controllers [10, 51] is to use the COG to manipulate the COP (ZMP) such that it remains within the support polygon. The COG in turn is manipulated by joint motions using the linear relationship between the COG velocity (task space vector) and joint velocities (joint space vector) that was described in Equation 3.2. Thus, the COG Jacobian is a key relationship that forms a “bridge” between the internal dynamics of the biped (the joint torque and the GRF) and the external dynamics (the GRF and COG motion).

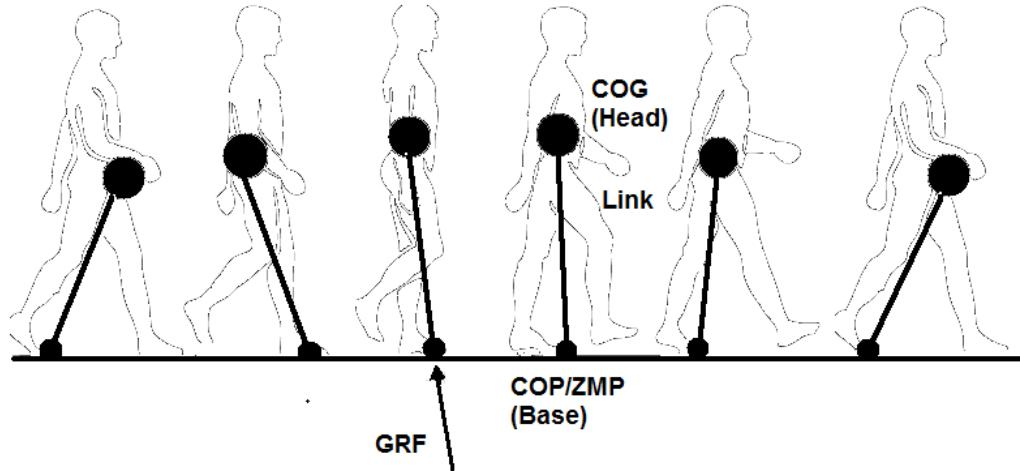


Figure 3.64: Modelling the dynamics of walking using an inverted pendulum model

In summary, several important relationships exist within the complex dynamics

of the biped:

1. The relationship between the joint torques and the GRF.
2. The relationship between the GRF, COP, and the COG.
3. The relationship between the joint motion and the COG motion.

The next section will present the derivation of the LIPM dynamics and a discussion on how it can be applied in a robot controller.

3.6.3 The 3-D Linear Inverted Pendulum Model (3-D LIPM)

The Equation of Motion of the 3-D LIPM

The LIPM introduced in the last section is an important part of realtime walking controllers. It allows the controller to focus on the essential dynamic relationships in bipedal walking and facilitate trajectory generation that would otherwise be too computationally expensive to be performed. As shown in Figure 3.65, the pendulum has a point mass m that is connected to the base of the pendulum by a massless link (models the stance leg). The base can translate along the horizontal axis and has a pivot point about which the pendulum can rotate. The location the point mass and the base of the LIPM corresponds to the COG position $\vec{X}_G = [x_G, y_G, z_G]^T$ and the COP (or equivalently, the ZMP position) $\vec{X}_{ZMP} = [x_{ZMP}, y_{ZMP}, z_{ZMP}]^T$ of the biped, respectively. The length of the link \vec{r} is given by the distance between the base of the pendulum and the point mass $\vec{r} = \vec{X}_G - \vec{X}_{ZMP}$.

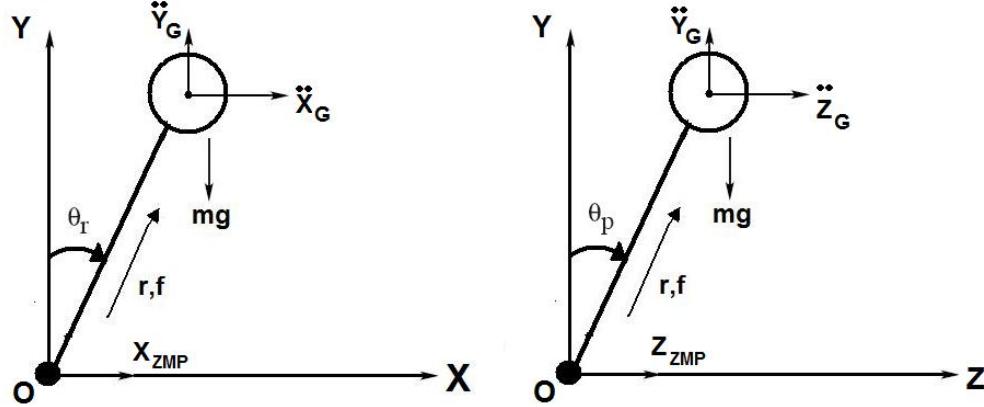


Figure 3.65: The 3-D LIPM projected in the lateral plane (left) and the sagittal plane (right).

The GRF is assumed to act on the pendulum along the massless link \vec{r} . As shown in Figure 3.65, this is equivalent to adding an linear actuator on the massless link and accelerating the point mass by an actuator force \vec{f} . Additionally, the acceleration of the point mass is also dependent on the gravitational force mg . If the pivot point is actuated, then μ_r and μ_p are the effective input forces acting on the point mass. Note that μ_r and μ_p are calculated from the nonlinear mapping between the point mass acceleration and the input torques (τ_p and τ_r) [49].

We can derived the moment equation of the 3D-LIPM in the sagittal and lateral plane as follows:

$$m\ddot{x}_G (y_G - y_{ZMP}) - m\ddot{y} (x_G - x_{ZMP}) = \mu_p - mg (x_G - x_{ZMP}), \quad (3.56a)$$

$$m\ddot{z}_G (y_G - y_{ZMP}) - m\ddot{y} (z_G - z_{ZMP}) = \mu_r - mg (z_G - z_{ZMP}), \quad (3.56b)$$

Simplifying the above, we get the equation of motion of the 3-D LIPM,

$$\ddot{x}_G = \left(\frac{\ddot{y} - g}{y_G - y_{ZMP}} \right) (x_G - x_{ZMP}) + \left(\frac{1}{m \cdot y} \right) \mu_p, \quad (3.57a)$$

$$\ddot{z}_G = \left(\frac{\ddot{y} - g}{y_G - y_{ZMP}} \right) (z_G - z_{ZMP}) + \left(\frac{1}{m \cdot y} \right) \mu_r, \quad (3.57b)$$

The inputs forces μ are calculated from the ankle torques τ of the biped. Based on the “ankle strategy” observed in human walking, the authors in [49] uses ankle torque to 1) track a reference COG trajectory and, 2) indirectly manipulate the ZMP to maintain balance. However, the proposed controller in Chapter 4 does not rely exclusively on the ankle torque to manipulate the COG; therefore, the input forces μ is excluded from the 3D-LIPM.

In this case, the equation of motion of the 3D-LIPM becomes:

$$\ddot{x}_G = \left(\frac{\ddot{y} - g}{y_G - y_{ZMP}} \right) (x_G - x_{ZMP}), \quad (3.58a)$$

$$\ddot{z}_G = \left(\frac{\ddot{y} - g}{y_G - y_{ZMP}} \right) (z_G - z_{ZMP}). \quad (3.58b)$$

The Simplifications and Assumptions of the 3-D LIPM

- The equation of motions (Equations 3.58a and 3.58b) are linearly independent. Therefore, the dynamics of the 3D-LIPM can be decoupled into two 2D-LIPMs in the sagittal the lateral plane and each of which can be controlled by a separate controller.
- It is desirable to limit the vertical motion of the COG in order to maintain a kinematic posture that is suitable for walking (ex: a low vertical COG position with respect to the world frame might lead to a “stooping” posture that eventually causes the biped to fall). In [40, 49, 52], the authors applied constraints

to the equation of motion of the 3D-LIPM (Equations 3.58a and 3.58b) and limited the motion of the pendulum to a horizontal plane at a specified height above the ground. However, vertical COG motions are useful in certain phases of the walking cycle. For example, a “pitching motion” is used when the biped shifts its weight from the back foot to front foot in DB. For this reason, we will use the unconstrained equation of motion of the LIPM (Equations 3.58a and 3.58b) and control the vertical position of the COG by a separate feedback controller (Chapter 4).

- The 3D-LIPM assumes that the total mass of the biped is concentrated at a single point with no moment of inertia. This implies that the model has no centroidal moment and that the line of action of the GRF is constrained along the vector between the COG and the COP \vec{r} . However, biomechanical studies in [53,54] show that a large class of human body movement involves non-zero centroidal moment. For example, as noted in [54], there is an increased spin angular momentum in the biped during SS and the line of action of the GRF does not necessarily pass through the COG. Since the swing leg has the largest fluctuation in forward velocity, it is the dominant source of spin angular momentum for the biped in SS. The spin angular momentum H_{cen} is given by the following equation:

$$H_{cen} = \sum_{i=1}^n r_{Gi} \times m_i v_i + I_i w_i \neq 0, \quad (3.59)$$

where n is the number of links in the biped, m_i and I_i are the mass and the moment of inertia of *link i*, r_{Gi} is the position of the i^{th} link COG with respect to the overall COG of the biped, and v_i and w_i are the linear and angular velocity of *link i* respectively.

From Equation 3.59, we see that unless there is a large linear/rotational acceleration in the swing leg (ex: a jerk in the swing leg during lift-off), or if the inertia of the leg is large, then the change in the spin angular momentum is small and the LIPM is a reasonable approximation of the actual biped dynamics.

In addition, the authors in [53] identified a case where the biped is required to generate spin angular momentum and centroidal moment in order to control the COG and regain dynamic stability. However, since the LIPM assumes zero centroidal moment at all times, it cannot be used to control the rotational dynamics of the biped to achieve stability.

- Closely related to the previous point, the 3D-LIPM ignores the effects of the distributed link masses of the biped and assume the leg to be massless. As noted in [53], the gravitational and inertial components of the unmodelled swing leg dynamics in the LIPM can cause the actual COG to deviate from the reference COG and the ZMP to move away from its assumed stable position.

Given the number of simplifying assumptions, the LIPM is a simple approximation of the full dynamics of the biped. In order to account for modelling errors, most realtime controllers [55] rely on feedback control. For example, since the spin angular momentum of the individual links of the biped H_{cen} is unaccounted for in the 3D-LIPM, the modelling error might cause unexpected “disturbances” to the COG motion. Therefore, a feedback controller is used to generate the necessary compensatory motions to reduce the difference between the actual and desired COG position.

The next section will provide a qualitative overview of the dynamic behaviours of the 3-D LIPM. These dynamic behaviours form the basis of the ZMP-based controller that will be presented in Chapter 4.

The Dynamic Behaviour of the 3-D LIPM

Let's examine the dynamic behaviour of the LIPM in a 2-D case where there are only conservative forces involved (ie: the only force that is affecting the pendulum is gravity). The inverted pendulum has an equilibrium point in the upright position when $\theta_r = \theta_p = r_X = r_Z = 0$. As shown in Figure 3.66, the acceleration of the pendulum will either increase or decrease depending on the direction that it is moving and its position with respect to the equilibrium point. If the pendulum starts at a position that is to the right of the equilibrium (pivot) point (ie: the COG is **ahead** of the ZMP and $Z_G - Z_{ZMP} > 0$) and the COG is moving in the positive Z direction, then the COG will accelerate ($\ddot{Z}_G > 0$). In this case, the gravitational potential energy of the pendulum will be converted to kinetic energy. If the pendulum starts at a position that is to the left of the equilibrium point (ie: the COG is **behind** of the ZMP and $Z_G - Z_{ZMP} < 0$) and it is moving towards the equilibrium point, then the COG will slow down ($\ddot{Z}_G < 0$) and the kinetic energy will be converted into potential energy. This behaviour can be expressed mathematically from the equation of motion of the LIPM with respect a polar coordinate frame (with θ and \vec{r} as state variables):

$$\ddot{\theta}_p = \frac{g}{\vec{r}} \sin \theta_p. \quad (3.60)$$

From Equation 3.60, it is clear that the pendulum will always accelerate in the same direction as its current position with respect to the equilibrium point. The LIPM exhibits the same behaviour in the case where there is a non-conservative force GRF acting on the point mass: the COG accelerates when it is moving away from the equilibrium point and slows down when it is moving towards it. This is due to two properties of the GRF: 1) The GRF cannot exert an attractive force and must always act in the positive vertical direction, and 2) the GRF is assumed to act along \vec{r} . Therefore, like gravity, the GRF will always accelerate the COG away

from the equilibrium (pivot) point (Figure 3.66).

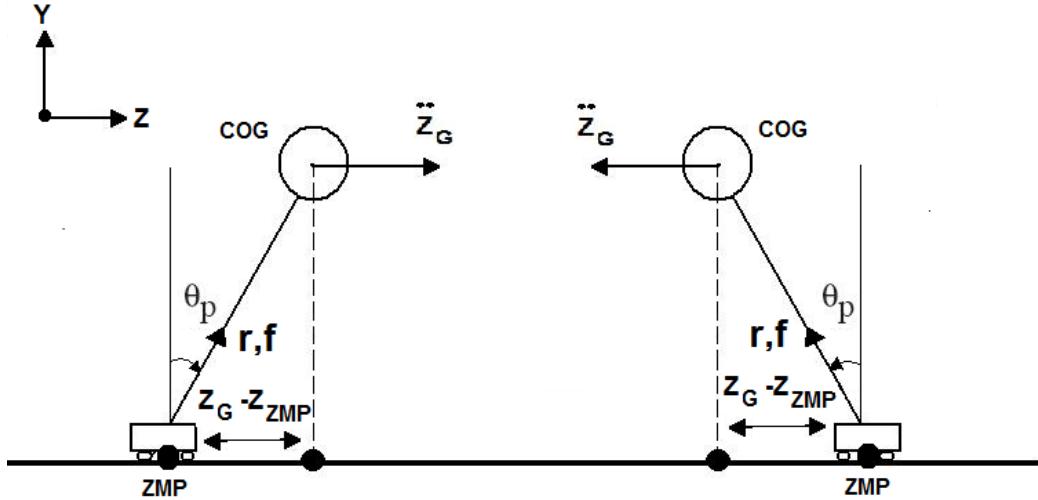


Figure 3.66: Dynamic behaviour of the LIPM: the distance between the COG and the ZMP causes the acceleration of the COG to increase or decrease.

An important insight from the LIPM is that we can control the motion of the COG motion by manipulating the relative distance between the ZMP (pivot point) and the COG. In addition, since the GRF is assumed to act along \vec{r} , its effects on the COG can be controlled by the COP. Based on the dynamic behaviour of the LIPM and the equation of motion (Equations 3.58a and 3.58b), we can design a controller that tracks a reference COG trajectory by ZMP manipulation. By moving the ZMP behind of the COG (left figure in Figure 3.66), the GRF and gravity will accelerate the COG forward; by moving the ZMP ahead of the COG (right figure in Figure 3.66), the GRF and gravity will act to oppose the COG motion and the COG will slow down. However, it is important to note that in order to maintain dynamic stability, the ZMP motion is limited to within the support polygon. This dynamic behavior in the LIPM forms the basis for the ZMP-based controller that will be presented in Chapter 4. The actual dynamics of the biped is not fully captured in the model. The effects of the swing leg dynamics is significant for ABL-BI, gravitational force, inertia

force (linear and angular) is not taken into account.

3.6.4 The ZRAM Model

Recent biomechanical studies in human walking [3, 56] show that the spin angular momentum is actively regulated by the human to achieve rotational stability throughout the walking cycle. Motivated by these studies, there is an increasing number of control strategies that use the spin angular momentum as a control variable to achieve dynamic and postural stability. The control objective of these controllers is to generate joint trajectories that track a desired spin angular momentum (usually set to zero). Alternatively, the spin angular momentum can be minimized by ensuring that its rate of change is zero. This model is called the zero rate of change of angular momentum (ZRAM) and it occurs when the net external moments about its COG is zero and the biped is rotationally stable. Mathematically, the ZRAM model can be expressed as a nonlinear relationship between the COG, the COP, and the GRF that is derived from the rotational dynamic equations of the biped. It is important to note that ZRAM only guarantees constant spin angular momentum. A non-zero constant spin angular momentum can cause the robot to tip over; therefore if the biped has a zero initial angular momentum, then ZRAM is a necessary but insufficient condition for maintaining dynamic balance.

The remainder of this section will focus on the derivation of the nonlinear relationship that defines the ZRAM model. Starting with the most general case (Figure 3.67), the forces and moments that are involved in the rotational dynamics of the biped include:

1. The GRF of the left foot F_l and right foot F_r .
2. The external disturbance force Q applied at the point S .

3. The external disturbance torque τ_d about the COG.
4. The gravitational force at the COG mg .
5. The centroidal moment \dot{H}_G .
6. The inertial force contributed by the linear acceleration of the COG $m\vec{a}$.

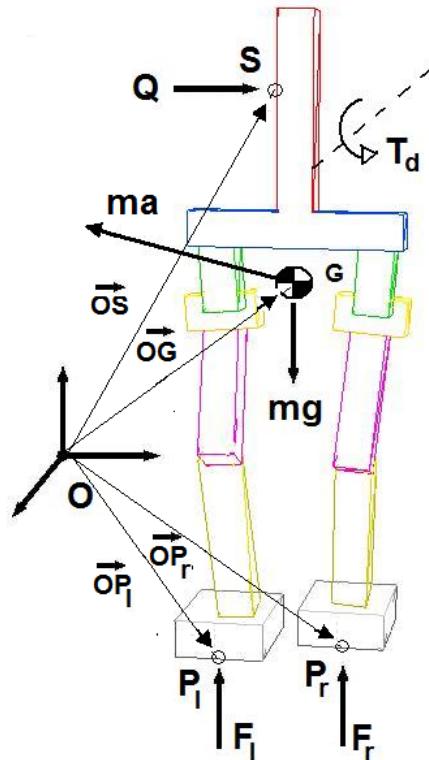


Figure 3.67: The external forces and moments that act on the biped.

In Figure 3.67, \overrightarrow{OP} denotes the vector between the origin and the COP, \overrightarrow{OG} is the vector between the origin and the COG, \overrightarrow{OS} is the vector between the origin and the point of action of the external force \vec{Q} and $\overrightarrow{r_{Gi}}$ is the vector between the local COG of link i and the overall COG.

Taking the moment about the origin O :

$$\overrightarrow{OP}_l + \times \vec{F}_l + \overrightarrow{OP}_r \times \vec{F}_r + \tau_d + \overrightarrow{OS} \times Q + \overrightarrow{OG} \times mg = \dot{H}_G + \overrightarrow{OG} \times m \vec{a}, \quad (3.61a)$$

$$\dot{H}_G = \sum_{i=1}^n I_i \dot{w}_i + \sum_{i=1}^n \vec{r}_{Gi} \times m_i \vec{a}_i, \quad (3.61b)$$

where I_i is the moment of inertia of link i , and w_i is the angular velocity of link i about the COG.

Assuming that the biped is walking on flat ground with no external disturbances, and combining \vec{F}_l and \vec{F}_r into a single resultant GRF \vec{F} that is acting at the COP:

$$\overrightarrow{OP} \times \vec{F} + \overrightarrow{OG} \times mg = \dot{H}_G + \overrightarrow{OG} \times m \vec{a}, \quad (3.62)$$

Taking the moment about the COG, Equation 3.62 further reduces to:

$$(\vec{r}_{COP} - \vec{r}_G) \times \vec{F} = \dot{H}_G, \quad (3.63a)$$

$$(\vec{r}_{CMP} - \vec{r}_G) \times \vec{F} = 0, \quad (3.63b)$$

In Equation 3.63, the only external force that is relevant in the rotational dynamics is the GRF. Equation 3.63a depicts the general case where the GRF induces a centroidal moment. The magnitude of \dot{H}_G in Equation 3.63a is proportional to the distance between the CMP and COP (Chapter 1). In the case of zero centroidal moment (Equation 3.63b), the GRF passes through the COG and the COP coincides with the CMP (ie: \vec{r}_{COP} in Equation 3.63a becomes \vec{r}_{CMP} in Equation 3.63b.)

Based on the hypothesis of spin angular momentum regulation in human walking, Equation 3.63b is able to model the rotational dynamics of human walking with a high degree of accuracy. In order to validate this hypothesis, the researchers in [3, 57] rearranged Equation 3.63b to calculate the horizontal components of the GRF of a human model with anthropomorphic parameters and mass distributions (Equations 3.64a and 3.64b):

$$F_x = \frac{F_y}{y_G} (x_G - x_{COP}) \quad (3.64a)$$

$$F_z = \frac{F_y}{y_G} (z_G - z_{COP}) \quad (3.64b)$$

The calculated values from Equations 3.64a and 3.64b were compared against experimental GRF values (measured from actual human walking trials) and the two sets of results matches very accurately (with coefficients of determination of over 0.95) [53] [53]. These results validate the nonlinear relationship between the COG, the CMP and the GRF (Equation 3.63b) and supports the use of spin angular momentum regulation as a means of bipedal walking control. Although it was stated in this section that ZRAM is a necessary condition for dynamic stability, in certain situations, the ZRAM condition might not be the optimal control objective. For example, a compensatory \dot{H}_G is required to counteract external disturbances. In addition, at the start of SS, a non-zero \dot{H}_G can help the foot to lift-off and swing forward. For this reason, in the robot controller, the ZRAM condition is relaxed in certain phases of the walking cycle (Chapter 5).

From 3.63b, we obtain equations of the CMP in terms of the COG and GRF:

$$x_{CMP} = x_{COG} - \frac{F_x}{F_y} y_{COG}, \quad (3.65a)$$

$$z_{CMP} = z_{COG} - \frac{F_z}{F_y} y_{COG}, \quad (3.65b)$$

Using Equations 3.65a and 3.65a, we can generate rotationally stable COG trajectories based on a pre-planned CMP trajectory and the actual GRF. Using the above equations, the rate of change of angular momentum can be indirectly regulated by COG manipulation. The details of the ZRAM-based controller will be presented in Chapter 5.

3.7 The Control Architecture

The purpose of this section is to provide a top level overview of the robot controllers before moving on to the detail algorithms in Chapters 4 and 5. Both the ZMP-based controller and the ZRAM-based controller have a feedback control structure that can be divided into five control blocks. In this section, we will ignore the details and instead focus on the roles of each block within the robot controller and the data flow between the blocks (the input and output of each block).

The overall control architecture of the robot controllers is presented in Figure 3.68.

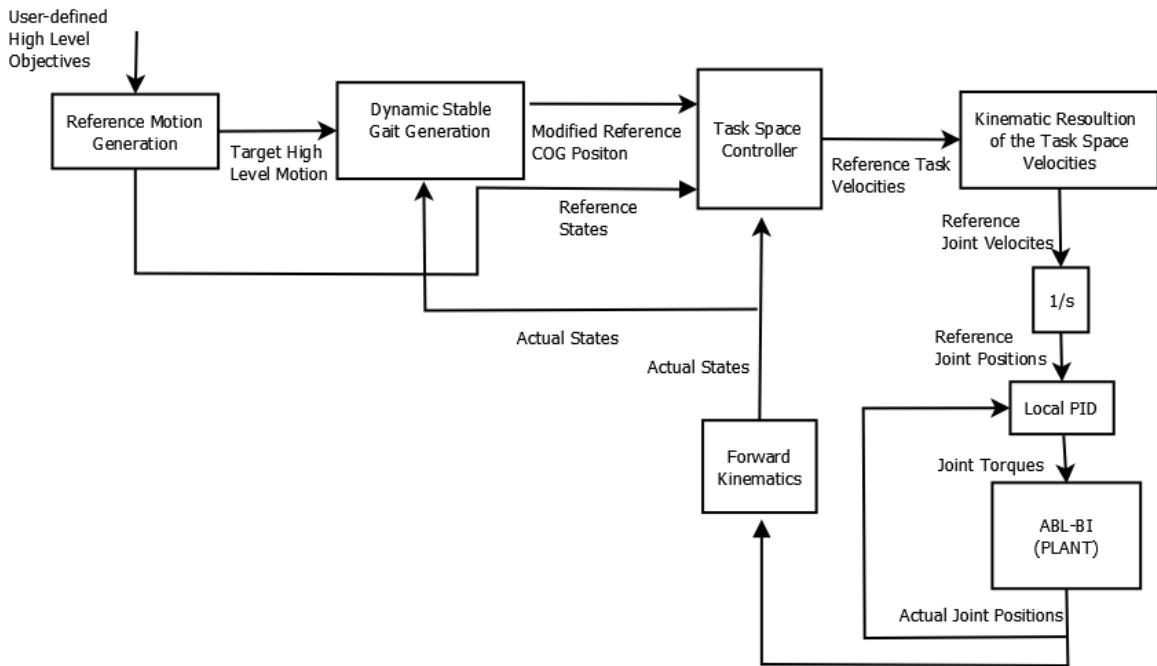


Figure 3.68: A schematic diagram of the control architecture of the robot controllers

A brief description of each control block and the input/output variables (in bold text) is given below:

- **Reference Motion Generation**— Given a set of **user-defined high level objectives** (ex: the walking velocity of the biped), the intermediate objectives for each locomotion state is calculated (ex: the step period, the step length, the reference COG/CMP position) and a reference trajectory for each state variable is generated from the cubic spline interpolation method (Chapter 4). The reference trajectories for the pelvis orientation and the swing foot position and orientation (referred to as the **reference states**) is directly used as the inputs for the *Task Space Controller*. On the other hand, the absolute reference COG/CMP trajectory (denoted as the **target high level motion**) is used to generate dynamically stable COG trajectories.

- **Dynamically Stable Gait Generation**— Given the **target high level motion** trajectory, a dynamically stable **modified reference COG** trajectory is generated. For the ZMP-based controller, the stability constraint is applied to the modified reference COG trajectory by a separate *ZMP manipulation* process (Chapter 4). For the ZRAM-based controller, the dynamic stability constraint is applied using the ZRAM model defined in Section 3.6.4. This control block is the most important part of the robot controller and will be described in Chapters 4 and 5.
- **Task Space Controller**— A PD controller that is used to track the modified COG position and other reference positions in the task space (such as the pelvis orientation and swing foot orientation and position) that are specified by the *Reference Motion Generation* block. The output of this controller is an actuating **task space velocity vector** that is used to manipulate the actual states (obtained from the *Forward Kinematics* block) to track the reference states.
- **Kinematic Resolution of the Task Space Velocities**— The objective of this block is to compute a joint space solution (expressed by the joint space velocity vector) that would allow multiple reference velocities (expressed by the **reference task velocity vector**) to be achieved by the biped simultaneously. In other words, this control block “decomposes” a given set of reference velocities (from the *Task Space Controller*) and outputs a set of **reference joint velocities**. In order to avoid conflicts between the high level tasks, the joint space solution is calculated from a task priority method. Basically, this framework allows the high priority tasks to be fulfilled in the expense of the low priority tasks. The low priority task can only be achieved as long as it does not interfere with the high priority tasks. Please refer to Section 3.5.3 of this Chapter for further details.

- **Joint Space Controller**– The reference joint positions can be obtained by integrating the reference joint velocities. Given the **reference joint positions**, the *Joint Space Controller* applies PID control to the actual joint positions (measured from the encoders of the biped) and outputs a set of **joint torque commands** for the actuators.
- **Forward Kinematics**– On the actual robot, most of the actual states (such as the COG position and the pelvis orientation) cannot be directly measured from onboard sensors. Instead, the actual states are first expressed as end-effectors of the kinematic chains defined in (Section 3.3.2). Then, the actual states are calculated from the kinematic equations that relates the **actual joint position** (obtained from the biped encoders) to the end-effector positions/orientations. **The actual states** are subsequently used in the various control loops in the robot controller. The forward kinematics of ABL-BI is described in details in Section 3.4 of this Chapter.

Chapter 4

A Walking Controller Based on the LIPM and the ZMP Stability Criterion

In the previous chapter, we developed the components that are essential to walking controllers such as the locomotion state machine, the inverse differential kinematic controller, and the reduced dynamics model for ABL-BI. In this chapter, we will integrate these elements into a control scheme (the ZMP-based controller) that generates stable walking gaits. The basic principles of the ZMP-based controller is based largely on the work by Sugihara and Nakamura in [55]. The ZMP-based controller uses the LIPM to simplify the dynamics of the biped, the ZMP criterion to generate dynamically stable COG trajectories, and the Whole Body Cooperation (WBC) method to decompose task space objectives to reference joint trajectories. In addition, modifications were made to the original controller in [55] to simplify high level motion planning and kinematic postural control of the biped. Simulation results show that ABL-BI can perform multi-step walking while [55] only reported results for one step cycle.

A walking test was performed in *Webots* to verify the performance of the ZMP-based controller. Although the results show that the ZMP-based controller was

able to generate stable walking gaits, the walking speed of ABL-BI was significantly lower than other bipeds. The simulation results also highlighted several issues in the ZMP-based controller and the ground model in *Webots*. Two important questions were raised by the simulation tests of the ZMP-based controller. What are the limitations of the ZMP-based controller and how can the design of the controller and the physical platform be modified in order to improve the walking performance of ABL-BI? These questions are addressed in the second part of this chapter. With supporting evidence from simulation trials, we identified serval factors that limits the performance of the ZMP-based controller and ABL-BI; this includes the mass distribution of the biped and the limitations of the LIPM. Some of these issues were addressed in a new ZRAM-based controller (the ZRAM-based controller) that will be presented in Chapter 5 of this thesis. Therefore, the ZMP-based controller can be regard as a “pilot test” for the subsequent controller developments of ABL-BI.

This chapter is organized as follows. Section 4.1 presents the control algorithm of the ZMP-based controller and describes how the dynamic relationship between the COG and ZMP is used to generate dynamically stable gaits. The simulated walking test results are provided in Section 4.2 to verify and evaluate the performance of the ZMP-based controller on ABL-BI. In Section 4.3, an in-depth discussion on the limiting factors of the ZMP-based controller and ABL-BI, supported by simulation results, is presented. Finally the main results and observations in this chapter are summarized in Section 4.4.

4.1 The Control Algorithm

The ZMP-based controller is based largely on a real-time controller that was proposed by Sugihara and Nakamura in [55]. The basic principles of Sugihara’s controller are

as follows:

- The complex dynamics of the biped is simplified using the LIPM to facilitate real-time motion generation and control. Dynamically stable walking gait (defined by the reference COG and swing foot trajectories) is generated based on 1) user-defined high level objectives, 2) the dynamic relationship between the ZMP and the COG that is defined in the LIPM, and 3) the ZMP stability criterion.
- Feedback control is used to attenuate modelling errors and external disturbances.
- The task space objectives (including the COG and swing foot trajectories) are decomposed into joint trajectories using the WBC method (described in Section 3.5.1).

The ZMP-based controller retained most of the basic principles of Sugihara's controller. However, two modifications were made to facilitate implementation on ABL-BI:

- The high level objectives in Sugihara's controller are specified in terms of reference COG velocity. However, it is more intuitive to plan the high level motion of the biped based on the COG position. Therefore, in the ZMP-based controller, motion planning takes place in the position level and the reference COG velocity is the output of an additional COG position controller.
- Sugihara's controller uses the weighted-pseudoinverse method [58] to obtain the joint-space solutions for the reference task-space velocities. However, the

elements of the weighting matrix must be manually tuned to achieve the desired effects. In contrary, the ZMP-based controller uses a more efficient task-prioritization method to assign different levels of priority to the high level objectives. This method is incorporated in the *Closed-loop Kinematic Controller* that was described in Section 3.5.3.

The essential elements that are common to the ZMP-based controller and the ZRAM-based controller has already been presented in Section 3.7 and Figure 3.68. The distinguishing difference between the ZMP-based controller and the ZRAM-based controller is the way in which the dynamically stable reference trajectory is generated; therefore, it will be the focus of this section. As shown in the the ZMP-based controller block diagram in Figure 4.1, the ZMP-based controller can be divided into four main portions. The *Stable Reference Trajectory Generation* block outputs a dynamically stable reference COG velocity that is based on a user-specified COG trajectory and the ZMP criterion. The reference COG velocity, along with other reference task-space velocities ${}^{ref}X_i$ (ex: reference linear/angular swing leg velocities) are decomposed into reference joint velocities using the *Closed-looped Kinematic Controller*. Finally, the motor torques of the biped are calculated from the *Local PID controller* and the actual states of the biped are calculated from the *Forward Kinematics* block. Note that aside from the *Stable Reference Trajectory Generation* block, the components of the ZMP-based controller were described previously in Section 3.7.

Let's focus more closely on the *Stable Reference Trajectory Generation* block in the ZMP-based controller. As shown in Figure 4.2, there are two control loops: 1) the COG position loop and, 2) the COG velocity loop. In the the COG position loop, the output of the COG position controller is an actuating COG velocity (henceforth referred to as the *absolute reference COG velocity* ${}^{ref}v_G$) that manipulates the actual COG position to the reference position. The *absolute reference COG velocity* is also

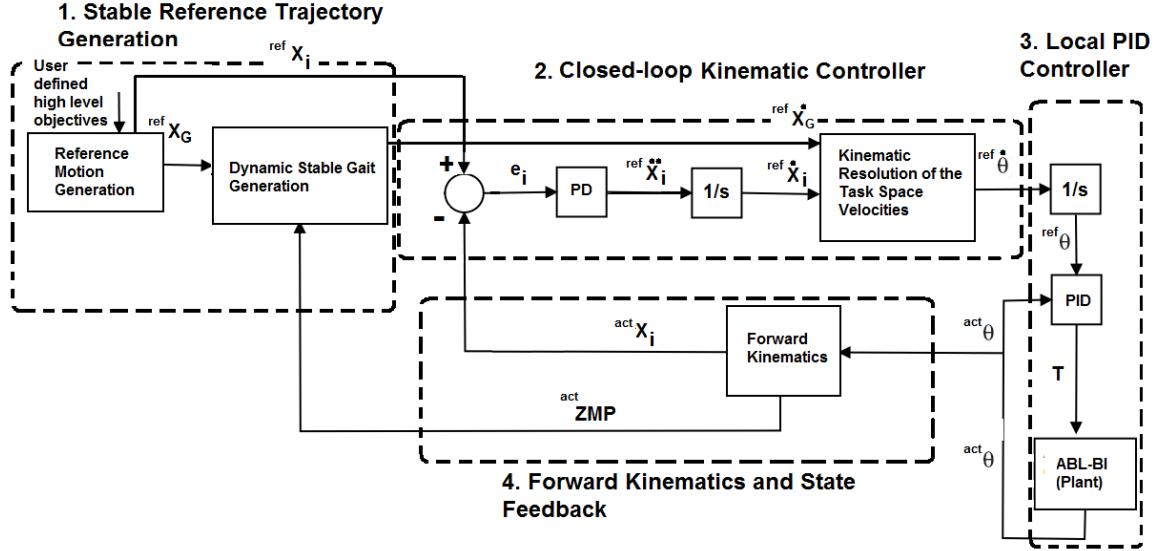


Figure 4.1: A schematic diagram of the control architecture of the ZMP-based controller.

the reference velocity of the COG velocity loop. In this loop, there are two main control blocks:

ZMP Planner The purpose of the ZMP planner is to generate a reference ZMP trajectory. Recall the dynamic behaviour of the LIPM that is described in Section 3.6.3: by moving the ZMP behind of the COG, the GRF and gravitational force will accelerate the COG forward; by moving the ZMP ahead of the COG, the COG will accelerate in the opposite direction. Based on this behaviour, the controller of the COG velocity loop (the *ZMP Planner*) generates a reference ZMP position that manipulates the actual COG velocity to the *absolute reference COG velocity*. If the reference ZMP position (generated from the ZMP planner) is outside of the support polygon (note: this occurs when the *absolute reference COG velocity* is large), it will be placed at a point that is closest to the support polygon.

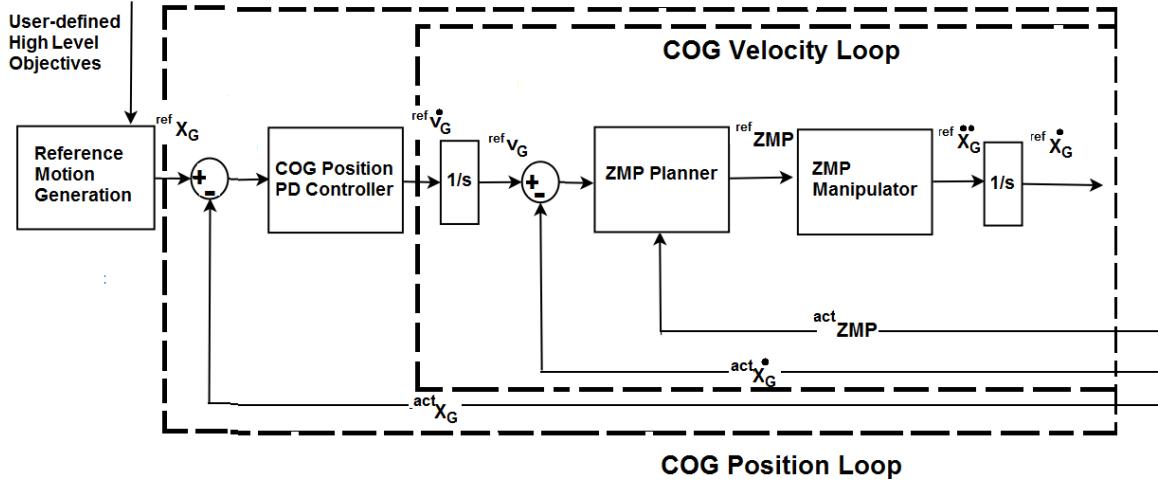


Figure 4.2: A schematic diagram of the Stable Reference Trajectory Generation block in the ZMP-based controller.

ZMP Manipulator Unlike the actual inverted pendulum, the ZMP can only be indirectly manipulated by accelerating the COG in the biped. Using the dynamic equation of the LIPM (Equation 3.58a and 3.58b), the necessary reference COG acceleration is calculated to manipulate the actual ZMP to the reference ZMP, which in turn manipulates the actual COG velocity to the *absolute reference COG velocity*.

$$\ddot{x}_G = \left(\frac{\ddot{y} - g}{y_G - y_{ZMP}} \right) (x_G - x_{ZMP}), \quad (3.58a, \text{ repeated})$$

,

$$\ddot{z}_G = \left(\frac{\ddot{y} - g}{y_G - y_{ZMP}} \right) (z_G - z_{ZMP}). \quad (3.58b, \text{ repeated})$$

Integrating the reference COG acceleration, we obtain the *modified reference COG velocity* ${}^{ref}\dot{X}_G$, which is the output of the *Stable Reference Trajectory Generation* block. It is important to note the difference between the *absolute reference COG*

velocity and the *modified reference COG velocity*. First, the *absolute reference COG velocity* is used to track the reference COG position that is specified by the user. However, in order to ensure that the *absolute reference COG velocity* is dynamically stable (based on the ZMP criterion), a *modified reference COG velocity* is generated from the *absolute reference COG velocity*, the ZMP planner, and the ZMP manipulator. Essentially, the ZMP Planner serves as a “saturation block” that applies stability constraint on the *absolute reference COG velocity* and the ZMP Manipulator generates the modified velocity. Note that a more detailed description of the ZMP Planner and the ZMP Manipulator can be found in [55].

4.2 Initial Walking Tests in Webots

The control objectives of the ZMP-based controller are 1) generate stable walking gaits for ABL-BI, and 2) maintain a desirable posture (ex: an upright pelvis and torso orientation) throughout the walking cycle. The performance of the ZMP-based controller was evaluated based on its ability to achieve the above control objectives in a walking test in *Webots*. The following sections will present and discuss the walking test results.

4.2.1 A Summary of the User-specified Parameters for the ZMP-based Controller

This section summarizes the control parameters for the ZMP-based controller that were used in the walking test in *Webots*. The control gains for the “COG position loop” and the “COG velocity loop” are presented in Table 4.1. As well, the control gains for the *Closed-looped Kinematic Controller* are summarized in Table 4.1. The gains were selected based on the observed behaviors of the simulated biped in the simulation trials. It was observed that the gains for the “COG velocity loop”

are very sensitive and cannot be varied by more than 2 – 3%. In addition, there is a coupling between the gains in the COG position loop and the COG velocity loop. Therefore, it is very difficult to tune the gains for the ZMP-based controller. The consequences of this problem will be discussed in Section 4.3. The ZMP-based controller uses the same damping factors (for the pseudoinverse Jacobians) as the step cycle test in Section 3.5.3. The damping factors are presented in Table 3.4.

Table 4.1: The control gains for the ZMP-based controller in the walking test.

High level Objective	Control Gains [X,Y,Z]
COG Position loop P-Gain (X)	$k_p = 0.8$ for DB and $k_p = 0.6$ for SS
COG Position loop D-Gain (X)	$k_d = 0.7$ for DB and $k_d = 0.7$ for SS
COG Position loop P-Gain (Z)	$k_p = 0.4$ for DB and $k_p = 1.0$ for SS
COG Position loop D-Gain (Z)	$k_d = 0.4$ for DB and $k_d = 0.7$ for SS
COG Velocity loop P-Gain (X)	$k_p = -0.25$ for DB and $k_p = -0.25$ for SS
COG Velocity loop D-Gain (X)	$k_d = -0.08$ for DB and $k_d = -0.10$ for SS
COG Velocity loop P-Gain (Z)	$k_p = -0.4$ for DB and $k_p = -0.6$ for SS
COG Velocity loop D-Gain (Z)	$k_d = -0.08$ for DB and $k_d = -0.1$ for SS
COG Position P-Gain (Y)	$k_p = 0.5$
Pelvis Orientation P-Gain	[1.25, 1.00, 1.25]
Swing Foot Position P-Gain	[2.50, 7.00, 5.00]
Swing Foot Orientation P-Gain	[5.00, 2.00, 2.00]

The walking parameters are used to generate reference COG and swing foot trajectories from the cubic spline interpolation method [8]. The walking parameters were selected based on: 1) external and internal constraints that are imposed on the intermediate objectives of the walking cycle (Section 3.2.2) (ex: the target lateral COG position of DB must be within the support polygon in order for the biped to

achieve static stability), and 2) qualitative observations from simulation trials. There are several limiting factors on the DB period, step period, and step length that were observed from the simulation trials. These factors will be discussed in Section 4.3. A summary of the key walking parameters is provided in Table 4.2.

Table 4.2: The key reference trajectory planning parameters for the ZMP-based controller.

Walking Parameters	Values
DB Period	7500 <i>ms</i> (excluding the TorsoRoll motion)
Step Period	6000 – 6500 <i>ms</i>
Step Length	15 <i>cm</i> (equals to the length of the foot)
Maximum height of the swing foot	5 <i>cm</i>

4.2.2 Results and Discussions

Reference COG Tracking

In the initial walking test in *Webots*, the biped was commanded to take four steps using the walking parameters that are summarized above. The duration of the walking test is 78000 *ms* and as shown in the time-elapsed freeze frames (Figure 4.3), the biped was able to walk in a stable manner. Notice that aside from the weight-shift and stepping motion that is found in regular walking, ABL-BI performs a “Torso Roll” motion at the start of each DB after the second step. This motion helps the biped in manipulating its COG in the lateral direction and will be discussed in further details in Section 4.3.1. First, the performance of the ZMP-based controller was evaluated based on its ability to track a user-specified reference COG motion. The comparison between the reference and actual COG position with respect to the world frame is presented in Figures 4.4, 4.5, and 4.6.

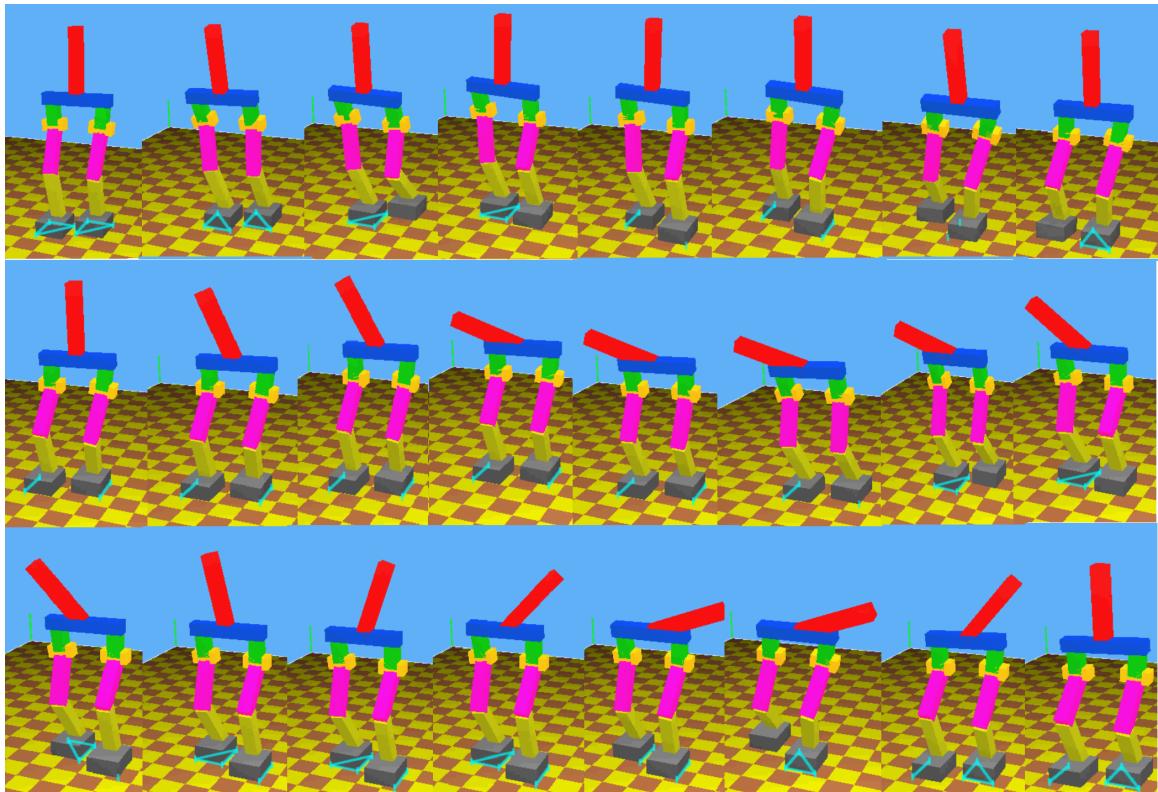


Figure 4.3: Walking test of the ZMP-based controller: Time-elapsed freeze frames for the walking test in Webots.

The key observations from the reference COG tracking results are:

1. The actual COG did not perfectly track the reference COG (with a maximum error of about 4 cm). The main sources of error are: 1) disturbances that were caused by the unmodelled dynamics of the swing leg in SS, especially at state transitions where the magnitude of the swing leg acceleration was the greatest (please refer to Section 4.3 for more details), 2) errors that were introduced by numerical filtering and integration of the reference COG and joint trajectories, and 3) task-space velocity tracking error that was contributed by the damping factor of the Jacobian pseudoinverse. Despite the inaccuracies in COG tracking, the biped was able to walk in a stable manner. This affirms the fact that rather

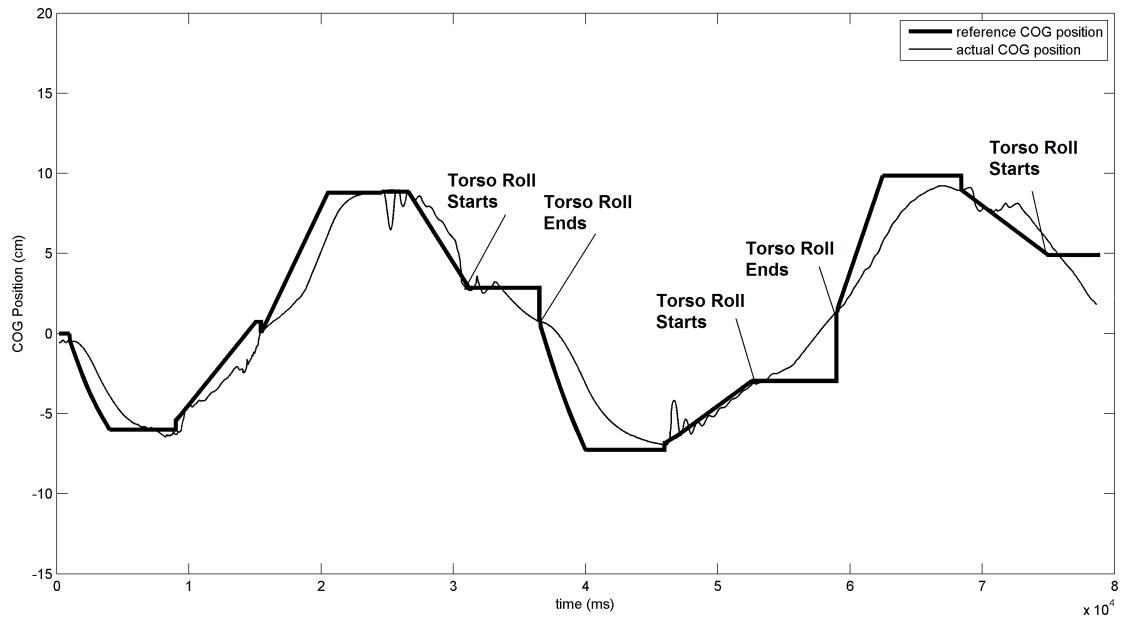


Figure 4.4: The comparison between the reference and actual COG position (X) respect to the world frame.

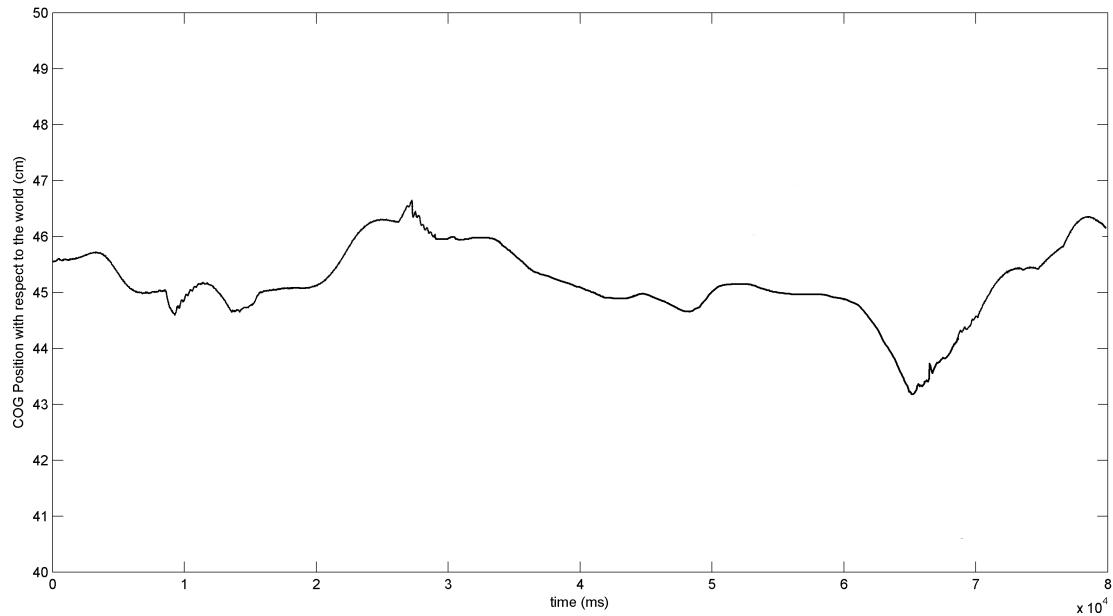


Figure 4.5: The actual COG position (Y) with respect to the world frame.

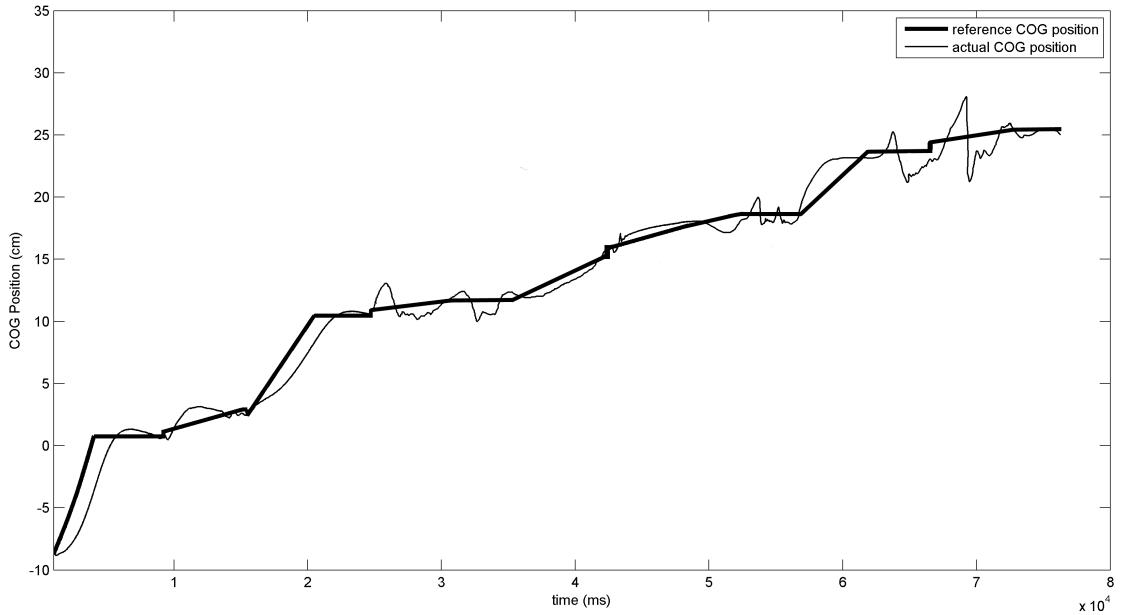


Figure 4.6: The comparison between the reference and actual COG position (Z) with respect to the world frame.

than performing perfect COG tracking, the objective of the controller should instead be: 1) ensure that the COG stays within the support polygon to achieve static stability, or 2) ensure that the acceleration of the COG does not cause the ZMP to move outside of the support polygon to achieve dynamic stability. Since the ZMP-based controller achieved these two objectives (note: the actual ZMP position of the biped will be presented later on) we conclude that it is a stable walking controller.

2. At state transitions, the COG position was perturbed by the angular momentum about the COG that was generated by the swing leg during lift-off (this effect will be discussed in further details in Section 4.3). The effects of this disturbance is shown by the deviations (maximum error of 2 – 3 cm) in the COG position immediately after the state transition in the result plots. However, the

ZMP-based controller was able to attenuate these disturbances through feed-back control.

3. As mentioned in the previous chapter, the vertical component of the COG is not controlled in DB. However, as shown in Figure 4.5, the result of that was only a slight decrease in the COG position (maximum of 2 cm). In the subsequent SS, the ZMP-based controller manipulated the vertical COG position back to its nominal position of around 46 cm .
4. As indicated in Figure 4.4, the duration of the Torso Roll motions was about $5000 - 6000\text{ ms}$ and it occurred at the start of DB after the second step cycle. The COG during this period was manipulated by the motion of the Torso and did not track the reference COG trajectory (the constant reference COG trajectory in Figure 4.4 only acts as a “placeholder”).

In summary, the ZMP-based controller was able to track the reference COG position closely enough to enable stable walking. Next, we will examine how the ZMP-based controller apply dynamic stability constraints on the user-specified reference COG trajectory.

The Final Reference ZMP Placement

Recall from Section 4.1 that the reference ZMP manipulates the actual COG velocity to the absolute reference COG velocity. However, in situations where there is a large difference between the reference and actual COG position (ex: after a disturbance), the PD controller for the “COG position loop” will generate large absolute reference COG velocities that cannot be achieved by the biped in a stable manner. Thus, the ZMP-based controller generates a modified reference COG velocity trajectory from a stable reference ZMP trajectory (please refer to Section 4.1 for more details). Figures 4.7 and 4.8 show the reference ZMP trajectories that were generated from the *ZMP Planner* (the PD controller for the “COG velocity loop”). If the reference ZMP position (the “actuating” output of the ZMP planner) was outside of the support polygon, then the ZMP-based controller will place it to the nearest point in the support polygon (“the final reference ZMP placement”).

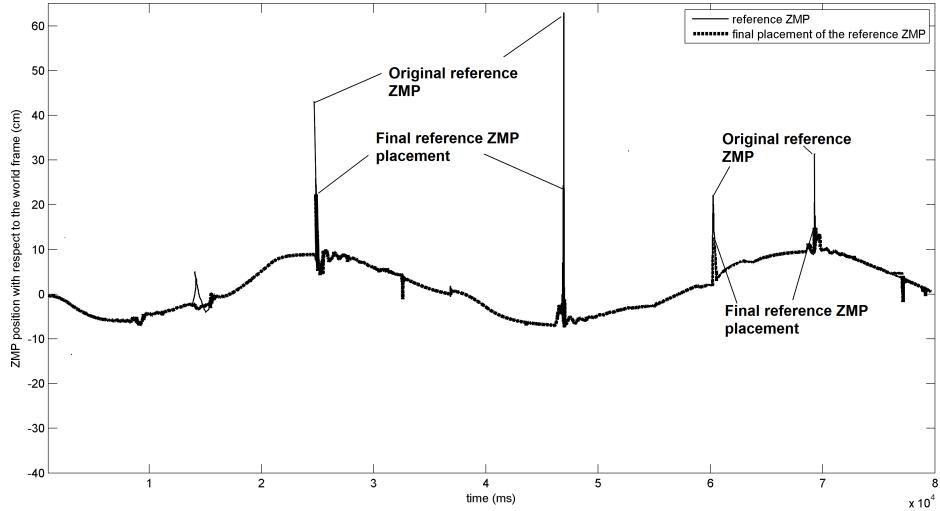


Figure 4.7: The comparison between the reference and modified ZMP position in the X direction with respect to the world frame.

As shown in the result plots, the reference ZMP position was only corrected by

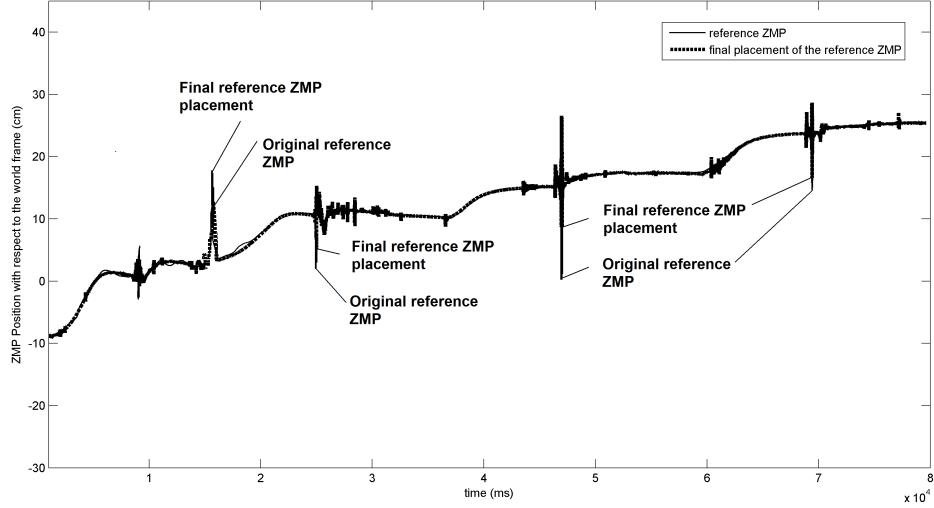


Figure 4.8: The comparison between the reference and modified ZMP position in the Z direction with respect to the world frame.

the ZMP-based controller at state transitions. The reasons are:

- The reference COG trajectory was planned such that it will not produce large dynamic effects that move the ZMP outside of the support polygon. Note that the reference trajectories in the ZMP-based controller were limited by factors that will be described in Section 4.3 (ie: we cannot plan reference motions at any arbitrary velocity).
- In the initial walking test, there were no external disturbances applied to the biped that can cause the actual COG position to deviate significantly from the reference COG position. As the result of that, the reference ZMP from the PD controller of the “COG velocity loop” remained within the support polygon. An exception to that occurred at state transitions where the swing leg inertia effects and large GRF caused large disturbances. In this case, the reference ZMP (generated from the ZMP planner) was outside of the support polygon and the ZMP-based controller had to place the reference ZMP at the nearest point in the support polygon.

The Modified Reference COG Velocity

The modified reference COG velocity that was generated from the ZMP manipulator also differed significantly from the original reference velocity at state transitions. This is indicated in Figures 4.9 and 4.10. Note that the slight difference between the absolute and modified reference COG velocity in DB and SS were due to errors that are caused by numerical filtering and integration.

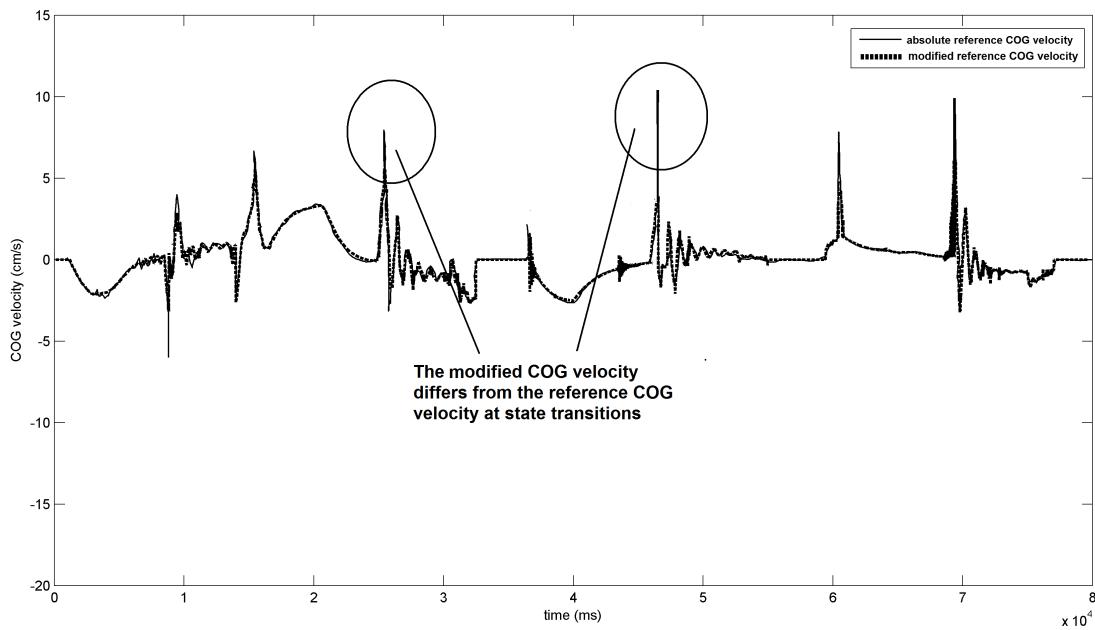


Figure 4.9: The comparison between the absolute reference and modified COG velocity in the X direction with respect to the world frame.

In summary, the above results show that the ZMP-based controller is largely dependent on the user-defined reference trajectories. The biped is able to perform stable walking using the ZMP-based controller if the following conditions are met:

- The user-defined trajectories (or the intermediate locomotion objectives) are planned according to the internal and external constraints that are described in Section 3.2.2.

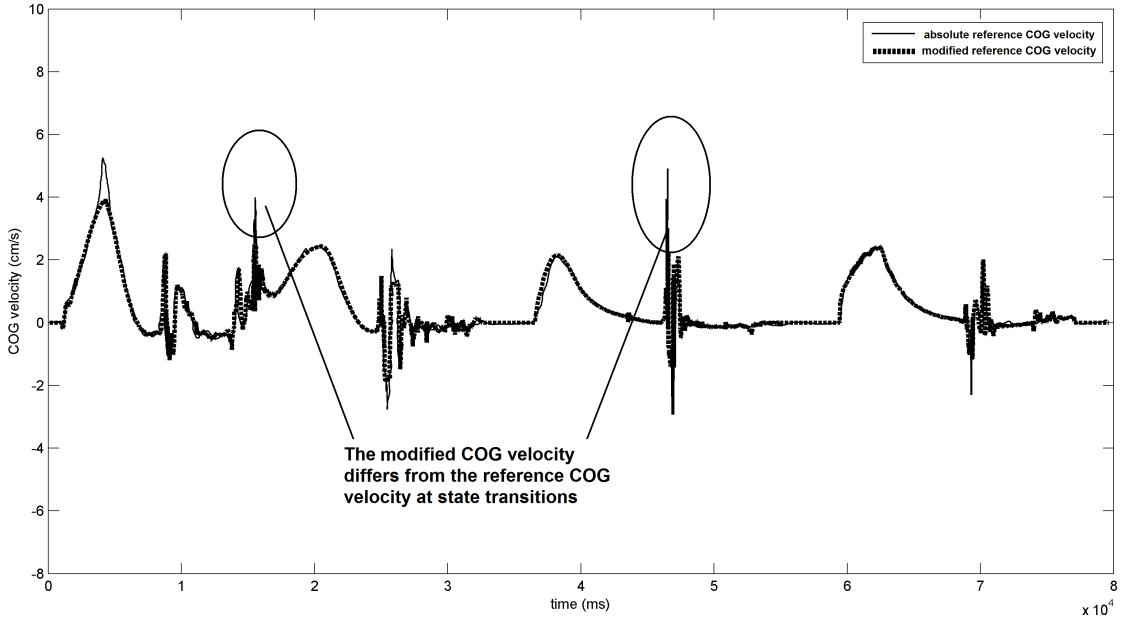


Figure 4.10: The comparison between the reference and modified COG velocity in the Z direction with respect to the world frame.

- The feedback controllers in the ZMP-based controller are able to track their respective reference trajectories and attenuate external disturbances.

The role of the *Dynamically Stable Gait Generation* block is actually less significant. The ZMP planner and the LIPM merely serves as a “saturation block” that limits the magnitude of the reference COG acceleration in order to ensure dynamic stability. For example, if the user-specified trajectory moves the COG *slowly* outside of the support polygon, the ZMP-based controller will not generate any compensatory motions to maintain static stability. Therefore, from the simulation results, we show that the ZMP-based controller relies heavily on stable reference trajectory planning to generate stable walking gaits.

The Pelvis Orientation Tracking

Next, we examine the pelvis orientation tracking of the ZMP-based controller. The comparisons between the reference and actual pelvis orientation are presented in Figures 4.11, 4.12 and 4.13.

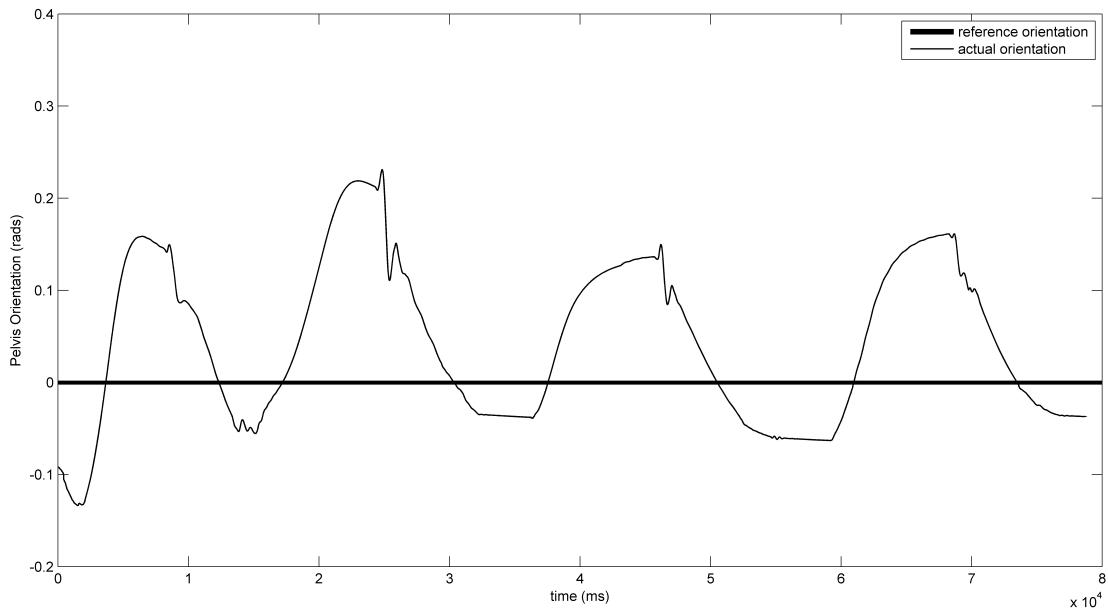


Figure 4.11: The comparison between the reference and actual pelvis orientation about the X axis with respect to the world frame.

The results show that in DB, the pelvis orientation was unable to remain at zero about the roll and pitch axis as the biped performed weight shift. In other words, the kinematic controller cannot achieve the secondary task of the *combined leg chain* and the error of the pelvis orientation tracking grew to a maximum value of about 11° at the end of DB. However, the simulation results show that these errors did not have significant effects on the walking gait and the dynamic stability of the biped in DB. Also, the pelvis orientation about the Y axis remained small (within 2°) throughout the walking cycle.

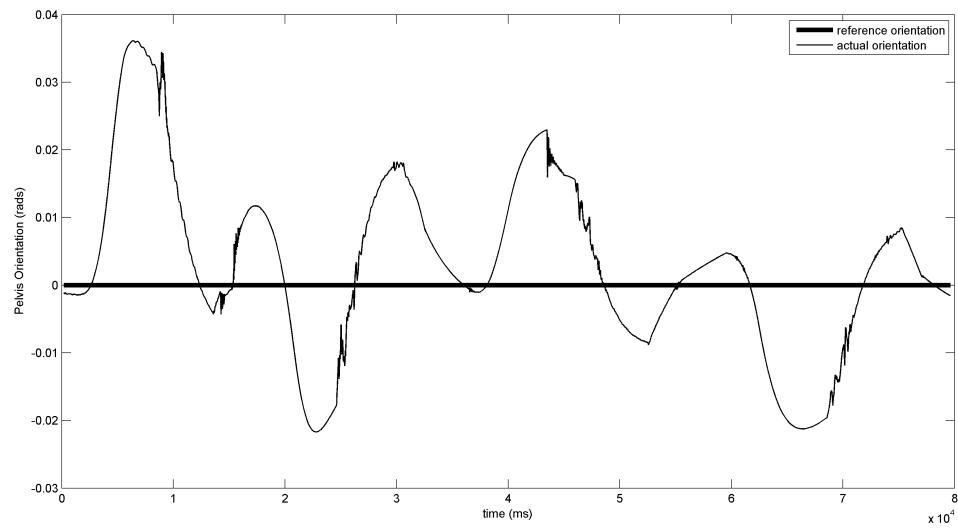


Figure 4.12: The comparison between the reference and actual pelvis orientation about the Y axis with respect to the world frame.

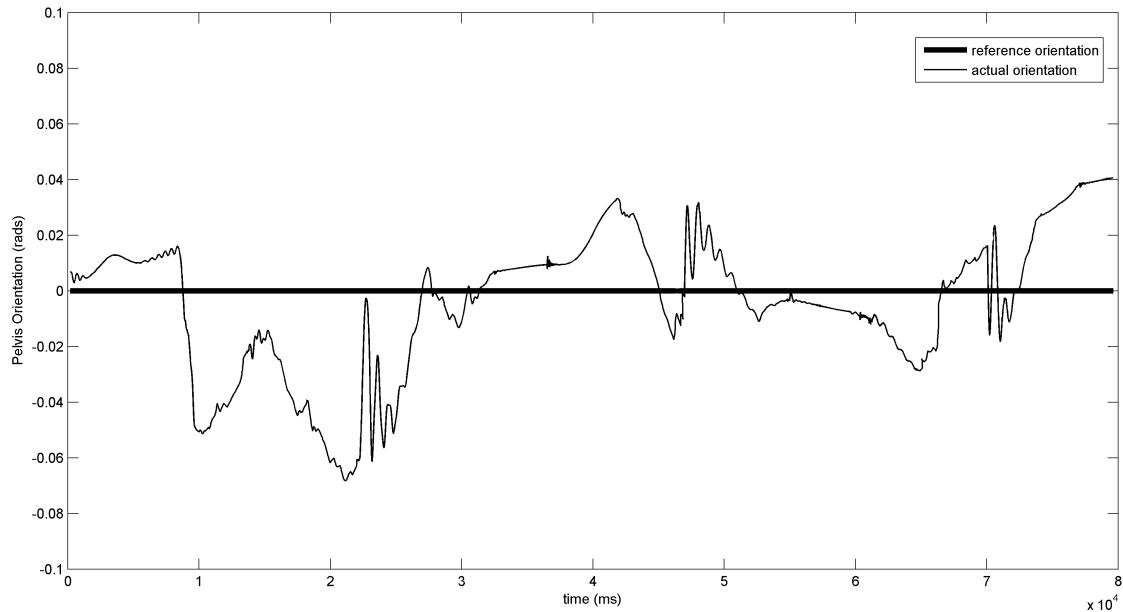


Figure 4.13: The comparison between the reference and actual pelvis orientation about the Z axis with respect to the world frame.

As shown in Figure 3.5.4, the task space errors in the pelvis orientation about the X and Z axis were corrected in SS. Unlike DB, the *Stance Leg Chain* in SS can fulfill both the primary and secondary high level tasks. Therefore, the kinematic controller generated angular velocity to reduce the pelvis orientation error axis towards the end of SS. (ex: the error of the pelvis orientation about the X axis was reduced from 11° to 3° at the end of SS).

The Swing Foot Position Tracking

As shown by Figures 4.14 to 4.19, the ZMP-based controller was able to accurately track the reference swing foot trajectory. Since the *Closed-loop Kinematic Controller* only controls the swing foot in SS, the reference swing foot trajectories only appear in SS in the result plots. The tracking error of the swing leg position was very small (less than 1 cm) throughout the walking cycle. Although there was a slight error (about 0.5 cm) in the lateral swing foot position at state transitions, it was quickly attenuated by the controller. In addition, although the stance foot was not explicitly controlled in DB, the results for the actual foot position (in the DB portions of the result plots) show that stance foot motions remained small. These movements were likely caused by the slight rotations in the stance feet during weight-shift. In summary, since the swing foot can be controlled accurately in all three translational directions by the ZMP-based controller, the biped can achieve accurate foot placements and potentially regain foot contact at the end of SS. The results also imply that if the biped cannot attain foot contact, it will likely be due to errors in the swing foot orientation rather than position.

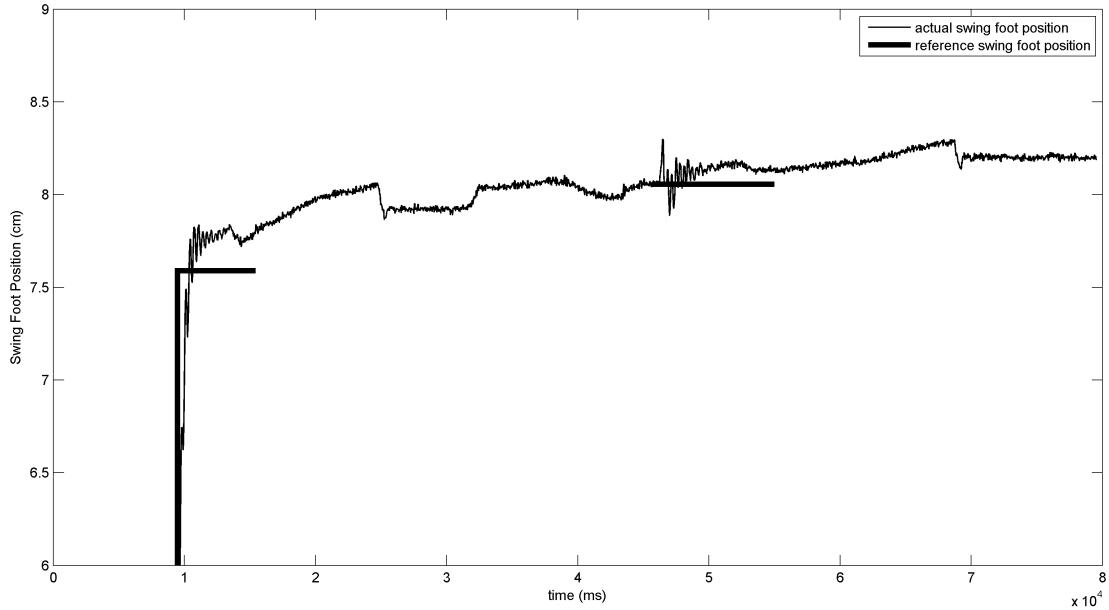


Figure 4.14: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the X direction (SSL).

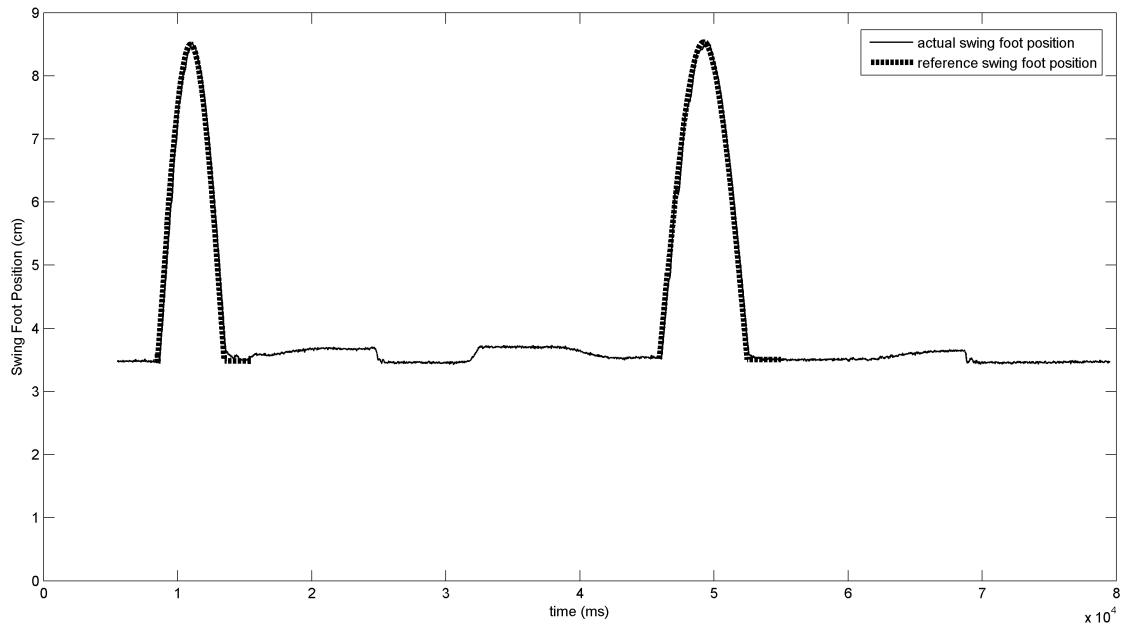


Figure 4.15: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Y direction (SSL).

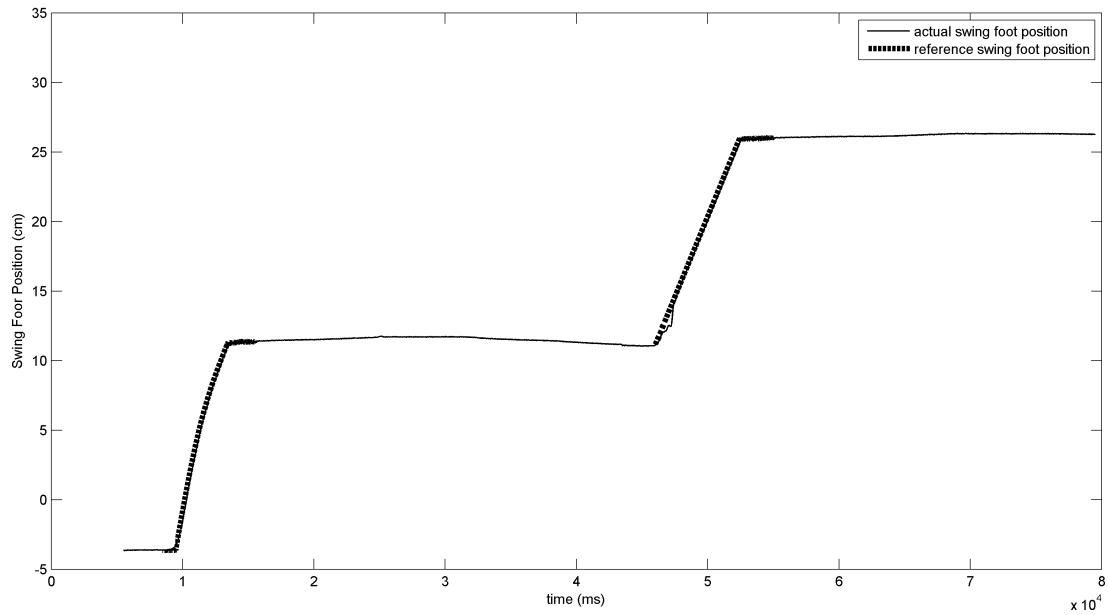


Figure 4.16: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Z direction (SSL).

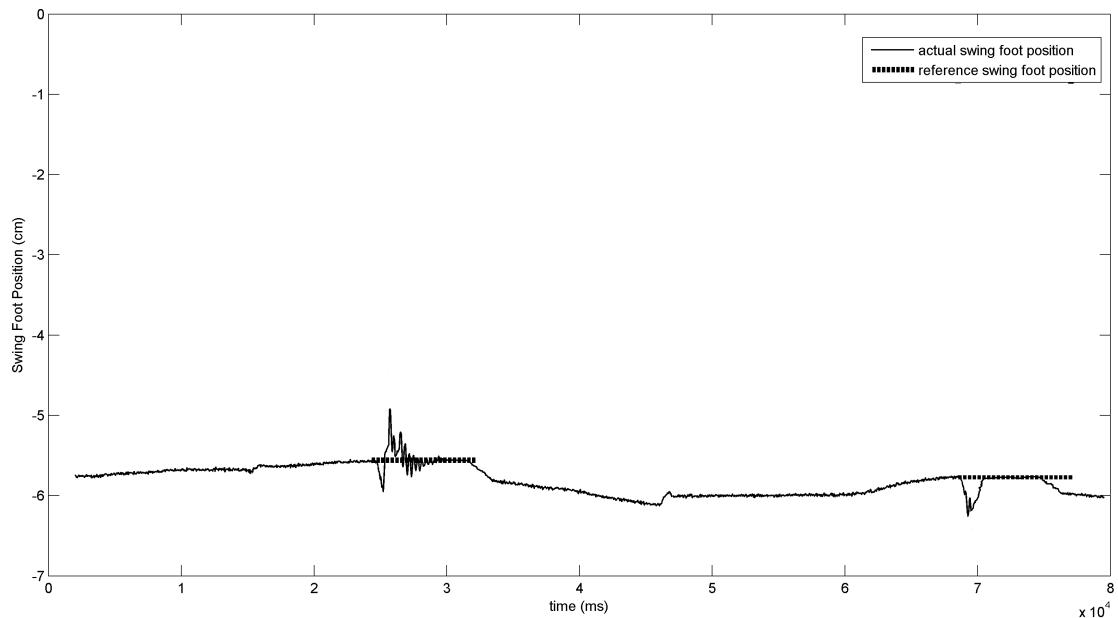


Figure 4.17: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the X direction (SSR).

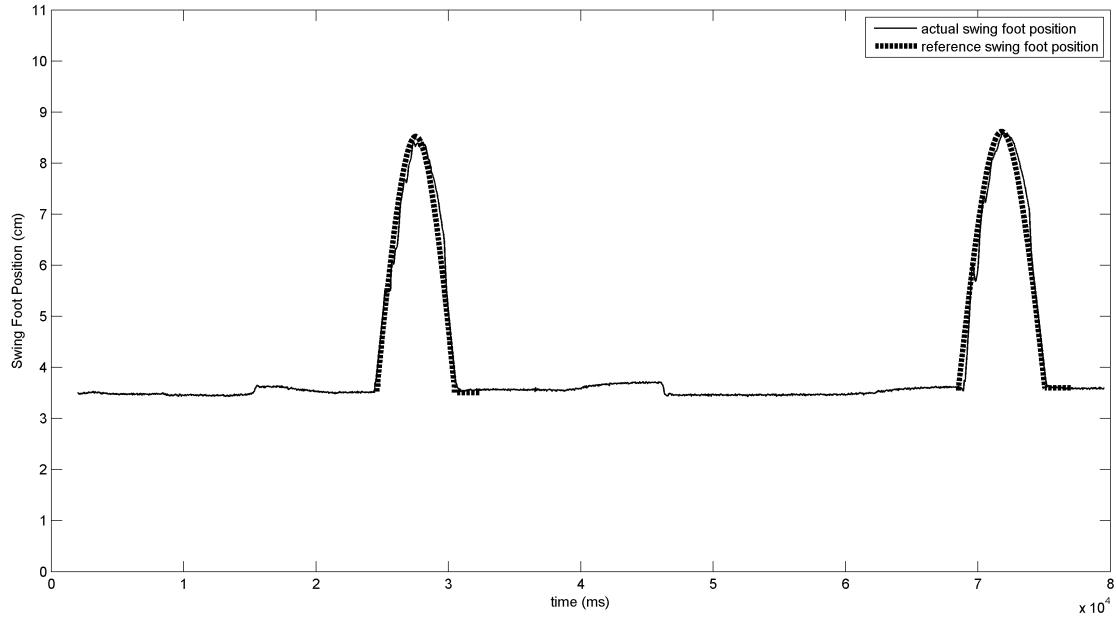


Figure 4.18: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Y direction (SSR).

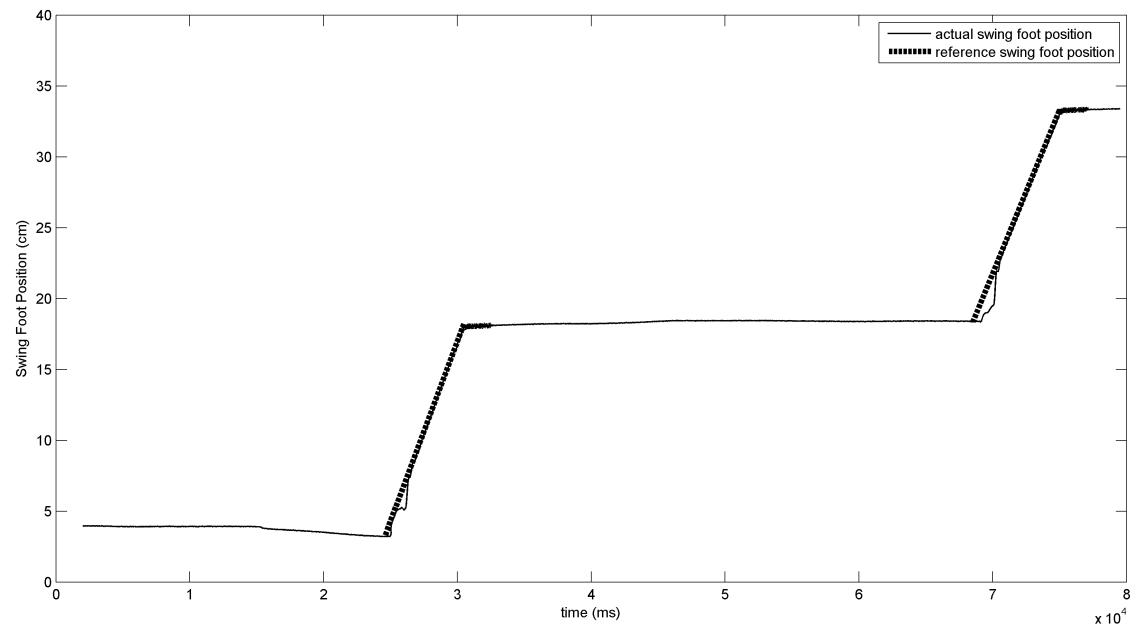


Figure 4.19: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Z direction (SSR).

The Swing Foot Orientation Tracking

The swing foot orientation tracking results are presented in Figures 4.20 to 4.25. The orientation about the X (roll) and Y (yaw) axis remained very small (within $\pm 3^\circ$ about the roll axis and $\pm 1^\circ$ about the yaw axis) throughout the walking cycle. The orientation about the Z (pitch) axis had a larger error (a maximum of $\pm 8.5^\circ$), which indicates that the angular motion of the swing foot about the pitch axis might be in conflict with the translational motion of the swing foot at certain parts of SS. However, in all three axes, the ZMP-based controller eventually generated compensatory angular velocities to bring the swing foot orientation to zero towards the end of SS. The abrupt convergence to zero for the swing foot orientation at the end of SS was likely enforced by the weight of the biped on the swing foot at touch-down.

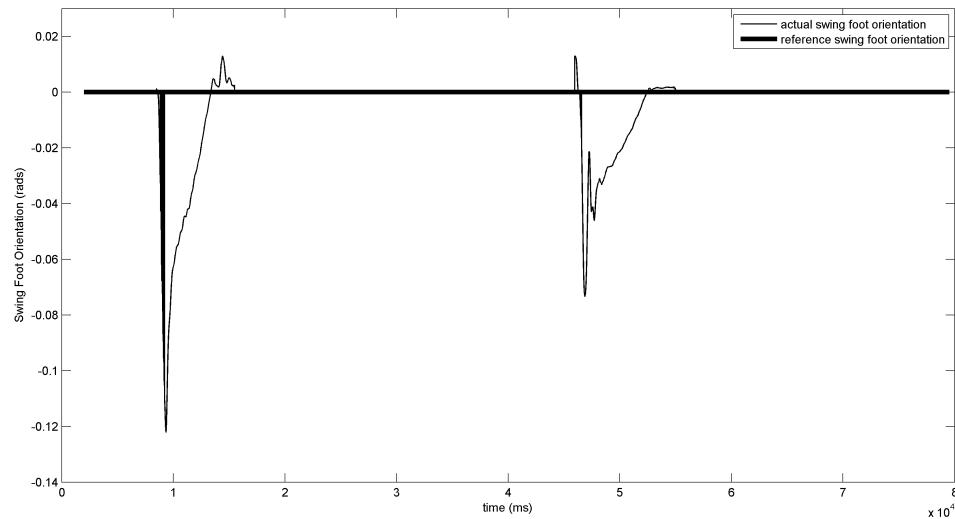


Figure 4.20: The comparison between the reference and actual swing foot orientation with respect to the world frame about the X axis (SSL).

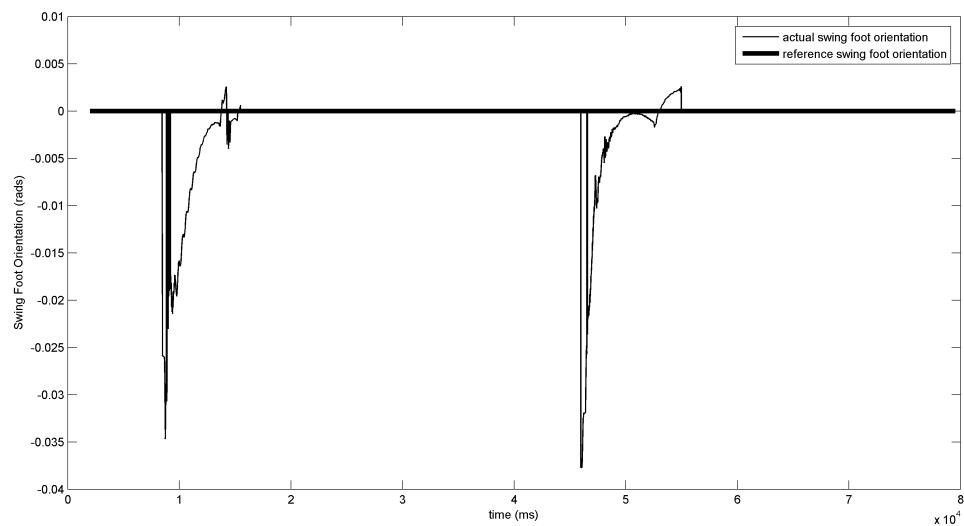


Figure 4.21: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Y axis (SSL).

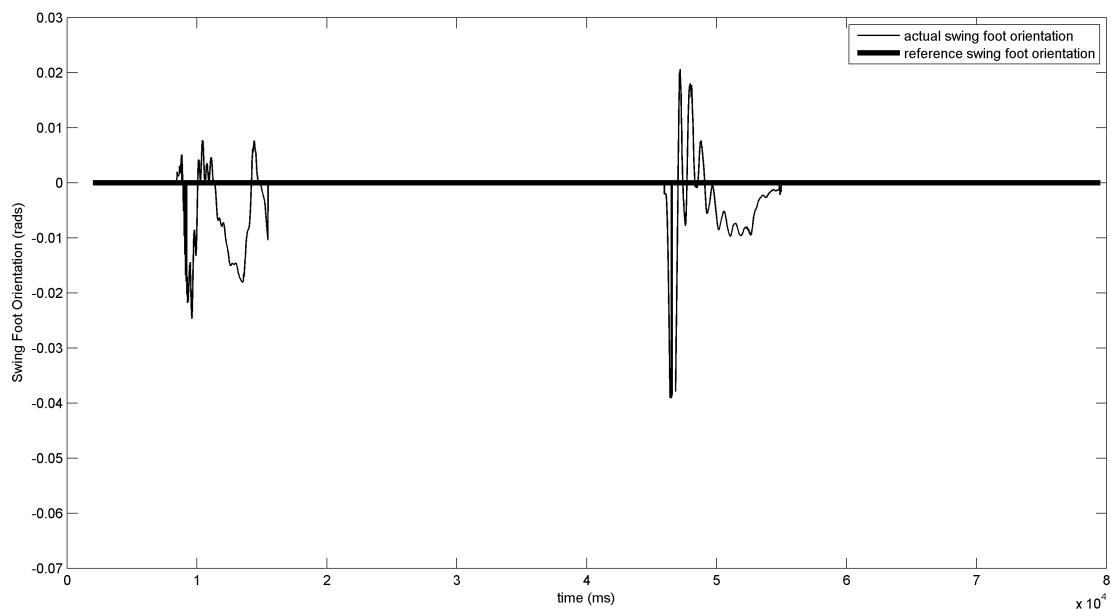


Figure 4.22: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Z axis (SSL).

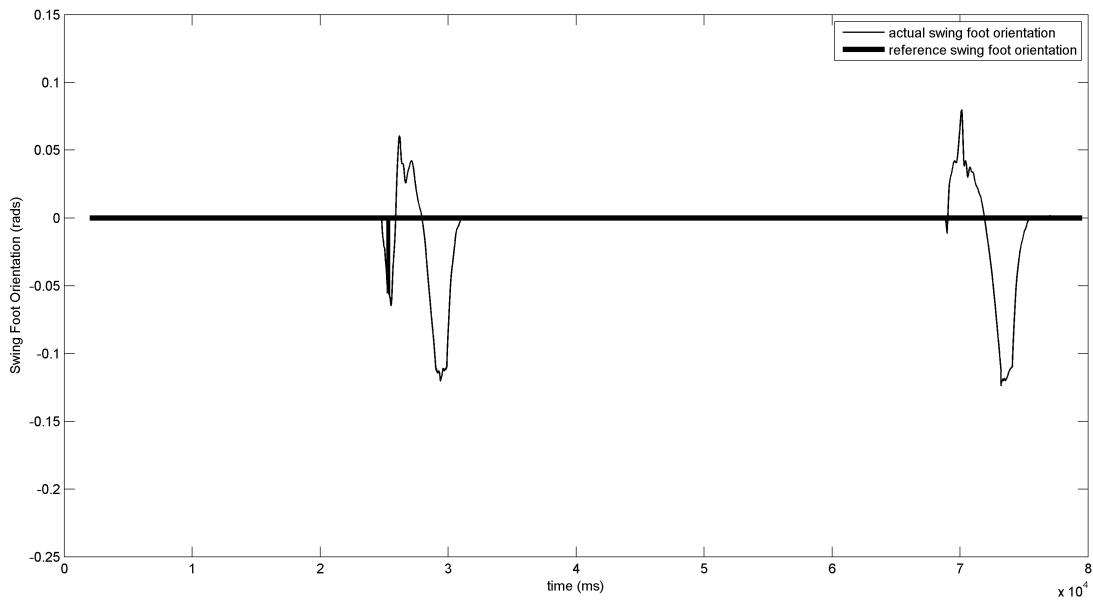


Figure 4.23: The comparison between the reference and actual swing foot orientation with respect to the world frame about the X axis (SSR).

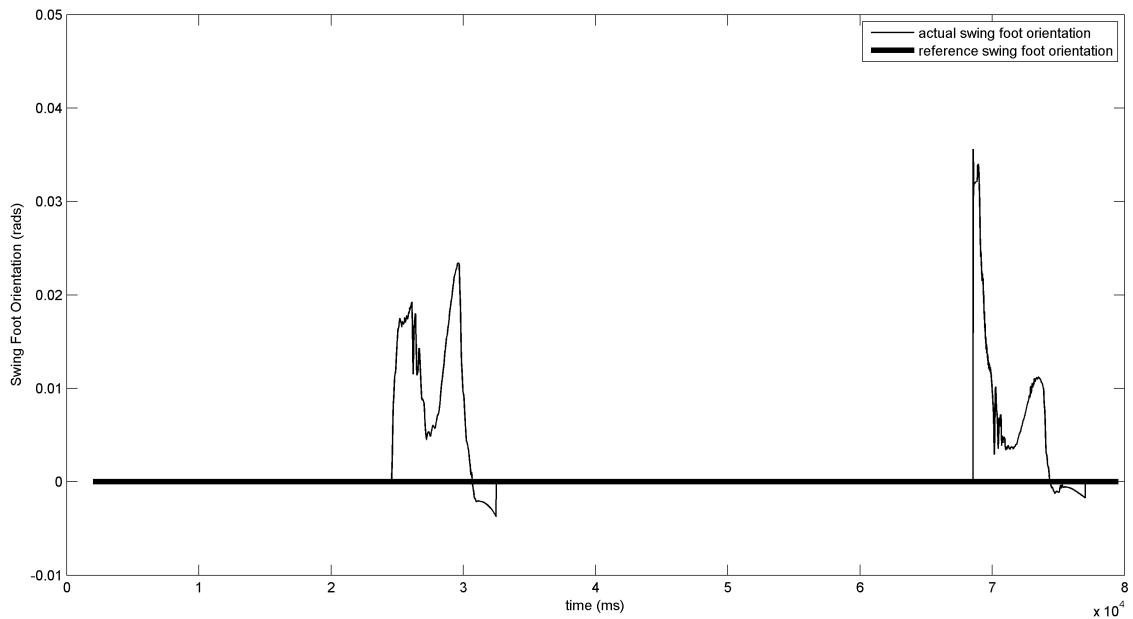


Figure 4.24: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Y axis (SSR).

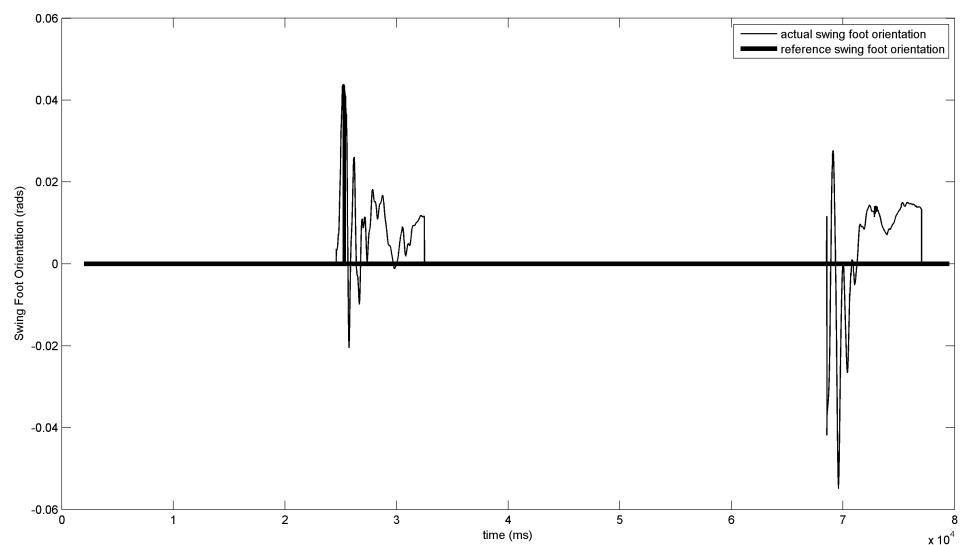


Figure 4.25: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Z axis (SSR).

In summary, the results from the walking tests show that the ZMP-based controller can generate and track stable reference trajectories. However, the walking velocity in ABL-BI was considerably lower (about $10\times$ lower) than other bipeds; in addition, the walking gait involves non-anthropomorphic *Torso Roll* motions that are not found in other bipeds. The reasons for this slow walking gait will be provided in the next section.

4.3 Limitations on the Walking Speed of ABL-BI

The walking performance of a robot controller differs considerably when it is applied to different bipeds. For example, although the ZMP-based controller generates a stable walking gait by adopting a large portion of the LIPM-based controller that is proposed in [55], the walking speed of ABL-BI is significantly lower than the speed that is reported in similar bipeds [17, 59, 60]. Also, the biped is not very robust to modelling errors and the abrupt changes in the reference trajectories and GRF at state transitions. From the simulation tests in *Webots*, we identified several factors that have decisive influence on the performance and stability of ABL-BI. Note that some of these factors are platform-specific, meaning that they only apply to our particular biped, while others are attributed to the limitations of the LIPM and the general shortcomings of the ZMP-based controller. The combined effects of these limiting factors are reflected in the simulation results and qualitative observations of different simulation trials.

Problems with the Mass Distribution of ABL-BI

As noted in [61], there are several factors that limit the maximum walking speed of bipeds: leg length, foot length, maximum joint velocity and torque and the

maximum foot traction force in the horizontal plane. These different factors can be encompassed into two limits for each step cycle [48]: 1) the maximum step length, and 2) the minimum step period. Therefore, in order to achieve fast walking speed, the walking controller should increase the step length and reduce the step period. In addition to the theoretical limitations in bipedal walking, there are specific limiting factors for the ZMP-based controller that are caused by the mass distribution of ABL-BI. In Table 4.3, the mass specification of ABL-BI is compared to other biped models and actual human data. These include biped models that are developed by Erbatur and Seven in [62], and Park and Kim in [11]). In addition, the mass distribution of a humanoid robot NUSIP II [63] and an average 10-year old child [64] are also included. One important difference is that the mass of ABL-BI is mainly concentrated in its legs whereas the mass for other biped models and humans is concentrated in the upper body. Unfortunately, this leads to two undesirable effects in the ZMP-based controller: 1) the motion of the upper body has less influence on the ZMP position than the motion of the legs, and 2) the unmodelled dynamics of the swing leg (which accounts for about 30% of the total mass compared to 20% in other bipeds) create large disturbances to the motions of the biped that are generated by the LIPM in SS.

4.3.1 Limiting Factors in Double Stance

We will first illustrate the effects of mass distribution by comparing the motions of ABL-BI with other bipeds in DB. Most stable walking gaits from existing walking controllers have a very short DB phase. As cited in [8], the DB phase accounts for less than 20% of the walking cycle. There are two main reasons for this:

- The objective of DB is to shift the weight of the biped (ie: manipulate the

Table 4.3: The mass distribution of ABL-BI, various biped models, a humanoid robot NUSBIP II, and a 10-year old child.

Link	link mass [kg]				
	Erbatur and Seven	NUSBIP II	Park and Kim	10-year old child	ABL-BI
Thigh	12.0 (13.0%)	2.4 (8.5%)	1.0 (6.2%)	4.1 (10.8%)	4.4 (19.9%)
Shank	0.5 (0.58%)	1.4 (4.9%)	1.0 (6.2%)	2.0 (2.0%)	1.3 (5.8%)
Foot	5.5 (6.3%)	1.2 (4.2%)	1.0 (6.2%)	2.5 (6.8%)	0.9 (4.1%)
Leg	18.0 (21.0%)	4.9 (17.0%)	3.0 (18.8%)	8.6 (22.9%)	6.55 (29.8%)
Lower Body	36.0 (41.0%)	9.9 (35.5%)	6.0 (37.5%)	17.2 (45.9%)	13.1 (59.7%)
Upper Body	50.0 (58.0%)	18.1 (64.5%)	10.0 (62.5%)	20.3 (54.1%)	8.85 (40.3%)
Total Mass	86.0	28.0	16.0	37.5	21.95

COG/ZMP position) from the support surface of the back stance foot to the forward stance foot. Generally, unless the stance feet are placed very far apart, weight-shift can be achieved easily by manipulating the torso and pelvis with the stance legs. Since most of the mass is in the upper body, a relatively small angular motion of the stance legs and translation of the torso is sufficient to shift the COG/ZMP to its desired position in a very short time. The support surfaces of the stance feet are usually smaller in the lateral direction and are separated by the width of the pelvis; therefore, the weight-shift motion mainly take place in the lateral direction. For this reason, some planar bipeds (with no lateral motion) in [49] assume instantaneous DB phase.

- In order to maximize the velocity of the biped, the legs must alternate their roles between swing and stance in the shortest time possible.

Problems with the Kinematics of ABL-BI

In contrary to other bipeds, ABL-BI has a very long DB phase that occupies close to 50% of the step cycle. The main reason is that the mass of the biped is concentrated in the legs instead of in the upper body. Rather than using the stance legs to

manipulate the pelvis and the torso over the forward stance foot, the stance legs themselves have to rotate significantly (especially in the lateral direction) in order to achieve weight-shift. In addition to that, the COG/ZMP of ABL-BI is required to move through a larger distance than other bipeds during weight shift. As shown in Figure 4.26, ABL-BI has a relatively wide pelvis (25 cm) and narrow feet (10 cm); therefore, the lateral distance (the predominant direction of motion in DB) between the initial and desired COG/ZMP position in DB is about 15 cm, which is larger than other bipeds.

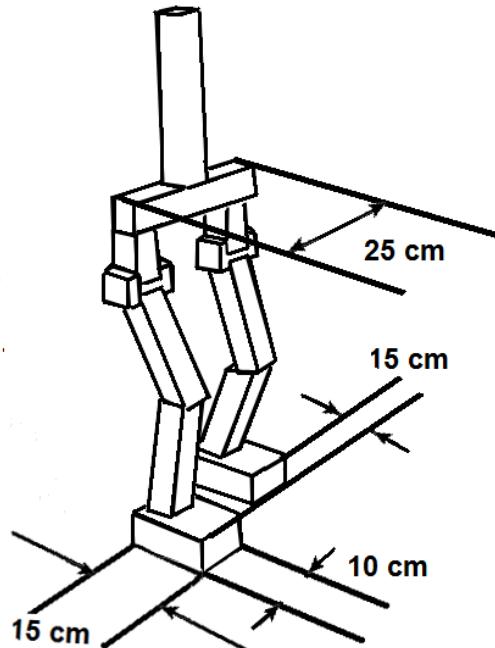


Figure 4.26: The pelvis and stance foot dimensions of ABL-BI.

For example, the biped in [65] has a narrower pelvis (16 cm); therefore, the required distance for weight-shift is also smaller (10 cm). As shown in the walking tests in [66,67], the lateral weight-shift distance for two humanoids: the RH-1 and the Sony

Qrio are also about 10 *cm*. Although the HRP-II humanoid has a reference lateral weight-shift distance of 15 *cm* [66] (same as ABL-BI); however, due to its high upper body mass, the hip roll and foot roll angles is only required to rotate about 12°. In contrast, as shown in Figure 4.27 and 4.28, the weight-shift of ABL-BI requires joint motions about the roll axis of about 0.35 *rads* (20°). Therefore, ABL-BI has to rotate its stance legs about 1.5 times more than HRP-2 to achieve the same weight-shift.

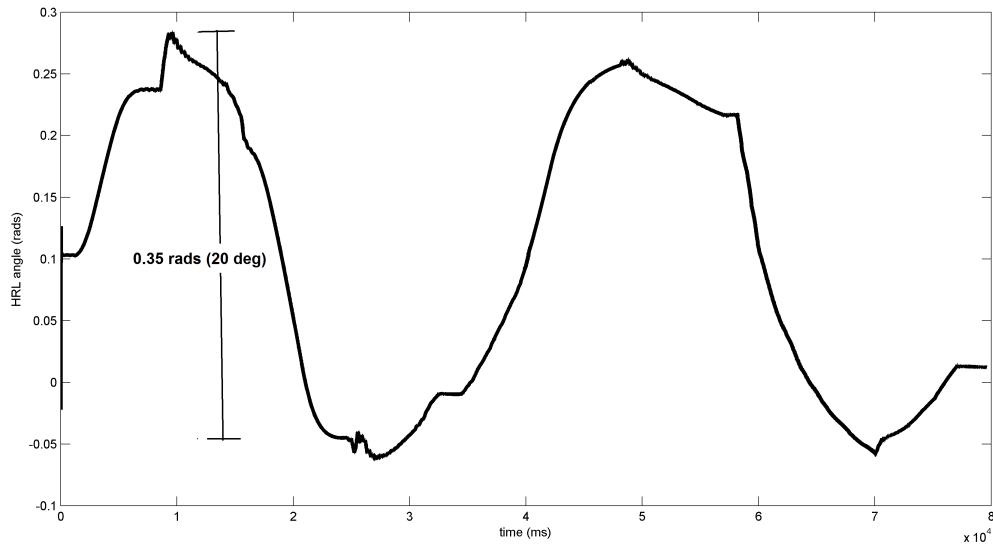
Problems with the Ground Model in Webots

From the above discussion, we pointed out that due to the mass distribution and the width of the pelvis/stance foot of ABL-BI, its required lateral weight-shift motion is larger than other bipeds. This in turn increases the DB period and reduces the walking speed of ABL-BI. However, the above factors still does not fully account for the extremely long duration of DB (about 9 – 10*s*). We purposely restricted the speed of ABL-BI in DB for two reasons: 1) the stance feet are unable to retain ground contact due to the stiffness of the ground model, and 2) the inability of the ZMP-based controller to handle large disturbances at state transitions. Let's examine these two factors more closely.

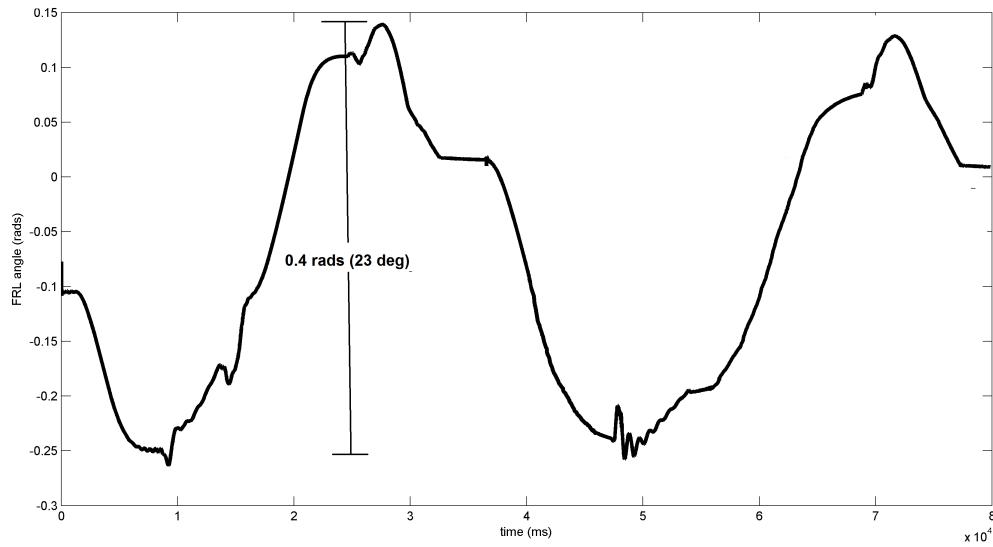
Recall from Section 2.2.4, the ground stiffness in *Webots* is dependent on two parameters:

Constraint Force Mixing (CFM): Determines the allowable error in the constraint of two rigid bodies.

Error Reduction Parameter (ERP): Determines the magnitude of the constraint force that corrects joint errors and realigns the constrained position of two rigid bodies that are connected by a joint.



(a) Joint angle of HRL in DB.



(b) Joint angle of FRL in DB.

Figure 4.27: Joint angles about the roll axis for the left leg in DB

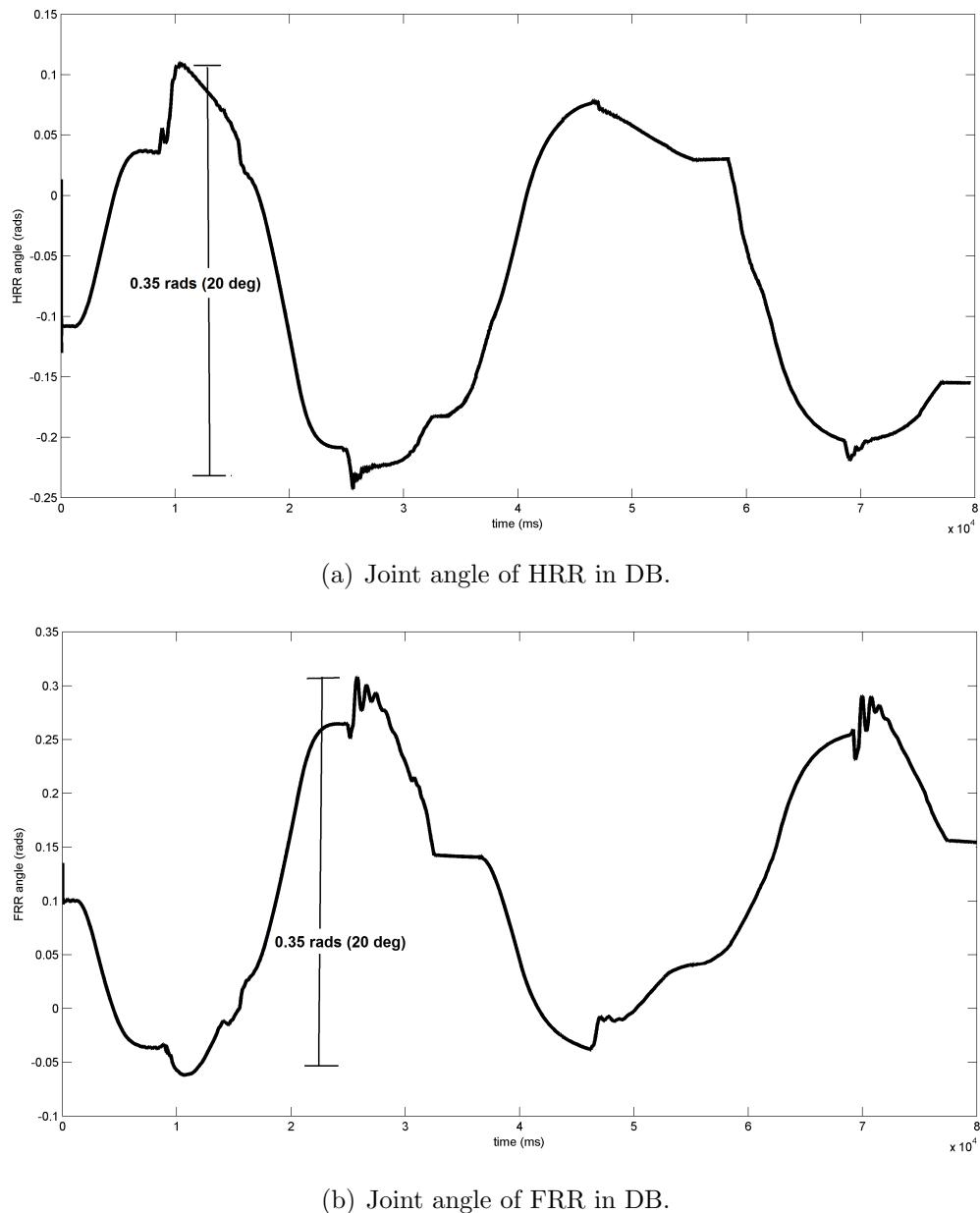


Figure 4.28: Joint angles about the roll axis for the right leg in DB

The parameters CFM and ERP from the Webots ground model can be used to calculate the proportional and derivative gains of the equivalent spring-damper model. Unfortunately, the ZMP-based controller was implemented in *Webots* 5.9.2 Educational Version, which does not allow the user to adjust the ERF and CFM. Therefore, the simulation of the ZMP-based controller uses an unrealistic stiff default ground model with very high proportional and derivative gains (2.5×10^8 and 8.0×10^4 respectively). As a consequence, the stance feet are subjected to very large GRFs and it is very difficult for the biped to retain a *surface contact* with the ground (left figure of Figure 4.29), especially during lift-off and touch-down. Since the penetration depth of the foot is small due to the high ground stiffness, the foot must be *perfectly* parallel to the ground surface in order to obtain full surface contact. However, this is a very stringent requirement for the controller since it has small, but unavoidable tracking errors. Added to this difficulty is the limited number of contact points that can be detected by the simulation. The maximum number of contact points in Webots varies with the type of the colliding primitives that are used to model the two colliding bodies. For the ZMP-based controller, we used a box primitive to model the foot and a plane to model the ground. Collision occurs when the two colliding primitives intersects. Unfortunately, the collision detection function in ODE can yield at most three contact points using the box-plane primitives. Therefore, due to the stiff ground contact and the limited number of contact points, the forward stance foot has a tendency to roll on its edge as the stance leg rotates in the lateral direction in DB. Qualitative observations from different simulation trials show that a large lateral acceleration will generate GRF that causes the forward stance foot to rotate significantly about the Z axis and have a *single edge contact* (right figure of Figure 4.29) at the end of DB. Without a firm ground contact, it is difficult for the biped the perform lift-off in a stable manner in the subsequent SS phase. Note that this problem is

a simulation issue that does not affect the weight-shift of the *actual* ABL-BI platform.

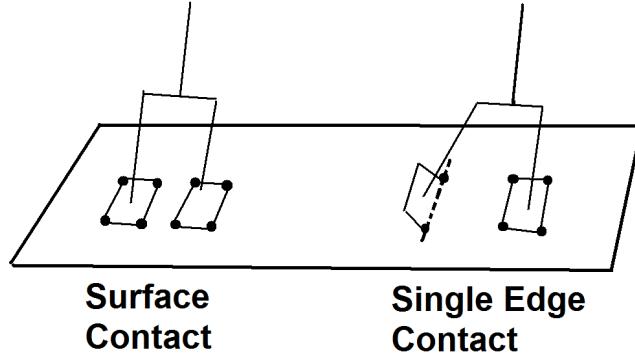


Figure 4.29: Types of foot contact. The dots represent contact points. The left figure show a biped with surface contact. In this case both feet are flat on the ground and the individual foot COP is within their respective support surfaces. On the right is a biped with single edge contact. One of the feet is not parallel to the ground and is supported by its edge. The COP for the edge support foot is outside its support surface.

Problems with the Control Architecture of the ZMP-based controller

Aside from poor ground contact, the ZMP-based controller also has to cope with disturbances in the state variables (ex: COG and ZMP) at state transitions that are caused by: 1) the abrupt changes in the reference COG velocity at lift-off and touch-down, 2) oscillations in the state leg that is induced by the dynamic effects of the swing leg at lift-off, and 3) the impact force at touch-down. As described in Section 4.1, the ZMP-based controller has a nested feedback loop with two sets of control gains: the first set is associated with the PD controller for the reference COG position and the second set controls the reference COG velocity. Unfortunately, this control architecture is one of the main drawbacks of the ZMP-based controller. The gains are manually tuned based on their observed effects in the simulation. The simulation trials indicate the following:

- The stability of the COG velocity loop (inner loop) is highly dependent on

the control gains and the output of the COG position loop (outer loop). For example, if there are significant changes in the control gains of the outer loop (by more than 50%) or in the reference COG velocity (output of the outer loop and input of the inner loop), then the gains of the inner loop must be changed in order to maintain the stability of the ZMP-based controller.

- From the equation of motion of the LIPM, a small change in the ZMP position induces a large change in the COG acceleration. Since the reference ZMP position is the output of the inner loop, the control gains of the inner loop has a large influence on the reference COG acceleration.

If the actual COG position significantly deviates from the reference position due to a very large disturbance, then it might be necessary to retune the control gains of *both* feedback loops. Since the reference COG acceleration is very sensitive to the reference ZMP position, an unstable output from the COG velocity loop can quickly cause the overall ZMP-based controller to become unstable.

Clearly, it is impossible to retune the control gains for every external disturbance. Therefore, the ZMP-based controller is not very robust and is only capable of attenuating small disturbances. To avoid gain scheduling, it is necessary to reduce the disturbance and the dynamic effects of the biped at state transitions. For this reason, the COG velocity is minimized at the end of DB and in the beginning of SS. However, a drawback is that due to the low velocity at state transitions, the controller has to maintain static stability and manipulate the COG over the support surface of the forward stance foot. In order to facilitate COG manipulation in the lateral direction and reduce the required stance leg rotation in DB, the *Torso* is rotated in the target lateral direction by about 70° at the start of DB (time-elapsed frames in Figure 4.30). As shown in the lateral COG position plot in Figure 4.31,

this rotation induces an additional COG movement of about 3 cm in the target lateral direction. Although rotating the torso is an non-anthropomorphic motion in a walking gait, it is more important to reduce the required lateral motion of the stance legs since it can cause rotation in the forward stance foot.

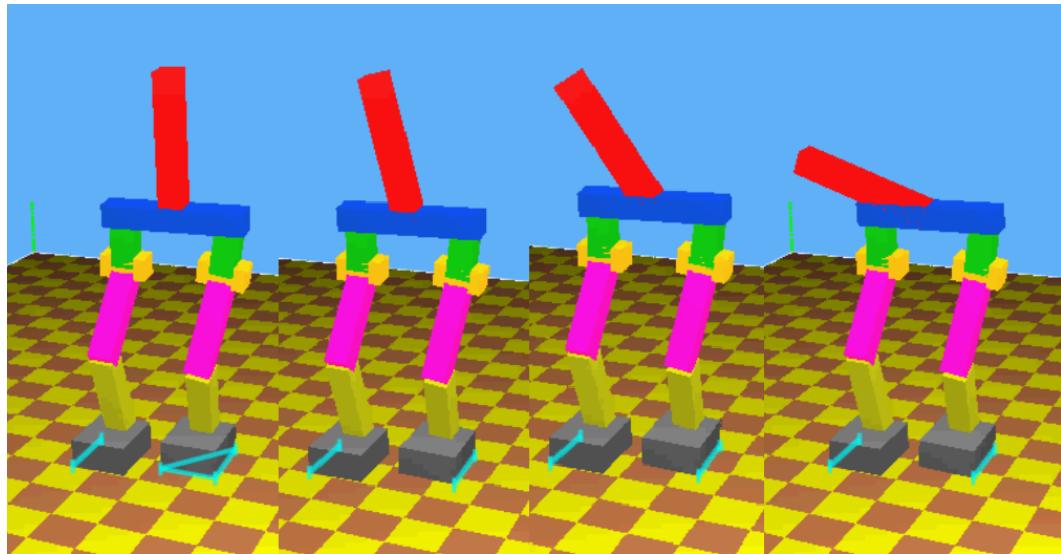


Figure 4.30: Time elapsed frames of the torso roll at the start of DB in Webots.

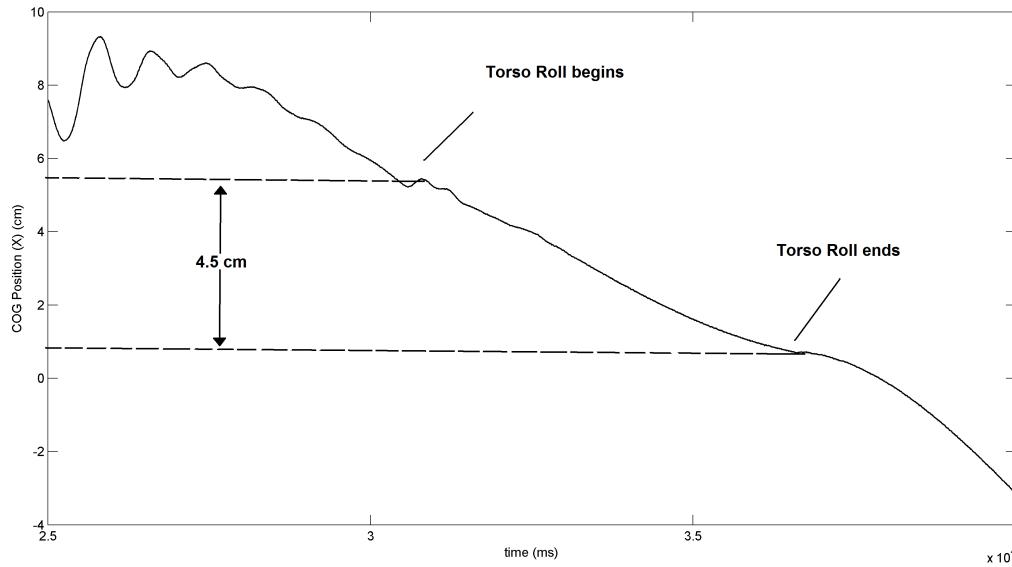


Figure 4.31: COG position (X) at the start of DB, showing the effects of the torso roll to the COG in the lateral (X) direction.

In summary, there are two limiting factors for the walking speed of ABL-BI that is associated with the mass and kinematic properties of the actual platform: 1) the mass distribution, and 2) the width of the pelvis and the stance feet. Both these factors increase the required motion of the biped in DB and therefore reduce the walking speed. The performance of the ZMP-based controller can be improved by modifying the physical properties of the biped in the simulation. However, in doing so, we would reduce the usefulness of the simulation results in validating the ZMP-based controller for actual platform implementation. Therefore, we refrained from modifying the simulated biped model as much as possible. Instead, the performance of ABL-BI can be improved by changing: 1) the parameters of the *Webots* ground model, and 2) the walking controller. The mass and kinematic specification of ABL-BI play a significant role in the performance of the ZMP-based controller; but at the same time, simulation results show that the ZMP-based controller is also unable to produce fast walking gaits and attenuate

large disturbances. In Chapter 5, we are able to design a simpler, but more robust controller that only use one set of control gains to control the reference COG velocity. In addition, by modifying the ground model and colliding primitives in *Webots* 6.2.4 Professional Version, the simulated biped in the ZRAM-based controller exhibits better ground contacts dynamics. Due to the improvement in the controller and ground model, the DB period of the ZRAM-based controller is reduced.

4.3.2 Limiting Factors in Single Stance

In addition to limitations in the DB phase, the walking speed of ABL-BI is also limited by the step length and step period in SS. For fast bipedal walking, it is desirable to have a large step length and small step period. However, for ABL-BI, there are limiting factors that place an upper bound on the maximum step length and a lower bound on the minimum step period. Let's examine each of these factors in turn.

Limitations in the Step Length

The step length in typical human walking speed is cited in [29] to be around 90% of the leg length. However, for ABL-BI, the step length is set to 15 *cm* (same as the length of the foot), which is only 22% of its leg length. The main reason is that unlike humans, the mass of the swing leg in ABL-BI has a large influence on the ZMP position of the biped. In SS, the ZMP must be kept within the support surface of the stance foot in order to maintain dynamic balance. As shown in Figure 4.32 and 4.33, moving the swing leg in the sagittal direction by about 15 *cm* will result in a forward ZMP movement of about 3 *cm*. Note that in this simulation trial, the COG is purposely left uncontrolled in order to isolate and observe the dynamic effects of the swing leg. Since the ZMP typically starts at about 2 *cm* ahead of

the center of the stance foot in SS and the length of the foot is 15 cm, the final COG position at the end of SS will be about 2.5 cm away from the boundary of the support polygon. This 2.5 cm stability margin serves as a “buffer region” for the biped in case of external disturbances at the end of SS. Therefore, although the step length of ABL-BI is much lower than its human counterpart, it is planned based on the stability constraint of the biped.

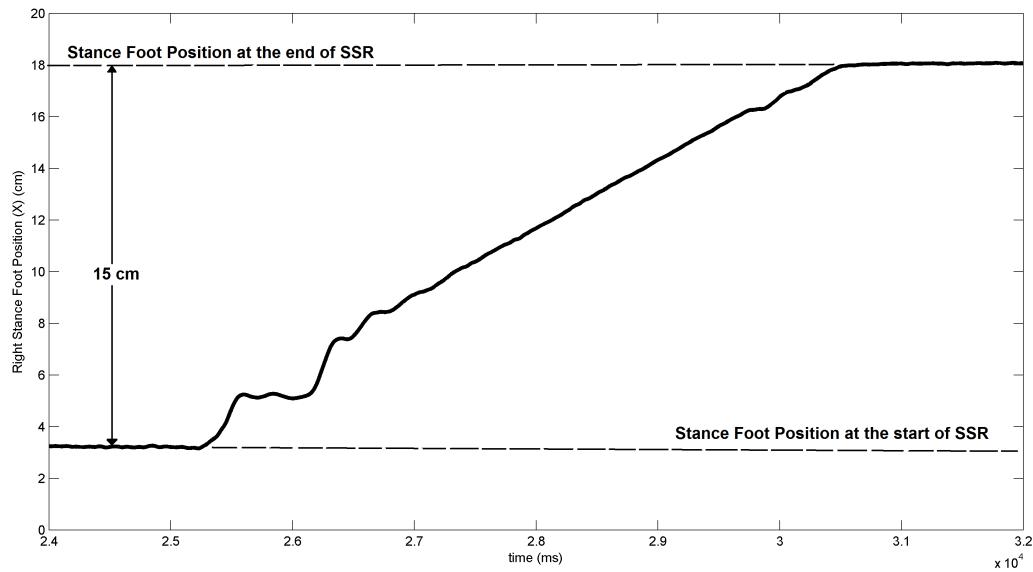


Figure 4.32: The sagittal swing foot position (Z) (origin of Frame 6) with respect to the stance foot frame (Frame 12).

Limitations in the Step Period

Typically, in other bipeds, the main limiting factor for the swing period are internal constraints such as: 1) maximum motor torques in the swing leg, 2) control bandwidth, and 3) leg inertia. Although these factors also apply to ABL-BI, simulation results show that the primary limiting factor is the unmodelled dynamics of the swing leg. A discussion of these results will be presented below.

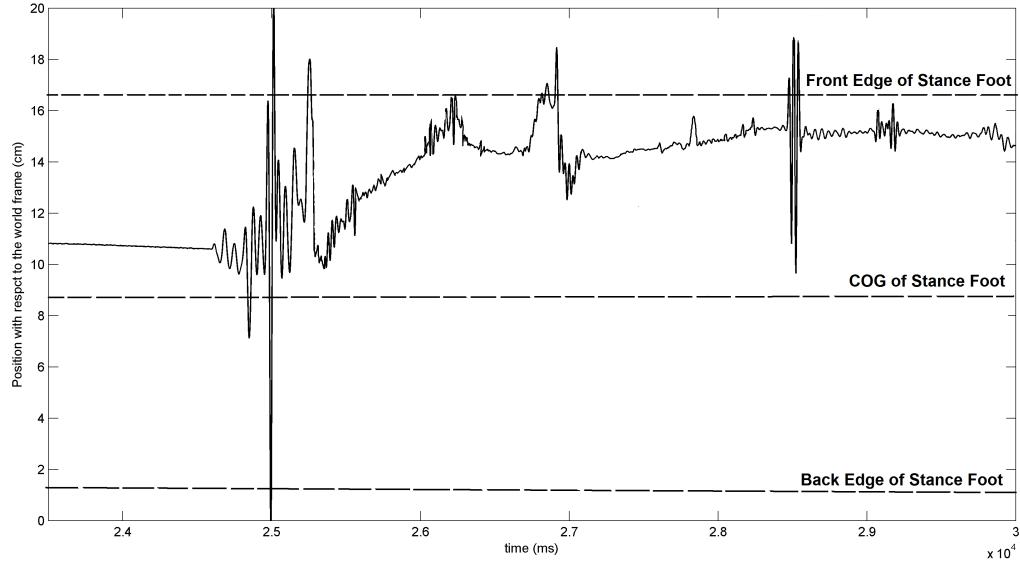


Figure 4.33: The ZMP position (Z) with respect to the world frame in SSR.

The main assumptions of the LIPM are 1) there are no centroidal moment throughout the walking cycle, which also implies that the line of action of the GRF passes through the COG , and 2) the legs are massless links. However, as pointed out in Section 3.6.3, these assumptions will be violated if the swing leg has a large linear/angular acceleration or if the swing leg has a large inertia. Since the mass of the swing leg is large (about 30% of the total mass of the biped), it is possible to for the unmodelled dynamics of the swing leg to induce large disturbances to the COG and ZMP during SS. For this reason, three simulation trials were performed to examine the effects of the swing leg dynamics in the sagittal plane (note: the swing leg motion takes place mostly in the sagittal plane in stable walking). In addition, in order to study the effects of the swing leg motion, *only* the reference trajectory of the swing foot was modified between each simulation trial.

In the first trial, the step period of the reference swing leg trajectory was set to 2200 ms (only the Z direction of the reference trajectory is shown in Figure

4.34), which is about 33% of the actual step period that is used in the ZMP-based controller (6500 ms). As shown in Figure 4.35 and 4.36 (COG position in the sagittal plane), the actual COG diverged from the reference COG in the middle of the SS, indicating that the inertial effects of the swing leg caused the COG to drift outside of the support polygon and the biped to fall.

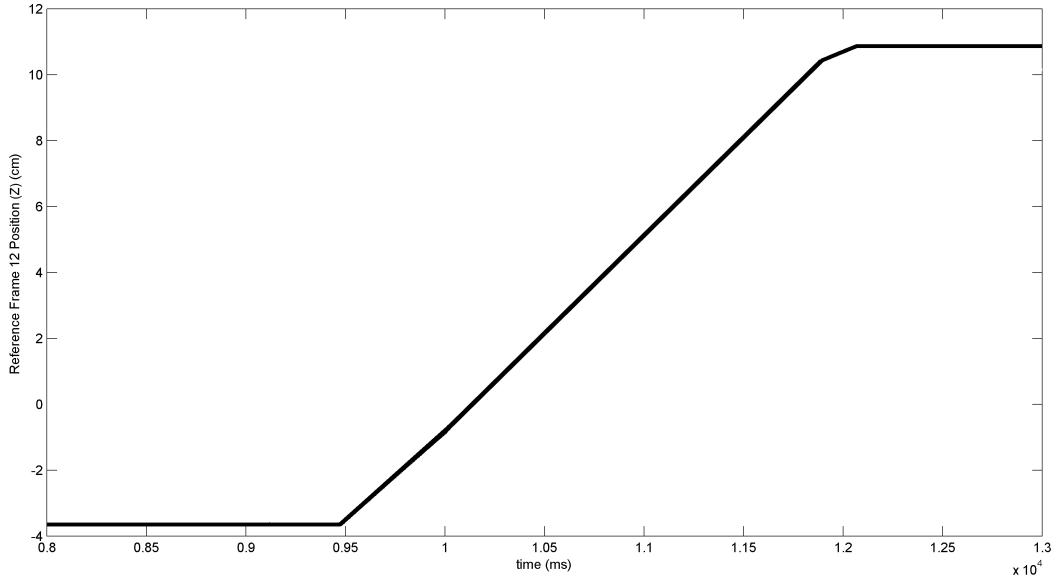


Figure 4.34: Swing leg inertial effect test 1: the reference swing foot trajectory in the Z direction in SSL.

In the second trial, the period of the swing leg trajectory was increased to 3000 ms (Figure 4.37), which is about 45% of the step period in the ZMP-based controller. In this trial, the disturbance was smaller (an initial overshoot of about 6.5 cm) and the feedback control of the ZMP-based controller was able to reduce its effects, but not enough to prevent the biped from landing on the edge of its foot. As indicated in Figure 4.38, the swing foot had a non-zero orientation about the roll axis at touch-down (at around 12500 ms). Without a firm ground contact, the biped falls in the subsequent DB phase as shown by the actual COG trajectories in Figure 4.39 and 4.40.

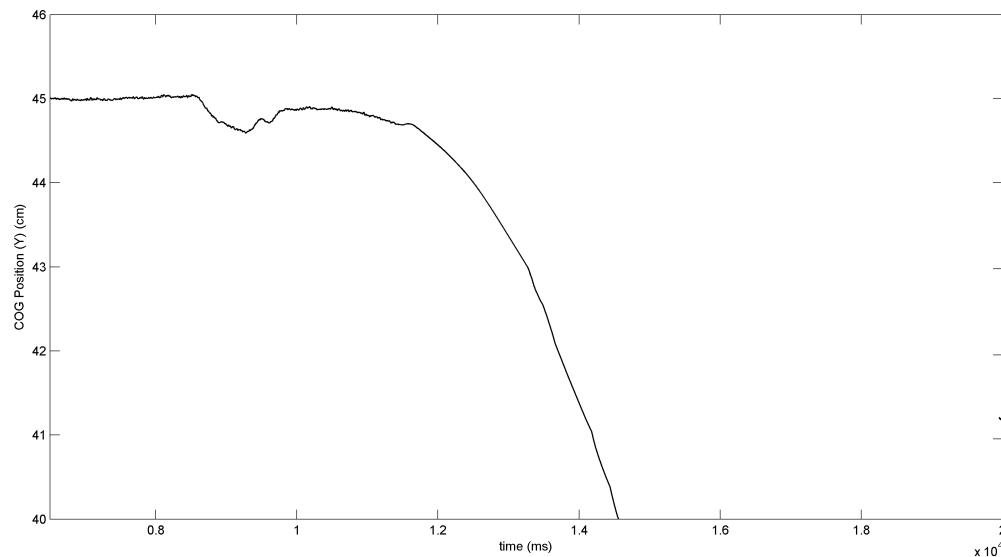


Figure 4.35: Swing leg inertial effect test 1: the reference and actual COG position in the Y direction.

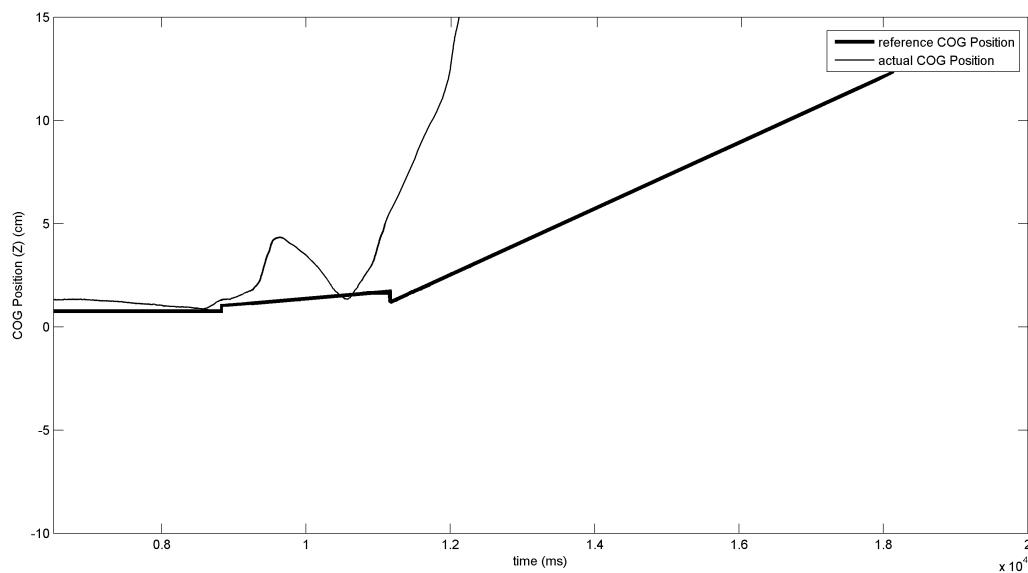


Figure 4.36: Swing leg inertial effect test 1: the reference and actual COG position in the Z direction.

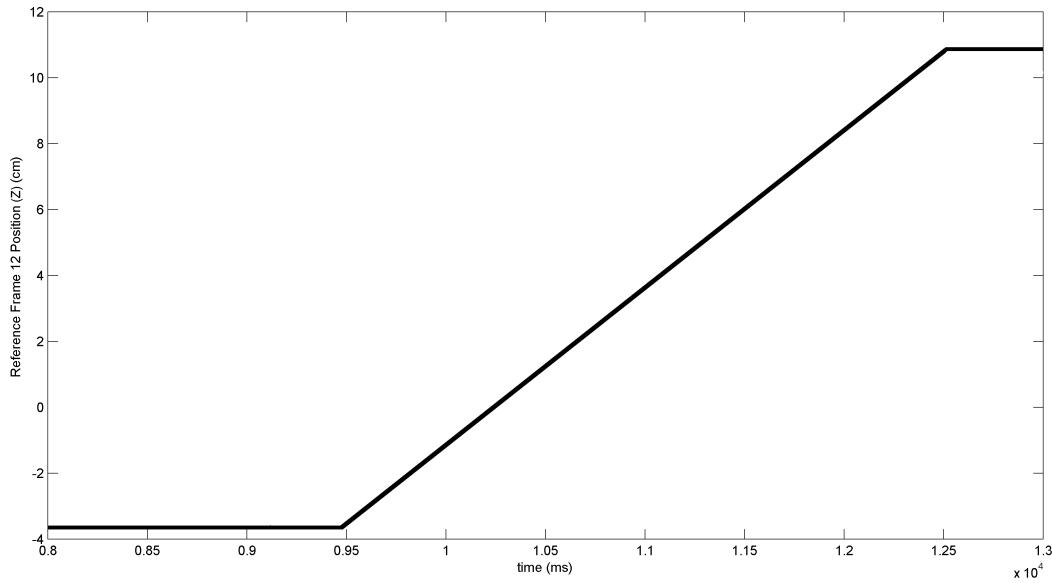


Figure 4.37: Swing leg inertial effect test 2: the reference swing foot trajectory in the Z direction in SSL.

Finally in the third trial, the step period was set to 6500 *ms* (Figure 4.41), which is equal to the step period that is used by the ZMP-based controller. As shown in the reference COG tracking plots in Figure 4.42 and 4.43, although the swing leg acceleration still caused a disturbance in the middle of SS, the feedback control was able to attenuate the disturbance and maintain the dynamic balance of the biped.

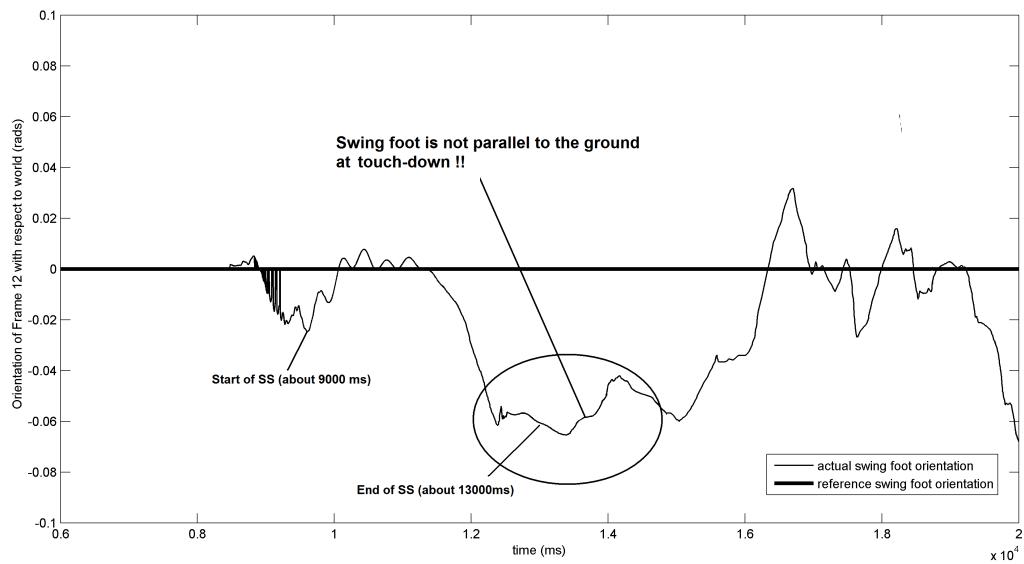


Figure 4.38: Swing leg inertial effect test 2: the orientation of Frame 12 (swing foot in SSL) about the Z axis with respect to the world.

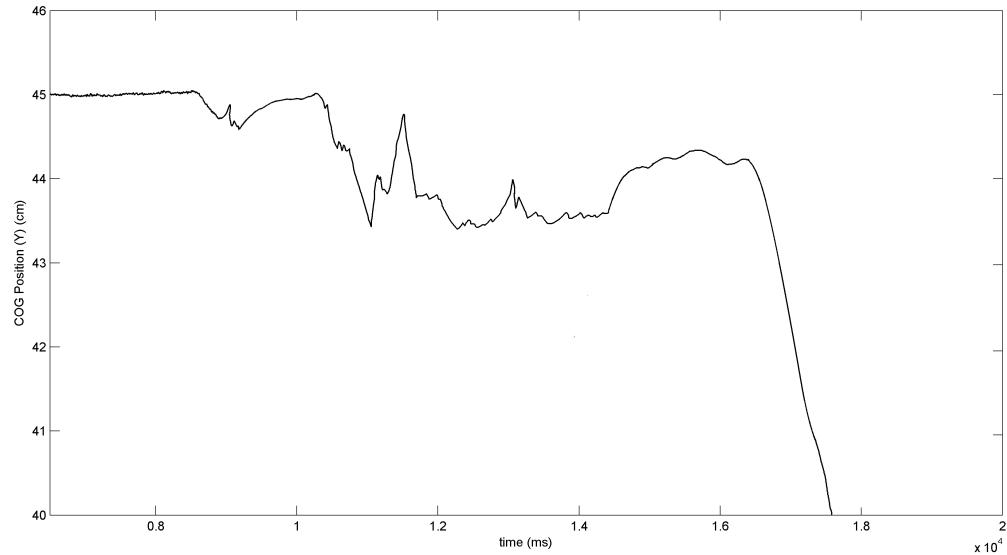


Figure 4.39: Swing leg inertial effect test 2: the reference and actual COG position in the Y direction.

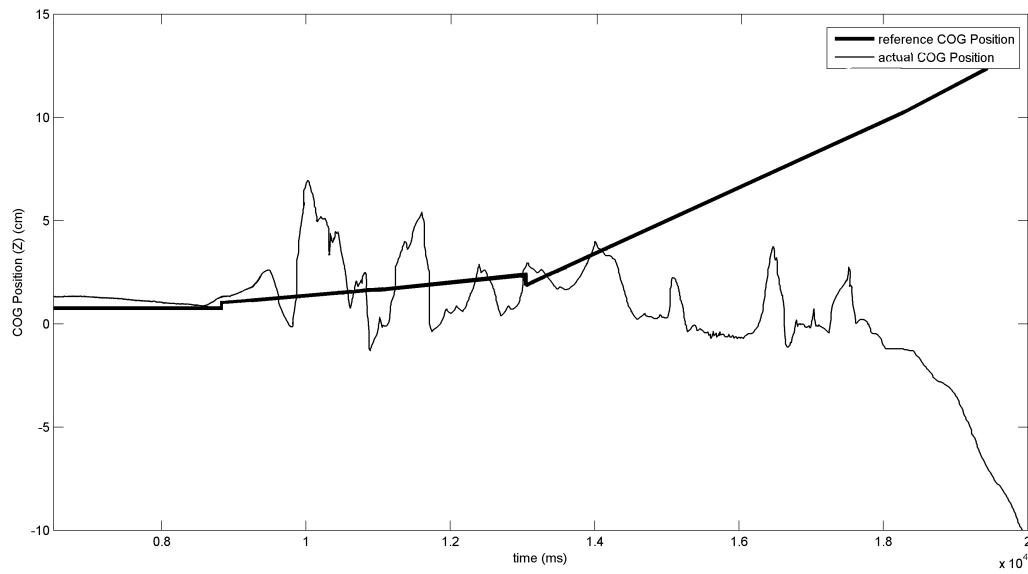


Figure 4.40: Swing leg inertial effect test 2: the reference and actual COG position in the Z direction.

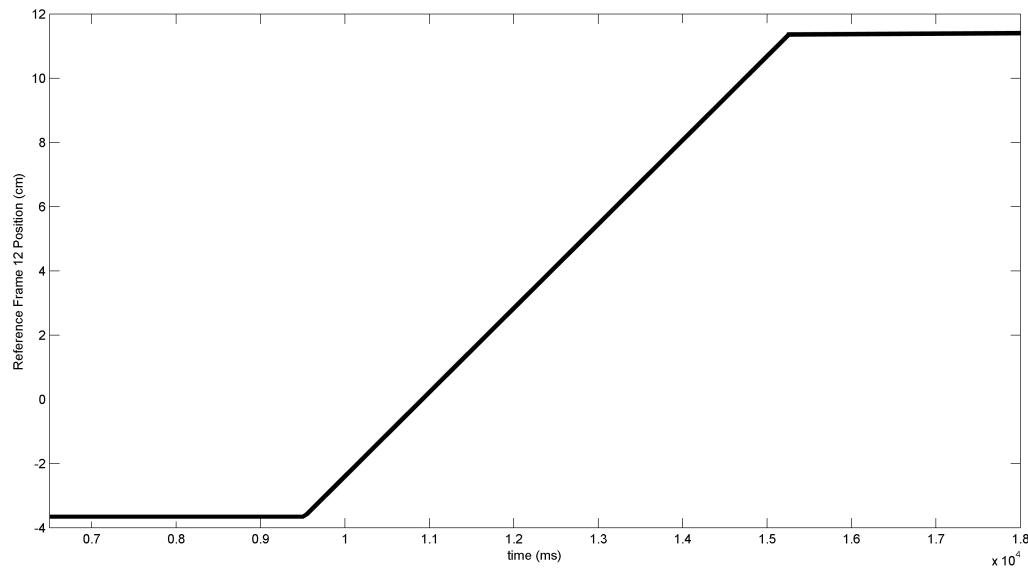


Figure 4.41: Swing leg inertial effect test 3: the reference swing foot trajectory in the Z direction in SSL.

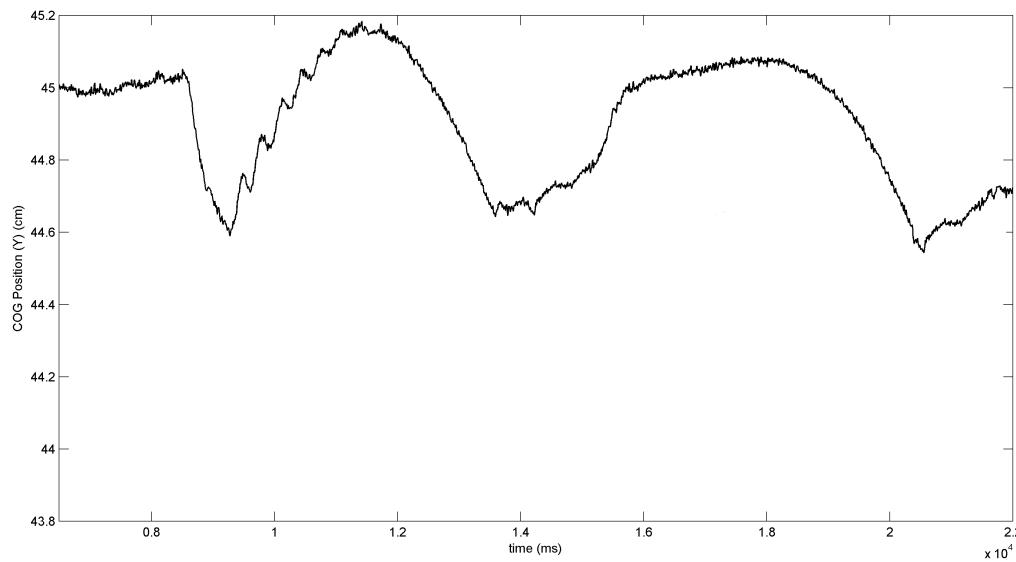


Figure 4.42: Swing leg inertial effect test 3: the reference and actual COG position in the Y direction.

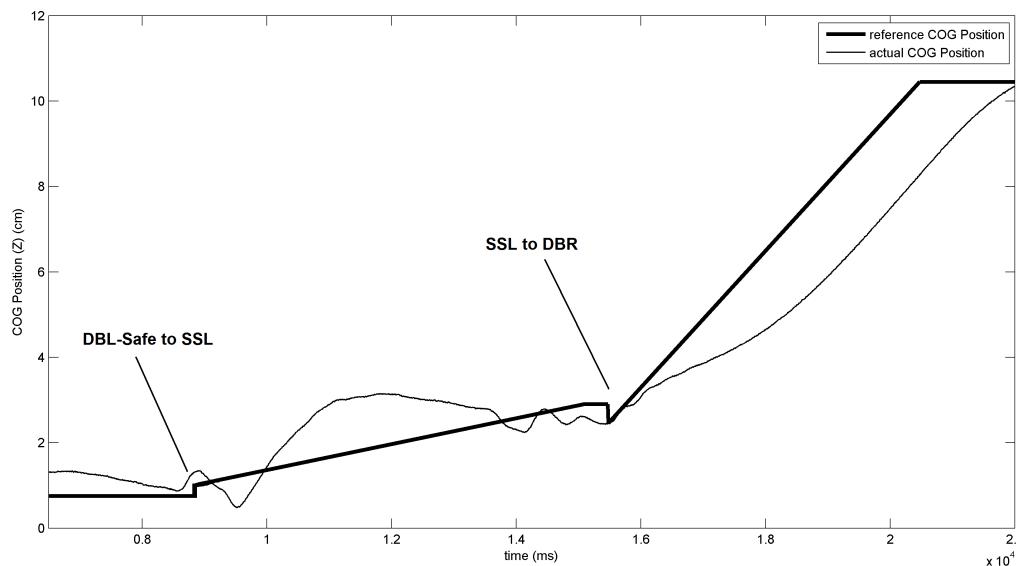


Figure 4.43: Swing leg inertial effect test 3: the reference and actual COG position in the Z direction.

In summary, the above simulation trials show that the inertial forces of the swing leg cause large disturbances to the COG in SS. In order to maintain stable walking, the disturbance from the swing leg must be reduced to a level that can be handled by the feedback controller in the ZMP-based controller. This can be done by three methods:

1. *Reduce the mass of the swing leg:* this reduces the gravitational and inertial effects of the swing leg. However, since the end-goal of the simulation tests is to approximate the effects of the ZMP-based controller on the actual platform. We avoided changing the mass properties of ABL-BI as much as possible.
2. *The use of a modified LIPM:* In [11,62], the authors proposed a two-mass LIPM with one mass representing the stance leg and the torso and the other representing the swing leg. Using this model, the rotational equilibrium equation was derived about the ZMP to generate stable walking trajectories. The results from [11] show that the two-mass model was able to reduce the modelling errors in SS and capture the dynamics of the swing leg. Although this model is not presently used, it might be useful to apply this model in the future to improve the performance of the ZMP-based controller.
3. *Reduce the velocity of the swing leg:* this method also reduces the inertia effects of the swing leg and is used by the ZMP-based controller.

Recall that due to the high stiffness of the ground model and the difficulty in tuning the control gains in the ZMP-based controller, the biped is prone to large disturbance. However, in Chapter 5, we were able to reduce the swing period to about 5500 ms due to the improved ground model and control architecture of the ZRAM-based controller. The simulation results in this section reaffirm the limitations

of the LIPM that is noted by various authors in [11, 49, 64] and verify that the inertial effects of the swing leg is the primary limiting factor of the step period in ABL-BI.

Finally, walking tests were conducted for two simulated bipeds in *Webots* 6.24 Professional Version. The first biped is the ABL-BI model that was used to validate the ZMP-based controller. The second biped is a *modified* ABL-BI model that has the following changes from the original model:

- The mass distribution of NUSBIP II [63] was applied to the modified ABL-BI model. While the total mass remained the same, the modified biped has a heavy upper body (14.2 kg) with relatively light legs (3.7 kg).
- The parameters of the ground model were modified to produce a softer ground. Similar to the values that were used in [68], the proportional term k_p was set to 30000 and the derivative term k_d was set to 2000.

In order to examine the combined effects of the limiting factors that were identified in this section, the lateral COG position of the original biped is compared to the modified biped in Figure 4.44.

By increasing the damping of the ground model, the modified biped was able to maintain surface ground contact and achieved faster weight-shift. As shown in Figure 4.44, the DB phase was shorten by $1000 - 2500\text{ ms}$ in comparison to the original biped. In addition, due to the reduced inertia effects of the swing leg, we were able to plan a faster swing leg trajectory for the modified biped and the step period was reduced by $3500 - 4500\text{ ms}$. By reducing the time period for DB and SS, the modified biped was able to take one more step than the original biped within the same period of time (30000 ms). Therefore, the results show that the mass distribution of the biped

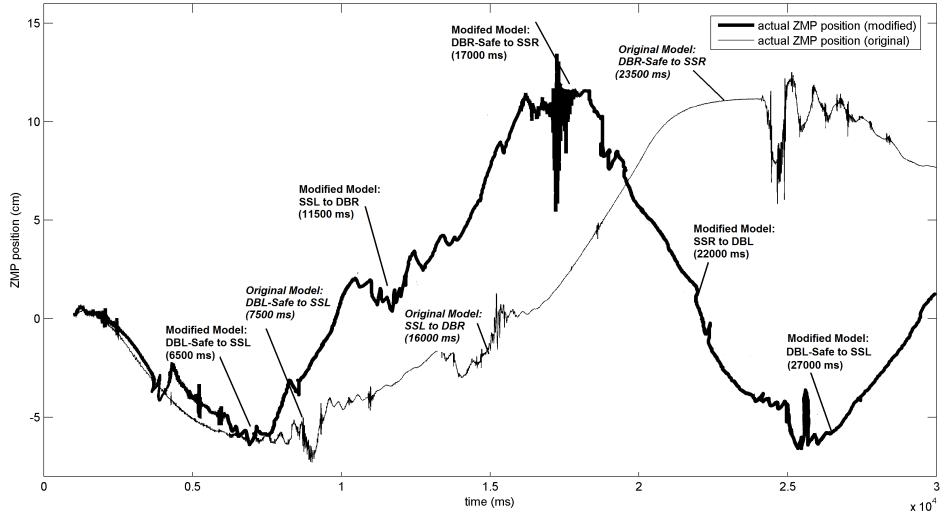


Figure 4.44: A comparison of the lateral ZMP position between two ABL-BI models, one with the original mass distribution and the other with a modified mass distribution.

and the ground model influence the performance of the ZMP-based controller and the walking speed of the biped.

4.4 Summary

In this chapter, an existing ZMP and LIPM-based controller (with slight modifications) was implemented on a simulated model of ABL-BI in *Webots*. The main purpose of the walking test for the ZMP-based controller is to: 1) gain insights to the physical limitations of ABL-BI (ex: find out the maximum velocity and the maximum lateral weight-shift distance of ABL-BI), 2) tune the control gains and walking parameters of ABL-BI, and 3) validate the performance of the *closed-looped kinematic controller*. The walking test results show that the ZMP-based controller was able to track the COG and swing leg reference trajectories accurately and generate stable walking gaits. ABL-BI was able to perform walking in the simulation, which is an improvement to the stepping motion that is

reported in the original controller [55]. It was also observed that the stability of the ZMP-based controller is largely dependent on stable reference trajectory planning.

The second focus of this chapter is to examine the factors that limit the velocity of ABL-BI in DB: 1) mass distribution, 2) the width of the pelvis and stance feet, 3) ground contacts, and 4) control gains. A short DB period can be achieved by adjusting the parameters in the ground model and improving on the design of the controller. Due to the high ground stiffness and the limited number of contact points from the plane-box colliding primitives, it is difficult to maintain surface ground contact. In order to prevent the forward stance foot from rotating about its edge at the end of DB, the reference COG velocity is reduced in the ZMP-based controller. In Chapter 5, we will implement the ZRAM-based controller in *Webots* 6.2.4 Professional Version, which allows the user to modify the ground stiffness and colliding primitives. By using a softer ground and box-box primitives, the biped has better ground contacts and the DB period is reduced from 10000 *ms* to 5000 *ms*. Another factor that limits the speed of ABL-BI in DB is the control gains of the ZMP-based controller. Simulation trials show that it is difficult to find constant control gains that provide satisfactory control for all situations in the walking biped. In the presence of large disturbance, the control gains need to be retuned in order to maintain the stability of the ZMP-based controller; however, it is impractical to find a set of gains for every possible situation for the controller. A solution is to limit the COG velocity near state transitions in order to minimize the disturbance at lift-off and touch-down. Admittedly, this is a very limiting solution for the ZMP-based controller; however, in Chapter 5, we were able to simplify the control architecture of the controller so that the reference COG trajectory is determined by only one PD controller and increase the robustness of ABL-BI.

For SS, the step length is limited by how far the swing leg can move without disrupting the balance that is maintained by the stance leg. The step length for the ZMP-based controller was selected to be 15 cm based on its effects on the ZMP and the stability margin that it provides in SS. The step period is limited by inertia effects of the swing leg. It was observed from simulation trials that a large swing leg acceleration can disturb the COG and cause the biped to fall. Therefore, in order to minimize the effects of the unmodelled dynamics, the acceleration of the swing leg is reduced, or equivalently, the step period was increased to 6500 ms for the ZMP-based controller.

Chapter 5

A Walking Controller Based on CMP Planning and COG Position Control

As the main contribution, this chapter develops a stable walking controller that is based on CMP planning and COG position control. High level motion planning of the biped is done by planning a reference CMP trajectory that lies within the support polygon. In order to ensure rotational stability, the controller regulates the rate of change of angular momentum of the biped (or equivalently, the distance between the COP and the reference CMP) through COG manipulation. A reference COG trajectory is first generated from the ZRAM model (Section 3.6), and then decomposed into joint velocities via a kinematic control scheme that is described in Section 3.5.3. Note that the ZMP-based controller and the ZRAM-based controller share a common control architecture (Figure 3.68) with a modular component: the *Dynamic Stable Gait Generation* block. Essentially, the ZMP-based controller and the ZRAM-based controller are distinguished by the stability criterion and reduced dynamic models that they use to facilitate real-time walking control. Therefore, the main focus of this Chapter will be on the *Dynamic Stable Gait Generation* block that generates a dynamically stable reference COG trajectory based on the ZRAM criterion and a reference CMP input. We will also briefly review other components of

the robot controller that are presented in Figure 3.68 (described in detail in Chapter 3), but more in the context of how their roles contribute to overall walking control of the biped. The ZRAM-based controller is first tested in a planar robot model as an initial proof-of-concept and then verified in a 3-D simulated biped in *Webots*. The main contribution of the ZRAM-based controller is that it offers a simple but effective method in generating stable walking gaits in real-time by exploiting the relationship between the CMP, GRF, and the COG.

This Chapter is organized as follows: Section 5.1 present a brief overview on the motivations and theoretical background of the ZRAM-based controller. Section 5.2 presents the control algorithm of the ZRAM-based controller, with special focus placed on the *Dynamic Stable Gait Generation* block. The simulation test results in the planar and the 3-D simulated robot are presented in Section 5.3 and 5.4 respectively. Additionally, an in depth discussion of each set of test results is presented. Finally, Section 6 summarizes the important results of the simulation tests and compares the ZRAM-based controller to a related bipedal walking controller.

5.1 Motivations and Theoretical Background

5.1.1 The ZRAM Stability Criterion

Although the LIPM is an useful abstraction of the complex dynamics of the biped, it is not based on the biomechanical principles of human walking and therefore cannot be used to generate biologically realistic gaits. An alternative approach in developing bipedal walking controllers is to study the biomechanics aspects of human walking. From these studies, we can identify key control objectives and strategies that human use to achieve walking and apply some of these elements in a biomimetic control

algorithm. Bipedal robots in turn can be used to verify the conclusions that are drawn from the biomechanical studies and identify new research objectives. Thus, the above example illustrates the reciprocal benefits between the study of human walking and the development of robot controllers that is first stated in Section 1.1.

One of the important result from biomechanical studies is that the spin angular momentum remain a conservative quantity despite the fact that the legged system (human or biped) is *not* an isolated system throughout a walking cycle (the GRF acts at the stance foot). Therefore, it has been asserted that one of the control strategies humans use to achieve dynamic stability is the active regulation of the spin angular momentum [35, 56]. In [35], the author suggests that the biped can achieve dynamic balance by regulating the centroidal moment ($\dot{H}_G = 0$) and minimizing the spin angular momentum. For this purpose, the ground reference point CMP is developed in [3] as a means of measuring and controlling the rotational stability of the biped at every time step (Section 1.2). In addition, the rotational equilibrium of the biped is expressed mathematically by a nonlinear coupling between the COG, CMP and the GRF in Equation 3.65. Therefore, a dynamically stable COG trajectory can be generated from the rotational equilibrium equation using 1) the reference CMP that defines the pre-planned high level motions of the biped, and 2) the actual (measured) GRF.

5.1.2 Related Control Methods

Applying the ZRAM stability criterion above, the authors in [35, 57] suggested several possible control methods that can achieve rotational stability; however, a controller that directly uses CMP with COG manipulation to generate stable walking with

disturbance compensation has not yet been realized. Kajita et al. [40] proposed a *Resolved Momentum Control* that directly controls the angular and linear momentum of the robot. The high level motion of the biped is planned using the reference linear momentum and the dynamic stability of the robot is maintained by tracking a zero reference angular momentum. In a similar work, the authors in [34] planned a reference COG trajectory and reduced the computational requirement of the controller by only using upper body motions to regulate the angular momentum of the robot. Komura et al. [12, 69] used a simplified dynamic model to explicitly calculate the additional angular momentum that is generated by external disturbances. A modified COG trajectory is then calculated to produce a counteracting moment that negates the additional angular momentum. Although this is a compact approach, a solution may or may not exist depending on the magnitude of the disturbance and the current states of the robot. Aside from walking, angular momentum regulation has also been applied to specific locomotion tasks such as turning [38], making an emergency stop [70], and taking a step to attenuate large disturbances [71]. In recent years, different simplified dynamic models that are based on the spin angular momentum have been developed. For example, in [72], an extension of the LIPM called the *Reaction Mass Pendulum* (RMP) is introduced to explicitly model the whole body inertia and the angular momentum of the biped. A similar model called the *Linear Inverted Dumbbell Model* (LIDM) approximates the rotational dynamics and the angular momentum of the robot [59]. However, walking controllers that are based on these newer angular momentum models have not yet been formulated. Although various control strategies based on angular momentum regulation and the ZRAM criterion have been suggested in the past, the ZRAM-based controller presented in this thesis is the first real-time walking implementation that exploits the relationship between CMP and COG.

5.1.3 Angular Momentum in Bipedal Locomotion

Based on the biomechanical findings of human movements in [53], the authors argue that ZRAM only ensures constant spin angular velocity and therefore is insufficient in maintaining the dynamic stability of the biped. Since a *constant non-zero* spin angular momentum can tip the biped over, the controller must also minimize the spin angular momentum along with the centroidal moment. In addition, their biomechanical findings support the fact that humans employ two different control strategies to maintain balance:

1. *ZRAM mode*: During normal walking and small rotational movements, the human subject regulates the centroidal moment and uses a strategy that is based on the ZRAM criterion.
2. *non-ZRAM (Disturbance Compensation) mode*: Under large external disturbances and rotational movements, the spin angular momentum of the human subject increases significantly. In this case, the human uses a control strategy that generates a non-zero centroidal moment to attenuate the additional spin angular momentum in the system and move the COP and COG towards the center of the support polygon.

Admittedly, since the ZRAM-based controller is based on the ZRAM criterion, there are limitations on the level of the external disturbances that it can handle. However, there are two additional methods that the ZRAM-based controller uses to attenuate the additional spin angular momentum in the biped without directly controlling the spin angular momentum:

1. During the first 3000 ms of SS, the reference COG is set to the reference CMP directly instead of being generated from the reference CMP using the rotational dynamic equilibrium equation. This implies that the spin angular momentum is

not precisely regulated at the start of SS. As noted in [3], there are certain types of human movement tasks such as swing foot lift off and hip swirling that might require non-zero spin angular momentum. In fact, the *Resolved Momentum Control* [40] omits the control of the spin angular momentum about the pitch and roll axis in SS in order to facilitate the stepping motion. In the ZRAM-based controller, the ZRAM condition might not be satisfied at the start of SS and the *Stance Leg Chain* is assigned a temporary objective of keeping the COG within the support polygon. In this span of time, the controller forces the perturbed COG (caused by the additional spin angular momentum that is induced by external disturbances) back to its reference position within the support polygon. Since the ZRAM condition is relaxed, a non-zero centroidal moment might be generated as the COG moves in the desired direction. In addition, the controller also reestablishes the pelvis to an upright position during SS. Thus, the COG and pelvis motion in SS can counteract the effects of the additional spin angular momentum in the biped.

2. The second method is actually observed from the simulation results in the planar model test. Under external disturbance, the reference CMP, actual CMP and the COP position will deviate from each other. By moving the COP to the reference CMP position through COG manipulation, the ZRAM-based controller negates the effects of the additional angular momentum in the biped. A detail explanation of this observation will be given in Section 5.2.

In summary, the ZRAM-based controller is a simple method that simultaneously 1) plans high level motions, 2) enforces the dynamic stability constraint, and 3) tracks the reference high level trajectories by using the WBC method (Section 3.5.1) and feedback control (Section 3.5.3).

5.2 The Control Algorithm

The section begins with a brief overview of the control architecture of the ZRAM-based controller. Although the common components of the ZMP-based controller and the ZRAM-based controller is summarized in Section 3.7 and Figure 3.68; for clarification purposes, a more detailed block diagram that only involves the state and output variables of the ZRAM-based controller is presented in Figure 5.1. The focus of this section will be on the reference COG trajectory generation that is based on the ZRAM criterion. In addition, we will also discuss how our method might be able to indirectly minimize the spin angular momentum of the biped.

5.2.1 The Control Architecture of the ZRAM-based Controller

The block diagram of the ZRAM-based controller is presented in Figure 5.1. Although the ZRAM-based controller has multiple control objectives such as the reference swing foot pose and pelvis orientation; for the sake of clarity, only the reference COG is shown in the block diagram. Like the ZMP-based controller, the overall controller of the ZRAM-based controller can be subdivided into four main portions. The *Stable Trajectory Generation* involves planning the high level motion of the biped (represented by the reference CMP) based on user-defined walking parameters and the cubic spline interpolation method. The walking parameters selection is mostly determined by the intermediate objectives and workspace constraints of the biped that were introduced in Section 3.2.2 and the limiting factors that were discussed in Section 4.3. Next, the pre-planned reference CMP is used to generate a dynamically stable reference COG trajectory based on the ZRAM criterion in the “Dynamic Stable Gait Generation” block (described later in this section). The *Closed-looped Kinematic Controller* is used to track the reference trajectories and decompose the actuating

task space velocities from the PD controller into joint velocities. Note that control architecture of the kinematic controller is given in Figure 3.45. Finally, the input torques of the motors are generated by the *Local PID Controller* and the output variables of the controller are approximated by the *Forward Kinematics* block.

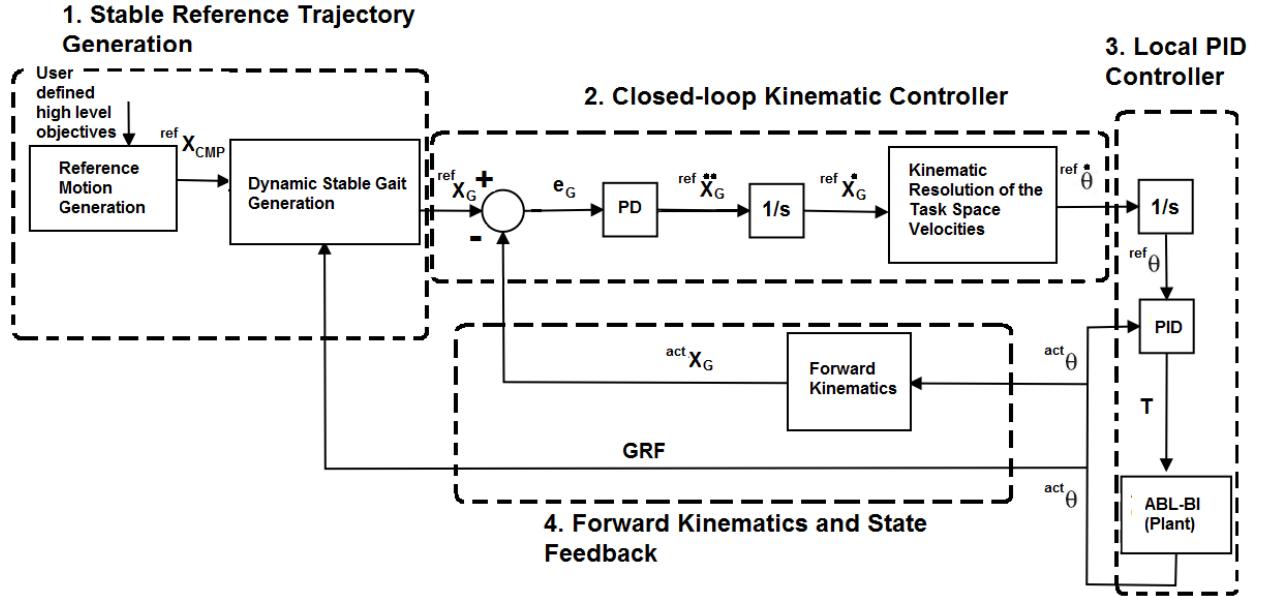


Figure 5.1: A schematic diagram of the control architecture of the ZRAM-based controller.

5.2.2 Dynamically Stable COG Generation based on ZRAM

Let's look more closely at the *Dynamic Stable Gait Generation* block. As shown in Figure 5.2, the inputs of the block are the reference CMP $\text{ref } \vec{X}_{\text{CMP}}$ (preplanned by the user) and the GRF (measured from the biped) and the output is the reference COG $\text{ref } \vec{X}_G$ that is generated based on the ZRAM criterion.

Recall that in Section 3.6.4, we derived the nonlinear relationship between the CMP, the COG, and the GRF based on the rotational dynamic equilibrium of the biped about the COG (Equation 3.65).

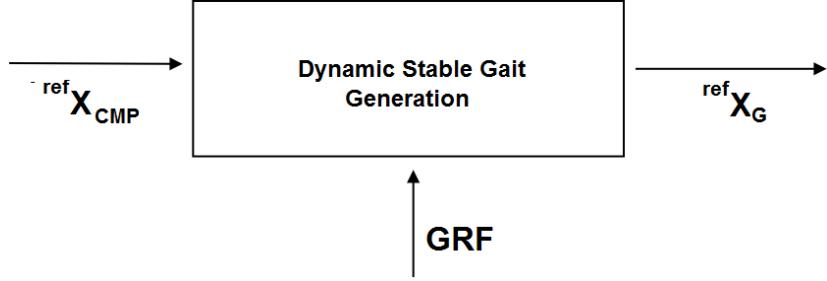


Figure 5.2: The Dynamic Stable Gait block in the ZRAM-based controller.

Rearranging Equation 3.65a and 3.65b, we get:

$${}^{ref}X_G = {}^{ref}X_{CMP} + \frac{F_x}{F_y}y_G \quad (5.1a)$$

$${}^{ref}Z_G = {}^{ref}Z_{CMP} + \frac{F_z}{F_y}y_G \quad (5.1b)$$

It is clear that from Equation 5.1, we can calculate the reference COG trajectory based on the ZRAM condition. If the actual COG of the biped closely tracks the reference COG, then the zero moment condition in Equation 3.65 is satisfied and the controller indirectly manipulates the COP to the reference CMP position. In this case, the COP (the *actual* point of action of the GRF), the actual CMP (the point at which the GRF must act in order to ensure zero moment about the COG, calculated by substituting the current COG position and GRF into Equation 3.65b) and the reference CMP will coincide and the moment about the COG is minimized (left figure of Figure 5.3).

However, in the presence of a disturbance, there will be a separation between 1) the COP and the reference CMP, 2) the actual CMP and the reference CMP, and 3) the COP and the actual CMP (right figure of Figure 5.3). From Equation 3.63a, a separation between the COP and the the actual CMP implies that there

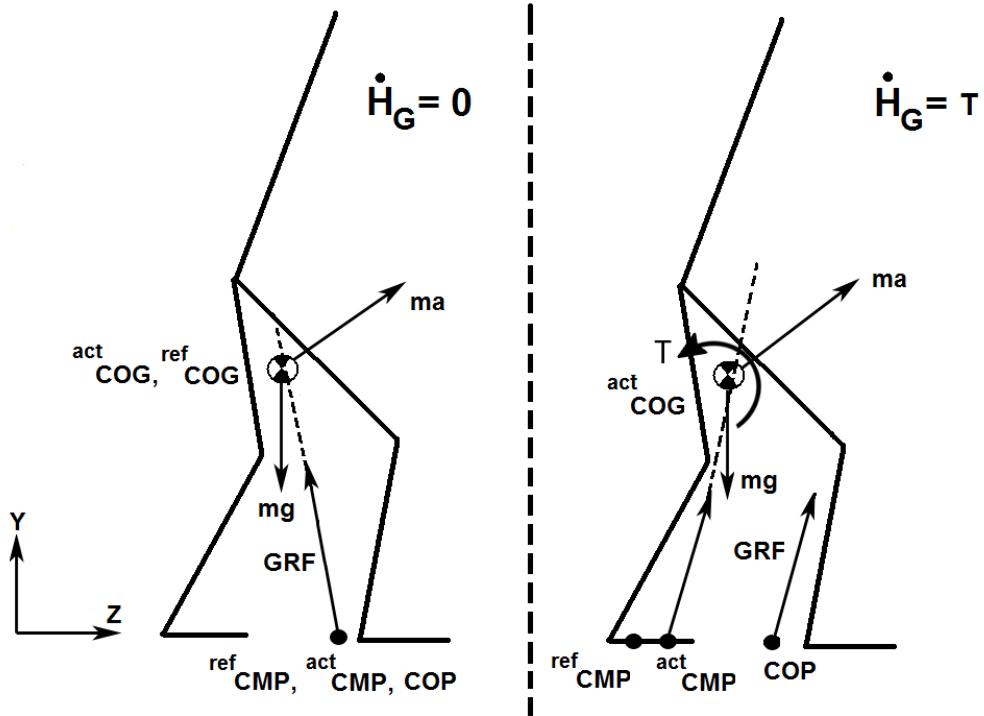


Figure 5.3: The relationship between the reference CMP, the actual CMP and the COP in the zero moment case (left) and the non-zero moment case (right).

is a uncompensated moment acting about the COG of the biped. This moment in turn generates additional spin angular momentum that can eventually tip the robot over. In this situation, the controller will attempt to recover from the disturbance by moving the actual COG to the reference COG that is generated from the reference CMP. Note that this will indirectly manipulate that COP to the reference CMP instead of the actual CMP. Whereas moving the COP to the actual CMP will ensure zero rate of change of angular momentum (Equation 3.63b), moving the COP to the reference CMP (which differs from the actual CMP) can induce a rate of change of angular momentum in the robot. Supported by the results of the planar simulation in Section 5.3, this rate of change of angular momentum is able to attenuate the additional angular momentum generated by the disturbance. Eventually, the COP and the actual CMP will converge to the reference CMP. Therefore, the rate of change of angular momentum will once again be regulated after the biped returns to a steady

walk.

5.3 Simulation Tests in the 2-D Simulation

The ZRAM-based controller is first implemented in the 2-D simulation as an initial proof of concept. Although the planar model is a simplification of the full kinematics and dynamics of ABL-BI, it allowed us to quickly implement, debug and modify a *simplified planar version* of the ZRAM-based controller. The results from the 2-D simulation also provide general insights on the effectiveness of the controller.

The main objectives of the 2-D simulation are:

- Verify that the ZRAM-based controller can track the reference high-level trajectories and enable the biped to walk in a stable manner.
- Observe how the ZRAM-based controller respond to external disturbances and verify that the biped can maintain dynamic balance by tracking a reference COG trajectory that is generated from the ZRAM model.

5.3.1 Planar Model

The 2-D simulator (based on Matlab and Simulink) and the planar biped model was developed by Olli Haavisto at the Helsinki University of Technology. The interested reader can refer to [73] for a detailed description of the 2-D simulation environment, the ground model, and the planar model. The following section will only provide a brief description on the control architecture of the *planar version* of the ZRAM-based controller and the major modifications that we made to the planar model.

The planar model consists of five links (two thighs, two shanks and one torso) and four joints (two hip pitch joints and two knee pitch joints). The links are assumed to be rigid bodies with concentrated masses that are located at their respective local COGs (note that the local COG, in turn, are assumed to be at the midpoint of the link). In order for the COP to move on the ground surface (and not constrained to a single point), the model is slightly modified to include two massless feet and ankle joints (Figure 5.4). Using the kinematics (link lengths and local COG positions) and the mass properties of ABL-BI, the *exact* dynamics of the planar model (a set of nine differential equations) is derived from the Lagrangian equation (please refer to [73] for the full derivation).

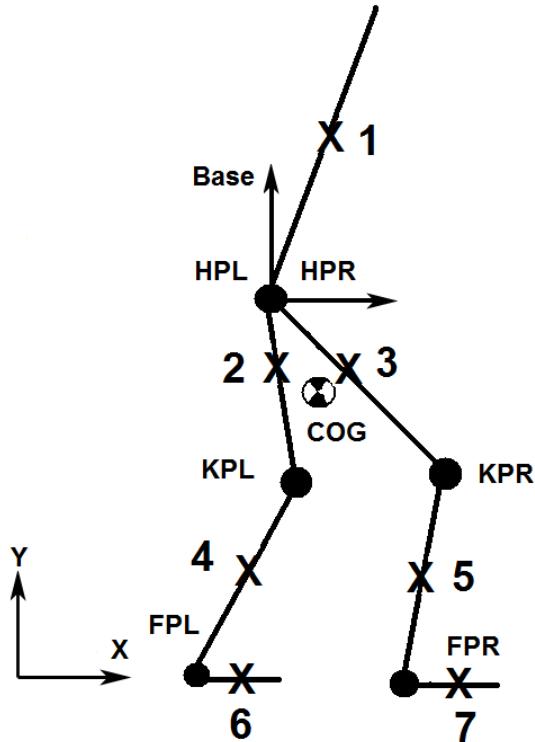


Figure 5.4: The Planar Model of ABL-BI. The crosses represent the link COG positions of the seven links and the dots represent the actuated joints. Note that in the 2-D simulation, the sagittal direction is represented by the X axis and a *base frame* is attached to the Hip Pitch joint of the biped.

The ground is modeled as a spring and damper system. The normal component of the GRF is proportional to distance and velocity in which the foot “penetrates” into the ground in the normal direction. Likewise, the tangential static friction force is proportional to foot penetration in the tangential direction. Once the static friction force exceeds the maximum threshold value, the kinetic friction force is calculated from the kinetic friction coefficient and the normal component of the GRF.

The planar implementation of the ZRAM-based controller in Simulink is summarized by the block diagram in Figure 5.5. Essentially, we “stripped off” the motion generation and stabilizing controller of the original control algorithm that is implemented in [73] and replaced it with the components of the ZRAM-based controller (Figure 5.1). These components include: 1) Stable Reference Trajectory Generation, 2) the Closed loop Kinematic Controller, and 3) Forward Kinematics and State Feedback. Thus, the modified planar controller in Simulink has the same control architecture as the ZRAM-based controller. The only elements that are retained from the original control algorithm are:

- The dynamic model of the biped that calculates the velocity and position of the *Torso* COG and the relative angular position between the links (the generalized coordinates of the dynamic model) from the input joint torques and the GRF. Note that the simulation uses a variable step *ode23tb* integrator that focuses on speed and computational efficiency over accuracy. It is also important to note that the input and output signals of the biped are discretized using the zero-order hold and the simulation has a sampling period of 5 *ms*.
- The ground contact model that calculates the GRF of the biped given the ankle position (the “toes” of the original planar model) with respect to the ground.
- The locomotion state machine that determines the state of the biped based on

the ankle position with respect to the ground.

- The local PID controller that generates input joint torques from the reference and actual joint positions.

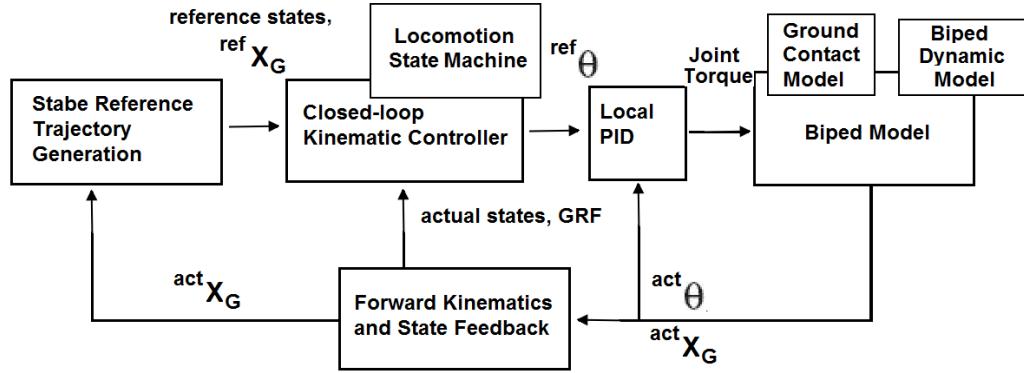


Figure 5.5: The control architecture of the 2-D simulation. The high level motion generator and the stabilizing controller is adopted from the ZRAM-based controller. The main components that are retained from the original controller are the ground model and the biped dynamics.

As indicated in Figure 5.5, the dynamic model of the biped and the ground contact model are included in the *Biped Model* block and the state machine is included in the *Closed-loop Kinematic Controller* block.

Finally, the mass and kinematic properties of the biped is presented in Table 5.1.

There is about a 30% difference between the total mass of the planar model and the 3-D model in *Webots*. The difference is due to fact that we omitted the mass of the feet, the pelvis link, and the *TorsoRoll* motor in the planar model. However, since the mass difference will only mostly affect the control gains of the *Closed-loop Kinematic Controller*, the planar model is still useful in validating the the walking gaits that are generated from the ZRAM-based controller.

Table 5.1: The kinematic and mass properties of the planar biped.

Link	Mass (kg)	Link Length (m)
1. Torso	5.0	0.4
2. Right Thigh	3.0	0.3
3. Left Thigh	3.0	0.3
4. Right Shank	3.0	0.3
5. Left Shank	3.0	0.3
6. Right Foot	0.0	0.15
7. Left Foot	0.0	0.15

5.3.2 Simulation Results and Discussions

Using the ZRAM-based controller on the planar model, the biped was able to walk continuously in a nine step cycle, 50 s (50000 ms) test. The step cycle period was set to 5.5 s (5500 ms), and the step length was 4 cm . It is observed that most existing bipeds and humans can achieve a faster walking speed. Please refer back to Chapter 4 for a full explanation on why it is difficult to achieve a faster performance on ABL-BI. Possible improvements will also be suggested in Chapter 6. For now, let's focus on the reference trajectory tracking of the ZRAM-based controller in the presence of external disturbance. The biped was given a 25 N push for 2 s in the middle of SS between 31 s and 33 s at a distance of 0.1 m above the *base frame* (see Figure 5.4). Since the vertical COG position is close to the *base frame*, the push would result in a 2.5 Nm centroidal moment disturbance, which is similar to the level of disturbance in [74]. The walking gait of the biped is shown in a series of freeze frame figures in Figure 5.6. Although the disturbance causes the biped to slightly deviate from its nominal walking gait, the biped gradually recovers to its initial gait within two step cycles.

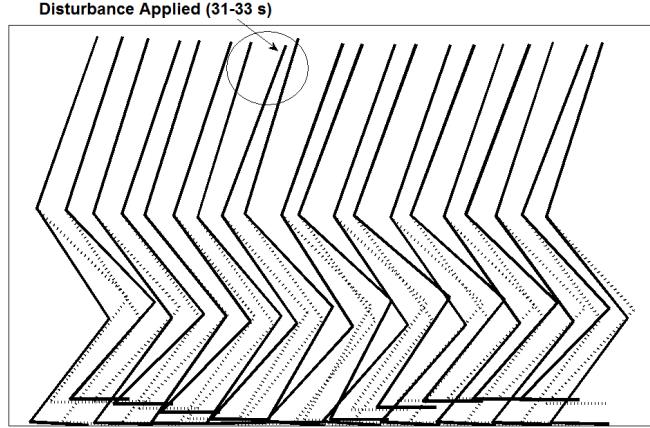


Figure 5.6: Elapsed time snapshot of the planar biped walking test. An external centroidal moment of 2.5 Nm is applied to the biped for two seconds during SS. The ZRAM-based controller was able to attenuate the disturbance and the biped returns to its original gait after two step cycles.

The response of the ZRAM-based controller to an external disturbance is best illustrated in Figure 5.7. Notice that 1) the actual COG tracks the reference COG accurately throughout the walking test (with or without external disturbance), and 2) the reference COG tracks the reference CMP accurately except during the perturbation at 31 s to 33 s . Also, the horizontal and vertical component of the GRF are presented in Figure 5.8. The magnitude of the horizontal GRF remain very small ($< 3 \text{ N}$) throughout the walking cycle but increase significantly to about 20 N when the external disturbance occurs. For the vertical GRF, the magnitude remain small throughout the walking cycle with sharp increases during lift-off and touch-down.

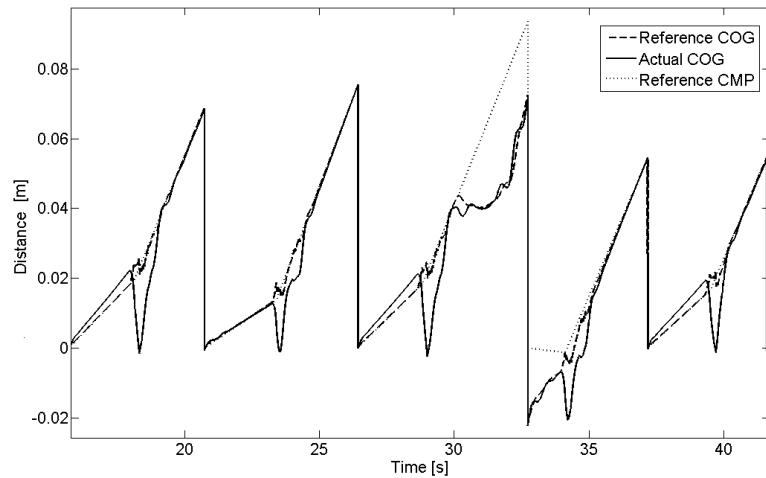


Figure 5.7: Planar test: The reference CMP, the reference COG, and the actual COG position (in the horizontal axis) with respect to the base frame.

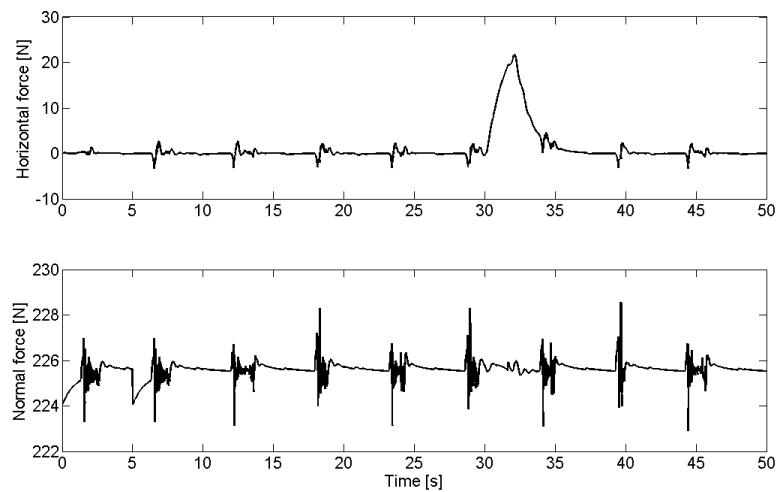


Figure 5.8: Planar test: The horizontal and vertical component of the GRF.

Several important points to note from the above results:

1. The biped can accurately track the COG while taking a step. This indicates that the high level objectives in the *Swing Leg Chain* in SS does not come in conflict with the objectives in the *Stance Leg Chain*. In addition, the results show that although the biped is treated as two kinematic chains, the *Stance Leg Chain* is able to track the COG and account for the effects of the swing leg inertia on the COG position.
2. The reference COG tracks the reference CMP closely when there are *no* external disturbances and will diverge from the reference CMP in the presence of external disturbances. As shown in Figure 5.8, the magnitude of the horizontal GRF is very close to zero in the stable phases of the walking test. In this case, the line of action of the GRF vector is approximately normal to the ground surface and the ZRAM condition is achieved if the horizontal components of the COG tracks the reference CMP (Section 1.2). When the biped is perturbed between 31 s and 33 s, there is a large increase in the horizontal GRF (Figure 5.8). Thus, the effect of the disturbance is reflected in the horizontal GRF and the reference COG must deviate from the reference CMP (by up to 4 cm in Figure 5.7) in order to satisfy the ZRAM condition (Figure 5.7). The effects of the horizontal GRF can also be seen from the second term of Equation 5.1. The reference COG will significantly diverge from the reference CMP to satisfy the ZRAM condition *only if* there is a large non-zero horizontal GRF. This is an advantageous characteristic of the ZRAM-based controller for the following reasons:
 - When there are no disturbances, the actual COG will essentially track a preplanned high level motion that is specified by the user (the reference CMP).

- During disturbance, the controller will automatically modify the reference trajectory in realtime to maintain dynamic balance. Therefore, instead of solely relying on feedback control to track a fixed reference trajectory (like the ZMP-based controller), the ZRAM-based controller allows the biped to track a modified reference COG that corresponds to the compensatory motion for external disturbances. After the disturbance is attenuated, the reference COG trajectory converges to the reference CMP trajectory and the biped returns to its original gait.

The difference between the reference CMP and the COP position ($^{ref}\delta r$) and between the actual CMP and the COP position ($^{act}\delta r$) are plotted in Figure 5.9. During the stable phases of the walking test, $^{ref}\delta r$ and $^{act}\delta r$ are very close to each other, this indicates that 1) the ZRAM-based controller is able to regulate the centroidal moment of the biped by minimizing the distance between the COP and the reference CMP position, and 2) The reference CMP, the actual CMP, and the COP closely tracks one another.

After the biped is perturbed, the reference CMP deviates from the actual CMP by about 6 cm (Figure 5.9). Since the reference COG is generated from the reference CMP, the ZRAM-based controller will indirectly manipulate the COP to the reference CMP (instead of to the actual CMP position) by tracking the reference COG. In the process of doing so, the motion of the biped will generate a restorative centroidal moment (since there is a separation between the reference and actual CMP) that attenuates the additional spin angular momentum. The effects of this can be seen in Figure 5.10. At perturbation (about 31 – 32.5 s) there is an increase in the spin angular momentum; however this is followed by a sharp decrease at about 32.5 – 35 s. This indicates that the ZRAM-based controller is generating a compensatory moment

to counteract the external disturbance. Finally, although the centroidal moment and the spin angular momentum are not perfectly regulated, we can see that throughout the walking cycle, the spin angular momentum is bounded to within very small values.

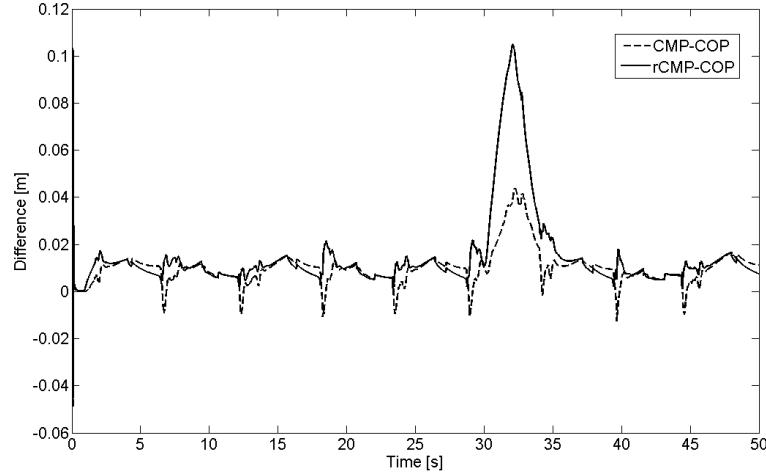


Figure 5.9: Planar test: The error between the reference CMP and the COP and the actual CMP and the COP.

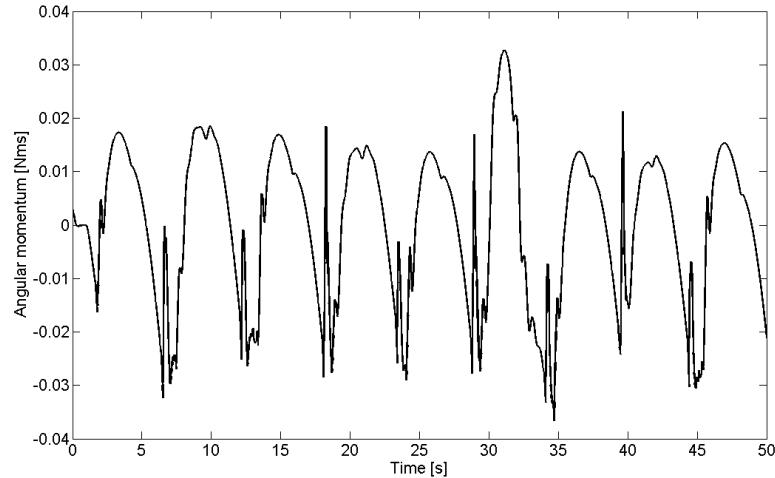


Figure 5.10: Planar test: The spin angular momentum of the planar biped.

In summary, the planar test results verified that 1) the ZRAM-based controller can track high-level reference trajectories and generate stable walking gaits, and 2) the

ZRAM-based controller can attenuate external disturbances by tracking a modified reference COG trajectory that is based on the ZRAM criterion.

5.4 Implementation on the 3-D Model

In this section, we perform walking tests on a 13-DOF, 7 link 3-D simulated biped in *Webots* that is described in Section 2.2. The control algorithm of the ZRAM-based controller that is used in the simulated biped is summarized in Section 5.2 and Figure 5.1. Extending the implementation of the ZRAM-based controller from the planar model to the 3-D model is not a trivial task. The following are some of additional challenges from the 3-D implementation:

1. The 3-D simulated biped is modelled as a system of rigid bodies with distributed mass. The mass and moment of inertia of each link is set according to the values of *the actual ABL-BI platform*. Unlike humans and some existing bipeds, the swing leg accounts for a large portion of the total inertia of ABL-BI. For this reason, the angular velocity of the swing leg contributes significantly to the centroidal moment of the biped. If the swing leg has a large angular velocity, the biped has to pitch the stance leg and pelvis forward to maintain ZRAM. Unfortunately, this compensatory pitching motion will result in a non-anthropomorphic gait in SS. In some cases, the biped cannot even perform these motions due to joint range limitations. Two methods were used to avoid this motion: 1) Relax the ZRAM condition in the first half of SS (Note: there are also other motivations for doing this, please refer to Section 5.1.3), and 2) reduce the angular velocity of the swing leg by planning a reference trajectory with a longer swing period.
2. The ZRAM-based controller has to deal with the full kinematics of the biped (7

links and 13 joints) and additional task space objectives that are associated with the lateral (X-Y) and frontal (X-Z) planes. For this reason, the forward and inverse kinematics of the 3-D biped is a lot more complex and computationally expensive than the planar case. To facilitate postural control of the biped, we use the kinematic chains and the kinematic redundancy resolution methods that were developed in Chapter 3.

3. In the 3-D biped, we have to additionally control the lateral stability. In DB, this is done by shifting the COG of the biped to the forward stance foot. However, this has proven to be problematic in Chapter 4 due to the weight distribution of the biped. The lateral stability of the biped must also be maintained in SS. Since the width of the foot (10 cm) is smaller than its length (15 cm), the biped has the smallest stability margin in the lateral direction. Therefore, the reference CMP trajectory must be carefully planned so that the biped can maintain dynamic stability in SS.

As noted in Section 3.6, the kinematics of the planar model is overly simplified and the actual dynamic behaviour of the biped cannot be reflected in the model. Therefore, despite the additional difficulties, we must validate the ZRAM-based controller in a 3-D simulation before it can be implemented on the actual platform. Two sets of simulation tests were performed in *Webots* to validate the ZRAM-based controller: 1) walking tests, and 2) perturbation tests.

5.4.1 Walking Tests

One of the main research objective of this project is to design and validate a ZRAM-based walking controller that would enable the biped to walk in a stable manner and attenuate external disturbances. Throughout this thesis, we have been developing different “tools” that are essential to a bipedal walking controller. In the previous

Chapters, we built the simulated biped in *Webots*, defined the locomotion states and the kinematic chains, derived the forward kinematics and developed the Closed-looped Kinematic Controller to control the posture of the biped. Also, in this Chapter, we applied the nonlinear coupling between the ground reference points and the GRF to generate dynamically stable COG trajectories based on the ZRAM criterion. Having integrated most of the essential components of bipedal walking in a unified control scheme, the final step is to evaluate the performance of the controller in walking tests.

The main objectives of the walking tests are the following:

1. Verify that the preplanned *intermediate objectives* (Section 3.2.2) would result in an overall stable walking gait. Also verify that the reference trajectories for the CMP and the swing foot position are feasible objectives that can be executed by the biped (ie: satisfies the internal constraints that are defined in Section 3.2.2).
2. Verify that the *Closed-looped Kinematic Controller* can achieve the multiple high level objectives that define the overall motion of the biped (ex: COG and swing foot position tracking). Also, observe whether or not the high level objectives conform to their respective priority levels that were assigned by the *task prioritization method* (Section 3.5.3).

A Summary of the User-specified Parameters for the ZRAM-based Controller

The control gains for each high level objective in the *Closed-looped Kinematic Controller* are summarized in Table 5.2. Like the ZMP-based controller, the gains are selected based on their effects on the tracking performance of the controller (overshoot, settling time etc.). It was determined from the simulation tests that the biped

can still perform stable walking if all the gains were varied by about 5 %.

Table 5.2: The control gains for the ZRAM-based controller in the walking test.

High level Objective	Control Gains [X,Y,Z]
COG Position P-Gain (X)	$k_p = 0.4$ for DB and $k_p = 1.0$ for SS
COG Position D-Gain (X)	$k_d = 0.4$ for DB and $k_d = 0.5$ for SS
COG Position P-Gain (Z)	$k_p = 0.4$ for DB and $k_p = 1.0$ for SS
COG Position D-Gain (Z)	$k_d = 0.4$ for DB and $k_d = 0.7$ for SS
COG Position P-Gain (Y)	$k_p = 0.3$
Pelvis Orientation P-Gain	[1.25, 1.00, 1.25]
Swing Foot Position P-Gain	[2.50, 7.00, 5.00]
Swing Foot Orientation P-Gain	[5.00, 2.00, 2.00]

The pseudoinverse damping factors λ were also manually tuned (summarized in Table 3.4). They were set to low values so that the reference task-space velocities can be accurately tracked.

The reference CMP represents the overall reference motion of the biped and are planned based on the *intermediate objectives* of each locomotion state. Some of the qualitative aspects of reference trajectories planning has already been discussed in Section 3.2.2. For example, the CMP must be planned within the support polygon and the target time period of the trajectory is limited by the internal constraints of the biped. The parameters for the reference trajectory planning were selected mainly based on the workspace constraints and the observed behaviour of the biped in simulation tests. Some of these parameters (in particular, the time period for the CMP in DB and SS) are limited for reasons that are discussed in 4.3. The key walking parameters are summarized in Table 5.3.

Table 5.3: The key reference trajectory planning parameters for the ZRAM-based controller.

Walking Parameters	Values
DB Period	5000 – 5800 ms
Step Period	4800 – 5500 ms
Step Length	15 cm (equals to the length of the foot)
Maximum height of the swing foot	5 cm

Results and Discussions

The biped is commanded to take six steps with walking parameters that are summarized in Table 5.3. The duration of the test is 70 seconds (70000 ms). As shown in the time-elapsed snapshots of the walking test in Figure 5.11, the biped had a natural walking gait. First, let's examine the results for the reference CMP and the reference COG tracking in Figures 5.12, 5.13, and 5.14.

We note several important points from the simulation test results.

1. The reference COG that is generated from the ZRAM model is essentially identical to the reference CMP (Figures 5.12 and 5.14). As demonstrated in the planar walking test, this is due to the very small horizontal GRF in stable walking. Since the swing foot motion induces centroidal moment during lift-off, there are no *a priori* reasons for the reference COG to track the reference CMP in SS. However, as noted earlier in this section, the effects of the centroidal moment are neglected by the controller in the first half of SS; thus, the reference COG tracks the reference CMP throughout the walking cycle. The tracking of the reference COG by the actual COG is not perfect (with a maximum error of 3 cm); particularly during lift-off where the actual COG fluctuates significantly before

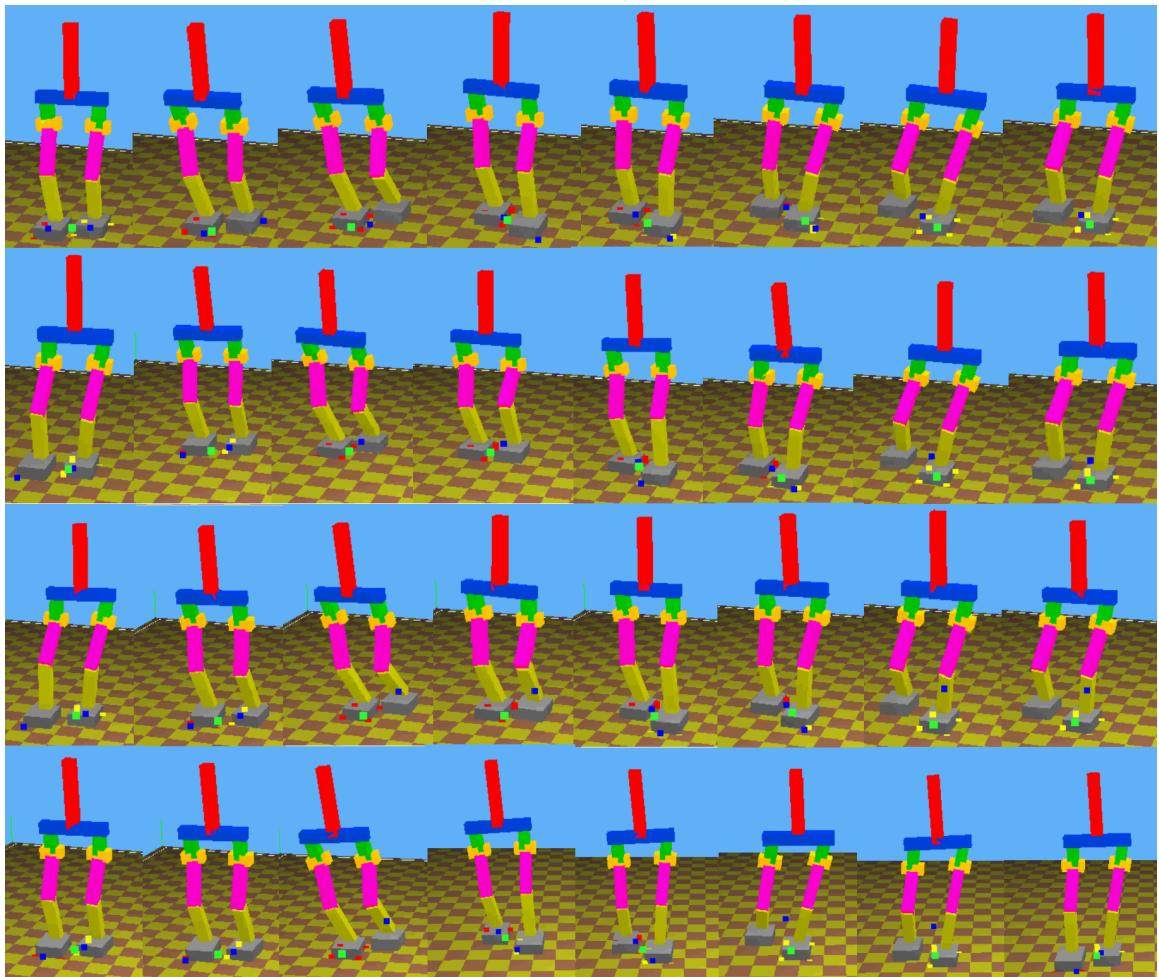


Figure 5.11: The time-elapsed snapshots of the six step walking test in Webots showing the major locomotion phases in the walking cycle.

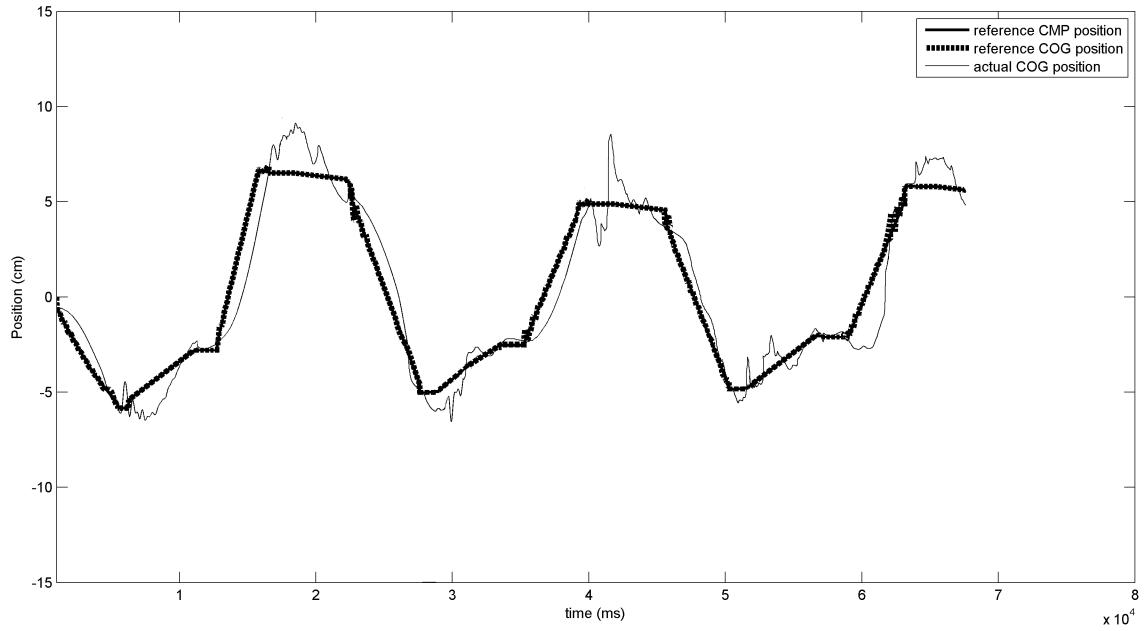


Figure 5.12: Walking test of the ZRAM-based controller: the reference CMP, reference COG and actual COG position with respect to the world frame (X).

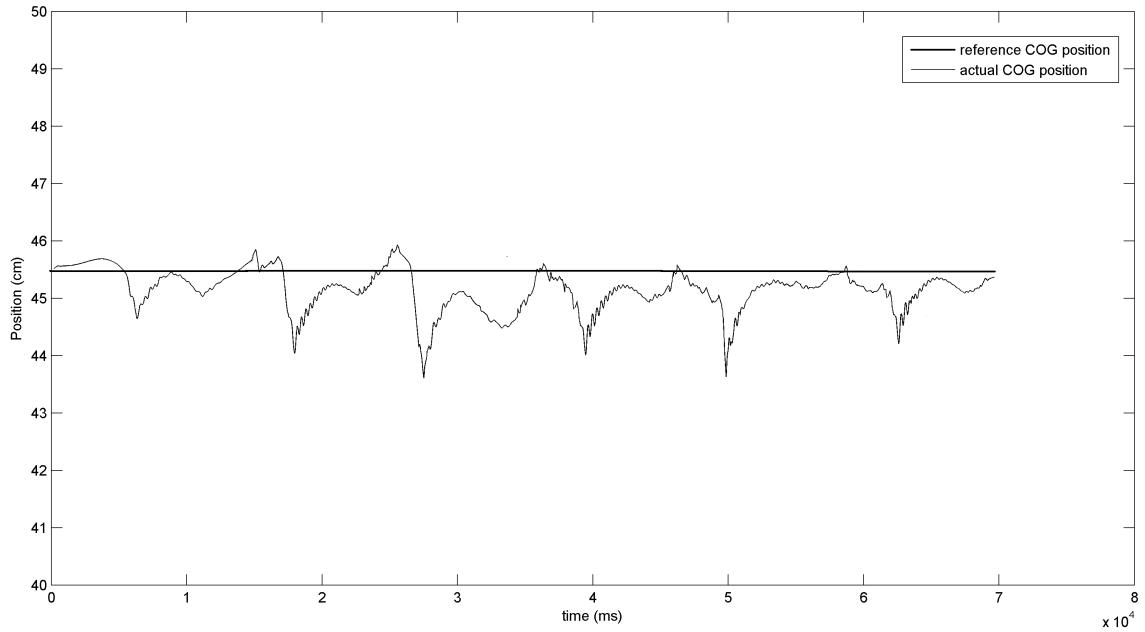


Figure 5.13: Walking test of the ZRAM-based controller: the reference COG and actual COG position with respect to the world frame (Y).

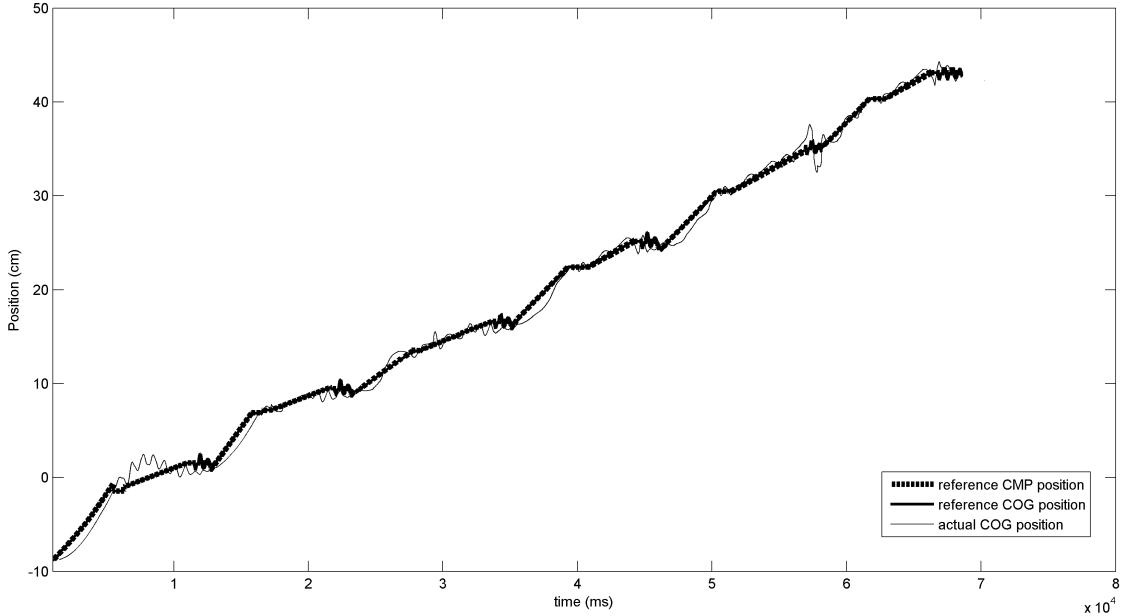


Figure 5.14: Walking test of the ZRAM-based controller: the reference CMP, reference COG and actual COG position with respect to the world frame (Z).

settling down (due to feedback control) towards the end of SS. This tracking error is due to:

- The sharp change in the reference COG trajectory at state transitions.
- The sudden increase in the angular velocity of the swing leg (which accounts for almost one-third of the total mass of the biped) during lift-off.
- The actuating reference COG acceleration from the COG feedback controller has to go through multiple levels of numerical integration and filtering before it reaches the reference joint position (Figure 5.1). Delays and inaccuracies are introduced at each level of integration and filtering.
- The reference COG velocity cannot be perfectly tracked by the joint space solutions from the inverse differential kinematics due to the damping factor in the pseudoinverse of the COG Jacobian.

For these reasons, we cannot expect the reference COG to be tracked exactly

by the controller. From Equation 5.1, we can see that the COG tracking error would result in small non-zero centroidal moment in the biped due to the slight separation between the COP and the reference CMP. However, as shown in the above results, the centoridal moment is not significant enough to disrupt the stable walking gait and can be attenuated by feedback control. In summary: 1) the *Closed-looped Kinematic Controller* is able to achieve its primary high level objective, 2) a small level of reference COG tracking error is acceptable, and 3) the ZRAM-based controller is able to maintain dynamic stability by tracking the reference COG trajectory.

2. Although the vertical (Y) COG position is not controlled in DB, it does not produce any detrimental effects on the dynamic stability and kinematic posture of the biped. As shown in Figure 5.13, the COG decreases slightly in the Y direction (by about 2 cm) due to the pitching motion in weight-shift. However, in the subsequent SS phase, the vertical COG position is controlled by a P-controller and gradually returns to its initial value. Although there is no explicit planning for the vertical COG position, the above control setup results in an arc like cycle that is similar to the vertical COG motion observed in human walking [75].

Next, we examine the results for the pelvis orientation tracking in Figures 5.15, 5.16, and 5.17.

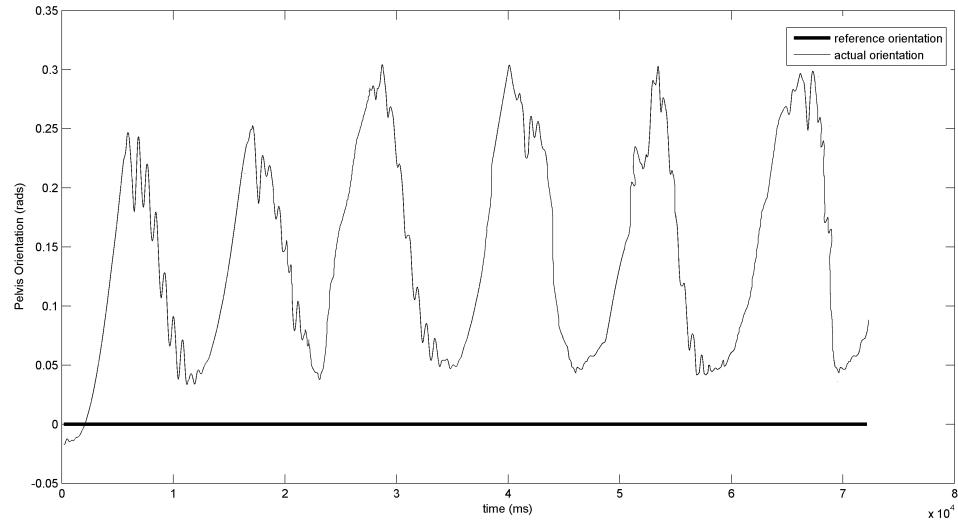


Figure 5.15: The comparison between the reference and actual pelvis orientation about the X axis with respect to the world frame.

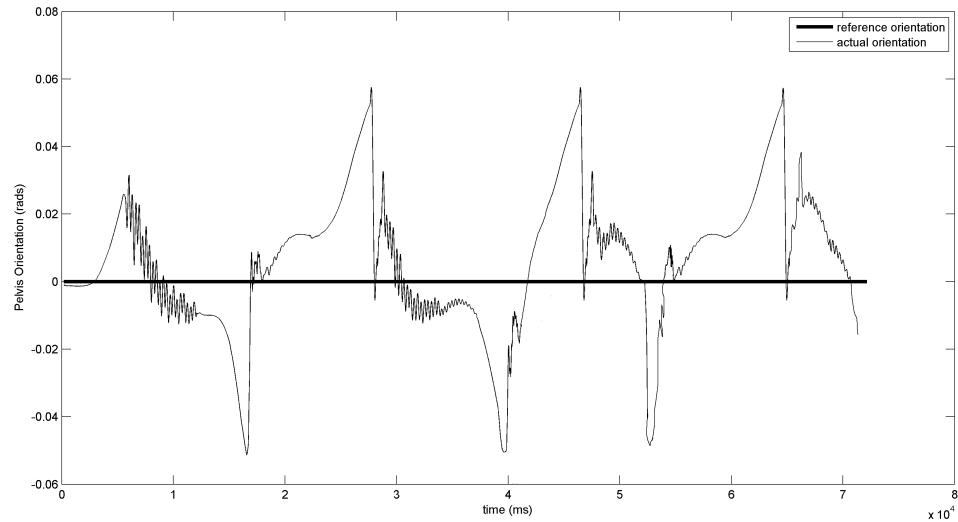


Figure 5.16: The comparison between the reference and actual pelvis orientation about the Y axis with respect to the world frame.

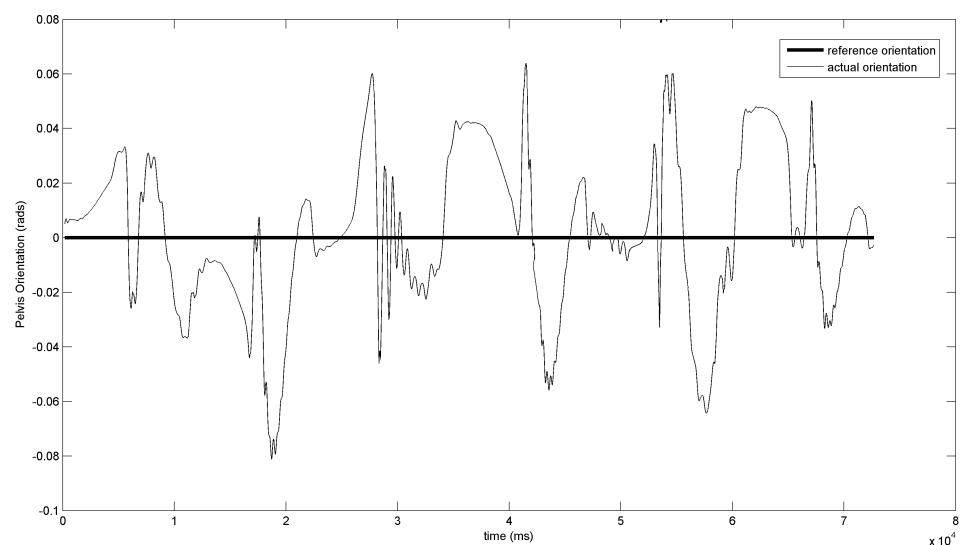


Figure 5.17: The comparison between the reference and actual pelvis orientation about the Z axis with respect to the world frame.

The above results are similar to the simulation tests of the ZMP-based controller in Chapter 4. Several important points for the simulation results of the pelvis orientation tracking test are noted below:

1. The pelvis orientation about the Y (yaw) axis and the Z (roll) axis are bounded to within small values by the ZRAM-based controller (the maximum angular position is about 4.5°).
2. As with the tests in the ZMP-based controller, the pelvis orientation is not tracked by the controller in DB due to its interference with the primary task (reference COG tracking). However, in SS, the pelvis gradually returns to an upright orientation. Again, this result demonstrates how the task prioritization method (Section 3.5.3) gives preference to the primary tasks (which are critical to all other tasks in the biped) over the less stringent secondary tasks. The results also show that the secondary task will eventually be fulfilled once it is not in conflict with the primary task.

We present the simulation results for the swing foot position and orientation tracking in 1) SSL, and 2) SSR.

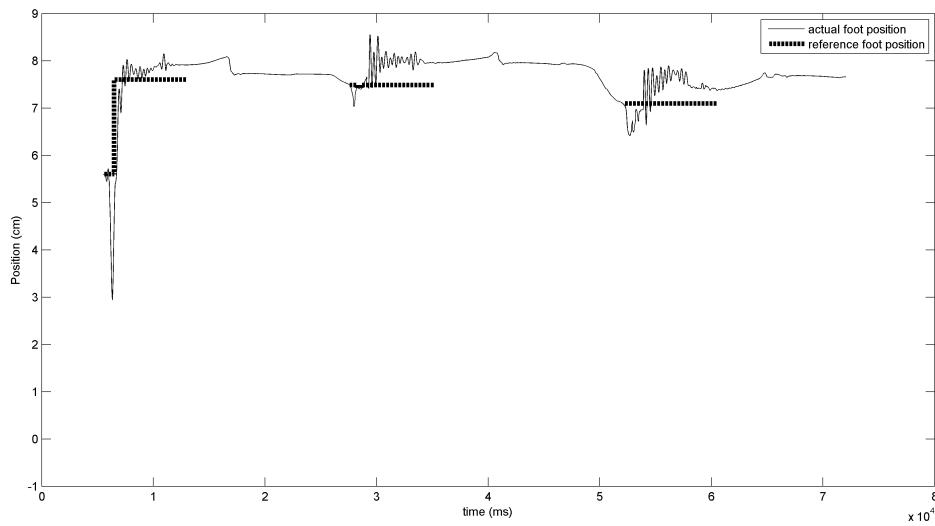


Figure 5.18: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the X direction (SSL).

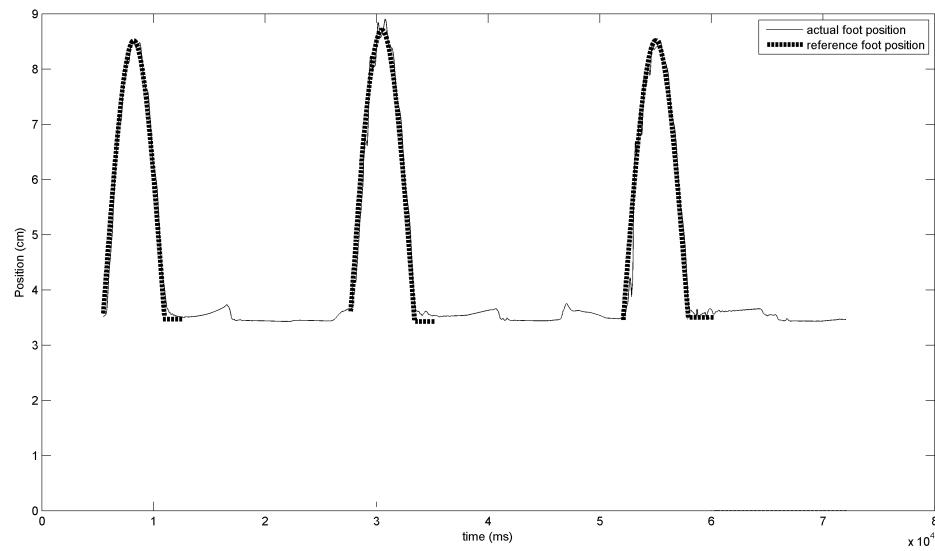


Figure 5.19: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Y direction (SSL).

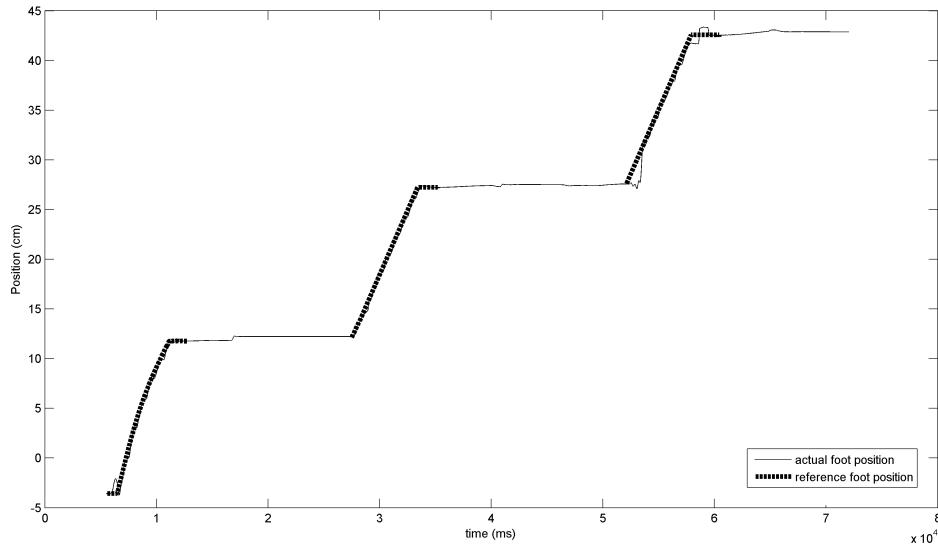


Figure 5.20: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Z direction (SSL).

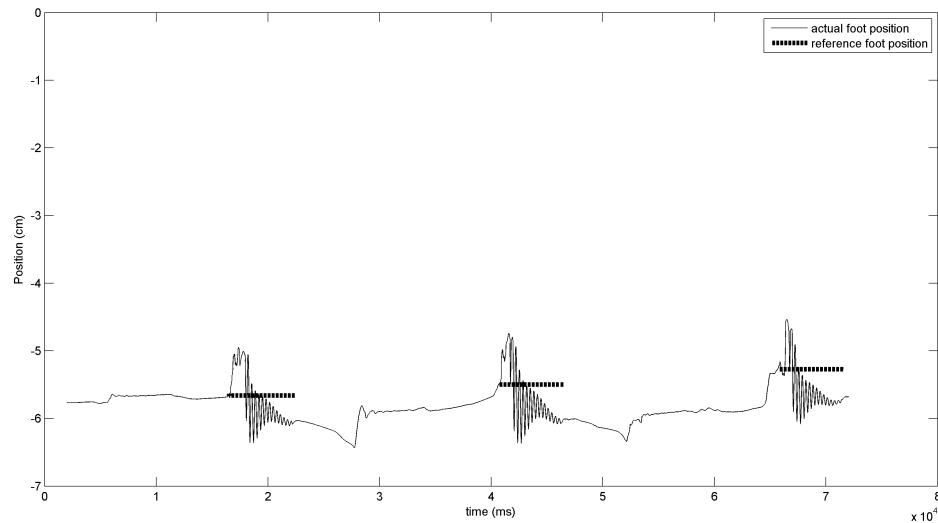


Figure 5.21: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the X direction (SSR).

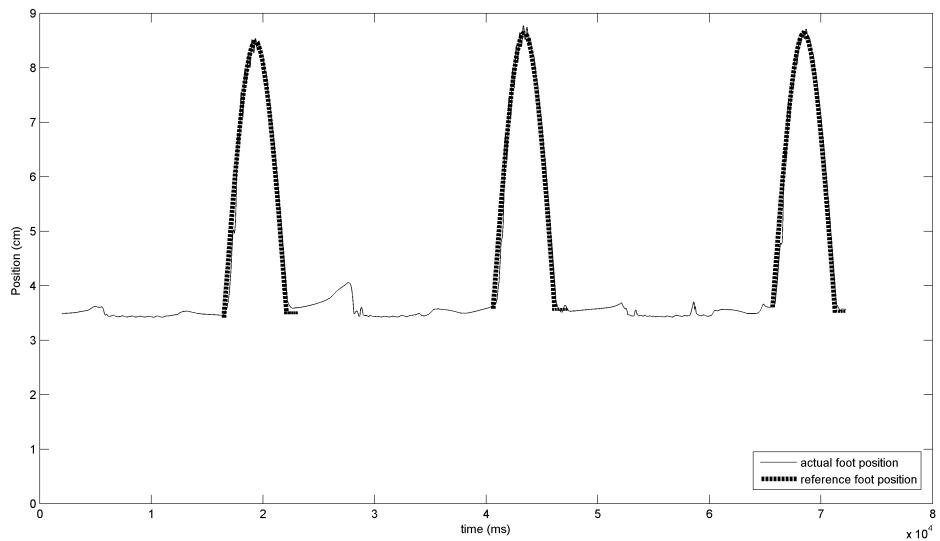


Figure 5.22: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Y direction (SSR).

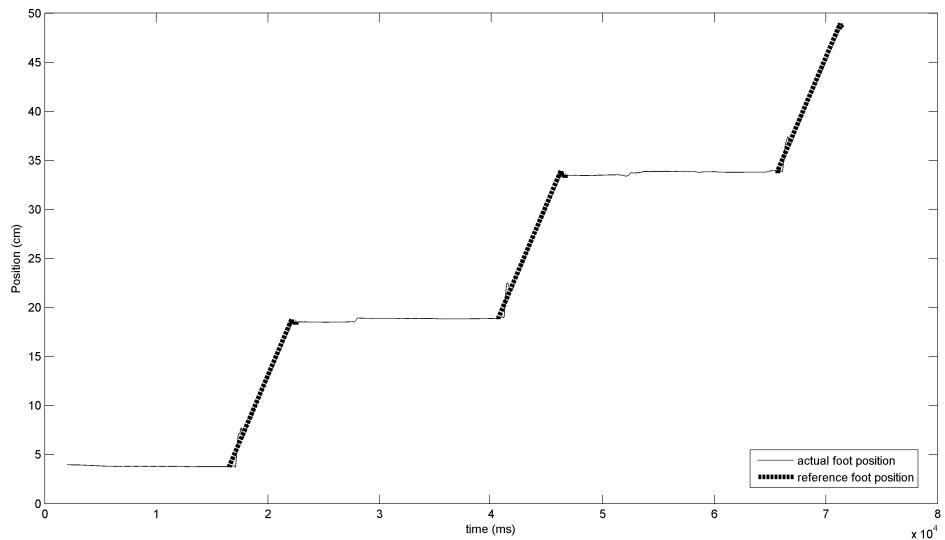


Figure 5.23: The comparison between the reference and actual swing foot position with respect to the stance foot frame in the Z direction (SSR).

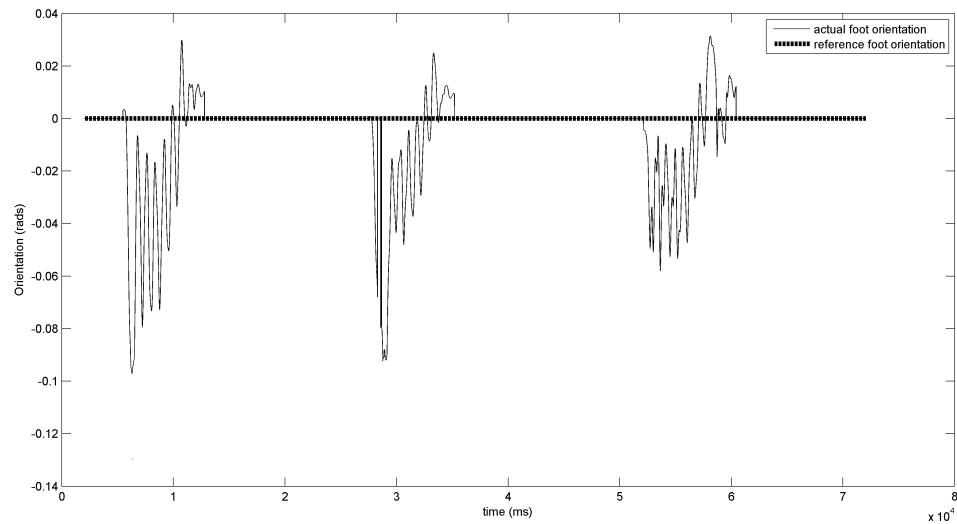


Figure 5.24: The comparison between the reference and actual swing foot orientation with respect to the world frame about the X axis (SSL).

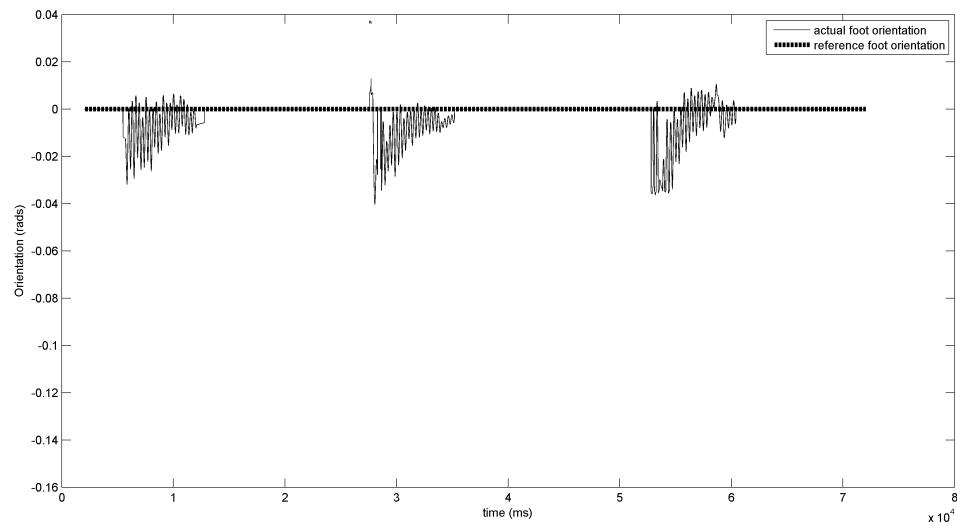


Figure 5.25: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Y axis (SSL).

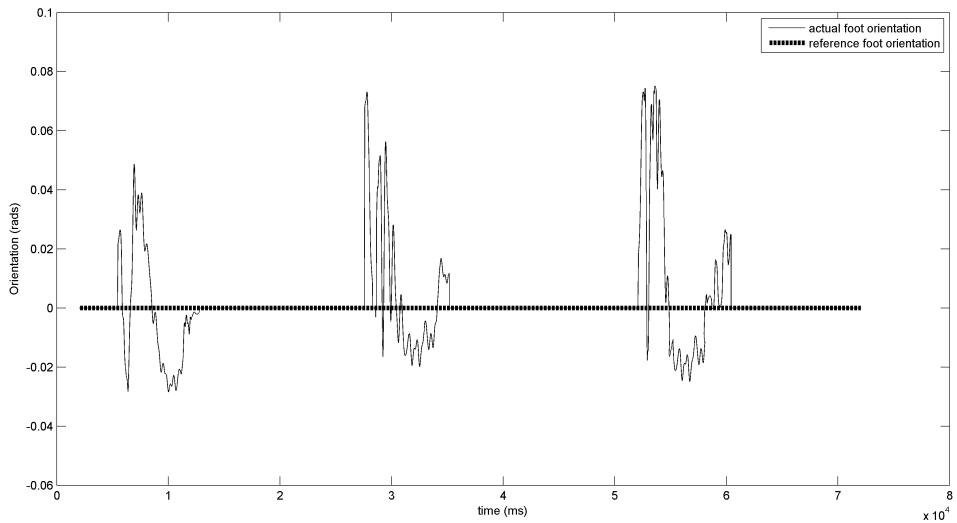


Figure 5.26: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Z axis (SSL).

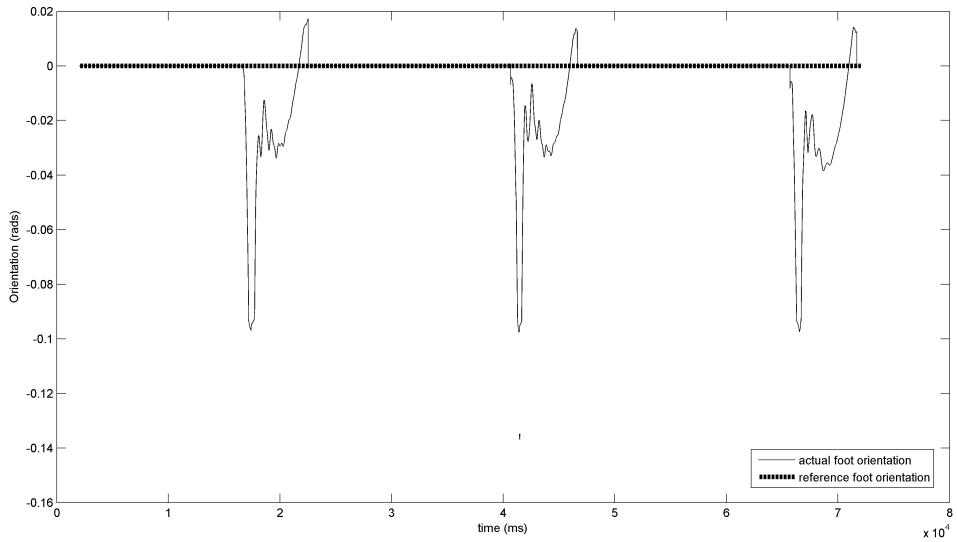


Figure 5.27: The comparison between the reference and actual swing foot orientation with respect to the world frame about the X axis (SSR).

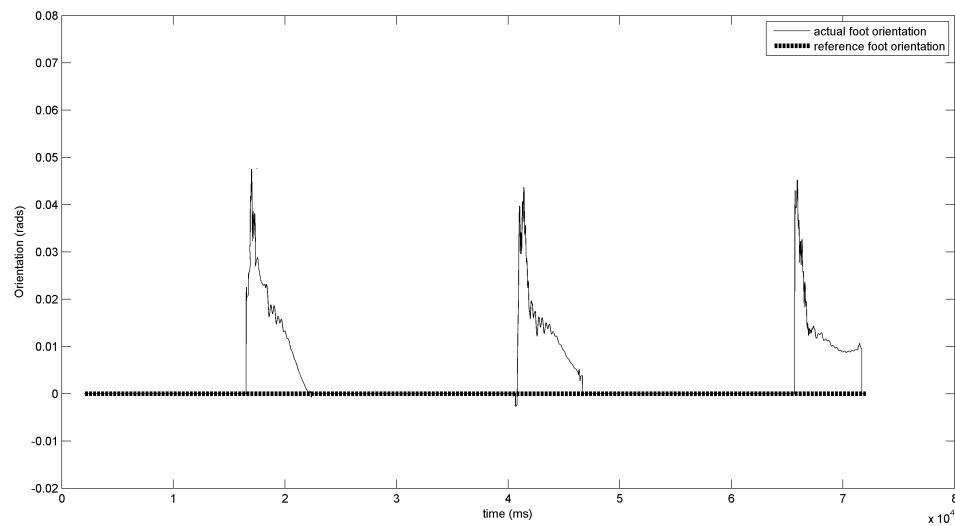


Figure 5.28: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Y axis (SSR).

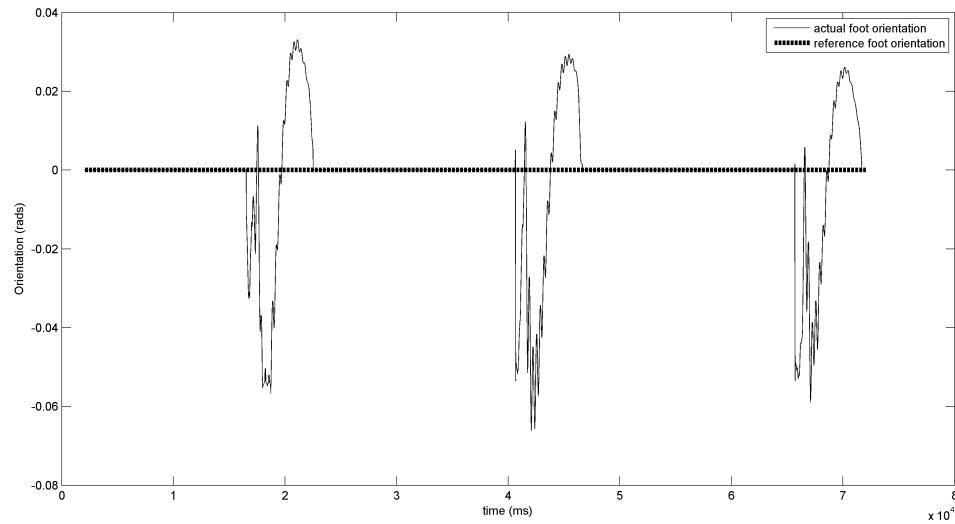


Figure 5.29: The comparison between the reference and actual swing foot orientation with respect to the world frame about the Z axis (SSR).

Like the ZMP-based controller, the swing foot pose with respect to the stance foot closely tracks the reference pose in SS. A few comments for the results in the ZRAM-based controller are noted below:

1. The swing foot position with respect to the stance foot is only explicitly controlled in SS. As noted in Section 3.4, the stance feet are treated as base frames for the kinematic chains in DB and are assumed to be fixed with respect to the ground. Thus, the controller will not produce joint space solutions that cause the stance foot to voluntarily lift off from the ground. The only conditions that would result in relative motion between the stance feet in DB are: 1) insufficient ground friction that causes the foot to slide in the horizontal plane and rotate about the vertical axis, and 2) the COP of *one of the stance foot* (and not the overall COP of the biped in DB) is at the edge of the support surface. This causes one of the stance foot to rotate about the horizontal axis while the other stance foot remains flat on the ground. As shown in the swing foot position plots (Figures 5.18 to 5.23), the relative position between the feet changes slightly in DB. This implies the stance feet does not remain perfectly stationary on the ground. Fortunately, the changes are small and has no significant effect on the dynamic stability of the biped.
2. In SS, the oscillatory motion of the swing foot in the X direction is caused by a brief interference between the foot and the ground right before lift off. However, the oscillatory motion eventually settles down at the end of SS. Although the foot placement of the biped in the X direction differs from the reference position by about 0.4 cm, this slight deviation does not have significant effects on the stability and posture of the biped at touch-down.
3. Similar to the ZMP-based controller, the orientation of the swing foot is non-zero at the start of SS but eventually reduces to close to zero at the end of SS.

The sudden convergence of the swing foot orientation (about the X and Z axis) to zero is enforced by the mass of the biped at touch-down.

Finally, in order to show that the ZRAM-based controller can produce sustainable walking cycles, we commanded the biped to take 18-steps in 240 seconds (240000 ms) using the same control gains and walking parameters in Table 5.2 and 5.3. As shown by the reference CMP, COG and actual COG plots, the biped can walk continuously in a stable manner.

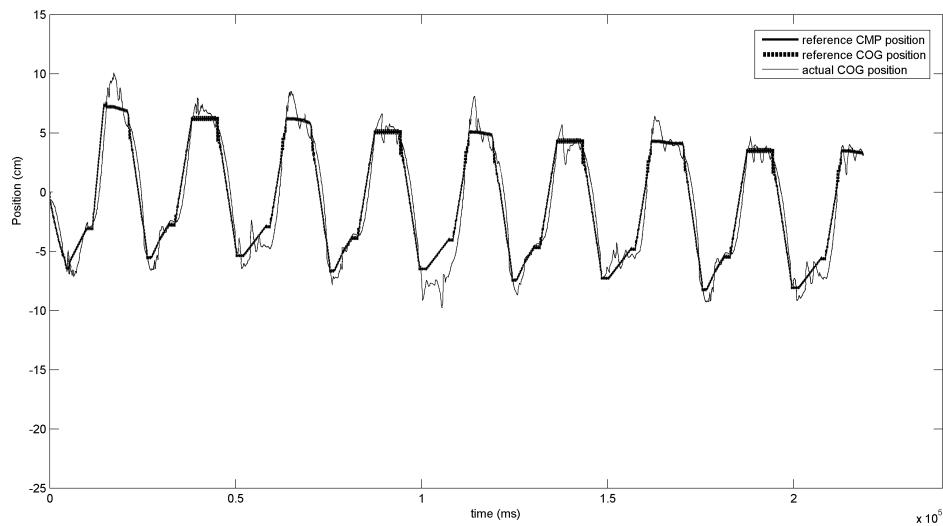


Figure 5.30: Walking test of the ZRAM-based controller: the reference CMP, COG and the actual COG for 18 step cycles (X).

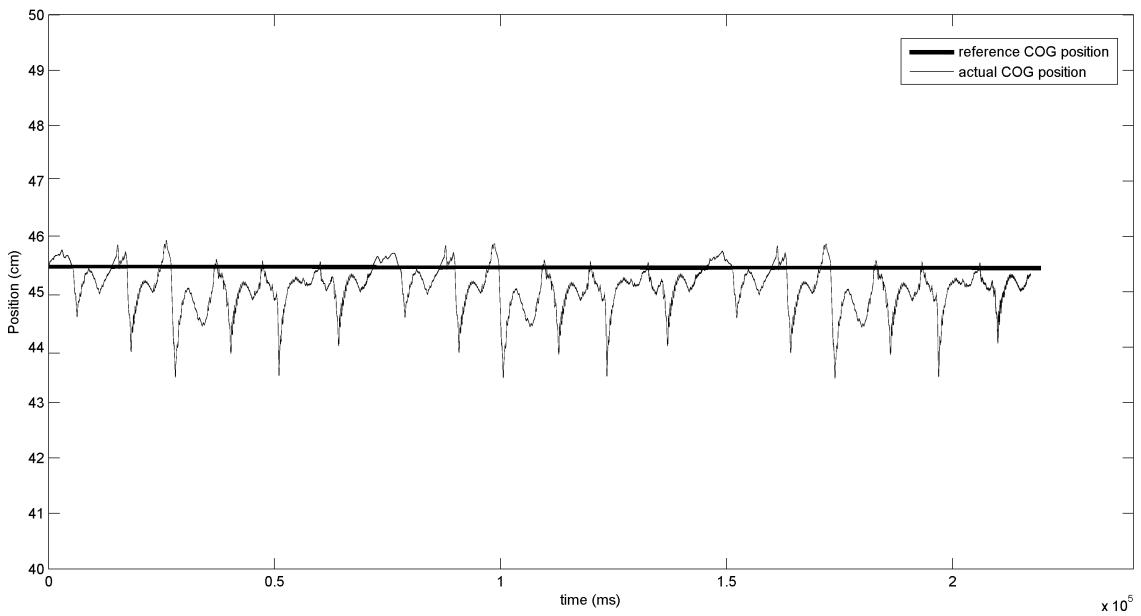


Figure 5.31: Walking test of the ZRAM-based controller: the reference COG and the actual COG for 18 step cycles (Y).

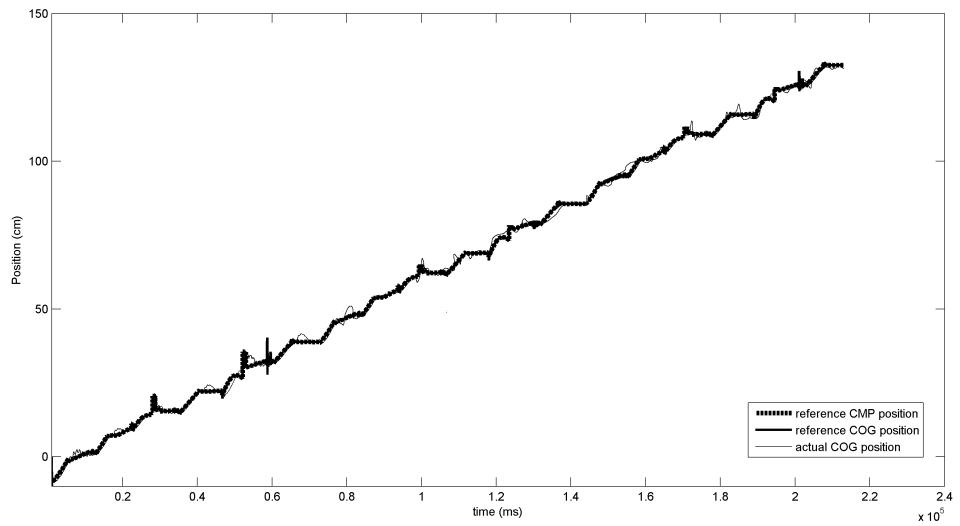


Figure 5.32: Walking test of the ZRAM-based controller: the reference CMP, COG and the actual COG for 18 step cycles (Z).

5.4.2 Perturbation Tests

The second part of the simulation tests in Webots involves demonstrating the robustness of the ZRAM-based controller to external disturbances. Recall the main observations from the walking test of the planar biped:

- The effects of small external disturbances on the biped are reflected in the horizontal components of the GRF. For example, if the biped is pushed forward (in the positive Z direction), there would be an increase in the opposing GRF in the negative Z direction.
- In stable walking, the reference COG is essentially equivalent to the preplanned high level motion that is specified by the reference CMP. On the other hand, a non-zero GRF will cause the ZRAM-based controller to generate a modified reference COG from the ZRAM model (Equation 5.1) that would preserve the rotational dynamic equilibrium of the biped.
- If the actual COG tracks the modified COG, then the COP of the biped will converge to the CMP and the ZRAM condition will be satisfied. Thus, the modified reference COG incorporates: 1) the preplanned high level motion (an external constraint), and 2) the ZRAM dynamic stability constraint that is enforced on the motion of the biped.

We conducted three separate simulation trials and applied a disturbance force at different phases of the walking cycle (Figure 5.33). The point of action of the applied force is the local COG of the *Torso* link, which is approximately 24 cm to 26 cm away from the overall COG of the biped in the vertical direction.

Test I: A 25 N external force (a step input) is applied at the start of DB for a duration of 2 seconds (2000 ms to 4000 ms). The line of action of the applied

force is parallel to the Z axis of the world frame. The resulting centroidal moment that is induced from the applied force is about 6.65 Nm about the X (pitch) axis.

Test II: A -20 N external force is applied in the middle of SS for a duration of 2 seconds (7500 ms to 9500 ms) in the Z direction. The resulting centroidal moment from the applied force is about 5.32 Nm about the X (pitch) axis.

Test III: A -30 N external force is applied in the middle of DB for a duration of 2 seconds (3000 ms to 5000 ms) in the X direction. The resulting centoridal moment from the applied force is about 7.98 Nm about the Z (roll axis).

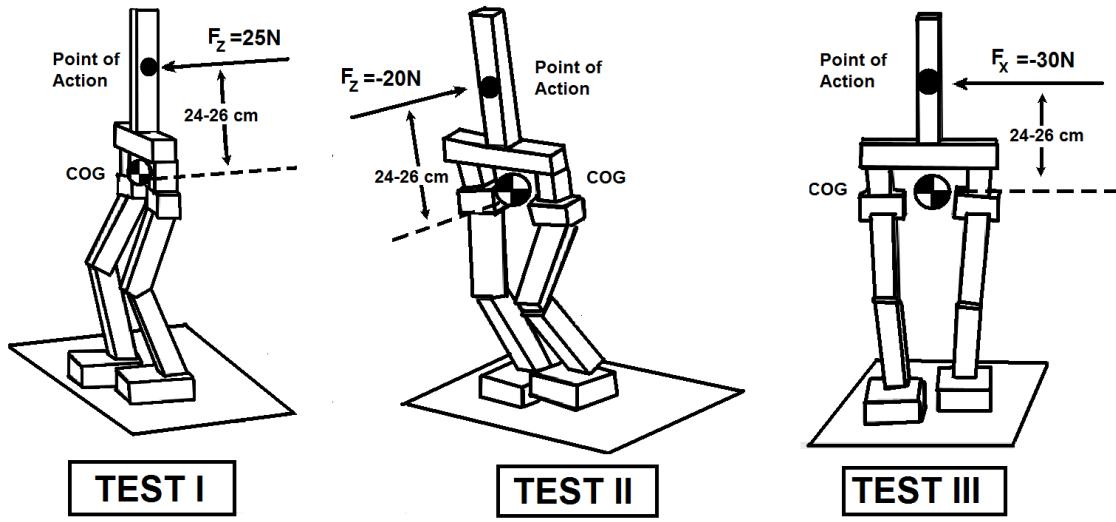


Figure 5.33: Disturbance forces for the three perturbation tests in Webots.

The magnitude of the disturbance forces in the above tests were selected based on the simulation test parameters in a related work [12]. In the perturbation tests for the ZRAM-based controller, we induce an additional angular momentum of 10 Nms to 15 Nms in the biped compared to 12 Nms in [12]. The authors in [12] also conducted a second simulation test with a stronger impact and proved that their

controller can attenuate a disturbance of 24 Nms. However, the controller in [12] is implemented on a simple planar human model and the mass of the model might be greater than ABL-BI (although the mass and dimension of the “human body model” is not stated in [12], it has significantly more links and joints than ABL-BI). A comparison between the ZRAM-based controller and the controller in [12] will be presented in Section 5.5.1.

The walking parameters of the biped in the perturbation tests are same as the ones that were presented in Table 5.3. The control gains, however, must be retuned slightly in order for the modified COG to be tracked accurately. The new control gains are summarized in Table 5.4.

Table 5.4: The control gains for the ZRAM-based controller in the perturbation tests.

High level Objective	Control Gains [X,Y,Z]
COG Position P-Gain (X)	$k_p = 1.0$ for DB and $k_p = 1.2$ for SS
COG Position D-Gain (X)	$k_d = 0.75$ for DB and $k_d = 0.6$ for SS
COG Position P-Gain (Z)	$k_p = 1.0$ for DB and $k_p = 1.2$ for SS
COG Position D-Gain (Z)	$k_d = 0.75$ for DB and $k_d = 0.6$ for SS
COG Position P-Gain (Y)	$k_p = 0.5$
Pelvis Orientation P-Gain	[1.25, 1.00, 1.25]
Swing Foot Position P-Gain	[2.50, 7.00, 5.00]
Swing Foot Orientation P-Gain	[5.00, 2.00, 2.00]

As shown in the time elapsed snapshots of the simulation test in Figure 5.34, the biped is perturbed at the beginning of DB in **Test I** but is able to recover from the disturbance and regain an upright pelvis orientation in SS. The effectiveness

of the ZRAM-based controller in attenuating the additional centroidal moment can be verified by observing the preplanned feedforward motion (reference CMP), the modified reference COG, and the actual COG in Figure 5.35 and the horizontal GRF plots in Figure 5.36.

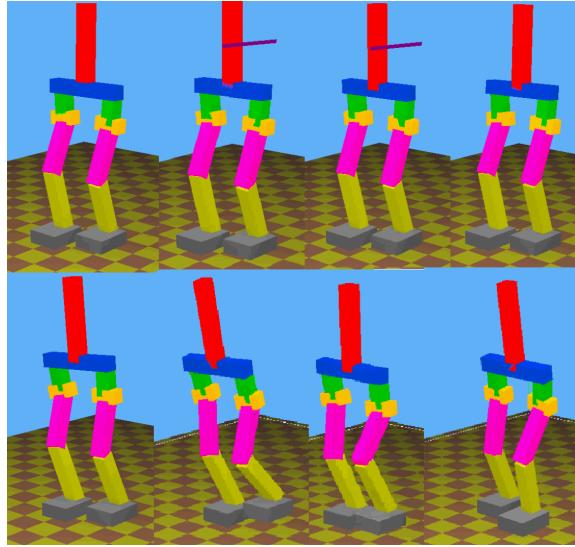


Figure 5.34: Test I: The time-elapsed snapshots of the perturbation test. The disturbance force in the sagittal direction is applied in the beginning of DB.

Unlike the stable walking tests in Section 5.4.1, the reference COG differs significantly from the reference CMP trajectory (maximum deviation of about 8 cm) in the presence of external disturbance. As shown in Figure 5.35 and 5.36, the divergence of the reference COG (Z) corresponds with the increase in the magnitude of the GRF (Z) during the external disturbance between 2000 ms and 4000 ms. Thus, from the negative GRF (imposed on the biped as it pitches forward) and Equation 5.1, the ZRAM-based controller is able to generate a backward reference COG motion to counteract the disturbance and satisfy the ZRAM criterion. Although there are sharp increases/decreases in the horizontal GRF in SS, the ZRAM condition is relaxed and the reference COG closely tracks the reference CMP trajectory. Thus, the reference COG is modified only in the presence of disturbances. Due to various

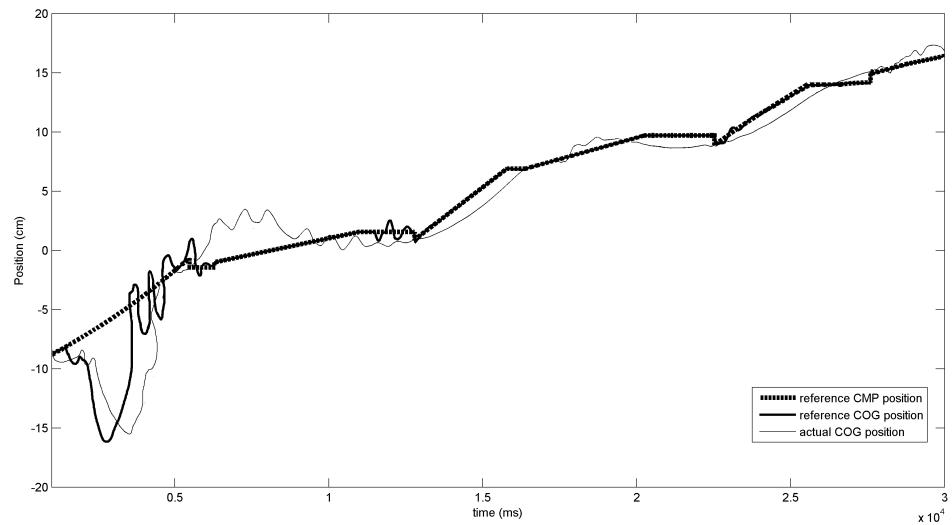


Figure 5.35: Test I: The reference CMP, reference COG and actual COG position with respect to the world frame (Z).

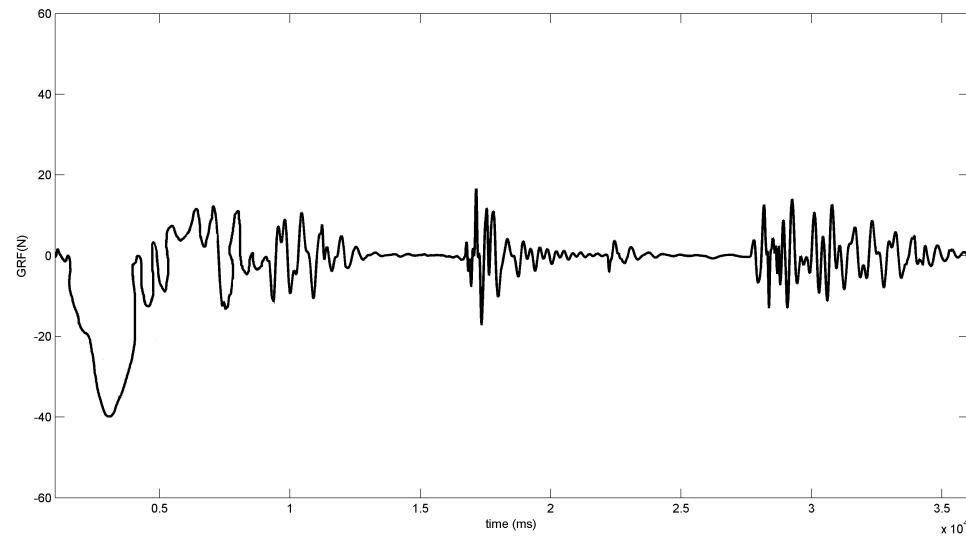


Figure 5.36: Test I: The GRF (Z) vs. time plot.

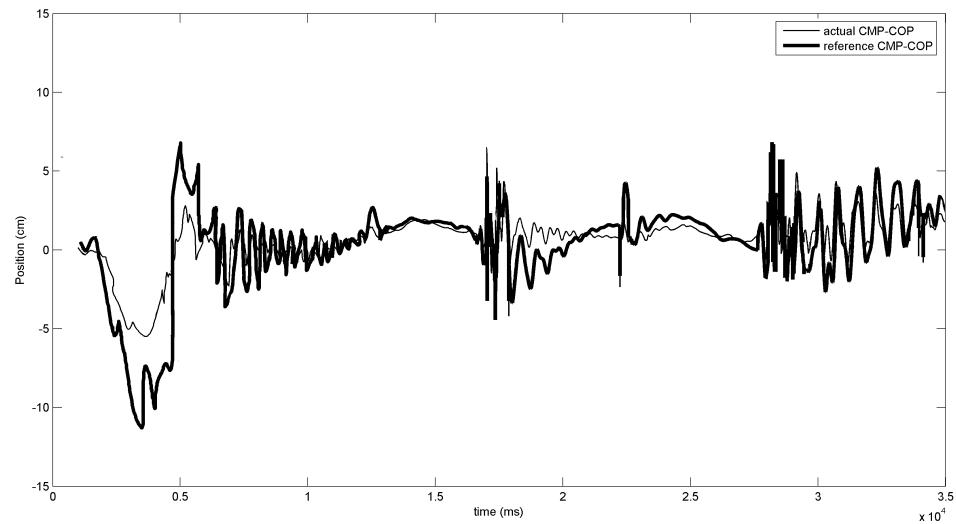


Figure 5.37: Test I: The error between the reference CMP and the COP and the actual CMP and the COP.

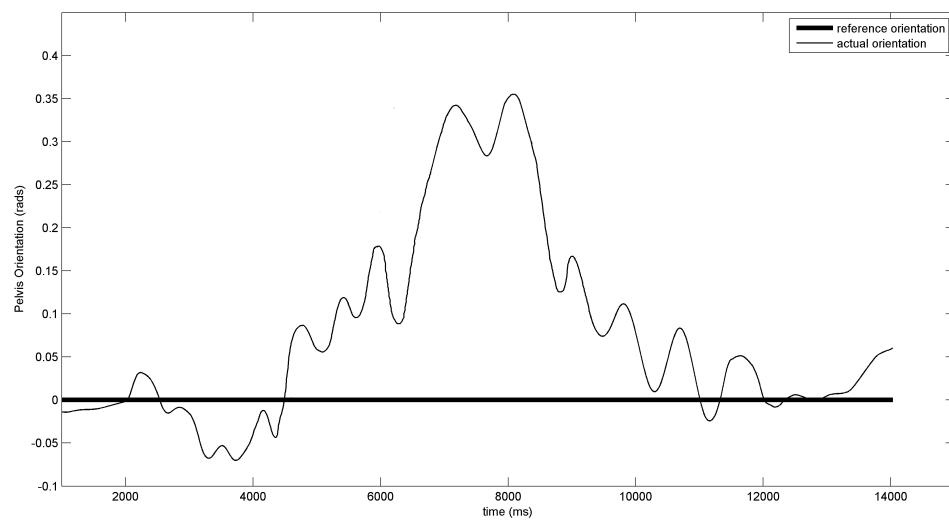


Figure 5.38: Perturbation Test I: The pelvis orientation about the X axis.

levels of integration and filtering in the *Closed-loop Kinematics Controller*, there is a $500 - 700\text{ ms}$ delay between the actual and reference COG. Although the tracking is not exact, the actual COG motion is able to attenuate the external disturbance. The compensatory effects of the actual COG motion is shown in Figure 5.37, where the COP converges to the reference CMP at about 7000 ms after the perturbation. Thus, the results in *Webots* coincide with the results in the 2D-simulation. In addition, as shown in 5.38, the *Closed-loop Kinematic Controller* is able to control the pelvis orientation in SS and regain an upright posture after the perturbation and the effects of the weight-shift in DB.

In summary, the above results verify that:

- The ZRAM-based controller is able to generate a modified feedforward motion in realtime to attenuate the external disturbance.
- The kinematics controller is able to track the modified COG reasonably well (with $500 - 700\text{ ms}$ delay).
- The distance between the COP and the reference CMP eventually reduces to zero after about 7000 ms , implying that the additional centroidal moment is attenuated and the rotational equilibrium of the biped is regulated by the ZRAM-based controller.

The time elapsed snapshots, ground reference points, and the horizontal GRF for **Test II** are plotted in Figure 5.39, 5.40, and 5.41 respectively.

The results are similar to the Test I:

- The external disturbance is reflected in the horizontal GRF as the biped is pushed backwards.
- In response to that, the ZRAM-based controller generates a modified reference COG trajectory (a forward motion) that satisfies the ZRAM criterion.

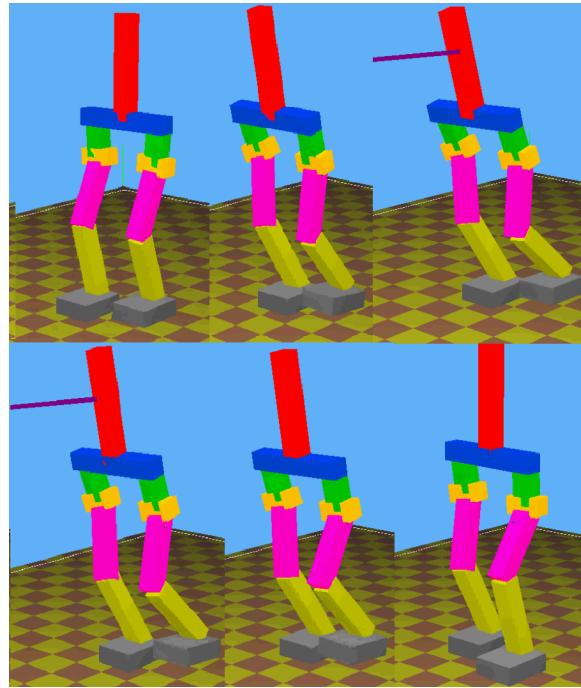


Figure 5.39: Test II: The time-elapsed snapshots of the perturbation test. The disturbance force in the sagittal direction is applied in the middle of SS.

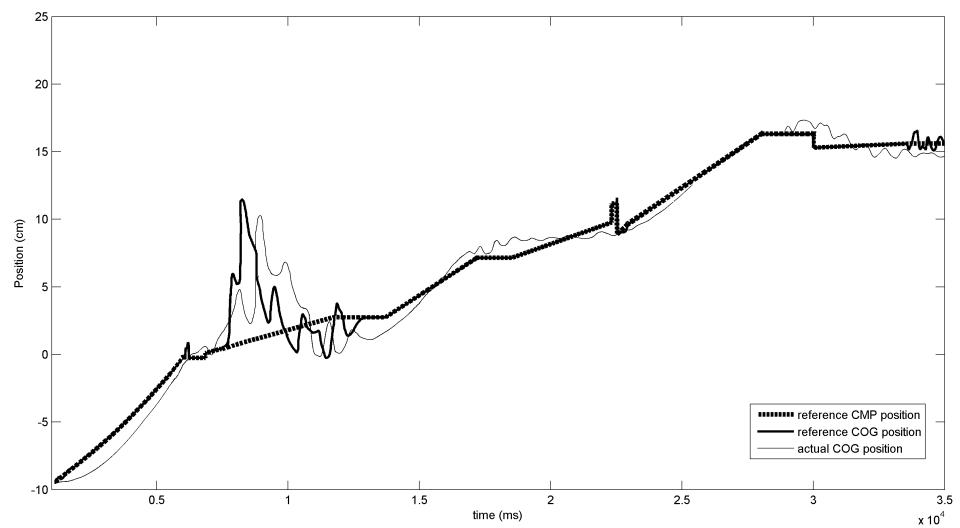


Figure 5.40: Test II: The reference CMP, reference COG and actual COG position with respect to the world frame (Z).

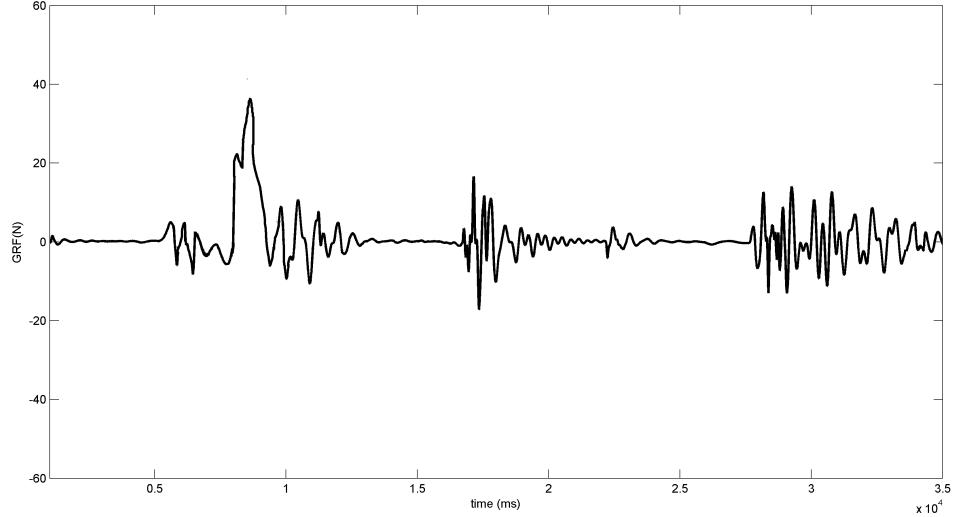


Figure 5.41: Test II: GRF (Z) vs. time plot

- The actual COG tracks the reference COG reasonably well with slight delays, and in doing so, the ZRAM-based controller was able to attenuate the disturbance. The reference COG of the biped reverts back to the user-specified high level motion (specified by the CMP) after 15000 *ms*.

Test III examines the effects of an external disturbance that is applied in the lateral direction of the biped (induces an centroidal moment about the Z axis). The time elapsed snapshots, ground reference points, and the horizontal GRF are plotted in Figure 5.42, 5.43, and 5.44 respectively. Since the biped has a greater stability margin in the lateral direction than in the sagittal direction in DB, the ZRAM-based controller can attenuate a slightly higher external force than the previous tests.

The simulation results show that the ZRAM-based controller can attenuate an external disturbance force that is applied in the lateral direction. Again, the reference COG is modified according to the ZRAM criterion and the actual COG can track the reference COG, albeit a slight delay. As the result, the centroidal moment is regulated and the biped eventually returns to its original feedforward motion after

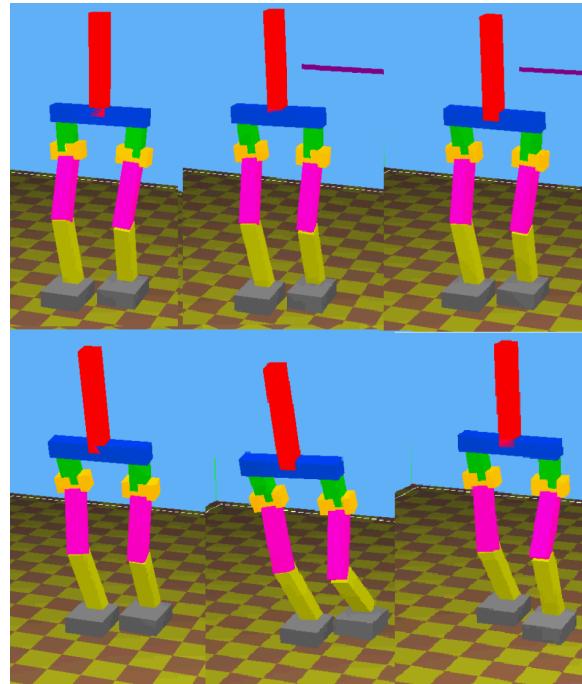


Figure 5.42: Test III: The time-elapsed snapshots of the perturbation test. The disturbance force in the lateral direction is applied in the middle of DB.

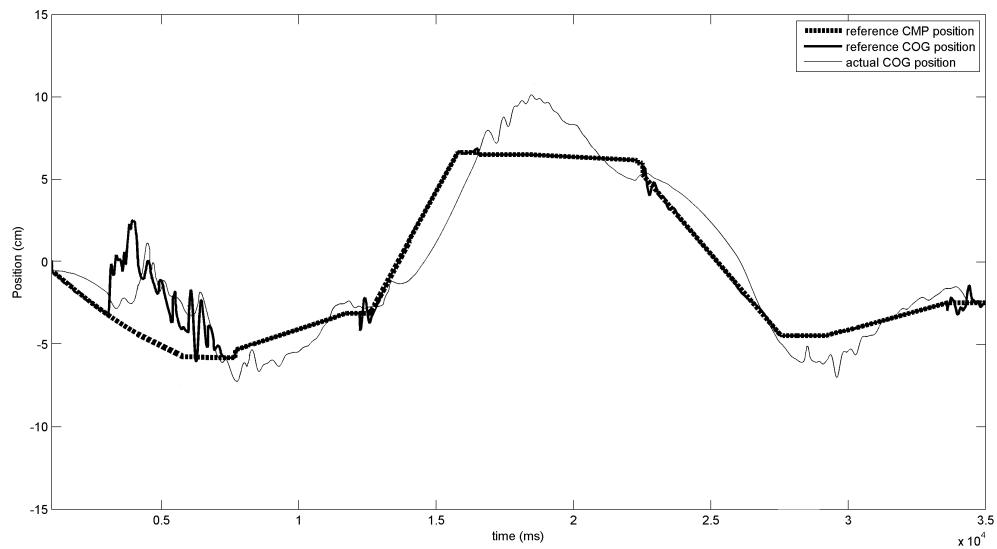


Figure 5.43: Test III: The reference CMP, reference COG and actual COG position with respect to the world frame (X).

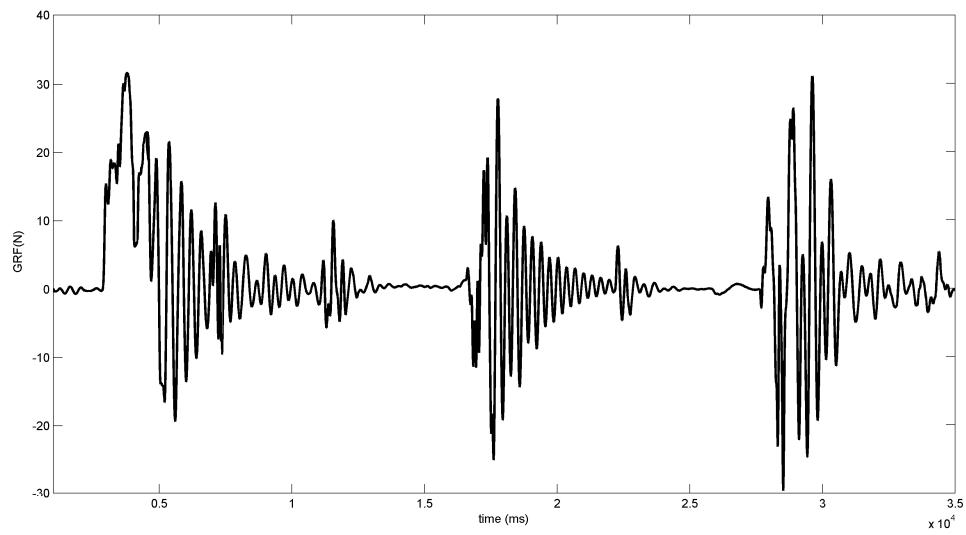


Figure 5.44: Test III: GRF (X) vs. time plot

7500 ms.

The perturbation tests described in this section were also conducted on the simulated biped using the ZMP-based controller. As shown in the time-elapsed snapshots of the simulation test in Figure 5.45, the ZMP-based controller was not able to attenuate the external disturbance and the biped tips over.

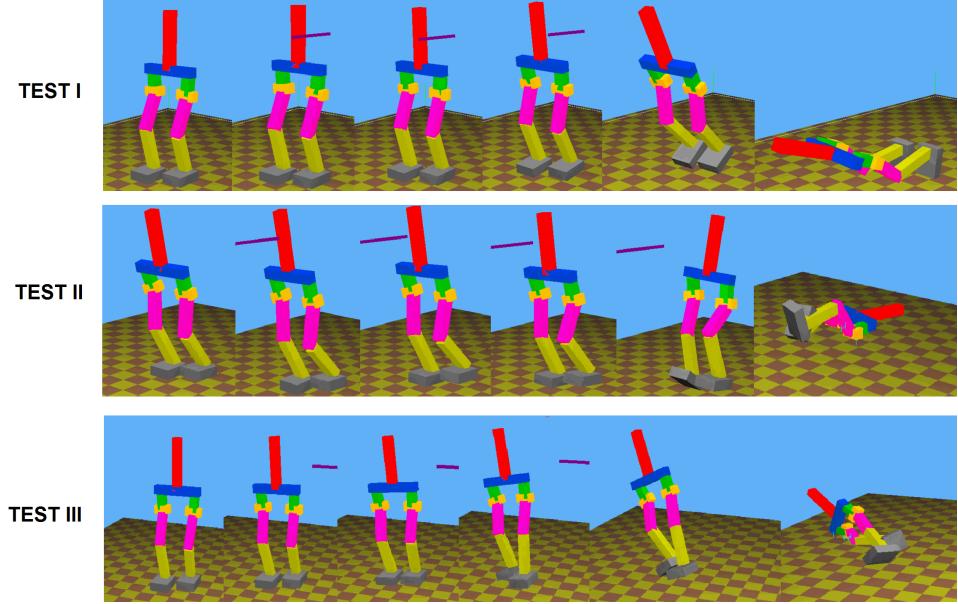


Figure 5.45: Perturbation tests: The time-elapsed snapshots of the perturbation test for the ZMP-based controller.

As mentioned previously, since the rigid feet of ABL-BI has an abrupt edge, there would be insufficient GRF for the biped to regain balance once it loses foot contact and starts to tip over. Therefore, it is important for the controllers in ABL-BI to respond effectively to external disturbances before their effects becomes critical. Unfortunately, due to the large number of feedback loops in the ZMP-based controller and the sensitively of the *COG velocity control loop* to changes in the control gains, it is very difficult to tune the control gains and obtain the required responses from the controller after the perturbation. As shown in Figure 5.46, the actual COG was not able to track the reference COG and the controller fails to attenuate the disturbance.

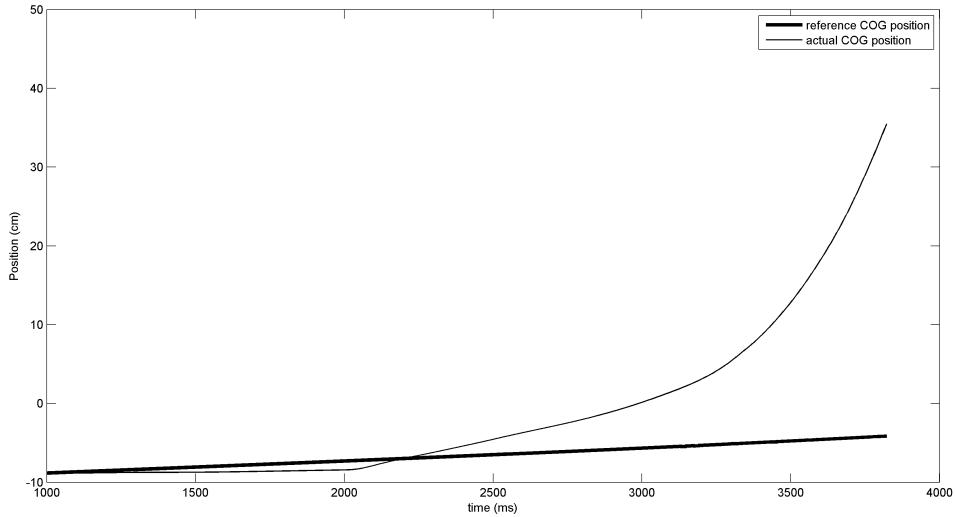


Figure 5.46: Perturbation Test I: Reference COG (Z) and actual COG (Z) position of the ZMP-based controller.

5.5 Summary

In this Chapter, we presented a new strategy that regulates the rate of change of angular momentum. This strategy is based on the ZRAM condition, which is an underlying biomechanical principle of human walking that is represented by the nonlinear coupling between the CMP, the COG and the GRF. During stable walking, the reference COG is almost identical to the reference CMP. In this case, the ZRAM-based controller is essentially a feedback controller that tracks a user-specified reference motion. In the presence of disturbances however, a modified reference COG trajectory is generated based on the ZRAM criterion to attenuate disturbances and regain rotational stability.

The walking tests on the planar biped verify that the ZRAM-based controller can regulate the centroidal moment of the biped by minimizing the distance between the COP and the reference CMP after it has been perturbed by an external force.

In addition, the results suggest that the additional spin angular momentum that is induced by disturbances can be attenuated indirectly by the ZRAM-based controller.

The walking tests in *Webots* verify that the ZRAM-based controller can generate a sustainable stable walking gait. The robustness of the ZRAM-based controller is further validated by a series of tests in *Webots* where the biped is perturbed at different phases of the walking cycle. In all cases, the biped is able to recover from the disturbance and return to its original gait. We will summarize the main qualities of the ZRAM-based controller below:

Gait Modification in Realtime: The main feature of the ZRAM-based controller is that it can generate a modified feedforward motion to maintain dynamic stability and attenuate external disturbances based on the ZRAM criterion. As shown in the perturbation tests in *Webots*, after the biped is perturbed, the ZRAM-based controller is able to actively replan the reference COG trajectory in real-time and guide the biped back to its nominal walking gait in a stable manner. In contrary, some ZMP-based controllers such as the ZMP-based controller and [51, 52, 76, 77] solely rely on feedback control to track a fixed reference COG trajectory. However, tracking a fixed reference COG might not be the most appropriate response for the biped after a disturbance (Figure 5.46). This is especially true when the actual COG deviates significantly from the reference COG. After a perturbation, the primary objective of the robot controller should be changed from tracking a pre-planned reference trajectory to generating compensatory motions to counteract the disturbance. In some cases, it is not even feasible to track a fixed reference trajectory due to the internal constraints of the biped. In addition, the control gains must be retuned after a perturbation in order to maintain the stability of the controller. However, it is impractical to prepare a set of control gains for every possible situation that

the biped might encounter. Therefore, robot controllers must rely on both feedback control and a modified feedforward motion in order to deal with external disturbances. For this reason, the ZRAM-based controller is more flexible and robust than the ZMP-based controller and other related controllers.

Consistent with Biomechanical Principles: The ZRAM-based controller regulates the centroidal moment and is consistent with the “ZRAM mode” that is observed in human walking in the presence of small external disturbance (please refer to Section 5.1.3). Therefore, the ZRAM-based controller can be classified as a bio-inspired control strategy and has the potential to produce biologically realistic gaits. Although some authors argue that it is necessary to directly control the spin angular momentum throughout the walking cycle (as done by the controllers in [40, 78]), we opt to regulate the centroidal moment for two reasons:

- Biomechanical studies in [35, 56] show that the centroidal moment is regulated for small rotational movements and external disturbances in human walking (note that similar observations were made in the walking tests in *Webots*). Therefore, it is sufficient to use the ZRAM condition in normal walking and only apply direct control on the spin angular momentum when there is a large external disturbance in the biped. The biomechanical studies also show that humans use a different controller to attenuate large disturbances. Unfortunately, this “non-ZRAM” controller mode was not implemented on ABL-BI in this thesis; however, it will a future development of this project and a possible control algorithm will be suggested in Chapter 6.
- Since the controller is based on the ZRAM criterion, we can simplify the design of the controller by exploiting the relationship between the ground

reference points.

Modular: The *Stable Reference Trajectory Generation* block in Figure 5.1 is a modular component of the ZRAM-based controller. Therefore, we can replace the ZRAM criterion with new reference trajectory generation methods and reuse the *Closed-loop kinematic controller* and the *Local PID controller* to control the posture of the biped. In addition, since the ZRAM-based controller is based on the fundamental principles and characteristics of bipedal locomotion (ex: ground reference points, conservation of angular momentum), it can be applied to bipeds/humanoids with an arbitrary number of DOFs and links. As discussed extensively in Chapter 4, the performance of any walking controller (including the ZRAM-based controller) also depends largely on the physical properties of the biped (ex: weight distribution, biped kinematics and actuation limits); therefore, the performance of the ZRAM-based controller will most likely vary between different robots.

5.5.1 Comparing the ZRAM-based Controller to a Similar Control Strategy

An angular momentum control strategy that is closely related to the ZRAM-based controller is proposed by Komura et al. in [12]. Similar to the ZRAM-based controller, Komura's controller uses a simplified model called the Angular Momentum Pendulum Model (AMPM) to approximate the dynamics of the biped. From AMPM, a nonlinear equation that relates the change in the angular momentum between two time instances (δM) as a function of the COG acceleration is derived. The main objective of the controller is to find a COG acceleration that counteracts δM (induced by external disturbances) in the biped. The main problem is that a close

form solution for the COG acceleration might not exist for the current state of the biped. Although an approximated solution can be calculated iteratively using numerical methods, this would further complicate the control strategy and increase the computational requirement. Also, Komura's controller was only implemented and tested on a planar human model; therefore, the question remains as to whether or not the controller is simple enough to be implemented in real-time on a 3-D biped. In contrary, the ZRAM-based controller is a simpler control strategy that only deals with the low-dimensional representation of the overall motion and properties of the biped (ex: the COG and the COP). The advantage of the ZRAM-based controller is that the dynamic balance of the biped can be maintained by regulating the distance between the COP and the CMP using the ZRAM model, which is a very simple relationship compared to the equation for the spin angular momentum in [12].

In addition, the posture controller in [12] involves a complicated process of calculating and comparing the overall moment of inertia of a *planar human model* with respect to the COG between two time instances. In contrast, the ZRAM-based controller uses a simple feedback controller to directly control the posture of the biped. The simulation test results in *Webots* show that by imposing external constraints on the vertical COG position and the pelvis orientation, the biped can maintain an upright posture.

The above comparison illustrates that although it is important to for a controller to generate stable walking gaits and attenuate external disturbances; at the same time, it must also be simple and efficient enough to be implemented in realtime on a 3-D biped. The ZRAM-based controller is a relatively simple controller because it exclusively relies on the nonlinear coupling between the ground reference points to

generate a stable reference feedforward motion in realtime. In addition, the ZRAM-based controller uses feedback control and kinematic redundancy resolution methods to track the reference high-level trajectories. The simulation test results in the planar biped and the *Webots* model show that the ZRAM-based controller is a simple but effective controller that is able to generate stable walking gaits in the presence of external disturbances.

Chapter 6

Conclusions

6.1 Summary

In this thesis, we implemented and validated two model-based realtime walking controllers on a simulated model of ABL-BI. The first controller (the ZMP-based controller) was a modification of an existing ZMP-based realtime controller. Using this controller, the simulated biped was able to take multiple steps in a 3-D simulation. The second controller (the ZRAM-based controller) was an innovative walking controller that was able to perform continuous walking while maintaining dynamic stability by regulating the rate of change of angular momentum. Although many angular momentum-based walking controllers had been suggested in the past, this is the first real-time walking implementation that exploits the relationship between CMP and COG. By regulating the distance between the COP and the reference CMP, the simulation results also suggest that the ZRAM-based controller can indirectly attenuate the additional spin angular momentum in the biped that is caused by disturbances.

During the course of developing and validating the walking controllers, several

other contributions were made. First, a simulated biped model based on the kinematic and mass specifications of the actual ABL-BI platform was constructed in a 3-D simulation environment. The forward kinematics of the model was derived to approximate the states of the biped using only sensory information that is available to the actual biped as long as the modelling of the biped is reasonably accurate. Thus, the controllers were designed to be transferrable from the simulation to the actual platform. Second, the kinematics of the biped was separated into different kinematic chains to facilitate high-level task assignment to different segments of the biped. A closed loop postural controller based on task prioritization was formulated and different priority levels were suitably assigned to the high level tasks. Simulation results show that the biped can achieve high-level tasks through the coordination of its joint motions. Third, supported by evidence from simulation tests, we were able to identify several features in the ABL-BI platform that affects the performance of the walking controllers. These results highlight specific areas of improvement in the mechanical and controller design of ABL-BI in the future developments of this project. The main accomplishments of this thesis are: 1) the kinematic and dynamic modelling of ABL-BI, 2) the design and implementation of two walking controllers, with innovative elements in the second controller, and 3) the validation of the walking controllers in simulation. These contributions establishes the groundwork for future developments of ABL-BI.

6.2 Conclusions

This section summarizes the key points and conclusions for each chapter in this thesis:

Chapter 2 This chapter provided introductory information on the ABL-BI platform, the simulated ABL-BI model, and the 3-D simulation environment *Webots*. The design and assembly of the mechanical structure, and the distributed software

and hardware control architecture were done in the ABL-Lab. Based on the mass and kinematic specifications of the ABL-BI platform, a simulated biped was constructed in *Webots*. The basic information of the kinematic model of ABL-BI was provided, this includes the frame assignments and the mass properties of the links. In addition, background information on the features and functionalities of *Webots* that were relevant to the ABL-BI simulation were also provided. This include: 1) simulated sensors that were available in *Webots* (ie: the physical states that can be “measured” directly from the simulation), 2) the control layers (robot controller, supervisory controller and the physics plugin), and 3) the ground model and collision detection method.

Chapter 3 This chapter presented components that are essential to most real-time modelled-based walking controllers, regardless of what stability criterion or simplified dynamics model they use. The locomotion state machine divided the walking cycle into sequential phases to simplify motion planning and control. The full kinematics of the biped was a “tree structure” that can be broken down into multiple open kinematic chains. Depending on the locomotion state, each segment of the biped assumed a specific role and different high level tasks were assigned to the “end-effectors” of each kinematic chain. A *Closed-loop Kinematics Controller* based on task prioritization was developed to facilitate biped postural control. Two simplified models were introduced to facilitate real-time control: the 3-D LIPM was used for the ZMP-based controller and the ZRAM model was used in the ZRAM-based controller. We conclude the findings of this chapter by the following points:

- There were several advantages in treating the kinematics of the biped as multiple kinematic chains: 1) it was intuitive to assign specific tasks to different segments of the biped, and 2) we can avoid applying complex

kinematic constraints on the biped, especially in DB where the motion of the two stance legs might come into conflict with one another.

- The forward kinematics equation for each kinematic chain were derived. The “end-effector” for each kinematic chain was associated with key state variables in the walking controller such as the swing foot frame and the COG position. The kinematic equations were validated by comparing the calculated values of the state variables (ex: swing foot position) with the “measured” values from the simulated sensors in *Webots*. The results showed that the calculated values closely tracked the measured values.
- The *Closed-loop Kinematic Controller* was validated by simulation tests. Given a set of task-space reference trajectories, the goal of the controller was to produce the appropriate joint motions to track the reference trajectories. Overall, the tests show that the error between the actual and reference trajectories were small and the high level tasks conformed to their assigned priority level.

Chapter 4 The the ZMP-based controller based on an existing ZMP-based real-time walking controller [55] was implemented. Two modifications were made to the original controller: 1) to facilitate motion planning *in the position level*, an outer COG position feedback loop was added to the controller, and 2) instead of using the weighted-pseudoinverse Jacobian to resolve the reference task-space velocities to reference joint velocities, a more efficient *Closed-loop Kinematics Controller* based on task prioritization was used. The ZMP-based controller was implemented in the simulated biped and was verified in a walking test in *Webots*. The following conclusions were drawn from the simulation results:

- The biped was able to walk in a stable manner in the simulation test, despite the fact that the COG was not accurately tracked at all phases of

the walking cycle. The error was attributed to the inertial effects of the swing leg and numerical filtering and integration. In addition, simulation results show that the ZMP-based controller was able to limit the reference COG velocity (generated by the COG position PD controller) and produce dynamically stable gaits based on the ZMP criterion.

- The reference swing foot position/orientation trajectories were closely tracked by the ZMP-based controller. This indicates that the biped was able to attain accurate foot placements and successfully perform lift-off and touch-down.
- The walking speed of the biped was significantly slower (about $10\times$ slower) than other bipeds. Additional simulation tests show that the mass distribution of ABL-BI was the main limiting factor of the walking speed. Other factors include: 1) the limitation of the LIPM (not being able to capture the dynamics of the swing leg), 2) the high stiffness of the ground model, and 3) the overly complicated, multi-loop control architecture of the ZMP-based controller.

Chapter 5 An innovative walking controller based on CMP planning and COG control was introduced (the ZRAM-based controller). This strategy is consistent with underlying biomechanical principles of human walking and therefore has the potential to produce biologically realistic gaits. By regulating the rate of change of angular momentum using the nonlinear relationship between the COG, CMP, and the GRF, the controller was able to generate stable walking by maintaining rotational equilibrium. A series of walking and perturbation tests were used to validate the ZRAM-based controller in *Webots*. The following are the important results from the simulation tests:

- The ZRAM-based controller was first validated in a 2-D simulation. The

results show that 1) the biped can generate stable walking gaits and closely track the reference trajectories, 2) the biped was able to generate and a track a modified reference COG trajectory in the presence of disturbance (an external moment of 2.5 Nm) to maintain dynamic balance, and 3) after a perturbation, the ZRAM-based controller was able to indirectly generate a compensatory moment by regulating the distance between the COP and the reference CMP (different from the actual CMP) to attenuate the additional spin angular momentum in the biped.

- The ZRAM-based controller was further verified in a 3-D walking test in *Webots*. Again, the results show that the biped was able to walk continuously in a stable manner. In addition, due to the improvement in the control architecture of the ZRAM-based controller and the increased damping in the ground model, the DB period was reduced from 7500 ms (without the Torso Roll motion) in the ZMP-based controller to about 5500 ms in the ZRAM-based controller. In addition, the swing period was shortened from 6500 ms in the ZMP-based controller to 5500 ms in the ZRAM-based controller.
- In order to evaluate the robustness of the ZRAM-based controller to external disturbances, the biped was subjected to perturbation at different phases in the walking cycle. The results showed that the biped was able to attenuate an external moment of up to 6 Nm about the COG. These results were comparable to similar tests there were conducted in [12]. Therefore, we concluded that the ZRAM-based controller was able to recover from small disturbances and return to its nominal walking gait.

6.3 Suggestions for Future Work

This section presents several suggestions for the future development of this project:

Modification of the ABL-BI Platform: The optimization of the mechanical structure of the biped is seldom taken into account in the literature. However, the results in Chapter 4 showed that the performance of a control strategy is largely dependent on the the mass distribution and kinematics of the biped. In the view of this, two improvements can be made to ABL-BI:

- Increase the width of the foot to enlarge the support polygon and reduce the weight-shift distance in the lateral direction. This can also be achieved by reducing the width of the pelvis (and thus the distance between the two stance legs).
- Modify the mass distribution of the biped. Since most of the mass is contributed by the motors, it is difficult to reduce the weight of the lower body. The only option is to introduce additional “dummy weights” (which represents the mass of the arms and head of ABL-BI) to the torso to attain a better mass ratio in the upper body.

Exploit the Passive Dynamics of the Biped: One important requirement of walking that was not examined in this thesis was the energy efficiency of the walking gait. The study in [79] show that conventional control approaches in powered bipeds use significantly more energy than their biological counterparts. In [35], the authors demonstrated that in order to generate natural swing leg motion, it is important to minimize both the motor torques and the spin angular momentum. Therefore, energy expenditure is an important aspect in bipedal walking. Several control approaches that are commonly used in passive walkers can be incorporated into the current walking controller of ABL-BI.

For example, instead of “enforcing our will onto the robot” and forcing it to track a pre-defined swing leg trajectory, the controller can rely on gravity to “automatically” generate a natural swing leg motion. In this case, the motors only need to provide positive work to the biped to initiate lift-off and provide negative work to decelerate the legs at touch-down. A compliant ankle joint is also useful for the biped to store potential energy in DB and transfer it to kinetic energy to initiate swing leg lift-off. However, extensive work is needed to design compliant leg mechanisms for ABL-BI. In summary, passive dynamics can be exploited to reduce the complexity of the controller and generate energy efficient gaits. Therefore it can be considered as one of the possible future directions of this project.

The Disturbance Recovery Mode in the ZRAM-based controller: As mentioned in Chapter 5, human has two distinct control modes. During the stable phases of walking, the ZRAM criterion is used maintain dynamic balance (ZRAM mode). However, in the presence of disturbance, humans actively generate a compensatory centrodial moment to regulate its spin angular momentum. Therefore, in order to mimic this aspect of human walking, the ZRAM-based controller that is proposed in this thesis must also have a “Disturbance Recovery (non-ZRAM) mode” that specifically handles external disturbances by regulating the spin angular momentum. Feedback control can be used to regulate the spin angular momentum. Based on the error between the reference and actual spin angular momentum, an actuating reference centroidal moment is generated. By substituting the reference centriodal moment, the measured GRF, and the reference CMP to Equation 3.63a, a reference COG can be calculated. By tracking this reference COG, the biped can generate the necessary

centrodial moment that attenuates the additional spin angular momentum. Although the Disturbance Recovery mode was not implemented or verified in this thesis, it is an extension of the current ZRAM-based controller that can be further developed in the future.

List of References

- [1] R. Kurzeil, *The Age of Spiritual Machines*. Penguin Books, 2000.
- [2] M. Vukobratovic and B. Borovac, “Zero-moment point- thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, pp. 157–153, 2004.
- [3] M. Popovic, A. Goswami, and H. Herr, “Ground reference points in legged locomotion: Definitions, biological trajectories and control implications,” *The International Journal of Robotics Research*, vol. 24, pp. 1013–1032, 2005.
- [4] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, “Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking,” in *Proc. IEEE Int Robotics and Automation Conf*, vol. 1, pp. 368–374, 1999.
- [5] A. Takanishi, H. Lim, M. Tsuda, and I. Kato, “Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface,” in *Proc. IEEE Int. Workshop Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications' IROS '90*, pp. 323–330, 1990.
- [6] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of honda humanoid robot,” in *Proc. IEEE Int Robotics and Automation Conf*, vol. 2, pp. 1321–1326, 1998.
- [7] A. Konno, K. Nagashima, R. Furukawa, K. Nishiwaki, T. Noda, M. Inaba, and H. Inoue, “Development of a humanoid robot saika,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems IROS '97. Conf*, vol. 2, pp. 805–810, 1997.
- [8] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, “Planning walking patterns for a biped robot,” vol. 17, no. 3, pp. 280–289, 2001.
- [9] T. Arakawa and T. Fukuda, “Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of ga, ep layers,” in *Proc. Conf. IEEE Int Robotics and Automation*, vol. 1, pp. 211–216, 1997.

- [10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, “A realtime pattern generator for biped walking,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA ’02*, vol. 1, pp. 31–37, 2002.
- [11] J. H. Park and K. D. Kim, “Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control,” in *Proc. IEEE Int. Robotics and Automation Conf.*, vol. 4, pp. 3528–3533, 1998.
- [12] T. Komura, H. Leung, S. Kudoh, and J. Kuffner, “A feedback controller for biped humanoids that can counteract large perturbations during gait,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2005*, pp. 1989–1995, 2005.
- [13] S. Shimmyo, T. Sato, and K. Ohnishi, “Biped walking pattern generation by using preview control with virtual plane method,” in *Proc. 11th IEEE Int Advanced Motion Control Workshop*, pp. 414–419, 2010.
- [14] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, “Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf.*, vol. 3, pp. 2684–2689, 2002.
- [15] G. A. Pratt and M. M. Williamson, “Series elastic actuators,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems 95. ’Human Robot Interaction and Cooperative Robots’ Conf.*, vol. 1, pp. 399–406, 1995.
- [16] J. Pratt, P. Dilworth, and G. Pratt, “Virtual model control of a bipedal walking robot,” in *Proc. Conf. IEEE Int Robotics and Automation*, vol. 1, pp. 193–198, 1997.
- [17] J. Pratt, C. Chew, A. Torres, P. Dilworth, and G. Pratt, “Virtual model control: An intuitive approach for bipedal locomotion.,” *International Journal of Robotics Research*, vol. 20, pp. 129–143, 2001.
- [18] Y. Mao, J. Wang, P. Jia, S. Li, Z. Qiu, L. Zhang, and Z. Han, “A reinforcement learning based dynamic walking control,” in *Proc. IEEE Int Robotics and Automation Conf.*, pp. 3609–3614, 2007.
- [19] H. Benbrahim, *Biped Dynamic Walking using Reinforcement Learning*. PhD thesis, University of New Hampshire, 1996.
- [20] T. McGeer, “Passive dynamic walking,” *Int. J. Robotics Research*, vol. 9, pp. 62–82, 1990.

- [21] M. Ahmadi, *Stable control of a one-legged robot exploiting passive dynamics*. PhD thesis, McGill University, 1998.
- [22] T. Narukawa, K. Yokoyama, M. Takahashi, and K. Yoshida, “A simple 3d straight-legged passive walker with flat feet and ankle springs,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2008*, pp. 2952–2957, 2008.
- [23] A. Goswami, B. Espiau, and A. Keramane, “Limit cycles in a passive compass gait biped and passivity-mimicking control laws,” *Journal of Autonomous Robots*, vol. 4(3), pp. 273–276, 1997.
- [24] M. Garcia, A. Chatterjee, and A. Ruina, “Speed, efficiency, and stability of small-slope 2d passive dynamic bipedal walking,” in *Proc. IEEE Int Robotics and Automation Conf*, vol. 3, pp. 2351–2356, 1998.
- [25] P. Channon, S. Hopkins, and D. Phan, “Derivation of optimal walking motions for a biped walking robot,” *Robotica*, vol. 10(2), pp. 165–172, 1992.
- [26] M. Rostami and G. Bessonnet, “Impactless sagittal gait of a biped robot during the single support phase,” in *Proc. IEEE Int Robotics and Automation Conf*, vol. 2, pp. 1385–1391, 1998.
- [27] L. Roussel, C. Canudas-De-Wit, and A. Goswami, “Generation of energy optimal complete gait cycles for biped robots,” in *Proc. IEEE Int Robotics and Automation Conf*, vol. 3, pp. 2036–2041, 1998.
- [28] O. Barker, R. Beranek, and M. Ahmadi, “Design of a 13 degree-of-freedom biped robot with a can-based distributed digital control system,” in *Proc. IEEE International Conference on Advanced Intelligent Mechatronics AIM ’10*, pp. 836–841, 2010.
- [29] D. A. Winter, *Biomechanics and motor Control of Human Movement*. John Wiley & Sons, 1990.
- [30] M. Gienger, K. Löffler, and F. Pfeiffer, “Towards the design of a biped jogging robot,” in *Proc. IEEE International Conference on Robotics and Automation*, 2001.
- [31] J. Craig, *Introduction to Robotics: Mechanics and Control, Third Edition*. Pearson Education, 2004.

- [32] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
- [33] G. Taga, “A model of the neuromusculo-skeletal system for human locomotion i. emergence of basic gait.,” *Biological Cybernetics*, vol. 73, pp. 97–111, 1995.
- [34] K.-H. Ahn and Y. Oh, “Walking control of a humanoid robot via explicit and stable com manipulation with the angular momentum resolution,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, pp. 2478–2483, 2006.
- [35] M. Popovic, A. Hofmann, and H. Herr, “Angular momentum regulation during human walking: biomechanics and control,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '04*, vol. 3, pp. 2405–2411, 2004.
- [36] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education Inc., 2010.
- [37] B. Siciliano, *Robotics: Modelling, Planning, and Control*. Springer-Verlag London Ltd, 2009.
- [38] M. T. Farrell and H. Herr, “Angular momentum primitives for human turning: Control implications for biped robots,” in *Proc. 8th IEEE-RAS Int. Conf. Humanoid Robots Humanoids 2008*, pp. 163–167, 2008.
- [39] J.-I. Yamaguchi, A. Takanishi, and I. Kato, “Development of a biped walking robot compensating for three-axis moment by trunk motion,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems '93 IROS '93*, vol. 1, pp. 561–566, 1993.
- [40] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Resolved momentum control: humanoid motion planning based on the linear and angular momentum,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, vol. 2, pp. 1644–1650, 2003.
- [41] I.-W. Park, J.-Y. Kim, and J.-H. Oh, “Online biped walking pattern generation for humanoid robot khr-3(kaist humanoid robot - 3: Hubo),” in *Proc. 6th IEEE-RAS Int Humanoid Robots Conf*, pp. 398–403, 2006.
- [42] C. Chevallereau and W. Khalil, “A new method for the solution of the inverse kinematics of redundant robots,” in *Proc. Conf. IEEE Int Robotics and Automation*, pp. 37–42, 1988.

- [43] J. Kieffer, “A path following algorithm for manipulator inverse kinematics,” in *Proc. Conf. IEEE Int Robotics and Automation*, pp. 475–480, 1990.
- [44] B. Benhabib, A. A. Goldenberg, and R. G. Fenton, “A solution to the inverse kinematics of redundant manipulators,” in *Proc. American Control Conf*, pp. 368–374, 1985.
- [45] O. Egeland, J. R. Sagli, I. Spangelo, and S. Chiaverini, “A damped least-squares solution to redundancy resolution,” in *Proc. Conf. IEEE Int Robotics and Automation*, pp. 945–950, 1991.
- [46] S. Chiaverini, “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators,” vol. 13, no. 3, pp. 398–410, 1997.
- [47] Napoleon, S. Nakaura, and M. Sampei, “Balance control analysis of humanoid robot based on zmp feedback control,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, vol. 3, pp. 2437–2442, 2002.
- [48] E. Westervelt, J. Grizzle, C. Chevallereau, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007.
- [49] S. Kajita and K. Tani, “Experimental study of biped dynamic walking in the linear inverted pendulum mode,” in *Proc. Conf. IEEE Int Robotics and Automation*, vol. 3, pp. 2885–2891, 1995.
- [50] T. Sugihara and Y. Nakamura, “A fast online gait planning with boundary condition relaxation for humanoid robots,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2005*, pp. 305–310, 2005.
- [51] J. H. Park and Y. K. Rhee, “Zmp trajectory generation for reduced trunk motions of biped robots,” in *Proc. Conf. IEEE/RSJ Int Intelligent Robots and Systems*, vol. 1, pp. 90–95, 1998.
- [52] S. Kagami, “A fast dynamically equilibrated walking trajectory generation method of humanoid robot,” *Autonomous Robots*, vol. 12, p. 7182, 2002.
- [53] H. Herr and M. Popovic, “Angular momentum in human walking,” *The Journal of Experimental Biology*, vol. 211, pp. 467–481, 2008.
- [54] A. Hofmann, M. Popovic, and H. Herr, “Exploiting angular momentum to enhance bipedal center-of-mass control,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '09*, pp. 4423–4429, 2009.

- [55] T. Sugihara, Y. Nakamura, and H. Inoue, “Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '02*, vol. 2, pp. 1404–1409, 2002.
- [56] M. Popovic, A. Hofmann, and H. Herr, “Zero spin angular momentum control: definition and applicability,” in *Proc. 4th IEEE/RAS Int Humanoid Robots Conf*, vol. 1, pp. 478–493, 2004.
- [57] A. Goswami and V. Kallem, “Rate of change of angular momentum and balance maintenance of biped robots,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '04*, vol. 4, pp. 3785–3790, 2004.
- [58] S. Chiaverini, O. Egeland, and R. K. Kanestrom, “Achieving user-defined accuracy with damped least-squares inverse kinematics,” in *Proc. Fifth Int Advanced Robotics 'Robots in Unstructured Environments', 91 ICAR. Conf*, pp. 672–677, 1991.
- [59] S. Lee, M. Kim, J. Kim, and M. Choi, “Receding horizon viability radius for stability of humanoid robot under external perturbation.,” *Journal of Mechanical Science and Technology*, vol. 24, pp. 1127–1139, 2010.
- [60] C.-L. Shih, J. W. Grizzle, and C. Chevallereau, “Asymptotically stable walking of a simple underactuated 3d bipedal robot,” in *Proc. 33rd Annual Conf. of the IEEE Industrial Electronics Society IECON 2007*, pp. 2766–2771, 2007.
- [61] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic, *Biped Locomotion: Dynamics, Stability and Application*. Springer-Verlag, 1990.
- [62] K. Erbatur and U. Seven, “An inverted pendulum based approach to biped trajectory generation with swing leg dynamics,” in *Proc. 7th IEEE-RAS Int Humanoid Robots Conf*, pp. 216–221, 2007.
- [63] E. Choong, C. Chew, A. Poo, and G. Hong, “Mechanical design of an anthropomorphic bipedal robot,” in *International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management*, 2003.
- [64] V.-H. Dau, C.-M. Chew, and A.-N. Poo, “Mechanical design of an anthropomorphic bipedal robot,” in *Proc. IEEE Conf. Robotics Automation and Mechatronics (RAM)*, pp. 575–580, 2010.

- [65] S. Ito, O. Amano, M. Sasaki, and P. Kulvanit, “A zmp feedback control for biped balance and its application to in-place lateral stepping motion,” *Jounral of Computers*, vol. 3, pp. 23–31, 2008.
- [66] M. R. A. Saavedra, *Stable Locomotion of Humanoid Robots based on Mass Concentrated Model*. PhD thesis, Carlos III University of Madrid, 2008.
- [67] Y. Bachar, “Developing controllers for biped humanoid locomotion,” Master’s thesis, University of Edinburgh, 2004.
- [68] C. Maufroy, *Generation and Stabilization Of Quadrupedal Dynamic Walk Using Phase Modulations Based On Leg Loading Information*. PhD thesis, The University of Electro-Communications, 2009.
- [69] S. Kudoh, T. Komura, and K. Ikeuchi, “Stepping motion for a human-like character to maintain balance against large perturbations,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2006*, pp. 2661–2666, 2006.
- [70] M. Morisawa, S. Kajita, K. Harada, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, “Emergency stop algorithm for walking humanoid robots,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2005)*, pp. 2109–2115, 2005.
- [71] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *Proc. 6th IEEE-RAS Int Humanoid Robots Conf*, pp. 200–207, 2006.
- [72] S.-H. Lee and A. Goswami, “Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots,” in *Proc. IEEE Int Robotics and Automation Conf*, pp. 4667–4672, 2007.
- [73] O. Haavisto and H. Hytyniemi, “Simulation tool of a biped walking robot model,” Tech. Rep. 138, Helsinki University of Technology, Helsinki, Finland, 2004.
- [74] K. Mitobe, G. Capi, and Y. Nasu, “A new control method for walking robots based on angular momentum,” *Mechatronics*, vol. 14, p. 163174, 2004.
- [75] A. Kuo, J. Donelan, and A. Runia, “Energetic consequences of walking like an inverted pendulum: Step-to-step transitions,” *Exercise and Sport Sciences Reviews*, vol. 33, pp. 88–97, 2005.

- [76] C. Zhu and A. Kawamura, “Walking principle analysis for biped robot with zmp concept, friction constraint, and inverted pendulum model,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, vol. 1, pp. 364–369, 2003.
- [77] J. H. Park and H. Chung, “Zmp compensation by online trajectory generation for biped robots,” in *Proc. IEEE Int Systems, Man, and Cybernetics Conf. IEEE SMC '99*, vol. 4, pp. 960–965, 1999.
- [78] K. Mitobe, S. Kaneko, T. Oka, Y. Nasu, and G. Capi, “Control of legged robots during the multi support phase based on the locally defined zmp,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2004)*, vol. 3, pp. 2253–2258, 2004.
- [79] N. Heglund and C. Taylor, “Speed, stride frequency and energy cost per stride: how do they change with body size and gait?,” *The Journal of Experimental Biology*, vol. 1, p. 301318, 1988.

Appendix A

Parameters for the ABL-BI Structural Model

Table A.1: Relative position between each successive frames in the Structural Model.

Successive Frames	Position (X,Y,Z) [cm]
Frame 1 wrt Frame 0 (${}_1^0FD$)	(-11.3,-1.7,7.2)
Frame 2 wrt Frame 1 (${}_2^1FD$)	(0.0,0.0,0.0)
Frame 3 wrt Frame 2 (${}_3^2FD$)	(0.0,-12.0,0.0)
Frame 4 wrt Frame 3 (${}_4^3FD$)	(-0.2,-18.0,0.0)
Frame 5 wrt Frame 4 (${}_5^4FD$)	(1.0,-22.5,0.0)
Frame 6 wrt Frame 5 (${}_6^5FD$)	(0.7,-2.5,7.0)
Frame 7 wrt Frame 0 (${}_7^0FD$)	(11.3,-1.7,-7.2)
Frame 8 wrt Frame 7 (${}_8^7FD$)	(0.0,0.0,0.0)
Frame 9 wrt Frame 8 (${}_9^8FD$)	(0.0,-12.0,0.0)
Frame 10 wrt Frame 9 (${}_{10}^9FD$)	(0.2,-18.0,0.0)
Frame 11 wrt Frame 10 (${}_{11}^{10}FD$)	(-1.0,-22.5,0.0)
Frame 12 wrt Frame 11 (${}_{12}^{11}FD$)	(-0.7,-2.5,7.0)
Frame 13 wrt Frame 0 (${}_{13}^0FD$)	(0.0,0.0,0.0)

Table A.2: Link COG position with respect to nearest frame.

Link COG and nearest frame	Position (X,Y,Z) [cm]
link COG 0 wrt Frame 0 (${}^0P_{G0}$)	(0.0,1.5,-13.6)
link COG 1 wrt Frame 1 (${}^1P_{G1}$)	(-0.2,-2.38,0.0)
link COG 2 wrt Frame 2 (${}^2P_{G2}$)	(-2.9,0.35,0.0)
link COG 3 wrt Frame 3 (${}^3P_{G3}$)	(-1.7,-14.6,0.0)
link COG 4 wrt Frame 4 (${}^4P_{G4}$)	(-0.95,-16.8,0.0)
link COG 5 wrt Frame 5 (${}^5P_{G5}$)	(1.2,-1.4,0.1)
link COG 6 wrt Frame 6 (${}^6P_{G6}$)	(0.0,-0.4,0.9)
link COG 7 wrt Frame 7 (${}^7P_{G7}$)	(0.2,-2.4,0.0)
link COG 8 wrt Frame 8 (${}^8P_{G8}$)	(-2.9,0.35,0.0)
link COG 9 wrt Frame 9 (${}^9P_{G9}$)	(1.7,-14.6,0.0)
link COG 10 wrt Frame 10 (${}^{10}P_{G10}$)	(0.95,-16.8,0.0)
link COG 11 wrt Frame 11 (${}^{11}P_{G11}$)	(1.2,-1.4,0.1)
link COG 12 wrt Frame 12 (${}^{12}P_{G12}$)	(0.0,-0.4,0.9)
link COG 13 wrt Frame 13 (${}^{13}P_{G13}$)	(0.0,14.3,-13.1)

Appendix B

The Transformation Matrices of the Biped Model

The homogenous transformation matrices of Frame i with respect to Frame $i - 1$ is presented in this section. Note that each transformation matrix is dependent on 1) the joint variable θ , and 2) the robot parameters ${}^{i-1}\overrightarrow{FD}_i$ that define the relative position of the origins between each successive frames (please refer to Table Table A.1).

$${}_1^0T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & {}_1^0FD_X \\ s\theta_1 & c\theta_1 & 0 & {}_1^0FD_Y \\ 0 & 0 & 1 & {}_1^0FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1a})$$

$${}_2^1T = \begin{bmatrix} c\theta_2 & 0 & s\theta_2 & \frac{1}{2}FD_X \\ 0 & 1 & 0 & \frac{1}{2}FD_Y \\ -s\theta_2 & 0 & c\theta_2 & \frac{1}{2}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1b})$$

$${}_3^2T = \begin{bmatrix} 1 & 0 & 0 & \frac{2}{3}FD_X \\ 0 & c\theta_3 & -s\theta_3 & \frac{2}{3}FD_Y \\ 0 & s\theta_3 & c\theta_3 & \frac{2}{3}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1c})$$

$${}_4^3T = \begin{bmatrix} 1 & 0 & 0 & \frac{3}{4}FD_X \\ 0 & c\theta_4 & -s\theta_4 & \frac{3}{4}FD_Y \\ 0 & s\theta_4 & c\theta_4 & \frac{3}{4}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1d})$$

$$\begin{aligned} {}^4T = & \begin{bmatrix} 1 & 0 & 0 & \frac{4}{5}FD_X \\ 0 & c\theta_5 & -s\theta_5 & \frac{4}{5}FD_Y \\ 0 & s\theta_5 & c\theta_5 & \frac{4}{5}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1e}) \end{aligned}$$

$$\begin{aligned} {}^5T = & \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & \frac{5}{6}FD_X \\ s\theta_6 & c\theta_6 & 0 & \frac{5}{6}FD_Y \\ 0 & 0 & 1 & \frac{5}{6}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1f}) \end{aligned}$$

$$\begin{aligned} {}^6T = & \begin{bmatrix} c\theta_7 & -s\theta_7 & 0 & \frac{6}{7}FD_X \\ s\theta_7 & c\theta_7 & 0 & \frac{6}{7}FD_Y \\ 0 & 0 & 1 & \frac{6}{7}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1g}) \end{aligned}$$

$$\begin{aligned} {}_8^7T = & \begin{bmatrix} c\theta_8 & 0 & s\theta_8 & {}_8^7FD_X \\ 0 & 1 & 0 & {}_8^7FD_Y \\ -s\theta_8 & 0 & c\theta_8 & {}_8^7FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1h}) \end{aligned}$$

$$\begin{aligned} {}_9^8T = & \begin{bmatrix} 1 & 0 & 0 & {}_9^8FD_X \\ 0 & c\theta_9 & -s\theta_9 & {}_9^8FD_Y \\ 0 & s\theta_9 & c\theta_9 & {}_9^8FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1i}) \end{aligned}$$

$$\begin{aligned} {}_{10}^9T = & \begin{bmatrix} 1 & 0 & 0 & {}_{10}^9FD_X \\ 0 & c\theta_{10} & -s\theta_{10} & {}_{10}^9FD_Y \\ 0 & s\theta_{10} & c\theta_{10} & {}_{10}^9FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1j}) \end{aligned}$$

$${}^{10}_{11}T = \begin{bmatrix} 1 & 0 & 0 & {}^{10}_{11}FD_X \\ 0 & c\theta_{11} & -s\theta_{11} & {}^{10}_{11}FD_Y \\ 0 & s\theta_{11} & c\theta_{11} & {}^{10}_{11}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1k})$$

$${}^{11}_{12}T = \begin{bmatrix} c\theta_{12} & -s\theta_{12} & 0 & {}^{11}_{12}FD_X \\ s\theta_{12} & c\theta_{12} & 0 & {}^{11}_{12}FD_Y \\ 0 & 0 & 1 & {}^{11}_{12}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1l})$$

$${}^0_{13}T = \begin{bmatrix} c\theta_{13} & -s\theta_{13} & 0 & {}^0_{13}FD_X \\ s\theta_{13} & c\theta_{13} & 0 & {}^0_{13}FD_Y \\ 0 & 0 & 1 & {}^0_{13}FD_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.1m})$$

Appendix C

Additional Test Results for the Forward Kinematics of the Biped

Additional validation test results for the swing foot position with respect to the stance foot is given below:

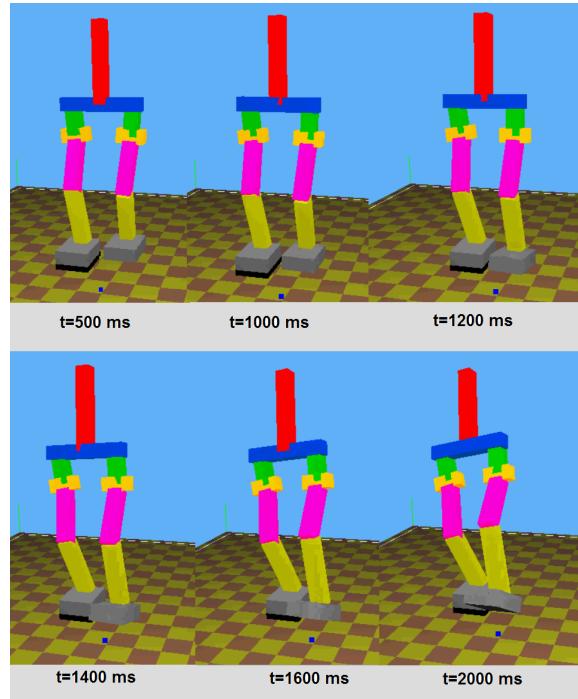


Figure C.1: In order to further validate the swing foot position with respect to stance foot in the X direction, the swing foot is given a reference motion in the lateral plane (SSL).

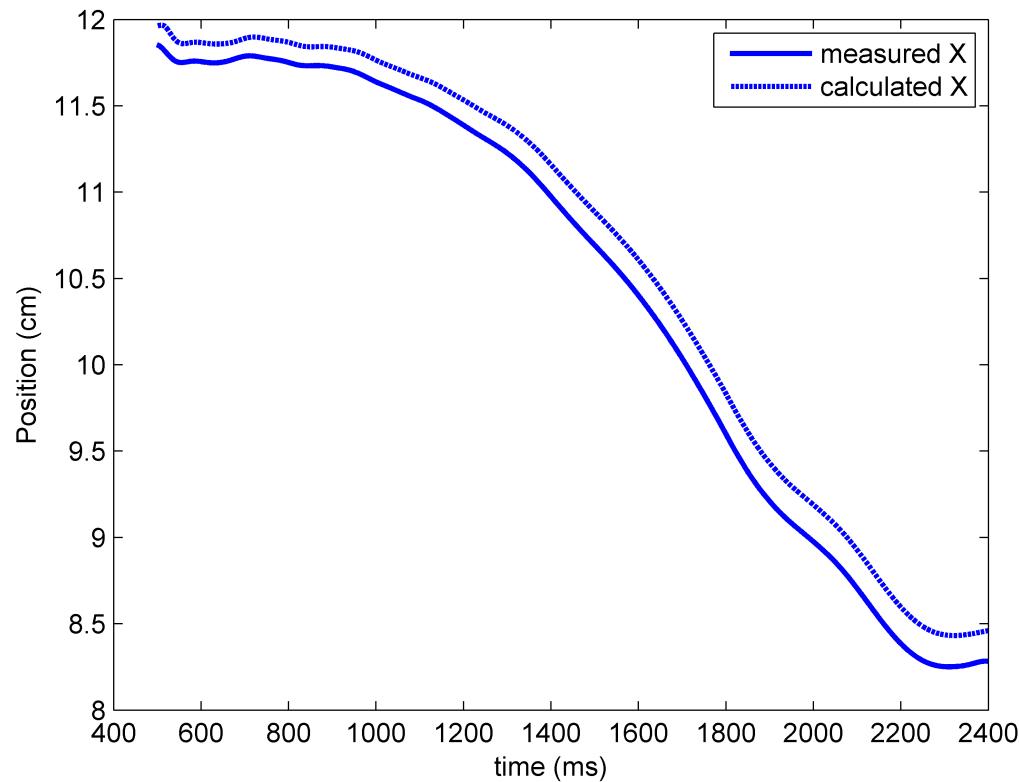


Figure C.2: The additional validation test results for the swing foot X position with respect to the stance foot for SSL.

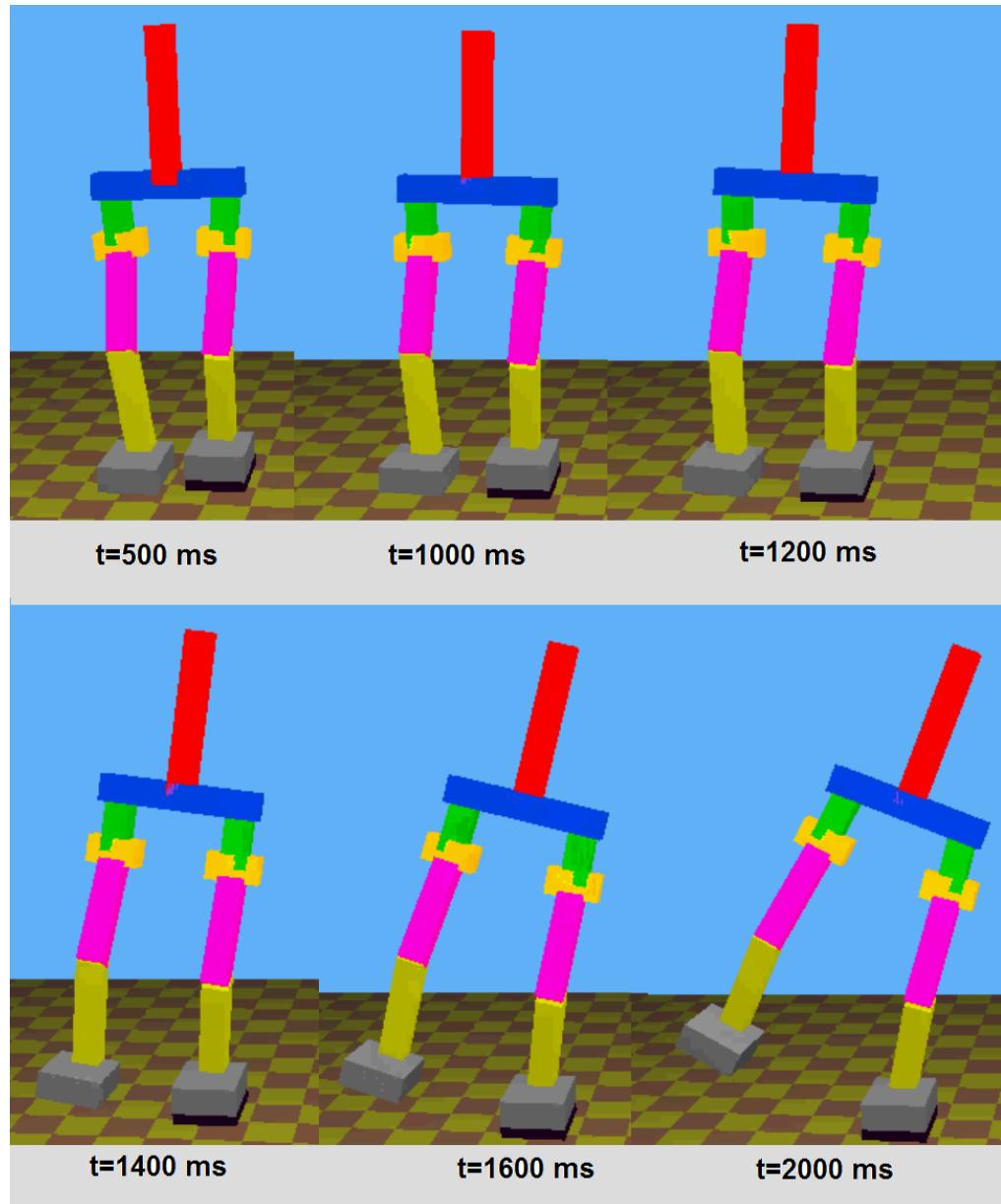


Figure C.3: In order to further validate the swing foot position with respect to stance foot in the X direction, the swing foot is given a reference motion in the lateral plane (SSR).

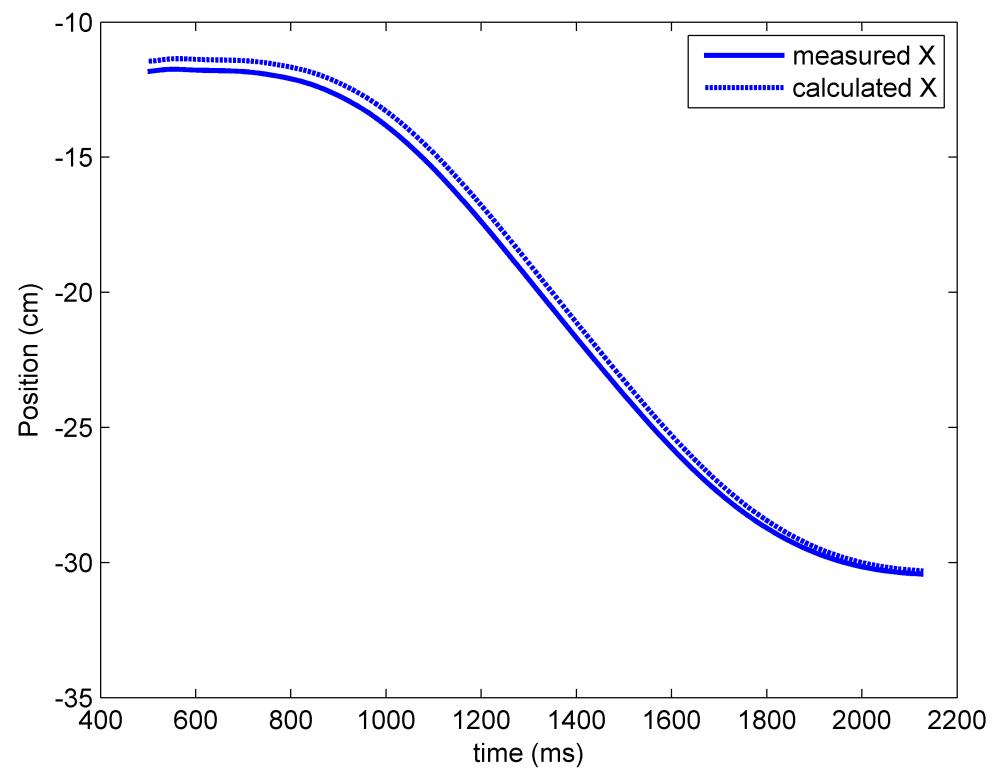


Figure C.4: The additional validation test results for the swing foot X position with respect to the stance foot for SSR.

The additional test results for the pelvis orientation with respect to the stance foot is presented in the following plots:

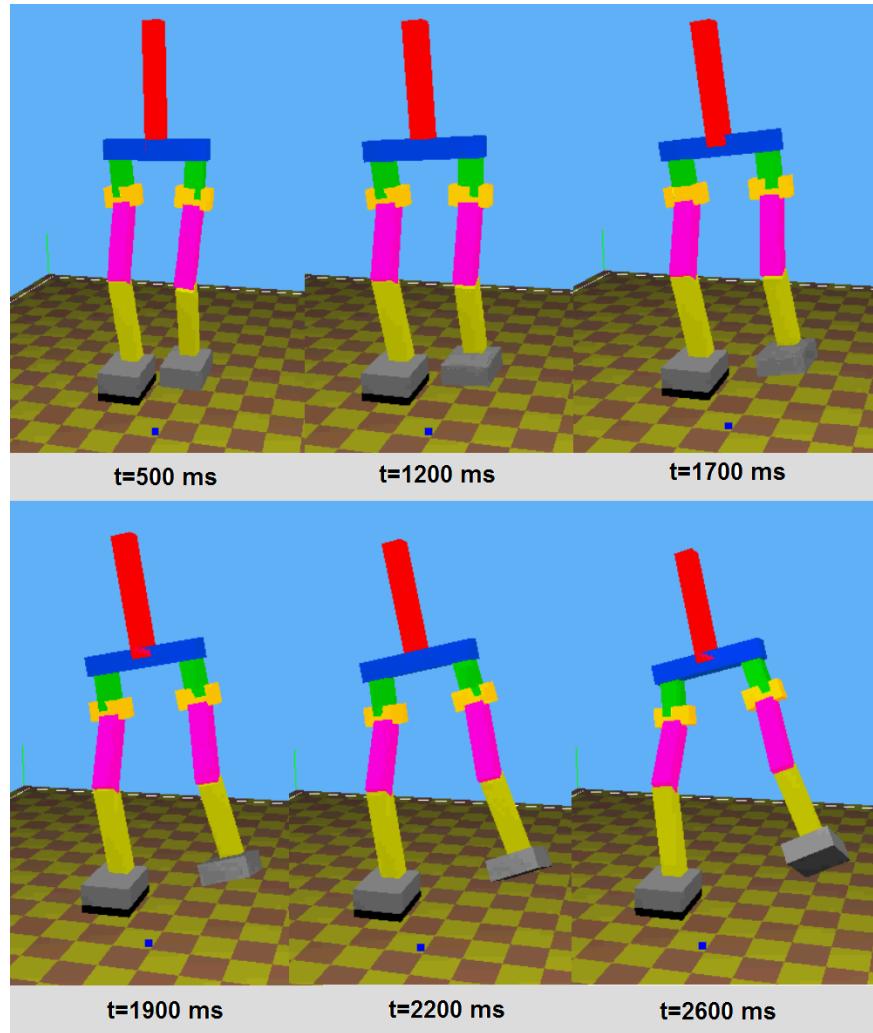
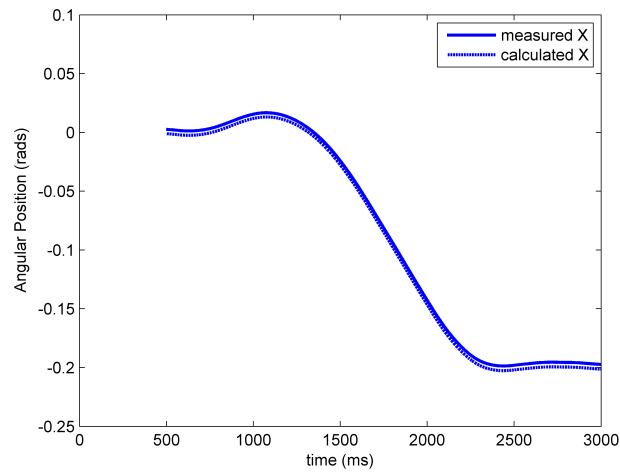
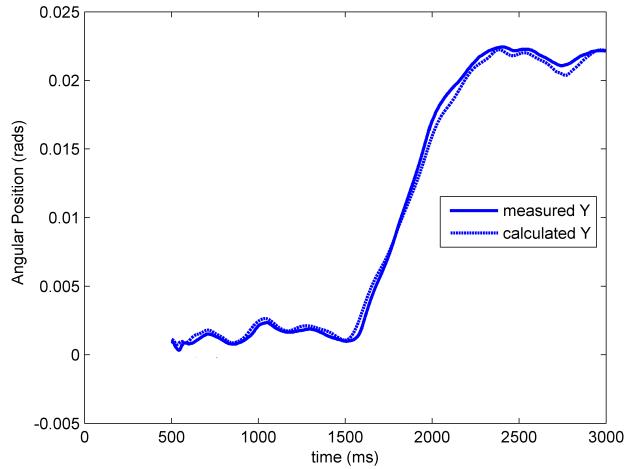


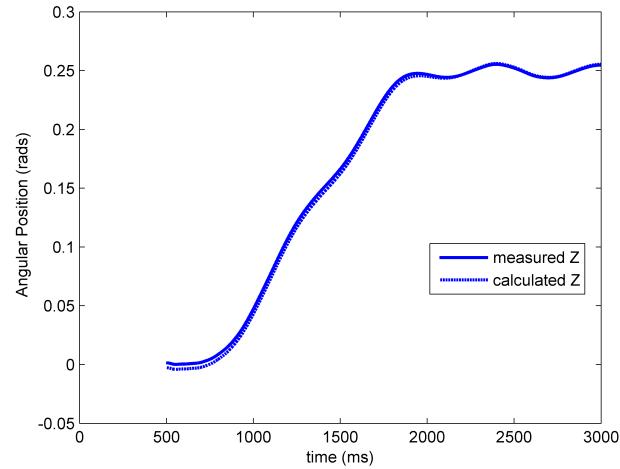
Figure C.5: The reference motion for the additional pelvis orientation test (SSL).



(a) Pelvis orientation about the X axis.



(b) Pelvis orientation about the Y axis.



(c) Pelvis orientation about the Z axis.

Figure C.6: The additional validation test results for the pelvis orientation in SSL.

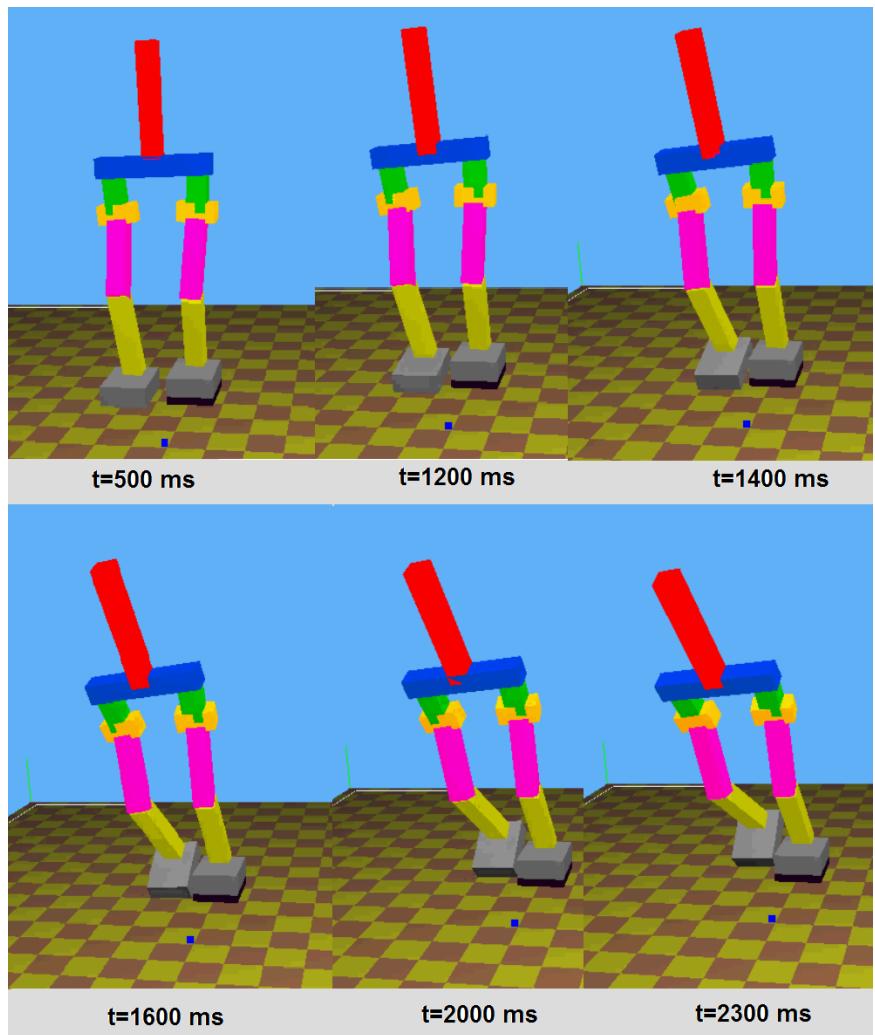
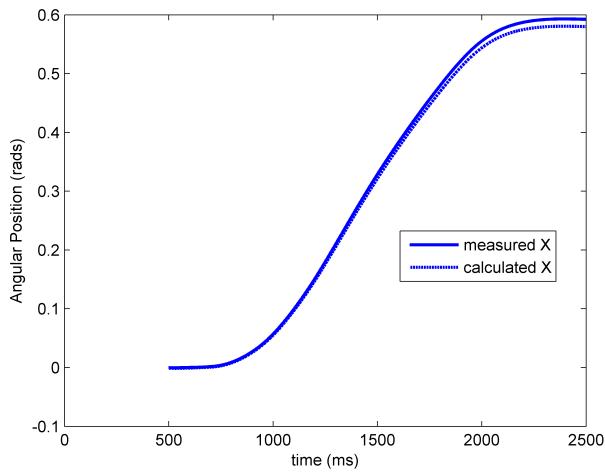
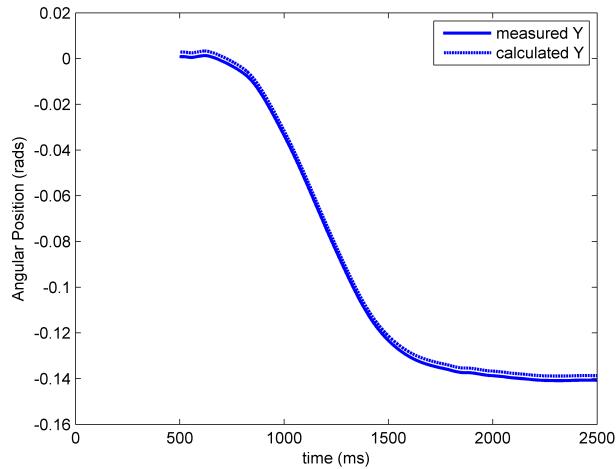


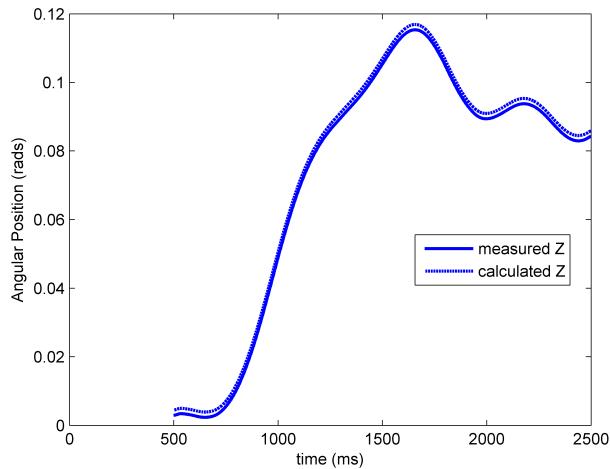
Figure C.7: The reference motion for the additional pelvis orientation test (SSR).



(a) Pelvis orientation about the X axis.



(b) Pelvis orientation about the Y axis.



(c) Pelvis orientation about the Z axis.

Figure C.8: The additional validation test results for the pelvis orientation in SSR.

Finally, the additional test results for the swing foot orientation with respect to the stance foot is presented below:

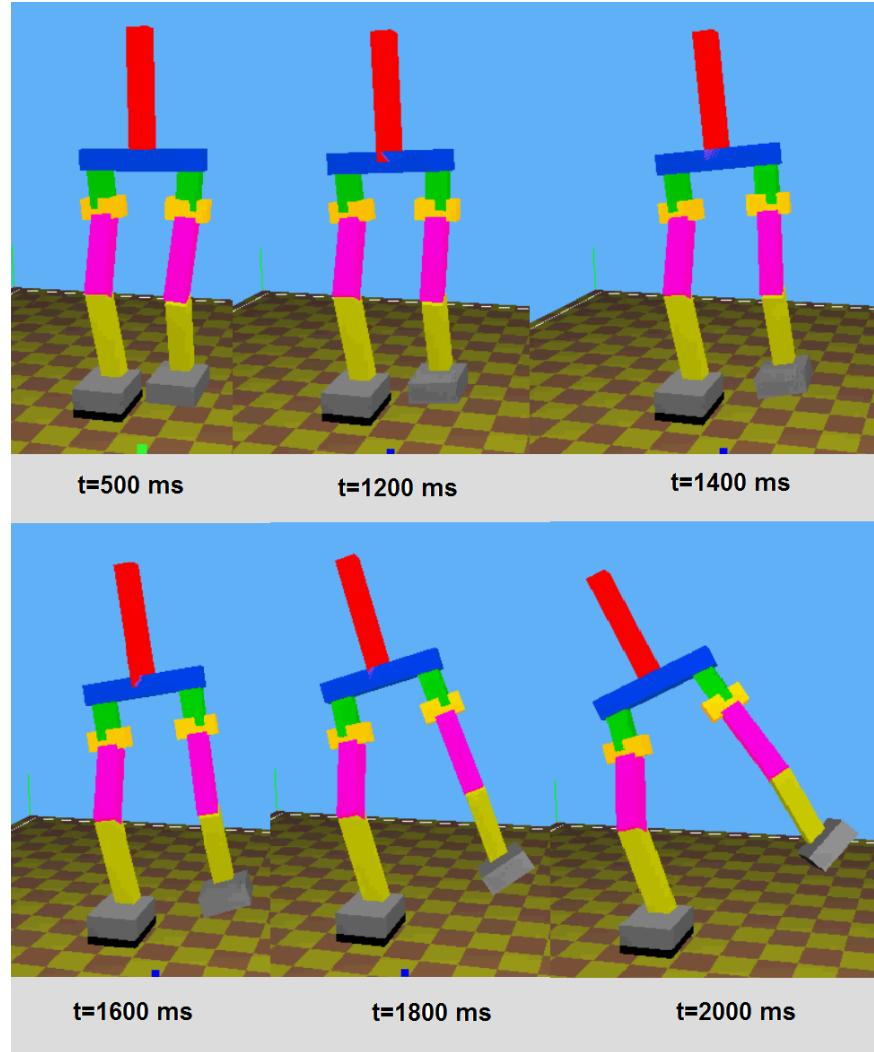
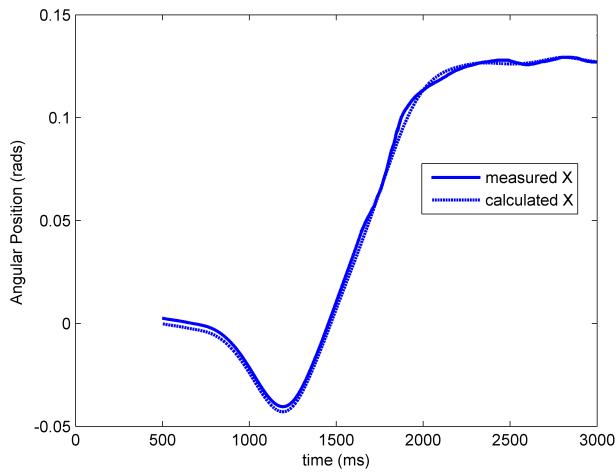
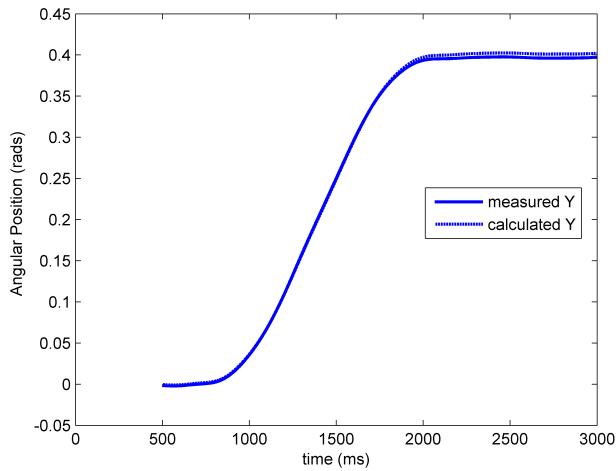


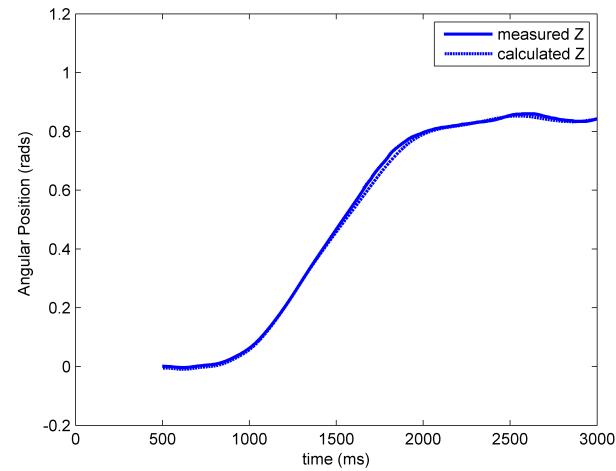
Figure C.9: The reference motion for the additional swing foot orientation test (SSL).



(a) Swing foot orientation about the X axis.



(b) Swing foot orientation about the Y axis.



(c) Swing foot orientation about the Z axis.

Figure C.10: The additional validation test results for the swing foot orientation in SSL.

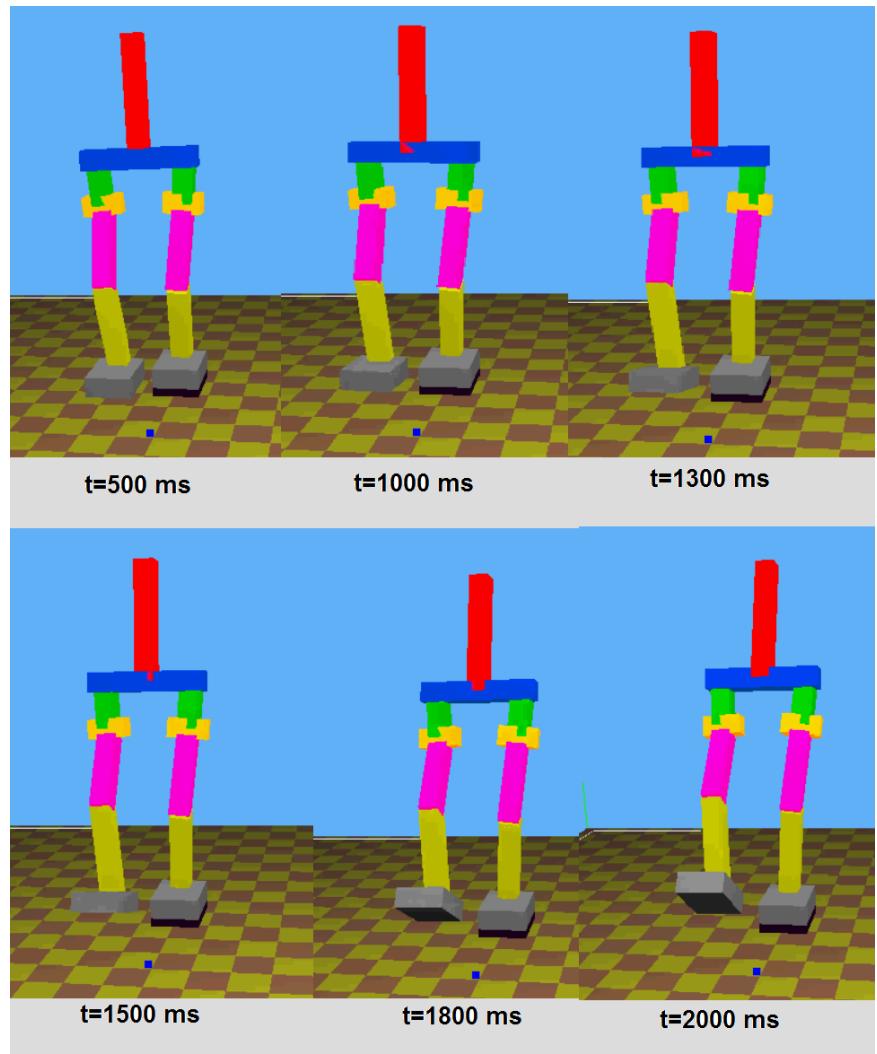
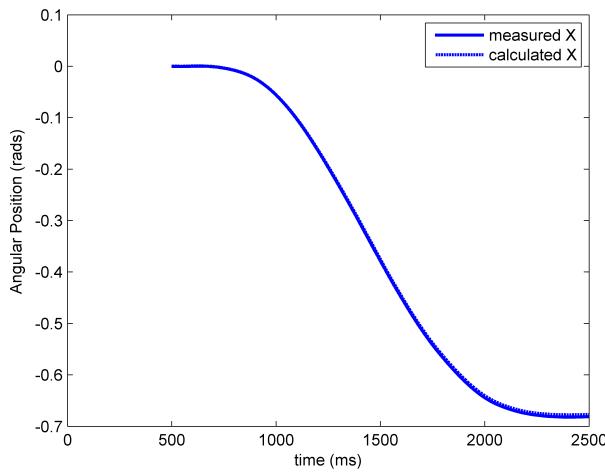
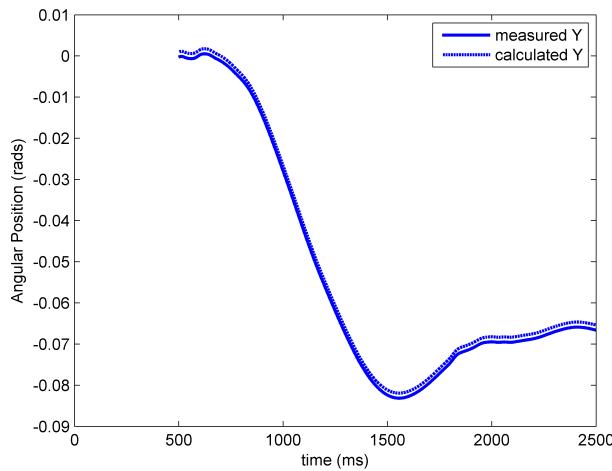


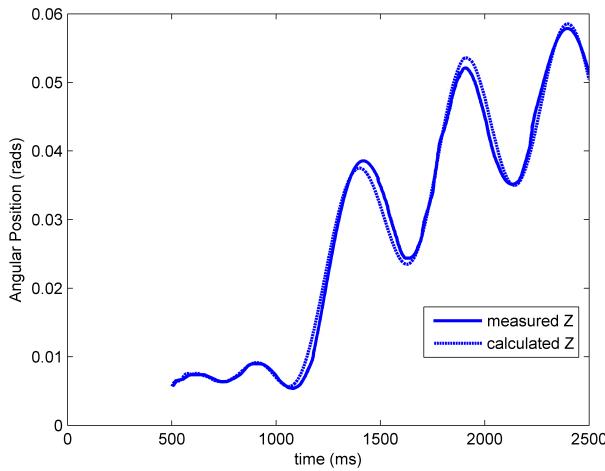
Figure C.11: The reference motion for the additional swing foot orientation test (SSR).



(a) Swing foot orientation about the X axis.



(b) Swing foot orientation about the Y axis.



(c) Swing foot orientation about the Z axis.

Figure C.12: The additional validation test results for the swing foot orientation in SSR.

Appendix D

The Derivation of the COG Jacobian

The COG Jacobian J_G defines the linear relationship between the COG velocity $\vec{\dot{X}}_G$ (a vector in the Cartesian space) and the joint velocities $\vec{\dot{\theta}}$ (a vector in the joint space). The COG Jacobian is formulated as a means of calculating the COG velocity using measured joint velocities. In addition, the inverse of the COG Jacobian is used to resolve a high level reference COG trajectory to reference joint trajectories. Using this method, the COG is manipulated by the joints similar to an end-effector in a robot manipulator. In [52], Kagami et al. proposed a numerical method for calculating the COG Jacobian; however, this method is too complex and computationally expensive for real-time controllers. Sugihara and Nakamura [55] proposed a simpler and faster numerical approach that is applicable to bipeds with an arbitrary number of joints and links. For this reason, it is adopted by the ZMP-based controller and the ZRAM-based controller in this thesis.

As stated in Section 3.4.1, the COG velocity is defined with respect to two different reference frames: the world frame (Frame W) for the simulated biped ${}^W\vec{\dot{X}}_G$ and the stance foot frame (fixed frame) for the actual biped ${}^F\vec{\dot{X}}_G$. However, since velocity is a *free* vector, it can be equivalently described by two reference frames with the same orientation but different origin locations. Therefore, provided that

there is no relative velocity between the world frame and the fixed frame (ie: the stance foot does not rotate throughout the walking cycle), ${}^W\vec{X}_G$ is equivalent to ${}^F\vec{X}_G$.

Similarly, WJ_G and FJ_G can be used interchangeably as long as the stance foot is fixed with respect to the ground. However, in the derivation of the COG Jacobian, most of the linear and angular velocities are calculated from the kinematics of the biped and are defined with respect to the fixed frame (Frame F).

The COG Jacobian with respect to Frame 0

The derivation of the COG Jacobian begins with the calculation of the COG velocity with respect to the pelvis frame (Frame 0) from the individual link masses and link COG velocities:

$${}^0\vec{P}_G = \frac{\sum_{i=1}^p m_i {}^0\vec{r}_i}{\sum_{i=1}^p m_i}, \quad (\text{D.1})$$

where ${}^0\vec{P}_G$ is the COG velocity with respect to Frame 0, m_i is the mass of link i , ${}^0\vec{r}_i$ is the COG velocity of link i with respect to Frame 0, and p and n are the number the number of links and joints in the biped respectively.

Each ${}^0\vec{r}_i$ can be expressed in terms of a link COG Jacobian matrix ${}^0J_{Gi}$, and the joint velocity vector $\vec{\theta}$:

$${}^0\vec{P}_G = \frac{\sum_{i=1}^p m_i {}^0J_{Gi} \vec{\theta}}{\sum_{i=1}^p m_i}, \quad (\text{D.2})$$

The link COG Jacobian matrices are derived from the partial derivatives of the link COG positions 0r_i with respect to the joint variables $\vec{\theta}_i$ in Equation D.3. Note that 0r_i are calculated from the forward kinematics of the biped.

$${}^0J_{Gi} = \frac{\partial {}^0r_i}{\partial \theta_{1..n}}, \quad (\text{D.3})$$

${}^0\vec{P}_G$ can be also expressed in terms of the COG Jacobian matrix with respect to Frame 0 (0J_G) and the joint velocity vector:

$${}^0\vec{P}_G = {}^0J_G \vec{\theta}. \quad (\text{D.4})$$

By comparing Equations D.2 and D.4, we obtain the Equation for 0J_G :

$${}^0J_G = \frac{\sum_{i=1}^p m_i {}^0J_{Gi}}{\sum_{i=1}^p m_i}. \quad (\text{D.5})$$

Relative Velocities between Frame F, Frame 0, and Frame W

The derivation of the COG Jacobian involves calculating the relative linear/angular velocities between Frame 0, Frame W, and Frame F and expressing each of these velocities as a product of a Jacobian matrix and the joint velocity vector.

The linear velocity of the fixed frame origin with respect to Frame 0 (${}^0\vec{P}_F$) is expressed by the following equation:

$${}^0\vec{P}_F = {}^0J_F \vec{\theta}, \quad (\text{D.6})$$

where the Jacobian matrix 0J_F is obtained from the partial derivatives of the fixed frame origin position ${}^0\vec{P}_F$ with respect to the joint variables $\vec{\theta}_i$. Note that ${}^0\vec{P}_F$ is calculated from the kinematics of the *single leg chain* (Section 3.4) with Frame 0 as the base frame and Frame F as the end-effector frame.

Similarly, the angular velocity of Frame 0 with respect to Frame W (${}^W\omega_0$) is

expressed by the following equation:

$${}^W\omega_0 = {}^WJ_{\omega_0} \vec{\dot{\theta}}, \quad (\text{D.7})$$

Although ${}^WJ_{\omega_0}$ defines the linear mapping between the joint velocities and the angular velocity of Frame 0 with respect to Frame W (${}^W\omega_0$), it cannot be calculated from the forward kinematics of the biped. As an alternative, we can express ${}^W\omega_0$ by the angular velocity of the fixed frame:

$${}^0\omega_F = -{}^W\omega_0 \quad (\text{D.8})$$

Note that the above equation is only valid if Frame F is fixed with respect to Frame W and ${}^0\omega_F$ is equivalent to the angular velocity of the world frame with respect to Frame 0 (${}^0\omega_F = {}^0\omega_W = -{}^W\omega_0$).

${}^0\omega_F$ is expressed by the following equation:

$${}^0\omega_F = {}^0J_{\omega_F} \vec{\dot{\theta}} \quad (\text{D.9})$$

${}^0\omega_F$ can be calculated from the kinematics of the *stance leg chain* by "propagating" the angular velocity from one link to another. The angular velocity of link i is equal to the angular velocity of link $i - 1$ (${}^{i-1}\omega_{i-1}$) plus an additional component that is contributed by the rotation of joint i :

$${}^{i-1}\omega_i = {}^{i-1}\omega_{i-1} + {}^{i-1}_iR |\dot{\theta}_i| \hat{Z}_i, \quad (\text{D.10})$$

where $|\dot{\theta}_i|$ is the magnitude of the angular velocity of joint i and \hat{Z}_i is a 3×1 vector that defines the axis of rotation of joint i . From Equation D.10, we can express ${}^0\omega_F$ as a function of the joint velocities (note that $|\dot{\theta}_i| \hat{Z}_i = \vec{\dot{\theta}}_i$). By isolating $\vec{\dot{\theta}}_i$ in

Equation D.10, we can obtain ${}^0J_{\omega F}$.

Substituting Equation D.9 into Equation D.8 and rearranging, we can express ${}^W\omega_0$ in terms of the fixed frame rotation Jacobian ${}^0J_{\omega F}$

$${}^W\omega_0 = -{}^0J_{\omega F} \vec{\theta} \quad (\text{D.11})$$

The Velocity of Frame 0 with respect to Frame W

The Velocity of Frame 0 with respect to Frame W can be calculated by first formulating the equation for the relative velocity between Frame W and Frame F (${}^W\vec{P}_F$). As shown in Figure Figure D.1, ${}^W\vec{P}_F$ is contributed by three components:

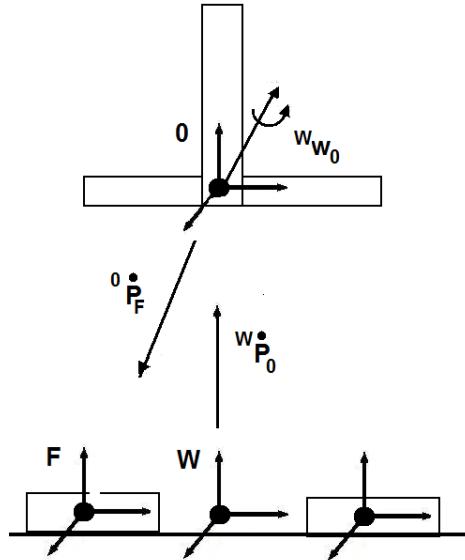


Figure D.1: The velocity of the fixed frame with respect to the world frame is contributed by the linear velocities ${}^W\vec{P}_0$ and \vec{P}_F , and the angular velocity ${}^W\omega_0$.

1. The linear velocity of Frame 0 with respect to Frame W (${}^W\vec{P}_0$).
2. The angular velocity of Frame 0 with respect to Frame W . Note that the resulting linear velocity is given by a vector cross product.

3. The linear velocity of Frame F with respect to Frame 0 (${}^0\vec{P}_F$).

${}^W\vec{P}_F$ is calculated by the following equation:

$${}^W\vec{P}_F = {}^W\vec{P}_0 - \left({}_0^W R {}^0\vec{P}_F \right) \times {}^W\omega_0 + {}_0^W R {}^0\vec{P}_F, \quad (\text{D.12})$$

Since the stance foot (Frame F) is assumed to be fixed with respect to Frame W throughout the walking cycle, ${}^W\vec{P}_F = 0$ and Equation D.12 becomes:

$$0 = {}^W\vec{P}_0 - \left({}_0^W R {}^0\vec{P}_F \right) \times {}^W\omega_0 + {}_0^W R {}^0\vec{P}_F \quad (\text{D.13})$$

Solving for ${}^W\vec{P}_0$ in Equation D.13, we get:

$${}^W\vec{P}_0 = \left({}_0^W R {}^0\vec{P}_F \right) \times {}^W\omega_0 - {}_0^W R {}^0\vec{P}_F \quad (\text{D.14})$$

The linear and angular velocities in Equation D.14 are expressed in terms of Jacobian matrices and the joint velocity vector by substituting D.9 and D.11 into D.14:

$${}^W\vec{P}_0 = {}_0^W R \left[-{}^0\vec{P}_F \times \left({}^W J_{\omega F} \vec{\theta} \right) - {}^0 J_F \vec{\dot{\theta}} \right] \quad (\text{D.15})$$

The COG Velocity with respect to Frame W

Finally, the COG velocity with respect to Frame W (${}^W\vec{P}_G$) is contributed by the following linear/angular velocities (Figure D.2):

1. The linear velocity of Frame 0 with respect to the world frame ${}^W\vec{P}_0$ (given by Equation D.15).
2. The angular velocity of Frame 0 with respect to Frame W .

3. The linear velocity of the COG with respect to Frame 0 (${}^0\vec{P}_G$)

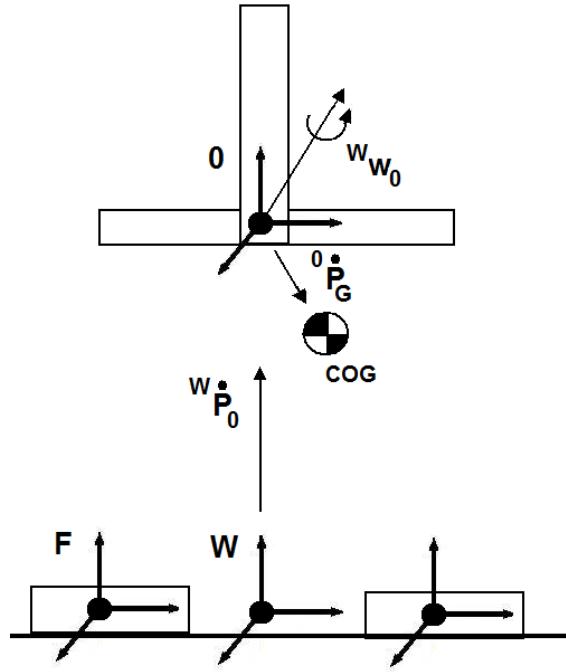


Figure D.2: The COG velocity with respect to the world frame is contributed by the linear velocities $w \vec{P}_0$ and ${}^0\vec{P}_G$, and the angular velocity $w \omega_0$.

$w \vec{P}_G$ is calculated by the following equation:

$$w \vec{P}_G = w \vec{P}_0 - ({}^0R w \vec{P}_G) \times w \omega_0 + {}^0R w \vec{P}_G \quad (\text{D.16})$$

Substituting Equation D.5 and Equation D.11 into Equation D.16 and expressing the linear and angular velocities in terms of Jacobian matrices and joint velocities, we get:

$$w \vec{P}_G = w \vec{P}_0 + {}^0R \left[{}^0\vec{P}_G \times \left({}^0J_{\omega F} \vec{\theta} \right) + {}^0J_G \vec{\theta} \right] \quad (\text{D.17})$$

Substituting $w \vec{P}_0$ (given in Equation D.15) into D.17, and rearranging:

$${}^W \vec{P}_G = {}^W_0 R \left[{}^0 J_G - {}^0 J_F + \left({}^0 \vec{P}_G - {}^0 \vec{P}_F \right)^\times {}^0 J_{\omega F} \right] \vec{\theta} \quad (\text{D.18})$$

Note that the cross product operation in Equation D.18 is expressed as a product of a skew symmetric matrix p^\times and a vector:

$$p^\times = \begin{bmatrix} 0 & -p_Z & p_Y \\ p_Z & 0 & -p_X \\ -p_Y & p_X & 0 \end{bmatrix} \quad (\text{D.19})$$

Finally, since ${}^W \vec{P}_G = J_G \vec{\theta}$, the COG Jacobian matrix J_G can be obtained from Equation D.18:

$${}^W \vec{J}_G = {}^W_0 R \left[{}^0 J_G - {}^0 J_F + \left({}^0 \vec{P}_G - {}^0 \vec{P}_F \right)^\times {}^0 J_{\omega F} \right] \quad (\text{D.20})$$

Two final important notes on the COG Jacobian:

1. The COG Jacobian matrix in Equation D.20 is expressed with respect to the world frame. However, in the actual robot, we cannot measure ${}^W \vec{P}_G$ and ${}^W_0 R$ directly. Therefore, the COG Jacobian and COG velocity is calculated with respect to Frame F by replacing ${}^W_0 R$ with ${}^W_F R$.
2. The fixed frame alternates between the left stance foot (Frame 6) and the right stance foot (Frame 12) between locomotion states. The COG Jacobian matrix that is derived using Frame 6 as the fixed frame is denoted as JGL and is used in DBL, DBL_Safe, and SSL. Similarly, the COG Jacobian matrix with Frame 12 as the fixed frame is denoted as JGR and is used in DBR, DBR_Safe, and

SSR.

Appendix E

The Differential Kinematics of the Kinematic Chains

The differential kinematics of the kinematic chains is used extensively in the forward and inverse kinematics of ABL-BI. This section presents the derivation of the Jacobian matrices that relate the high level tasks of each kinematic chain with the joint velocities. The pseudoinverse of these Jacobian matrices is used in conjunction with a redundancy resolution framework to facilitate kinematic postural control of ABL-BI. Note that the Jacobian matrices in this section is derived with the aid of *Maple* (version 12.0), a symbolic mathematical computation software.

There are two high level tasks in **DB**: 1) COG position tracking, and 2) Pelvis orientation tracking. The biped has a closed-loop kinematic configuration in DB; however, it can be regarded as two separate chains that is shown in Figure 3.15. Although this kinematic setup is used to delegate tasks to the kinematic chains (Section 3.3.1), it can also be used to produce cooperative motions between the two chains. Let's clarify this with an example. As shown in Figure 3.15, the pelvis frame (Frame 0) is the end-effector of both the *Left Leg Chain DBL/DBR* and the *Right Leg Chain DBL/DBR*. By setting the same reference orientation ${}^{foot}_0\Omega$ for the common

end-effector Frame 0 with respect to the foot frames (Frame 6 and 12), the stance leg chains will coordinate their joint motions to track the reference orientation; at the same time, the motion of one chain will not be in conflict with the other (ie: the stance legs will not move in opposite directions and generate internal forces that can pull the biped apart). In other words, the stance leg motions will not violate the kinematic constraint that is imposed by the rigid pelvis link between the two stance leg chains. This example is similar to two cooperative manipulators moving a rigid object at a desirable trajectory.

The COG position is treated as a common end-effector position of the two stance leg chains in Figure 3.3.2. In DBL, the COG velocity $\vec{\dot{X}G}_{DBL}$ (a 3×1 vector) is related to the joint velocity of the *Left Leg Chain DBL* by the following equation:

$$\vec{\dot{X}G}_{DBL} = JGlc_{DBL} \vec{\dot{\theta}}_{1..6,13} \quad (\text{E.1})$$

The COG Jacobian $JGlc_{DBL}$ is a 3×7 matrix that is constructed from the elements of the full COG Jacobian matrix (with the left foot as the fixed frame) JGL that is derived in Appendix D. $JGlc_{DBL}$ is given by the following equation:

$$JGlc_{DBL} = \begin{bmatrix} JGL_{1,1} & \cdots & JGL_{1,6} & JGL_{1,13} \\ JGL_{2,1} & \cdots & JGL_{2,6} & JGL_{2,13} \\ JGL_{3,1} & \cdots & JGL_{3,6} & JGL_{3,13} \end{bmatrix}_{3 \times 7}, \quad (\text{E.2})$$

Note that $JGlc_{DBL}$ only include elements that correspond to the joints in the Left Leg Chain DBL so that $\vec{\dot{X}G}_{DBL}$ will only be manipulated by the Left Leg Chain DBL.

Similarly, $\vec{\dot{X}}G_{DBL}$ is related to the joint velocity of the Right Leg Chain DBL by the following equation:

$$\vec{\dot{X}}G_{DBL} = JGrc_{DBL} \vec{\dot{\theta}}_{7..12} \quad (\text{E.3})$$

$JGrc_{DBL}$ corresponds to the COG Jacobian matrix that involves the joints in the Right Leg Chain DBL only:

$$JGrc_{DBL} = \begin{bmatrix} 0 & \dots & 0 & JGL_{1,7} & \dots & JGL_{1,12} \\ 0 & \dots & 0 & JGL_{2,7} & \dots & JGL_{2,12} \\ 0 & \dots & 0 & JGL_{3,7} & \dots & JGL_{3,12} \end{bmatrix}_{3 \times 6}, \quad (\text{E.4})$$

In order for the left leg chain and the right leg chain to generate cooperative motions to track the COG position, a common reference COG velocity ${}^{ref}\vec{\dot{X}}_G$ is assigned to the task velocity vectors $\vec{\dot{X}}G_{DBL}$ in Equations E.1 and E.3. Concatenating the task velocity vectors, we get:

$$\vec{\dot{X}}G_{DB} = \begin{bmatrix} {}^{ref}\vec{\dot{X}}_G \\ {}^{ref}\vec{\dot{X}}_G \end{bmatrix}_{6 \times 1} \quad (\text{E.5})$$

Combining Equation E.1 and Equation E.3, we get the following expression:

$$\vec{\dot{X}}G_{DB} = JG_{DB} \vec{\dot{\theta}}_{1..13}, \quad (\text{E.6})$$

where JG_{DBL} is obtained by combining the Jacobian matrices that correspond to

the Left Leg Chain $JGlc_{DBL}$ and Right Leg Chain $JGrc_{DBL}$:

$$JG_{DBL} = \begin{bmatrix} JGlc_{1,1} & \cdots & JGlc_{1,6} & 0 & \cdots & 0 & JGlc_{1,13} \\ JGlc_{2,1} & \cdots & JGlc_{2,6} & 0 & \cdots & 0 & JGlc_{2,13} \\ JGlc_{3,1} & \cdots & JGlc_{3,6} & 0 & \cdots & 0 & JGlc_{3,13} \\ 0 & \cdots & 0 & JGrc_{1,7} & \cdots & JGrc_{1,12} & 0 \\ 0 & \cdots & 0 & JGrc_{2,7} & \cdots & JGrc_{2,12} & 0 \\ 0 & \cdots & 0 & JGrc_{3,7} & \cdots & JGrc_{3,12} & 0 \end{bmatrix}_{6 \times 13}, \quad (\text{E.7})$$

Inverting JG_{DBL} , we obtain an equation that relates the reference COG velocities to the joint velocities:

$$\vec{\dot{\theta}} = JG_{DBL}^\lambda \cdot \vec{\dot{X}G_{DB}} \quad (\text{E.8})$$

Similarly, in DBR, the COG velocity is related to the joints in the Left Leg Chain DBR and Right Leg Chain DBR by the Jacobian matrices $JGlc_{DBR}$ and $JGrc_{DBR}$ respectively. The Jacobian matrices are given by the following equations:

$$JGlc_{DBR} = \begin{bmatrix} JGR_{1,1} & \cdots & JGR_{1,6} \\ JGR_{2,1} & \cdots & JGR_{2,6} \\ JGR_{3,1} & \cdots & JGR_{3,6} \end{bmatrix}_{3 \times 6}, \quad (\text{E.9})$$

$$JGr_{cDBR} = \begin{bmatrix} 0 & \cdots & 0 & JGR_{1,7} & \cdots & JGR_{1,13} \\ 0 & \cdots & 0 & JGR_{2,7} & \cdots & JGR_{2,13} \\ 0 & \cdots & 0 & JGR_{3,7} & \cdots & JGR_{3,13} \end{bmatrix}_{3 \times 7}, \quad (\text{E.10})$$

Combining $JGlc_{DBR}$ and JGr_{cDBR} , we obtain a 3×13 COG Jacobian matrix JG_{DBR} that relates $\overrightarrow{\dot{X}G_{DB}}$ to $\dot{\theta}_{1 \dots 13}$:

$$JG_{DBL} = \begin{bmatrix} JGlc_{1,1} & \cdots & JGlc_{1,6} & 0 & \cdots & 0 & 0 \\ JGlc_{2,1} & \cdots & JGlc_{2,6} & 0 & \cdots & 0 & 0 \\ JGlc_{3,1} & \cdots & JGlc_{3,6} & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & JGr_{c1,7} & \cdots & JGr_{c1,12} & JGr_{c1,13} \\ 0 & \cdots & 0 & JGr_{c2,7} & \cdots & JGr_{c2,12} & JGr_{c2,13} \\ 0 & \cdots & 0 & JGr_{c3,7} & \cdots & JGr_{c3,12} & JGr_{c3,13} \end{bmatrix}_{6 \times 13}, \quad (\text{E.11})$$

Inverting JG_{DBR} , we obtain the joint space solution to a given $\overrightarrow{X}G_{DB}$:

$$\overrightarrow{\dot{\theta}} = JG_{DBR}^\lambda \cdot \overrightarrow{X}G_{DB} \quad (\text{E.12})$$

In **DB**, the second high level task for the stance leg chains is pelvis orientation tracking. The angular velocity of the pelvis with respect to the left stance foot (Frame 6) is related to the joint velocities by the following equation:

$${}^6\omega_0 = {}^6J_{\omega 0} \overrightarrow{\dot{\theta}} \quad (\text{E.13})$$

The Jacobian matrix ${}^6J_{\omega 0}$ (a 3×6 matrix) is obtained by computing the rotation axis of each joint \hat{Z}_i ($i = 1 \dots 6$) with respect to Frame 6 (3×1 column vectors):

$${}^6J_{\omega 0} = \begin{bmatrix} {}^6\hat{Z}_1 & \dots & {}^6\hat{Z}_6 \end{bmatrix}_{3 \times 6} \quad (\text{E.14})$$

${}^6\hat{Z}_i$ is obtained by simply premultiplying \hat{Z}_i with the rotation matrix of Frame i with respect to Frame 6:

$${}^6\hat{Z}_i = {}^6R_i \cdot {}^6\hat{Z}_i \quad (\text{E.15})$$

For the right stance leg chain, the angular velocity of the pelvis with respect to Frame 12 is calculated by the following equation:

$${}^{12}\omega_0 = {}^{12}J_{\omega 0} \overrightarrow{\dot{\theta}} \quad (\text{E.16})$$

Again, the Jacobian matrix ${}^{12}J_{\omega 0}$ is calculated from the rotation axis of joints $7 \dots 12$ with respect to Frame 12:

$${}^{12}J_{\omega 0} = \begin{bmatrix} {}^{12}\hat{Z}_7 & \dots & {}^{12}\hat{Z}_{12} \end{bmatrix}_{3 \times 6} \quad (\text{E.17})$$

Similar to the COG position, the pelvis can be cooperatively manipulated by the stance leg chains by setting the same reference pelvis angular velocity for each chain. Combining ${}^{12}J_{\omega 0}$ and ${}^6J_{\omega 0}$ into a task space velocity vector $\vec{\omega}_{DB}$, we get:

$$\vec{\omega}_{DB} = \begin{bmatrix} {}^6\omega_0 \\ {}^{12}\omega_0 \end{bmatrix}_{6 \times 1} \quad (\text{E.18})$$

Combining Equation E.13 and E.16, we get the following kinematic equation:

$$\vec{\omega}_{DB} = J\omega_{DB} \vec{\dot{\theta}} \quad (\text{E.19})$$

In Equation E.19, the Jacobian matrices ${}^6J_{\omega 0}$ and ${}^{12}J_{\omega 0}$ are combined into a 6×13 Jacobian matrix $J\omega_{DB}$:

$$J\omega_{DB} = \begin{bmatrix} {}^6J_{\omega 0}(1,1) & \cdots & {}^6J_{\omega 0}(1,6) & 0 & \cdots & 0 & 0 \\ {}^6J_{\omega 0}(2,1) & \cdots & {}^6J_{\omega 0}(2,6) & 0 & \cdots & 0 & 0 \\ {}^6J_{\omega 0}(3,1) & \cdots & {}^6J_{\omega 0}(3,6) & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & {}^{12}J_{\omega 0}(1,1) & \cdots & {}^{12}J_{\omega 0}(1,6) & 0 \\ 0 & \cdots & 0 & {}^{12}J_{\omega 0}(2,1) & \cdots & {}^{12}J_{\omega 0}(2,6) & 0 \\ 0 & \cdots & 0 & {}^{12}J_{\omega 0}(3,1) & \cdots & {}^{12}J_{\omega 0}(3,6) & 0 \end{bmatrix}_{6 \times 13}, \quad (\text{E.20})$$

Note that the last (13^{th} column) of $J\omega_{DB}$ is zero; therefore, the angular velocity of the pelvis is only controlled by the twelve joints in the stance legs.

Finally, the reference angular pelvis velocity is resolved to joint velocities by the DLS-pseudoinverse of $J\omega_{DB}$:

$$\vec{\theta} = J\omega_{DB}^\lambda \cdot \vec{\omega}_{DB} \quad (\text{E.21})$$

As shown in Figure 3.3.2, the kinematics of the biped is separate into two kinematic chains in **SS**: 1) Stance Leg Chain SSL/SSR, and 2) Swing Leg Chain SSL/SSR. Like DB, the high level tasks of the stance leg chain are COG position tracking and pelvis orientation tracking. The COG Jacobian matrices for SSL and SSR are given by Equations E.2 and E.10 respectively, and the Jacobian matrices for the pelvis angular velocity are given in Equations E.14 and E.17 respectively.

For the swing leg chain, the high level objectives are swing foot position and orientation tracking. In **SSL**, the swing foot velocity \vec{X}_{12}^6 (a 3×1 vector) is related to the joint velocities by the following equation:

$$\vec{X}_{12}^6 = {}_{12}^6 J \vec{\theta} \quad (\text{E.22})$$

The swing foot position Jacobian matrix ${}_{12}^6 J$ (a 3×13 matrix) is obtained from the partial differentiation of the position vector \vec{P}_{12}^6 with respect to the joint positions.

$${}_{12}^6 J = \frac{\partial {}_{12}^6 \vec{P}}{\partial \theta_{1..13}}, \quad (\text{E.23})$$

Neglecting the elements of ${}_{12}^6 J$ that are associated with the joints in the stance leg chain ($\theta_{1..6}$), we get a modified Jacobian matrix:

$$Jv_{SSL} = \begin{bmatrix} {}_{12}^6 J_{1,1} & \cdots & {}_{12}^6 J_{1,6} \\ {}_{12}^6 J_{2,1} & \cdots & {}_{12}^6 J_{2,6} \\ {}_{12}^6 J_{3,1} & \cdots & {}_{12}^6 J_{3,6} \end{bmatrix}_{3 \times 6}, \quad (\text{E.24})$$

The joint space solution that is produced by the pseudoinverse of Jv_{SSL} will only involve joint velocities in the swing foot chain. In other words, the high level task \vec{X}_{12}^6 in Equation E.22 is assigned to the joints in the swing leg chain.

$$\vec{\theta} = Jv_{SSL} \cdot \vec{X}_{12}^6 \quad (\text{E.25})$$

Similarly, in **SSR**, the swing foot position is determined from the following equation:

$${}^{12}_6 \vec{X} = {}^{12}_6 J \vec{\theta} \quad (\text{E.26})$$

The swing foot position Jacobian matrix ${}^{12}_6 J$ is obtained from the partial differentiation of the position vector ${}^{12}_6 \vec{P}$ with respect to the joint positions.

$${}^{12}_6 J = \frac{\partial {}^{12}_6 \vec{P}}{\partial \theta_{1..13}}, \quad (\text{E.27})$$

Like **SSL**, the elements that are associated with the stance foot joints ($\theta_{7..12}$) in ${}^{12}_6 J$ are eliminated to obtain the modified Jacobian matrix Jv_{SSR} :

$$Jv_{SSR} = \begin{bmatrix} {}^{12}_6 J_{1,7} & \dots & {}^{12}_6 J_{1,12} \\ {}^{12}_6 J_{2,7} & \dots & {}^{12}_6 J_{2,12} \\ {}^{12}_6 J_{3,7} & \dots & {}^{12}_6 J_{3,12} \end{bmatrix}_{3 \times 6}, \quad (\text{E.28})$$

Jv_{SSR} is used to assign the high level task to the swing foot joint in the following inverse differential kinematics equation:

$$\vec{\theta} = Jv_{SSR}^\lambda \cdot {}^{12}_6 \vec{X} \quad (\text{E.29})$$

In **SS**, the second high level task of the swing foot chain is the swing foot orientation tracking. The angular velocity of the swing foot with respect to the stance foot frame in **SSL** ${}^{12}_6 \omega$ (a 3×1 vector) is related to the joint velocities by the following equation:

$${}_{12}^6\omega = {}_{12}^6J\omega \vec{\theta} \quad (\text{E.30})$$

The Jacobian matrix ${}_{12}^6J\omega$ is constructed by the axis of rotation vectors ${}^6\hat{Z}_i$, $i = 1 \dots 13$.

$${}_{12}^6J\omega = \begin{bmatrix} {}^6\hat{Z}_1 & \dots & {}^6\hat{Z}_{13} \end{bmatrix} \quad (\text{E.31})$$

As before, the Jacobian matrix ${}_{12}^6J\omega$ is modified by eliminating the elements that are associated with the stance leg joints ($\theta_{1\dots 6}$) and the pseudoinverse of $J\omega_{SSL}$ is used to assign the high level task ${}_{12}^6\omega$ to the swing leg joints (Equation E.33).

$$J\omega_{SSL} = \begin{bmatrix} {}_{12}^6J\omega_{1,1} & \dots & {}_{12}^6J\omega_{1,6} \\ {}_{12}^6J\omega_{2,1} & \dots & {}_{12}^6J\omega_{2,6} \\ {}_{12}^6J\omega_{3,1} & \dots & {}_{12}^6J\omega_{3,6} \end{bmatrix}_{3 \times 6}, \quad (\text{E.32})$$

$$\vec{\theta} = J\omega_{SSL}^\lambda \cdot {}_{12}^6\vec{X} \quad (\text{E.33})$$

In **SSR**, the angular velocity of the swing foot with respect to the stance foot ${}^{12}{}_6\omega$ is related to the joint velocities by the following equation:

$${}^{12}{}_6\omega = {}^{12}{}_6J_\omega \vec{\theta} \quad (\text{E.34})$$

The Jacobian matrix ${}^{12}{}_6J_\omega$ is constructed by the axis of rotation vectors ${}^{12}\hat{Z}_i$,

$i = 1 \dots 13$.

$${}_{\hat{6}}^{12}J_{\omega} = \begin{bmatrix} {}_{\hat{6}}^{12}\hat{Z}_1 & \dots & {}_{\hat{6}}^{12}\hat{Z}_{13} \end{bmatrix} \quad (\text{E.35})$$

The Jacobian matrix ${}_{\hat{6}}^{12}J_{\omega}$ is modified and we obtain $J\omega_{SSR}$ that is only associated with the swing leg joints ($\theta_{1\dots 6}$).

$$J\omega_{SSR} = \begin{bmatrix} {}_{\hat{6}}^{12}J\omega_{1,7} & \dots & {}_{\hat{6}}^{12}J\omega_{1,12} \\ {}_{\hat{6}}^{12}J\omega_{2,7} & \dots & {}_{\hat{6}}^{12}J\omega_{2,12} \\ {}_{\hat{6}}^{12}J\omega_{3,7} & \dots & {}_{\hat{6}}^{12}J\omega_{3,12} \end{bmatrix}_{3 \times 6}, \quad (\text{E.36})$$

The joint space solution for the reference angular velocity of the swing foot is calculated by the following equation:

$$\vec{\dot{\theta}} = {}_{\hat{6}}^{12}J\omega_{SSR}^{\lambda} \cdot {}_{\hat{6}}^{12}\vec{X} \quad (\text{E.37})$$