

Design and Control of a Humanoid Robot

Wee Teck Chew

Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In the subject of

Robotics and Control Systems

Control and Power Research Group

Department of Electrical and Electronic Engineering

Imperial College London

June 2014

Declaration of Originality

I declare that this thesis is my own original research work and that all information derived from the published and unpublished work of others is appropriately referenced in the bibliography.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

Design and control of the bipedal humanoid robot locomotion are challenging areas of research. Accurate models of the kinematics and dynamics of the robot are essential to achieve bipedal locomotion. Bipedal walking can be achieved either with flat-foot or toe-foot walking. Flat-foot walking is more stable but slower, whereas toe-foot walking produces more natural and faster motion. Furthermore in toe-foot walking it is possible to perform stretch knee walking. The mechanical structure of the robot is designed with compact modular parts so that the robot kinematics can be modelled as a multi-point-mass system, and its dynamics are modelled applying the inverted pendulum model and the zero-moment-point concept. The optimality in the gait trajectory is achieved exploiting augmented model predictive control methods taking into consideration the trade-off between walking speed and stability. The robustness and stability of the walking gaits and posture in the presence of internal or external disturbances are enhanced by adopting angular compensation with joint control techniques. The thesis develops a flat-foot optimal walking gait generation method. The effectiveness of the control technique and the passive-toe design is validated by simulation tests with the robot walking on slope, stepping over an obstacle and climbing a stair. The walking gaits are implemented on a mid-size (1.6 meter, 58 kg) bipedal robot. Experiments demonstrate the effectiveness of the new proposed Augmented Model Predictive Control (AMPC) method has improved and produced a smoother gaits tracking trajectory in comparison with existing LQR and preview methods; and at the same time the proposed algorithms are able to reduce noise interference.

Acknowledgements

A deep thank-you to my supervisor, Prof. Alessandro Astolfi, who has played a crucial role in my research work, with his continuous support, advices and guidance throughout the course whilst allowing me the room to work in my own way, and also for his suggestions and inputs into the papers we co-authored.

Special thanks to my co-supervisor, Prof. Xie Ming from NTU Singapore, for his support and advice throughout the course of my study.

Thank you to all the researchers of LOCH group from NTU Singapore, for the good and fruitful time we spent together in the Singapore Armed Forces unit SCME; at the RSAF Open House Road-show; conducting of numerous field trials; and demonstration of the robot to VIPs and visitors.

In addition, I would like to thank our senior research group administrator (C&P group) Michelle Hammond, for her excellence administrative support throughout the course of research.

Thank you to all the members of the Control & Power Research group, especially Pedro, in providing an enjoyable working and research environment for me during my course of work.

I would like to thank Chee Chong Woh for his help and constructive advices to me in writing this thesis.

This PhD study was funded by the Singapore Ministry of Education (MOE) and Temasek Polytechnic, and their financial support is gratefully acknowledged.

My aspiration is that the bipedal humanoid robot will one day help to improve our elderly community in providing elderly-care service and living support. However, there is still much to be done to achieve this dream goal. Humanoid robot will evolve mechanically, electronically and be more intelligent progressively, and my research will not stop here, I will continue to develop and improve TPinokio capabilities, to realize my aspiration.

Contents

Abstract

Acknowledgements

1 Introduction

1.1 Background	14
1.2 Motivation and Objectives	14
1.3 Robot and Definitions	15
1.4 Methodology	16
1.5 Thesis Outline	18
1.6 Research Contributions	20
1.7 Publications	21

2 Literature Review and Background Theory

2.1 General Information	22
2.2 Bipedal Development History by Various Research Institutions	23
2.3 Recent Developments on Bipedal Robots	25
2.4 Robot Mechanical Design	30
2.5 Robot Configuration	32
2.6 Forward and Inverse Kinematics Model	33
2.7 Dynamic Model	37
2.8 Center of Mass, Zero Moment Point and Support Convex Polygon	40
2.9 Walking Gait and Path Planning	43
2.10 Trajectory Generation and Control	45
2.11 Stabilizer	51
2.10 Chapter Summary	52

3 Mechanical and System Architectures

3.1 Mechanical Structure	54
3.2 Hardware Architecture	59

3.3 Software Architecture	61
3.4 Modular Joint Design Concept	62
3.5 Force, Power and Stress Analysis of TPinokio Robot	68
3.6 Chapter Summary	82
4 Kinematics and Dynamics Model	
4.1 Kinematics Model of TPinokio Robot	83
4.2 Dynamic Model of TPinokio Robot	91
4.3 Momentum and Motion	97
4.4 Motion Path and Trajectory Generation	99
4.5 Chapter Summary	107
5 Control and Walking Pattern Generation	
5.1 Augmented Model Predictive Control	108
5.2 Robot with Passive Toe-foot	126
5.3 Feasibility Study of Robot Waling on Different Surface	128
5.4 Chapter Summary	148
6 Stability and Posture Control	
6.1 Stability Margin and Support Polygon	149
6.2 Stabilization Based on Linear Inverted Pendulum	150
6.3 AMPC Stability Analysis	150
6.4 Balancing Control with Sensors Feedback	157
6.5 Chapter Summary	161
7 Conclusion and Continuous Research	
7.1 Research Summary	162
7.2 Continuous Research Activities and Recommendations for Future Works	163

Bibliography

Appendixes

- A** TPinokio Robot Mechanical Design
- B** Theory of Applied Robotics

List of Tables

Table 3.1. Maximum step size.

Table 3.2. Dimension and weight of robot arm and hand.

Table 4.1. TPinokio robot joint link coordinate.

Table 4.2. ZMP trajectory and profile.

List of Figures

- Figure 1.1. TPinokio Bipedal humanoid robot and simulation model.
- Figure 1.2. Definition of robot main coordinate and robot parts.
- Figure 1.3. Feedback and stabilizer control.
- Figure 2.1. Pictures of 19th century robots.
- Figure 2.2. The 1960/70 robots.
- Figure 2.3. Waseda university WL series robots.
- Figure 2.4. Robots designed and built by Japanese researchers.
- Figure 2.5. Japan Robots.
- Figure 2.6. HRP series robots.
- Figure 2.7. KHR series robots.
- Figure 2.8. Robots designed by France researchers.
- Figure 2.9. Robots from various European universities.
- Figure 2.10. Robots designed by USA researchers.
- Figure 2.11. Robots designed by Singapore institutions.
- Figure 2.12. LOCH (NTU) robot with electric revolute-joint design concept.
- Figure 2.13. LOLA robot with Linear actuator design concept.
- Figure 2.14. LUCY [ref] Artificial muscle design concept.
- Figure 2.15. The robot planes of section.
- Figure 2.16. A typical robot walking cycle.
- Figure 2.17. 3D Inverted pendulum model and compass-gait motion.
- Figure 2.18. Location of the robot CoM and ZMP.
- Figure 2.19. Area of the SCP is the convex hull of the red envelop.
- Figure 2.20. A typical bipedal robot support polygon stable region.
- Figure 2.21. A typical biped walking pattern and its CoM path trajectory.
- Figure 2.22. Trajectory of robot locomotion, the red dotted line is the CoM trajectory,
- Figure 2.23. Human walking gait data recording technique.
- Figure 2.24. Typical ZMP trajectories, the red boxes are the robot's foot step.
- Figure 2.25. LQR with observer waveform generation block diagram.
- Figure 2.26. LQR tracking waveforms.
- Figure 2.27. Preview controller block diagram.
- Figure 2.28. Preview Gain G_p .
- Figure 2.29. A typical feedback and stabilizer control block diagram.
- Figure 3.1. TPinokio Targeted robotic market sector.
- Figure 3.2. Mechanical structure of TPinokio.
- Figure 3.3. Advantages of having toe-feet design.
- Figure 3.4. TPinokio robot step size.
- Figure 3.5. Human living environment and space.
- Figure 3.6. General specification of the TPinokio robot.
- Figure 3.7. Movable range of the TPinokio robot.
- Figure 3.8. Overall control of the TPinokio robot.
- Figure 3.9. Hardware System Configurations.
- Figure 3.10. TPinokio GUI with LabVIEW.
- Figure 3.11. Compact modular housing.

Figure 3.12. Compact modular-joint parts.

Figure 3.13. Simple cable connections between each compact module.

Figure 3.14. Foot mechanism with passive toe.

Figure 3.15. Full bridge load cell sensing circuit.

Figure 3.16. Single support phase (SSP) and the ZMP location.

Figure 3.17. Double support phase (DSP) ZMP location.

Figure 3.18. FTS measurement and estimation of ZMP location.

Figure 3.19. Jumping sequences key frames.

Figure 3.20. Jumping height and Velocity

Figure 3.21. Robot body bending parameters.

Figure 3.22. Acceleration and angle of the robot ankles.

Figure 3.23. Phases of landing.

Figure 3.24. CoM location of the robot head .

Figure 3.25. Angular velocity of the pelvis three axis.[H.J. Luinge et al., 2002].

Figure 3.26. Robot maximum yaw and pitch angle of the robot.

Figure 3.27. Dimension and CoM location of arm joints.

Figure 3.28. Angular velocity of a human upper arm [H.J. Luinge et al., 2002].

Figure 3.29. Grasping forces analysis of a robot hand.

Figure 3.30. Gait cycle of walking.

Figure 3.31. (a) Ankle roll to single support standing, (b)Hip roll in to double support.

Figure 3.32. Robot making a 90 degree turning.

Figure 4.1. TPinokio simplified point-mass model.

Figure 4.2. Similar robot with actuator mounted on the joint link.

Figure 4.3 TPinokio D-H model and link frames.

Figure 4.4 TPinokio robot kinematics stick diagram.

Figure 4.5 TPinokio robot joint and kinematic links.

Figure 4.6. Robot initial posture is determined by applying forward kinematics .

Figure 4.7. Individual joint angle position is determined by inverse kinematics.

Figure 4.8. Inverted pendulum model superimposed on to the robot.

Figure 4.9. TPinokio point mass distribution model.

Figure 4.10. Double IPM model

Figure 4.11. Momentum and motion reference coordinate.

Figure 4.12. Interpolation and trajectory pattern simulation.

Figure 4.13. Typical robot walking cycle.

Figure 4.14. Typical robot walking gait.

Figure 4.15. Trajectory profile with cycloid function.

Figure 4.16. Trajectory profile with seven degree polynomial function.

Figure 4.17. ZMP trajectory parameters.

Figure 4.18. Generic ZMP trajectory.

Figure 5.1. Typical AMPC set up.

Figure 5.2. Continuous time poles and zeros plot.

Figure 5.3. Discrete time poles and zeros plot.

Figure 5.4. Block diagram of a discrete time AMPC.

Figure 5.5. P-Z plot of a discrete time LIPM-AMPC system.

Figure 5.6. Augmented MPC model tracking, (a) y-axis, (b) x-axis, (c) xy plane.

Figure 5.7. System jerk is reduced to zero in both (a) x-axis and (b) y-axis direction.

Figure 5.8. Comparison of MPC and Preview control, (a) y-axis, (b) x-axis.

Figure 5.9. Robot simulation model.

Figure 5.10. Robot joint parameters.

Figure 5.11. AMPC with ZPETC controller block diagram.

Figure 5.12. AMPC-ZPETC waveform, (a) x-axis, (b) y-axis, (c) xy-axis.

Figure 5.13. AMPC-PZC control signal, (a) y-axis, (b) x-axis.

Figure 5.14. Piece-wise single step response.

Figure 5.15. AMPC with DMIPM waveform, the green box in (c) represent the robot foot step.

Figure 5.16. TPinokio robot with passive foot-toe design.

Figure 5.17. Modified toe-foot walking gait and its ZMP position in sagittal plane.

Figure 5.18. (a) Flat-feet step length. (b) Toe-feet step length.

Figure 5.19. Toe-off and heel-strike walking trajectory in sagittal plane.

Figure 5.20. Inverted pendulum walking on slope model.

Figure 5.21. Posture of robot walking on slope.

Figure 5.22. Walking on slope joints' trajectory, (a) Left leg, (b) Right leg.

Figure 5.23. Geometrical measurement of robot stepping over obstacle parameters.

Figure 5.24. Foot trajectories for stepping over a rectangular obstacle.

Figure 5.25. TPinokio step over obstacle posture model.

Figure 5.26. Basic sequences for stepping over obstacle with statically balancing mode.

Figure 5.27. Leg joints' trajectories for stepping over obstacle, (a) Left leg, (b) Right leg.

Figure 5.28. Geometrical measurement of robot stepping on and over obstacle parameters.

Figure 5.29. Geometrical parameters of robot climbing stairs.

Figure 5.30. Stairs climbing sequences.

Figure 5.31. Simulation of robot walking upstairs and its joints angle.

Figure 5.32. ZMP-CoM model for walking up and down stairs.

Figure 5.33. The robot geometrical boundary and position for 90° turn.

Figure 5.34. The sequences and robot CoM (pelvis) and foot position for spot-turning.

Figure 5.35. (a) simulation sequences, (b) Left leg joints angle, (c) Right leg joints angle.

Figure 6.1 Controller and stabilizer block diagram.

Figure 6.2. The support convex polygon (SCP) and the ZMP boundary.

Figure 6.3. Control variables response with constraints.

Figure 6.4. AMPC tracking trajectory with constraints.

Figure 6.5. AMPC model tracking trajectory with noisy input.

Figure 6.6. AMPC model system jerk response with noisy waveform.

Figure 6.7. AMPC-PZC tracking waveform simulation with noise.

Figure 6.8. AMPC-PZC control signal simulation with noise.

Figure 6.9. Block diagram of AMPC with observer.

Figure 6.10. Roll angle compensation of swing leg for frontal plane.

Figure 6.11. Sensor feedback compensation on slope surface for sagittal plane.

Figure 6.12. Sensor feedback compensation on slope surface for frontal plane.

Figure 6.13. Spring-mass model of a robot with bending legs.

Figure 6.14. Damping response of the shock absorber with $T=1$ second.

Figure 7.1 Kneeling down sequences.

Figure 7.2. Slippage turning with toe.

Figure 7.3. Modification of CoG and ZMP position.

Figure 7.4. Human running motion and flight phase.

List of Acronyms

2D: Two Dimensional

3D: Three Dimensional

AMPC: Augmented Model Predictive Control

CoM: Center of Mass

DMIPM: Double Mass Inverted Pendulum Model

DoF: Degree of Freedom

DSP: Double Support Phase

GCoM: Ground Projection of the Center of Mass

IPM: Inverted Pendulum Model

LHP: Left Half Plane

LQR: Linear Quadratic Regulator

MP: Minimum Phase

MPC: Model Predictive Control

NMP: Non-Minimum Phase

PID: Proportional Integral Derivative

PZC: Pole Zero Cancellation

RHP: Right Half Plane

RHS: Right Hand Side

SCP : Support Convex Polygon

SISO: Single Input Single Output

SSP: Single Support Phrase

ZMP: Zero Moment Point

Chapter 1

Introduction

1.1 Background

The development of bipedal robots began more than thirty years ago. A stable walking motion in a bipedal humanoid robot requires effective gait balancing and robust posture correction algorithms. Despite the several research efforts, it is still an extremely challenging task to develop and implement intelligent motion algorithms to control a bipedal robot, particularly with regards to adaptability, robustness and stability of the robot. This chapter presents the objectives, the scope of work and the methodologies under taken in this research. It will also give a brief summary of the research as well.

1.2 Motivation and Objectives

It is a challenging task to control and stabilize bipedal robot locomotion because of its constrained dynamic characteristics, especially with its walking gaits trajectories. The motivation and objectives of this research are:

- To design and build a teen-size bipedal humanoid robot with simple kinematics structure, for education and research purpose.
- To develop and implement an effective control algorithm to achieve a stable static and dynamic walking posture.

The scope of this research includes the mechanical design, the electronics interface, the software development and the control algorithm implementation for both static and dynamic walking.

For a robot to walk steadily and maintain good balancing and stability, the projected ground Center of Mass (GCoM) of the bipedal must stay inside the support convex polygon (SCP) of the

supporting stance foot during walking or stay within both feet SCP when the robot is supported by both leg. There are various approaches for stable walking found in literature. This research proposes an alternative approach that applies the Augmented Model Predictive Control (AMPC) algorithm and the Zero-Moment Point (ZMP) concept, together with the Inverted Pendulum Model (IPM) method to generate and synthesize the walking gait, and accompanied with stabilizer and sensory feedback to achieve a stable locomotion. External noise and disturbances are added into the simulation to test its robustness. As depicted in Figure 1.1, the proposed approaches are experimented with the TPinokio bipedal humanoid robot to test its effectiveness.

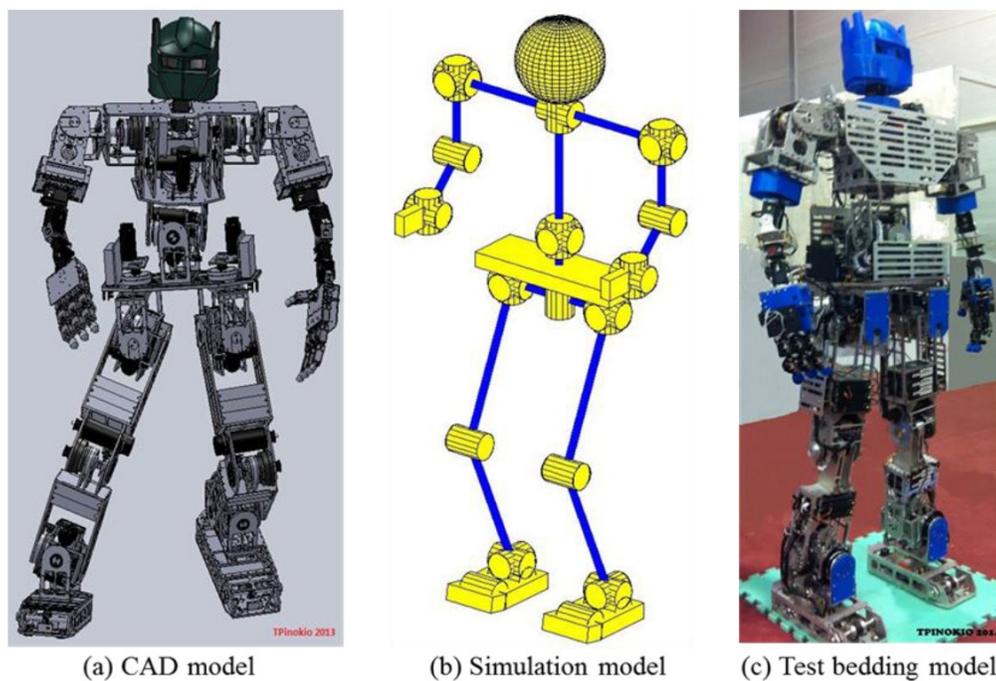


Figure 1.1. TPinokio Bipedal humanoid robot and its simulation model.

1.3 Robot and Coordinate Definitions

The hypothesis on the bipedal robot, the walking gait and the impact conditions are as follows.

The bipedal robot:

- Has two symmetrical legs, connected to the hip and the end of each leg is terminated with forward facing foot.

- Comprises of rigid revolute joints connected by light weight links to form a kinematic chain. The rigid links are assumed to have zero mass and inertia.
- The revolute joints are independently actuated.

Walking gait:

- Have a fully actuated Single-Support-Phase (SSP) and Double-Support-Phase (DSP). An extra Under-Actuated-Phase (UAP) is included if the robot has a passive toe-feet.
- During the support phases, the support foot or feet remains still on the contact surface with no slippage.

Impact conditions:

- The impact force generated by the landing foot and the contact surface is instantaneous.
- The velocity and momentum of the swing leg is zero immediately upon impact, no rebound and slippage.
- The impact forces can be represented by impulses.

The definition of the main coordinate system and the robot body parts used in the thesis is depicted in Figure 1.2.

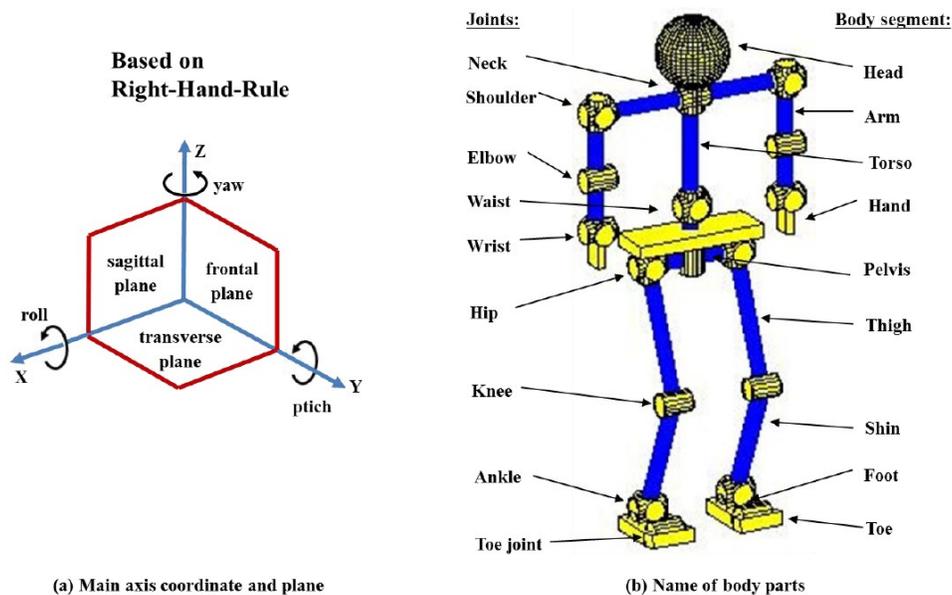


Figure 1.2. Definition of robot main axis coordinates and body parts.

1.4 Methodology

In general, locomotion of a bipedal robot consists of motion planning and balancing control. Although humanoid robots have different designs, especially in their leg mechanism, there are still some common features found in the humanoid robot. Some general features include: At least six fully articulated and position controlled joints for each leg, there is an intelligent motion control algorithm and the robot's feet have embedded sensors to measure the feet-surface contact points and contact forces.

The approach taken in this research is to apply an Augmented Model Predictive Control (AMPC) algorithm with on-line feed forward to process future data to optimize a control signal. It is done by minimizing a cost function so that the robot's Center of Mass (CoM) is able to track the reference Zero Moment Point (ZMP) as close as possible, and at the same time, this action is able to limit the motion jerk that will improve the stability of the robot's walking. Furthermore, the walking gait of the robot is improved with on-line sensing of its environmental conditions with sensory feedback and fine tuning of its motion parameters.

The bipedal robot is first linearized and modeled as a Linear Inverted Pendulum Model (LIPM). The transfer function is discretized and the system's '*jerk*' parameter is introduced into the equation as a control signal (ΔU). The final system's model is represented by a 3×3 state-space model. A reference trajectory, in this case, is the desire ZMP, is fed into the AMPC algorithm to produce a resultant CoM trajectory that will track the ZMP as close as possible. The following cost function is used in the optimization process to investigate the system's performance:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U .$$

Other performance algorithms including walking with passive-toe, walking on slope, stairs climbing, step over an obstacle, are also evaluated and simulated with MATLAB in this research.

A mid-size humanoid robot, TPinokio, with passive toe-feet has been designed and built for this research project. It is designed with light weight joint-link and with compact joint housing

mechanism. As a result, it has a simple kinematics structure and it can be modeled as a ‘single-mass’ system. Without stabilization, the robot will not walk steadily and may fall and damaged. Thus, we need a feedback controller with build-in stabilization algorithms, and in turn it modifies the reference trajectory with the help of sensors information and eventually improves the overall system stability. The robot’s feet are fitted with force/torque sensors to measure the reaction forces magnitude and location; the measured values provide feedback signals to the controller to further improve the system stability. The illustration is as shown in Figure 1.3. An effective balancing controller and various control algorithms are also proposed and discussed in this thesis.

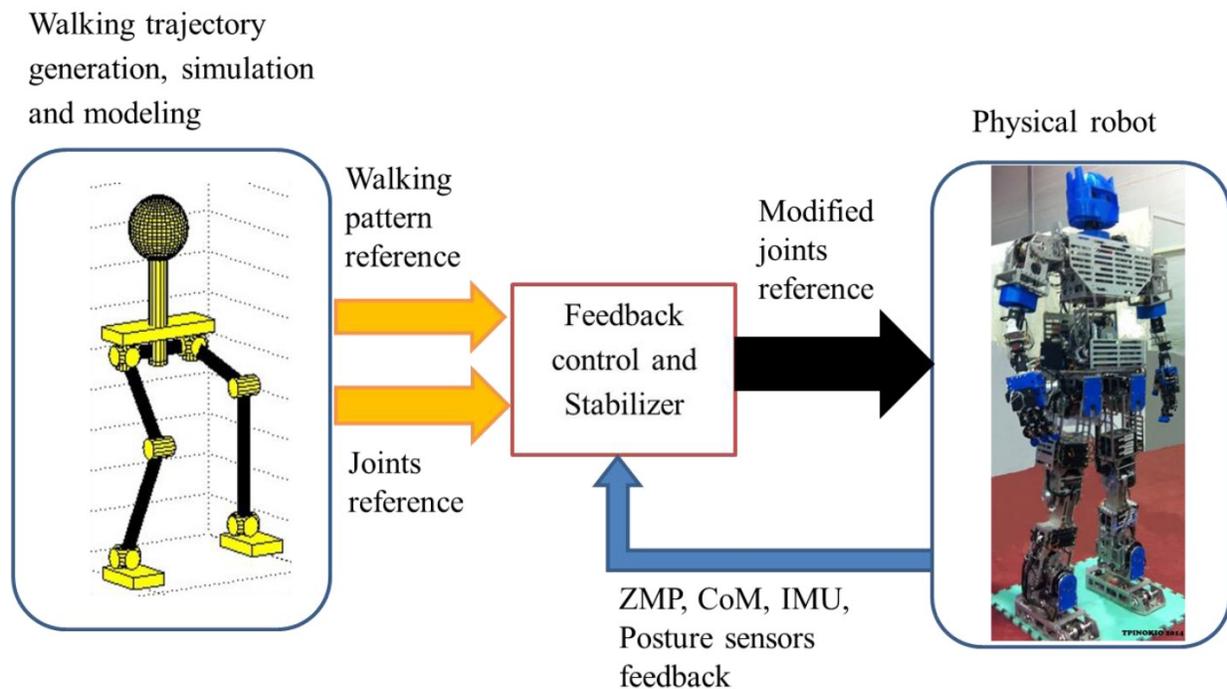


Figure 1.3. Feedback and stabilizer control.

1.5 Thesis Outline

This report is organized in Chapters with the following contents:

Chapter 1

This chapter will state the objectives, motivations and a summary of the author's research works.

Chapter 2

This chapter will provide a brief literature review of different robots' design concept. It will also present literature review of bipedal humanoid robot research that is relevant and related to this research and lastly to outline and categorise bipedal walking into different groups and the research concepts of each group.

Chapter 3

This chapter will discuss the mechanism and design concept of the TPinokio bipedal robot. The robot waist and above upper body has more than 18 DoF, its legs and hip has 13 Degrees of Freedom (DoF). The design of the joint link is modular, compact and has zero backlashes. In addition, this chapter will also present the robot's passive toe-feet design. This concept provides a better walking posture and is able to achieve a higher stairs climbing and obstacle clearance step.

Chapter 4

This chapter explain the kinematics and dynamics modelling of the TPinokio bipedal robot. The multi-point-mass kinematics and dynamics modelling methodology is first explained, then deriving the simplified 3D Inverted Pendulum Model (IPM) and other models used in the trajectories generation. This chapter also discusses the inverse kinematic algorithms and its implementation.

Chapter 5

Here, it will explain the control and walking pattern generation of the bipedal robot. Besides that, this chapter will also present the proposed motion control algorithms and description of the walking gait generation methods and explain the simulation tools used in the design. Applying the Augmented Model Predictive Control (AMPC) technique to control the bipedal robot is motivated by its capability in handling constraints. This method is effective for walking gait pattern synthesis and controlling of bipedal robot locomotion, which is subjected to disturbances or constraints in an unstructured environment.

Chapter 6

Bipedal locomotion control consists of a gait pattern generation block and a stabilization controller block. Without stabilization control, the bipedal cannot walk firmly. In this chapter, the stabilization concept which works along a given walking pattern and the posture control technique will be presented to realize a stable biped walking.

Chapter 7

A summary of the results finding and a conclusion is made in this chapter. This chapter also provides suggestions for future works.

Appendices:

It contains some supplementary technical information, and also a list of all MATLAB TPinkio bipedal Toolbox that was developed during the course of this research.

1.6 Research Contributions

This thesis documents the author's progress and achievements made in the research works done on the humanoid bipedal robot TPinkio in last three year's.

During the first year, the author focused on the robot mechanical and actuating mechanism, including the design, fabrication and testing of the mechanical structure. The objective is to construct a robot with a simple kinematics structure, so that it can be modeled as a simplified 'point-mass' system. After the mechanical problems are solved, the 2nd and 3rd year duration concentrated on the motion gaits, trajectories generation and balancing control. The ultimate goal is to explore and implement a set of suitable stability and control algorithms, which will produce a stable gait for a bipedal robot to walk steadily on different terrains like a flat ground, slope and step up stairs. A new approach is proposed and simulations have been done using the combination of Augmented Model Predictive Control (AMPC) method and the Inverted Pendulum Model (IPM) concept, to solve the Center-of-Mass (CoM) and Zero-Moment-Point (ZMP) tracking issues.

The Research Contributions in this research are

- A new method using AMPC/IPM/ZMP concepts with optimized gait control algorithm has been proposed and validated.
- A unique compact modular-joint, with simple kinematic structure, mid-size humanoid robot has been designed and built.

1.7 Publications

Some of the research presented in this thesis has appeared in the following publications.

- [1] T. C. Wee, A. Astolfi, M. Xie, Modular Joint Design of a TPINOKIO Bipedal Robot, 8th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2011. pp.185-190.
- [2] T.C. Wee, A. Astolfi, M. Xie, Modular-Joint Design for Efficient Walking and Modeling of a Bipedal Robot TPinokio, 14th International Conference on Climbing and Walking Robots, CLAWAR 2011. pp.493-500.
- [3] T. C. Wee, A. Astolfi, M. Xie , Control of Bipedal Robot (TPinokio) with Augmented Model Predictive Control and Inverted Pendulum Method, 2nd International Conference on Electronics, Communications and Control, ICECC 2012, pp.2781-2784.
- [4] T.C. Wee, A. Astolfi, M. Xie, Robust Control of Bipedal Humanoid (TPinokio), Journal Procedia Engineering Vol. 41, Elsevier 2012, pp. 643-649.
- [5] T.C. Wee, A. Astolfi, M. Xie, Design and Control of a Bipedal Robot with Sensory feedback, International Journal of Advanced Robotic Systems, 2013, pp.1-12.
- [6] T.C. Wee, Mechanical design and optimal control of humanoid robot (TPinokio), The Journal of Engineering, IET 2014, pp. 1-12.

Chapter 2

Literature Review and Background Theory

This chapter presents the development history and also discuss on the literature review of bipedal humanoid robot research that is relevant and related to this research works. The literature review sections also outlined and categorized bipedal walking into different groups and the design concept for each group.

2.1 General Information

In designing the control of the ‘walking’ of a stable bipedal humanoid, one has to take into consideration of its complexity and at the same time, not to overlook the primary need to ensure that the biped is at all times maintain its balance while walking on unstructured environment. It is therefore a challenging task indeed.

Many studies and works were done from the 70s. Some of them are mentioned below, Vukobratovic [M. Vukobratovic et al., 1973] was one of the pioneers in studying bipedal locomotion, and Miyazaki [F. Miyazaki, S. Arimoto et al., 1980] proposed reduced-order equations of motion model for dynamic walking. Miura [H. Miura, I. Shimoyama et al., 1984] developed the inverted pendulum model concept and applied it on bipedal walking, and the stability analysis conducted had shown that dynamically stable walking mode can be achieved. Mita [T. Mita, T. Yamaguchi et al., 1984] applied modern optimal regulator control theory on plane dynamic biped locomotion. Raibert [M. Raibert , 1986] implemented different landing control algorithms on a one-leg hopping robot with rhythmical high speed walk. Furusho [J. Furusho, A. Sano et al., 1990] started experiments on 3D bipedal robot with force/torque sensor-based feedback control. Li [Q. Li, A. Takanishi et al., 1991] applied the “zero-moment-point“ concept to control the motion of the body in a full 12 DoF bipedal robot (the WL-12RIII). Kajita

[S. Kajita, T. Yamaura et al., 1992] proposed an interesting concept of potential energy conserving orbit to formulate new control laws to experiment on their bipedal with an average speed of 0.2m/s. Miller [W.T. Miller, 1994] and Grishin [A. Grishin, A. Formal et al., 1994] applied modern control theory, real-time neural network and adaptive control algorithm to produce stable dynamic walking. The study of bipedal walking on rugged environment began in the early 90s. Zheng [Y. Zheng , J. Shen et al., 1990] studied on walking characteristic on slope surface, and Yamaguchi [J. Yamaguchi, A. Takanishi, et al., 1995] designed a bipedal robot with shock absorber to walk on rough terrains. Since the late 90s, various control strategies and synthesizing methods to generate bipedal walking trajectories were initiated. These strategies will consist of taking data and records on the human walking characteristics, defining the finites states machine and not forgetting having feedback sensory systems. Besides that, we will have strategies that give passive gaits generation with gravity inertia and optimal gait trajectories with modern control laws. They have enriched the bipedal research community and grow by leaps and bounds.

2.2 Bipedal Development History by Various Research Institutions

Many automatic structures were built to mimic human movements in the early 18th century. Leonardo de Vinci was one of the first people to create the automatic structures in form of a humanoid as shown in Figure 2.1a. In the early 20th century, the Westinghouse Society built the first electrical structure and coined is as “Elektro humanoid”, as shown in Figure 2.1b.



(a) Leonardo de Vinci robot



(b) Elektro robot

Figure 2.1. Pictures of 19th century robots.

It was in the 1970s that human-like bipedal robots started to appear. The University of Lomonosov in Moscow built a legged robot OstRover as shown in Figure 2.2a. In the USA, the MIT built a jumping robot ‘Spring Flamingo’ was one of the pioneers to walk and run with a dynamically stable gait as shown in Figure 2.2b. In the 20th century, Japan Waseda University carried out studies on bipedal robotic walking. As depicted in Figure 2.3, subsequently, the team in the University developed a series of bipedal humanoid robots, namely the Wabot-1, WL-10 and WABIAN-2R

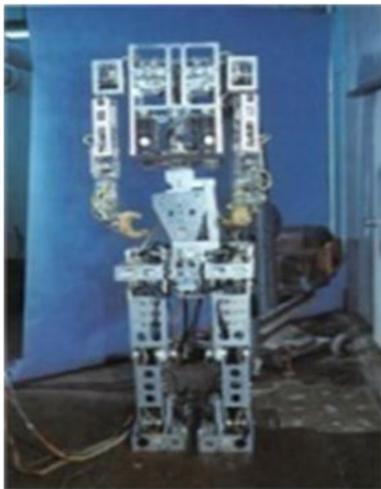


(a) OstRover

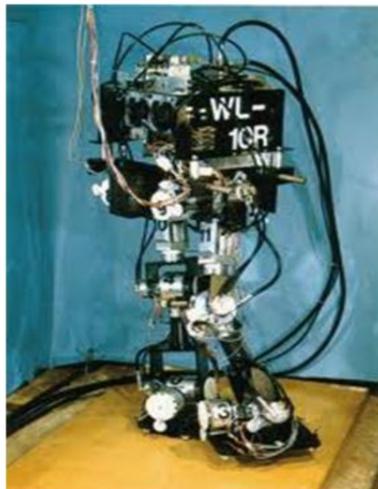


(b) MIT Spring Flamingo

Figure 2.2. The 1960/70 robots.



(a) Wabot-1



(b) WL-10



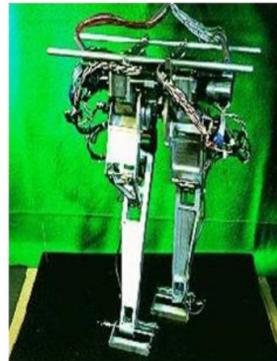
(c) WABIAN-2R

Figure 2.3. Waseda university WL series robots.

In the 1980s, Japanese researcher developed the bipedal robot BIPER, as depicted in Figure 2.4a. This robot BIPER, can perform dynamic walking and had passive articulation at the ankles with a large feet to maintenance balance. In 1990, Kajita and Tani constructed the MELTRAN-II robot with the concept of an inverted pendulum motion, as shown in Figure 2.4b.



(a) BIPER



(b) MELTRAN-II

Figure 2.4. Robots designed and built by Japanese researchers.

2.3 Recent Developments on Bipedal Robots

More recent humanoids will exhibit emotions and are able to ‘learn while they walk’ as they learn through autonomous interaction with the environment. Bipedal humanoid robotics offers an alternative research tool in studying the human behavior and cognitive science. The largest concentration of research activity of service robots are presently in Japan and Korea. Japan’s Mitsubishi Research Institute predicts that the future industry for humanoid service robots is forecasted to worth around USD 50 billion by 2020, and each household would own at least one service robot by 2020.

2.3.1 Japan

In Japan the development of bipedal and humanoid robots started very early. They were spearheaded by the industrial companies. The Humanoid Robotics Project (HRP) was supported by the Japan Ministry of Economics, Commerce and Industry (METI). Honda had built a series

of bipedal robots since the 90s. Honda began to design P1 to P3 and then later on progress to the design of the ASIMO. The ASIMO robot is 1.4 meter high and has more than 25 DoF, as depicted in Figure 2.5a. The Toyota humanoid robot, as shown in Figure 2.5c is capable of playing musical instrument and the SONY Qrio robot, as shown in Figure 2.5d is the first kid-size bipedal robot built in 2003.

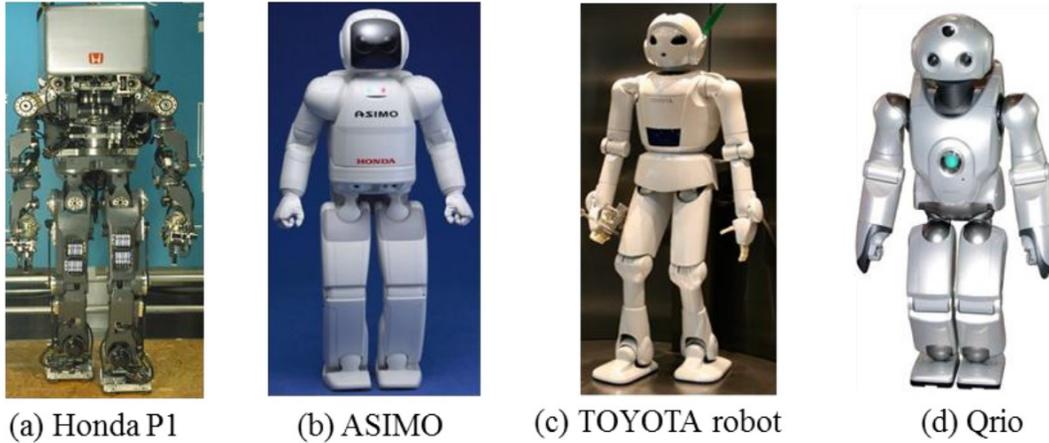


Figure 2.5. Japan Robots.

The HRP series robots, as depicted in Figure 2.6, were developed by Kawada Industries and the AIST institute. The HRP-2 robot has more than 30 DoF, is approximately 1.52m high, and weighs around 58 kg. It can achieve a walking speed of around 2.5 km/h, and comes with in-built vision cameras, force sensors with attitude sensors. With such a design, the robot was able to control its own stability, whilst it controls its tasks. The latest version of the series is HRP-4.

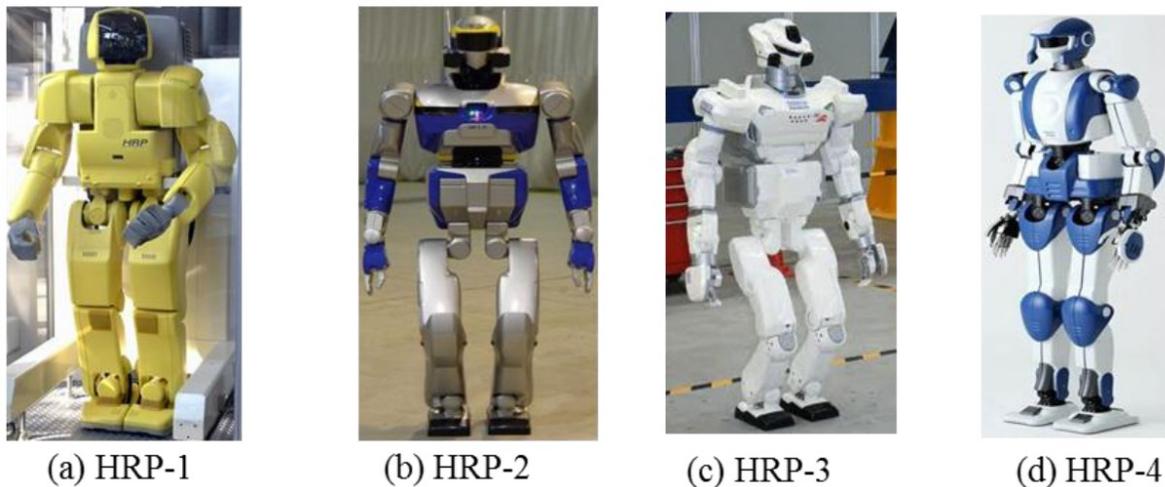
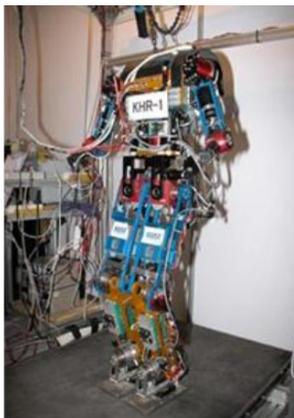


Figure 2.6. HRP series robots.

2.3.2 Korea

In Korea, as depicted in Figure 2.7, bipedal robots KHR-series have been developed in Korea by Korea Advanced Institute of Science and Technology (KAIST) since 2000. The KHR-1 was designed with no hands and head and consists of 21 DoF. KHR-2 is a 41 DoF humanoid which could walk with human-like movement. The latest development of HUBO in 2010 is more refined, with its movements more graceful like human and also come with more human-like features.



(a) KHR-1



(b) KHR-2



(c) HUBO

Figure 2.7. KHR series robots.

2.3.3 France

Strasbourg University LSIT laboratory in France carried out the initial studies on legged robots. Their bipedal robot had six joints with only legs and feet. As depicted in Figure 2.8(a), the University of Versailles LIRIS lab built a biped ROBIAN for test bedding. The BIP2000 robot, is a joint R&D works from INRIA Rhone-Alpes and LMS Poitiers, as depicted in Figure 2.8(b), it is approximately 1.82m tall and weighs around 100kg. The Rabbit robot, with point feet, as shown in Figure 2.8(c), was built in 1998 in collaboration with the University of Michigan (USA) to study planar dynamics motion. The Nao robot, as depicted in Figure 2.8(d), was built in 2006 by the company Aldebaran, is a 58-cm tall humanoid robot and is successfully used in computer and science classes for education. The France-Japanese Joint Robotics Laboratory (JRL) was formed recently to enable the French robotics researchers to enhance their researches on legged robot.

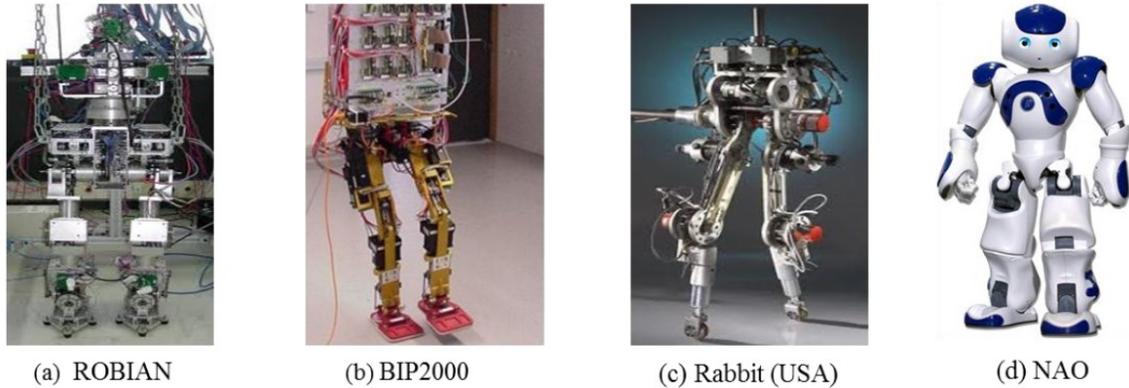


Figure 2.8. Robots designed by France researchers.

2.3.4 Other European Countries

In Europe, as shown in Figure 2.9, humanoid robots built by researchers and universities include Johnnie/LOLA from University of Munich Germany, which is powered by linear actuators, and LUCY by Vrije Universiteit Brussel, using pneumatic muscle, and a two-year old child like humanoid robot called iCub from University of Genova Italy. The iCub, as depicted in Figure 2.9(c) was developed as part of the EU project RobotCub and subsequently adopted by more than 20 laboratories worldwide.

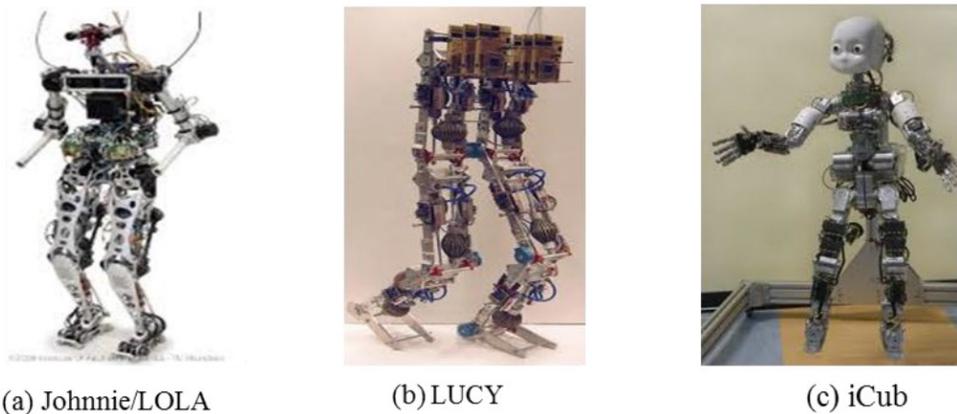


Figure 2.9. Robots from various European universities.

2.3.5 USA

In the USA, as shown in Figure 2.10, among the numerous humanoid robots that were developed in the different varsities, researchers at NASA have developed a robonaut robot

‘Valkyrie’, measuring in at around 1.9m and around 125kg, the robot is capable of manipulating objects in space. Organized by the United States military’s Defense Agency, The DARPA Robotics Competition 2014 [<http://www.darpa.mil/default.aspx>] was the center of attraction. This project is centered on the development of autonomous humanoid robots to explore disaster areas and to deploy in search and rescue missions.

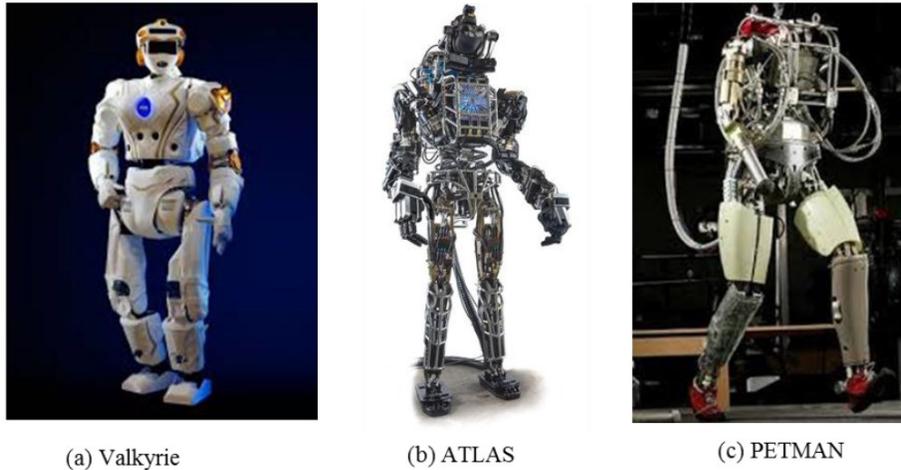


Figure 2.10. Robots designed by USA researchers.

2.3.6 Singapore

In Singapore, as shown in Figure 2.11, the design of bipedal robots was carried out at the National University of Singapore, at the Nanyang Technologies University and at the Temasek Polytechnic since 2007.

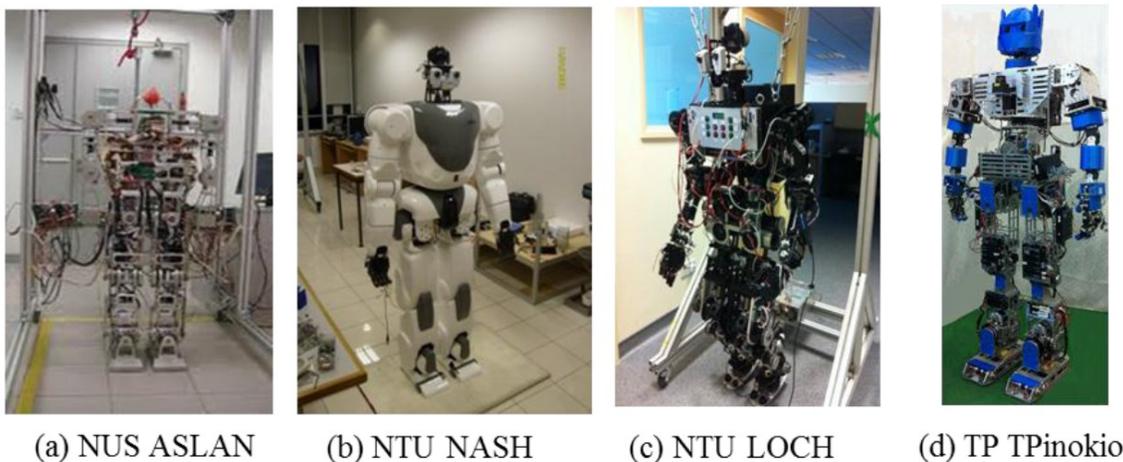


Figure 2.11. Robots designed by Singapore institutions

2.4 Robot Mechanical Design

There are many criteria to consider when developing a bipedal robot. For instance, we need to consider the mechanical structure, the actuator type, the weight to power ratio, the sensors types, the dynamics, the kinematics and, inevitably, the cost. The three most current types of joint mechanism are the electric revolute joint, the linear actuator joint and the artificial muscle joint.

2.4.1 Electric Revolute Joint

Electrical motors with Harmonic Gear (HG) drives are commonly used in many revolute-joint designs. The advantage of such gear is that it is compact, has zero-backlash and high torque ratio. Most of the fully 3D articulated bipedal robots such as HUBO, HRP and LOCH have electric revolute-joints, as depicted in Figure 2.12.

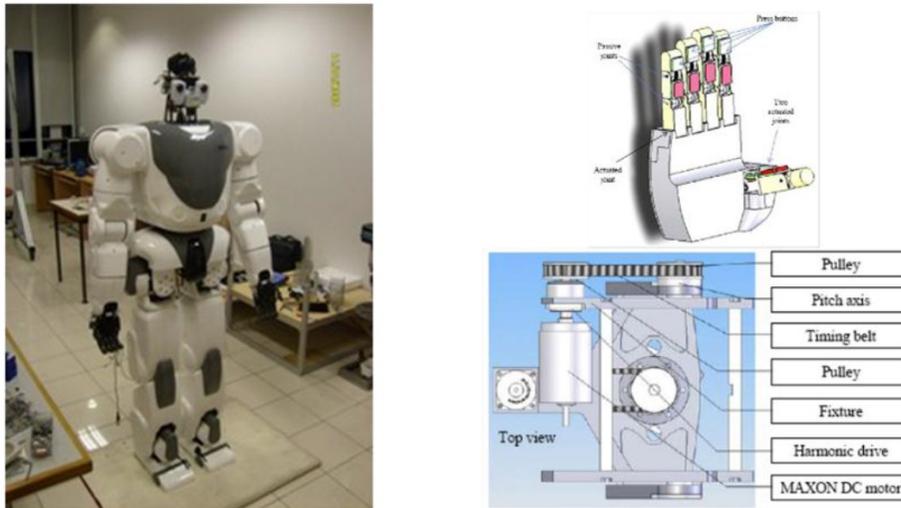


Figure 2.12. LOCH (NTU) robot with electric revolute-joint design concept.

2.4.2 Linear Actuator Joint

Robot LOLA [L. Sebastian, B. Thomas et al., 2009] as depicted in Figure 2.13, is a 3D biped and each leg has six DoF. It is designed with ball-screw based linear actuator joints concept. The limitation with such design is that the ball-screw mechanism is coupled to the ankle joint in a parallel configuration. As such, the mechanism is much more complex and difficult to control than the HG revolute-joints.

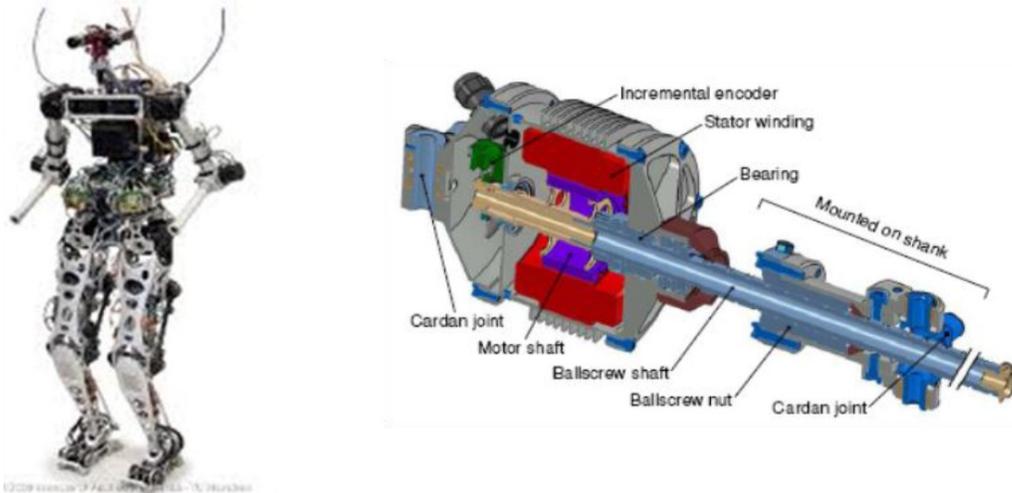


Figure 2.13. LOLA robot with Linear actuator design concept.

2.4.3 Artificial Muscle Joint

Robot LUCY [B. Vanderborcht, B. Verrelst et al., 2005], as depicted in Figure 2.14, is a seven link planar (sagittal plane) walking robot. Its mechanical structure employs the Pleated Pneumatic Artificial Muscles (PPAM) drive. As the artificial muscles are pneumatically driven, there is only a ‘pull’ action. The robot will need a bi-directional joint. To create a working ‘bi-directional’ revolute joint movement, two muscles must be coupled antagonistically to create the motion. It is to note that the kinematics model and torque characteristic are different from the electric revolute joint design.



Figure 2.14. LUCY Artificial muscle design concept.

2.5 Robot Configurations

2.5.1 Coordinate

Most of the current bipedal humanoid robots are designed to have fewer Degrees of Freedom (DoF) than human being. This is because the current robotics technology is unable to reproduce the complex of the human skeletal structure. The number of DoF for each joint is limited to operational requirement and thus the mobility of a bipedal robot is designed to provide only the essential or critical movement. Refer to Figure 2.15, in the robot; the design of its structure will be referenced by three coordinate planes, namely, the x-sagittal plane, the y-frontal plane and the z-transversal plane. The right hand XYZ Cartesian coordinate is adopted.

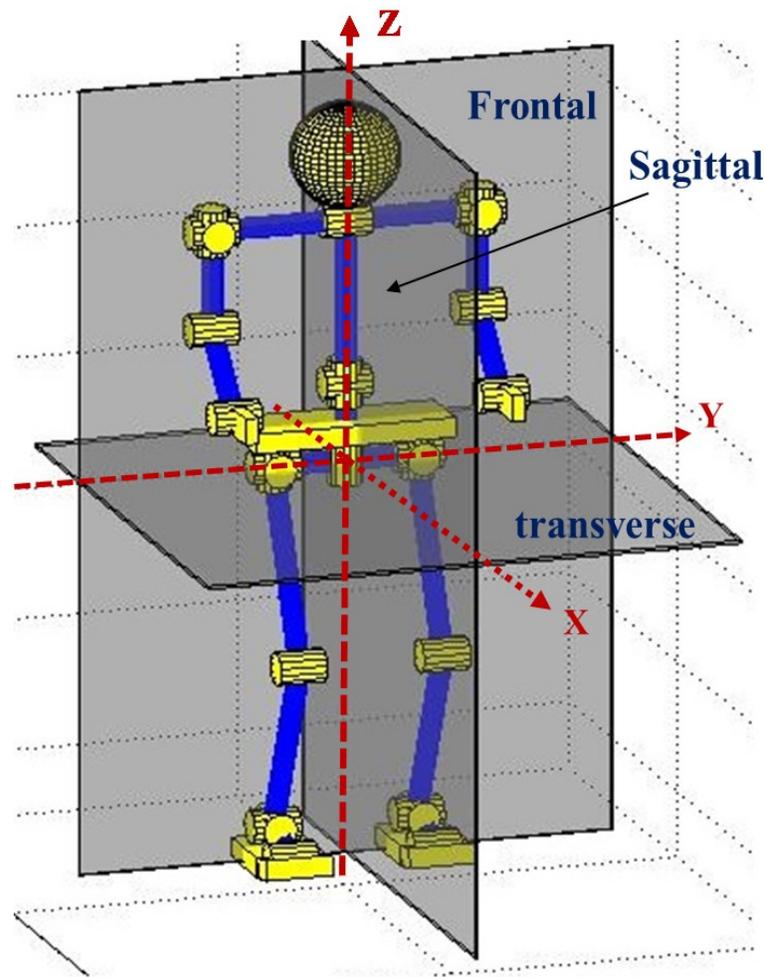


Figure 2.15. The robot planes of section and Cartesian coordinate.

2.5.2 Foot Movement

A robot walking gait is represented by eight sequences [M. Xie, Z.W. Zhong et al., 2009] and divided into two phases in a loop cycle as illustrated in Figure 2.16, with the following definition:

- Stance : the duration when both feet is in contact with the surface.
- Swing : the duration when one of the feet is lifted from the surface.
- Single support : the moment when only one leg is supporting the robot.
- Double support : the moment when both legs are supporting the robot.
- Heel-strike : the moment when the heel of the swing foot contact the surface.
- Toe-off : the moment when the foot swings off the contact surface.

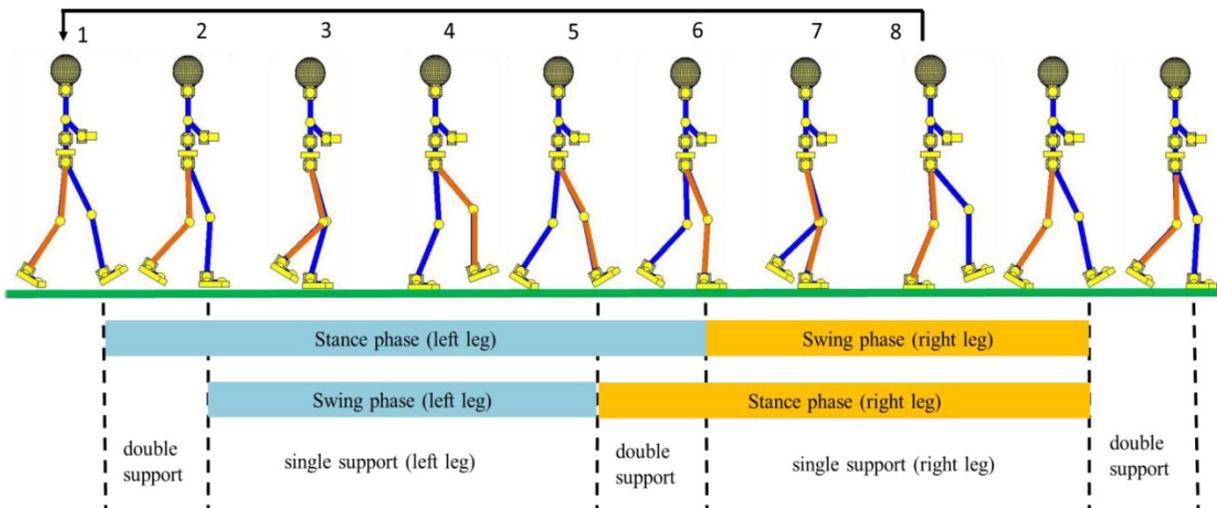


Figure 2.16. A typical robot walking cycle.

2.6 Forward and Inverse Kinematics Model

Kinematics, in robotic mechanism, describes the motion of a body without considering the force/torque acting on its body or causes the motion. Forward kinematics is a process of mapping a robot manipulator from Joint-space to Cartesian-space (sometimes known as task space). For instance, if the coordinates of a manipulator is defined in vector form, the joint angles is $\theta \in \mathcal{R}^n$, and the orientation/position of the manipulator's end effector is $x \in \mathcal{R}^m$, the forward kinematic function can be expressed as $x = f(\theta)$. Similarly, the inverse relationship is $\theta =$

$f^{-1}(x)$. Note that in a redundant manipulator ($n > m$), the solution will be non-unique, and even with $n = m$, multiple solutions may exist [R. N. Jazar, 2010].

2.6.1 Forward Kinematics

A bipedal robot resemble a serial-chain mechanism, the forward kinematics of the joint link can be determined by transforming the kinematic information of the robot's Joint-space to the Cartesian-space. In general, the coordinate of a robot end-effector transformation matrix with respect to the base frame can be determined by the Denavit-Hartenberg (D-H) method [R.N. Jazar, 2010].

2.6.1.1 Denavit-Hartenberg (D-H) transformation

This method is to transform between two adjacent coordinate frames. Applying the D-H convention, the coordinate frames B_i is transformed to the coordinate frames to B_{i-1} by the homogenous transformation matrix ${}^{i-1}T_i$:

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}d_i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.1)$$

where ${}^{i-1}R_i$ is a 3×3 matrix representing rotation and ${}^{i-1}d_i$ is a 3×1 vector representing translation. Similarly, the inverse transformation equation to transform the coordinate frame $B_i(x_i, y_i, z_i)$ to coordinate frame $B_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ is

$$\begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \\ 1 \end{bmatrix} = {}^{i-1}T_i \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}.$$

The forward position kinematics of a joint coordinate with reference to the base frame can be solved from the combined transformation matrix

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \dots \dots {}^{n-1}T_n(q_n). \quad (2.2)$$

Please refer to Appendix B for details.

2.6.2 Inverse Kinematics

In bipedal robotics research, solving for inverse kinematics is one of the challenging areas because of its high complexity and degrees of freedom (DoF). There are three primary methods to solve for inverse kinematics, namely, the conventional algebraic, geometric and iterative methods. Each method has its own limitations and advantages. Both the geometric method and algebraic method give a definite set of solutions. The iterative method provides generalized numerical solutions and converges to one solution space. All three methods are able to provide real-time computation solutions of inverse kinematics.

2.6.2.1 Closed-Form Solutions

Solutions that are closed-form take a shorter time to derive when compared with when finding for numerical solutions. However, in the case for closed-form solution, they are robot dependent. Generally, the closed-form solutions can be derived in a 6-DOF system with special kinematic geometric features [B. Siciliano, O. Khatib, 2008]. In general, algebraic and geometric methods will result in closed-form solution.

2.6.2.1.1 Algebraic Methods

Algebraic method is to identify significant equations that contain the joint variables and then these equations will be manipulated to a soluble form. After which, reducing to a single variable equation is often the way to solve such problems [B. Siciliano, O. Khatib, 2008].

2.6.2.1.2 Geometric Methods

A closed-form solution may also be derived using the geometry of the bipedal robot with known geometry. The constraint is that the closed-form solution can be applied only to a specific geometry. [D. Goswami, P. Vadakkepat et al., 2009], [T. Nguyen, W.K. Dae et al., 2008] used this method to derive the solution for a 13 DoF humanoid robot.

2.6.2.2 Numerical Methods

A polynomial equation with degree four or less will be able to produce a closed-form solution. Thus any higher degree polynomial will use numerical methods to obtain solutions. The added

advantage is that this method can be used in any kinematic structure as it is not robot dependent. Symbolic elimination methods and iterative methods are commonly used besides the continuation method under the classification for Numerical Methods

2.6.2.2.1 Symbolic Elimination Methods

Symbolic elimination method is to eliminate and reduce the system's nonlinear equations variables to a set of manageable equations by using analytical manipulations. Raghavan [M. Raghavan, B. Roth, 1990] applied dialytic elimination to determine the inverse kinematics equation of a 6 DoF manipulator. Manocha [D. Manocha, J. Canny, 1992] further improved this technique to a generalized eigenvalue problem.

2.6.2.2.2 Continuation Methods

These techniques are applied to inverse kinematics problems by exploiting the intrinsic properties of polynomial systems to derive the possible solutions [L.W. Tsai, A.P. Morgan, 1985]. These methods will basically track the solution path of the system. Tracking will start from a system with known solutions, to solve for the desired target system recursively. Solutions will be obtained when the original system is progressively translated to the target system. [C.W. Wampler, A.P. Morgan et al., 1990].

2.6.2.2.3 Iterative Methods

Different types of approaches with iterative methods can also solve the inverse kinematics problem. Often, these approaches are able to converge to one solution that is based on a good initial guess, the time taken to converge depend on the 'quality' of that initial guess. Pieper [D. Pieper, 1968] was among the pioneers to apply the *Newton-Raphson* iterative method. Other methods include using 'resolved motion rate control' method, 'modified predictor-corrector algorithm' method and using 'interval analysis' method that allows convergence to a solution [B. Siciliano, O. Khatib, 2008]. The derivation of the Jacobian is needed in some minimization algorithms, and the pseudo-inverse for inverting a non-square and singular Jacobian. The inverse kinematics can be seen as identifying and solving for q_k from a set of nonlinear matrices

$${}^0T_n = \mathbf{T}(q) = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \dots$$

$${}^0T_n = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.3)$$

or $r_{ij} = r_{ij}(q_k) \quad k = 1, 2, \dots, n.$

where n denote the DoF.

Equation (2.3) is the *Newton-Raphson* method used to determine the joints' variables (q_k).

The kinematic equations is solved by applying the iterative technique, by setting

$$\mathbf{T}(q) = \mathbf{0}, \quad (2.4)$$

and start with an initial guess for the joints' variables \mathbf{q} ,

$$\mathbf{q}^* = \mathbf{q} + \delta\mathbf{q}. \quad (2.5)$$

The configuration of the end-effector frame can be obtained from the joint variables equation

$$\mathbf{T}^* = \mathbf{T}(\mathbf{q}^*). \quad (2.6)$$

The error between the desired configurations and the calculated values using the forward kinematics, which is called 'residue' must be minimized:

$$\delta\mathbf{T} = \mathbf{T} - \mathbf{T}^*. \quad (2.7)$$

The Jacobian matrix of the set of equations is

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \frac{\partial T_i}{\partial q_j} \end{bmatrix}. \quad (2.8)$$

Please refer to Appendix C for a details description of inverse kinematics iteration algorithm and technique.

2.7 Dynamic Model

Dynamic equations of motion describe the relationships between the actuation and contact forces exerting on the motion of a rigid mechanism object, with the acceleration and motion trajectories that are due to the reacting forces.

2.7.1 Lagrange Method

Bipedal robots are walking mechanism with rigid legs. Torque commands to the joint are required to allow each joint motion to follow a desired trajectory. The desired torque/motion can be derived by using the Lagrange equations of motion [Reza N. Jazar, 2010],[E.R. Westervelt, J.W. Grizzle et al., 2007]. The equation is written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{U} . \quad (2.9)$$

Where \mathbf{q} is the joints angle, the matrix \mathbf{M} is the inertia matrix, \mathbf{V} is the velocity coupling matrix (Coriolis, centrifugal forces etc), \mathbf{G} is the gravity matrix, \mathbf{B} is the joints' torque / forces mapping matrix, and finally \mathbf{U} is the applied torques.

A robot mechanical system is *fully-actuated* when the numbers of independent actuators is the same as the numbers of DoF, and if the numbers of actuators are fewer than the DoF, then it is said to be *under-actuated*.

2.7.1.1 Fully-actuated Phase

The dynamics model of a bipedal robot during the *fully-actuated* phase resemble a N DoF rigid-link mechanism actuated at the support foot ankle [E.R. Westervelt, J.W. Grizzle et al., 2007], and without slipping and is motionless during this phase. The stance foot acts as an *actuated* pivot. Let $\mathbf{q}_a = [q_1, q_2 \dots ; q_N]^T$ denote the joints' variables, where q_N is the support foot ankle joint angle, and q_1, q_2, \dots, q_{N-1} denote the angles of all other joints. The fully actuated dynamic equation is

$$\mathbf{M}_a(\mathbf{q}_a)\ddot{\mathbf{q}}_a + \mathbf{V}_a(\mathbf{q}_a, \dot{\mathbf{q}}_a)\dot{\mathbf{q}}_a + \mathbf{G}_a(\mathbf{q}_a) = \mathbf{B}_{a1}\mathbf{U}_b + B_{a2}u_A , \quad (2.10)$$

where $u_A = u_N$ is the ankle joint input and $\mathbf{U}_b = [u_1, u_2, \dots, u_{N-1}]^T$ is the forces/torques exerted at all the other remaining joints.

2.7.1.2 Under-actuated Phase

During *under-actuated* phase, the stance heel of the swing leg left the contact surface and the whole robot begins to roll over at the stance feet toe, and the stance toe act like an *un-actuated* pivot with no actuation between the ground and the stance feet toe, the dynamics of the robot in this phase is equivalent to an N+1 DoF robot with *un-actuated* point feet. The dynamic equation can be determined by applying Lagrange method, this yields

$$\mathbf{M}_u(\mathbf{q}_u)\ddot{\mathbf{q}}_u + \mathbf{V}_u(\mathbf{q}_u, \dot{\mathbf{q}}_u)\dot{\mathbf{q}}_u + \mathbf{G}_u(\mathbf{q}_u) = \mathbf{B}_u\mathbf{U}_u, \quad (2.11)$$

where $\mathbf{q}_u=[q_1, q_2, \dots, q_{N+1}]^T$ are the joint angles, and $\mathbf{U}_u=[u_1, u_2, \dots, u_N, 0]^T$ is the forces/torques exerted at all the joints exclude the stance toe.

2.7.1.3 Double-Support (impact) Phase

The dynamic model of the instantaneous impact forces and reaction torques exerting at the foot ankle when the swing foot first touches the contact surface during the DSP/impact phase can be written as

$$\mathbf{M}_d(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{V}_d(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{G}_d(\mathbf{q}_d) = \mathbf{B}_d\mathbf{U}_d + \boldsymbol{\tau}_{ext}, \quad (2.12)$$

where $\boldsymbol{\tau}_{ext}$ represents the vector of external torques acting on the robot.

Applying inverse kinematics, the joints angle (\mathbf{q}) for the desired geometric path of the end-effectors (Pelvis, Left-foot, right-foot) of the robot can be determined. And the joint's motor torque command can be derived by substituting the joints angle (\mathbf{q}) into Equations (2.10), (2.11), (2.12).

2.7.2 3D Linear Inverted Pendulum Dynamic model

The human walk resembles a compass gait, as depicted in Figure 2.17. The 3D Linear Inverted Pendulum Model (3D-LIPM) which applied the compass gait technique has led to the development of simplified IPM walking model stated below:

- Human walk resembles a ‘compass gait’ where the CoM vaults over rigid legs.
- Most biological structures seem to prefer a walking pattern that has the characteristic of two peaks of the vertical ground reaction force which reassemble a compass gait.

This method was first studied by McGeer [McGeer, 1990] to explain the characteristics and the dynamics of walking. The 3D-LIPM, which is a derivation of the Inverted Pendulum Model, was proposed by Kajita [S. Kajita, K. Tani et al., 1995], and by applying the preview control algorithm, they were able to generate on-line stable walking, turning, jumping, and even running gait for bipedal robot. More details of the application of the 3DLIPM will be discussed in Chapter 5.

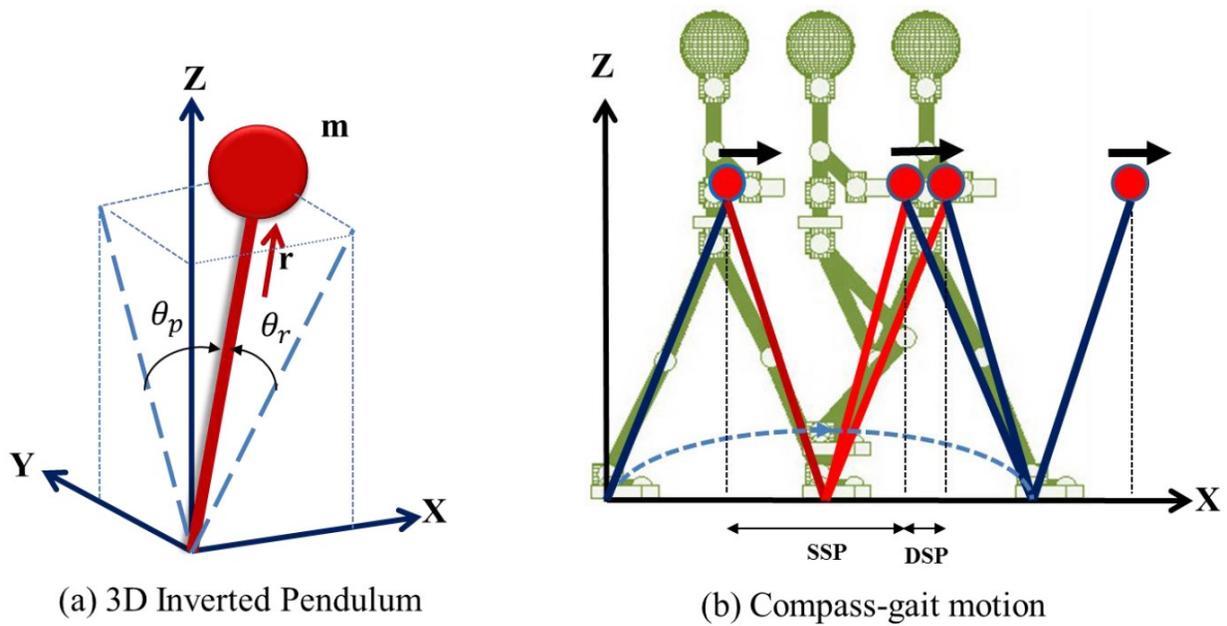


Figure 2.17. 3D Inverted pendulum model and compass-gait motion.

2.8 Center of Mass (CoM), Zero Moment Point (ZMP) and Support Convex Polygon (SCP)

The Centre of Mass (CoM) is the global position of the total weight of a rigid body as depicted in Figure 2.18. The CoM coordinate points $(x_{com}, y_{com}, z_{com})$ can be calculated by the following equations

$$x_{com} = \frac{\sum_i^n m_i x_i}{\sum_i^n m_i} \quad , \quad y_{com} = \frac{\sum_i^n m_i y_i}{\sum_i^n m_i} \quad , \quad z_{com} = \frac{\sum_i^n m_i z_i}{\sum_i^n m_i} \quad . \quad (2.13)$$

Where m_i denote the mass of the particles, p_i denote the coordinate point and n is the number of mass particles.

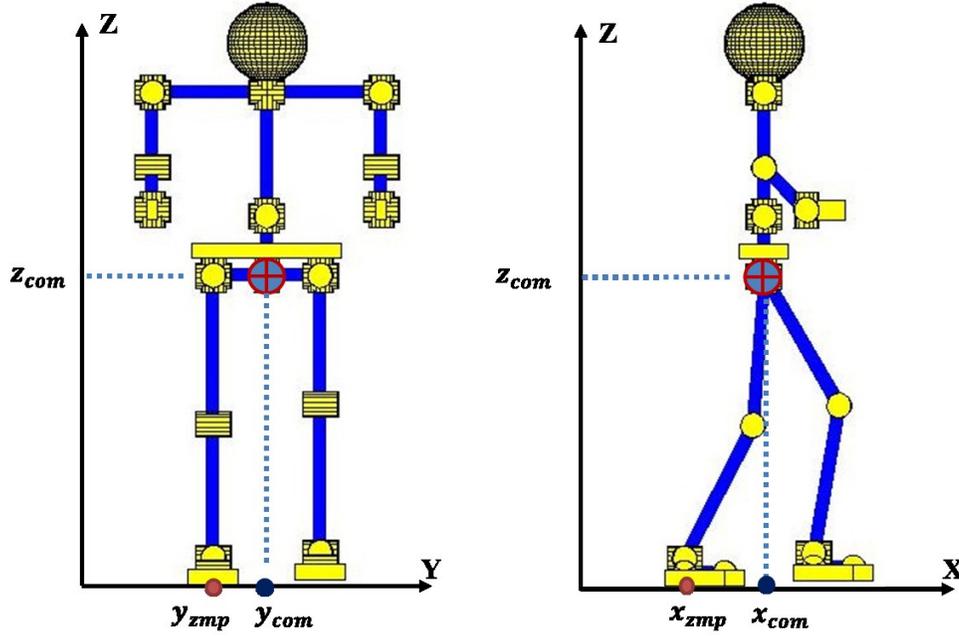


Figure 2.18. Location of the robot CoM and ZMP.

Refer to above Figure 2.18. A popular concept using the Zero-Moment-Point (ZMP) and CoM trajectory tracking method has been developed to achieve stable walking. The ZMP [M. Vukobratovic, B. Borovac, 2004] can be defined as a rotating point at the ground contact surface which the sums of all the moments exerting on this point are equal to zero. The ZMP variables [W. Pengfei, H. Bo et al., 2005] can be expressed as

$$x_{zmp} = x_{com} + \frac{\sum_i m_i \ddot{z}_i x_i - \sum_i m_i \dot{x}_i \dot{z}_i}{g \sum_i m_i} + \frac{\sum_i M_{iy}}{g \sum_i m_i},$$

$$y_{zmp} = y_{com} + \frac{\sum_i m_i \ddot{z}_i y_i - \sum_i m_i \dot{y}_i \dot{z}_i}{g \sum_i m_i} - \frac{\sum_i M_{ix}}{g \sum_i m_i}. \quad (2.14)$$

The ZMP ($x_{zmp}, y_{zmp}, 0$) is related to the ground projection [D. Goswami, P. Vadakkepat et al., 2009] of the CoM by the equations

$$x_{zmp} = x_{com} + \frac{\sum_i m_i \ddot{z}_i x_i - \sum_i m_i \ddot{x}_i z_i}{g \sum_i m_i} + \frac{\sum_i M_{iy}}{g \sum_i m_i},$$

$$y_{zmp} = y_{com} + \frac{\sum_i m_i \ddot{z}_i y_i - \sum_i m_i \ddot{y}_i z_i}{g \sum_i m_i} - \frac{\sum_i M_{ix}}{g \sum_i m_i}. \quad (2.15)$$

Where m_i is the mass of the particles, M_{ix} and M_{iy} , are the moments of the links about the *sagittal*-axis and the *frontal*-axis respectively.

The stable region, also known as the Support Convex Polygon (SCP) as illustrated in Figure 2.19, is the convex polygon of the contact area between the contact surface and the support leg foot-print. In Single Support Phase (SPP), the support polygon is the shape of the stance foot, whereas during the Double Support Phase (DSP) the support polygon is the convex area of both feet and the tangents lines.

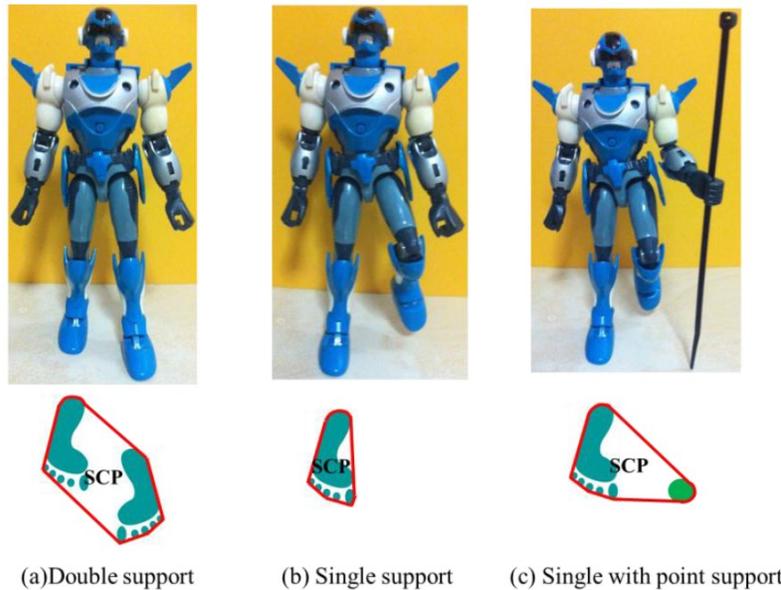
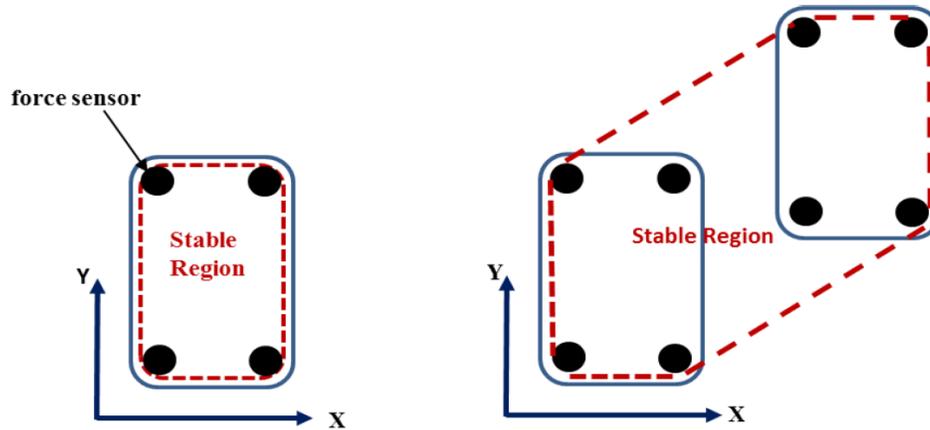


Figure 2.19. Area of the SCP is the convex area of the red envelop.

As illustrated in Figure 2.20, in order to maintain dynamic equilibrium, the ZMP must be inside the support convex polygon (SCP).



(a) Stable region of Single Support (b) Stable region of Double Support.

Figure 2.20. A typical bipedal robot support polygon stable region.

2.9 Walking Gait and Path Planning

A walking gait consists of three basic phases: the double support phases (DSP), the single support phases (SSP), and the impact phase (IMP). For a bipedal robot designed with toe-feet, the ‘rotational phase’ is also taken into consideration. A typical walking gait is illustrated in Figure 2.21. Planning for humanoid robot walking gait includes defining a trajectory curve for the swing feet and the Center of Mass (CoM), its rotational motion, and its orientations. The Cartesian path of the feet movement is expressed in time function to determine its velocity and acceleration [M. Xie, Z.W. Zhong et al., 2009]. As shown in Figure 2.20(a), the basic parameters are pelvis height (Z), frontal-shift (F), step-length (S) and foot lifting-height (H). A basic forward walking sequence assuming right feet swing first is as follows:

- At the beginning of the walking step the robot CoM shifts to the left, so that the Zero Moment Point (ZMP) is inside the left foot SCP, now the robot is in the SSP;
- The right foot leaves the ground and the right leg begins to swing forward with predefined trajectory;
- During the ending of the swinging, once the right foot contacts the ground, the robot is supported by two legs and it is now in the DSP;
- During the DSP, the robot CoM shifts from the left to that of the right foot support. The ZMP is now inside the support convex polygon of the right foot-print; and

- The right leg is now the stance leg, and the left feet lifts up and swings forward to execute the next step.

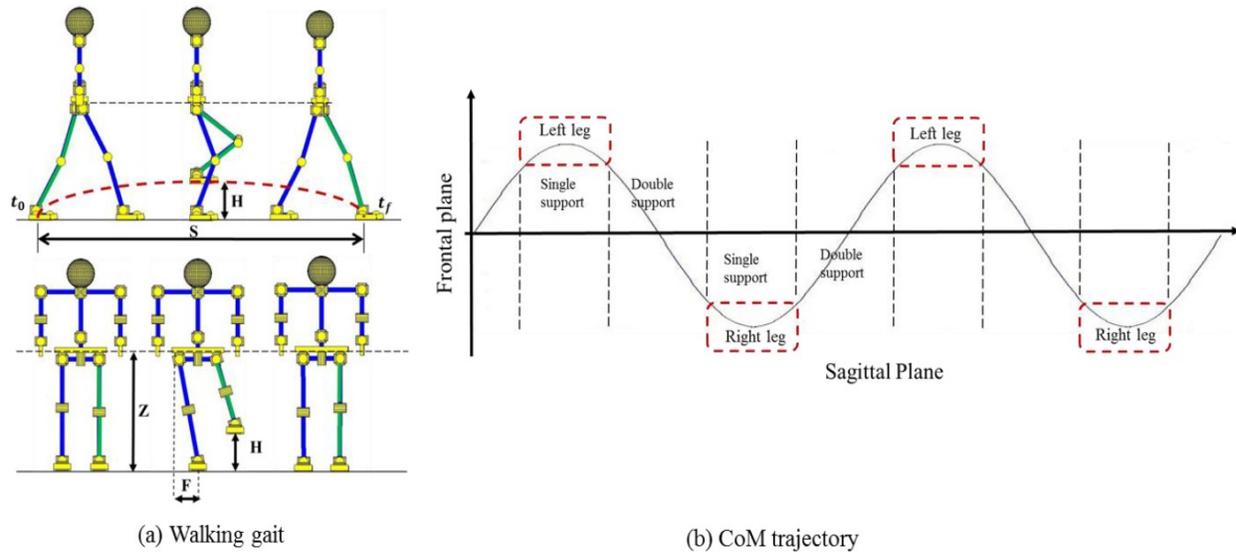


Figure 2.21. A typical biped walking pattern and its CoM path trajectory.

The desired trajectory of the swinging feet and that of the CoM is as depicted in above Figure 2.21(b). By applying the inverse kinematics, the required angular trajectory of each articular joint can be obtained. A fast walking motion requires dynamic balance while a slow walking motion may walk statically. A fast walking action also requires more robustness in the controller as disturbances will have a greater effect on the system dynamics and simultaneously there is less reaction time to correct for the disturbances.

2.9.1 Static Gait

In static walking gait, as depicted in Figure 2.22(a), the robot Center of Mass (CoM) is always remain in the Support Convex Polygon (SCP) when the robot moves its swing leg, and this usually applies to slow walking motion. Static walking motion has better stability because the CoM is always stays inside the feet/foot SCP, and the structure inertial momentum does not cause too much disturbance during walking. The location of the ZMP is very close or at the same point to the CoM's ground projection point.

2.9.2 Dynamic Gait

In the dynamic walking gait, as depicted in Figure 2.22(b), the ground projection CoM point may not necessary always be located within the foot support polygon. When the walking becomes faster, the CoM will be located near to the boundary of the SCP. As such, the biped requires a faster legs motion speed to recover from an unstable state. Although such motion gait is smoother and it resembles a more human-like walking gait, due to the inertial and momentum effect, it may produce jerky motion and gives rise to stability problems. This is a challenge for bipedal robotic researchers.

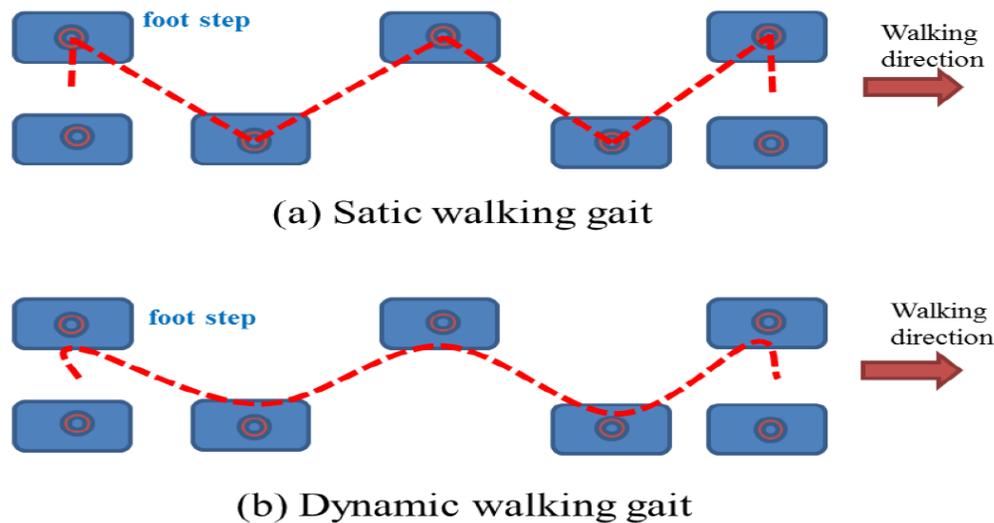


Figure 2.22. Trajectory of robot locomotion, the red dotted line is the CoM trajectory.

2.10 Trajectory Generation and Control

There are three main methods that can be used to compute the desire trajectories: capturing human motion, rhythmic based and purely computer generated.

2.10.1 Human Gait Motion Capture and Playback

A typical motion capture method [http://www.frontiernet.net/~imaging/gait_model.html/] is as shown in Figure 2.23, the Honda humanoid robot P2 and P3 uses this technique. The walking motion is achieved by playing back the trajectories recorded from human walking on different terrains. These trajectories are then modified to suit the individual robot characteristics. Although this method is able to achieve a stable walking gait but often yields unnatural motions [Q. Yuan,

I.M. Chen, 2013]. This is due to the discrepancy in gait mapping between a human walking and a robot walking. Furthermore, such reverse-engineering technique is not portable and is computational intensive.



Figure 2.23. Human walking gait data recording technique.

2.10.3 Biologically Inspired Rhythmic Manipulation

Recently, rhythmic based walking control capable of generating periodic motions such as the Central Pattern Generators (CPG) is gaining attention. A typical approach with the CPG is the composition of coupled nonlinear equations to achieve a stable limit cycle walking pattern that generate trajectory signals. The CPG can generate effective walking gaits pattern by adjusting the walking cycle rhythm from sensory feedback information and mimicking virtual neurons to generate signals to learn to improve on the walking gaits (hypothesis that how babies first learn to walk). Based on the results of the many studies on humans walking do exhibit some form of CPG-like structure, CPG are suitable for bipedal locomotion because it has the ability to generate rhythmic gaits patterns. [A. Herali, K. Wolff et al., 2007], [M.A. Carla, J.A. Tenreiro Machado et al., 2010], [A.G.Baydin, 2012].

2.10.4 Computer Generated Trajectory

A typical model-based control algorithm is depended on the mathematical model of the bipedal robot. Computer based control techniques are popular due to it intelligent and high speed algorithms solving capabilities. The most common control strategy is to track and confine the Zero-Moment-Point (ZMP) to remain within the footprint of the Support Convex Polygon (SCP) to achieve stable walking. This technique was first introduced by Vukobratovic [M. Vukobratovic, B. Borovac, 2004]. There are numerous control approaches that are in use for trajectory modeling, for example, Genetic Algorithms, Neural Networks, Fuzzy Logic and

hybrid techniques such as fuzzy-genetic and neuro-fuzzy algorithms. Each of the proposed approaches has its own advantages and limitations and will not be discussed here.

2.10.5 Direct Waveform Walking Pattern Generation

Bipedal robot walking sequences can be seen as a periodic phenomenon composed of various phases. The common approach for walking planning is to execute it phase by phase. This strategy will allow the biped to change its gait patterns at the end of each walking phase when there are changes in environmental conditions. The basic concept of a walking gait pattern generator is to generate the CoM trajectory and the swing foot trajectory for the robot. For a stable walking posture, the ZMP of the robot always stays within the SCP. Figure 2.24(c) shows a typical walking footprint and the ZMP zigzag reference trajectory of a bipedal robot.

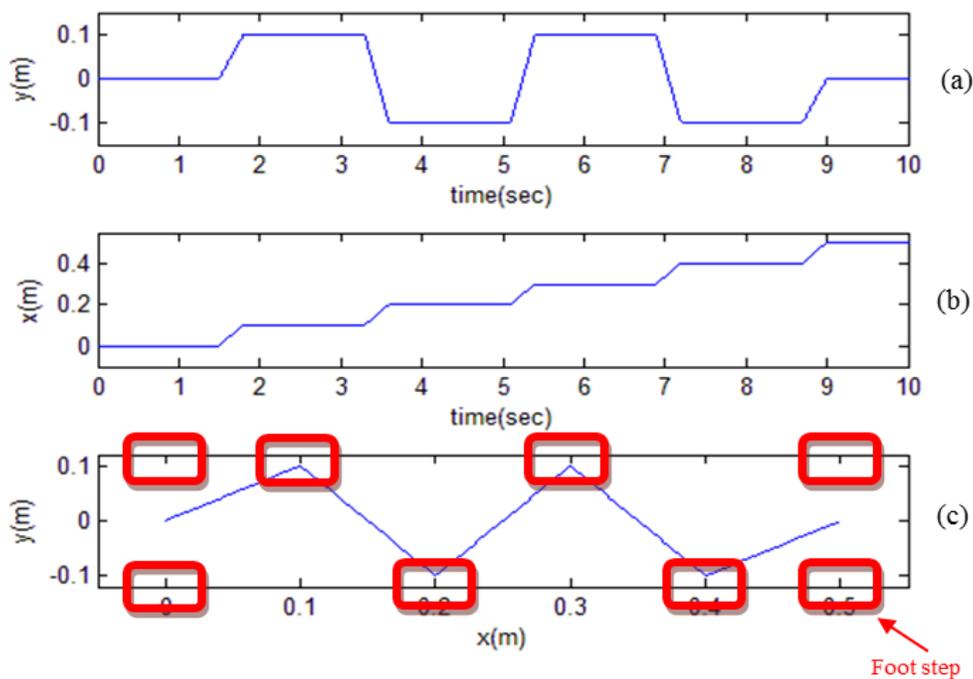


Figure 2.24. Typical ZMP trajectories, the red boxes are the robot's foot step.

The CoM waveform may be produced through piece-wise connection using polynomial functions, cyclic function generators or tracking based intelligent pattern generation algorithm. Some of the common trajectory-based paths found in literature are:

$$\begin{aligned}
\text{Cubic path:} & \quad q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 . \\
\text{Polynomial path:} & \quad q(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n . \\
\text{Harmonic path:} & \quad q(t) = a_0 + a_1 \cos a_2 t + a_3 \sin a_2 t . \\
\text{Cycloid path:} & \quad q(t) = q_0 + \frac{q_f - q_0}{\pi} \left(\frac{\pi(t-t_0)}{t_f - t_0} - \frac{1}{2} \sin \left(\frac{2\pi(t-t_0)}{t_f - t_0} \right) \right) .
\end{aligned} \tag{2.16}$$

The cycloid and the harmonic functions are most commonly used due to their simple expressions; however, the disadvantage is their nonlinearity. The cubic path function can provide a good trajectory motion, however, any increase in the end point constraints, such as low jerk or zero acceleration, higher degree of polynomials are needed.

2.10.6 ZMP Tracking Gait Pattern Generation

Although there are various proposed methods found in literature, but the two most widely applied methods are the LQR [T. Nguyen, W.K. Dae et al., 2008] and the preview control approach [S. Kajita, F. Kahehiro et al., 2003].

2.10.6.1 State Space Model

If a constraint (z-axis) control is applied to an inverted pendulum as discussed in previous section, such that the mass can only move horizontally with constant height along the z-plane, we obtain a simplified Linear Inverted Pendulum Model (LIPM). Please refer to Appendix B for details description. The ZMP equation in the x-axis (sagittal plane) can be derived as

$$x_{zmp} = x_{com} - \frac{z_{com}}{g} \ddot{x}_{com} . \tag{2.17}$$

The LIPM equation(2.17) can be re-arranged into the ZMP *state space* form by defining a new variable, the jerk ($\ddot{\ddot{x}}$) of the LIPM in the x-axis, to be the control input u , the ZMP equations can now be expressed in state-space form

$$u_x = \ddot{\ddot{x}}_{com} ,$$

$$\frac{d}{dt} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x,$$

$$x_{zmp} = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix}. \quad (2.18)$$

2.10.6.2 LQR Control Method

Refer to Figure 2.25. A full state LQR with observer feedback tracking system is applied to perform the ZMP tracking control. The parameters (Q, R, K_r) need to be tuned and optimized during the tracking process. The sample trajectories of the ZMP and CoM waveforms tracking are as shown in Figure 2.25. The controller is able to track the reference waveform with some overshoot.

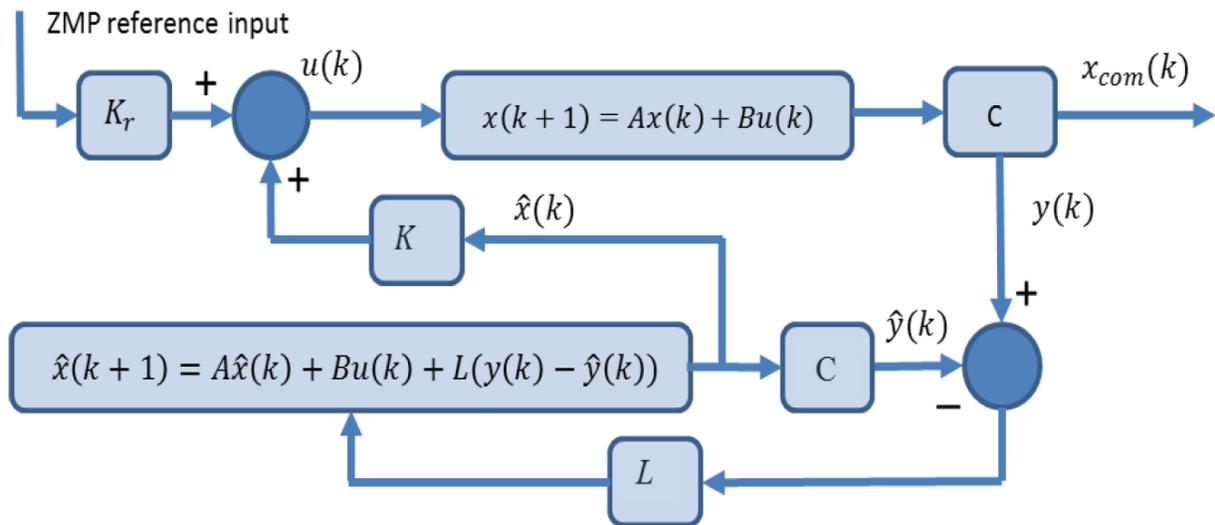


Figure 2.25. LQR with observer waveform generation block diagram.

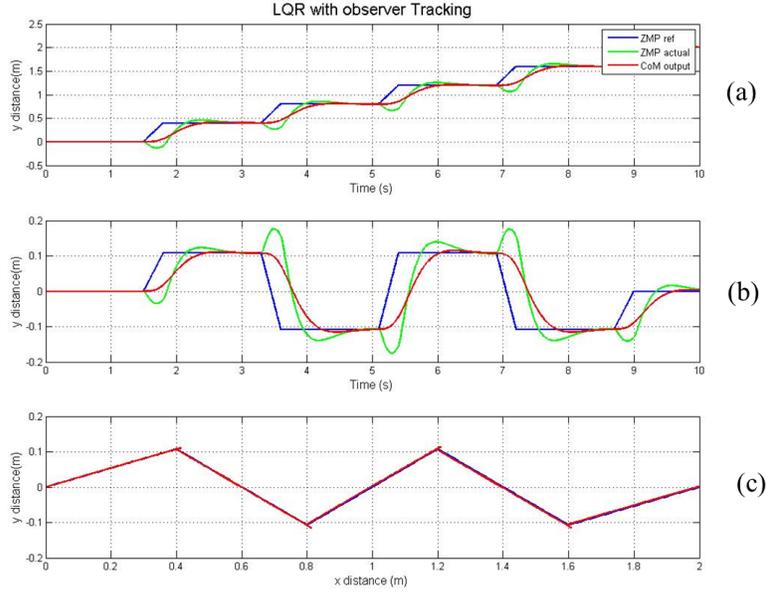


Figure 2.26. LQR tracking waveforms.

2.10.6.3 Preview Control Method

This method was first proposed by Kajita [S. Kajita, F. Kahehiro et al., 2003], as depicted in Figure 2.27, using the techniques mentioned in Thomas [T.B. Sheridan, 1966] *preview control* concept. By applying equation (2.19), defining a cost function and solving the Riccati equation, an optimal controller may be designed. The performance index is specified as

$$J = \sum_{j=k}^{\infty} \{ \mathbf{Q}_e \mathbf{e}(i)^2 + \Delta \mathbf{x}^T(i) \mathbf{Q}_x \Delta \mathbf{x}(i) + R \Delta \mathbf{u}^2(i) \}, \quad (2.19)$$

where $\Delta \mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$ denote the state vector, $\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$ is the control input, and $\mathbf{Q}_e, R > 0$ and \mathbf{Q}_x is a 3×3 symmetric non-negative definite matrix. The preview controller which minimizes the cost function index with preview for future N_L steps is determined by

$$\mathbf{u}(k) = -\mathbf{G}_i \sum_{i=0}^k \mathbf{e}(i) - \mathbf{G}_x \mathbf{x}(k) - \sum_{j=1}^{N_L} \mathbf{G}_p(j) p^{ref}(k+j), \quad (2.20)$$

where \mathbf{G}_i is the error gain, and \mathbf{G}_x is the state feedback gain. Note that $\mathbf{G}_p(j)$ is the preview gain introduce into the control feedback loop to overcome the overshoot, as depicted in Figure 2.28.

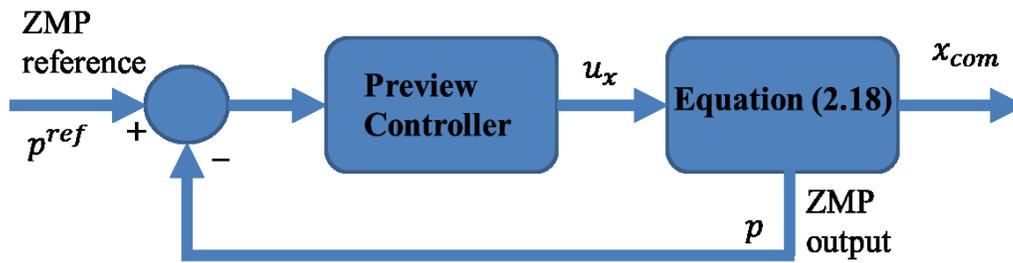


Figure 2.27. Preview controller block diagram.

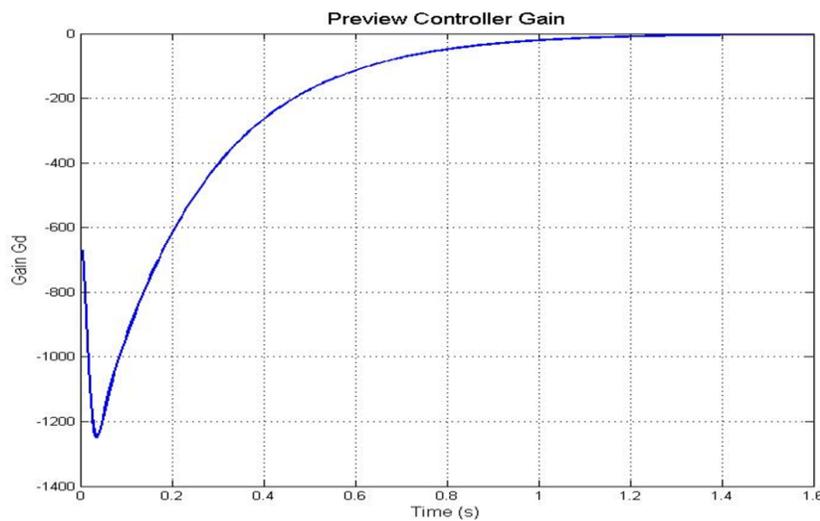


Figure 2.28. Preview Gain G_p .

2.11 Stabilizer

In general, bipedal robot is not a perfect rigid mechanism, and the ground condition is on a even flat ground. Due to modeling errors, uneven ground conditions and disturbances from surroundings, the robot will not walk steadily without a stabilization controller. In most situations, based on sensors information, a real-time feedback control with build-in stabilization algorithms is used to modify the biped gait trajectories to achieve stable walking, this is illustrated in Figure 2.29. Such stabilizers were discussed extensively in [J. Yamaguchi, A. Takanishi et al., 1995], [K. Hirai, M. Hirose, Y. Haikawa et al., 1998], [K. Hashimoto, Y. Sugahara et al., 2006], [Y. Choi, D. Kim et al.,2007].

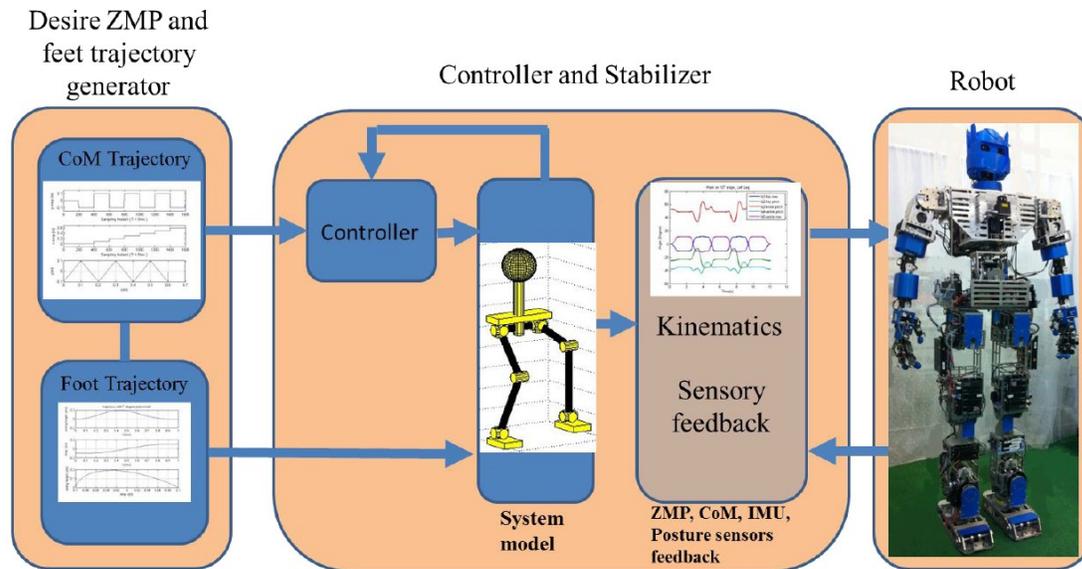


Figure 2.29. A typical feedback and stabilizer control block diagram.

2.12 Chapter Summary

This chapter summarised state of the art engineering research and developments of humanoid robotics. Humanoid robotics addresses the emerging field of intelligent systems equipped artificial cognitive skills and is becoming popular in recent years; anthropomorphic bipedal robots are now studied by a variety of scientific disciplines such as biomechanics, mechatronics, machine learning and biomedical engineering. There are a multitude of institutional teams and industry researchers creating and studying the bipedal robots behaviors worldwide. With the ageing population, the needs in terms of elderly-care robots and service robots with innovating cognitive technologies and artificial intelligence created a professional application arena for humanoid robotics.

Chapter 3

Mechanical and System Architectures

A life-size humanoid robot is complex and expensive from we see in the market till now. To take a step ahead in this area of research, it is obvious to seriously look into the robotic complexity and the cost effectiveness. Therefore, in this research, an important challenge is to design and develop a comparatively low cost mid-size humanoid bipedal that has a simple anthropomorphic mechanical structure. At the same time, we need to ensure that this robot platform is suitable for use in both engineering research and educational learning. Refer to Figure 3.1on the proposed targeted sector. This chapter gives an overview on the approach taken on design of the humanoid bipedal robot TPinokio. The robot is designed and developed over the last three years as an experimental and educational platform for studying of robot walking behavior. The design concept of the robot mechanical structure, the modular joint mechanism, the passive-toe and the control electronics architectures are presented in subsequent sections.

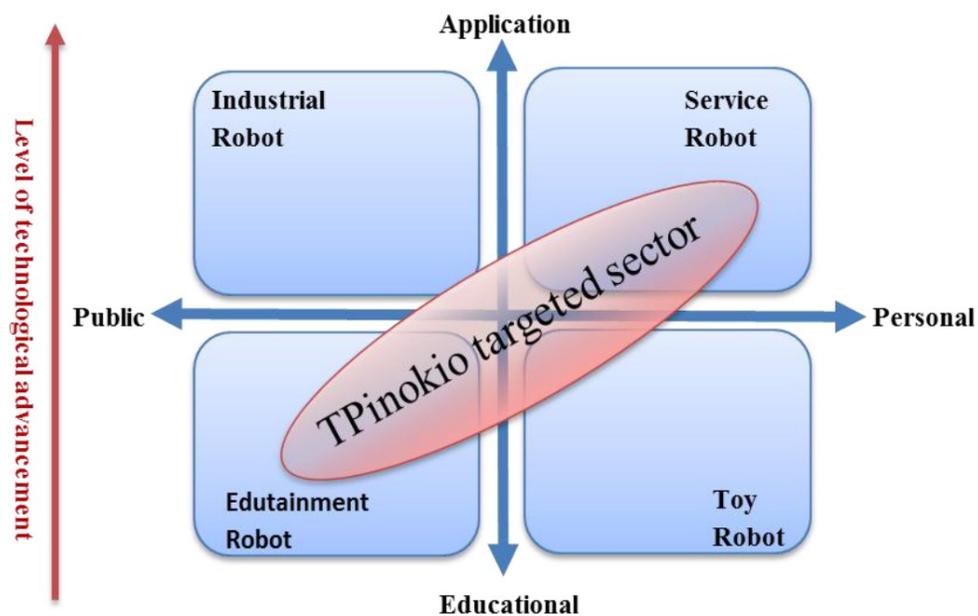


Figure 3.1. TPinokio Targeted robotic market sector.

3.1 Mechanical Structure

TPinokio is a mid-size robot with height approximately 1.6 meter and weighs approximately 58kg as depicted in Figure 3.2. A compact and modular-joint mechanical structure has been successfully implemented on this robot. Another special feature of this robot is that it has feet with simple passive-toe mechanism, and there is a ‘feet contact forces’ feedback. These feedback mechanisms are the load cells mounted on the foot sole and a 6-axis force-torque sensor mounted at the joint ankle. Some of the key design concepts and features of TPinokio are:

- A mid-size human-like body with 55 Degree of Freedom (DoF) revolute actuators.
- A modular-joint design which will lead to the robot to have simple kinematics and the body structure can then be easily modeled as a multiple point- mass system.
- A set of uncomplicated kinematics and dynamics models of the robot that are simpler to implement mathematically and also are easier to simulate using MATLAB software programs.

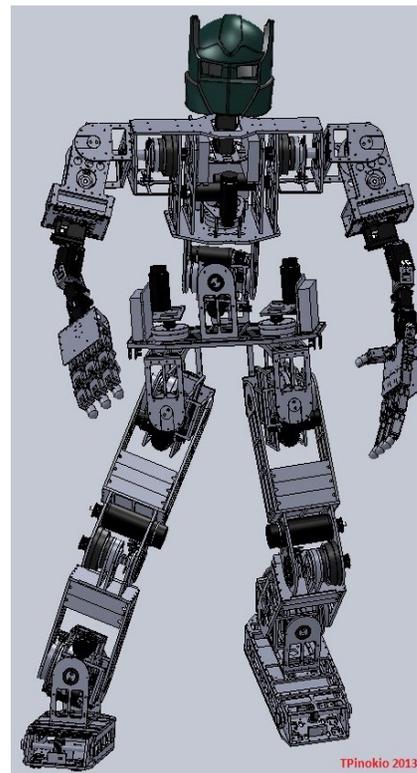
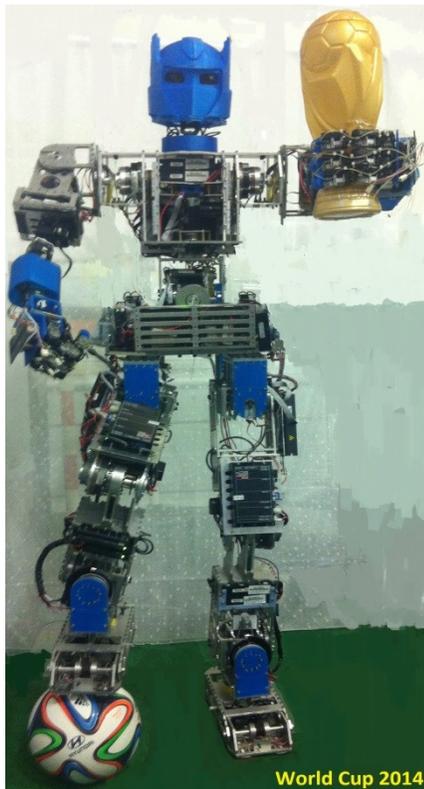


Figure 3.2. Mechanical structure of TPinokio.

3.1.1 TPinokio Design Concept

As discussed in previous section, the primary features in the design of TPinokio are to be a low cost, modular and light weight robot. The rationale in pinning to these considerations is to enhance the robot, that is under research, to be as commercially viable as much as possible and also be useful to mankind in a sense that it is convenient and practical to use the robot, and not just an ‘ideal’ robot but not practical to use due to size and complexity. The ‘practical to use’ feature stands out well especially with its flexible modularity and portability design concept. Upon successful completion of the robot with these features, it is no doubt that the outcome of the research works will be useful and widely used by all. Following are the salient features of TPinokio robot:

- Comparatively low cost: All components selected were carefully evaluated prior to procurement. Amongst the Bill of Material (BoM) for the robot, the most expensive component is the harmonic gear system.
- For Research: The modular design of TPinokio provides a good test-bedding platform for researcher, it can be easily modeled as multi-body point-mass system, and this is a common assumption adopted by most literatures.
- For Education: A wide range of low cost sensors and electronic gadgets may be added to the robot, with the user friendly LabVIEW interface, many interesting and interactive educational software can be easily developed for beginners.
- Simple mechanism: The mechanical structure is designed for easy maintenance and system modeling.

The passive toe-feet design is to achieve a more natural walking posture and to achieve a higher step clearance as depicted in Figure 3.3 and Figure 3.4.

The maximum step the robot can reach with and without toe-feet is as listed in Table 3.1.

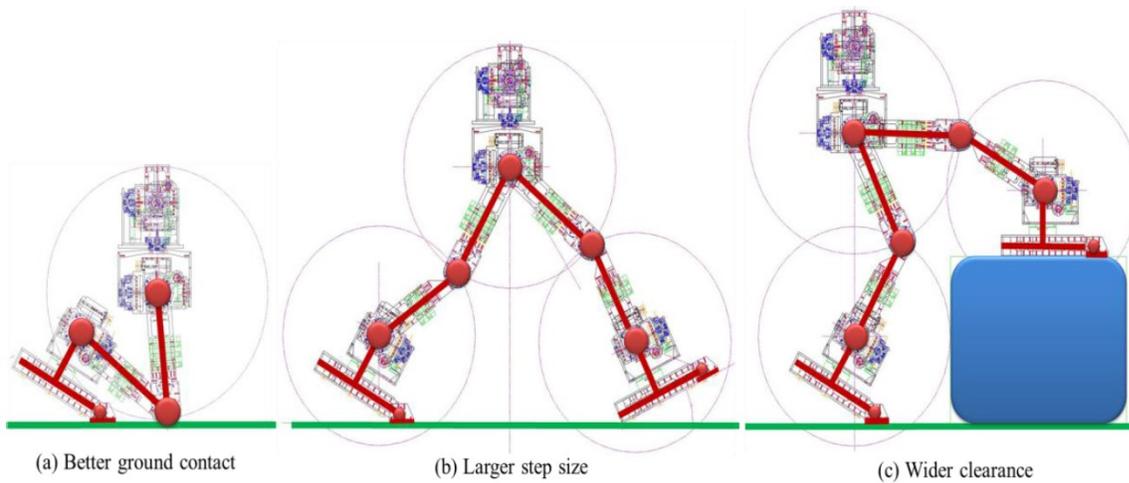


Figure 3.3. Advantages of having toe-feet design.

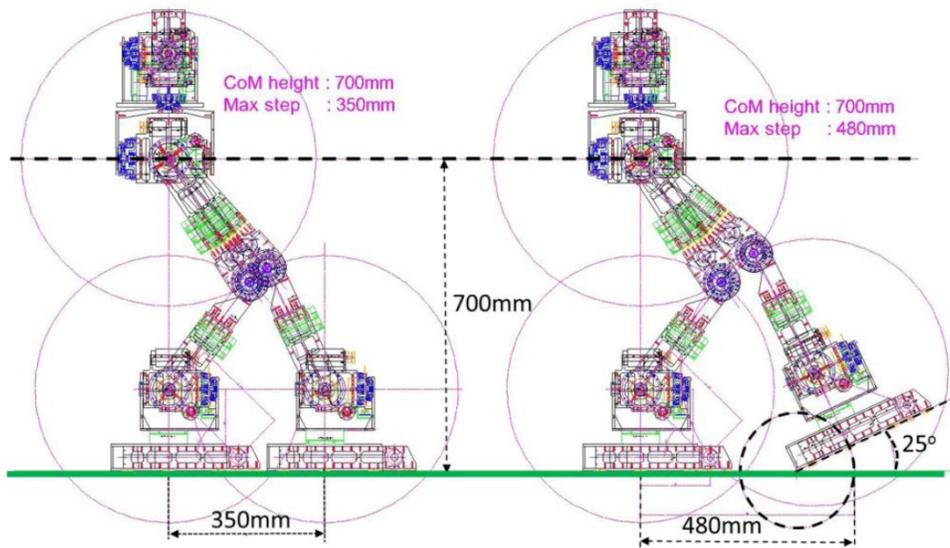


Figure 3.4. TPinokio robot step size.

Table 3.1. Maximum step size.

Pelvis (CoM) height	Maximum step size with flat-feet	Maximum step size with toe-feet
700mm	350mm	480mm
600mm	450mm	570mm

3.1.2 Other Design Considerations

Other general considerations are:

- Real-time operation: The communication protocol should be real-time with high update rate for the controller.
- Power requirement: The energy consumption should be optimized.
- Environmental safety: The walking and drives mechanism must not damage the floor or ground surface.
- Modular: The mechanical structure should be in modular form, to allow for easy maintenance, testing and upgrades.
- Size and Weight: The weight and size of the robot should be manageable for lab and field trial testing.

3.1.3 Configuration and Degree of Freedom (DoF)

TPinokio size and structure was carefully designed to allow the robot to maneuver safely in a normal human daily life living environment space and also appears user-friendly as illustrated in Figure 3.5.

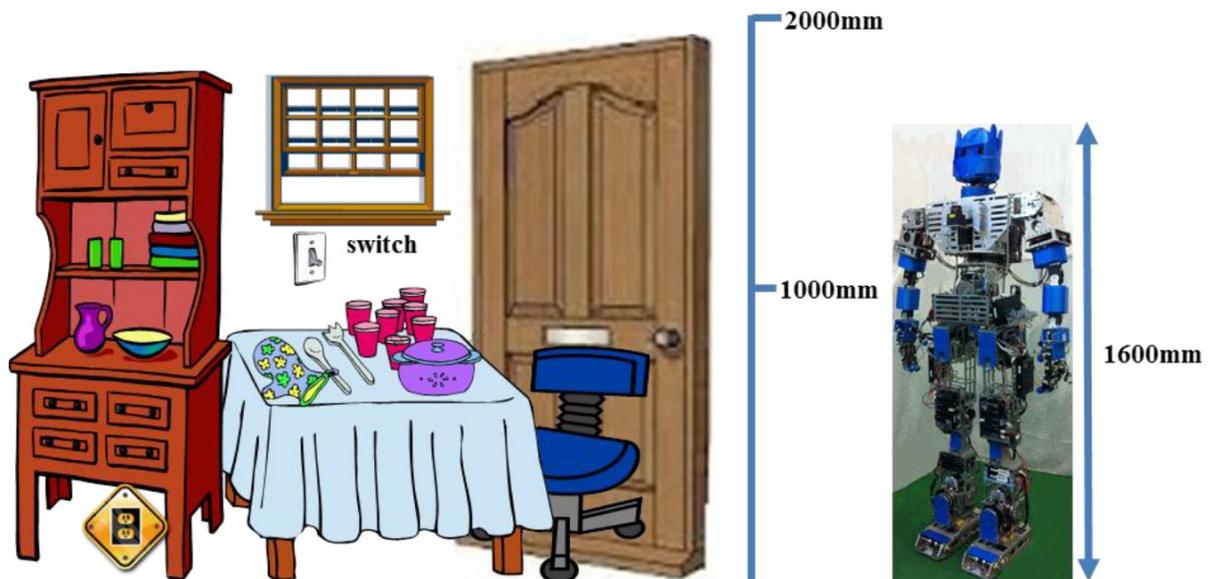


Figure 3.5. Human living environment and space.

When designing the robot, the author took into consideration of the work space of the robot's arm and hand so that the robot's hand can reach for light switches, table top and other objects in the human living indoor environment. To assimilate a mid-size human body structure, the shoulder height of the TPInokio is approximately 1200 mm from the ground. This shoulder height will allow the robot to reach to objects at a height of 1700mm (for example, items from the kitchen cupboard) with the arm extended. The robot's legs are 800mm long so that it can easily clear a step height of 50(w) x 100(H) mm (for example curb in some houses). In doing so, the robot can easily climb stairs and to perform motion maneuver similar to human being. The design also allows the robot to pass through doorways and corridors. The actuator motors and gears are selected based on the maximum possible torque. The power required by each joints, the general specifications and key components of TPInokio are outlines in Figure 3.6. The movable ranges of the joints are shown in Figure 3.7.

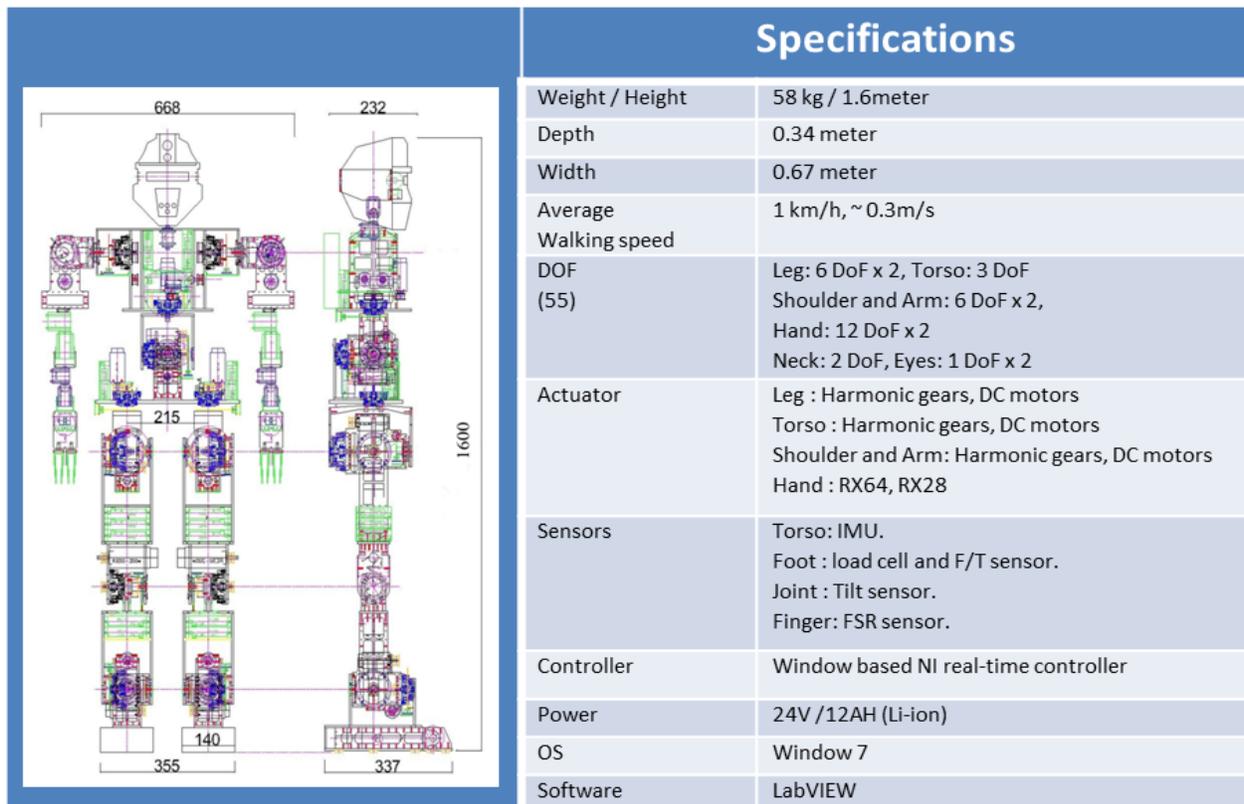
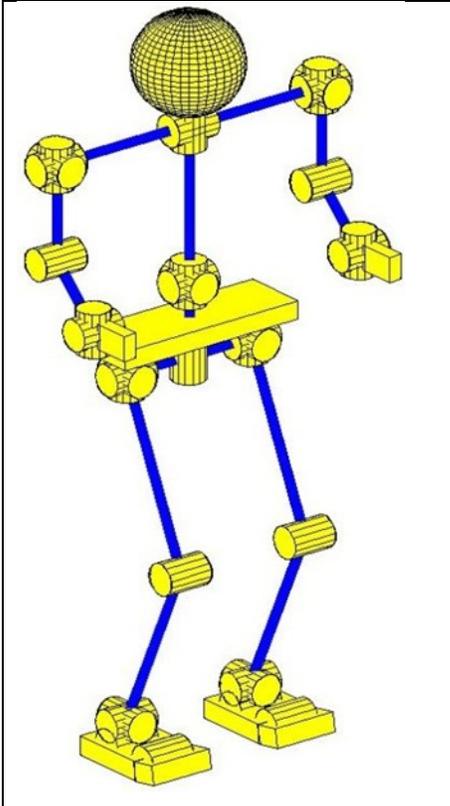


Figure 3.6. General specification of the TPInokio robot.



Body segment	Joint	Angle Ranges (similar to a Human)
Neck	Pitch Yaw	-30° to 90° -80° to 80°
Shoulder	Roll Pitch Yaw	-90° to 90° -120° to 120° -90° to 90°
Elbow	Pitch	0° to 130°
Wrist	Pitch Yaw	-30° to +30° -45° to +45°
Waist	Roll Pitch Yaw	-50° to 50° -20° to 70° -70° to 70°
Hip	Roll Pitch Yaw	-60° to 45° -130° to 45° -90° to 60°
Knee	Pitch	0° to 150°
Ankle	Roll Pitch	-20° to 30° -30° to 80°
Passive toe	Pitch	0 to 45°

Figure 3.7. Movable ranges of the TPinokio robot.

3.2 Hardware Architecture

The hardware architecture of the TPinokio is as illustrated in Figure 3.8. The overall control and hardware system configuration is shown in Figure 3.9. The robot is designed with scalability, modularity and standardized interfaces. The main communication protocol interface between the main controller and the individual modules are based on the CANbus and USB. CANBus was chosen as the main communication protocol because it provides high speed, good noise immunity and good fault tolerance. The main controller chosen is a light weight 10" note book computer because of its light weight, small size, reasonable battery life, and easy window OS programming environment. It also has great variety of extension through USB interface, for instance, digital and analog I/O cards. The main controller tasks are to calculate the gait trajectory patterns, synchronize the motor motion, updates of sensory data and control the joint position. It also provides data communication between joystick command and the base station through wireless WLAN.

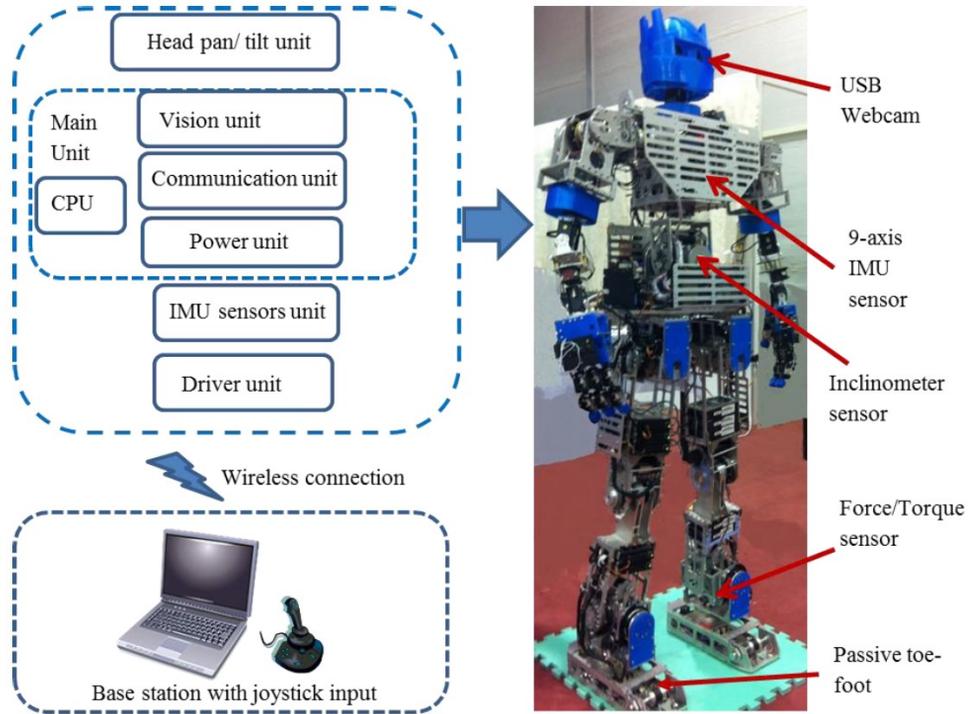


Figure 3.8. Overall control of the TPinokio robot.

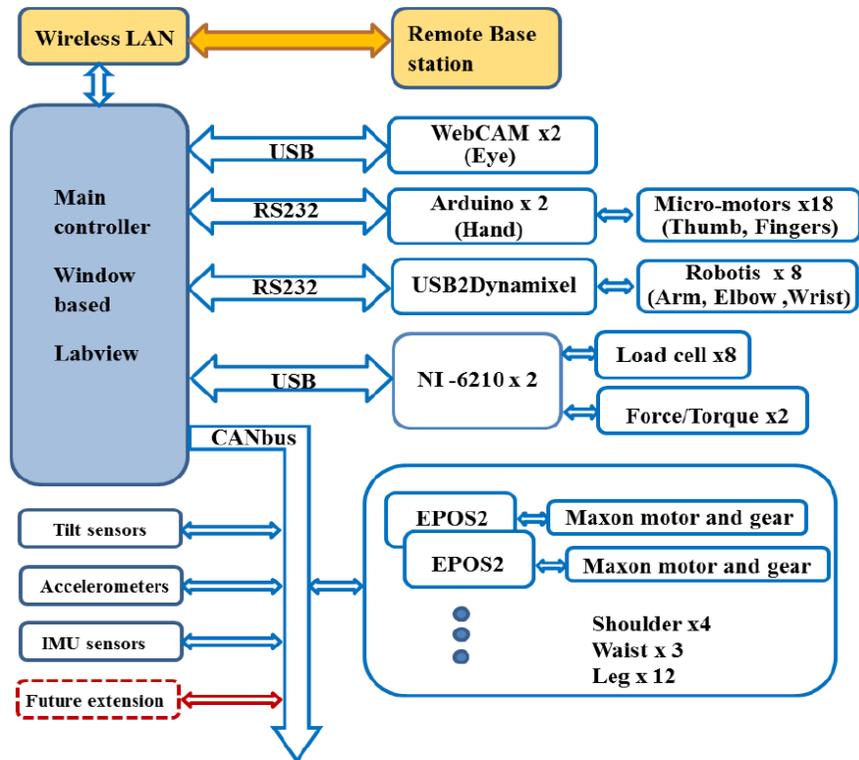


Figure 3.9. Hardware System Configurations.

3.2.1 Motion Control Structure

The motion control system includes the subsystem controllers, joint servo, motor drivers, angle encoders and the homing sensors. Eight load cell sensors are mounted on each corner edge of the foot-print to provide contact forces feedback on the ground surface, and a better accuracy 6-axis F/T sensor is mounted on the ankle-foot joint to provide active forces feedback. There is one 9-axis IMU sensor located on the robot waist to provide posture/balancing feedback. Each rotatory joint is control by a 150w DC motor fitted with harmonic gear and pulley-belt. The DC motors are controlled by EPOS2, and the EPOS2 is located near the modular joint housing, this is to eliminate long wiring and signals interference. The main features of the EPOS2 controller is compact, lightweight, high power output with built-in CANbus.

3.3 Software Architecture

The system software comprises of three main Modules. The first module is for vision and image processing. The second module is for walking pattern planning and gait generation. The third module is for lower level PID motion control.

3.3.1 Computer System Structure

The overall control system of the TPinkio comprises of the sensory modules and the motion control modules. The operating system is PC based Windows 7 and the programming software is LabVIEW. The communication between the main controller and sub-modules is through CANbus protocol at 1Mbps. The system can also be controlled through wireless LAN Ethernet 100Base-T by the mobile base station.

3.3.2 Graphic User Interface

The GUI is written in NI LabVIEW, as shown in Figure 3.10. The NI LabVIEW is user-friendly and is an effective programming platform for both researchers and students. The price of the NI LabVIEW software is reasonable and affordable for educational purposes.

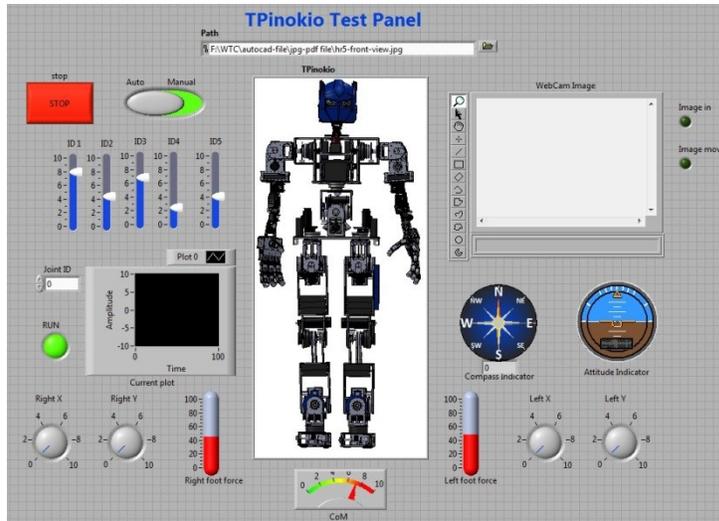


Figure 3.10. TPinokio GUI with LabVIEW.

3.4 Modular-Joint Design Concept

The joint shaft of the TPinokio is driven by DC motor fitted with harmonic drive and pulley-belt. As shown in Figure 3.11, the encoder sensors, controller electronics and wiring harness are housed in a compact housing. Most of the parts and components are standardized with 80% of similarity in specifications, except for the motors and the harmonic gears. The motors and harmonic gears have different specification due to the requirement for different torques at different joints. As depicted in Figure 3.12, with such a design concept, the modular-joint can be easy swapped, reconfigured, or dismantled for maintenance.

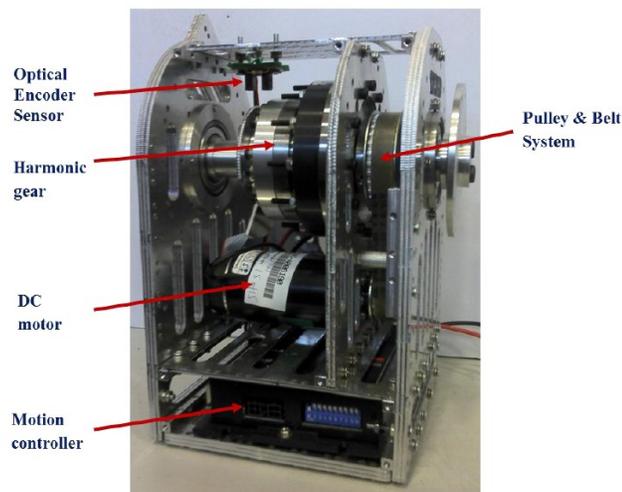


Figure 3.11. Compact modular housing.

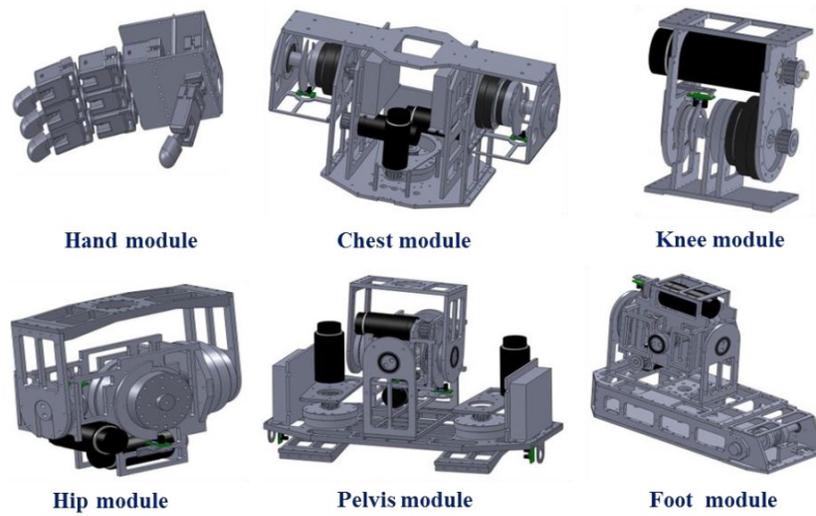


Figure 3.12. Compact modular-joint parts.

The connections between each compact module are depicted in Figure 3.13. The communication between joints is through CANbus interface with a single five core cable, three wires for the CANbus protocol, and two wires for the power supply. Such a design greatly eliminated the messy cabling that is found in many existing robots.

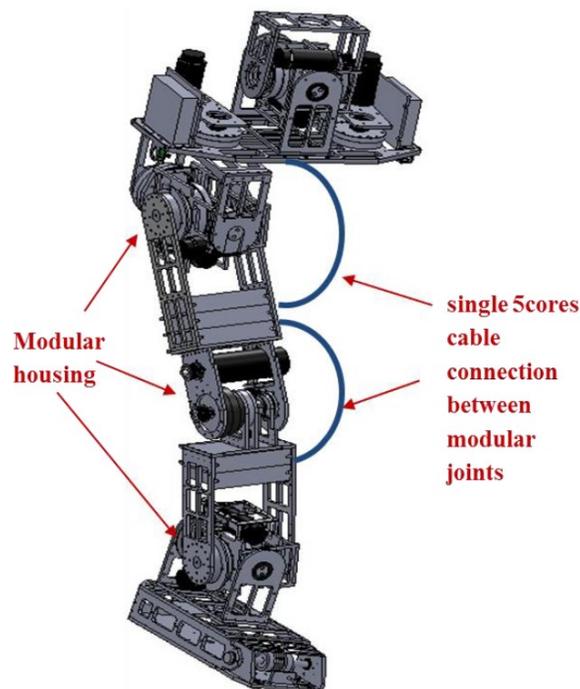


Figure 3.13. Simple cable connections between each compact module.

3.4.1 Feet with Passive-Toe and Force Sensor

The passive-toe design is shown in Figure 3.14. In the design, there are 8 load cells, the toe joint torsion spring constant is designed with 30Nm/rad, and the rotating angle of the passive toe joint is limited to 45°. The leg mechanism is designed to cater for ‘stretched walking’ and jumping. Two torsion springs are included in the design to increase the spring constant when there are running and bouncing activities. Incorporating the toe-feet design into the robot design, we can achieve a faster walking gait motion, a smoother climbing motion, and also gives a lower bending posture with good ground contact from the robot.

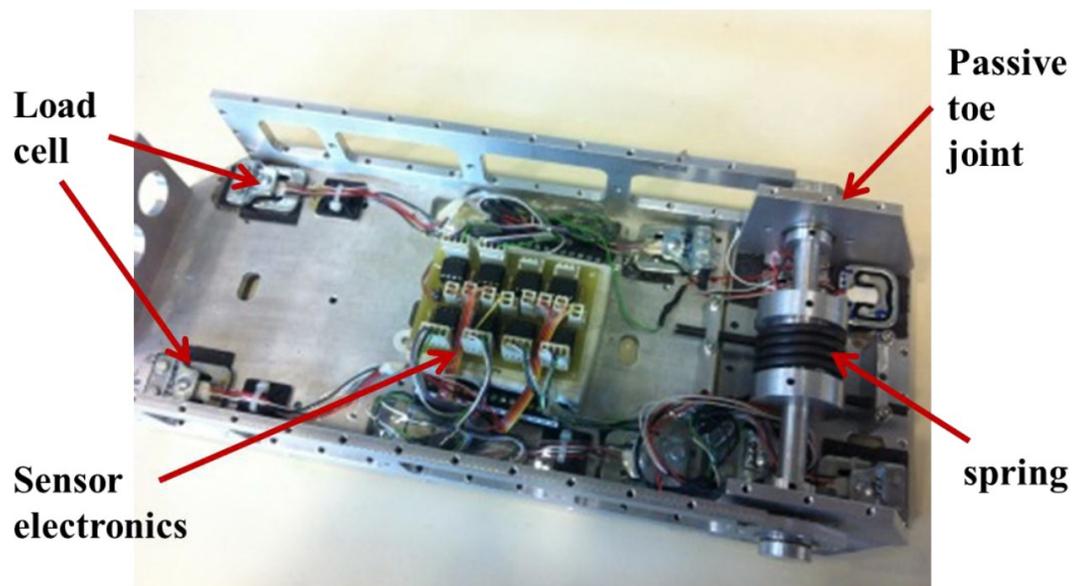


Figure 3.14. Foot mechanism with passive toe.

3.4.2 Foot force Sensor and Measurement

3.4.2.1 Load Cell Sensor

Every foot sole and the toes are fitted with four load-cell sensors to measure the ground reaction forces, a total of eight load-cell sensors (matching pairs, from commercially available digital weighing machine) per foot. A full bridge circuit is as shown in Figure 3.15. The maximum output value is around 4.5v.

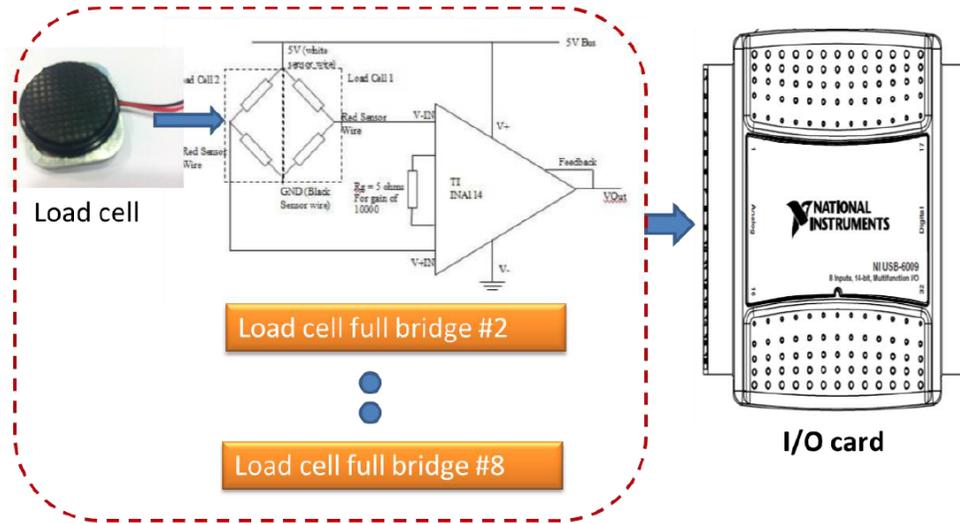


Figure 3.15. Full bridge load cell sensing circuit.

During the SSP or DSP phase, the Zero-Moment-Point (ZMP) can be calculated by measuring the individual ground contact reaction forces.

For single foot support, as illustrated in Figure 3.16, the ZMP location can be calculated as

$$x_{zmp} = \frac{\sum_{i=1}^8 p_{ix} f_i}{\sum_{i=1}^8 f_i} \quad , \quad y_{zmp} = \frac{\sum_{i=1}^8 p_{iy} f_i}{\sum_{i=1}^8 f_i} \quad . \quad (3.1)$$

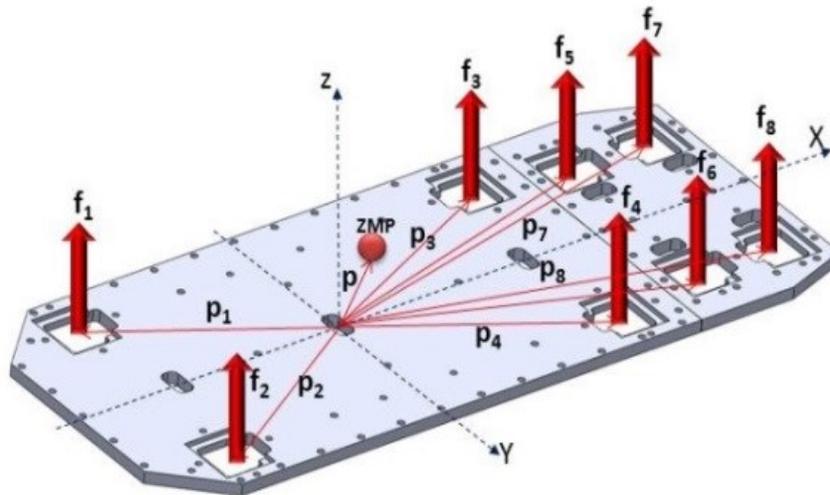


Figure 3.16. Single Support Phase (SSP) and the ZMP location.

For double feet support, as illustrated in Figure 3.17, the ZMP location can be calculated as

$$x_{zmp} = \frac{p_{Lx}f_{Lz} + p_{Rx}f_{Rz}}{f_{Lz} + f_{Rz}}, \quad y_{zmp} = \frac{p_{Ly}f_{Lz} + p_{Ry}f_{Rz}}{f_{Lz} + f_{Rz}}. \quad (3.2)$$

Where $f_R = [f_{Rx} \quad f_{Ry} \quad f_{Rz}]^T$, $f_L = [f_{Lx} \quad f_{Ly} \quad f_{Lz}]^T$,
and $P_R = [P_{Rx} \quad P_{Ry} \quad P_{Rz}]^T$, $P_L = [P_{Lx} \quad P_{Ly} \quad P_{Lz}]^T$.

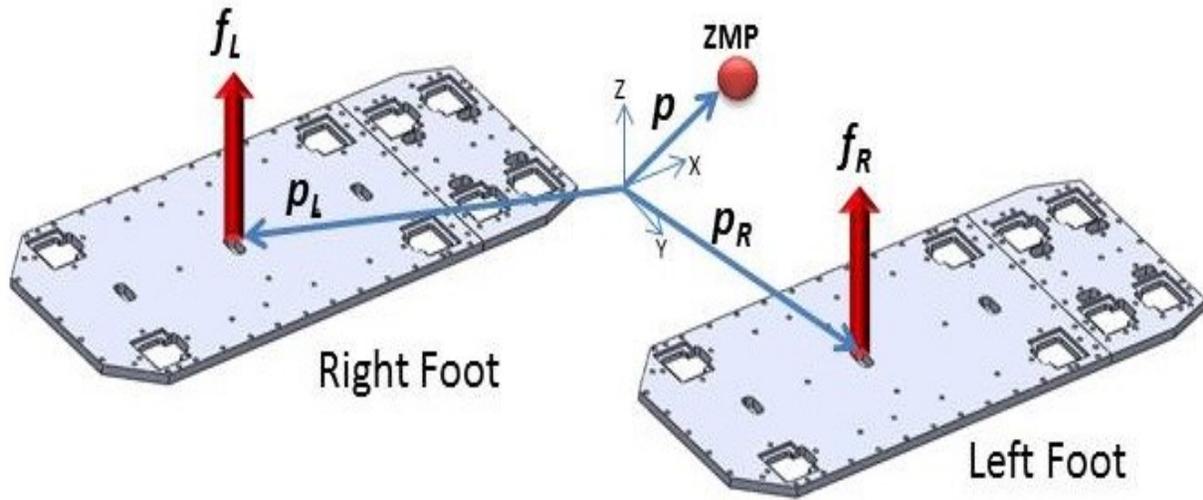


Figure 3.17. Double support phase (DSP) ZMP location.

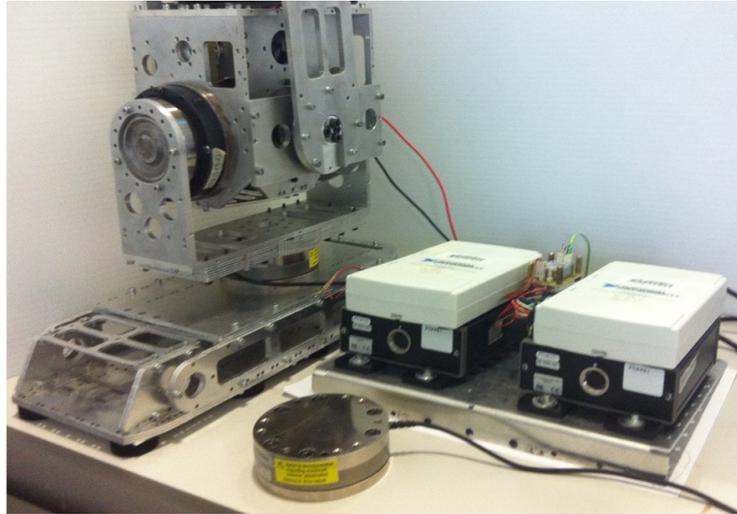
3.4.2.2 Six-axis Force Torque Sensor

A six-axis force torque sensors (FTS) as shown in Figure 3.18(a) are mounted at the robot ankle-foot joint to measure the forces and torque that are acting on the feet. The FTS equations to measure and estimate the ZMP location as illustrated in Figure 3.18(b) can be expressed as

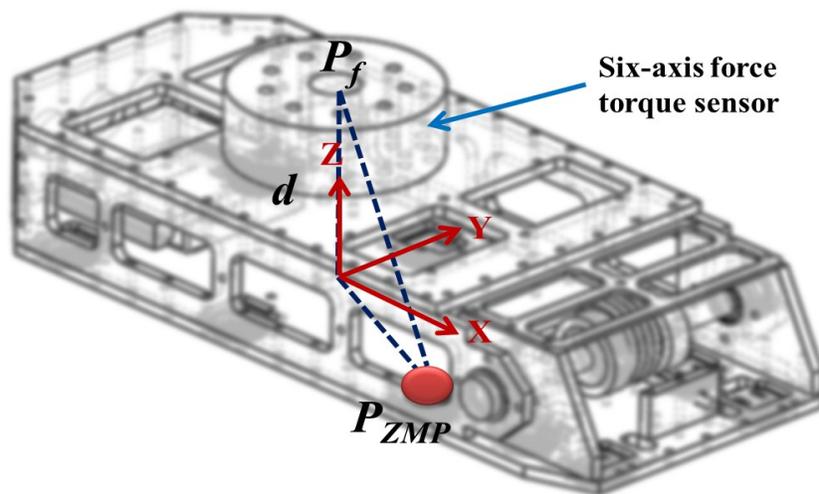
$$x_{zmp} = \frac{(-\tau_y - f_x d)}{f_z}, \quad y_{zmp} = \frac{(\tau_x - f_y d)}{f_z}. \quad (3.3)$$

where d is the distance of the sensor from the ground. f_x , f_y and f_z are the forces in the x-axis, y-axis and z-axis respectively. τ_x , τ_y are the torques about the x and y-axis respectively. The coordinate of all the points about the x-axis are

$$\begin{aligned}
 p_{zmp} &= [x_{zmp} \quad y_{zmp} \quad 0]^T, \\
 p_f &= [0 \quad 0 \quad d]^T.
 \end{aligned}
 \tag{3.4}$$



(a) Force/Torque sensor measurement setup.



(b) Using FTS to estimate ZMP location.

Figure 3.18. FTS measurement and estimation of ZMP location.

3.5 Force, Power and Stress Analysis of TPinokio

In this section, the maximum power required by each joint in order to perform basic movements, jumping and objects handling tasks will be derived. These values will help in the sizing of actuator power and gear ratio for each joint. Some assumptions were made to the modeling to simplify the calculation.

3.5.1 Jumping Stress Analysis

During jumping action, the whole body goes through four main phases of the actions. The key frames of the phases are presented in Figure 3.19.

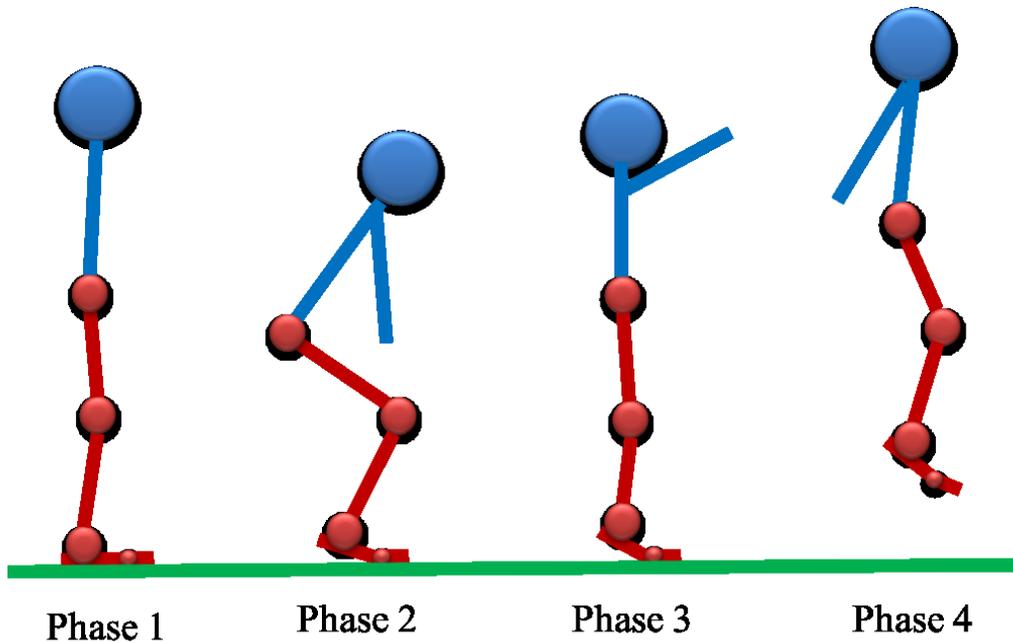


Figure 3.19. Jumping sequences key frames.

Refer to Figure 3.20, let the CoM of the robot is at the pelvis location point E, when the whole body is about to take off, the velocity of the CoM is v_c . If the jumping height is $H = 0.15m$, the jumping velocity v_c can be calculated with the equation $v_c^2 = 2gH$, where g is the gravity constant.

$$v_c = \sqrt{2gH} = 1.72 \text{ m/s.} \quad (3.5)$$

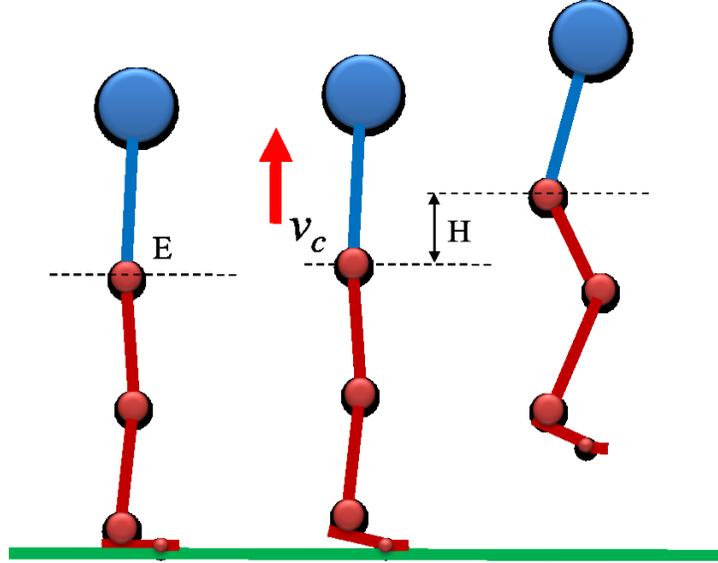


Figure 3.20. Jumping height and Velocity

For phase 2, as depicted in Figure 3.21, according to the biomechanical research in jumping literature, an efficient posture for jumping takeoff are at the angles $\theta_A, \theta_B, \theta_C$, and these values are $90^\circ, 100^\circ$ and 75° respectively. Assume from phase 2 to phase 4, the average action time is $\Delta t = 0.2s$, which is the time taken from the moment the robot bends its posture from standing position to the maximum angle to the time the toes leave the ground.

Let the maximum power required by the single hip pitch, knee pitch and ankle pitch actuators to perform jumping be P_h, P_k, P_a respectively, and $L_1 = 0.329m, L_2 = 0.25m, \theta_1 = 65^\circ, \theta_2 = 15^\circ$. Applying the following equations, the total power consumption can be calculated.

The bending height is given by

$$\Delta H = L_1(1 - \cos\theta_1) + L_2(1 - \cos\theta_2) = 0.198m . \quad (3.6)$$

The total energy required by both legs is

$$2(P_h + P_k + P_a) \cdot \Delta T = mg \cdot \Delta H + \frac{1}{2}mv_c^2 . \quad (3.7)$$

And the total power required by a single leg is

$$P_h + P_k + P_a = \frac{\left(mg \cdot \Delta H + \frac{1}{2}mV_c^2\right)}{2\Delta t} = \frac{58 \times 9.81 \times 0.198 + \frac{1}{2} \times 58 \times 1.17^2}{2 \times 0.2} = 380.89 w. \quad (3.8)$$

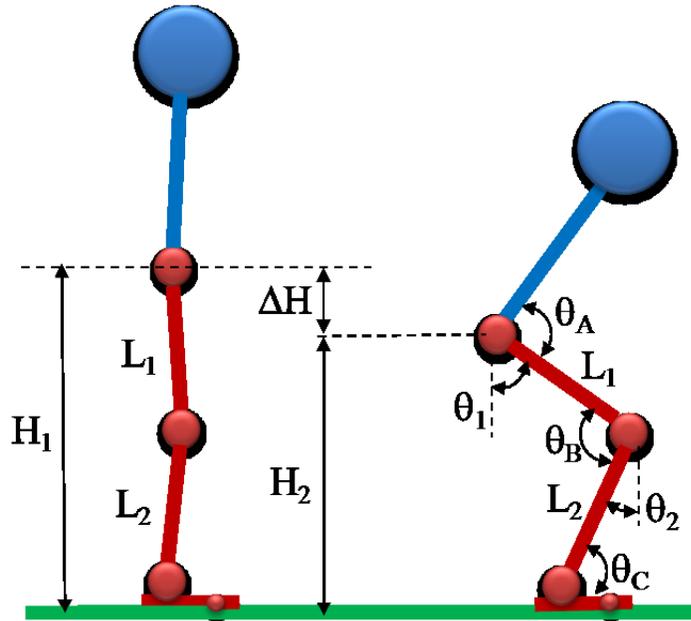


Figure 3.21. Robot body bending parameters.

By observing the human jumping posture, as illustrated in Figure 3.22(a), the ankle angular velocity changes from 0 at Phase 2 to a maximum value at the end of Phase 3, and the acceleration from maximum to 0 value, see Figure 3.22(b).

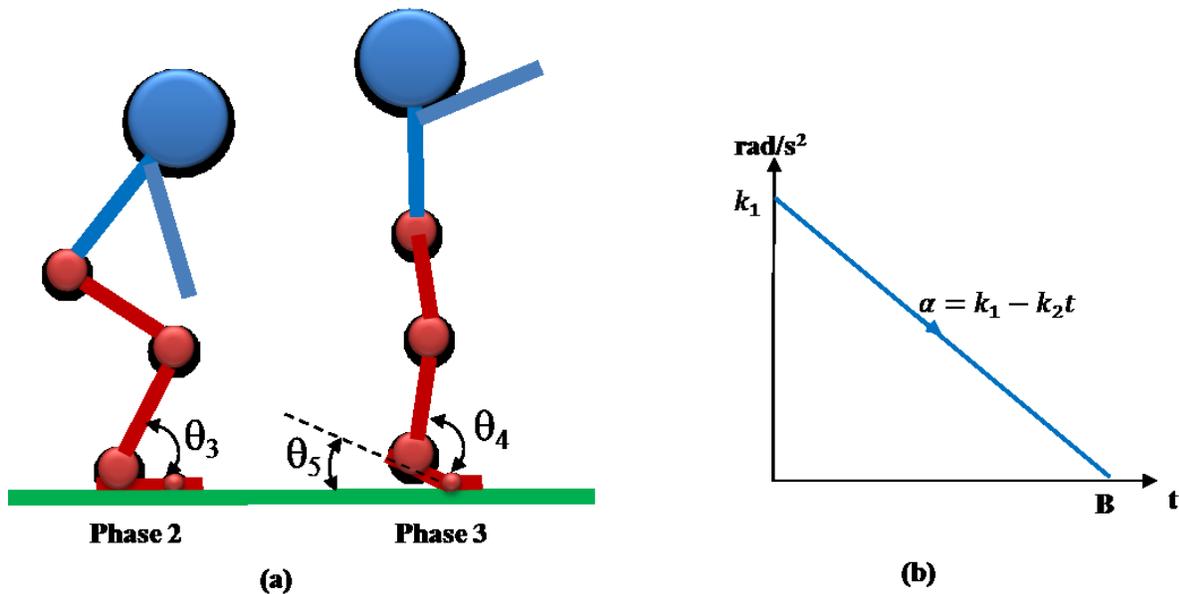


Figure 3.22. Acceleration and angle of the robot ankles.

The derivation is a complex process. We will simplify the procedure to one variable acceleration motion for easy calculation as follows:

Let the slope be $\alpha = k_1 - k_2 t$, and $t = 0.2s$, at point B

$$\alpha = k_1 - k_2 \times 0.2 = 0, \quad k_2 = 5k_1. \quad (3.9)$$

The angular velocity and the angle can be derived from (3.9) as

$$\omega = \int \alpha dt = k_1 t - \frac{1}{2} k_2 t^2. \quad (3.10)$$

For $t = 0.2s$,

$$\Delta\theta = \theta_4 - \theta_3 = \int_0^{0.2} \omega dt = \frac{0.04}{3} k_1. \quad (3.11)$$

Let set $\Delta\theta \approx 15^\circ$, from (3.9) and (3.10), we can derive

$$k_1 = 19.63, \quad \omega|_{t=0.2} = 1.963 \text{ rad/s}.$$

At point B, the maximum ankle torque for both ankles, where $l = 0.157m$ is the sole length, is

$$\cos\theta_5 = \frac{mgl}{2\tau_a}. \quad (3.12)$$

Let assume $\theta_5 \approx 35^\circ$, and the ankle torque can be determined by

$$\tau_a = \frac{Mgl}{2\cos\theta_5} = \frac{58 \times 9.81 \times 0.157}{2\cos35^\circ} = \frac{89.33}{1.638} = 54.53Nm, \quad (3.13)$$

$$P_a = \tau_a \cdot \omega = 54.53 \times 1.963 = 107.04w.$$

For the hip and knee, they both have the same starting velocity as well as the ending velocity, and the force is $f_h = f_k = (\alpha + g)m$, the average power ratio can be deduced as

$$\frac{P_h}{P_k} \approx \frac{f_h}{f_k} \approx \frac{m_{u_{body}}}{m_{u_{body}} + m_h} = \frac{20.3}{20.3 + 10.5} = 0.66. \quad (3.14)$$

From equation (3.8)

$$P_h + P_k = 380.89 - P_a = 380.89 - 107.04 = 273.84w. \quad (3.15)$$

And from the force ratio equation (3.14)

$$P_k = 164.96w, P_h = 108.87w. \quad (3.16)$$

When the body is in Phase 3, the angular velocities of hip pitch and knee pitch approach 0, same as end of Phase 2. Thus, the two joints go through an accelerated and decelerated motion. We can calculate the average angular velocity

$$\omega_h = \frac{\Delta\theta}{\Delta t} = \frac{\pi - \angle A}{\Delta t} = \frac{\pi - 0.5\pi}{0.2} = 7.85 \text{rad/s}, \quad (3.17)$$

and

$$\omega_k = \frac{\Delta\theta}{\Delta t} = \frac{\pi - \angle B}{\Delta t} = \frac{\pi - 0.55\pi}{0.2} = 6.98 \text{rad/s}. \quad (3.18)$$

For $P = \tau\omega$, the torque requirement for the knee and hip can be determined.

$$\tau_h = \frac{P_h}{\omega_h} = \frac{108.87}{7.85} = 13.87 \text{Nm},$$

and

$$\tau_k = \frac{P_k}{\omega_k} = \frac{164.96}{6.98} = 23.63 \text{Nm}.$$

(3.19)

3.5.2 Landing Stress Analysis

When the robot lands on the floor from the highest point, the posture of the body goes through four phases as depicted in Figure 3.23.

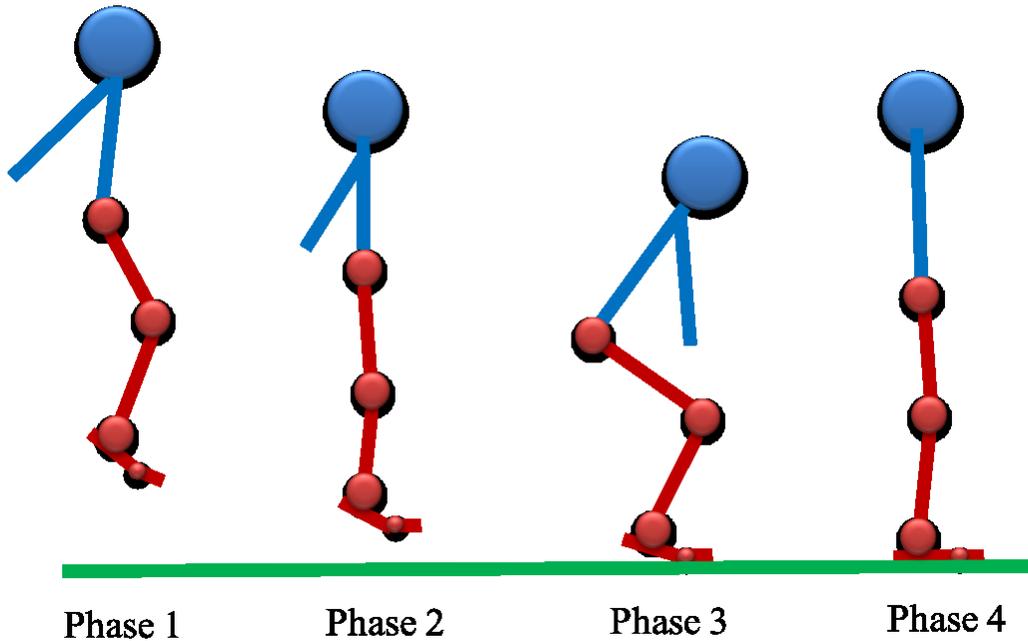


Figure 3.23. Phases of landing.

Assumed that the average impulse force when the robot landed is F_l , and the velocity of the body at the instant the toes just touching the floor is $v_d = 1.72 \text{m/s}$, we write the following equations

$$\frac{1}{2}mv_d^2 = mgH_J ,$$

$$(F_I - mg)\Delta t_d = mv_d . \quad (3.20)$$

Based on the observation from our in-house findings, the time taken is approximately $\Delta t_d \approx 0.25s$, which takes a little longer time than the jumping up case, and the impulse force exerted on both legs can be determined by

$$F_I = \frac{mv_d}{\Delta t_d} + mg = \frac{58 \times 1.72}{0.25} + 58 \times 9.81 \approx 968N . \quad (3.21)$$

Base on this calculated value and with a safety factor of 1.5, the feet sole shock absorber material and the toe-spring mechanism must be able to withstand an impulse force on single leg is

$$F_s = 1.5 \times \frac{F_I}{2} \approx 726N . \quad (3.22)$$

With the derivations made above, we conclude the jumping and landing force and stress analysis of the robot.

3.5.3 Force Analysis of the Robot Head Pitch and Yaw Motion

As depicted in Figure 3.24. The maximum total mass of the robot head, including webcam and other sensors, is approximately $m_h \approx 4kg$, and the offset from the center axis is $l_h = 0.0565m$. The maximum neck pitch torque τ_{np} required is

$$\tau_{np} = m_h g l_h = 4 \times 9.81 \times 0.0565 = 2.217Nm . \quad (3.23)$$

The maximum angular turning velocity of a human neck is around $160^0/s$, we design our robot to have $\omega_{np} = 180^0/s$ to extend its performance, and the power required is

$$P_{np} = \tau_{np} \times \omega_{np} = 2.21 \times \pi = 6.96w . \quad (3.24)$$

When the robot is detecting for objects or obstacles on the ground, the head will bend over and the maximum torque required from the neck yaw is the same as the pitch movement. Thus, the power requirement from the neck yaw actuator is

$$P_{ny} = P_{np} = 6.96w . \quad (3.25)$$

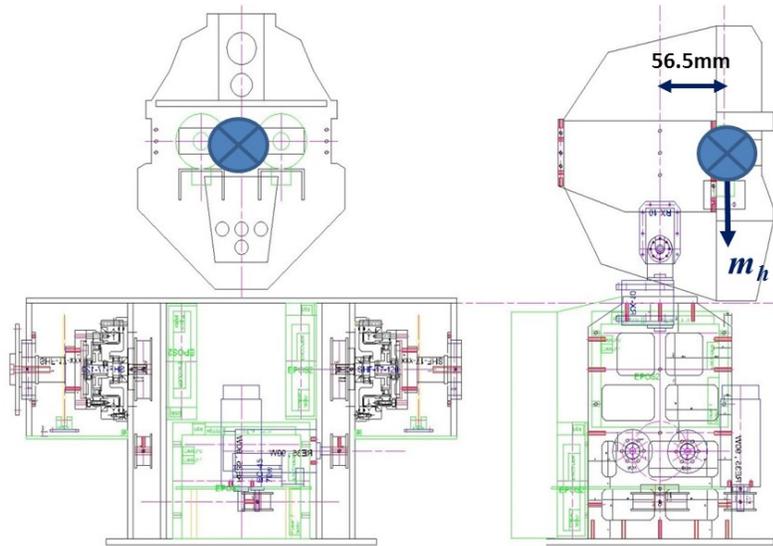


Figure 3.24. CoM location of the robot head.

3.5.4 Force Analysis of Pelvis and Waist Motion

The robot waist joint (pelvis) is designed with three DoF with pitch, yaw and roll movement. A typical human motion measured by using IMU sensors [H.J. Luinge et al., 2002] is depicted in Figure 3.25, the values shows that the maximum angular velocity of the pelvis roll, pitch and yaw movements are $\omega_{px} = 1 \text{ rad/s}$, $\omega_{py} = 2.5 \text{ rad/s}$ and $\omega_{pz} = 1.5 \text{ rad/s}$ respectively.

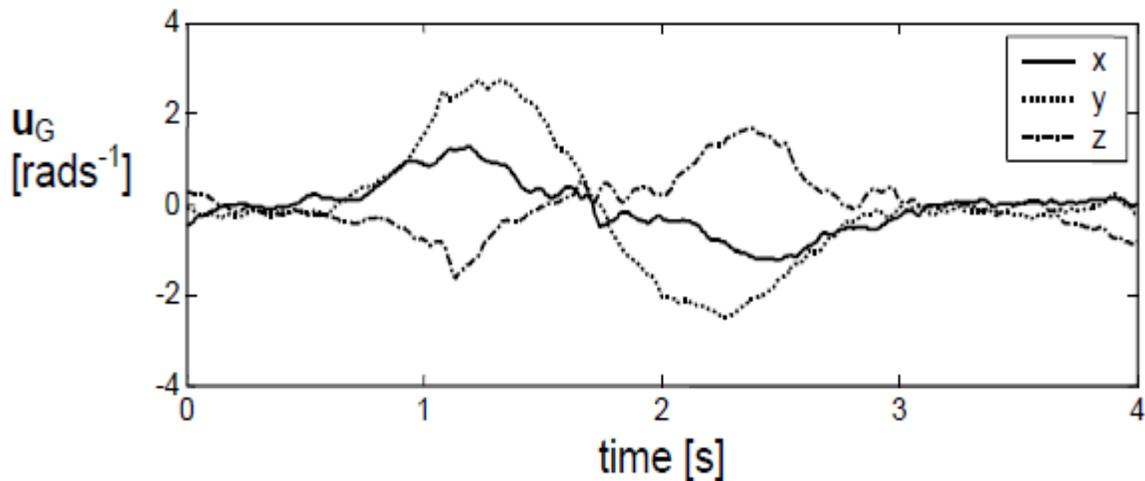


Figure 3.25. Angular velocity of the pelvis three axis.[H.J. Luinge et al., 2002].

The mass of the upper body of our robot is $m_{u_body} = 20.3 \text{ kg}$, and the maximum pitch and roll angle are 45° and 30° respectively, as depicted in Figure 3.26. The power requirement for the actuators can be determined with the following set of equations.

With $\omega_{px} = 1 \text{ rad/s}$, $\omega_{py} = 2.5 \text{ rad/s}$, $\omega_{pz} = 1.5 \text{ rad/s}$,

The maximum roll torque is

$$\tau_{px} = m_{u_body} \times 0.124 \times 9.81 = 24.69 \text{ Nm} ,$$

so the maximum power required from the roll actuator can be determined as

$$P_{px} = \tau_{px} \times \omega_{px} = 24.69 \times 1 = 24.69 \text{ w} .$$

The maximum yaw and pitch torque is

$$\tau_{py} = \tau_{pz} = m_{u_body} \times 0.1755 \times 9.81 = 34.95 \text{ Nm} ,$$

so the maximum power required from the yaw actuator can be determined as

$$P_{pz} = \tau_{pz} \times \omega_{pz} = 34.95 \times 1.5 = 52.42 \text{ w} .$$

The maximum power required from the pitch actuator is

$$P_{py} = \tau_{py} \times \omega_{py} = 34.95 \times 2.5 = 87.38 \text{ w} .$$

(3.26)

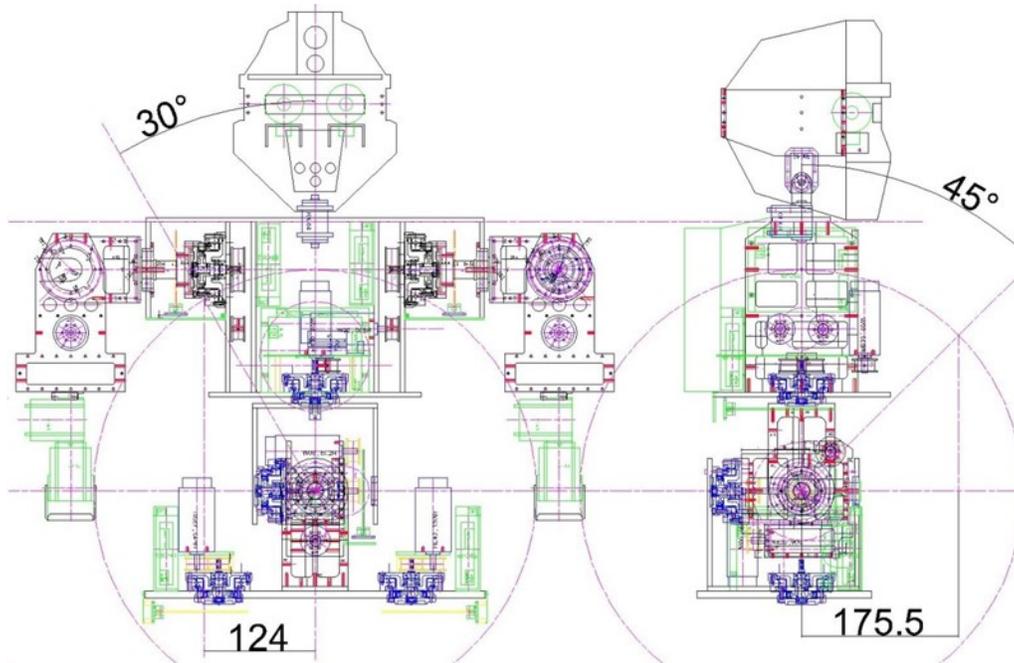


Figure 3.26. Robot maximum yaw and pitch angle of the robot.

3.5.5 Force Analysis of Arm and Hand

The maximum torque exists for the case when the arm is fully extends horizontally. We set the maximum payload that the hand can handle is 2kg, and the weight and dimension of the arm links are listed in Table 3.2.

Table 3.2. Dimension and weight of robot arm and hand.

Segment	Joint	Length (m)	Mass(kg)
Arm	elbow	0.244	0.31
	wrist	0.115	0.22
	Hand	0.1	0.25

Let denote the weight of the payload, elbow joint, wrist joint and the hand be m_l, m_e, m_w, m_h respectively as depicted in Figure 3.27. Then the power requirement for individual segment when the arm is fully extended horizontally can be determined as follows:

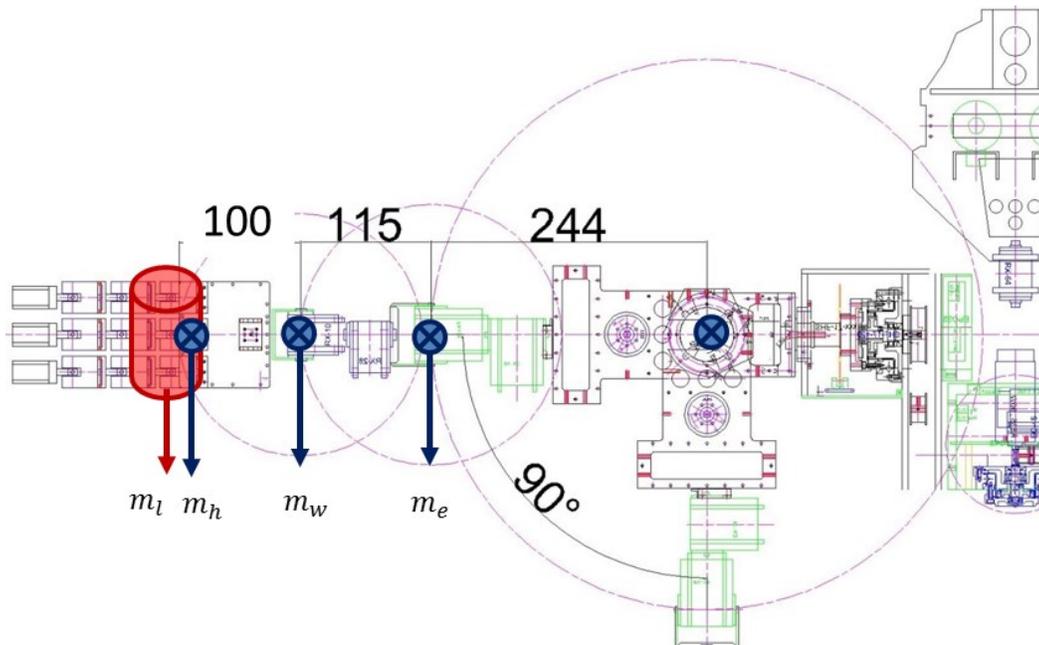


Figure 3.27. Dimension and CoM location of arm joints.

For the wrist: $\tau_w = (m_l + m_h)g \times 0.1 = (2 + 0.25) \times 9.81 \times 0.1 = 2.207Nm$.

For the elbow: $\tau_e = (2.25 \times 0.215 + 0.22 \times 0.115) \times 9.81 = 4.99Nm$.

For the shoulder: $\tau_s = (2.25 \times 0.459 + 0.22 \times 0.359 + 0.31 \times 0.244) \times 9.81 = 11.64Nm$.

(3.27)

In order to determine the power requirement for each joint, parameters from Figure 3.28 were taken as reference. It shows an experiment data of the angular velocity of a human upper arm [H.J. Luinge et al., 2002]. The measurement is based on some general tasks such as drinking and combing hair.

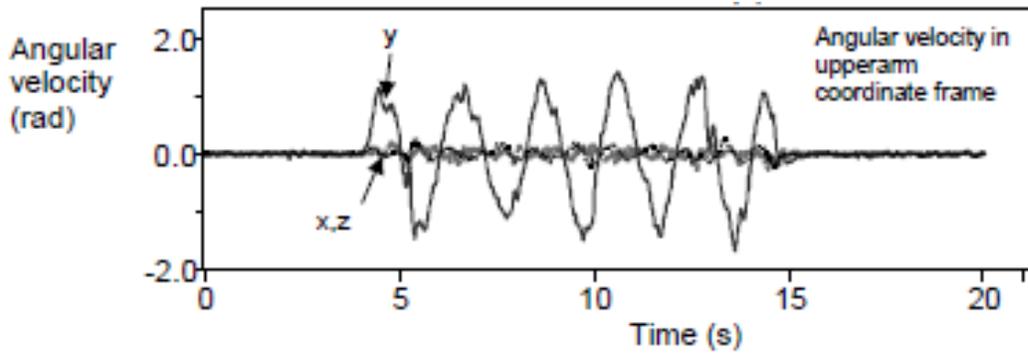


Figure 3.28. Angular velocity of a human upper arm [H.J. Luinge et al., 2002].

From the diagram above, we can see that the maximum angular velocity is about 1.5 rad/s . So using the maximum value, the power required by each joint can be estimated as:

For the wrist: $P_w = \tau_w \times \omega = 2.205 \times 1.5 = 3.3w$.

For the elbow: $P_e = 4.99 \times 1.5 = 7.49w$.

For the shoulder: $P_s = 11.64 \times 1.5 = 17.46w$.

(3.28)

3.5.6 Force Analysis of Hand and Fingers

There are different ways of grasping an object, the type as illustrated in Figure 3.29 is an extreme way to grasp an object because more grasping force is required by the fingers to prevent slippage. The forces relationship are denoted as follows: F_f is the friction force between the

fingers and the object, F_p is the normal force acting on the contact object interface, T_f is the torque generated by the normal force. The weight of the object is $m = 2kg$, the distance from the midpoint of the hand is $d = 0.05m$, $g = 9.81m/s^2$ is the gravity acceleration and $\mu = 0.5$ denotes the static friction coefficient of the rubber fingers. The maximum power required from the fingers can be determined as

$$F_g = mg = 2F_f ,$$

$$F_p = \frac{F_f}{\mu} = \frac{mg}{2\mu} . \quad (3.29)$$

The normal torque from the fingers to grasp is 2kg object is

$$T_f = F_p \times d = \frac{mg}{2\mu} \times d = \frac{2 \times 9.81}{2 \times 0.5} \times 0.05 = 0.98Nm . \quad (3.30)$$

The maximum power required by the fingers is

$$P_f = T_f \omega = 0.98 \times \frac{\pi}{2} = 1.54w . \quad (3.31)$$

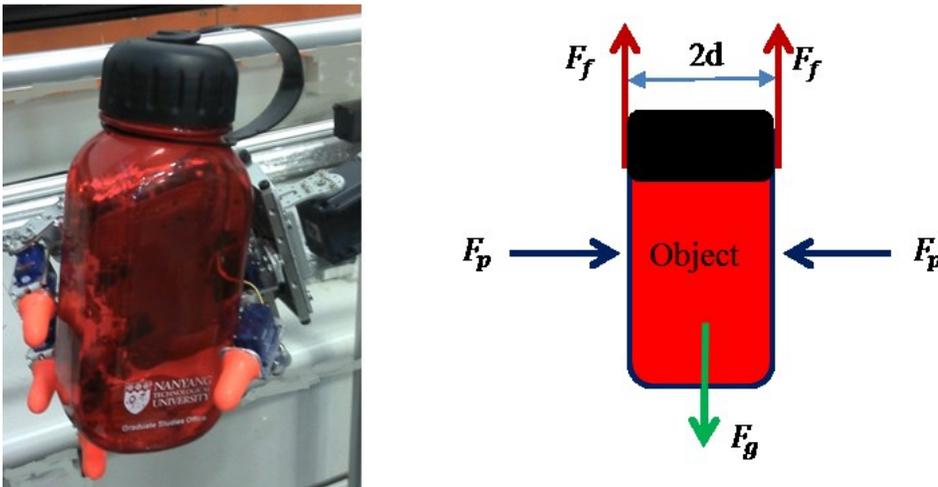


Figure 3.29. Grasping forces analysis of a robot hand.

3.5.7 Walking Energy Requirement

As discussed in Chapter 2, gait cycle has two phases: double leg support and single leg support phase. As illustrated in Figure 3.30, the walking cycle usually starts from the heel ground contact of the swing foot and heel-off of the stance foot.

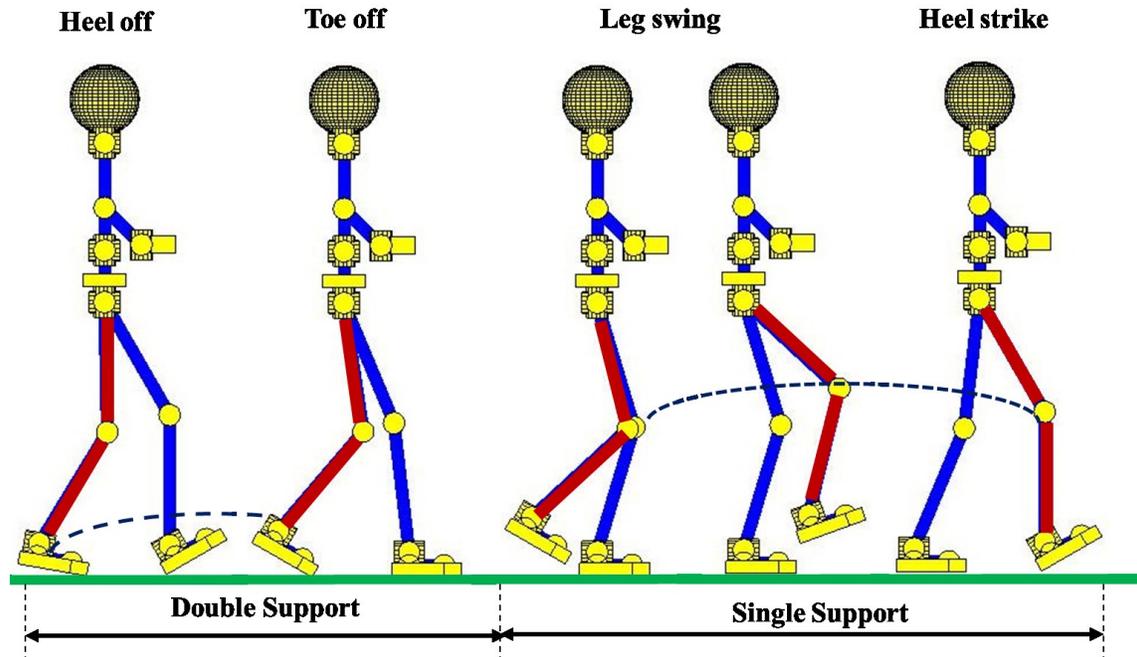


Figure 3.30. Gait cycle of walking.

From biomechanics literature [H.J. Luinge et al., 2002], the average walking energy required by a human, expressing in terms of walking speed, can be approximated by the following equation

$$P = m(2.23 + 1.26v^2), \quad (3.32)$$

where m is the mass, v is the velocity of the centre of gravity in m/s and P is the power consumption in watt (w). Apparently, the power consumption is lesser for a slower walking speed. This equation will be used as a reference to calculate the power consumption of the TPinokio robot. The total mass of the robot is 58kg, and its average walking speed is 0.3m/s, from equation (3.32), we can determine the power

$$P = 58(2.23 + 1.26 \times 0.3^2) = 135.92w. \quad (3.33)$$

From literature, considering most similar size robots should operate for around 2 hrs, using Lithium-ion battery with the capacity of 24v-12AH, the maximum operational time for TPinokio robot is around 2 hrs.

3.5.8 Hip and Ankle Power Analysis

In the walk motion, the pelvis of the bipedal robot will swing from side to side, forward and backward depending on posture requirement. The most frequently executed motion is to swing

the pelvis and upper body from single support leg to double leg position, and these actions need sufficient power to execute as illustrated in Figure 3.31.

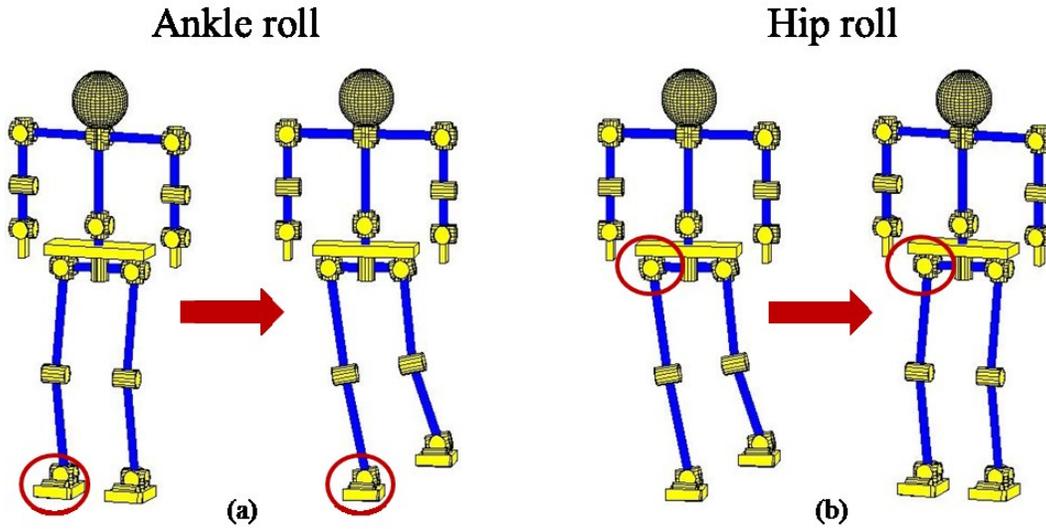


Figure 3.31. (a) Ankle roll to single support standing, (b) Hip roll in to double support.

3.5.8.1 Ankle Roll Power Requirement

As depicted in Figure 3.31(a), the robot is in right leg support after the left leg swing off the ground. The right ankle joint actuator will need to drive the pelvis and upper body to a stable right leg support position. The weight of the robot is 58kg and the width between the two legs is $l = 0.215m$, the maximum torque can be determined as

$$\tau_{max} = mg \cdot \frac{l}{2} = 58 \times 9.81 \times \frac{0.215}{2} = 61.17Nm . \quad (3.34)$$

This action is estimated to take about 0.2~0.4s from normal standing to single support standing, and the angle change is about 7° , we use the average of 0.3s, so the average angular velocity is

$$\omega = \frac{\Delta\theta}{\Delta t} = \frac{7\pi/180}{0.3} \approx 0.41rad/s . \quad (3.35)$$

And the maximum power of the ankle roll can be calculated

$$P_{ankle_roll} = \tau_{max} \cdot \omega = 61.17 \times 0.41 = 25.07w . \quad (3.36)$$

3.5.8.2 Hip Roll Power Requirement

As depicted in Figure 3.31(b), assume right leg single support, the robot now recovers to normal double support standing. The width between the two legs is $l = 0.215m$ and the total mass of the upper body and the right leg is $m_{up_lg} = 30.14kg$. The maximum torque required by the hip roll actuator can be determined as

$$\tau_{max} = m_{up_lg} \cdot g \cdot \frac{l}{2} = 30.14 \times 9.81 \times \frac{0.215}{2} = 31.78Nm . \quad (3.37)$$

And the average angular velocity is

$$\omega = \frac{\Delta\theta}{\Delta t} = \frac{7\pi/180}{0.3} \approx 0.41rad/s , \quad (3.38)$$

The maximum torque required by the hip roll actuator is

$$P_{hip_roll} = \tau_{max} \cdot \omega = 31.78 \times 0.41 = 13.03w . \quad (3.39)$$

3.5.8.3 Hip Yaw Power Requirement

When the robot does a 90° turn, the hip yaw movement of the robot can be divided to in to two stages as depicted in Figure 3.32. In the first stage, the right leg turns from the lateral direction to the frontal side with the upper body turning about 45° and the left leg remains at stance position. While in the second stage, the right leg keeps still, the left leg and the upper body turns another 45° to the frontal direction.

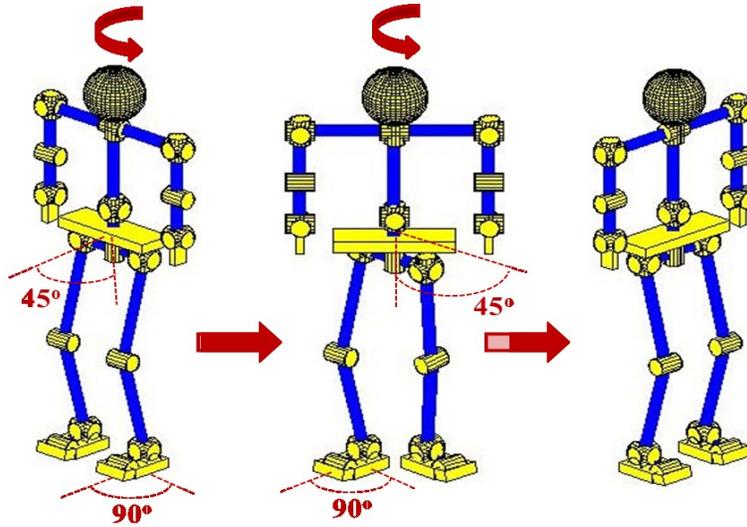


Figure 3.32. Robot making a 90 degree turning.

During the first stage of turning, the turn angle of the upper body $\theta_{up_body} = \frac{\pi}{4}$, the turning time is $t = 0.5s$, and the angular acceleration of the turning is $\varphi = 2rad/s^2$. Given that the rotary inertia (I_{RR}) with respect to the upright axis of the left leg is $I_{HY} = 7.84kgm^2$, which consist the rotary inertia of the upper body and right leg, except the left leg. The turning torque can be determined as follows:

$$T = I_{HY} \cdot \varphi = 15.68Nm, \quad (3.40)$$

and the average angular velocity is

$$\omega = \frac{\theta_{up_body}}{t} = \frac{\pi}{2}rad/s. \quad (3.41)$$

The total power required for hip yaw is

$$P = T\omega = 24.62w. \quad (3.42)$$

With the above literature, it concludes the power requirement and torque calculation of the TPinokio robot. The power requirement of all the joints motor is listed in Appendix A.

3.6 Chapter Summary

In this chapter the design and torque analysis of a new mid-size bipedal humanoid walking robot with 55 DoF has been presented. The design is optimized for an average walking speed of around 0.3m/s and a jumping height of around 0.15m. The hand is able to hold an object weight around 2kg. This biped is designed with passive-toe, so it produces more human-like walking posture. Another novel feature of the design is that it has compact modular joints with minimum mechanical parts. Furthermore, most parts are identical for each joint. This compact modular design leads to simple kinematic modeling and easy maintenance.

The interfacing electronics has also been described in details, including load-cell and force/torque sensors board, which is connected to the main controller via USB interface. A single 5-core wire connection between each compact module reduces the messy wire harnessing problems. Using a Notebook as a main controller and window based OS solve lots of programming and interfacing issues. LabVIEW real time programming platform provide excellent Graphic User Interface (GUI) and by adopting CANbus as the main communication protocol provide high speed motion control to the EPOS2 motor controller. The EPOS2 is able to provide position, speed and current torque control to the robot joints.

CHAPTER 4

KINEMATICS AND DYNAMICS

MODEL

In this chapter, the detail kinematics and dynamics model of the TPinokio robot is presented. Kinematics of the motion of a body in a robotic mechanism does not consider the force/torque that acting on the body or causes the motion. Dynamic equations of motion show the relationships between the forces and momentum exerting on the robot mechanism, and the resulting motion trajectories that due to the reacting forces. This chapter will only describe and discuss the kinematics models and the Degree of Freedom (DoF) of the bipedal robot lower limbs (the hip and both legs), which is the serial-chain of hip-knee-ankle-foot.

4.1 Kinematics Model of TPinokio Robot

TPinokio is designed with simple revolute joints and is assumed to be a rigid-body system consists of multiple symmetrical shape (spherical, cylindrical, rectangular) rigid-objects joined together by simple links. Its joints and mechanism were designed to produce a simple kinematics model. One feature to be highlighted is that the compact modular housing design (comprise of motors, gears, sensors, electronics) and the links to each modular housing joint are fabricated with lightweight aluminum. Due to the feature in the robot, the mass and inertia of the links are negligible compared to the weight of the modular housing joints. As shown in Figure 4.1, all the links may be assumed to have a ‘point-mass’ located at the distal end. Thus, the ‘point-mass’ kinematics model of the robot can be accurately modeled because its position can be measured accurately.

It is noted that other similar robots (LUCY, HRP, LOLA) that use the ‘point mass’ assumption in their modeling may not provide an accurate model as compared to our design. The rationale is that these robots’ actuators and motors are mounted on the joint’s links, especially for the knee joint actuator and the ankle actuator as depicted in Figure 4.2, this is inaccurate to assume zero inertia for the joint link.

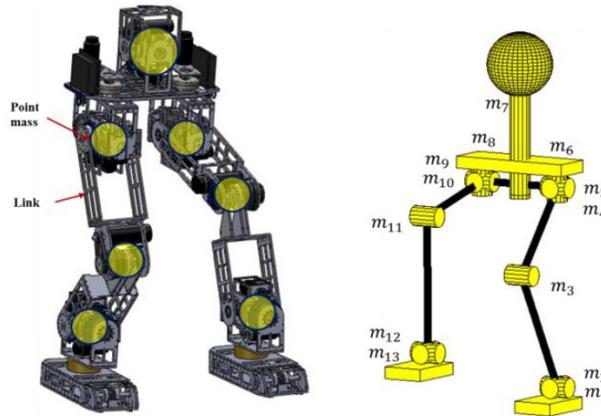


Figure 4.1. TPinokio simplified point-mass model.

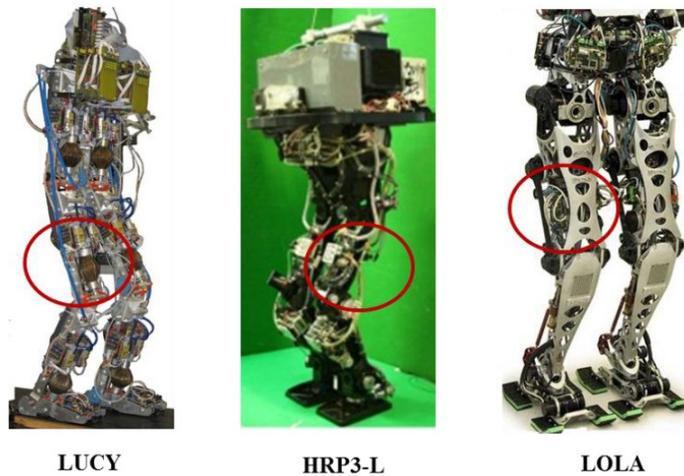


Figure 4.2. Similar robot with actuator mounted on the joint link.

In forward kinematics, the end-effector’s position and its orientation can be easily determined once the entire joints’ variables are defined. In the case for the inverse kinematics, each joint parameter is calculated from the end-effector position and orientation, and this does not necessary provide a closed-form solution.

4.1.1 Forward Kinematics with Denavit-Hartenberg Model

The forward kinematics transforms the kinematic information of a robot joint-link from the joint space to the Cartesian space. The most common method is to apply the Denavit Hartenberg (DH) transformation. As illustrated in Figure 4.3, the geometric structure and kinematics-chain of the TPinokio robot is converted to the D-H notation. The angle rotating direction is based on right-hand rule. Please refer to Appendix A for more details.

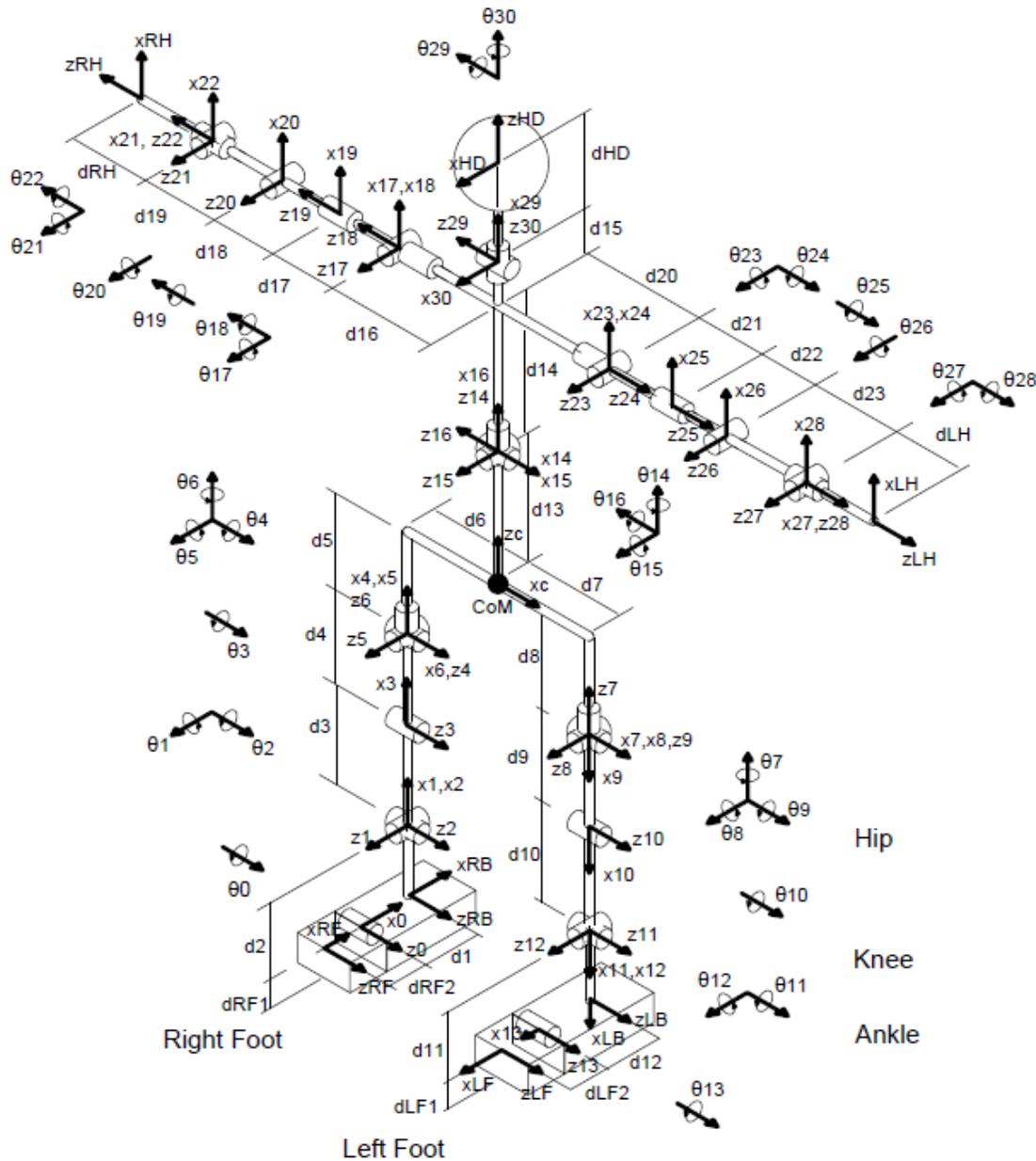


Figure 4.3 TPinokio D-H model and link frames.

4.1.2 Inverse Kinematics

As mentioned in Chapter 2, there are several ways to solve inverse kinematics problem. The three commonly used methods are: geometric, iterative and algebraic. For this research, the first two methods will be investigated and applied to the TPinokio robot.

4.1.2.1 Geometric Method

It is assumed that the masses of the upper body of TPinokio are concentrated at point masses located at the center of the pelvis/waist; this assumption is generally acceptable with the simple inverted pendulum model (IPM) approach. TPinokio robot has 6 DoF for each leg, the kinematics parameters are defined as shown in Figure 4.4. The right leg close-form based on geometrical method is presented here without details derivations leading to the solutions.

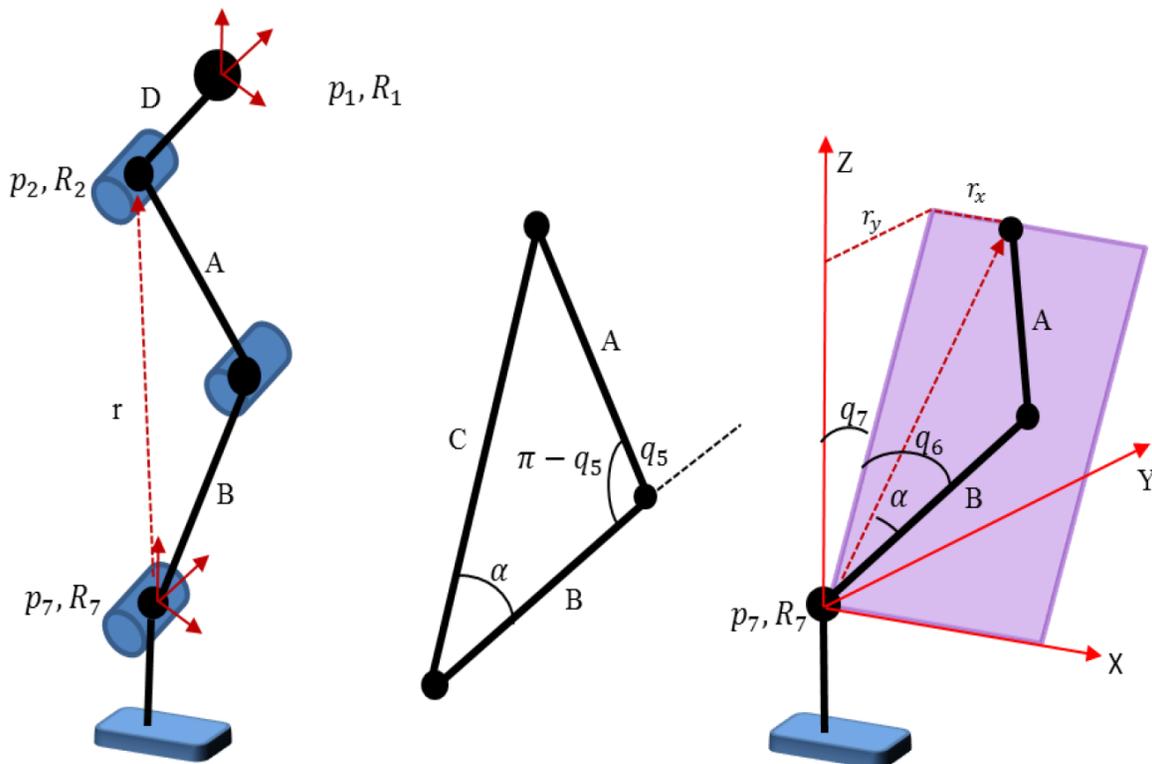


Figure 4.4 TPinokio robot right leg kinematics stick diagram.

Referring to Figure 4.4, the inverse kinematic parameters are as shown in the following set of equations without details derivation:

$$p_2 = p_1 + R_1 \begin{bmatrix} 0 \\ D \\ 0 \end{bmatrix}, \quad (4.1)$$

$$r = R_7^T(p_2 - p_7) \equiv [r_x \quad r_y \quad r_z]^T,$$

$$c = \sqrt{r_x^2 + r_y^2 + r_z^2},$$

$$C^2 = A^2 + B^2 - 2AB \cos(\pi - q_5),$$

$$\textit{knee pitch: } q_5 = -\cos^{-1}\left(\frac{A^2 + B^2 - C^2}{2AB}\right) + \pi, \quad (4.2)$$

$$\frac{C}{\sin(\pi - q_5)} = \frac{A}{\sin \alpha},$$

$$\alpha = \sin^{-1}\left(\frac{A \sin(\pi - q_5)}{C}\right),$$

$$\textit{ankle pitch: } q_6 = -\text{atan2}\left(r_x, \text{sign}(r_z) \sqrt{r_y^2 + r_z^2}\right) - \alpha, \quad (4.3)$$

$$\textit{ankle roll: } q_7 = \text{atan2}(r_y, r_z), \quad (4.4)$$

$$R_7 = R_1 R_z(q_2) R_x(q_3) R_y(q_4) R_y(q_5 + q_6) R_y(q_5 + q_6),$$

$$\begin{bmatrix} c_2 c_4 - c_2 s_3 s_4 & -s_2 c_3 & c_2 s_4 + s_2 s_3 c_4 \\ s_2 c_4 + c_2 s_3 s_4 & c_2 c_3 & s_2 s_4 - c_2 s_3 c_4 \\ -c_3 s_4 & s_3 & c_3 s_4 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix},$$

$$\text{where } c_* \equiv \cos q_*, \quad s_* \equiv \sin q_*,$$

$$\textit{hip yaw: } q_2 = \text{atan2}(-R_{12}, R_{22}), \quad (4.5)$$

$$\textit{hip roll: } q_3 = \text{atan2}(R_{32}, -R_{12} s_2 + R_{22} c_2), \quad (4.6)$$

$$\textit{hip pitch: } q_4 = \text{atan2}(-R_{31}, R_{33}). \quad (4.7)$$

4.1.2.2 Iterative Method

In Chapter 2, it was shown how the *Newton-Raphson* iterative technique [R.N. Jazar, 2010] is applied to solve inverse kinematics problem to determine the closet possible solution θ from a series of nonlinear equations. This method may not produce the correct solution because for nonlinear equations there will be normally multiple solutions, and each solution is dependent on the initial value. As such, we need to first set the initial position of the robot joints and use these positions as the starting values to initiate the iteration in order for the search algorithm to perform correctly.

Please refer to the Appendix B for detail description of the inverse kinematics iteration algorithm and technique. The kinematics link of each joint of the TPinokio robot is as shown in Figure 4.5. Its link coordinate and numbering notation is listed in Table 4.1.

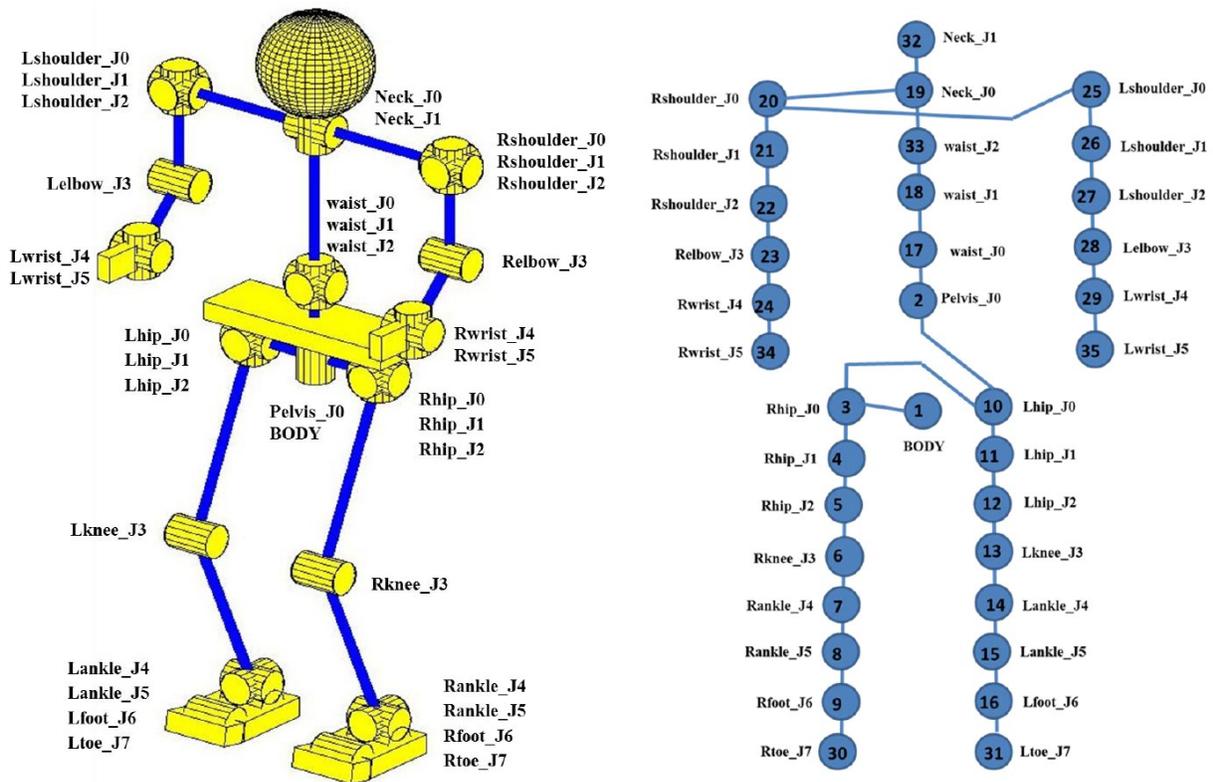


Figure 4.5. TPinokio robot joint and kinematic links.

Table 4.1. TPinokio robot joint link coordinate.

Lower Body				Upper body			
ID	Segment name	link 2	link1	ID	Segment name	link2	link1
1	BODY	0	3	19	neck_J0	32	20
2	Pelvis_J0	0	17	32	neck_J1	0	0
3	Rhip_J0	10	4	20	Rshoulder_J0	25	21
4	Rhip_J1	0	5	21	Rshoulder_J1	0	22
5	Rhip_J2	0	6	22	Rshoulder_J2	0	23
6	Rknee_J3	0	7	23	Relbow_J3	0	24
7	Rankle_J4	0	8	24	Rwrist_J4	0	34
8	Rankle_J5	0	9	34	Rwrist_J5	0	0
9	Rfoot_J6	0	30	25	Lshoulder_J0	0	26
30	Rtoe_J7	0	0	26	Lshoulder_J1	0	27
10	Lhip_J0	2	11	27	Lshoulder_J2	0	28
11	Lhip_J1	0	12	28	Lelbow_J3	0	29
12	Lhip_J2	0	13	29	Lwrist_J4	0	35
13	Lknee_J3	0	14	35	Lwrist_J5	0	0
14	Lankle_J4	0	15				
15	Lankle_J5	0	16				
16	Lfoot_J6	0	31				
31	Ltoe_J7	0	0				
17	waist_J0	0	18				
18	waist_J1	0	33				
33	waist_J2	0	19				

4.1.3 Kinematics Model Verification

Forward and inverse kinematics is essential tool for a robot to create motion. The TPinokio forward and inverse kinematics model is verified by MATLAB simulation. In the simulation, the forward kinematics is used to determine the posture of the robot that is derived from the predefined joints' angle, and the inverse kinematics is applied to determine the individual joints' position to achieve a predefined posture.

4.1.3.1 Initial Posture Forward Kinematics Verification

The initial posture of the robot is determined by using forward kinematics. The initial value for each joint is pre-defined and then using the forward kinematics to derive the robot posture, this is verification is shown in Figure 4.6.

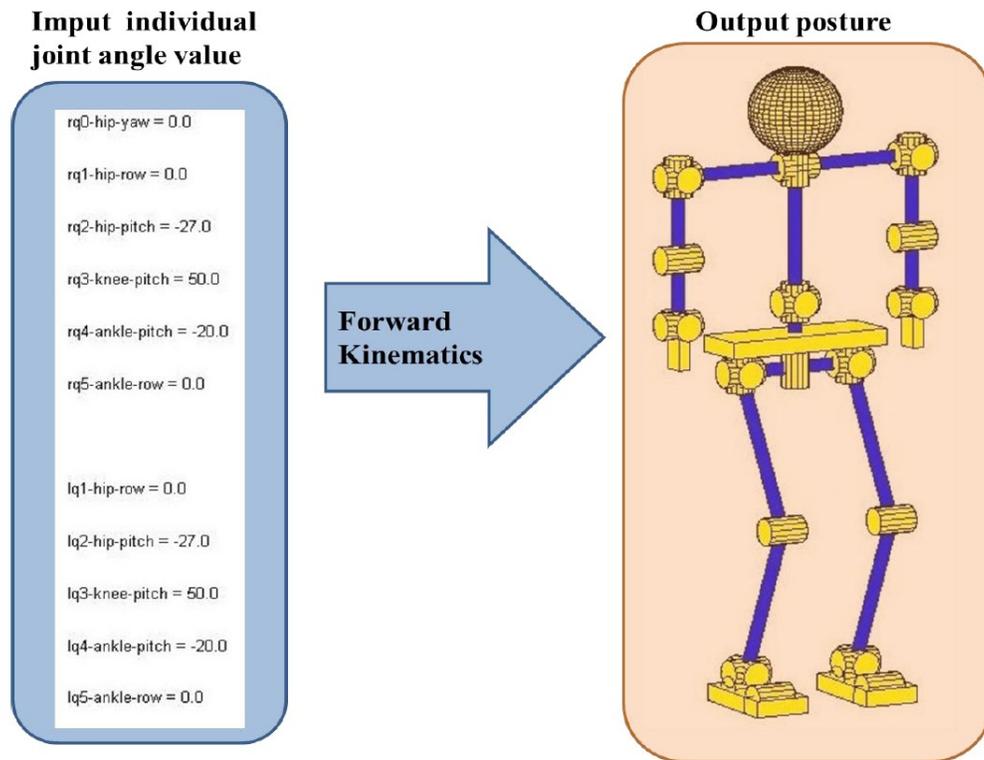


Figure 4.6. Robot initial posture is determined by applying forward kinematics .

4.1.3.2 Walking Gait Generation Inverse Kinematics Verification

The individual joint angle during walking is determined by applying inverse kinematics (IK). Once the orientation of the robot pelvis, right foot and left foot (end-effectors) position is

defined, the individual joint may be determined using *Newton-Raphson* method. This is verified as depicted in Figure 4.7.

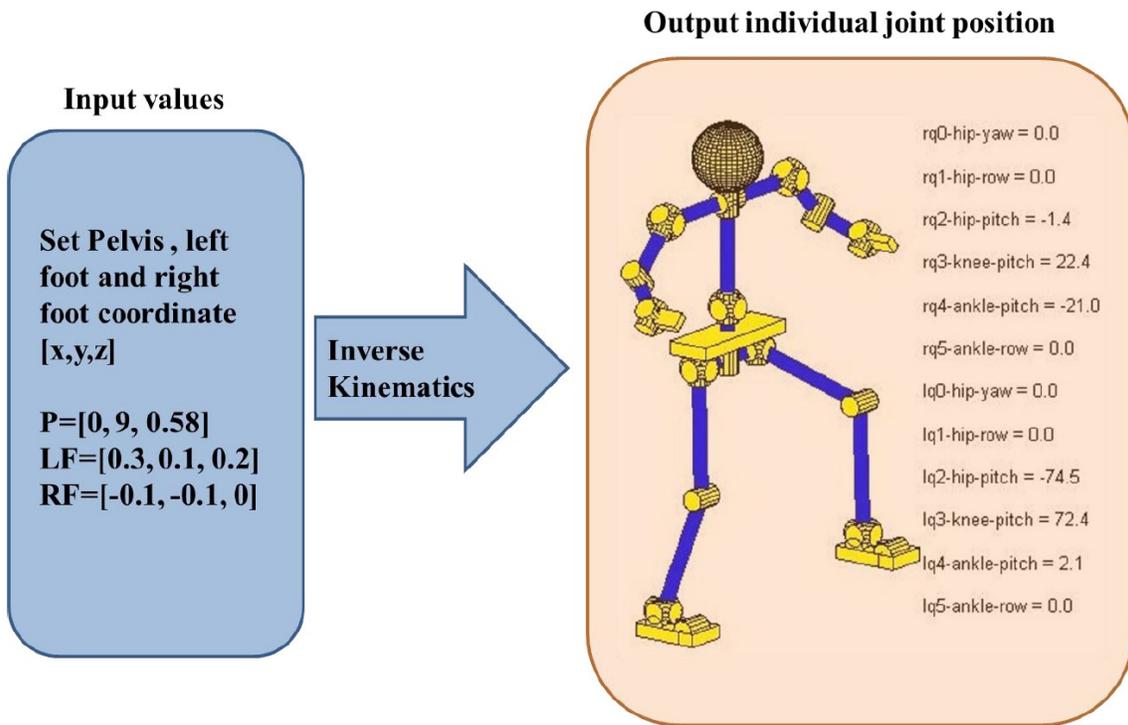


Figure 4.7. Individual joint angle position is determined by inverse kinematics.

4.2 Dynamic Model of TPinokio Robot

As discussed in Chapter 2, there are various methods to model a bipedal robot walking dynamic. In this research, the 3D inverted Pendulum model (3DIPM) is adopted in our analysis due to its simplicity and is also found to be a good approximation of our TPinokio kinematics and dynamics model.

4.2.1 3D Inverted Single Mass Pendulum Model

During the swing phase, the Centre of Mass (CoM) of the robot may be modeled as a ‘point mass’ system, and it is connected to the support leg like an inverted pendulum as illustrated in Figure 4.8. In order to define the relationship of the ZMP and the CoM trajectories, a simplified ZMP with Inverted Pendulum Model (IPM) is adopted in this analysis.

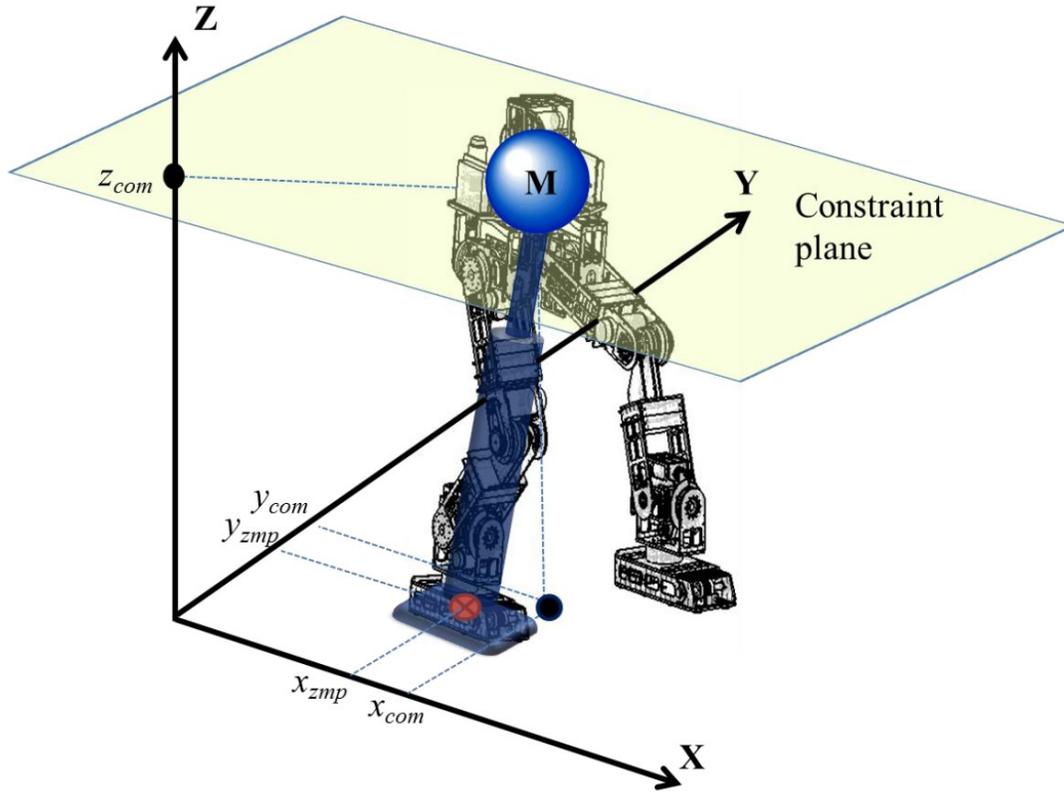


Figure 4.8. Inverted pendulum model superimposed on to the robot.

Referring to Figure 4.8, by definition, for a plane given by the equation $ax + by + cz + d = 0$, the vector (a, b, c) is a normal. In order to model the IPM for bipedal walking, certain constraints need to be applied to limit the motion of the inverted pendulum mass. For instance, we define the robot walking frontal is in the x-axis direction, and limit the motion by a constraint z-plane with vector parameters $(k_x, k_y, -1)$ and intersect on the z-axis at $z = z_{com}$. The constraint plane can be written as

$$d = z_{com} - k_x x - k_y y . \quad (4.8)$$

When the constraint plane is set to $(k_x = k_y = 0)$, i.e. horizontal to the ground surface, we can apply this constraint control with equation (4.8) to limit the mass movement, the mass can only on this horizontal plane. This technique is used to determine the dynamic behaviors and to derive the equations for the IPM system to walk on flat ground, slope and on stairs. This is a good

approximation compared to other more complicated approaches. The application of this technique will be explained in details in Chapter 5.

An approximation approach can be used to estimate the multi-point mass bipedal robot as shown in Figure 4.9, the CoM ($x_{com}, y_{com}, z_{com}$) of the robot can be calculated as

$$x_{com} = \frac{\sum_i m_i x_i}{\sum_i m_i},$$

$$y_{com} = \frac{\sum_i m_i y_i}{\sum_i m_i},$$

$$z_{com} = \frac{\sum_i m_i z_i}{\sum_i m_i},$$

(4.9)

where m_i is the mass and $p_i = (x_i, y_i, z_i)$ denote the coordinate point. The GCoM ($x_{com}, y_{com}, 0$) is defined as the ground projection of the CoM.

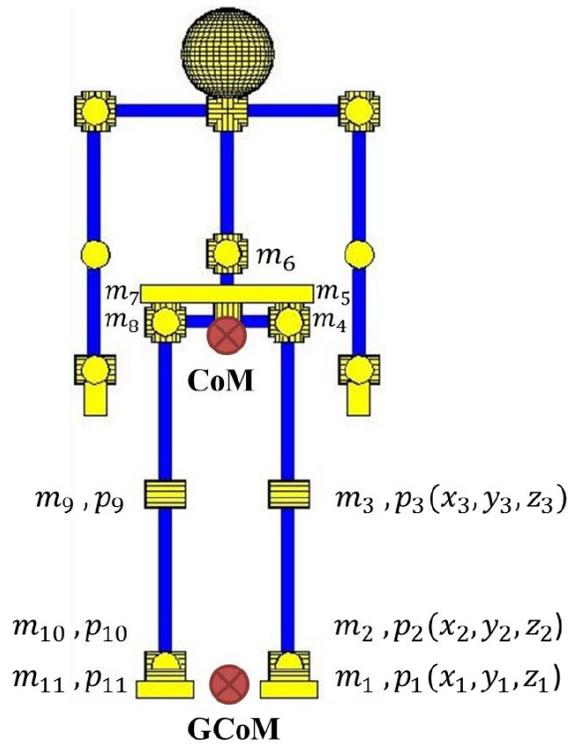


Figure 4.9. TPinokio point mass distribution model.

As discussed in Chapter 2, the ZMP is defined as a position on the contact surface where the sum of all the exerting forces and moments is reduced to zero. For a multi-mass model, where m_i represents the mass of the particles, the ZMP can be expressed as

$$\begin{aligned}x_{zmp} &= \frac{\sum_{i=1}^N m_i \{(\ddot{z}_i + g)x_i - z_i \ddot{x}_i\}}{\sum_{i=1}^N m_i (\ddot{z}_i + g)}, \\y_{zmp} &= \frac{\sum_{i=1}^N m_i \{(\ddot{z}_i + g)y_i - z_i \ddot{y}_i\}}{\sum_{i=1}^N m_i (\ddot{z}_i + g)}.\end{aligned}\tag{4.10}$$

For our robot design, the masses are concentrated at the links end and the links inertia is assumed to be zero. Taking these into consideration, from equations (4.9) and equation (4.10), the ZMP is related to the CoM by

$$\begin{aligned}x_{zmp} &= x_{com} - \frac{(z_{com})\ddot{x}_{com}}{\ddot{z}_{com} + g}, \\y_{zmp} &= y_{com} - \frac{(z_{com})\ddot{y}_{com}}{\ddot{z}_{com} + g}.\end{aligned}\tag{4.11}$$

For constant pelvis height, the $\ddot{z}_{com} = 0$, so equation (4.11) can be simplified to

$$\begin{aligned}x_{zmp} &= x_{com} - \frac{(z_{com})\ddot{x}_{com}}{g}, \\y_{zmp} &= y_{com} - \frac{(z_{com})\ddot{y}_{com}}{g}.\end{aligned}\tag{4.12}$$

It is noted that in the situation where a robot moves at a slow or constant speed, by setting the acceleration term \ddot{y}_{com} and \ddot{x}_{com} to zero, equation (4.12) can be further simplified to

$$\begin{aligned}x_{zmp} &= x_{com}, \\y_{zmp} &= y_{com}.\end{aligned}\tag{4.13}$$

Equation (4.12) will be used in the controller design and walking gait generation in Chapter 5.

4.2.2 3D Inverted Double Mass Inverted Pendulum Model

In this section, a new double-mass inverted pendulum model is presented. The lower mass, which represents the robot lower body Center-of-Mass (CoM), is constrained to a horizontal plane Z. The second mass, which represents the upper body CoM of the bipedal robot, is constrained to deviate in small angle variation in Z-axis. The two mass models are depicted in Figure 4.10.

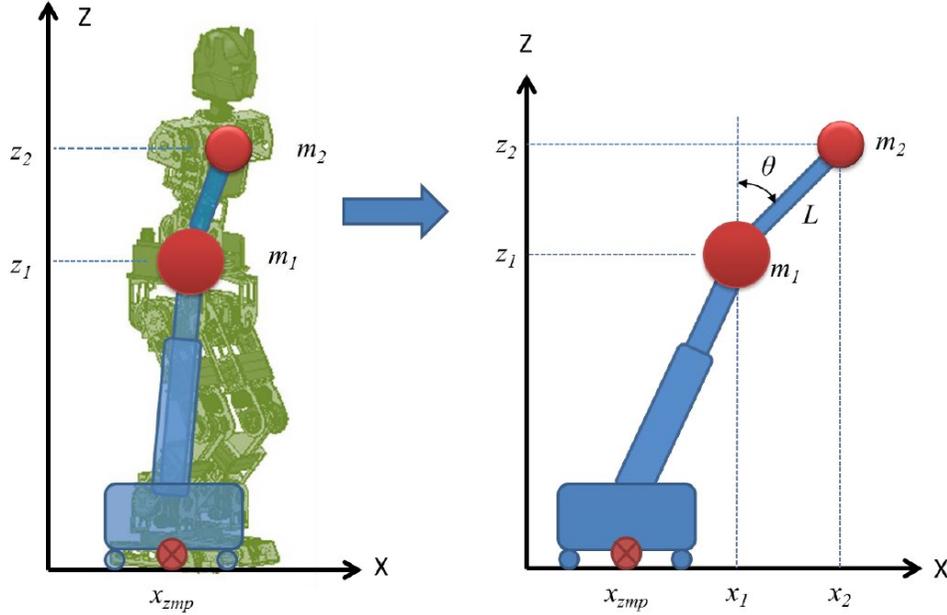


Figure 4.10. Double IPM model.

Referring to Figure 4.10, the torques and moments equation about the ZMP point is derived as

$$m_1 g(x_1 - x_{zmp}) + m_2 g(x_2 - x_{zmp}) = m_1 z_1 \ddot{x}_1 + m_2 z_2 \ddot{x}_2 . \quad (4.14)$$

Where θ and L is the angle and length of the second link respectively, and $x_2 = x_1 + L \sin \theta$, $z_2 = z_1 + L \cos \theta$. Substituting these expressions into equation (4.14), we obtain

$$m_1 g(x_1 - x_{zmp}) + m_2 g(x_1 + L \sin \theta - x_{zmp}) = m_1 z_1 \ddot{x}_1 + m_2 (z_1 + L \cos \theta) \ddot{x}_2 . \quad (4.15)$$

Based on the constrain that θ with small deviation ($\theta \approx 0^\circ$) , we can linearized equation (4.15), with $x_2 = x_1 + L\theta$ and $z_2 = z_1 + L$ the ZMP equation is derived as

$$x_{zmp} = x_1 - \frac{m_1 z_1 + m_2 (z_1 + L)}{(m_1 + m_2)g} \ddot{x}_1 + \frac{m_2 L}{m_1 + m_2} \theta - \frac{m_2 L (z_1 + L)}{(m_1 + m_2)g} \ddot{\theta} , \quad (4.16)$$

which can be arranged in matrix form

$$x_{zmp} = \begin{bmatrix} 1 & \frac{m_1 L}{m_1 + m_2} & 0 & 0 & -\frac{m_1 z_1 + m_2(z_1 + L)}{(m_1 + m_2)g} & -\frac{m_2 L(z_1 + L)}{(m_1 + m_2)g} \end{bmatrix} \begin{bmatrix} x_1 \\ \theta \\ \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix}. \quad (4.17)$$

To convert the equation (4.17) into a standard state space form, we introduce a control input $u = [\ddot{x} \ \ddot{\theta}]^T$, and the state $x = [x_1 \ \theta \ \dot{x}_1 \ \dot{\theta} \ \ddot{x}_1 \ \ddot{\theta}]^T$, the state space model of the double inverted pendulum can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \theta \\ \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \ddot{\theta} \end{bmatrix}, \quad (4.18)$$

$$\underbrace{y}_{P_{zmpx}} = \begin{bmatrix} 1 & \frac{m_1 L}{m_1 + m_2} & 0 & 0 & -\frac{m_1 z_1 + m_2(z_1 + L)}{(m_1 + m_2)g} & -\frac{m_2 L(z_1 + L)}{(m_1 + m_2)g} \end{bmatrix} \begin{bmatrix} x_1 \\ \theta \\ \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix}. \quad (4.19)$$

The determinant of the double mass IPM model is

$$\det = \begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix}, \quad (4.20)$$

and the transfer function for the double mass IPM is

$$G(s) = C(sI - A)^{-1}B = \begin{bmatrix} 1 - \frac{m_1 z_1 + m_2(z_1 + L)}{(m_1 + m_2)g} s^2 & \frac{m_1 L}{m_1 + m_2} - \frac{m_2 L(z_1 + L)}{(m_1 + m_2)g} s^2 \\ \frac{1}{s^3} & \frac{1}{s^3} \end{bmatrix}. \quad (4.21)$$

We note that the system does not have any zeros [Napoleon, N. Shigeki et al., 2002], [H. Dallali, M. Brown et al., 2009]. Equation (4.18) & (4.19) will be used in the gait trajectory generation and further discussion in Chapter 5.

4.3 Momentum and Motion

The transformation of the dynamic motion of the robot end effector (the feet and the hip) trajectories into an individual link joint trajectory can be carried out with the momentum calculation method [S. Kajita, F. Kanehiro et al., 2003]. As illustrated in Figure 4.11, the four reference frames of the robot are defined as $[\Sigma W \ \Sigma H \ \Sigma L \ \Sigma R]$, which denote the world frame, the pelvis frame, the left-foot frame and the right-foot frame respectively. The position and orientation of a point in space can be represented by a translational vector $p_{3 \times 1}$ and an orientation matrix $R_{3 \times 3}$ in the form of

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{p} \\ (\dot{R}R^T)^v \end{bmatrix}. \quad (4.22)$$

The foot velocity of the two legs can be expressed as

$$\begin{bmatrix} v_L \\ \omega_L \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & -\hat{r}_{HL} \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} v_H \\ \omega_H \end{bmatrix} + J_{legL} \dot{q}_{legL}, \quad (4.23)$$

$$\begin{bmatrix} v_R \\ \omega_R \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & -\hat{r}_{HR} \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} v_H \\ \omega_H \end{bmatrix} + J_{legR} \dot{q}_{legR},$$

where $[v_H, \omega_H]^T$ is the hip velocity, J_{legL} and J_{legR} are a (6×6) Jacobian matrix of the robot's leg, \dot{q}_{legL} and \dot{q}_{legR} are the (6×1) joint speed vector of the leg, and $r_{HL} = p_L - p_H$ and $r_{HR} = p_R - p_H$ are the position vectors. The $\hat{*}$ is an operator, which translates the (3×1) vector r_{HL} and r_{HR} into a (3×3) skew symmetric matrix \hat{r}_{HL} and \hat{r}_{HR} respectively. Referring to Figure 4.11, for a given hip and feet velocity, the leg joints angle can be determined as

$$\dot{q}_{legL} = J_{legL}^{-1} \begin{bmatrix} v_L \\ \omega_L \end{bmatrix} - J_{legL}^{-1} \begin{bmatrix} I_{3 \times 3} & -\hat{r}_{HL} \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} v_H \\ \omega_H \end{bmatrix}, \quad (4.24)$$

$$\dot{q}_{legR} = J_{legR}^{-1} \begin{bmatrix} v_R \\ \omega_R \end{bmatrix} - J_{legR}^{-1} \begin{bmatrix} I_{3 \times 3} & -\hat{r}_{HR} \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} v_H \\ \omega_H \end{bmatrix}.$$

The linear momentum $\mathcal{L}_{(3 \times 1)}$ and the angular momentum $\mathcal{A}_{(3 \times 1)}$ of the CoM of the robot can be computed as

$$\begin{bmatrix} \mathcal{L} \\ \mathcal{A} \end{bmatrix} = \overbrace{\begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\hat{r}_{HC} \\ 0 & \tilde{\mathbb{I}}_{3 \times 3} \end{bmatrix}}^{\text{Hip}} \begin{bmatrix} v_H \\ \omega_H \end{bmatrix} + \overbrace{\begin{bmatrix} M_L \\ H_L \end{bmatrix}}^{\text{Left Leg}} \dot{q}_{legL} + \overbrace{\begin{bmatrix} M_R \\ H_R \end{bmatrix}}^{\text{Right Leg}} \dot{q}_{legR}, \quad (4.25)$$

where m is the robot's total mass, $\tilde{\mathbb{I}}_{3 \times 3}$ is the inertia matrix with respect to the CoM, \hat{r}_{HC} is the skew symmetric matrix of r_{HC} , M and H are the inertial matrices of the joints related to the linear and the angular momentum respectively. To generate a stable walking motion in sagittal plane, the angular momentum of the CoM_x is set to zero, that is no rotation, and the translation movement of the CoM_x can be obtained from Equation (4.25). By definition $\mathcal{L} = mC\dot{O}M_x$, and we set $\mathcal{A} = 0$, yields

$$\begin{bmatrix} mC\dot{O}M_x \\ 0 \end{bmatrix} = \overbrace{\begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\hat{r}_{HC} \\ 0 & \tilde{\mathbb{I}}_{3 \times 3} \end{bmatrix}}^{\text{Hip}} \begin{bmatrix} v_H \\ \omega_H \end{bmatrix} + \overbrace{\begin{bmatrix} M_L \\ H_L \end{bmatrix}}^{\text{Left Leg}} \dot{q}_{legL} + \overbrace{\begin{bmatrix} M_R \\ H_R \end{bmatrix}}^{\text{Right Leg}} \dot{q}_{legR}. \quad (4.26)$$

For a given foot velocity ($[v_L, \omega_L]^T, [v_R, \omega_R]^T$), from equation (4.26), the hip velocity in sagittal plane can be derived as

$$\begin{bmatrix} v_H \\ \omega_H \end{bmatrix} = \begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\hat{r}_{HC} \\ 0 & \tilde{\mathbb{I}}_{3 \times 3} \end{bmatrix}^{-1} \left(\begin{bmatrix} mC\dot{O}M_x \\ 0 \end{bmatrix} - \begin{bmatrix} M_L \\ H_L \end{bmatrix} \dot{q}_{legL} - \begin{bmatrix} M_R \\ H_R \end{bmatrix} \dot{q}_{legR} \right). \quad (4.27)$$

Equation (4.25) and (4.26) are useful in generating robot hopping and running gait.

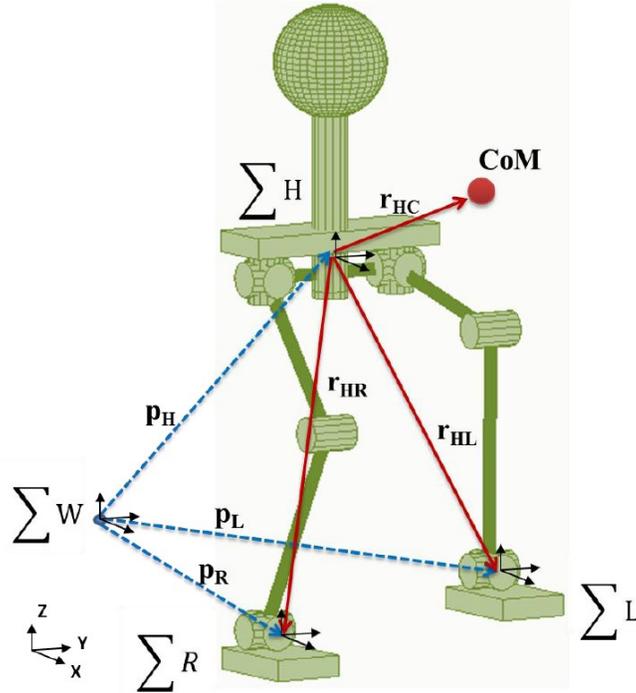


Figure 4.11. Momentum and motion reference coordinate.

4.4 Motion Path and Trajectory Generation

In this section, the motion path and trajectory generation algorithm for joint position, feet movement and ZMP/CoM trajectory will be discussed in detail.

4.4.1 Joint Trajectory Generation with Interpolation Method

The robot's joint trajectory is generated using seven degrees of polynomial to ensure that all the joints start and stop at the same time. The position, velocity, acceleration and jerk parameters are all zero when it reaches the desired angular position. For example, to move the angular position of a joint with a zero jerk start-stop path for $q(t_0) = 10^0$ and $q(t_f) = 45^0$ in $T=1s$, the unknown coefficient $q_0, a_1, a_2, \dots, a_7$ can be found with the following set of polynomial equations by setting the eight boundary conditions. The motion trajectories are depicted in Figure 4.12. For

$$q(t) = a_0 + a_1 \frac{t}{T} + a_2 \left(\frac{t}{T}\right)^2 + a_3 \left(\frac{t}{T}\right)^3 + a_4 \left(\frac{t}{T}\right)^4 + a_5 \left(\frac{t}{T}\right)^5 + a_6 \left(\frac{t}{T}\right)^6 + a_7 \left(\frac{t}{T}\right)^7, \quad (4.28)$$

and boundary conditions

$$\begin{aligned} q(t_0) &= q_0 = 10, & \dot{q}(t_0) &= 0, & \ddot{q}(t_0) &= 0, & \dddot{q}(t_0) &= 0. \\ q(t_f) &= q_f = 45, & \dot{q}(t_f) &= 0, & \ddot{q}(t_f) &= 0, & \dddot{q}(t_f) &= 0. \end{aligned} \quad (4.28)$$

The polynomial functions, with $u = \frac{t}{T}$, yields

$$\begin{aligned} q(t) &= a_0 + a_1 u + a_2 u^2 + a_3 u^3 + a_4 u^4 + a_5 u^5 + a_6 u^6 + a_7 u^7, \\ \dot{q}(t) &= a_1 + 2a_2 u + 3a_3 u^2 + 4a_4 u^3 + 5a_5 u^4 + 6a_6 u^5 + 7a_7 u^6, \\ \ddot{q}(t) &= 2a_2 + 6a_3 u + 12a_4 u^2 + 20a_5 u^3 + 30a_6 u^4 + 42a_7 u^5, \\ \dddot{q}(t) &= 6a_3 + 24a_4 u + 60a_5 u^2 + 120a_6 u^3 + 210a_7 u^4. \end{aligned} \quad (4.29)$$

The coefficients (a_0 to a_7) with $t_0 = 0$ and $t_f = 1$, can be solved with the following matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} q_0 = 10 \\ 0 \\ 0 \\ 0 \\ q_f = 45 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (4.30)$$

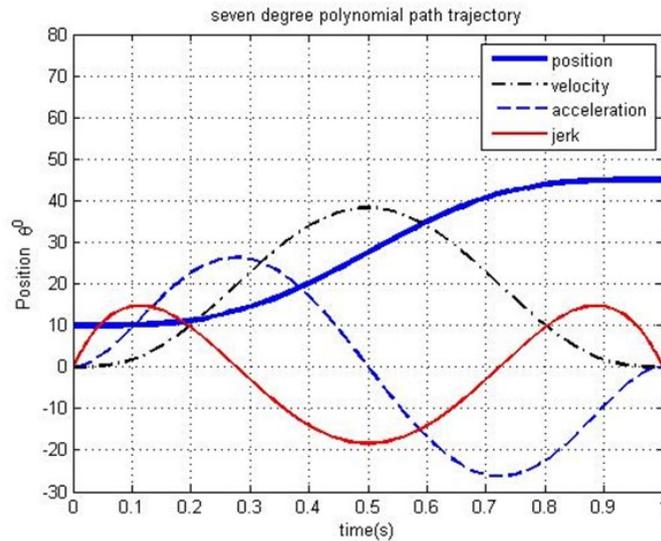


Figure 4.12. Interpolation and trajectory pattern simulation.

4.4.2 Foot Trajectory

The walking gait as illustrated in Figure 4.13, can be seen as a periodic stance of the swing of both the legs. One swing makes up a complete cycle of walking. Figure 4.14 shows The basic parameters for a biped walking are step length S , pelvis height Z , foot lifting height H and frontal-shift F .

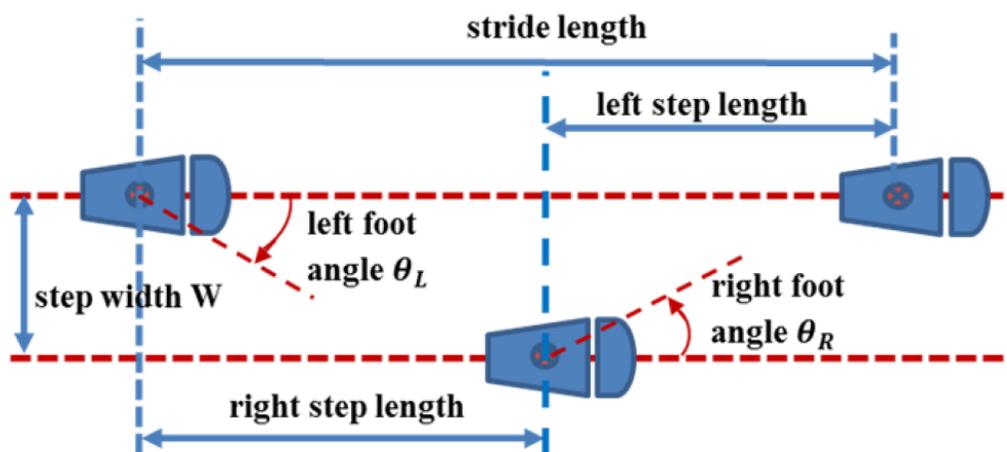


Figure 4.13. Typical robot walking cycle.

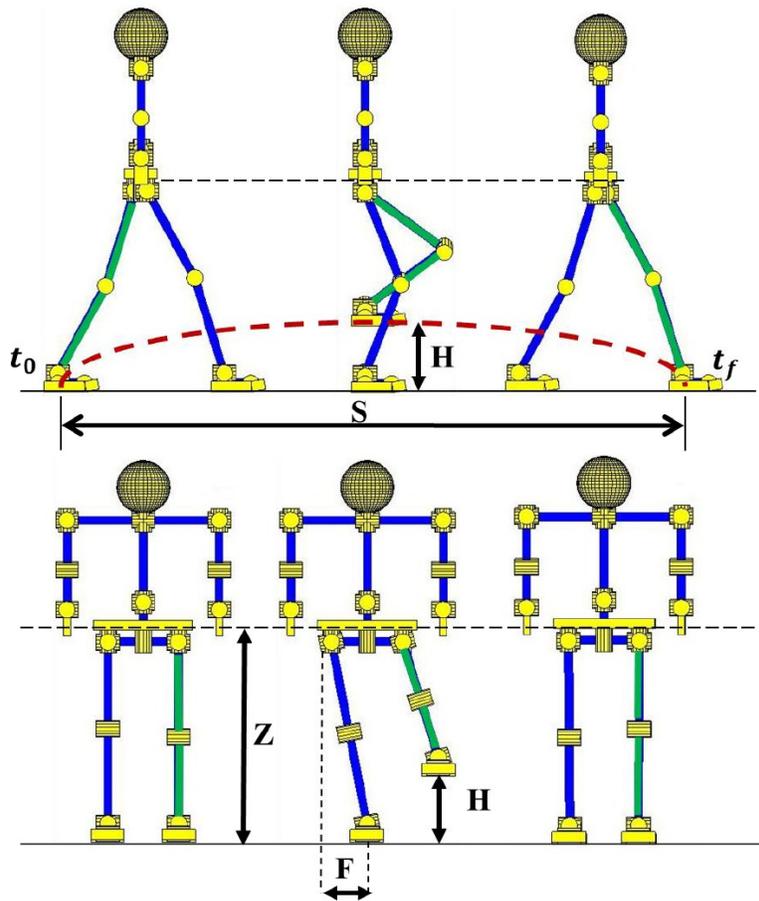


Figure 4.14. Typical robot walking gait.

4.4.2.1 Foot Trajectory for Walking on Flat Ground

The swing foot trajectory of a robot walking on flat ground may be modelled by a simple cycloid function. The equations that describe the parabolic trajectory of the swing foot are given by

$$\begin{aligned}
 x(t) &= -S \cos(\omega t), \\
 Z(t) &= \frac{H}{2} [1 - \cos(2\omega t)], \quad \text{for } 0 \leq t \leq T. \quad (4.31)
 \end{aligned}$$

Where S is the stride, H is the maximum foot height, T is the one step period and the stride frequency is $\omega = \frac{\pi}{T}$. These functions are commonly used to generate a robot swing foot trajectory. A sample plot of $T = 1s, S = 0.2m, H = 0.1m$ is shown in Figure 4.15.

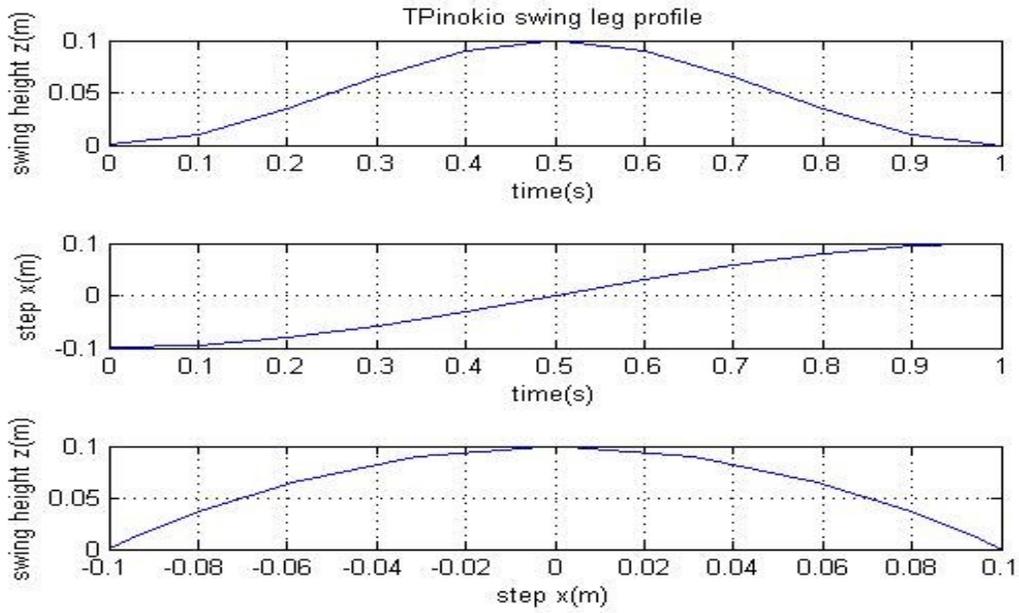


Figure 4.15. Trajectory profile with cycloid function.

4.4.2.2 Foot Trajectory for Walking on Uneven Ground

The swing foot trajectory of a robot walking on uneven surface may not be realised by using simple cycloid functions, especially if the robot is climbing stairs or stepping over an obstacle. Under such situations, we may need to define a path trajectory with intermediate points, and the seven degrees of polynomial with one intermediate point can be realised by setting the eight boundary conditions with

$$\begin{aligned}
 q(t) &= a_0 + a_1u + a_2u^2 + a_3u^3 + a_4u^4 + a_5u^5 + a_6u^6 + a_7u^7 \Big|_{u=\frac{t}{T}}, \\
 q(t_0) &= q_0, \quad \dot{q}(t_0) = 0, \quad \ddot{q}(t_0) = 0, \\
 q(t_1) &= q_1, \\
 q(t_f) &= q_f, \quad \dot{q}(t_f) = 0, \quad \ddot{q}(t_f) = 0, \quad \ddot{\ddot{q}}(t_f) = 0.
 \end{aligned} \tag{4.32}$$

For example, for time T from 0 to 1sec, to generate a swing profile with the following parameters:

axis	start point/time	midpoint/time	end point/time
x-axis	-0.1m, t=0s	0m, t=0.5s	0.1m, t=1s
z-axis	0m, t=0s	0.15m, t=0.3s	0m, t=1s

for x-axis, to determine the coefficients (a_0 to a_7) with $t_0 = 0, t_1 = 0.5$ and $t_2 = 1$, the coefficients matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5^1 & 0.5^2 & 0.5^3 & 0.5^4 & 0.5^5 & 0.5^6 & 0.5^7 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0 \\ 0 \\ 0.1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (4.33)$$

and for z-axis, the coefficients matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3^1 & 0.3^2 & 0.3^3 & 0.3^4 & 0.3^5 & 0.3^6 & 0.3^7 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.15 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (4.34)$$

The trajectory profile is as shown in Figure 4.16.

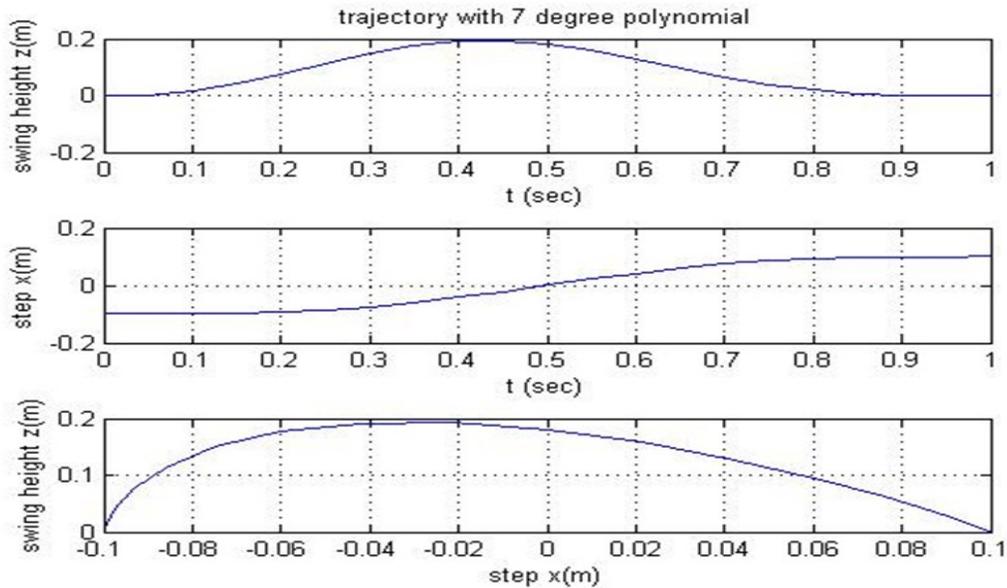
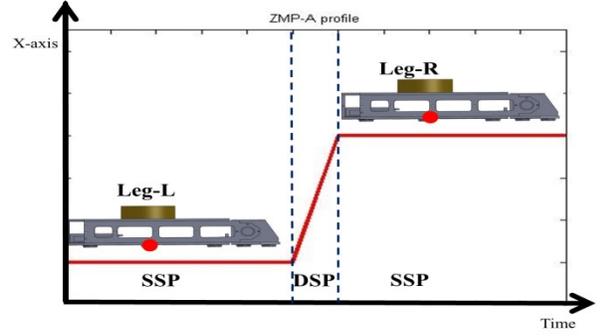
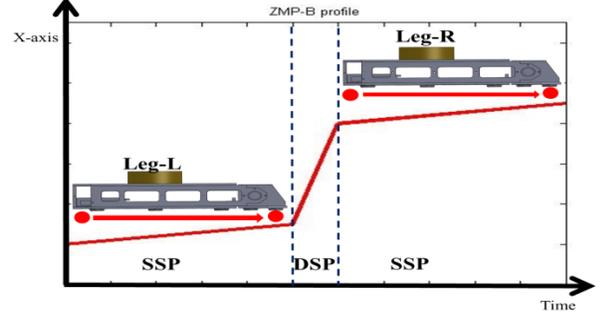
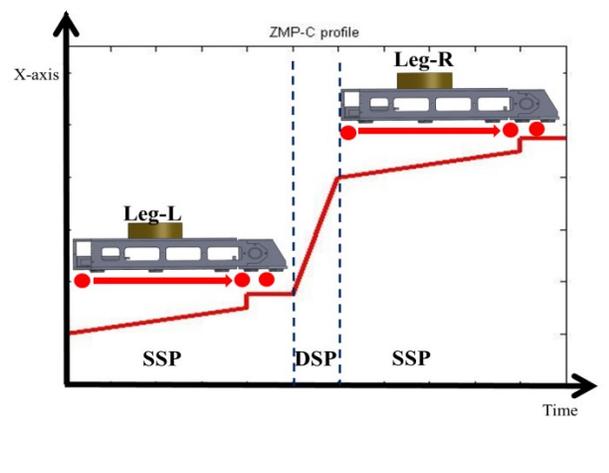


Figure 4.16. The trajectory profile of a seven degree polynomial function.

4.4.3 ZMP Trajectory for Robot with Flat-foot or Toe-foot Walking

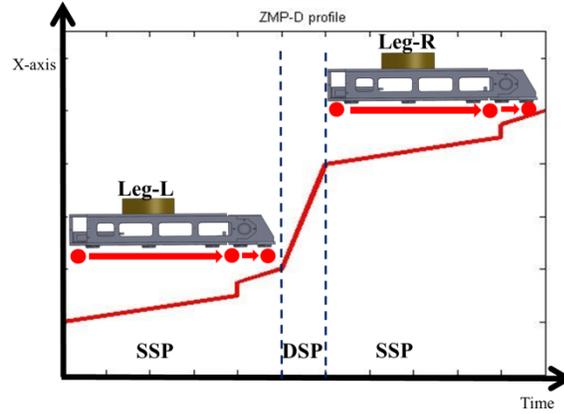
Referring to Table 4.2. There are various types of ZMP trajectories. For flat-foot walking, ZMP-A trajectory is suitable. But for fast walking, hopping forward or running, either ZMP-B, ZMP-C or ZMP-D trajectory is more suitable to produce a longer strike.

Table 4.2. ZMP trajectory and profile.

Different type of ZMP trajectory	Sagittal Plane (x-axis) ZMP profile
<p><u>ZMP-A trajectory:</u> The ZMP is always position at the center of the stance foot (Leg-L) support polygon during the SSP and switches to the center point of the foot support convex polygon (SCP) of the next support foot (Leg-R) during the double support phase (DSP).</p>	 <p>The graph shows the ZMP position on the x-axis over time. It is divided into three phases: SSP (Single Support Phase) for Leg-L, DSP (Double Support Phase), and SSP for Leg-R. During the first SSP, the ZMP is constant at the center of the left foot. At the start of the DSP, it jumps to the center of the right foot and remains constant through the second SSP. Red dots on the robot diagrams indicate the ZMP position at each phase.</p>
<p><u>ZMP-B trajectory:</u> At the start of SSP, the ZMP is positioned at the heel of the foot support polygon (Leg-L) and shift linearly to the front edge to the toe of the SCP during SSP, and switches to the center point of the foot support polygon of the next support foot (Leg-R) during the DSP.</p>	 <p>The graph shows the ZMP position on the x-axis over time. During the first SSP (Leg-L), the ZMP starts at the heel and moves linearly to the toe. At the start of the DSP, it jumps to the center of the right foot and remains constant through the second SSP. Red dots on the robot diagrams indicate the ZMP position at each phase.</p>
<p><u>ZMP-C trajectory:</u> Robot with feet-toe, it is more efficient in walking and it is able to achieve longer strike. At the start of SSP, the ZMP is positioned at the heel of the stance feet (Leg-L) SCP, during the SSP, it shift linearly to the front edge of the sole support, and switches to the center point of the toe support convex hull, and switches to the heel of the foot SCP of the next support foot (Leg-R) during the DSP.</p>	 <p>The graph shows the ZMP position on the x-axis over time. During the first SSP (Leg-L), the ZMP starts at the heel and moves in a series of steps to the toe. At the start of the DSP, it jumps to the center of the right foot and remains constant through the second SSP. Red dots on the robot diagrams indicate the ZMP position at each phase.</p>

ZMP-D trajectory:

Suitable for hopping, fast walking or even running. At the start of SSP, the ZMP is positioned at the heel of the stance foot (Leg-L) SCP, during the SSP, the ZMP shifts linearly to the front edge of the sole, and subsequently switches to the back edge of the toe SCP and shifts linearly to the front tip edge of the toe, and start to rotate around the toe-tip, before switching over to the heel of the foot SCP of the next support foot (Leg-R) during the DSP.



4.4.4 ZMP Pattern Generation for Different Ground Condition

The ZMP of the sagittal and frontal direction can be generated with the following generic set of algorithms [Y. Choi, D. Kim et al., 2007], [T.H.S. Li, Y.T. Su et al, 2012] for walking on different terrains:

$$Y_{zmp} = \begin{cases} (-1)^{n-1} \frac{(E-C_y)}{t_d} (t - (n-1)T) + \text{mod}(n, 2)C_y + (1 - \text{mod}(n, 2))E, \\ \text{for } (n-1)T \leq t < (n-1)T + t_d. \\ (-1)^{n-1} E_y \sin\left(\frac{\pi(t-(n-1)T-t_d)}{T-t_d}\right) + (1 - \text{mod}(n, 2))C_y + \text{mod}(n, 2)E, \\ \text{for } (n-1)T + t_d \leq t < nT. \end{cases} \quad (4.34)$$

$$X_{zmp} = \begin{cases} \frac{(S_x - E_x)}{t_d} (t - (n-1)T) + (n-1)S_x + E_x + C_x, \\ \text{for } (n-1)T \leq t < (n-1)T + t_d. \\ \frac{E_x}{T-t_d} (t - (n-1)T - t_d) + nS_x + C_x, \\ \text{for } (n-1)T + t_d \leq t < nT. \end{cases}$$

Where n is number of step, S_x is the step size, E, E_x, C_x , are some fine-tuning parameters to change the shape of the ZMP trajectory waveform. The trajectory parameters are as depicted in Figure 4.17 and the generic trajectory is as shown Figure 4.18.

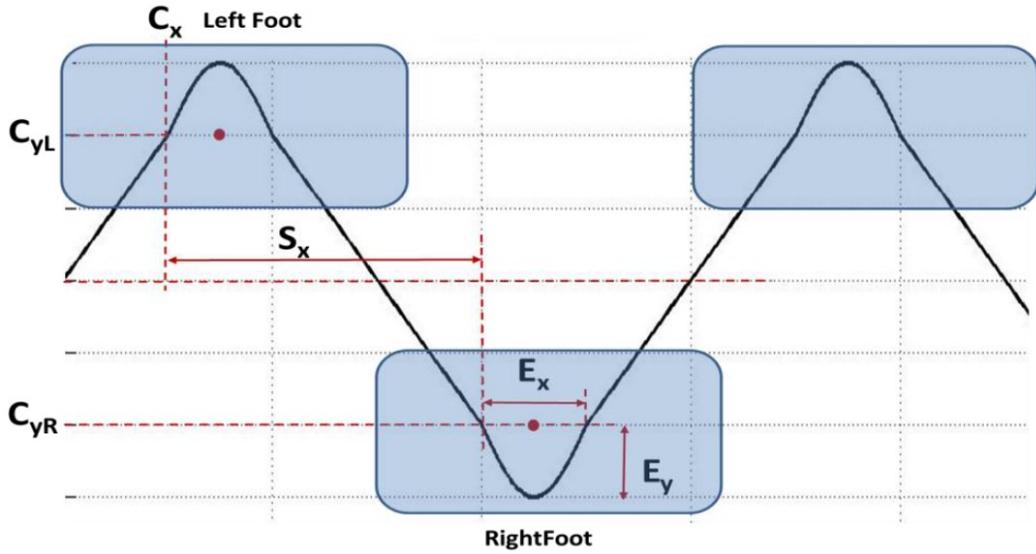


Figure 4.17. ZMP trajectory parameters.

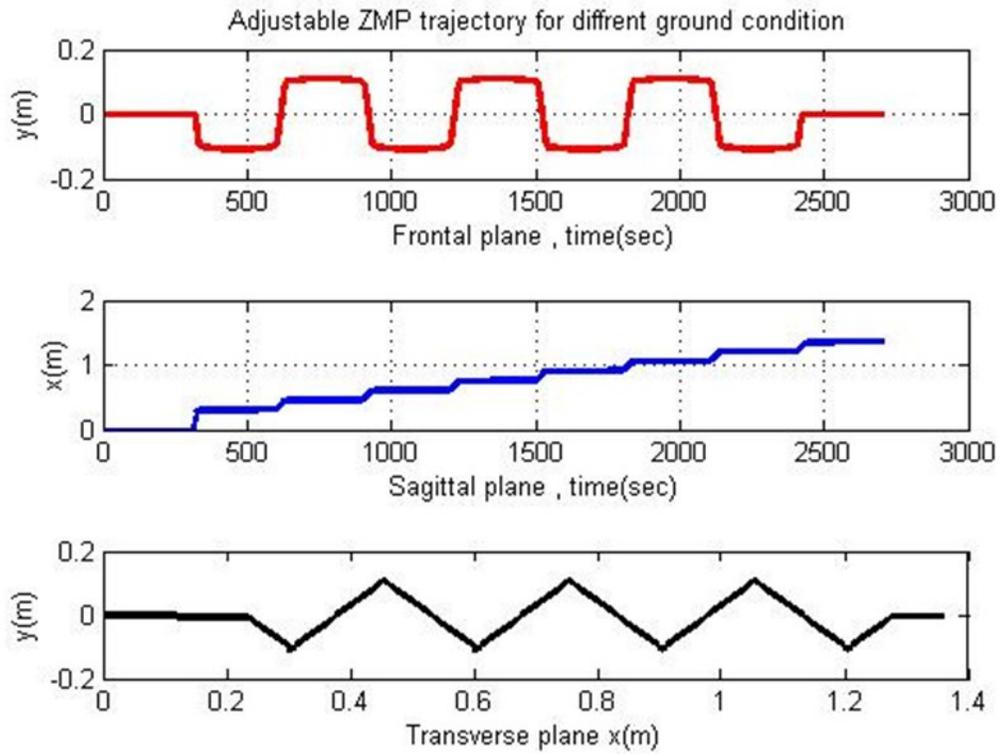


Figure 4.18. Generic ZMP trajectory.

4.5 Chapter Summary

In this chapter, the kinematics and dynamics model of our TPinokio robot is presented and discussed. Iterative method is adopted to solve the inverse kinematics problems because the advantage towards the close-form method is that we can add extra links. For example, adding extra link to the fingers, eyes or toe to increase the DoF of our robot can be done without too much modification to the forward and inverse kinematic equations. The accuracy of the forward and inverse kinematics is verified through simulations. A simple inverted pendulum dynamic model is as effective as a multi-mass or whole body dynamic model. Seven degrees polynomial is adopted for path trajectory planning to ensure that the all the joints start and stop at the same time. Furthermore, the position, velocity, acceleration and jerk components are all zero when the trajectory reaches the desired angular position and this will improve the walking stability. A generic ZMP generating equation is also developed and by tuning the waveform parameters, this ZMP trajectory can be easily modified to different shape for walking on different terrains.

Chapter 5

Control and Walking Pattern Generation

An admissible and optimal trajectory reference signal must first be generated before one can implement a controller for a bipedal robot to be stable in walking. In this chapter, we will consider the control issues of the bipedal robot TPinokio, its kinematics and dynamics, its gait synthesis and implementation. The goal is to generate efficient walking pattern for the robot to walk stably either in static or dynamic-mode on flat surface, step over obstacle, ascending and descending stairs. Control of bipedal walking can be classified into two main tasks:

- To control the position of the bipedal robot CoM and ZMP position and its trajectory to maintain a stable walking posture. This task can be achieved by various tracking algorithms; and
- To control the position, orientation and motion trajectory of the moving legs and body to produce minimum-jerk walking posture.

This chapter presents the proposed motion control algorithms and description of the walking gait generation methods.

5.1 Augmented Model Predictive Control

The use of the Augmented Model Predictive Control (AMPC) technique to control TPinokio is motivated by its capability in handling the control or the state variables constraints. The three main terms in the AMPC are prediction horizon, control horizon and moving horizon window respectively. This approach is suitable in designing a stable walking gait pattern and solving the control problems of a bipedal that is subjected to disturbances and constraints in an unstructured

environment. The human walking behavior provides some observation guide for our controller design:

- Perfect joint trajectory tracking from step-to-step is not necessary.
- Adapting to its environment and able to avoid obstacle.
- Walking from point to point with minimum energy consumption.
- Have good capability of anticipation and prediction during locomotion.

These behaviors fit nicely into the AMPC approach which minimizes the robot body CoM jerks ($\ddot{x}_{com}, \ddot{y}_{com}$), while tracking its ZMP position (x_{zmp}, y_{zmp}) as near as possible to the desired trajectory. For example, tracking the reference input from the vision system or the path planning controller. The control strategy is illustrated in Figure 5.1, the optimized gaits and the constraints performance criteria were introduced to define the gait pattern generation as an optimization problem with constraints. The optimization computes the predicted vector $U(k)$ over the control horizon, only the first component $u(k+1)$ is feedback to the robot input to ensure a smoother motion in an obstacle-filled environment. Then the optimization process restarts.

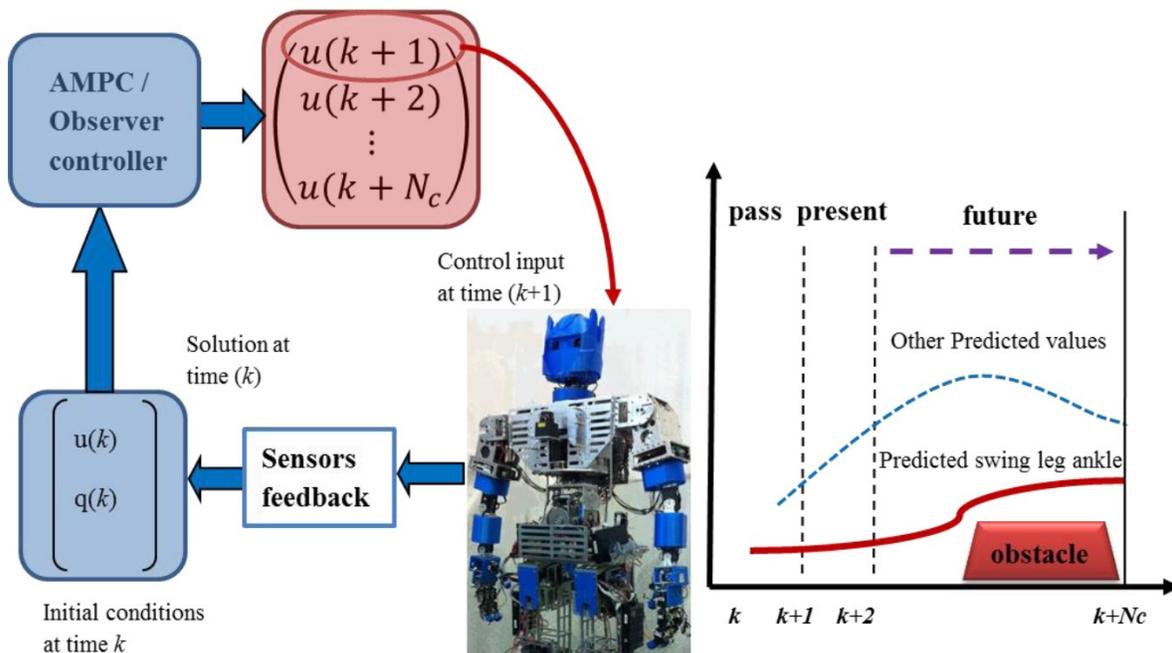


Figure 5.1. Typical AMPC set up.

5.1.1 Linear Inverted Pendulum Model

5.1.1.1 Discrete LIPM model

As discussed in Chapter 4, the 3D-LIPM model ZMP equation for the sagittal (xz-plane) and the frontal (yz-plane) can be expressed as

$$x_{zmp} = x_{com} - \frac{z_{com}}{g} \ddot{x}_{com} \quad , \quad y_{zmp} = y_{com} - \frac{z_{com}}{g} \ddot{y}_{com} . \quad (5.1)$$

Note that the above equations are decoupled, and they can be expressed in 2D-LIPM and analyzed independently. Let now define a new variable u (system jerk) in the following form:

$$u_x = \frac{d}{dt} \ddot{x}_{com} = \dddot{x}_{com} \quad , \quad u_y = \frac{d}{dt} \ddot{y}_{com} = \dddot{y}_{com} . \quad (5.2)$$

Combining equation (5.1) and (5.2), the ZMP equation can be re-arranged into the following state-space form:

$$\begin{bmatrix} \dot{x}_{com} \\ \dot{y}_{com} \\ \ddot{x}_{com} \\ \ddot{y}_{com} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \quad , \quad \begin{bmatrix} \dot{y}_{com} \\ \dot{y}_{com} \\ \ddot{y}_{com} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{com} \\ \dot{y}_{com} \\ \ddot{y}_{com} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_y , \quad (5.3)$$

$$x_{zmp} = \begin{bmatrix} 1 & 0 & -\frac{z_{com}}{g} \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix} \quad , \quad y_{zmp} = \begin{bmatrix} 1 & 0 & -\frac{z_{com}}{g} \end{bmatrix} \begin{bmatrix} y_{com} \\ \dot{y}_{com} \\ \ddot{y}_{com} \end{bmatrix} .$$

The set of equations in (5.2) and (5.3) can be applied to synthesize the dynamic characteristic of the 3D-LIPM system, which is a reasonable approximation as compared to other more complicated approaches. In the subsequent sections, the LIPM model will be analyzed in the xz-plane (sagittal plane). The same methodologies can be applied to the yz-plane (frontal plane). The above equations (5.3) are discretized with constant time T , and $t = kT$ for $k = 1, 2, 3 \dots$. The discrete form of the xz-plane can be derived as:

$$\begin{bmatrix} x_{com}(k+1) \\ \dot{x}_{com}(k+1) \\ \ddot{x}_{com}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{com}(k) \\ \dot{x}_{com}(k) \\ \ddot{x}_{com}(k) \end{bmatrix} + \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix} u_x(k),$$

$$x_{zmp}(k) = \begin{bmatrix} 1 & 0 & -\frac{z_{com}}{g} \end{bmatrix} \begin{bmatrix} x_{com}(k) \\ \dot{x}_{com}(k) \\ \ddot{x}_{com}(k) \end{bmatrix}. \quad (5.4)$$

5.1.1.2 Characteristic of the Discrete LIMP Model

The root-locus plot of the continuous time LIPM, referring to equation (5.3), is as depicted in Figure 5.2, showing an undesired *zero* at the RHP. The discrete time poles-zeros plot, referring to equation (5.4), is as depicted in Figure 5.3, showing the undesired *zero* appears outside the unit circle. As such, we can conclude that the LIPM model is a non-minimum phase (NMP) system. This undesired zero will produce a reverse action undershoot transient response [B.H. Jesse, S.B. Dennis et al., 2007] to a step input. Care should be taken to address this initial ‘water-bed’ effect, as this may cause the bipedal robot to lose balance and fall.

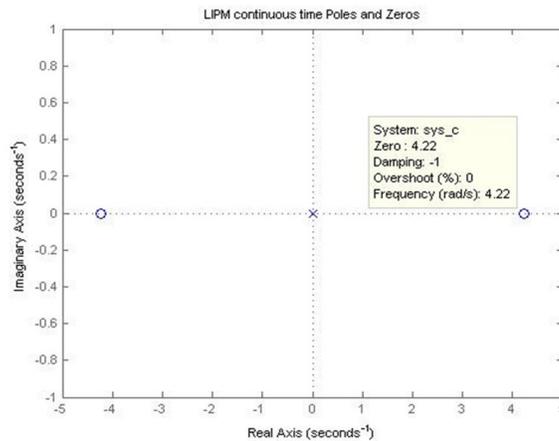


Figure 5.2. Continuous time poles and zeros plot.

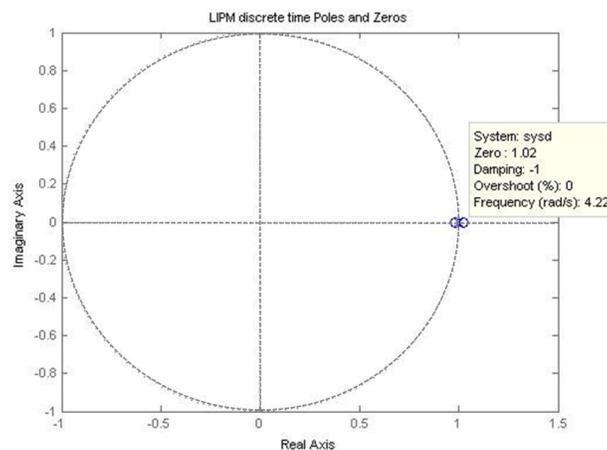


Figure 5.3. Discrete time poles and zeros plot.

5.1.2 Augmented Model

Augmented Model Predictive Control (AMPC) state-space model tracking system design will be presented in this section. The solutions is recursive, as such it is suitable for implementation with microcontrollers that have limited computing power and memory storage. In addition, our approach does not need to solve *Riccati equation* (comparison with most of the optimization methods), therefore reducing the complexity of the solving process.

The proposed AMPC model is an augmented state-space model with an embedded integrator [L. Wang, 2009]. The CoM is represented in the state variables. The set of equations (5.4) can be re-written into the standard state-space form:

$$\begin{aligned} x_D(k+1) &= A_D x_D(k) + B_D u(k), \\ y(k) &= C_D x_D(k). \end{aligned} \quad (5.5)$$

Where x_D is the discrete-time state variable of the CoM, u is the discrete-time control input variable of the system jerk, and y is the discrete-time ZMP output. To augment equation (5.5), by taking a different operation on both sides of the equation, we obtain

$$x_D(k+1) - x_D(k) = A_D(x_D(k) - x_D(k-1)) + B_D(u(k) - u(k-1)). \quad (5.6)$$

The difference in the state variable and the difference in the control variable are given by

$$\begin{aligned} \Delta x_D(k+1) &= x_D(k+1) - x_D(k), \\ \Delta x_D(k) &= x_D(k) - x_D(k-1), \\ \Delta u(k) &= u(k) - u(k-1), \\ \Delta x_D(k+1) &= A_D \Delta x_D(k) + B_D \Delta u(k). \\ y(k+1) - y(k) &= C_D(x_D(k+1) - x_D(k)) = C_D \Delta x_D(k+1) = C_D A_D \Delta x_D(k) + C_D B_D \Delta u(k) \\ y(k+1) &= y(k) + C_D A_D \Delta x_D(k) + C_D B_D \Delta u(k) \end{aligned} \quad (5.7)$$

Combining equation (5.5),(5.6) and (5.7), we obtain the augmented state-space model:

$$\begin{aligned} \overbrace{\begin{bmatrix} \Delta x_D(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} &= \overbrace{\begin{bmatrix} A_D & O_D^T \\ C_D A_D & 1 \end{bmatrix}}^A \overbrace{\begin{bmatrix} \Delta x_D(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_D \\ C_D B_D \end{bmatrix}}^B \Delta u(k), \\ y(k) &= \overbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}}^C \overbrace{\begin{bmatrix} \Delta x_D(k) \\ y(k) \end{bmatrix}}^c, \end{aligned} \quad (5.8)$$

where $O_D = [0 \ 0 \ 0]$ is the dimension of the state variable, and the triplet (A, B, C) is the augmented model.

5.1.3 Prediction of the Robot CoM (the state) and the ZMP (the output variable)

The control input (system jerk) is denoted as

$$\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_c-1), \quad (5.9)$$

where N_c is the control horizon and N_p is the length of the prediction horizon. The future CoM state variables are predicted for N_p samples. Let denote the future CoM state variables as

$$x(k+1|k), x(k+2|k), \dots, x(k+m|k), \dots, x(k+N_p|k). \quad (5.10)$$

Note that $x(k+m|k)$ is the predicted future state variable at $(k+m)$ with the given current system state variables $x(k)$. Normally, we set the control horizon N_c to $N_c \leq N_p$.

The stage variables can be computed recursively as follows:

$$\begin{aligned} x(k+1|k) &= A_D x(k) + B_D \Delta u(k) \\ x(k+2|k) &= A_D x(k+1|k) + B_D \Delta u(k+1) = A_D^2 x(k) + A_D B_D \Delta u(k) + B_D \Delta u(k+1) \\ x(k+N_p|k) &= A_D^{N_p} x(k) + A_D^{N_p-1} B_D \Delta u(k) + A_D^{N_p-2} B_D \Delta u(k+1) + \dots \\ &\quad + A_D^{N_p-N_c} B_D \Delta u(k+N_c-1). \end{aligned}$$

Similarly, all the predicted output values are based on current information of the state value $x(k)$ and the future predicted control values $\Delta u(k+j)$, where $j = 0, 1, 2, \dots, N_c-1$, can be arranged into the following vectors form:

$$\begin{aligned} Y &= [y(k+1|k) \quad y(k+2|k) \quad y(k+3|k) \quad \dots \quad y(k+N_p|k)]^T, \\ \Delta U &= [\Delta u(k) \quad \Delta u(k+1) \quad \Delta u(k+2) \quad \dots \quad \Delta u(k+N_c-1)]^T. \end{aligned} \quad (5.11)$$

The output variables are computed recursively as:

$$\begin{aligned}
y(k+1|k) &= C_D A_D x(k) + C_D B_D \Delta u(k) \\
y(k+2|k) &= C_D A_D^2 x(k) + C_D A_D B_D \Delta u(k) + C_D B_D \Delta u(k+1) \\
y(k+N_p|k) &= C_D A_D^{N_p} x(k) + C_D A_D^{N_p-1} B_D \Delta u(k) + C_D A_D^{N_p-2} B_D \Delta u(k+1) + \dots \\
&\quad + C_D A_D^{N_p-N_c} B_D \Delta u(k+N_c-1).
\end{aligned}$$

The dimensions of Y and U are N_p and N_c respectively. We can arrange the parameters into the following matrix form:

$$Y = Fx(k) + \Phi \Delta U, \quad (5.12)$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix},$$

and

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2 B & CAB & CB & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1} B & CA^{N_p-2} B & CA^{N_p-3} B & \dots & CA^{N_p-N_c} B \end{bmatrix}.$$

5.1.4 ZMP Tracking and Optimization

This optimization process is then translated to find an optimal control vector U such that the error between the input reference ZMP set-point trajectory and the CoM output trajectory is minimized. The vector that contains the set-point data information is written as

$$R_s = \overbrace{[1 \quad 1 \dots \quad 1]^T}^{N_p} r(k+t_a) = V_s r(k+t_a), \quad (5.13)$$

where $V_s = \overbrace{[1 \quad 1 \dots \quad 1]^T}^{N_p}$ and t_a is the adjustable feed forward time-shift constant respectively.

The cost function J is defined as

$$J = (R_s - Y)^T(R_s - Y) + \Delta U^T \bar{R} \Delta U, \quad (5.14)$$

where the first term is linked to the objective of minimizing the errors between the predicted ZMP output and the reference ZMP set point, and the second term reflects the consideration given to the rate of change in the system jerk (ΔU) when the objective function J is made as small as possible.

The diagonal matrix $\bar{R} = r_w I_{N_c \times N_c}$ ($r_w \geq 0$) is used as a tuning parameter to improve the closed-loop performance. For the case where $r_w = 0$, the goal would be solely to make the ZMP error $(R_s - Y)^T(R_s - Y)$ as small as possible. For the case of a large r_w , the cost function is interpreted as it is carefully consider how large the rate of change in the system jerk ΔU might be and cautiously reduce the ZMP error. To determine the optimal ΔU which minimizes J , we substitute equation (5.12) into (5.14) to obtain

$$J = (R_s - Fx(k))^T (R_s - Fx(k)) - 2\Delta U^T \Phi^T (R_s - Fx(k)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U. \quad (5.15)$$

The minimum J is computed by

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k)) + 2(\Phi^T \Phi + \bar{R}) \Delta U = 0. \quad (5.16)$$

By re-arranging equation (5.16), the optimal solution can be determined as

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (V_s r(k + t_a) - Fx(k)). \quad (5.17)$$

We note that the optimal solution of the control signal is linked to the ZMP reference set-point signal $r(k + t_a)$ and the state variable CoM by equation (5.17). Applying the AMPC concept, which is only the first value of the control parameters ΔU at time k is taken to be the subsequent control input variable, from equation (5.17) we obtain

$$\begin{aligned} \Delta u(k) &= \overbrace{[1 \quad 0 \dots 0]}^{N_c} (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (V_s r(k + t_a) - Fx(k)) \\ &= K_y r(k + t_a) - K_{mpc} x(k), \end{aligned}$$

$$(5.18)$$

where K_y and K_{mpc} is the first element of $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T V_s$ and first row of $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T F$ respectively. With K_{mpc} represent the state feedback control gain, equation (5.18) resemble a linear time-invariant system. As such, for the AMPC state variables expression $x(k+1) = Ax(k) + B\Delta u(k)$, and substituting (5.18) into the augmented system equation, the closed-loop system can be determined as

$$x(k+1) = Ax(k) - BK_{mpc}x(k) + BK_y r(k+t_a) = (A - BK_{mpc})x(k) + BK_y r(k+t_a). \quad (5.19)$$

We observe that K_y is also the last element of K_{mpc} , and the states $x(k_i) = [\Delta x_D(k)^T \ y(k)]^T$, and $K_{mpc} = [K_x \ K_y]$, with K_x and K_y being the gain variables of $\Delta x_D(k)$ and $y(k)$ respectively. Figure 5.4 shows the predictive system closed-loop block diagram, where z^{-1} and $\frac{1}{1-z^{-1}}$ denotes the backward shift operator and discrete-time integrator respectively. The closed-loop characteristic equation is

$$\det(\lambda I - (A - BK_{mpc})) = 0. \quad (5.20)$$

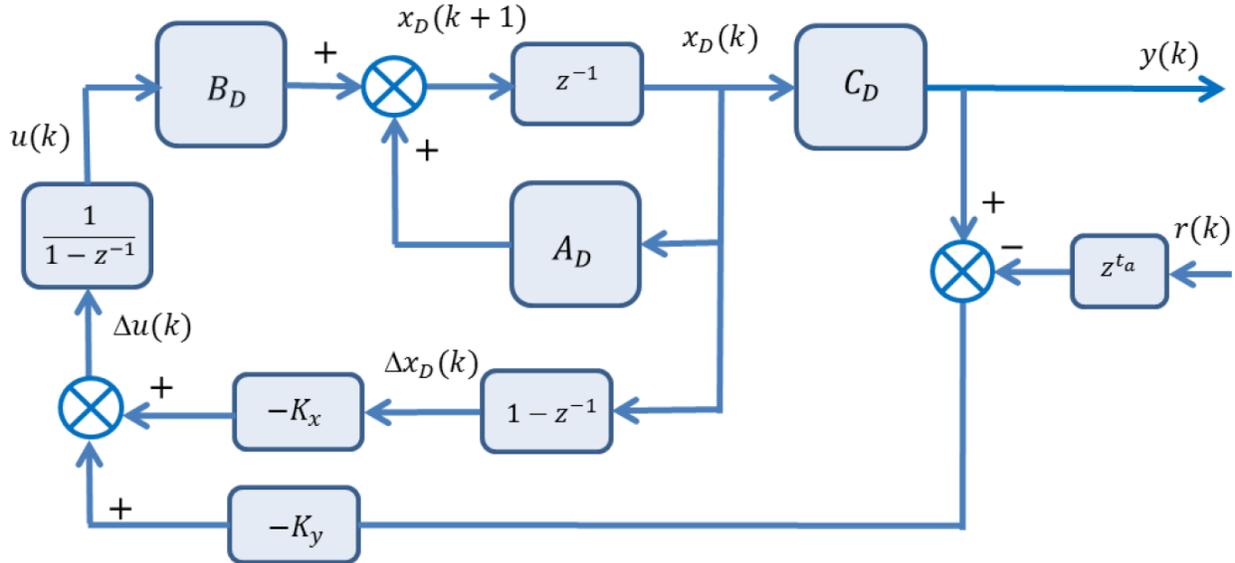


Figure 5.4. Block diagram of a discrete time AMPC.

The AMPC p-z plot, as depicted in Figure 5.5, indicated that there is a *zero* at location $z=1.02$, so the undershoot transient waterbed effect is expected.

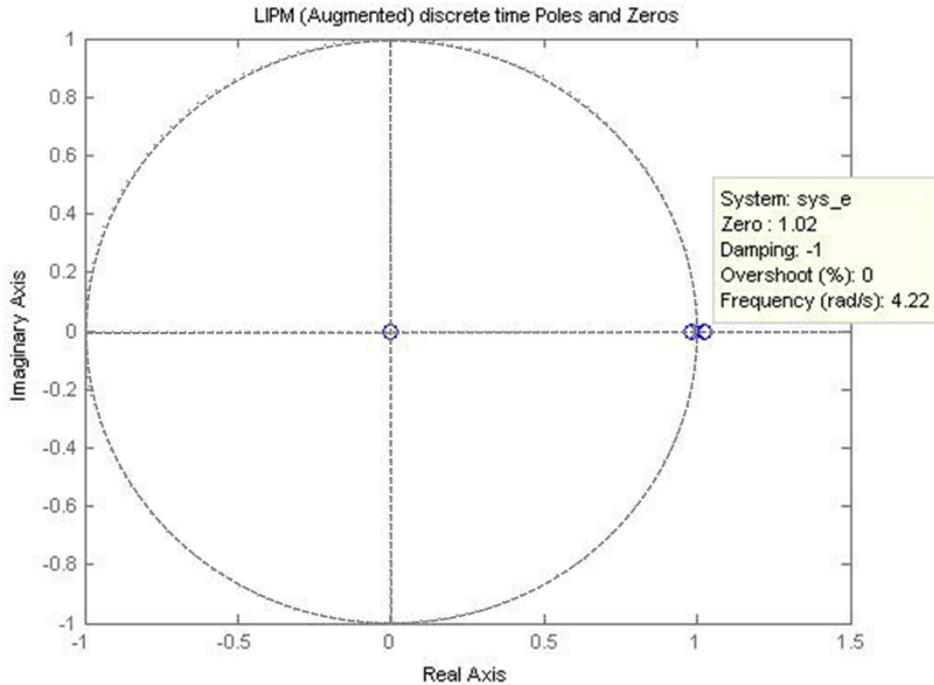


Figure 5.5. P-Z plot of a discrete time LIPM-AMPC system.

5.1.5 Simulation

Referring to Figure 5.6, the simulation shows good tracking of the desired ZMP and CoM trajectory. The robot parameters are: $T=5\text{ms}$, $Z_{\text{com}}=0.55\text{m}$. The AMPC parameters are: $N_c = 100$, $N_p = 300$, $t_a = 80$, $r_w = 10^{-5}$. As discussed in section 5.1.4, the undershoot transient response is expected because of the *zero* effect. Although AMPC exhibits an initial undershoot, which is due to the inherent non-minimum phase (NMP) characteristics of IPM, but the CoM output (the desired signal, in red) is smooth and is also following closely to the reference ZMP (blue waveform) trajectory. The algorithm is still very effective, and the jerk in both the X and Y direction is minimized to zero as shown in Figure 5.7.

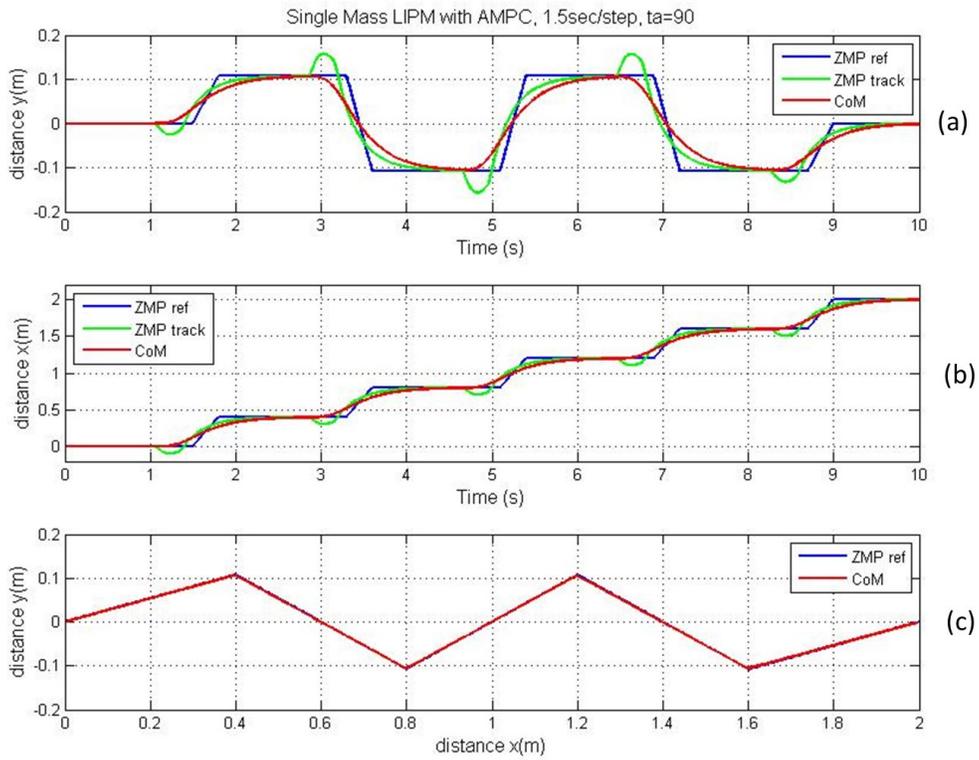


Figure 5.6. Augmented MPC model tracking trajectory, (a) y-axis, (b) x-axis, (c) xy plane.

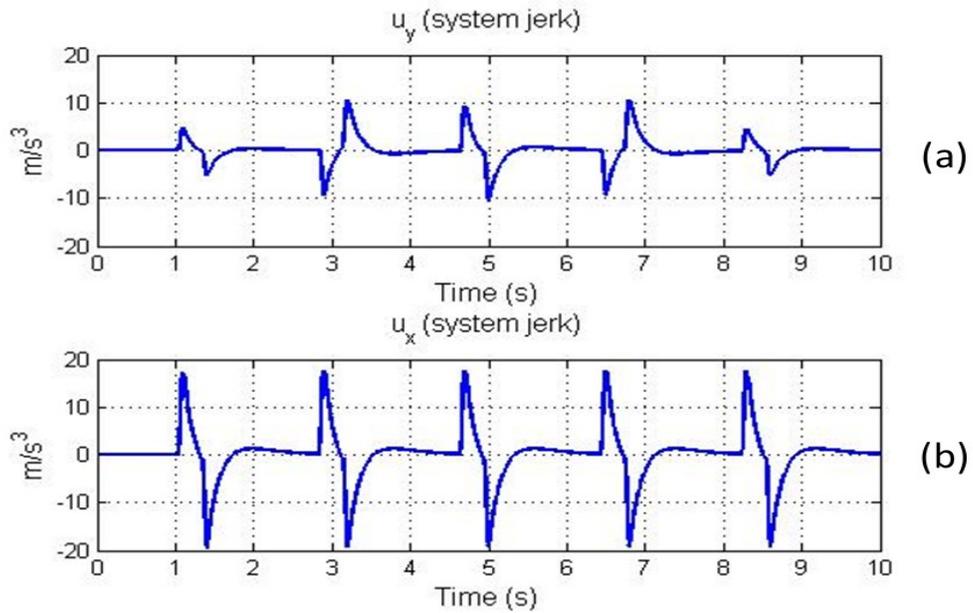


Figure 5.7. AMPC system jerk is reduced to zero in both (a) x-axis and (b) y-axis direction.

5.1.6 Comparison of AMPC and Preview Control Method

Referring to Figure 5.8, with a closer examination of the CoM trajectory produced using the AMPC tracking, it can be seen that the CoM curve (in red) moves slowly toward the reference ZMP curve (in blue). This can be interpreted as the controller trying to reduce the jerk and impact force during swing foot landing and cautiously reduce the ZMP error. As a result, the weight of CoM transfers from the supporting foot to the landing foot (new supporting foot) during the double support phase (DSP) is slow and gentle. In comparison with the preview control technique [S. Kajita, F. Kahehiro et al., 2003],[Thomas B. Sheridan, 1966], which produces a symmetrical sinusoidal waveform (in green), the AMPC algorithm produces a *truncated* sinusoidal trajectory during swing foot landing due to its ‘soft landing, fast take off’ characteristics. The other advantages of using the AMPC model is that the optimization process does not need to solve the *Riccati* equation and the optimization window can be easily adjusted to fine-tune the system response. Since AMPC uses the recursive iteration method, with continuing improvement in computer hardware and software architecture, this in turn reduces system-processing time and therefore, it is possible to implement it in real-time and in those low-end embedded systems which have limited computing power and memory storage.

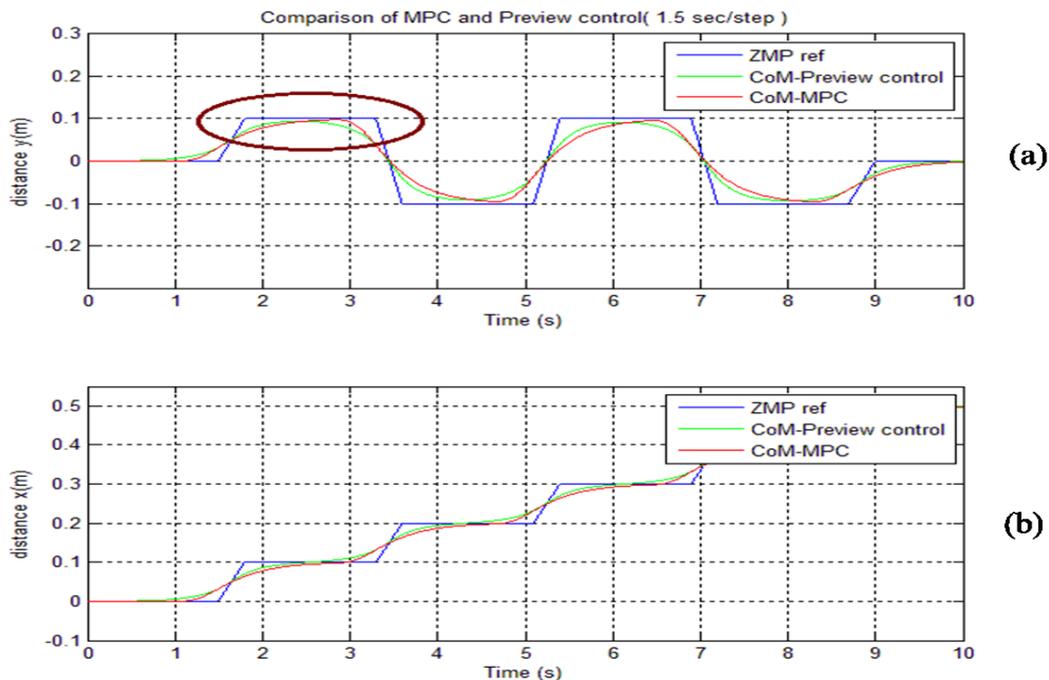


Figure 5.8. Comparison of MPC and Preview control, (a)y-axis, (b)x-axis.

The simulation of the biped dynamic walking is depicted in Figure 5.9 and its joint angle using the AMPC method is depicted in Figure 5.10.

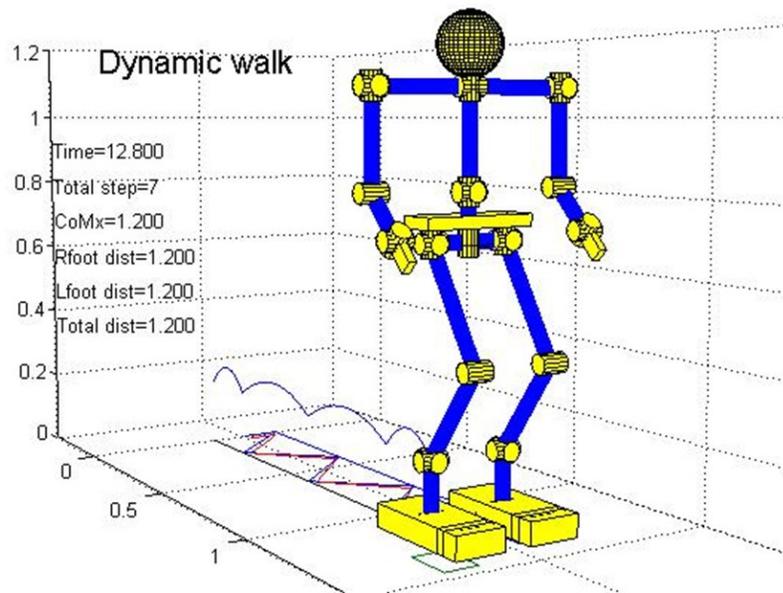
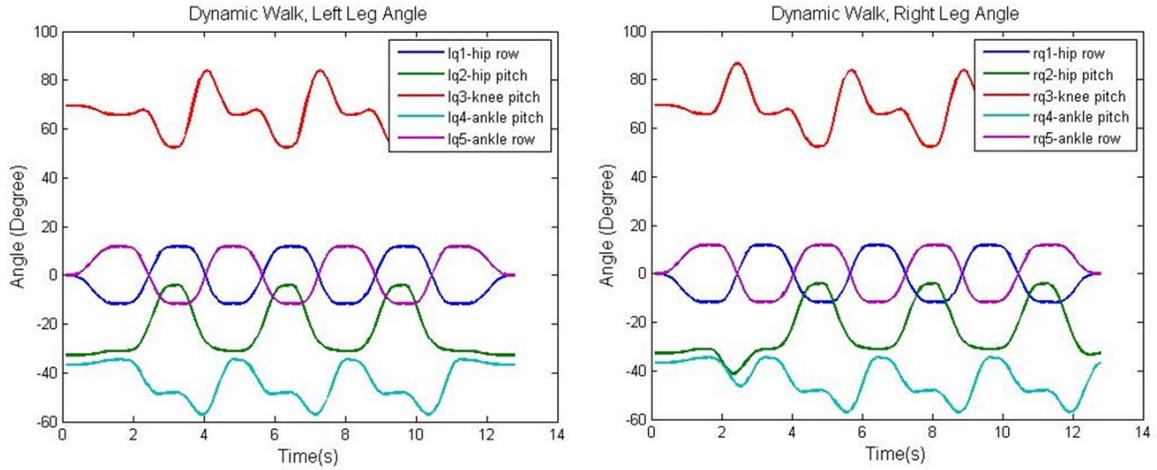


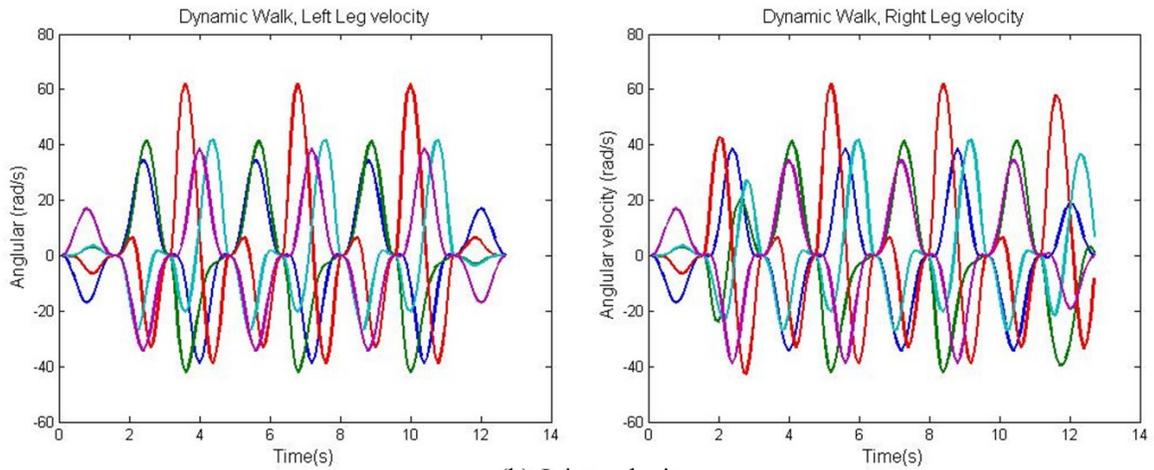
Figure 5.9. Robot simulation model.

The required joint torque can be calculated using Euler-Lagrange equation of motion with the above simulation values of the joint angular position, velocity and acceleration. Please refer to Appendix A for details.

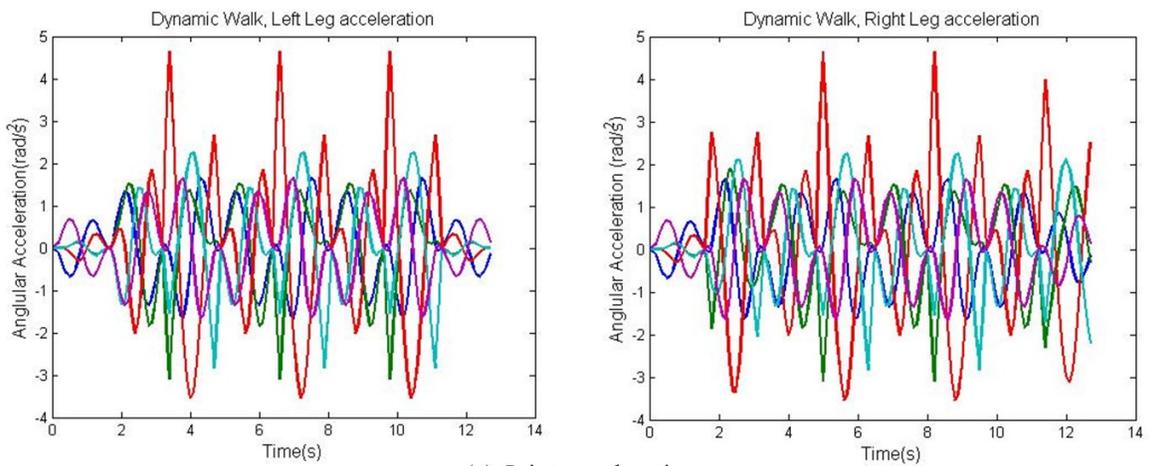
Another approach to reduce the water-bed effect is to design a feed forward controller using pole-zero cancelling technique, where the NMP *zero* is approximated with Taylor series expansion to achieve a less overshoot trajectory tracking. This method will be discussed in the following sections.



(a) Joint angle position



(b) Joint velocity



(c) Joint acceleration

Figure 5.10. Robot joint parameters.

5.1.7 AMPC-PZC Control Method

The difficulties to control non-minimum phase (NMP) systems' undershoot effect can be reduced by using the pole zero cancellation (PZC) techniques, the details approach can be found in literatures [Masayoshi Tomizuka, 1987], [Eric G., Massayoshi T., 1994]. Figure 5.11 illustrate a schematic diagram of the proposed AMPC-PZC controller. The PZC feed forward controller is defined as

$$G_{PZC}(z) = \frac{G_c A(z)}{B^-(z)} \times \frac{1}{B^+(z)}, \quad (5.21)$$

where $A(z)$ are the system poles, $B^-(z)$ is the polynomial of the system stable zeros, $B^+(z)$ is the polynomial of unstable zeros, and G_c is the dc gain.

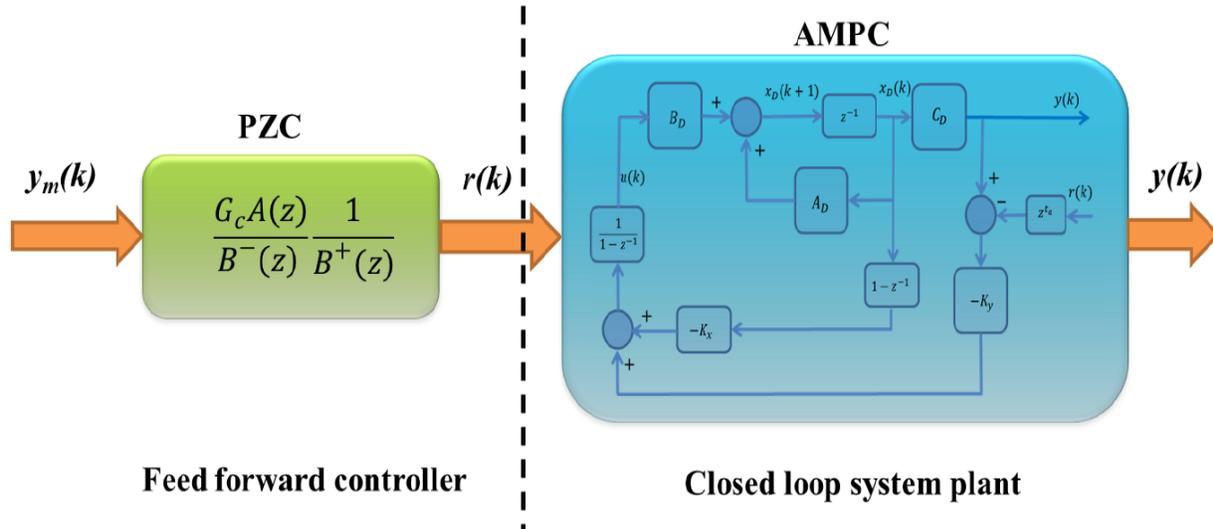


Figure 5.11. AMPC with PZC controller block diagram.

As depicted in Figure 5.11, the closed-loop system of the single mass inverted pendulum with AMPC is

$$\begin{aligned} GL(z)_{AMPC} &= \frac{G_p B^-(z) B^+(z)}{A(z)} \\ &= \frac{-1.423(z)(z - 0.9791)(z - 1.021)}{(z + 0.0003786)(z - 0.0003784)(z - 0.9567)(z - 0.9853)}, \end{aligned} \quad (5.22)$$

where $N_{mp}(z)$, $N_{nmp}(z)$ are the minimum and non-minimum phase parts respectively, G_p is the constant gain, and $D(z)$ is the plant denominator polynomials. The feed-forward PZC tracking controller can be derived from equation (5.21) and (5.22):

$$\begin{aligned}
 G_{PZC} &= \frac{1}{G_p} \times \frac{A(z)}{B^-(z)} \times \frac{1}{B^+(z)} \\
 &= (-0.703) \times \frac{(z + 0.0003786)(z - 0.0003784)(z - 0.9567)(z - 0.9853)}{(z)(z - 0.9791)} \\
 &\quad \times \frac{1}{(z - 1.021)},
 \end{aligned} \tag{5.23}$$

and the NMP zero($\frac{1}{(z-1.021)}$) is approximated by Taylor series expansion

$$\frac{1}{z - z_1} = -\frac{1}{z_1} - \frac{1}{z_1^2}z - \frac{1}{z_1^3}z^2 + \dots \quad \text{for } |z| < |z_1| = 1.021|.$$

The simulation waveform is shown in Figure 5.12 and the control signal is shown in Figure 5.13 respectively. This method produces a higher spike in the control signal.

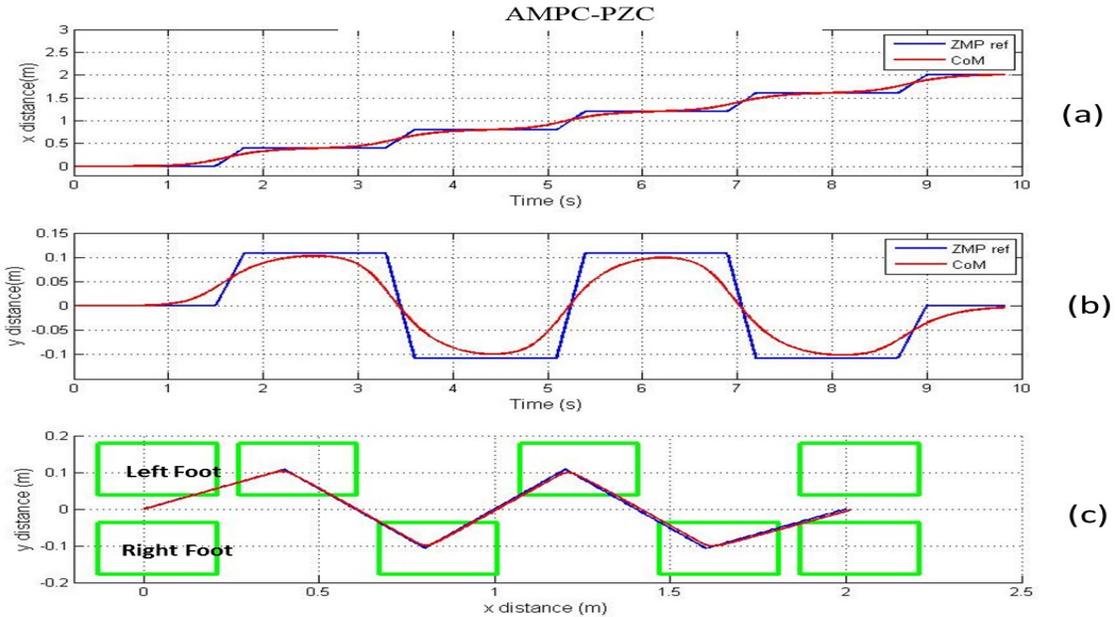


Figure 5.12. AMPC-PZC waveform, (a) x-axis, (b) y-axis, (c) xy-axis.

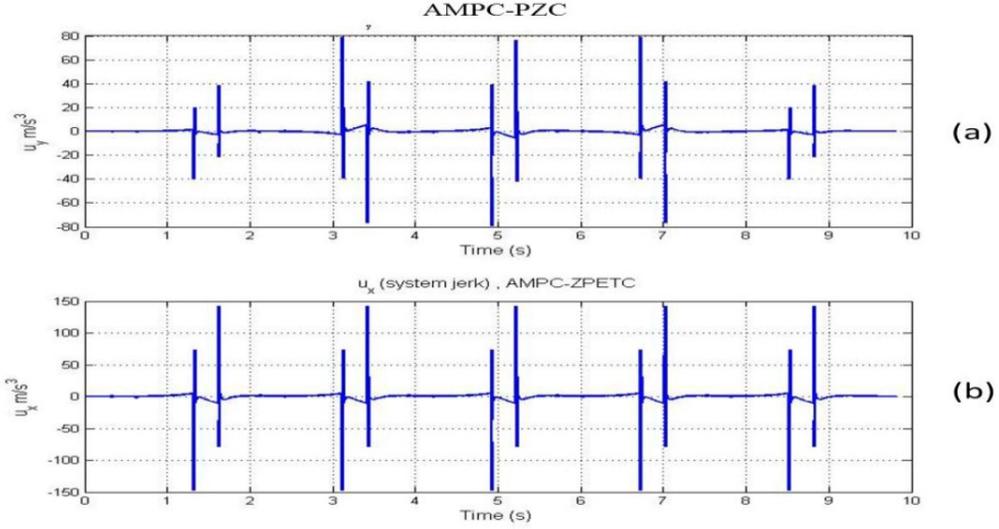


Figure 5.13. AMPC-PZC control signal, (a) y-axis, (b) x-axis.

5.1.8 Double Mass Inverted Pendulum with Model Predictive Control

A robot with torso can be modeled with the double inverted pendulum method. The challenge now will be designing a controller to control the motion of two point masses. As explained in chapter 4, the state-space of a double mass inverted pendulum model (DMIPM) can be written the form

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ \theta \\ \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_B \underbrace{\begin{bmatrix} u \\ \ddot{\theta} \end{bmatrix}}_u, \quad (5.24)$$

$$\underbrace{y}_{P_{zmpx}} = \underbrace{\begin{bmatrix} 1 & \frac{m_1 L}{m_1 + m_2} & 0 & 0 & -\frac{m_1 z_1 + m_2(z_1 + L)}{(m_1 + m_2)g} & -\frac{m_2 L(z_1 + L)}{(m_1 + m_2)g} \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1 \\ \theta \\ \dot{x}_1 \\ \dot{\theta} \\ \ddot{x}_1 \\ \ddot{\theta} \end{bmatrix}}_x. \quad (5.25)$$

This system has double control input $[\ddot{x} \quad \ddot{\theta}]^T$ variables, so note that the B matrix now has two columns. The cost function can be derived by using the same approach from section (5.1.2). The

simulation is performed with the following parameters: $L=0.273\text{m}$, $z_1=0.55\text{m}$, $m_1=35\text{kg}$, $m_2=27\text{kg}$, $\theta=0.15^\circ$, $g=9.81\text{m/s}^2$. In Chapter 4, we shows that the DMIPM does not have a *zero* at the RHP, as such, there is no undershoot or water-bed effect at the output ZMP waveform and this model also managed to track the reference ZMP waveform with good accuracy. Note that the simulation is done piece-wise with single step response. As illustrated in Figure 5.14, the angle θ of the upper pendulum need to reset to initial value with every support step switch over during the DSP. This method is suitable to control robot walking on stairs or stepping over obstacle that require piecewise motion. The simulation waveform is depicted in Figure 5.15.

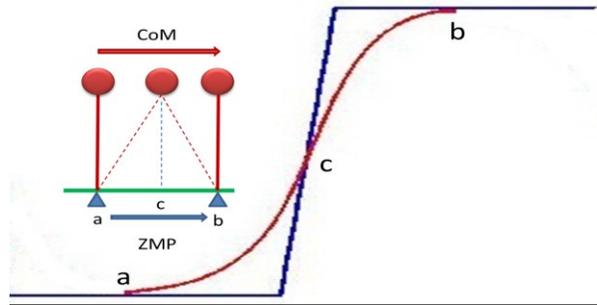


Figure 5.14. Piece-wise single step response.

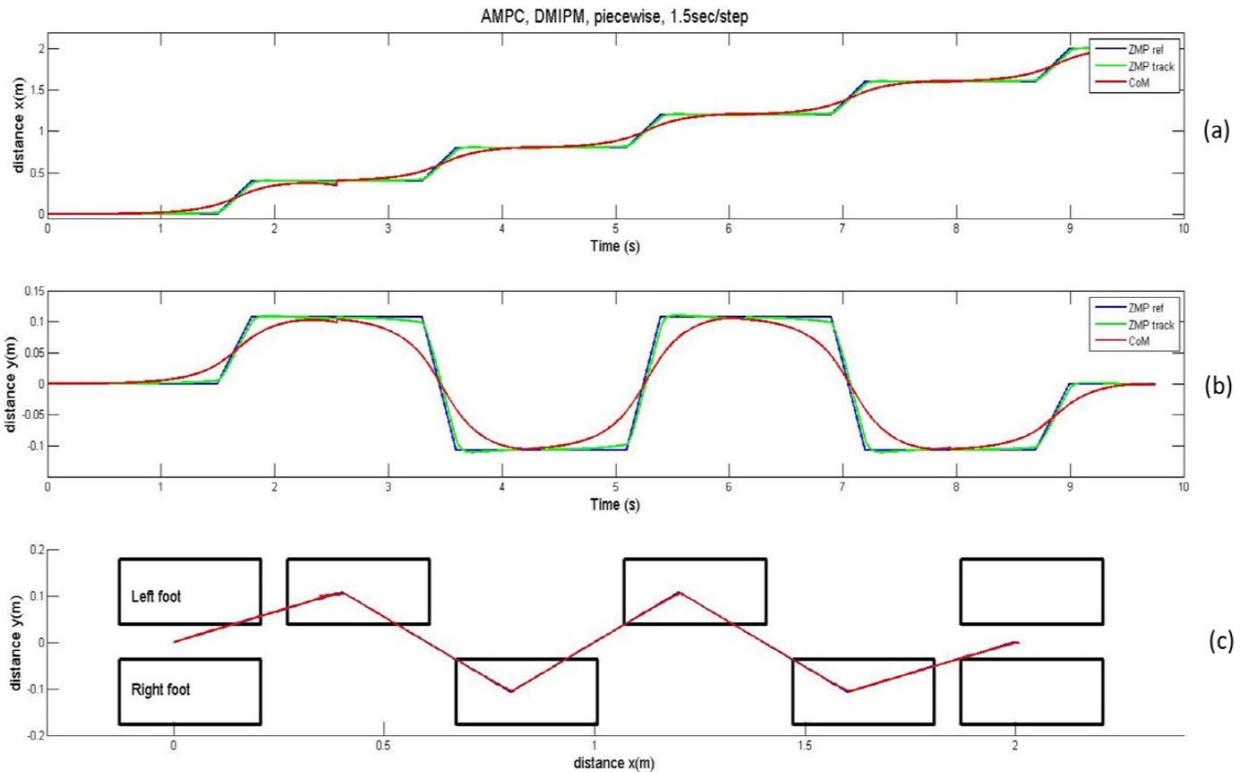


Figure 5.15. AMPC with DMIPM waveform, the black box in (c) represent the robot foot step.

5.2 Robot with Passive Toe-foot

In this section we study the advantage of a robot having passive toe-foot.

5.2.1 Passive Toe

Figure 5.16 shows a typical toe-foot design. With the combined actions of the CoM's natural oscillation frequency and passive toe, we can improve the robot walking efficiency and achieve a longer step strike. The TPinokio robot is designed with a passive spring-toe.

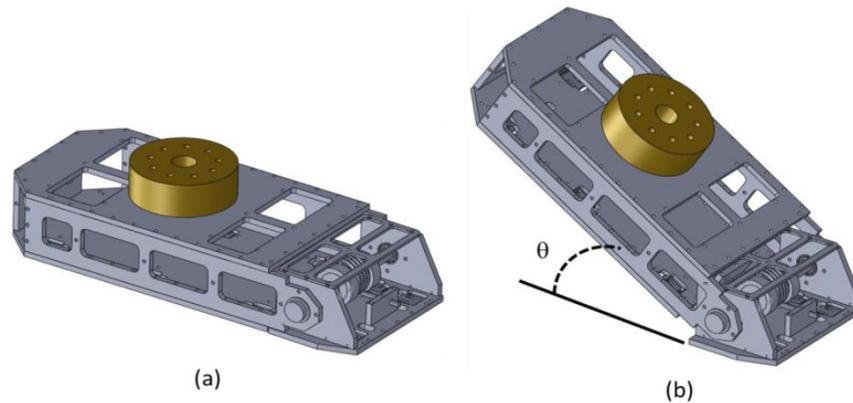


Figure 5.16. TPinokio robot with passive foot-toe design.

5.2.2 Modified Toe-foot gait

A foot with toe will reduce the spark in maximum speed of the knee joint and can produce more natural walking posture. Furthermore, the walking speed as well as the maximum step-length can also be increased. As illustrated in Figure 5.17(a), in sagittal plane, the ZMP trajectory of the toe-foot is different from the flat-foot trajectory and it now has three phases: Under-Actuated-Phase (UAP), Single-Support-Phase (SSP) and Double-Support-Phase (DSP). The modified trajectory in sagittal plane is as shown in Figure 5.17(b), starting from the SSP, the support feet ZMP shift linearly from the heel to the toe, and changed over to the new support feet to the end of the DSP.

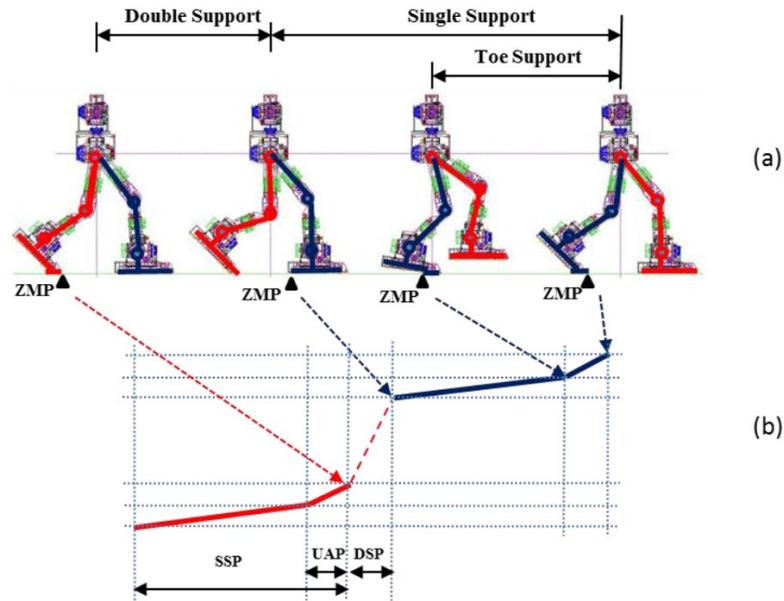


Figure 5.17. Modified toe-foot walking gait and its ZMP position in sagittal plane.

The advantage of having a toe-feet is illustrated in Figure 5.18. With a flat-foot and CoM height of 0.55m, the TPinokio robot can reach a maximum step size of 0.45m and a walking pace of 1.5s/step, which is around 1.1km/h. In the case where there is a toe-feet, as shown in Figure 5.18(b), the maximum step size is 0.6m, achieving a walking speed of around 1.4km/h, an increase of around 27% in walking speed.

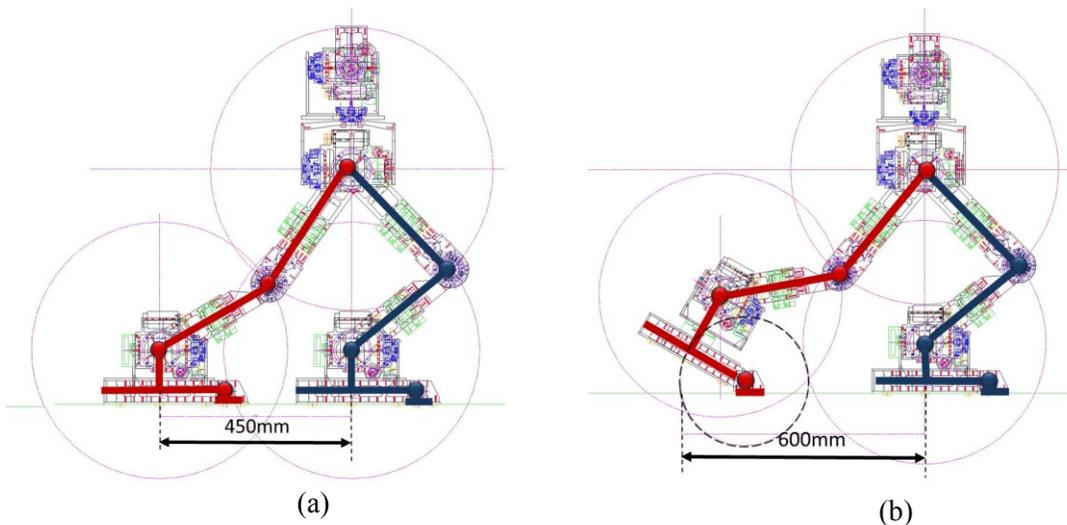


Figure 5.18. (a) Flat-foot step length. (b) Toe-feet step length.

Referring to Figure 5.19, another advantage of the toe-foot design is that the robot can perform stretch walking with toe-off and heel-strike motion. The normal value for a human walking with toe-off and heel-strike is 30° and 26° respectively.

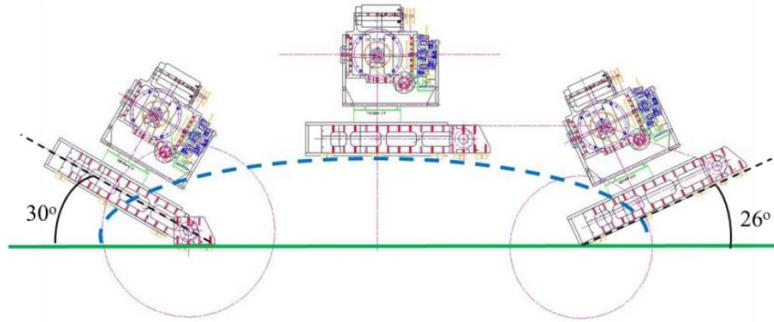


Figure 5.19. Toe-off and heel-strike walking trajectory in sagittal plane.

5.3 Feasibility Study of Robot Walking on Different Surface

In this section, we conduct a feasibility study of the performance of the TPinokio robot with considering the kinematics structure, the limitation of the robot legs and the geometry of the obstacle [G.Q. Zhang, M. Xie et al., 2009]. These kinematics limitations and characteristics can then either be stored or retrieved for future motion planning.

5.3.1 Walking On Slope

The IPM concept is used to model a bipedal walking on slope. Recall from Chapter 4, the ZMP-CoM equations (Sagittal plane, xz -axis) of a robot walking on flat surface with constraint in the z -plane are:

$$\begin{aligned}
 k_y &= 0, \\
 z &= z_c + k_x x + k_y y, \\
 m(\ddot{z}_{com} + g)(x_{com} - x_{zmp}) &= m(\ddot{x}_m)(z_{com} - z_{zmp}). \tag{5.26}
 \end{aligned}$$

Referring to Figure 5.20, for walking on either up slope or down slope, the constraint z -plane is now a slope with angle θ , and the constraint parameters are:

For walking up a xz -plane slope: $k_x = \tan \theta$, $k_y = 0$, $z = z_c + k_x x$.

For walking down a xz -plane slope: $k_x = -\tan \theta$, $k_y = 0$, $z = z_c - x \tan \theta$.

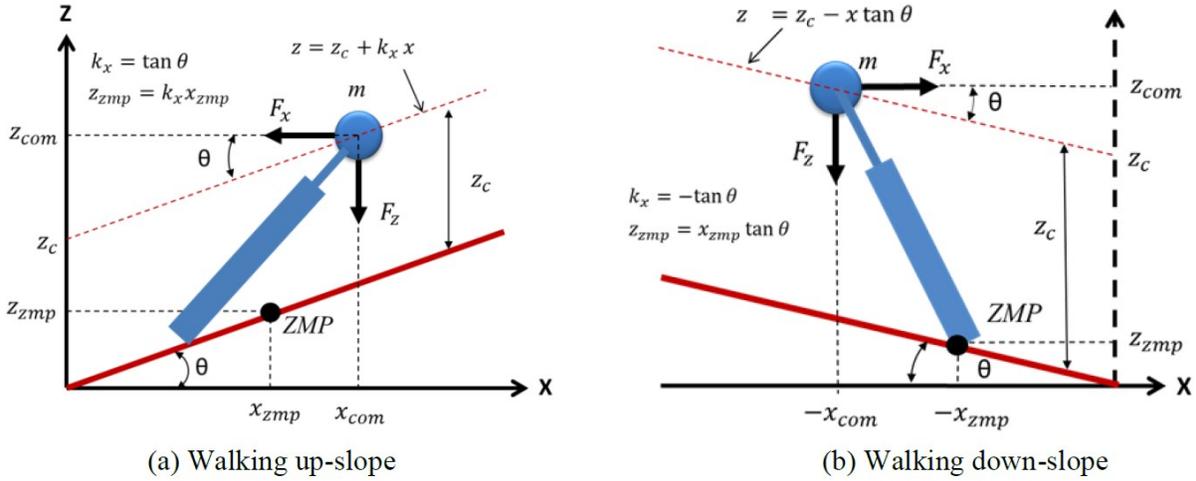


Figure 5.20. Inverted pendulum walking on slope model.

5.3.1.1 Walking Up Slope

In this section we derive the equations for a robot walking down slope as illustrated in above Figure 5.20(a). The constraint parameters are:

$$k_x = \tan \theta, \quad k_y = 0, \quad z = z_c + k_x x. \quad (5.27)$$

With $z_{zmp} = k_x x_{zmp}$, $\ddot{z}_{com} = k_x \ddot{x}_{com}$, from equation (5.26) we obtain

$$\begin{aligned} m(\ddot{z}_{com} + g)(x_{com} - x_{zmp}) &= m(\ddot{x}_{com})(z_{com} - z_{zmp}), \\ (k_x \ddot{x}_{com} + g)(x_{com} - x_{zmp}) &= \ddot{x}_{com}(z_c + k_x x_{com} - k_x x_{zmp}). \end{aligned} \quad (5.28)$$

Simplify equation (5.28), the dynamic equation for xz-plane is

$$x_{zmp} = x_{com} - \left(\frac{z_c}{g}\right) \ddot{x}_{com}. \quad (5.29)$$

Similarly, the yz-plane dynamic equation is

$$y_{zmp} = y_{com} - \left(\frac{z_c}{g}\right) \ddot{y}_{com}. \quad (5.30)$$

Note that the walking up the slope ZMP equations (5.29) and (5.30) is the same as the ZMP equations for walking on flat surface. Therefore, the same CoM/ZMP trajectory tracking technique used for the flat ground can also be implemented for walking up a slope. Also note that now the z_{com} is not a constant with respect to the horizontal ground, but it is a constant with respect to the slope and is now a function of the robot pelvis height (z_c) and the slope angle (θ).

As illustrated in Figure 5.21, for stable up slope walking, we need to adjust the leg ankle's pitch angle to $\theta_{ankle_pitch} = \theta_{slope}$, so that the robot foot is parallel to the slope when walking up the slope. The simulation results of the robot joints' trajectory is depicted in Figure 5.22.

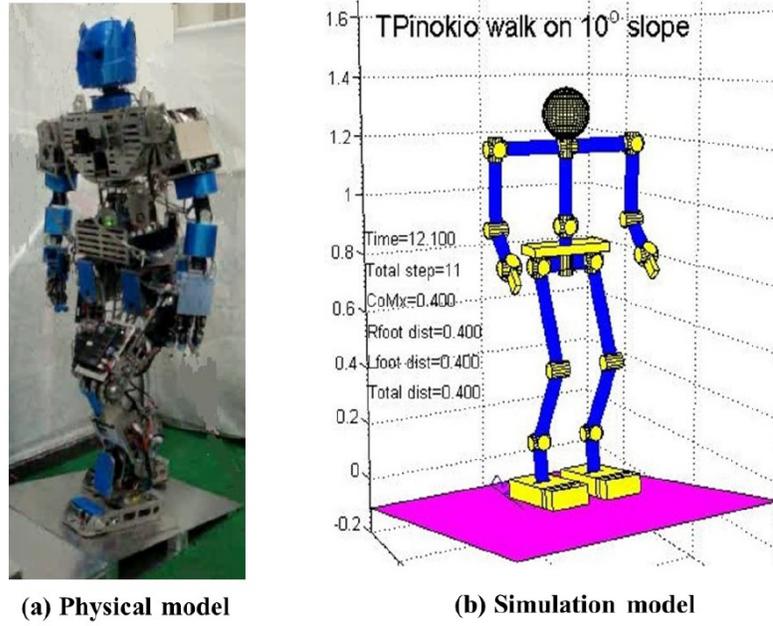


Figure 5.21. Posture of robot walking on slope.

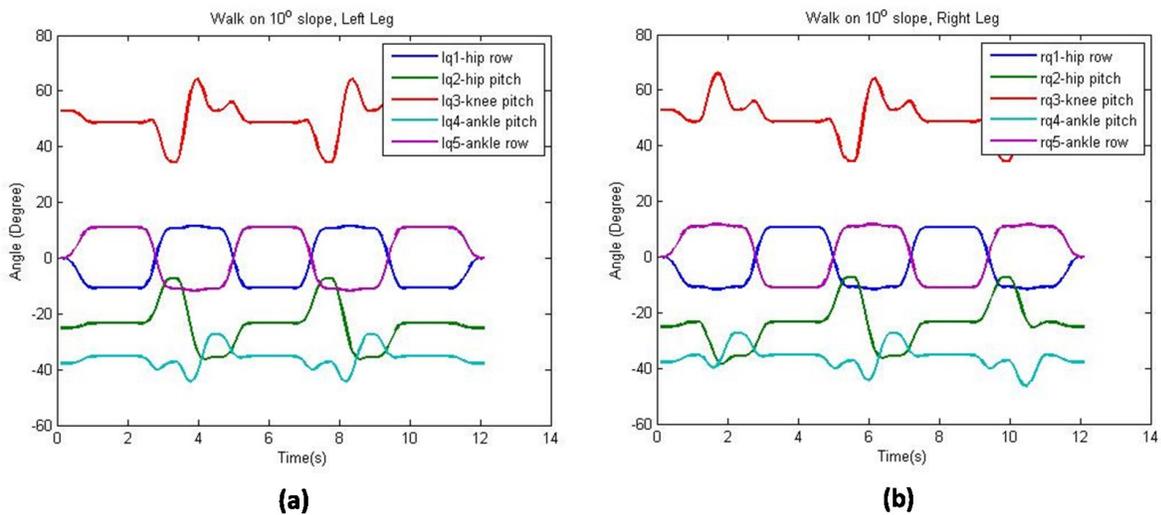


Figure 5.22. Robot walking up slope joints' trajectory, (a) Left leg, (b) Right leg.

5.3.1.2 Walking Down Slope

Referring to previous Figure 5.20(b), the constraint plane is now a negative slope with angle θ , and the parameters are:

$$k_x = -\tan \theta \quad , \quad k_y = 0 \quad , \quad z = z_c - x \tan \theta . \quad (5.31)$$

With $z_{zmp} = -x_{zmp} \tan \theta$, $\ddot{z}_{com} = -\ddot{x}_{com} \tan \theta$, from the ZMP equation (5.26), we obtain

$$(k_x \ddot{x}_{com} + g)(x_{com} - x_{zmp}) = \ddot{x}_{com}(z_c + k_x x_{com} - k_x x_{zmp}). \quad (5.32)$$

Simplify the above equation (5.32), the dynamic equations for the xz-plane and yz-plane are as follows:

$$\begin{aligned} x_{zmp} &= x_{com} - \left(\frac{z_c}{g}\right) \ddot{x}_{com} , \\ y_{zmp} &= y_{com} - \left(\frac{z_c}{g}\right) \ddot{y}_{com} . \end{aligned} \quad (5.33)$$

Note that the ZMP-CoM equation is the same as on walking on flat ground. The only required modification is to adjust the ankle pitch angle to $\theta_{ankle_pitch} = -\theta_{slope}$. So that the robot foot is parallel to the slope when walking down the slope.

5.3.2 Step Over Obstacle in Statically Balancing Mode

To control a bipedal robot to step over an obstacle without losing balance is another challenging task. The general procedure is to first define the constraints on the leg joints limited by its legs' mechanical structure and kinematics, follow by formulating the balance and collision-free constraints, and finally derive the mathematical models. To do this, we adopted the statically balancing walking strategy with the CoM of the robot always remains in the stance feet support convex polygon when one leg leave the ground. The CoM will only moves when both of the robot foot is on the ground during DSP. There are two basic constraints, the balance constraint and the collision free constraint must be observed in this stepping over obstacle movement. The following assumptions are made to simplify our analysis:

- A typical obstacle is a rectangular box shape. The minimal rectangular box is used to approximate a nonrectangular obstacle.

- To achieve balance constraint, we only consider static balancing and kinematics constraint.
- To achieve collision free constraint, the feet move in parallel to the sagittal plane, the frontal distance between the two legs remain constant and no yaw movement.

5.3.2.1 Geometrical Method to Determine the Obstacle Size

A geometrical method to determine the obstacle size is introduced in this section, as depicted in Figure 5.23. This is to determine the robot step length, the step height, the foothold positions and the maximum obstacle size which the robot can step over. Some critical geometric parameters must be considered carefully during trajectory planning:

- The distance of the bipedal feet and the obstacle must have good clearance;
- The swing feet must be lifted high enough to cross over the obstacle top; and
- The step length must be sufficient to maintain good balancing and avoiding knee-overstretch.

This method is simple and effective. Besides that, other advantages are that no complicated optimization programming is needed and the parameters obtained are near to optimal value. To determine the parameters, firstly, the ground projection of the robot CoM (GCoM) must be within the rear support foot toe supporting polygon 'a'. Secondly, the swing foot is stretched to maximum position 'b', which is limited by the rear feet toe angle θ and the leg length L_1 and L_2 , and the maximum step size 's' is obtained. Finally the maximum obstacle profile can be determined by direct measurement. In this research, referring to Figure 5.23, we chose the GCoM to be at the center of the rear foot toe, point 'a'. This will give good stability when the swing foot and the robot CoM are moving across the obstacle. For the Tpinokio robot, with $Z=682\text{mm}$, $\theta=30^\circ$, $L_1=331\text{mm}$, $L_2=300\text{mm}$ and $s=504\text{mm}$, the maximum obstacle profile is found to be $w=130\text{mm}$, $w_1=59\text{mm}$, $w_2=49\text{mm}$, $h_1=206\text{mm}$ and $h_2=322\text{mm}$.

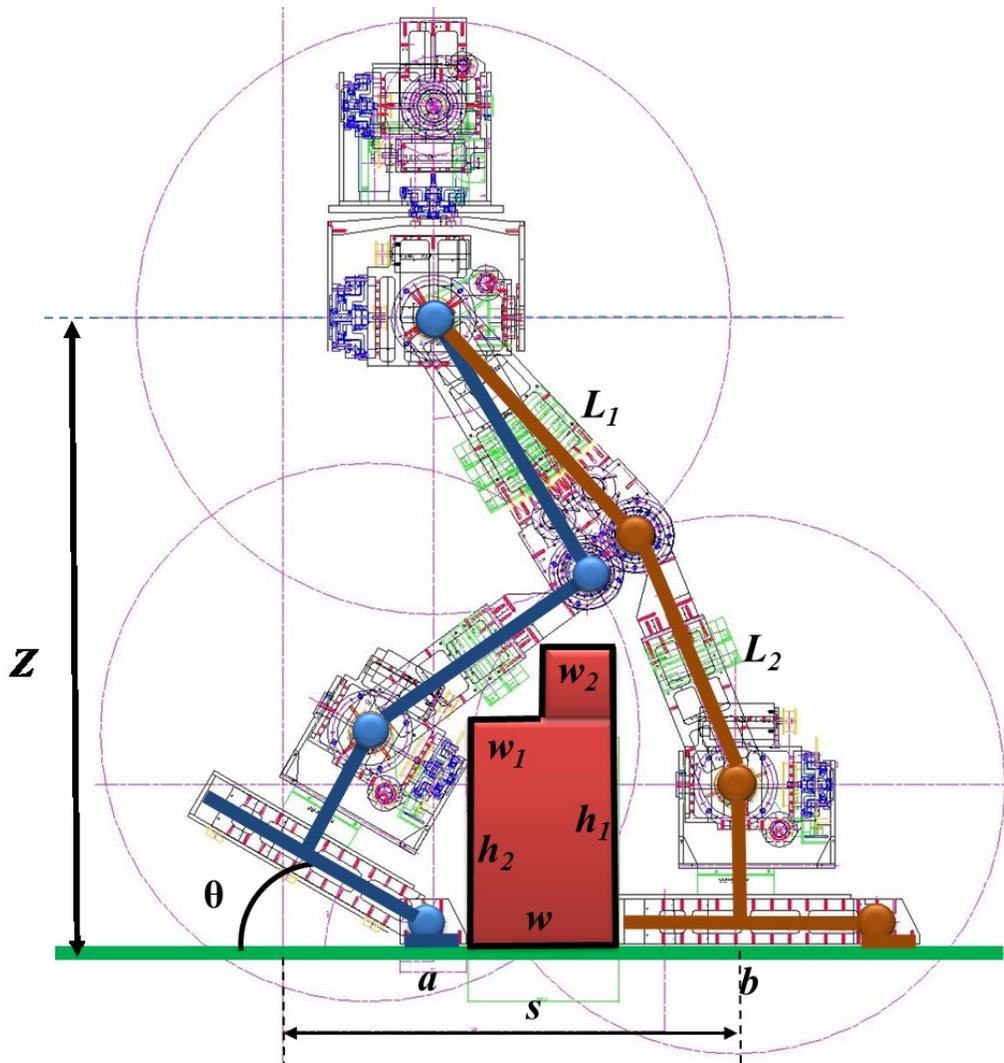


Figure 5.23. Geometrical measurement of robot stepping over obstacle parameters.

Once the feet and pelvis trajectories are determined, the individual joint-space trajectory can be determined from the inverse kinematics transformation. In order to reduce the motion jerk, a seven degree polynomial function is used to set the swing feet path, so that the velocity, acceleration and jerk of the motion is reduced to zero at the beginning and end of the trajectory.

The TPinokio robot is able to step over a rectangular obstacle profile of 130mm (w) x 186mm (h), with clearance 20mm on both sides for the toe and heel. The basic sequence for stepping over a rectangular obstacle is depicted in Figure 5.24, and described in algorithm 5.1, the simulation is depicted in Figure 5.25, 5.26 and 5.27 respectively.

Algorithm 5.1 Pseudo-code for stepping over obstacle in statically balancing mode.

Input : Initial position, sequence of foot and pelvis position.

Output : Trajectory of CoM and all joints position.

Start

/* 1: initialization

set robot to initial posture, ready to step over obstacle.

/* 2: double support

robot move its pelvis point (CoM) to the left supporting leg position.

/* 3: single support

robot lifts its right swing leg, moves the swing foot over the obstacle.

/* 4 : double support

robot lands its swing leg on the other side of the obstacle.

/*5: double support,

robot moves its pelvis (CoM) toward the front support leg.

/*6: single support

robot lifts its rear leg, swing it upward and over to clear the obstacle.

/*7: double support

robot move back to initial posture, ready for next action.

End

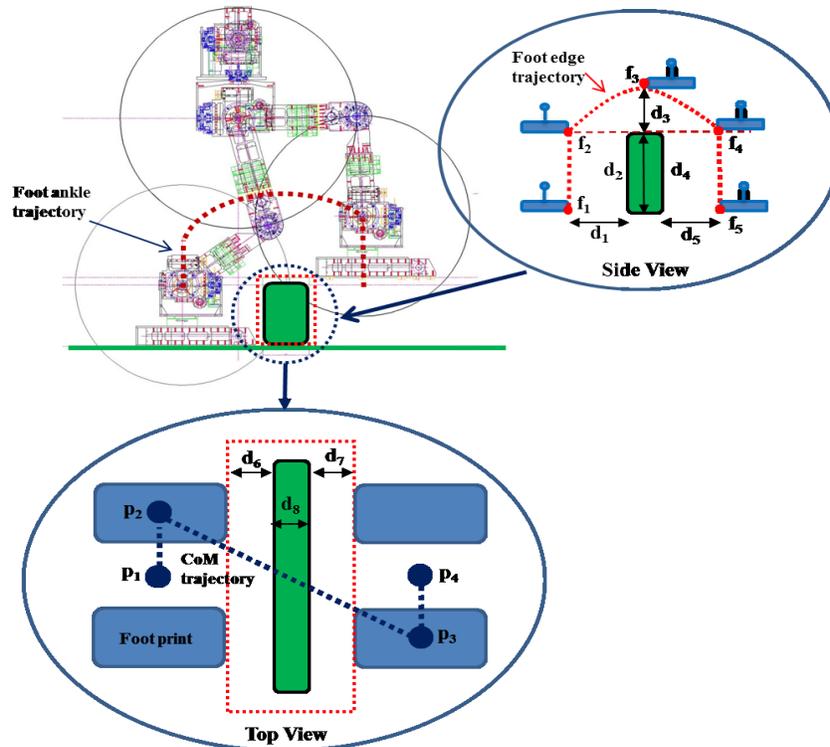
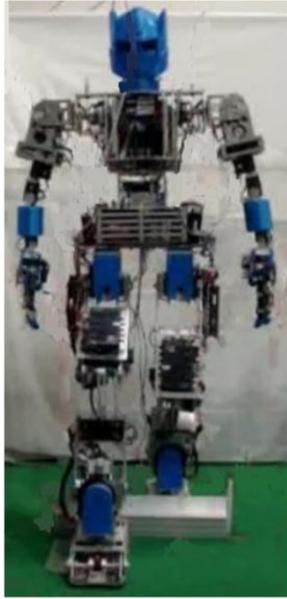
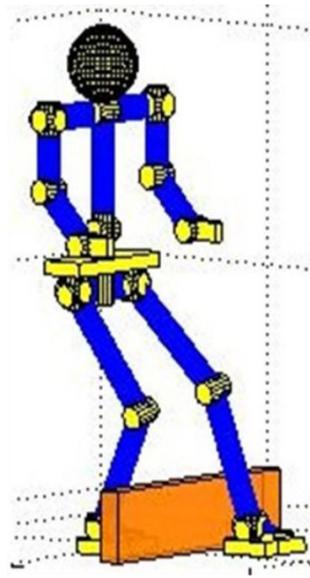


Figure 5.24. Robot foot trajectories for stepping over a rectangular obstacle.



(a) Physical model



(b) Simulation model

Figure 5.25. TPinokio step over obstacle posture model.

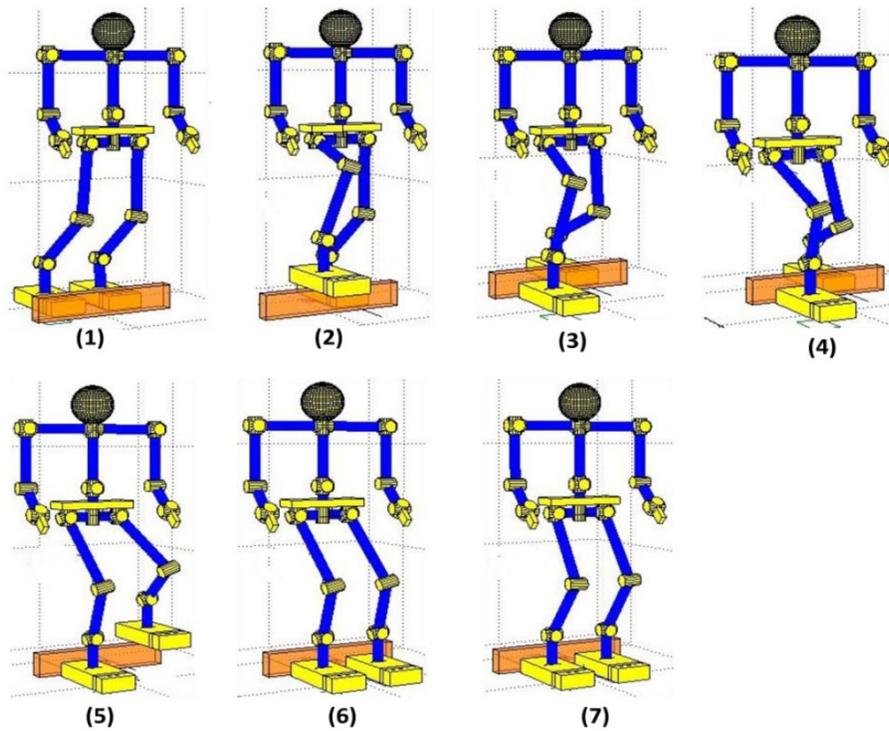


Figure 5.26. Basic sequences for stepping over obstacle with statically balancing mode.

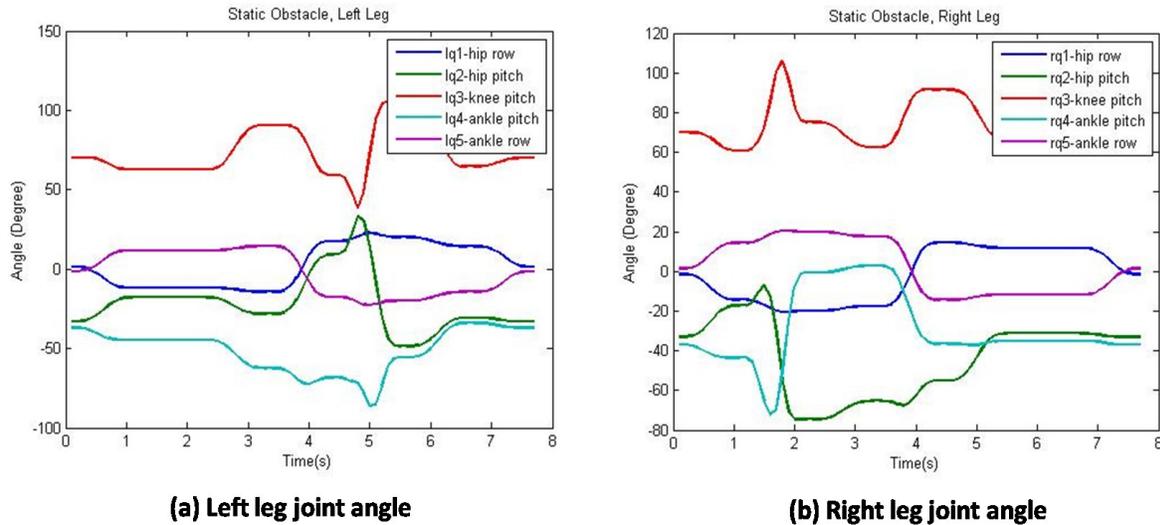


Figure 5.27. Leg joints' trajectory for stepping over obstacle, (a) Left leg, (b) Right leg.

5.3.3 Step On and Over Obstacle in Statically Balancing Mode

Another common method to cross a higher obstacle as shown in Figure 5.28, is by first stepping on the obstacle top surface and then lower down the support leg to clear the obstacle. With this method, the robot is able to clear a higher obstacle.

5.3.3.1 Geometrical Method to Determine the Obstacle Size

To determine the obstacle size, first we need to identify the front leg landing position at point 'a', and the obstacle width 'w'. For this analysis, the minimum obstacle width is chosen to be the same to the robot feet length so that the robot can stand firmly on the obstacle surface with the rear support leg, the robot pelvis Ground Center of Mass (GCoM) position is placed to be in line with the toe joint. Below are some critical geometric parameters must be considered carefully during trajectory planning:

- The position of the bipedal robot and the obstacle must have good clearance;
- The swing feet must be lifted high enough to reach the obstacle surface top and to avoid collision; and
- The height of the Pelvis must be able to maintain good balancing and to avoid knee-overstretch.

Once the position ‘ a ’ and ‘ b ’ is determined, we can begin to determine the other parameters. Slowly move the pelvis (CoM) vertically up along the z -axis, until just before the ground support foot limited by the leg length L_1 and L_2 . and at this position, the maximum obstacle height h can be determined as depicted in Figure 5.28. For our Tpinokio robot with $z=760\text{mm}$, the maximum obstacle dimension the robot can step over is $w=334\text{mm}$ and $h=325\text{mm}$,.

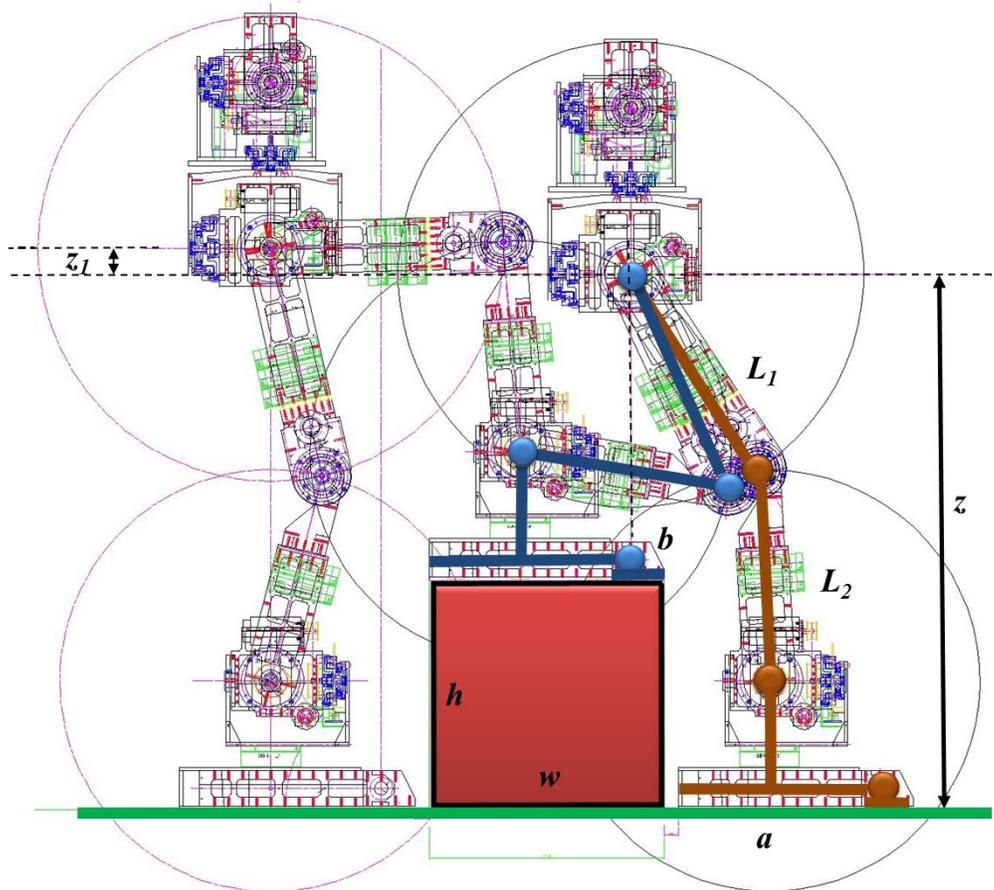


Figure 5.28. Geometrical measurement of robot stepping on and over obstacle parameters.

5.3.4 Stairs Climbing

Stair climbing is an important capability for bipedal robot [C.L. Shih, 1999]. As illustrated in Figure 5.29, a stairs-climbing gait consists of double and single support phase. During the SSP, the biped resemble an open kinematic serial-chain rigid body, because only one stance foot is supporting the robot and the other swing foot moves forward to the front stairs-step. During DSP, the bipedal can be regarded as a closed kinematic chain because both feet are standing in stationary position on the stairs-step.

Static climbing on stairs is slower but is stable; we only need to ensure that the CoM/ZMP is within the support convex polygon at all times and that the position of the pelvis and both feet location meet the kinematic constraints while generating the motion trajectory. Dynamic climbing is faster but is more challenging because we need to track the CoM/ZMP closely to prevent falling over, and we also need to ensure that the pelvis and both feet position meet the kinematic constraints while generating the trajectory.

In this section, we shall outline the sequences that a bipedal robot needs to take when ascending and descending stairs, and the constraints that must be satisfied in the procedures.

5.3.4.1 Geometrical Method to Determine the Stairs Step Size

Referring to Figure 5.29, in order to determine the stairs step size and the robot foothold position, below are some critical geometric parameters that must be considered carefully during our trajectory planning stage:

- The distance of the robot foot-toe to the stairs vertical surface must have good clearance;
- The swing foot must be lifted high enough to step over to reach the next stair-step without collision; and
- The foot-step length must be sufficient to maintain good balancing and to avoid knee-overstretching.

To determine the parameters geometrically, first the ground projection of the robot CoM (GCoM) must be within the front foot supporting polygon, we chose it to be at the center of the ankle point ' a ', the width of the stair-step is chosen to be slightly greater to then the width of the foot ' w ', this will give good stability when the swing foot and the robot CoM are moving up the stairs. Next, the swing foot is stretched back to position ' b ' to determine the foot-step length ' s '. With the rear foot position fixed, the robot CoM is moved up to position ' z ' vertically, which is limited by the leg length L_1 and L_2 , to determine the maximum stair-step height ' h '. And finally, the robot CoM movement during stairs climbing is determined by the virtual slope angle θ . For the Tpinokio robot, with $w=354\text{mm}$, $s=354\text{mm}$, $L_1=331\text{mm}$ and $L_2=300\text{mm}$, the maximum stair-step height is measured to be $h=183\text{mm}$ and the virtual slope angle $\theta=27^\circ$, which is about the standard stair-step dimension of most building.

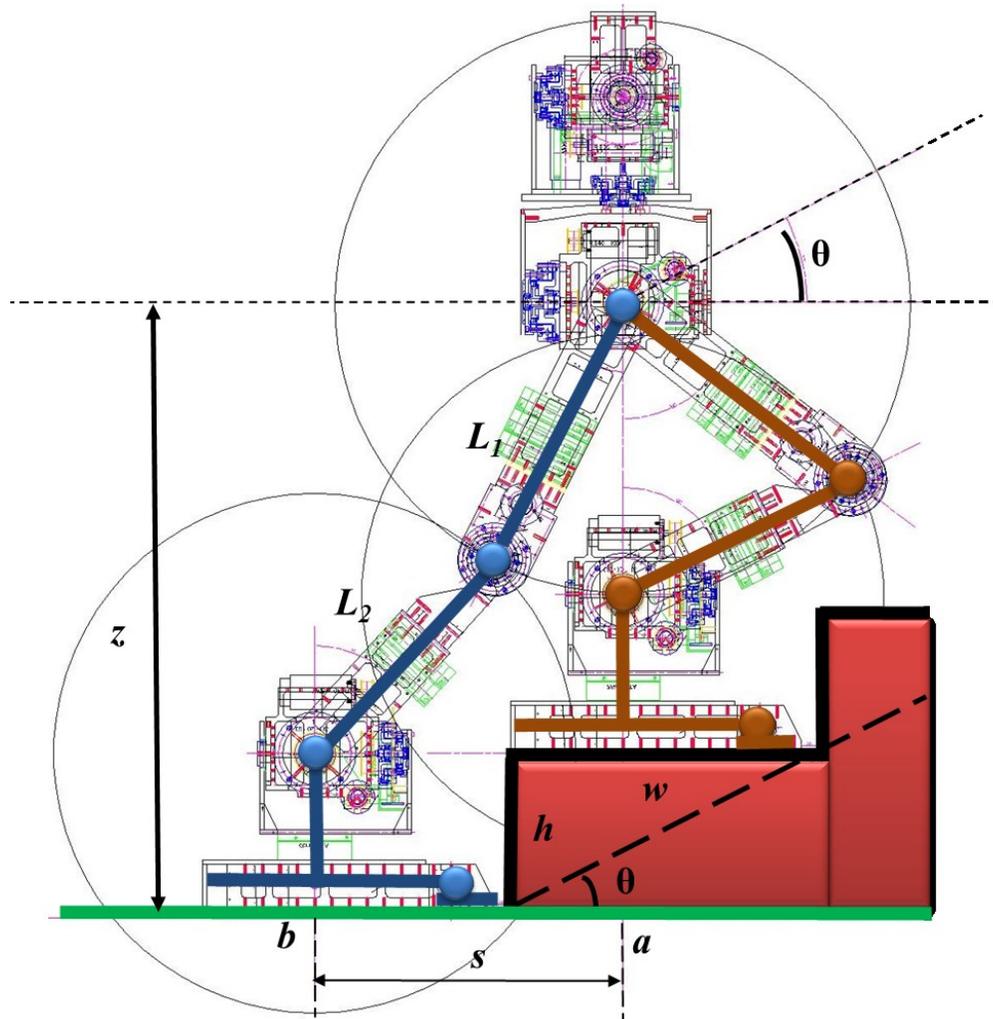


Figure 5.29. Geometrical parameters of robot climbing stairs.

5.3.4.2 Statically Balance Stairs Climbing

A sequential algorithm is implemented to generate the trajectory for the pelvis, stance and swing feet to achieve a stable static stairs climbing. This approach will guarantee stability because we can dictate the motion trajectory to be with the support polygon. The movement of the pelvis in sagittal plane as shown in Figure 5.30, is set to follow an imaginary slope with an angle θ defined by the stairs step height. A seven degree polynomial function is used to generate the path trajectories to achieve a zero jerk start-stop path. The unknown coefficient $a_0, a_1, a_2, \dots, a_7$ can be found by solving the following matrix:

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\
 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210
 \end{bmatrix}
 \begin{bmatrix}
 a_0 \\
 a_1 \\
 a_2 \\
 a_3 \\
 a_4 \\
 a_5 \\
 a_6 \\
 a_7
 \end{bmatrix}
 =
 \begin{bmatrix}
 U_1 \\
 0 \\
 0 \\
 0 \\
 U_2 \\
 0 \\
 0 \\
 0
 \end{bmatrix}.
 \tag{5.35}$$

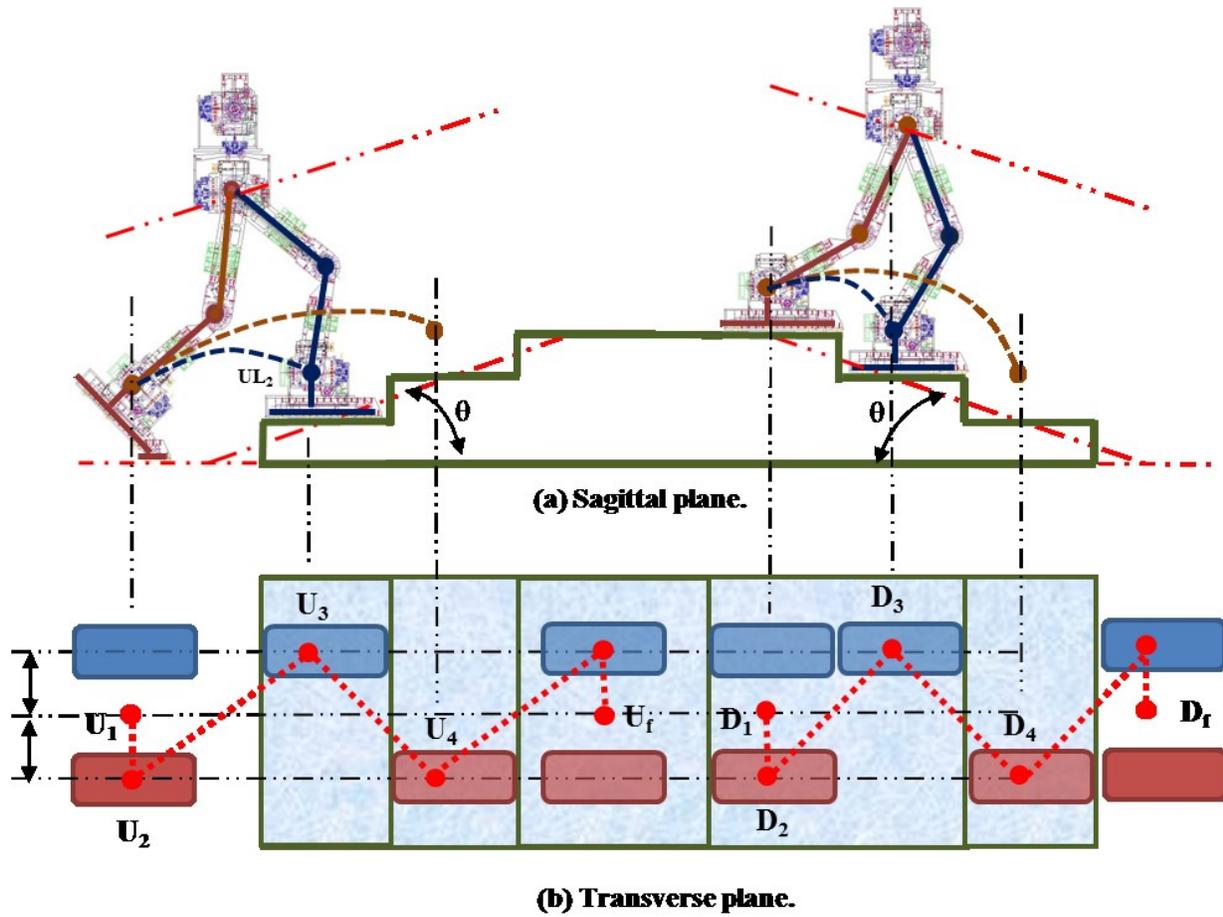


Figure 5.30. Stairs climbing sequences.

The basic sequences for climbing up and down stairs are described in algorithm 5.2 and 5.3 respectively. And the simulation is depicted in Figure 5.31.

Algorithm 5.2 Pseudo -code for walking up stairs in statically balancing mode.

Input : Initial position, number of stairs, step size, sequence of foot and pelvis position.

Output : Trajectory of CoM and all joints position.

start

*/*1: double support*

robot in initial posture, ready to step up stair, CoM at U_1 position.

*/*2: double support*

robot move its pelvis (CoM) to the supporting leg U_2 position.

*/*3: single support*

the robot lifts its swing leg, moves the swing foot up one stair to U_3 position.

*/*4: double support*

the robot moves its pelvis (CoM) to U_3 position.

*/*5: single support*

the robot lifts its rear leg, swing it upward one stair to U_4 position.

*/*6: do until end of stairs*

If not end of stairs

repeat 4 and 5 until end of stairs.

end

*/*7: double support*

robot move back to initial posture, CoM at U_f position.

End

Algorithm 5.3 Pseudo-code for walking down stairs in statically balancing mode.

Input : Initial position, sequence of foot and pelvis position.

Output : Trajectory of CoM and all joints position.

start

/ 1: double support*

robot in initial posture, ready to step up stair. CoM at D_1 position.

*/*2: double support*

robot move its pelvis (CoM) to the supporting leg D_2 position.

*/*3: single support*

robot lifts its swing leg, moves the swing foot down one step D_3 position.

*/*4: double support*

robot moves its pelvis (CoM) to D_3 position.

*/*5: single support*

robot lifts its rear leg, swing it down one step down to D_4 position.

*/*6: repeat 4 and 5 until end of stairs.*

If not end of stairs

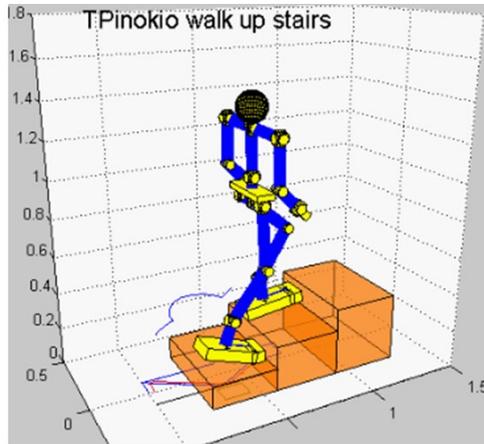
repeat 4 and 5 until end of stairs.

end

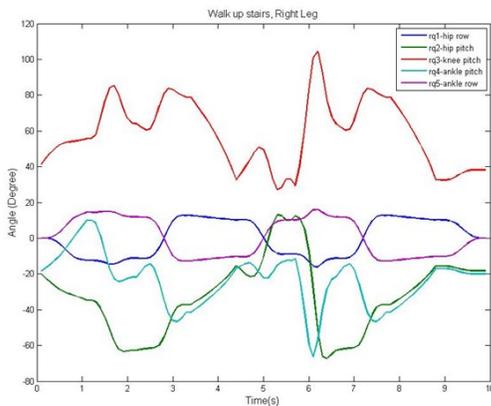
*/*7: double support*

robot move back to initial posture. CoM at D_5 position.

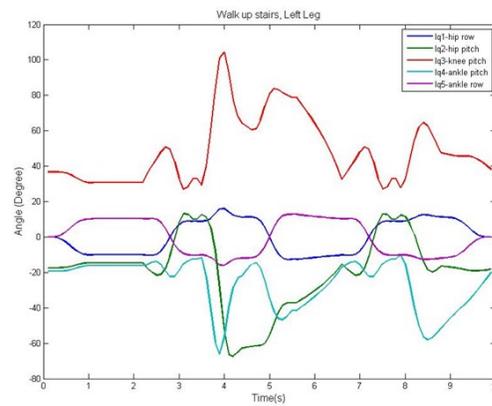
end



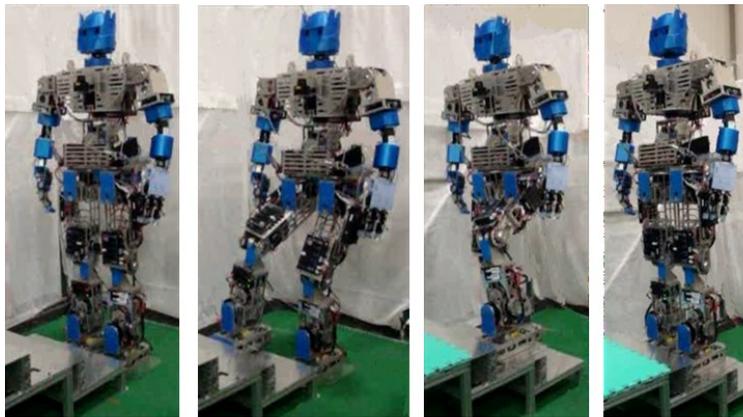
(a) Simulation model



(b) Right leg angle



(c) Left leg angle



(d) Walking up stair experiment with TPinokio.

Figure 5.31. Simulation of robot walking upstairs and its joints angle.

5.3.4.3 Dynamic Stairs Climbing

In dynamic stairs climbing, a simple strategy is to apply the Inverted Pendulum Model (IPM) concept. As illustrated in Figure 5.32, the movement of the IPM in sagittal plane is now modeled to follow an imaginary slope with an angle θ define by the stairs step height and depth. With this assumption made, the dynamic equations for walking up and down on the stairs are similar to that of walking up and down on a slope, except that the walking step size is now constrained by the stairs' depth.

Referring to Figure 5.32, an imaginary line is added with gradient equal to $\tan \theta = \frac{h}{w}$, where h denote the height and w is the depth and of the step. In order to avoid swing feet collision with the stairs' step, additional constraints are imposed on the robot's swing leg trajectory with the robot foot step size $S_s = w$ and the foot step height to $S_h > h$.

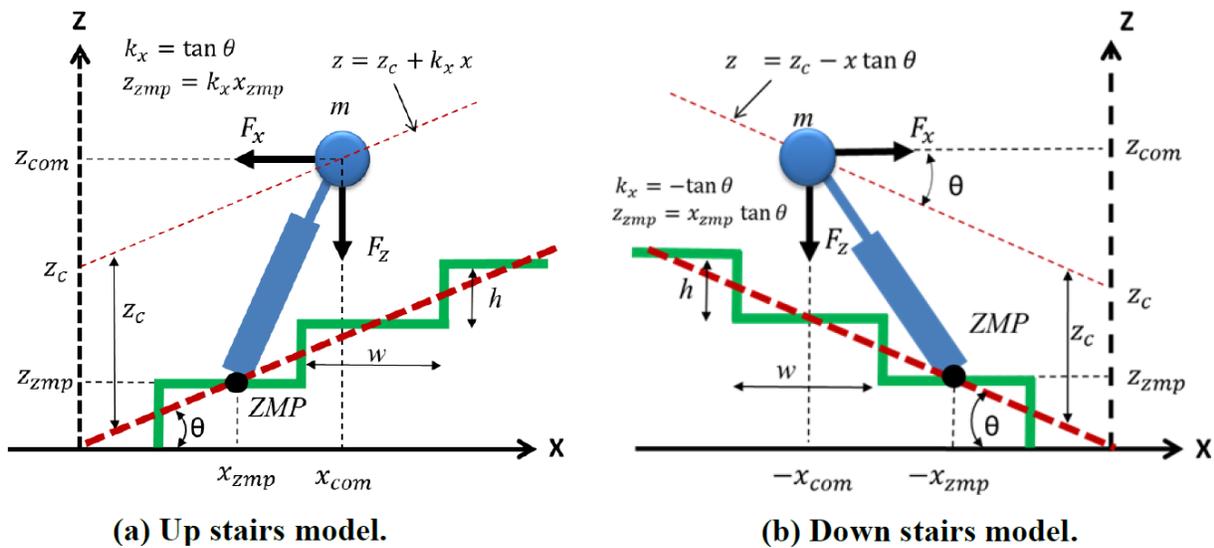


Figure 5.32. ZMP-CoM model for walking up and down stairs.

The constraints xz-plane and its parameters are set as follows:

$$\begin{aligned}
 &\text{up-stairs: } k_x = \tan \theta, \quad k_y = 0, \quad z = z_c + k_x x. \\
 &\text{down-stairs: } k_x = -\tan \theta, \quad k_y = 0, \quad z = z_c - x \tan \theta.
 \end{aligned}
 \tag{5.35}$$

The constraints to the swing feet are

$$S_s = w, \quad S_h > h. \quad (5.36)$$

The parameters and dynamic equations for ZMP-CoM tracking for walking on stairs are :

$$\text{Sagittal plane: } xz\text{-axis : } x_{zmp} = x_{com} - \left(\frac{z_c}{g}\right) \ddot{x}_{com}. \quad (5.37)$$

$$\text{Frontal plane: } yz\text{-axis : } y_{zmp} = y_{com} - \left(\frac{z_c}{g}\right) \ddot{y}_{com}. \quad (5.38)$$

5.3.5 Turning on The Spot

Turning on the spot is another challenging task for a bipedal robot. The robot needs to keep its balance while turning its foot to prevent leg collision. The follow sequences explain how a robot can make a 90° turn in eight steps geometrically. As illustrated in Figure 5.33, in order to make the robot turn 90° , two circles and intermediate position (A, B, C, D, E, F, G, H) are drawn, these set points are the robot foot ankle landing spot, the inner circle is the distance of the foot ankle, the outer circle is the boundary for the foot ankle to avoid leg/foot collision. The sequences and position of the robot CoM (pelvis) and both foot to turn 90° on the spot is illustrated in Figure 5.34. To turn 180° , the robot will have to execute the same sequence twice.

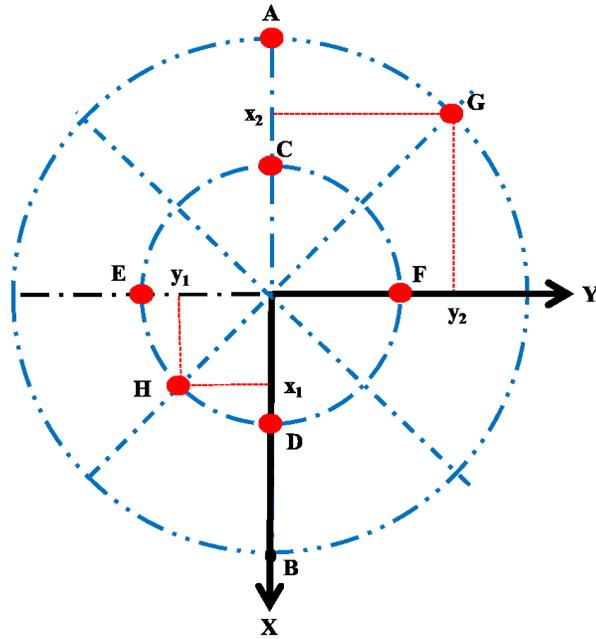


Figure 5.33. The robot geometrical boundary and position points for 90° turn.

The turning sequences are described in algorithm 5.4 and illustrated in Figure 5.34. The simulation sequences are depicted in Figure 5.35.

Algorithm 5.4 Pseudo-code for Turning 90° on the spot.	
Input	: Initial position, set points.
Output	: Trajectory of CoM and all joints position.
<i>Start</i>	
<i>/*1:</i>	Initialisation <i>robot moves its CoM to the required height (Z_h), && to the center of both foot, as depicted in Figure 5.34(1).</i>
<i>/*2:</i>	<i>CoM shift to right foot support point, Figure 5.34(2).</i>
<i>/*3:</i>	<i>Left foot ankle moves to point $A(-x_A, 0)$ and turn 90° through midpoint $G(x_2, y_2)$, && at the same time, the CoM turn 45°, Figure 5.34(3).</i>
<i>/*4:</i>	<i>CoM move to left foot support point $A(-x_A, 0)$, Figure 5.34(4).</i>
<i>/*5:</i>	<i>CoM turn another 45°, to face the final turning direction. Right foot move to point $D(x_D, 0)$ through midpoint $H(x_1, -y_1)$, && at the same time turn 90°, Figure 5.34(5).</i>
<i>/*6:</i>	<i>CoM move to right foot support point $D(x_D, 0)$, Figure 5.34(6).</i>
<i>/*7:</i>	<i>Left foot moves to point $C(-x_C, 0)$, Figure 5.34(7).</i>
<i>/*8:</i>	<i>CoM move to center point of both foot, Figure 5.34(8).</i>
<i>end</i>	

To turn the robot in any other angle of turning angle (θ), we can apply the general spline interpolation method to achieve the required motion. The sagittal and lateral trajectories of the foot ankle can be calculated with the following formula:

$$P_i(x_i, y_i) := \begin{cases} x_i = x_o + R \sin\left(\frac{\theta}{n}\right) \\ y_i = y_o + R \cos\left(\frac{\theta}{n}\right) \end{cases}, \quad (5.39)$$

where (x_o, y_o) is the initial position, R is the tuning radius, and n is the number of steps to complete the turn.

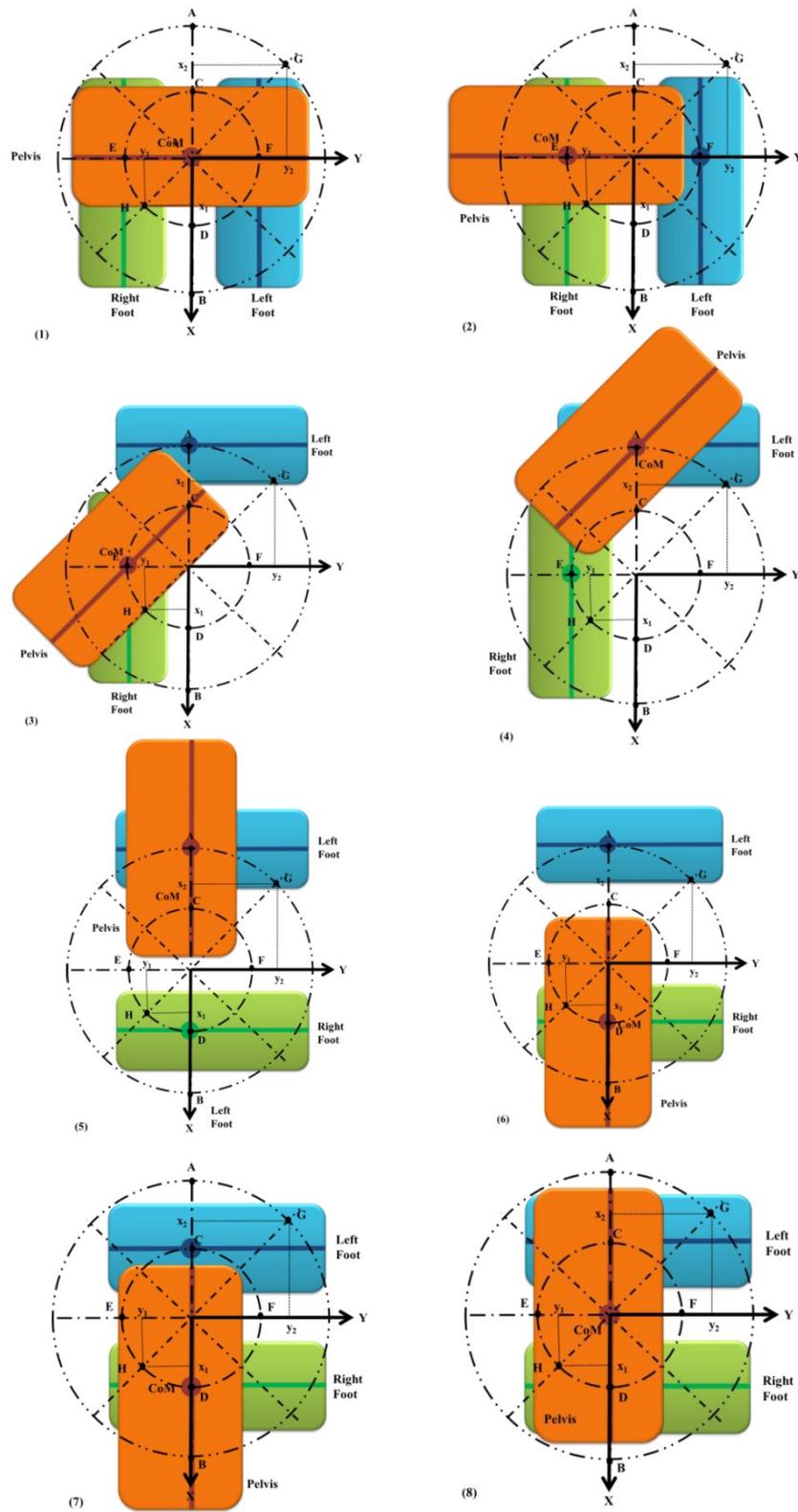
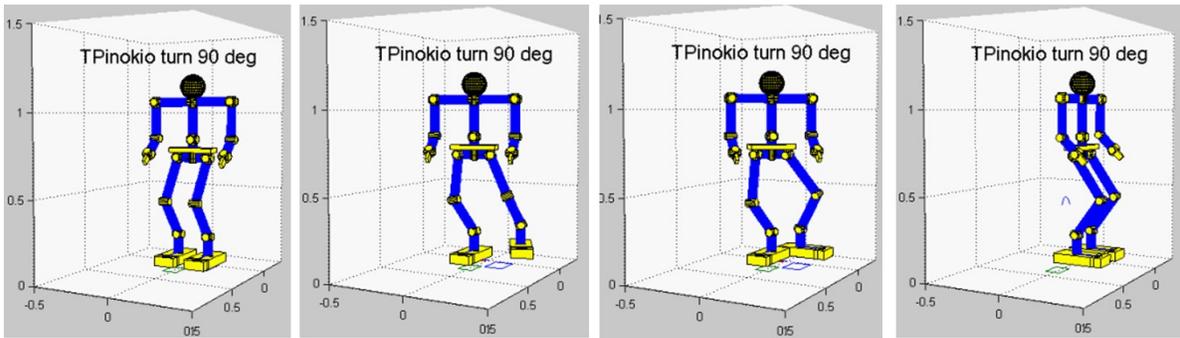
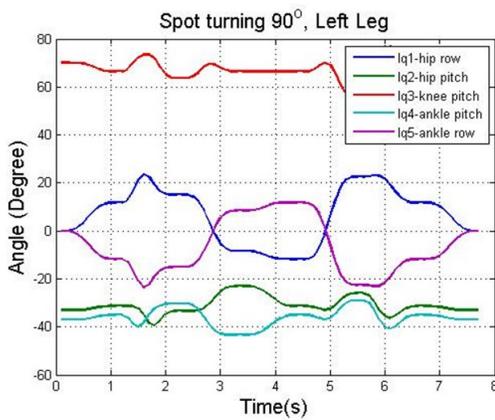


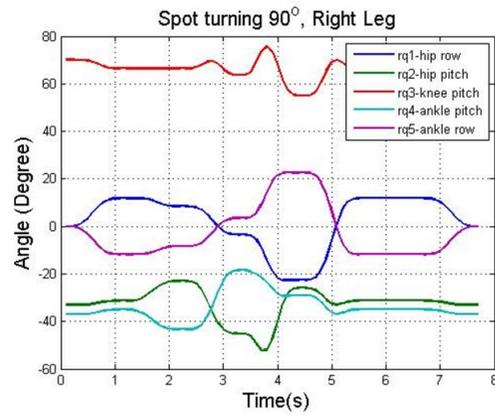
Figure 5.34. The sequences and robot CoM (pelvis) and foot position for spot-turning.



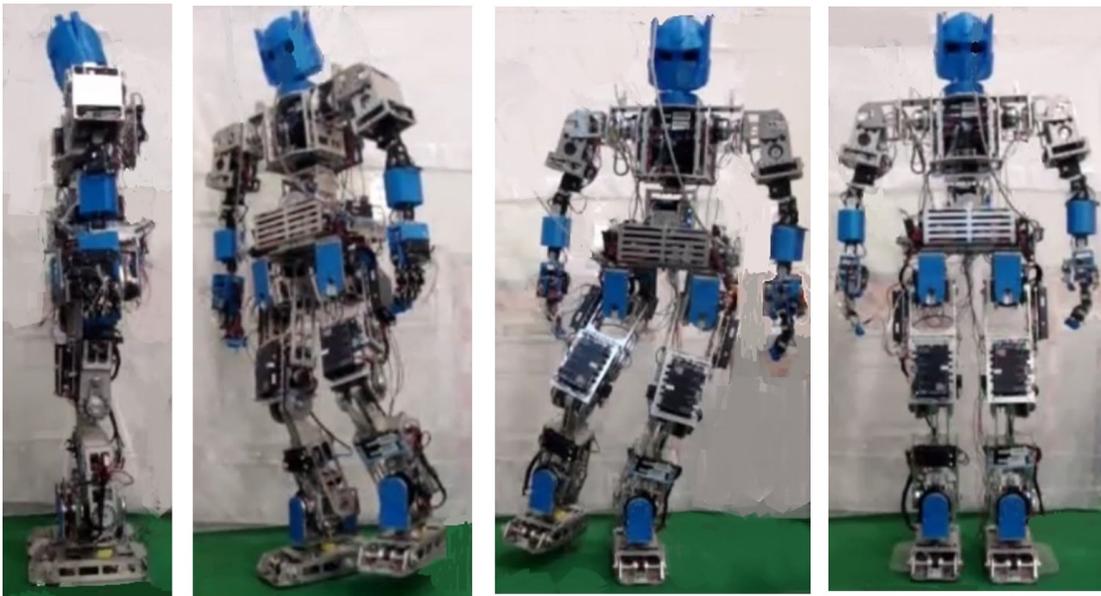
(a) Simulation model



(b) Left leg angle



(c) Right leg angle



(d) Turning experiment with robot

Figure 5.35. (a) simulation sequences, (b) Left leg joints angle, (c) Right leg joints angle, (d) Experiment with robot.

5.4 Chapter Summary

This chapter presents an optimal walking gait generation using the Augmented Model Predictive Control (AMPC) and the Inverted Pendulum Model (IPM) concept, which is a new approach based on the principle of preview control. The optimality in gait trajectory is achieved by model predictive control method that considers a trade-off between walking speed and stability. Piece-wise single step trajectory with double-mass inverted pendulum model is also discussed. The double-mass model method is suitable for a robot to make single large foot step or stepping over obstacle with dynamic balancing. The AMPC-PZC method reduces the model's undershoot water-bed effect by filtering the input ZMP reference waveform, this method produces a smooth output waveform and is suitable for general tracking. It is also presented that a robot with toe-feet can produce a more natural stretch walking posture, and can achieve a larger walking stride in comparison to a flat-feet design. This chapter also presented the geometrical method in determining the near-optimal parameters of a robot to perform stairs climbing and stepping over obstacle. The optimal walking gaits were experimented on TPinokio robot. The effectiveness of the AMP/ZMP/IPM techniques and the passive toe-feet design are validated by simulation testing with the robot walking on slope, stepping over obstacle, climbing stairs and turning on the spot.

CHAPTER 6

STABILITY AND POSTURE CONTROL

The Zero Moment Point (ZMP) concept and the Inverted Pendulum Model (IPM) were introduced and discussed in previous chapters. It was mentioned that bipedal locomotion control consists of a gait pattern generation block and a stabilizer block and was highlighted that without stabilization control, the bipedal cannot walk firmly. In this chapter, as illustrated in Figure 6.1, the stabilization concept which works along a given walking pattern and the posture control technique will be presented.

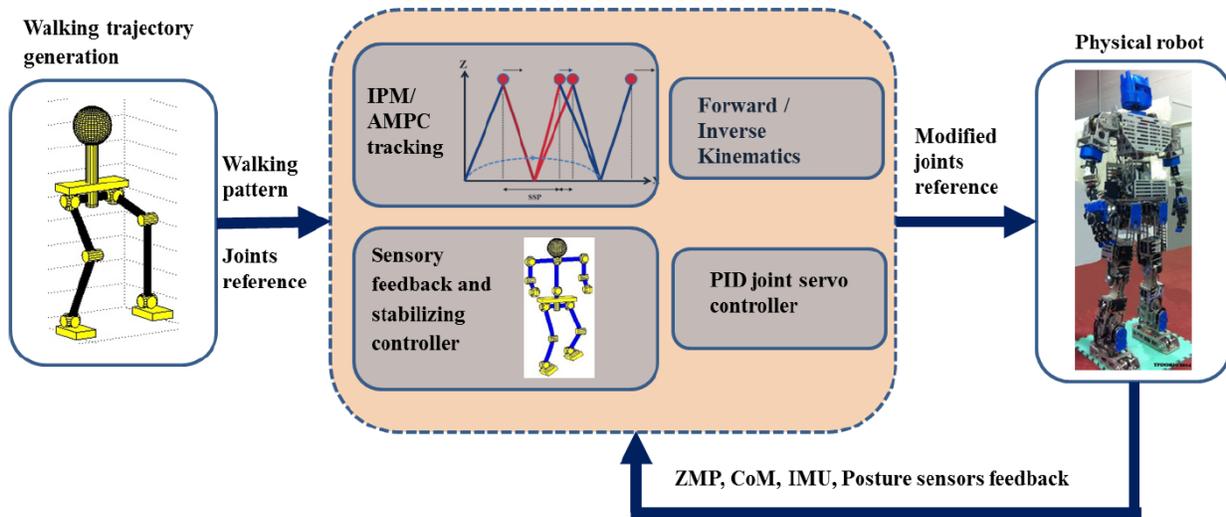


Figure 6.1. Controller and stabilizer block diagram.

6.1 Stability Margin and Support Convex Polygon (SCP)

The CoM/ZMP criterion specifies a bipedal robot to be dynamically stable if the Zero-Moment-Point remains within the support convex polygon (SCP) at all times [M. Vukobratovic, B. Borovac, 2004]. As depicted in Figure 6.2, if the ZMP remain within the SCP at all times, the biped is said to be statically stable. The stability margin of a bipedal is defined as how much the

ZMP is away from the center point of the SCP. The boundary of the constraint on the CoM in order for the biped to maintenance a stable posture is

$$ZMP_{min} \leq CoM \leq ZMP_{max} . \quad (6.1)$$

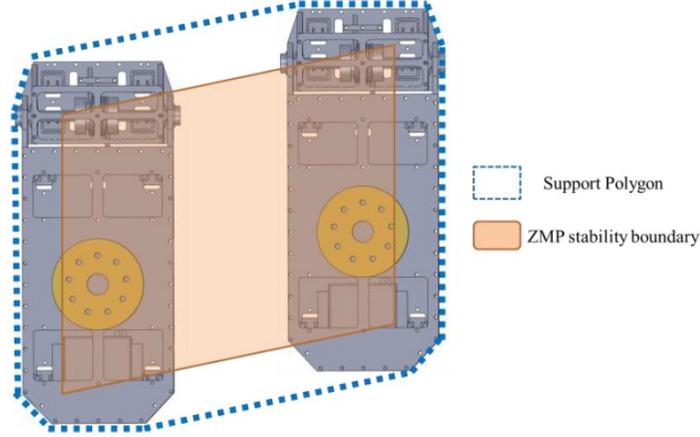


Figure 6.2. The support convex polygon (SCP) and the ZMP boundary.

6.2 Stability of Linear Inverted Pendulum Model

The dynamic equation of the inverted pendulum model in sagittal plane can also be expressed as

$$x_{zmp} = x_{com} - \frac{1}{\omega_n^2} \ddot{x}_{com} , \quad (6.2)$$

where $\omega_n = \sqrt{\frac{g}{z_{com}}}$ is the natural frequency of the pendulum.

The above equation shown that if the robot moves in a fast pace, the CoM amplitude will be reduced, which corresponds to dynamic walking mode. If the robot moves in a slower pace, the ZMP and CoM trajectories are almost similar, this corresponds to static walking. A lower CoM will improve the robot walking stability. With $z_{com} = 0.6m$, the natural frequency of the TPinokio robot is 4 Hz.

6.3 AMPC Stability Analysis

There are various approaches to measure and determine the stability of a robot locomotion gait. The general and basic approach is to study the CoM and ZMP trajectories to assess the stability

of the walking gait and this approach is feasible for both off-line and on-line gait assessments. In Chapter 5, various gait generation algorithms were introduced. However, due to unpredictable surface conditions and external disturbances, stabilizers with sensory feedback are being introduced into the controller to improve the system robustness. This section presents the stability analysis of the controllers designed in Chapter 5.

6.3.1 AMPC Model with Constraints on the Amplitude of the Control Variable

To achieve a stable walking, the system jerk should be kept as small as possible and this can be done by setting limits to constraint the amplitude of the control variable U by demand that

$$u_{min} \leq u(k) \leq u_{max} . \quad (6.3)$$

To demonstrate this, we done a simulation with AMPC model by limiting the x and y control variables with constraints $(-5 \leq u_y(k) \leq 5)$ and $(-10 \leq u_x(k) \leq 10)$, the control variables waveform is as shown in Figure 6.3. It is observed from Figure 6.4 that constraints on the control signal U are satisfied, the output CoM (red curve) response remains close to the desire ZMP (blue curve). We can modify the constraints to suit different working environment.

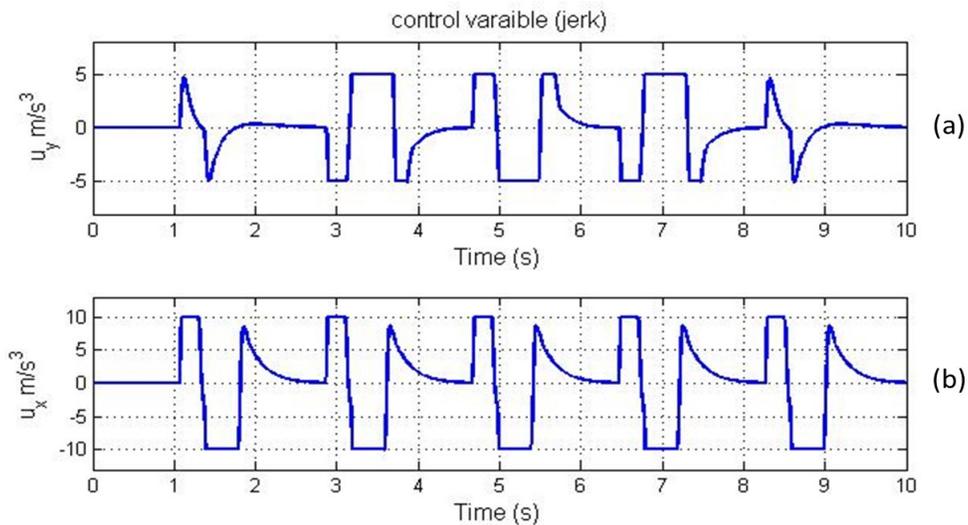


Figure 6.3. Control variables response with constraints.

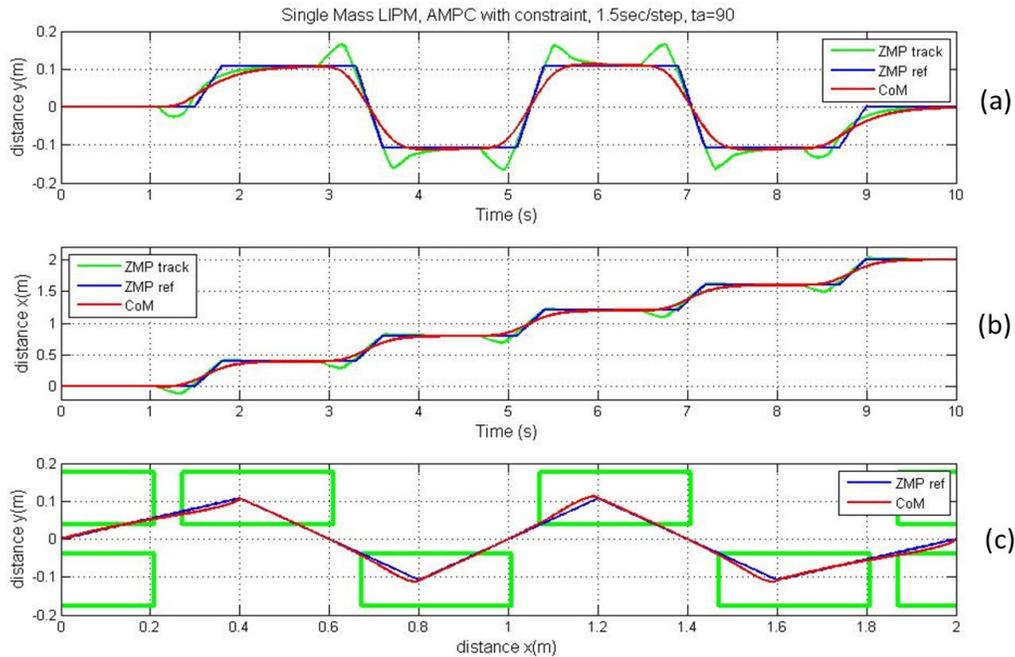


Figure 6.4. AMPC tracking trajectory with constraints.

6.3.2 Augmented MPC Model with Noise

One of the key features of AMPC predictive control is the ability to handle constraints in the design. To check the robustness of the AMPC model in tracking the desired ZMP, about random noise ($\pm 0.01\text{m}$) of about 10% of the reference signal was added to the reference ZMP waveform. The simulation results were as depicted in Figure 6.5. The results show that the AMPC model is still able to produce a good tracking even in a noisy environment.

Referring to Figure 6.6, it was observed that the control signal in the sagittal plane (u_x) produced a higher jerk than that in the frontal plane (u_y). This is because the bipedal robot takes a larger step (0.4m/step) in the sagittal-plane than the frontal-plane ($\pm 0.102\text{m}$).

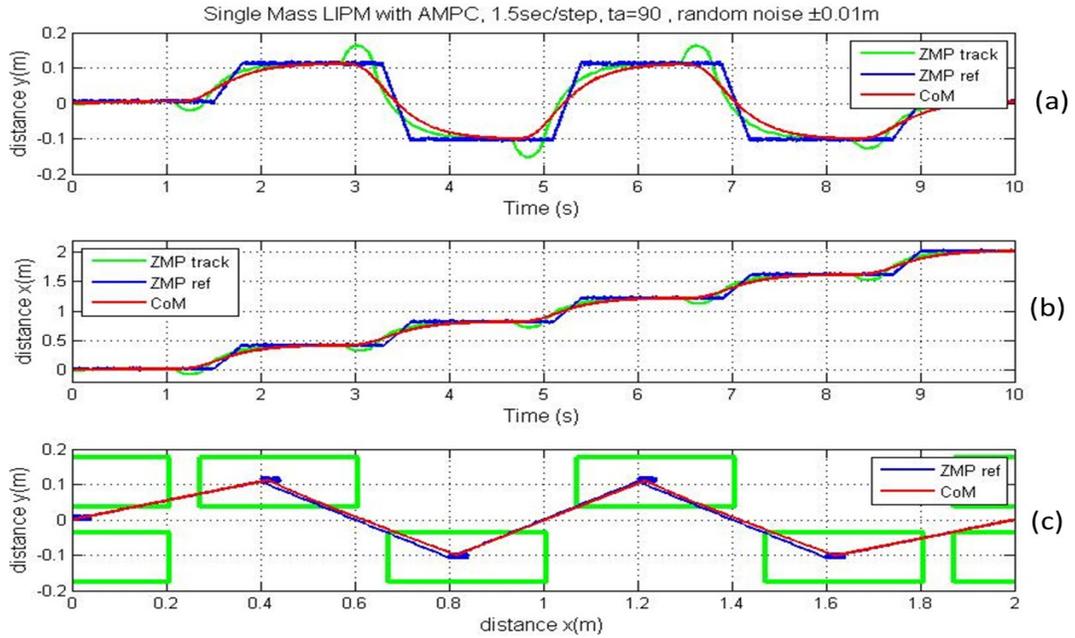


Figure 6.5. AMPC model tracking trajectory with noisy input.

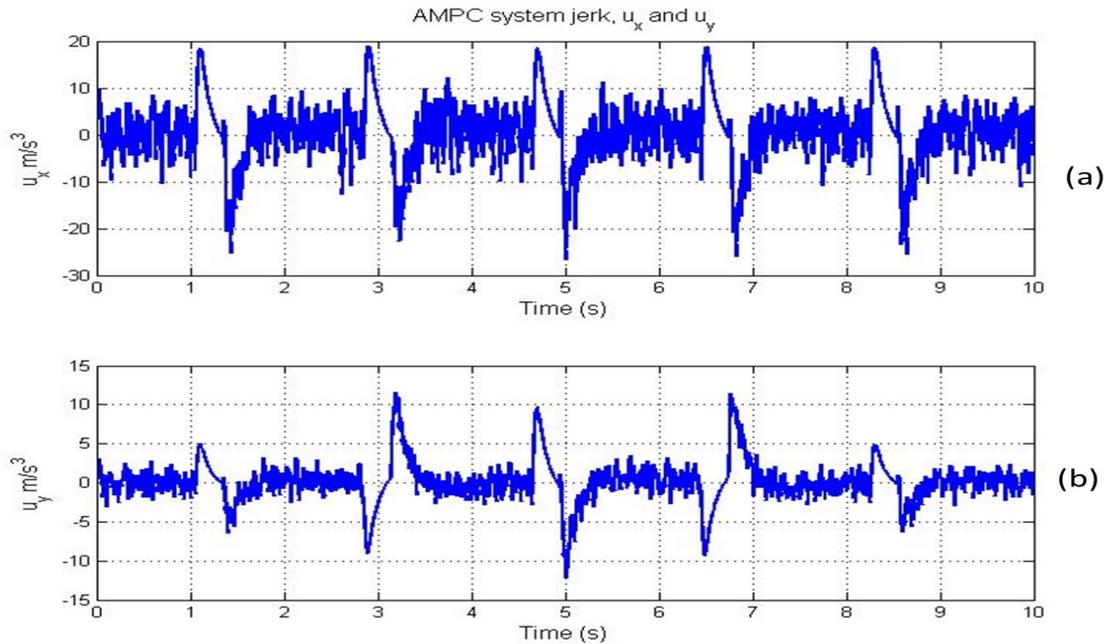


Figure 6.6. AMPC model system jerk response with noisy waveform.

6.3.3 AMPC-PZC Model with Noise

The same random noise of $\pm 0.01\text{m}$ is injected into the AMPC-PZC system, the simulation results are depicted in Figure 6.7 and Figure 6.8. It is observed that although the control signal had higher fluctuation, but the output trajectory still remain stable.

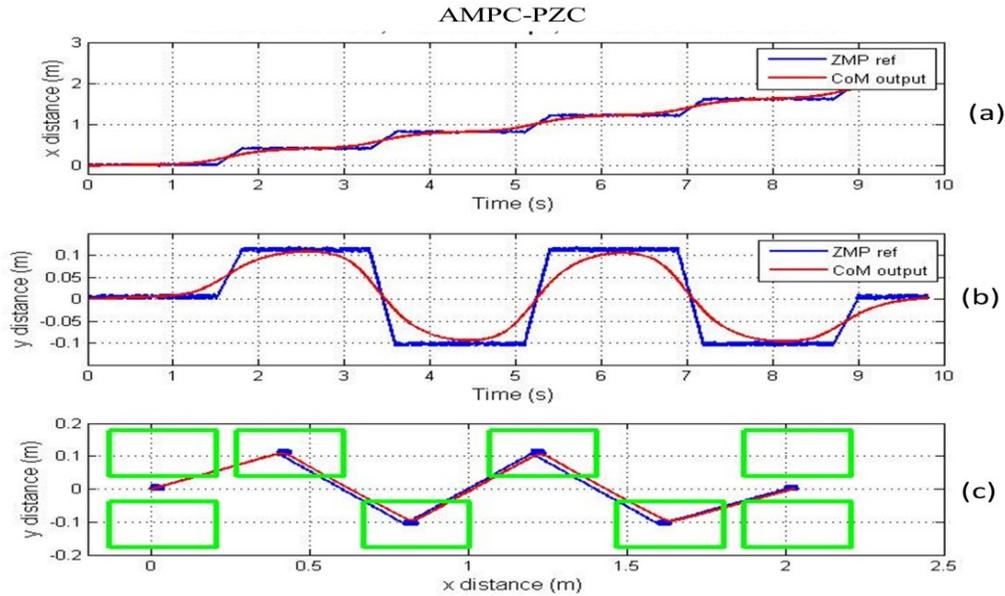


Figure 6.7. AMPC-PZC tracking waveform simulation with noise.

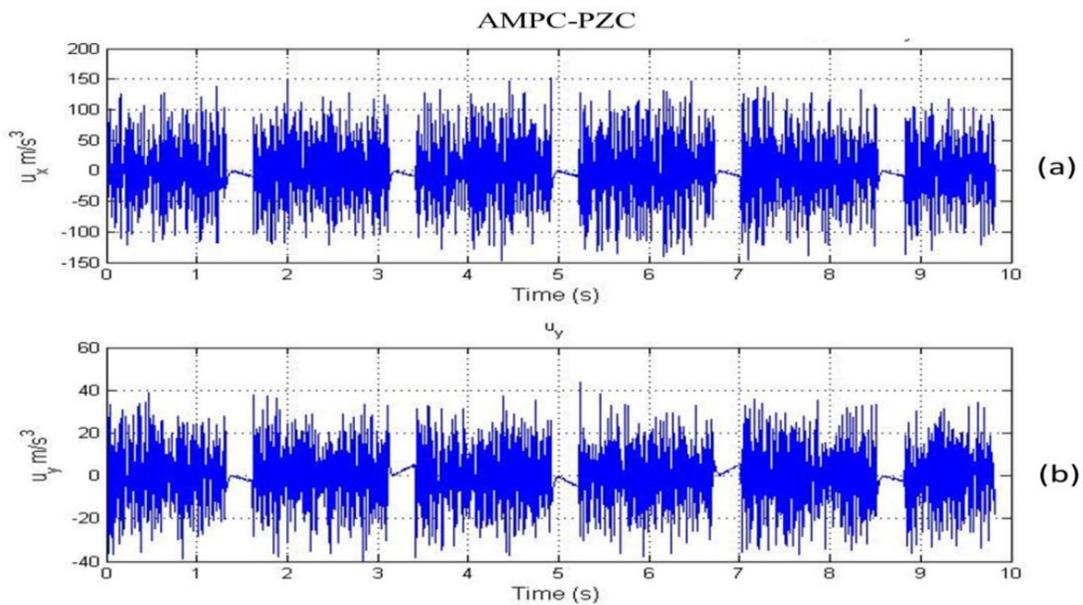


Figure 6.8. AMPC-PZC control signal simulation with noise.

6.3.4 Observer-based Model Predictive Control

An observer, as illustrated in Figure 6.9, may be used in a situation where the state variable $x(k)$ is not measurable, for instance, the CoM, is not measurable due to noisy environment. The robot state $x(k_i)$ can be estimated by an observer [L. Wang, 2009] of the form:

$$\hat{x}(k+1) = A\hat{x}(k) + B\Delta u(k) + K_{ob}(y(k) - C\hat{x}(k)). \quad (6.4)$$

The predictive control law is similar to the AMPC model discussed in chapter 5. We find ΔU by minimizing the cost function

$$J = (R_s - F\hat{x}(k))^T (\bar{R}_s r(k + t_a) - F\hat{x}(k)) - 2\Delta U^T \Phi^T (R_s - F\hat{x}(k)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U. \quad (6.5)$$

The optimal solution can be derived as

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - F\hat{x}(k)). \quad (6.6)$$

The optimal solution for $\Delta u(k)$ at time k is

$$\Delta u(k) = K_y r(k + t_a) - K_{mpc} \hat{x}(k). \quad (6.7)$$

The closed-loop characteristic equations and the eigenvalues of the closed-loop system are obtained as follows: nb

$$x(k+1) = Ax(k) + B\Delta u(k) = Ax(k) + BK_y r(k + t_a) - BK_{mpc} \hat{x}(k), \quad (6.8)$$

substitute $\Delta u(k)$ with equation (6.7), and define the close loop observer error equation as

$$\tilde{x}(k+1) = (A - K_{ob}C)\tilde{x}(k), \quad (6.9)$$

where $\tilde{x}(k) = x(k) - \hat{x}(k)$, equation (6.8) can be rewritten as

$$x(k+1) = (A - BK_{mpc})x(k) + BK_{mpc}\tilde{x}(k) + BK_y r(k + t_a). \quad (6.10)$$

Combining equation (6.9) and (6.10) leads to

$$\begin{bmatrix} \tilde{x}(k+1) \\ x(k+1) \end{bmatrix} = \begin{bmatrix} A - K_{ob}C & 0_{n \times n} \\ BK_{mpc} & A - BK_{mpc} \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} 0_{n \times m} \\ BK_y \end{bmatrix} r(k + t_a). \quad (6.11)$$

The characteristic equation of the close-loop state space system can be obtained as

$$\det \left[\lambda I - \begin{bmatrix} A - K_{ob}C & 0_{n \times n} \\ BK_{mpc} & A - BK_{mpc} \end{bmatrix} \right] = 0, \quad (6.12)$$

and the eigenvalue can be determined which is equivalent to

$$\det(\lambda I - (A - K_{ob}C)) \det(\lambda I - (A - BK_{mpc})) = 0. \quad (6.13)$$

This means that the robot closed-loop model and the state estimator have two independent characteristic equations

$$\begin{aligned} \det(\lambda I - (A - K_{ob}C)) &= 0, \\ \det(\lambda I - (A - BK_{mpc})) &= 0. \end{aligned} \quad (6.14)$$

As such, one can design the predictive control law and the observer separately.

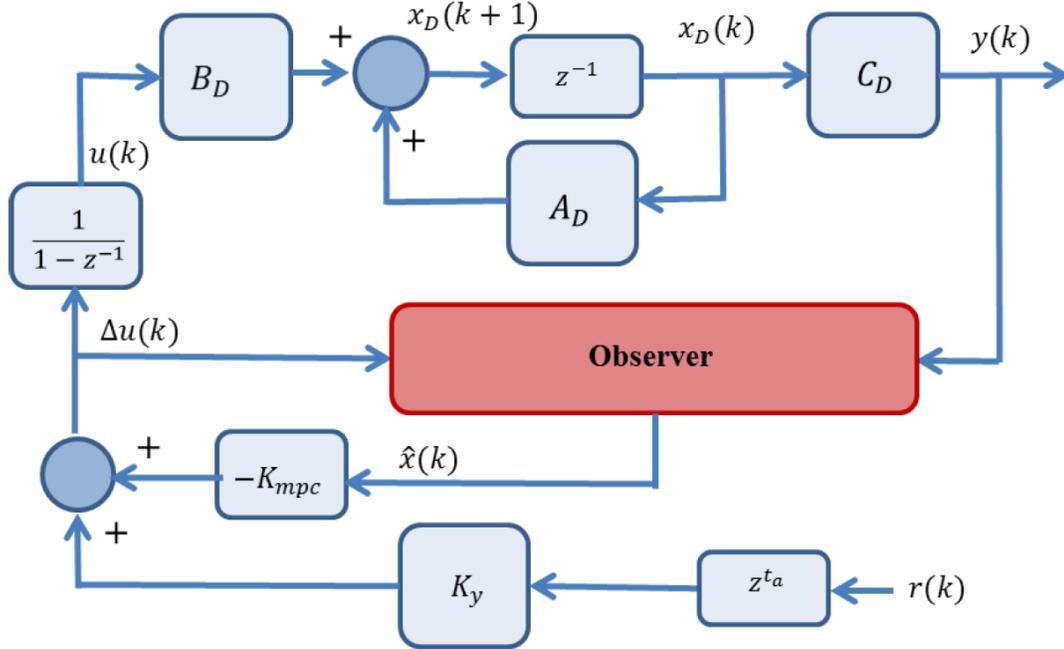


Figure 6.9. Block diagram of AMPC with observer.

6.4 Balancing Control with Sensors Feedback

During the walk cycle, the hip/pelvis link of the robot may not always maintain in a horizontal position due to gravitational, vibration and other mechanical structural limitations. As a result, the swing leg is prone to collide with the stance leg. Standing or walking on slope poses another difficulty in maintaining a stable posture. This section will discuss and propose some simple corrective actions to improve the robot stability.

A nine-axis Inertial Measurement Unit (IMU) is mounted on the robot pelvis center position to measure the robot roll, pitch and yaw motion and another two three-axis tilt sensors are mounted on the robot ankle joint to measure the floor inclination during walking, and also to provide feedback when the robot swing foot touch down on ground surface. These readings from the sensors will compensate for the ground condition and any system's errors accordingly.

6.4.1 Collision Avoidance Control

During locomotion, leg collision must be avoided in order to achieve a stable walking at all times when the robot swings its legs. As illustrated in Figure 6.10, a bipedal robot is not a complete *rigid-body* as was assumed, and in fact some compliances and flexing were observed in the joint and link when a robot walks. To compensate for the hip angle roll error θ_{roll} , an angular compensator can be easily implemented using a P-type controller, and the feedback error angle can be monitored and measured by using an IMU sensor. This can be done as follows:

Compensation angle for right hip roll, where k_r is the gain, is

$$\begin{aligned}\Delta\theta_{r1} &= k_r|\theta_{roll}|, \\ \theta_r &= \theta_{r1} \pm \Delta\theta_{r1}.\end{aligned}\tag{6.15}$$

And, compensation angle for left hip roll, where k_l is the gain, is

$$\begin{aligned}\Delta\theta_{l1} &= k_l|\theta_{roll}|, \\ \theta_l &= \theta_{l1} \pm \Delta\theta_{l1}.\end{aligned}\tag{6.16}$$

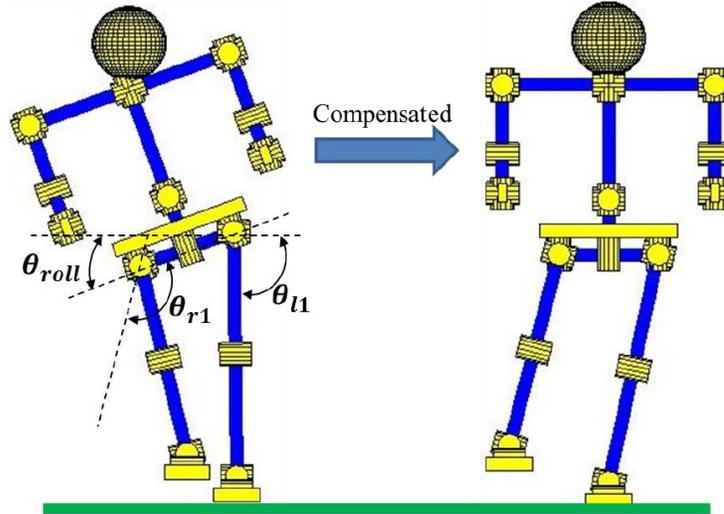


Figure 6.10. Roll angle compensation of swing leg for frontal plane.

6.4.2 Slope Balancing Control

In order to achieve a more stable walking or balancing on a slope, the robot torso and ankle angle must be monitored and compensated accordingly based on the IMU feedback data. This controller can be implemented with simple P-type or a PI-type controller, as illustrated in Figure 6.11 and Figure 6.12.

Slope compensator (Sagittal plane), where k is the gain, is

$$\theta_4 = \theta'_4 \pm k|\theta_{pitch}|. \quad (6.17)$$

Slope compensator (Frontal plane), where k_p and k_r is the gain, is

$$\begin{aligned} L_l &= L'_l + \left(k_{pl} + \frac{k_{ll}}{s}\right)\phi_{roll}, \\ L_r &= L'_r + \left(k_{pr} + \frac{k_{lr}}{s}\right)\phi_{roll}, \\ \phi_5 &= \phi'_5 + \tan^{-1}\left(\frac{L_r - L_l}{d}\right). \end{aligned} \quad (6.18)$$

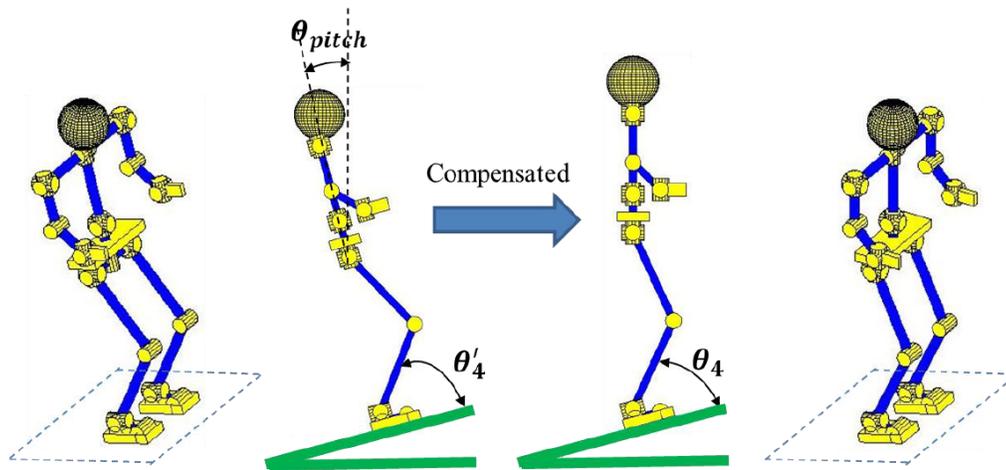


Figure 6.11. Sensor feedback compensation on slope surface for sagittal plane.

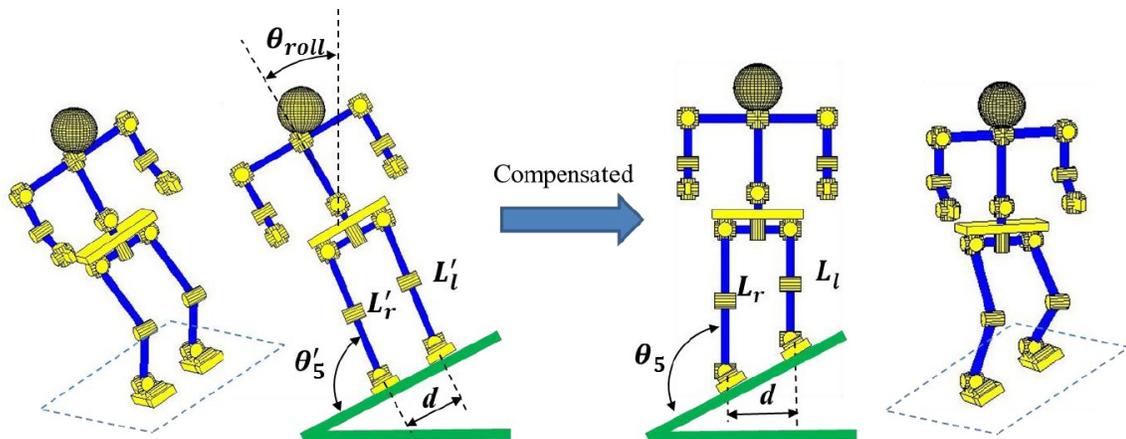


Figure 6.12. Sensor feedback compensation on slope surface for frontal plane.

6.4.3 Foot Impact Shock Absorber Stabilizer

During walking, once the swing foot of the biped contacts the ground, the impact reacting force from the ground may cause stability issues because the impact will generate an undesired push-off force from the ground. As depicted in Figure 6.13, compliance or soft material is used on the sole of the bipedal to soften the landing impact force, and the F/T sensor provides feedback to the stabilizing controller to bend the leg to act as like a spring-mass damper shock absorber to

reduce the ground impact force. The damping response simulation is as shown in Figure 6.14, the damping parameters $b=7000$ and $k=60000$ are chosen experimentally.

$$z_{dp} = \frac{F_z(s)}{ms^2 + bs + k} \quad (6.19)$$

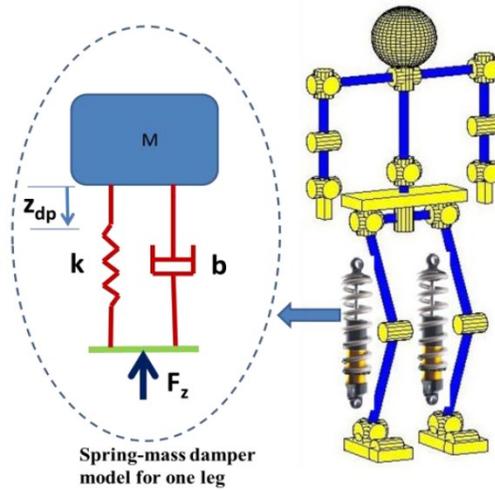


Figure 6.13. Spring-mass model of a robot with bending legs.

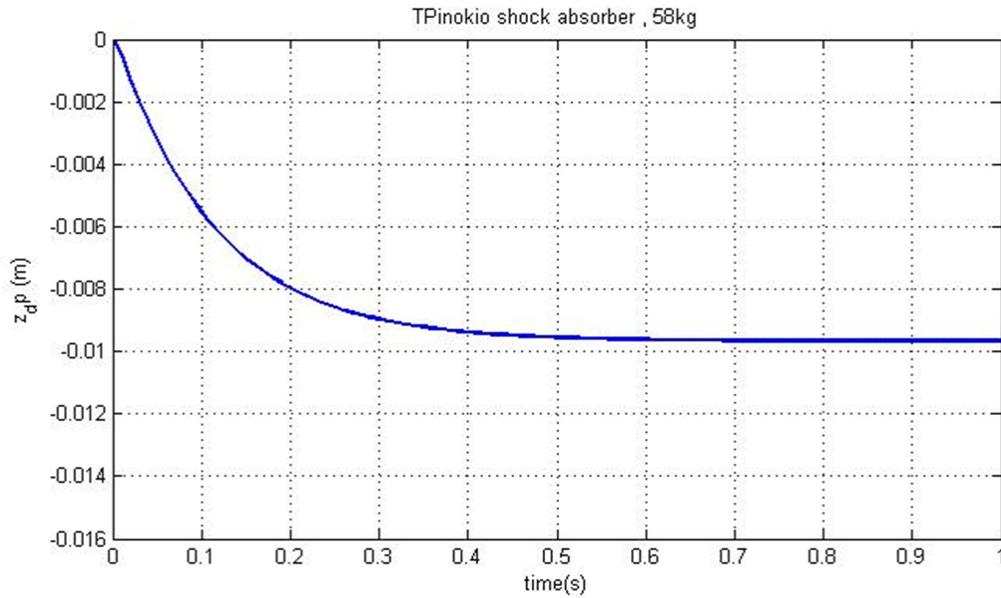


Figure 6.14. Damping response of the shock absorber with $T=1$ second.

6.5 Chapter Summary

This chapter presents the differences in balance and stability measurement for the bipedal robot. The AMPC method with noise injected into the reference waveform and the measured ZMP are identified and the method to correct the CoM position is proposed. The ZMP criterion is used to determine the stability margin. We also analyzed the method of accessing stability and balance based on the Support Convex Polygon (SCP) boundaries for the TPinkio robot. Discrepancies between the measured CoM through the inverse kinematics method are corrected using posture compensation controller with IMU and F/T sensors.

CHAPTER 7

CONCLUSION AND CONTINUOUS RESEARCH

Stable humanoid bipedal locomotion is still an ongoing and challenging research activity for researchers from the robotics community. The research works of the TPinokio humanoid robot began in 2011, and has made significance progress till now. The primary objective and milestone of this research was to design, build and develop a modeling framework for a teen-size bipedal humanoid robot that can walk in a steady manner both statically and dynamically in an unstructured environment such as on slope and stairs. In this chapter, some discussion on bipedal walking issues and to what extent the future continuous research direction should be formulated is presented. Please refer to <http://www.tpinokio.com> for continuous update on the research works.

7.1 Research Summary

The aim of this thesis is to present the works done during the past three years. During the first one and a half year, most of the effort was concentrated on developing the mechanical structure, interfacing the motion control electronics, assembling the gear-drive mechanism and testing the sensors interface. In the following two years, research efforts were spent on the system modeling, walking gait generation, simulation, stability analysis and experimental testing. The accomplishments for this research are as mentioned below:

- The primary objective of developing a humanoid bipedal robot with toe-feet, its sensory system and drive mechanisms has been successfully implemented;
- The kinematics and dynamics model of the robot is successfully developed and implemented with MATLAB for simulation and testing of various control algorithms;
- A stable gait generation and control method using AMPC/LIPM model has been proposed and simulated with MATLAB, and tested on the physical robot;

- Simplified model of the robot walking on slope, stairs, and step over and obstacle is implemented and simulated with MATLAB, and tested on the physical robot.

7.2 Continuous Research Activities and Recommendations for Future Works

Although in this thesis, different techniques and suggestions to achieve stable bipedal walking on different surfaces are proposed and simulated, a more detail study and modeling of the robot walking characteristics will be needed in order to develop a more advance human-like walking posture and mechanisms. The ultimate goal is to develop a bipedal robot capable of becoming a domestic helper in area of assistive robotics in the elderly care sector. Several continuous activities and directions for future research with the TPinkio robot are briefly discussed in the subsequent sections. A number of interesting extensions of the research work presented in this thesis are now suggested as future research works.

7.2.1 Kneeling Down and Recovery Posture

Kneeling down to pick up an object on the floor is the next natural task that should be performed by a bipedal robot. To kneel down, pick up an object and recovery to it standing position with the robot still maintains good stability is a challenging task [K. Fujiwara, F. Kanahiro et al., 2003], because this involves whole body movement, coordination of legs, hip, upper body and arms motion and not forgetting that the dynamic of the robot is changed after picking up an object. Due to the limitation of the fingers joints grasping capability, the TPinkio is designed to pick up object that weighs less than 2kg. Figure 6.1 depicts the kneeling down sequences. Note that the CoM and support polygon is different for every different motion frame segment.

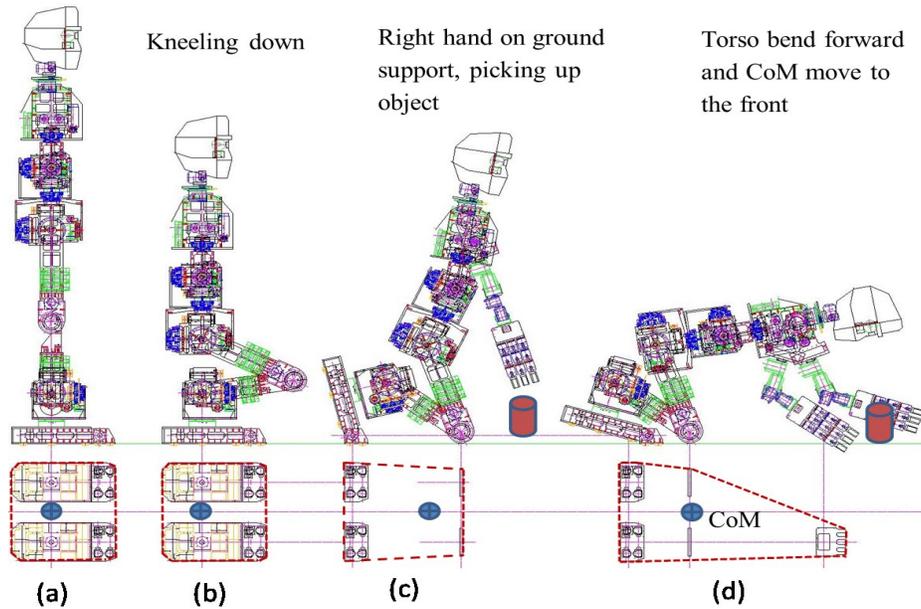


Figure 7.1. Kneeling down sequences.

7.2.2 Landing Stability

Bipedal locomotion is sensitive to the environment perturbation. For an ideal structural mechanism and a perfect structured even surface, the robot is able to execute smoothly with any prescribed motion trajectories. However, this is not the case in a real environment. The stability conditions need to be established, especially for a stable foot landing on unknown and uneven terrain. Landing stability analysis is important in establishing a stable walking posture, for example, the foot-ground contact may be modeled like a spring-damper mechanism. These aspects are worth for a detail study to increase the robot locomotion robustness.

7.2.3 Active Toe-joint and Toe-foot Rotation Characteristics

As discussed in Chapter 5, robot with toe-foot results in the reduction of rotation speed of knee angle and the increase in step length. Besides that, toe-foot produced more natural walking gait and posture. The TPinokio robot is designed with passive toe-foot and is capable for stretch walking. Bipedal walking with toe-foot rotation is under-actuated by one DoF. Due to the passive DoF at the joint between the toe and the contact surface, the biped is modeled as an

under-actuated system. The stability aspects and energy efficiency of such configurations are worth further research study especially with active-toe-foot.

7.2.4 Rotation of Posture Using Slippage Turning Technique

A simple and effective spot-turning method is proposed in Chapter 5. The assumption is that there is no slippage between the contact surface and the stance foot/feet. In such a case, the robot tends to take smaller steps when turning and resulting in a slower movement because the swing foot has to be completely lifted from ground, and this is not energy efficient. Another method which is using foot slippage and ground frictional forces to assist in body turning is an interesting research topic [K. Miura, F. Kanehiro et al.,2013]. To effectively implement the slippage technique, the robot spinning dynamic model, contact forces and slippage phenomenon worth further investigation and study.

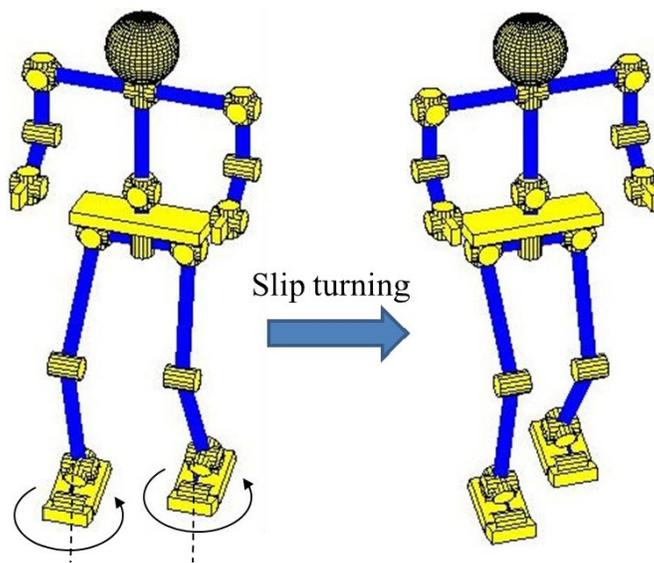


Figure 7.2. Slippage turning with toe.

7.2.5 Motion with external object

Some typical examples of a robot using multiple non-coplanar contacts to perform a dynamic motion and task are illustrated in Figure 7.3. In such a situation, the CG and ZMP position will be modified by the whole body mass and the external object and the hand reaction force will

modify the force distribution in the robot body [L. Saab, O.E Ramos et al., 2013]. The generation of motion and controlling of the robot in such environment is a challenging problem worth further investigation and study.

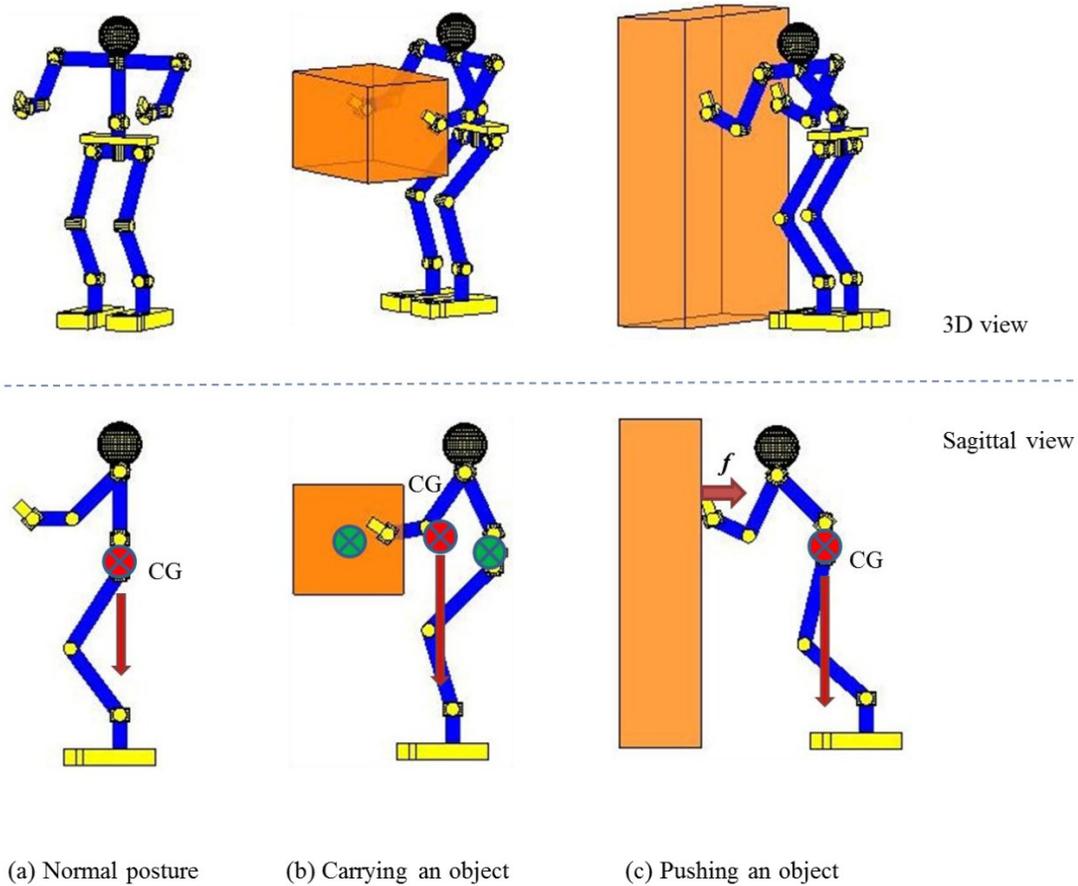


Figure 7.3. Modification of CoG and ZMP position.

7.2.6 Running Motion and Control

The most obvious difference between walking and running gait is that during walking, one leg always stays on the contact surface while the other is swinging in air. In running, there is typically a *flight* phase (Edward Muybridge's 1887 motion study of an athlete running) as depicted in Figure 7.4, where the robot is airborne with both feet in the air [R. Tajima, D. Honda et al., 2009], [S. KAJITA, TAKASHI et al., 2007], [K. Arikawa, T. Mita, 2002]. Making a bipedal robot that is capable of running is an extremely challenging task with current knowledge

and technologies. As illustrated in Figure 7.4, there are three major technical issues that need to be looked into: high impact reaction force; floor slippage while pushing and jumping forward; and the spinning reaction force generated by the fast motion. To address and solve these challenging issues, a faster controller to response to the reaction force is needed to stabilize the whole-body coordination to prevent slippage, and to coordinate the robot's upper body with arm motion to counter act the spinning force. Further study and research activities are needed to understand both the complexity and the motion characteristics of the bipedal body flight dynamic.

Edward Muybridge's 1887 motion study of an athlete running.

[http://www.nytimes.com/2004/11/17/health/17cnd-run.html?_r=0,]©

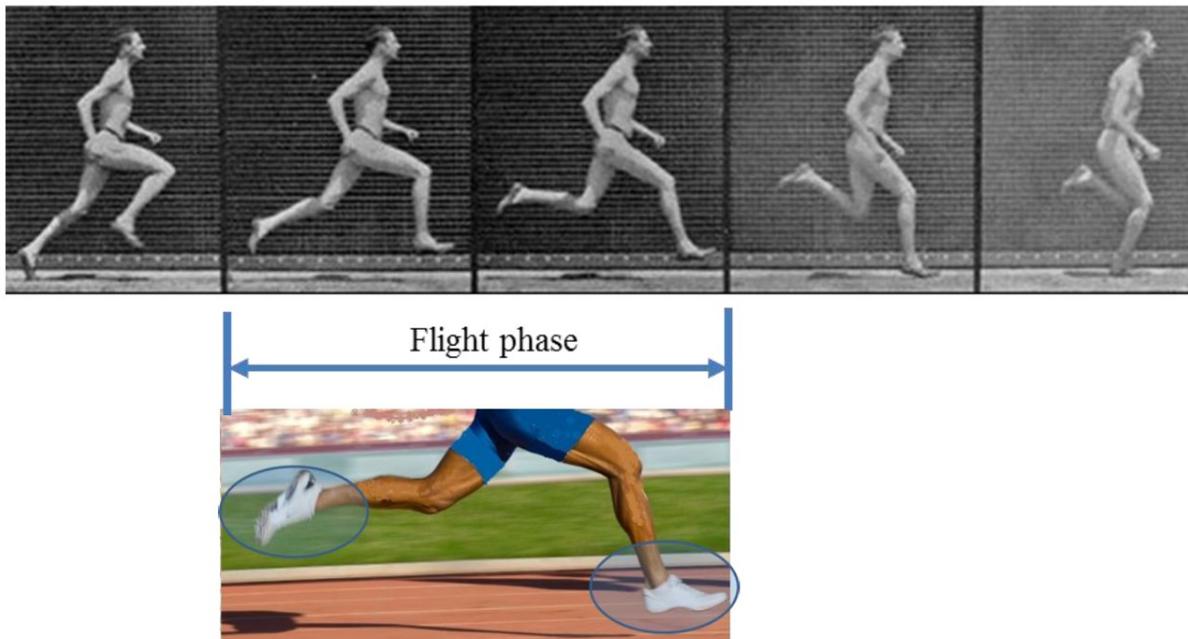


Figure 7.4. Human running motion and flight phase.

Bibliography

[A. Astolfi, R. Ortega et al., 2009] A. Astolfi, R. Ortega, A. Venkatraman, *A globally exponentially convergent immersion and invariance speed observer for n degrees of freedom mechanical systems*, Proceedings of the 48th IEEE Conf on Decision and Control, Shanghai China, 2009, pp. 6508-6513.

[A. Astolfi, R. Ortega et al., 2010] A. Astolfi, R. Ortega, A. Venkatraman, *Global observer design for mechanical systems with non-holonomic constraints*, American Control Conference (ACC), 2010, pp. 202-207.

[A.G.Baydin, 2012] A.G. Baydin, Evolution of central pattern generators for the control of a five-link bipedal walking mechanism, PALADYN Journal of Behavioral Robotics, 2012, pg 1-11.

[A. Goswami, 1999] A. Goswami, *Postural stability of biped robots and the foot-rotation indicator(FRI) point*, Int. J. Robot. Res. 18(6), 1999, pp. 523–533.

[A. Herali, K. Wolff et al., 2007] A. Herali, K. Wolff, M. Wahde, Central Pattern Generators for Gait Generation in Bipedal Robots, Humanoid Robots, New Developments, I-Tech Education and Publishing, Book, 2007, pg 285-305.

[A. Sabanovic, K. Ohnishi, 2011] A. Sabanovic, K. Ohnishi, *Motion Control Systems*, Book, John Wiley & Sons, IEEE Press 2011.

[B.H. Jesse, S.B Dennis et al., 2007] B.H. Jesse, S.B. Dennis, *Nonminimum-Phase Zeros : Much To Do About Nothing, Part II*, Humanoid Robots, IEEE Control System Magazine, June 2007, pp. 45-57.

[B. Siciliano, O. Khatib, 2008] B. Siciliano, O. Khatib, *Springer Handbook of Robotics*, 2008.

[B. Vanderborght, B. Verrelst et al., 2005] B. Vanderborght, B. Verrelst, R.V. Ham, M.V. Damme, D. Lefeber, *A pneumatic biped: experimental walking results and compliance adaptation experiments*, Humanoid Robots, 5th IEEE-RAS International Conference, 2005, pp. 44-49.

[C. Chevallereau, G. Bessonnet et al., 2009] C. Chevallereau, G. Bessonnet, G. Abba, Y. Aoustin, *Bipedal Robots: Modeling, Design and Walking Synthesis*, Book, Wiley, 2009.

[C. Chevallereau, J.W. Grizzle et al., 2010] C. Chevallereau, J.W. Grizzle, C.L. Shih, *Steering of a 3D Bipedal Robot with an Underactuated Ankle*, IROS, Taipei, Taiwan, October 2010.

- [C.L. Shih, 1999] C.L. Shih, *Ascending and Descending Stairs for a Biped Robot*, IEEE Transactions on system, man, and cybernetics, part A: system and humans, Vol.29, No.3, 1999, pp 255-268.
- [C.W. Wampler, A.P. Morgan et al., 1990] C.W. Wampler, A.P. Morgan, A.J. Sommese, *Numerical continuation methods for solving polynomial systems arising in kinematics*, ASME J. Mech. Des. 112, 1990, pp. 59-68.
- [C. Y. Hwan, O. Yonghwan et al.,2008] C. Y. Hwan, O. Yonghwan, K. Doik, H. eokmin, *Balance Control in Whole Body Coordination Framework for Biped Humanoid Robot MAHRUR*, Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication 2008, pp. 401-406.
- [D. B. Marghitu, 2009] D.B. Marghitu, *Mechanisms and Robots Analysis with MATLAB*, Book, Springer 2009.
- [D.J. Braun, J.E. Mitchell et al. 2012] D.J. Braun, J.E. Mitchell, M. Goldfarb, *Actuated Dynamic Walking in a Seven-Link Biped Robot*, IEEE/ASME Trans. on Mechatronics, vol. 17, no. 1, Feb 2012, pp. 147-156.
- [D. Manocha, J. Canny, 1992] D. Manocha, J. Canny, *Real Time Inverse Kinematics for General 6R Manipulators* Tech. rep. University of California, Berkeley, 1992.
- [D. Tlalolini, Y. Aoustin et al., 2010] D. Tlalolini, Y. Aoustin, C. Chevallereau, *Design of a walking cyclic gait with single support phases and impacts for the locomotor system of a thirteen-link 3D biped using the parametric optimization*, Multibody System Dynamic, 2010, pp. 33-56.
- [D. Pieper, 1968] D. Pieper: *The Kinematics of Manipulators Under Computer Control*. Ph.D. Thesis, Stanford University, Stanford. 1968.
- [E. Gross, M. Tomizuka, 1994] E. Gross, M. Tomizuka, *Experimental Flexible Beam Top Tracking Control with a Truncated Series Approximation to Uncancelable Inverse Dynamics*, IEEE Trans. on Control, Systems Technology, Vol 2, No. 4, 1994, pp.382-391.
- [E.R. Westervelt, J.W. Grizzle et al., 2007] E.R. Westervelt, J.W. Grizzle, C. Chevallereau, J.H. Choi, *Feedback Control of Dynamic Bipedal Robot Locomotion*, Book, CRC Press 2007.
- [F. Miyazaki, S. Arimoto et al., 1980] F. Miyazaki, S. Arimoto, *A control theoretic study on dynamical biped locomotion*, ASME J. Dyn., Syst., Meas., Contr., vol. 103, 1980. pp. 233–239
- [G. End, J. Nakanishi et al., 2005] G.End, J. Nakanishi, J. Morimoto, G. Cheng, *Experimental Studies of a Neural Oscillator for Biped Locomotion with QRIO*, International Conference on Robotics and Automation Barcelona, Spain, April 2005, pp. 596-602.

[G. Capi, M. Yokota et al., 2006] G. Capi, M. Yokota, *Optimal Multi-Criteria Humanoid Robot Gait Synthesis: An Evolutionary Approach*, International Journal of Innovative Computing, Information and Control, Volume 2, Number 6, December 2006, pp. 1249-1258.

[G. Dip, V. Prahlad et al. 2009] G. Dip, V. Prahlad, P.D. Kien, *Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed*, Robotica volume 27, 2009, pp. 355-365 .

[G.Q. Zhang, M. Xie et al., 2009] G.Q. Zhang, M. Xie, H. Yin, L. Wang, H.J. Yang, *Planning and Control of Biped Walking along Curved Paths on Unknown and Uneven Terrain*, ICIRA 2009, LNAI 5928, 2009, pp. 1032–1043.

[H. Chr, S. Thoss et al., 2006] H. Chr, S. Thoss, F. Schmelz, E.Aucktor, *Universal Joints and Driveshafts, Analysis, Design, Applications*, Book, Springer 2006

[H. Dallali, M. Brown et al., 2009] H. Dallali, M. Brown, B. Van, *Using the Torso to Compensate for Non-Minimum Phase Behaviour in ZMP Bipedal Walking*, advance in robotics research, book, springer 2009, pp191-202.

[H.J. Luinge, 2002] H.J. Luinge, *Inertial Sensing of Human Movement*, Twente University Press, 2002.

[H. Khalil, 2002] H. Khalil, *Non-linear Systems*, book, Prentice Hall, 2002.

[H. Miura, I. Shimoyama et al., 1984] H. Miura, I. Shimoyama, *Dynamic walk of biped locomotion*, Int. J. Robot. Res., vol. 3, no. 2, pp. 60–74, 1984.

[H.Y. Anthony, N. Michel et al., 1998] H.Y. Anthony, N. Michel, L. Hou, *Stability Analysis of Systems with Impulse Effects*, IEEE Trans. on Automatic Control, VOL. 43, NO. 12, Dec 1998, pp 1719-1723.

[I.W. Park, J.Y. Kim et al., 2005] I.W. Park, J.Y. Kim, J. Lee, J.H. Oh, *Mechanical Design of Humanoid Robot Platform KHR-3 (KAIST Humanoid Robot - 3: HUBO)*, Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots, pp. 321-326.

[I.W. Park, J.Y. Kim et al., 2006] I.W. Park, J.Y. Kim, J.H. Oh, *Online Biped Walking Pattern Generation for Humanoid Robot KHR-3 (KAIST Humanoid Robot-3: HUBO)*, Humanoid 2006, pp. 398-403.

[J.A. Rossiter, 2004] J.A. Rossiter, *Model Based Predictive Control, A Practical Approach*, Book, CRC press, 2004.

[J. Furusho, A. Sano et al., 1990] J. Furusho, A. Sano, *Sensor-based control of a nine-link biped*, Int. J. Robot. Res, 1990, vol. 9, no. 2, pp. 83-98.

[J. Craig, 2005] J. Craig, *Introduction to Robotics*, Pearson Education, Prentice Hall, 2005.

- [J.P. Ferreira, M. M. Crisóstomo et al., 2013] J.P. Ferreira, M. M. Crisóstomo, A.P. Coimbra, *Adaptive PD Controller Modeled via Support Vector Regression for a Biped Robot*, IEEE Trans. on Control Systems Technology, vol. 21, No. 3, May 2013, pp. 941-949.
- [J.U. Cho, J.S. Yeon et al., 2012] J.U. Cho, J.S. Yeon, J.H. Park, *Stable Running Velocity Change of Biped Robot Based on Virtual Torque*, IEEE Int. Conf. on Robotics and Biomimetics, Dec 2012, pp.301-306.
- [J.W. Grizzle, C. Chevallereau et al., 2010] J.W. Grizzle, C. Chevallereau, A.D. Ames, R.W. Sinnet, *3D Bipedal Robotic Walking: Models, Feedback Control, and Open Problems*, NOLCOS, Bologna, Italy, September 2010.
- [J.W. Grizzle, G. Abba et al., 2001] J.W. Grizzle, G. Abba, F. Plestan, *Asymptotically stable walking for biped robots: analysis via systems with impulse effects*, Vol. 46, No. 1, IEEE Trans. Automat. Control, 2001, pp. 51-64.
- [J. Yamaguchi, A. Takanishi et al., 1995] J. Yamaguchi, A. Takanishi, I. Kato, *Experimental development of a foot mechanism with shock absorbing material for acquisition of landing surface position information and stabilization of dynamic biped walking*, IEEE Int. Conf. Robot. Autom. (1995), pp. 2892–2899.
- [K. Arikawa, T. Mita, 2002] K. Arikawa, T. Mita, *Design of Multi-DOF Jumping Robot*, Proceedings of the 2002 IEEE International Conference on Robotics & Automation, May 2002, pp. 3992-3997.
- [K. Hashimoto, Y. Sugahara et al., 2006] K. Hashimoto, Y. Sugahara, H. Sunazuka, C. Tanaka, A. Ohta, M. Kawase, H. Lim, A. Takanishi, *Biped landing pattern modification method with nonlinear compliance control*, IEEE Int. Conf. Robot. Autom. (Orlando 2006) pp. 1213–1218.
- [K. Hashimoto, Y. Yoshimura et al., 2011] K. Hashimoto, Y. Yoshimura, H. Kondo, H.K. Lim, A. Takanishi, *Realization of Quick Turn of Biped Humanoid Robot by Using Slipping Motion with Both Feet*, in Proc. IEEE Int. Conf. on Robotics and Automation, 2011, pp. 2041–2046.
- [K. Hirai, M. Hirose et al., 1998] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, *The development of honda humanoid robot*, IEEE Int. Conf. Robot. Autom. (1998) pp. 1321–1326.
- [K. Miura, F. Kanehiro et al., 2010] K. Miura, F. Kanehiro, K. Harada, K. Kaneko, K. Yokoi, S. Kajita, *Slip Turn Generation of Humanoid Robots Based on an Analysis of Friction Model*, in Proc. 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR2010), 2010, pp. 761–768.
- [K. Miura, F. Kanehiro et al., 2013] K. Miura, F. Kanehiro, K. Kaneko, S. Kajita, K. Yokoi, *Slip-Turn for Biped Robots*, IEEE Trans on Robots, VOL. 29, NO. 4, Aug 2013, pp. 875-887.

[K. Yokoi, F. Kanehiro et al.,2004] K. Yokoi, F. Kanehiro, K. Kaneko, S. Kajita, K. Fujiwara, H. Hirukawa, *Experimental study of humanoid robot HRP-1S*, Int. J. Robot. Res. 23(4-5), 2004, pp. 351–362 .

[K. Miura, F. Kanehiro et al.,2012] Kanako Miura, Fumio Kanehiro, Kenji Kaneko, Shuuji Kajita, Kazuhito Yokoi *Quick Slip-Turn of HRP-4C on its Toes*, 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 3527-3528.

[K. Harada, Y. EiiChi et al. 2010] K. Harada, Y. Eiichi, Y. Kazuhito, *Motion Planning for Humanoid Robots*, Boook, Springer 2010.

[K. Kaneko, S. Kajita et al., 2002] K. Kaneko, S. Kajita, F. Kanehiro, K. Yokoi, K. Fujiwara, H. Hirukawa, T. Kawasaki, M. Hirata, T. Isozumi, *Design of Advanced Leg Module for Humanoid Robotics Project of METI*, ICRA 2002, pp 38-45

[K. Fujiwara, F. Kanahiro et al., 2003] K. Fujiwara, F. Kanehiro, S. Kajita, *The First Human-size Humanoid that can Fall Over Safely and Stand-up Again*, Proceedings of the IEEE/RSJ, Intl. Conference on Intelligent Robots and Systems, October 2003, pp 1920 -1926.

[K. Nishiwakiz S. Kagami et al., 2002] K. Nishiwakiz, S. Kagami, Y. Kuniyoshiz, Masayuki, I. Hirochika, *Toe Joints that Enhance Bipedal and Fullbody Motion of Humanoid Robots*, IEEE International Conference on Robotics & Automation, 2002, pp. 3105-3110.

[L. Saab, O.E Ramos et al., 2013] L. Saab, O.E. Ramos, F. Keith, N. Mansard, P. Soueres, J.Y. Fourquet, *Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints*, IEEE Trans on Robotics, wol. 29, No. 2, 2013, pp. 346-362.

[L. Sebastian, B. Thomas et al., 2009] L. Sebastian, B. Thomas, U. Heinz, *Design and Control of Anthropomorphic Walking Robot LOLA*, Vol.14, No.6, IEEE/ASME Transaction on Mechatronics, Dec2009, pp. 658-666.

[L. Wang, 2009] L. Wang, *Model Predictive Control System Design and Implementation using MATLAB*, Book, Springer 2009.

[L.W. Tsai, A.P. Morgan, 1985] L.W. Tsai, A.P. Morgan: *Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods*, ASME J. Mechan. Transmission Autom. Design 107, 1985, pp. 189-195.

[L.W. Tsai, 1999] L.W. Tsai, *Robot Analysis: The mechanics of serial and parallel manipulators*, book, John Wilry & Sons, USA, 1999.

[M.A. Carla, J.A. Tenreiro Machado et al., 2010] M.A. Carla, J.A. Tenreiro Machado, *Fractional central pattern generators for bipedal locomotion*, Nonlinear Dyn (2010) 62, 2010, pg 27–37.

- [M. Koeda, T. Ito et al., 2009] M. Koeda, T. Ito, T. Yoshikawa, *Shuffle turn with both feet of humanoid robot by controlling load distribution of soles*, in Proc. 12th International Conference on Climbing and Walking Robots (CLAWAR2009), 2009, pp. 1007–1014.
- [M. Raghavan, B. Roth, 1990] M. Raghavan, B. Roth: *Kinematic analysis of the 6R manipulator of general geometry*, 5th Int. Symp. Robot. Res. 1990.
- [M. Raibert, 1986] M. Raibert, *Running with symmetry*, International Journal of Robotics, vol. 5, no.4, pp. 3–19, 1986.
- [M. Vukobratovic, B. Borovac, 2004] M. Vukobratovic, B. Borovac, *ZMP - Thirty five years of its life*. International Journal of Humanoid Robotics, 2004, pp. 157–173.
- [M. Vukobratovi'c, J. Stepanenko, 1972] M. Vukobratovi'c, J. Stepanenko, *On the stability of anthropomorphic systems*, Mathematical Biosciences, vol. 15, 1972, pp. 1–37.
- [M. Xie, G.Q. Zhang et al., 2009] M. Xie, G.Q. Zhang, H. Ying, *Planning and Control of Biped Walking along Curved Paths on Unknown and Uneven Terrain*, ICIRA 2009, Springer-Verlag, 2009, pp. 1032-1043.
- [M. Xie, Z.W. Zhong et al., 2009] M. Xie, Z.W. Zhong, L. Zhang, L. B. Xian, L. Wang, H. J. Yang, C. S. Song, J. Li, *A Deterministic Way of Planning and Controlling Biped Walking of LOCH Humanoid Robot*, International Journal of Industrial Robot 36/4, 2009, pp. 314–325.
- [M. Xie, 2003] M. Xie, *Fundamentals of robotics: linking perception to action*, Book, World Scientific Publishing, 2003.
- [M. Tomizuka, 1987] M. Tomizuka, *Zero Phase Error Tracking Algorithm for Digital Control*, Journal of Dynamic Systems, Measurement and Control, Vo. 109,1987, pp.65-68.
- [Md. Akhtaruzzaman, A.A. Shafie, 2010] Md. Akhtaruzzaman, A.A. Shafie, *Evolution of Humanoid Robot and Contribution of Various Countries in Advancing the Research and Development of the Platform*, International Conference on Control, Automation and Systems 2010, pp.1021-1028.
- [N.T. Phuong, D.W. Kim et al., 2008] N.T. Phuong, D.W. Kim, H.K. Kim, S.B. Kim, *An Optimal Control Method for Biped Robot with Stable Walking Gait*, 2008 8th IEEE-RAS International Conference on Humanoid Robots, 2008, pp. 211-218.
- [Napoleon, N. Shigeki et al., 2002] Napoleon, N. Shigeki, S. Mitsuji, *Balance Control Analysis of Humanoid Robot based on ZMP Feedback Control*, Intl. Conference on Intelligent Robots and Systems, October 2002, pp.2437-2442.
- [P.B. Weiber, 2009] P.B. Weiber, *Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations*, INRIA, Version 1, Jun 2009.

[P.B. Wieber, C. Chevallereau, 2006] P.B. Wieber , C. Chevallereau , *Online adaptation of reference trajectories for the control of walking systems*, Robotics and Autonomous Systems 54(7), 2006, pp. 559-566.

[Q. Li, A. Takanishi et al., 1991] Q. Li, A. Takanishi, I. Kato, *A biped walking robot having a ZMP measurement system using universal force-moment sensors*, IEEE/RSJ Intel. Workshop Intelligent Robots Systems IROS '91, Osaka, Japan, 1991, pp. 1568–1573.

[Q. Yuan, I.M. Chen, 2013] Q. Yuan, I.M. Chen, *3-D Localization of Human Based on an Inertial Capture System*, IEEE Trans. on Robotics, VOL. 29, NO. 3, June 2013, pp. 806-812.

[R. Featherstone, 2008] R. Featherstone, *Rigid Body Dynamics Algorithms*, book, springer 2008.

[R. Kelly, V. Santibáñez et al., 2005] R. Kelly, V. Santibáñez, A. Loría, *Control of Robot Manipulators in Joint Space*, Book, Springer 2005.

[R. Tajima, D. Honda et al., 2009] R. Tajima, D. Honda , Keisuke Suga, *Fast Running Experiments Involving a Humanoid Robot*, 2009 IEEE International Conference on Robotics and Automation, May 12-17, 2009. pp.1571-1576.

[R.N. Jazar, 2010] R.N. Jazar, *Theory of Applied : Robotics, kinematics, Dynamics, and control*. Book Springer 2010.

[S. Kwon, Y.H. Oh et al., 2007] S. Kwon, Y.H. Oh, *Estimation of the Center of Mass of Humanoid Robot*, International Conference on Control, Automation and Systems, 2007, pp. 2705-2709.

[S. Lohmeier, T. Buschmann et al., 2009] S. Lohmeier, T. Buschmann, H. Ulbrich, *Humanoid Robot LOLA*, 2009 IEEE International Conference on Robotics and Automation, Kobe International Conference Center Kobe, Japan, May 12-17, 2009, pp. 775-780.

[S. Kajita, F. Kanehiro et al., 2003] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, *Resolved Momentum Control: Humanoid Motion Planning based on the Linear and Angular Momentum*, Proceedings of the 2003 IEEE/RSJ, Intl. Conference on Intelligent Robots and Systems, October 2003, pp. 1644-1650.

[S. Kajita, T. Nagasaki et al., 2007] S. Kajita, T. Nagasaki, K. Kaneko, H. Hirukawa, *ZMP-Based Biped Running Control*, IEEE Robotics & Automation Magazine, JUNE 2007, pp. 63-72.

[S. Kajita, F. Kanehiro et al., 2003] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, *Biped Walking Pattern Generation using Preview Control of the Zero-Moment-Point*, Vol. 2, IEEE Conference on Robotics and Automation, 2003, pp. 1620-1626.

[S. Kajita, K. Kaneko et al., 2007] S. Kajita, K. Kaneko, M. Morisawa, *ZMP-based Biped Running Enhanced by Toe Springs*, IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007. pp 3963-3969.

[S. Kajita, M.Morisawa et al., 2010] S.Kajita, M.Morisawa, K.Miura, S.Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, K.Yokoi, *Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking*, IEEE/RSJ Conference on Intelligent Robots and Systems 2010, pp. 4489-4496.

[S. Kajita, M.Morisawa et al., 2010] S.Kajita, M.Morisawa, K.Miura, S.Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, K.Yokoi, *Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking*, IEEE/RSJ Conference on Intelligent Robots and Systems 2010, pp. 4489-4496.

[S. Kajita, T. Yamaura et al., 1992] S. Kajita, T. Yamaura, A. Kobayashi, *Dynamic walking control of a biped robot along a potential energy conserving orbit*, IEEE Trans. Robot. Automat., vol. 8, 1992, pp. 431-438.

[S. Shimmyo, T. Sato et al., 2013] S. Shimmyo, T. Sato, K. Ohnishi, *Biped Walking Pattern Generation by Using Preview Control Based on Three-Mass Model*, IEEE Trans. on Industrial Electronics, vol. 60, No. 11, Nov 2013, pp. 5137-5147.

[S. Shimoda, Y. Yoshihara et al., 2013] S. Shimoda, Y. Yoshihara, H. Kimura, *Adaptability of Tacit Learning in Bipedal Locomotion*, IEEE Trans. on Autonomous Mental Development, vol. 5, No. 2, June 2013, pp. 152-161.

[T.B. Sheridan, 1966] T.B. Sheridan, *Three Models of Preview Control*, IEEE transactions on human factors in electronics, 1966, pp. 91-102.

[T.H.S. Li, Y.T. Su et al, 2012]T.H.S. Li, Y.T. Su, S.H. Liu, J.J. Hu, C.C. Chen, *Dynamic Balance Control for Biped Robot Walking Using Sensor Fusion, Kalman Filter, and Fuzzy Logic*, IEEE Trans. on Ind. Electronics, vol. 59, No. 11, Nov 2012, pp. 4394-4408.

[T.C. Wee, A. Astolfi et al., 2011] T. C. Wee, A. Astolfi, M. Xie, *Modular Joint Design of a TPINOKIO Bipedal Robot*, ICINCO 2011. pp.185-190.

[T.C. Wee, A. Astolfi et al., 2011] T.C. Wee, A. Astolfi, M. Xie, *Modular-Joint Design for Efficient Walking and Modeling of a Bipedal Robot TPinokio*, CLAWAR 2011. pp.493-500.

[T.C. Wee, A. Astolfi et al., 2012] T. C. Wee, A. Astolfi, M. Xie , *Control of Bipedal Robot (TPinokio) with Augmented Model Predictive Control and Inverted Pendulum Method*, ICECS 2012, pp.2781-2784.

[T.C Wee, A.Astolfi at al., 2012] T.C. Wee, A. Astolfi, M. Xie, *Robust Control of Bipedal Humanoid (TPinokio)*, Journal Procedia Engineering Vol. 41, Elsevier 2012, pp. 643-649.

[T.C. Wee, A. Astolfi et al., 2013] T.C. Wee, A. Astolfi, M. Xie, *Design and Control of a Bipedal Robot with Sensory feedback*, International Journal of Advanced Robotic Systems, 2013. pp.1-12.

[T. Mita, T. Yamaguchi et al.,1984] T. Mita, T. Yamaguchi, T. Kashiwase, T. Kawase, *Realization of a high speed biped using modern control theory*, Int. J. Contr., vol. 40, no. 1, 1984 pp. 107–119.

[T. Sato, S. Sakaino et al., 2011] T. Sato, S. Sakaino, E. Ohashi, K. Ohnishi, *Walking Trajectory Planning on Stairs Using Virtual Slope for Biped Robots*, IEEE Trans. On Industrial electronics , Vol. 58, No. 4, April 2011, pp. 1385-1396.

[V. Duindam, S. Stramigioli, 2009] V. Duindam, S. Stramigioli, *Modeling and Control for Efficient Bipedal Walking Robots: A Port-Based Approach*, Book, Springer 2009.

[W. Pengfei, H. Bo et al., 2005] W. Pengfei, H. Bo, S. Lining, *Walking Research on Multi-motion Mode Quadruped Bionic Robot Based on Moving ZMP*, Proceedings of the IEEE International Conference on Mechatronics & Automation, 2005, pp. 1935-1940.

[Y. Zheng , J. Shen et al., 1990] Y. Zheng, J. Shen, *Gait synthesis for the SD-2 biped robot to climb sloped surface*, IEEE Trans. Robot. Automat., vol. 6, no. 1, 1990, pp. 86–96.

[Y. Choi, D. Kim et al., 2007] Y. Choi, D. Kim, Y. Oh, B.J. You, *Posture/Walking Control for Humanoid Robot Based on Kinematic Resolution of CoM Jacobian With Embedded Motion*, IEEE Trans on Robotics, VOL. 23, NO. 6, 2007, pp. 1285-1293.

[Y.J. Kim, J.Y. Lee et al., 2012] Y.J. Kim, J.Y. Lee, J.J. Lee, *A Balance Control Strategy of A Walking Biped Robot in An Externally Applied Force*, Proceeding of the IEEE International Conference on Information and Automation, June 2012, pp. 572-577.

[X. Yu, F. Iida, 2014] X. Yu, F. Iida, *Vertical-Hopping Robot Minimalistic Models of an Energy-Efficient*, IEEE Trans. on Industrial Electronics, vol. 61, No. 2, Feb 2014, pp. 1053-1062.

[Z. Qin, F.C. Xiang et al., 2013] Z. Qin, F.C. Xiang, Y. Tao, *Action Generation of a Biped Robot Climbing Stairs*, Proceedings of IEEE International Conf. on Mechatronics and Automation, 2013, pp. 1069-1073.

Appendix A

TPinokio Robot Mechanical System Design

A.1 TPinokio CAD drawing

The mechanical structure of TPinokio is as depicted Figure A.1.1. Additional information can be found at www.tpinokio.com.

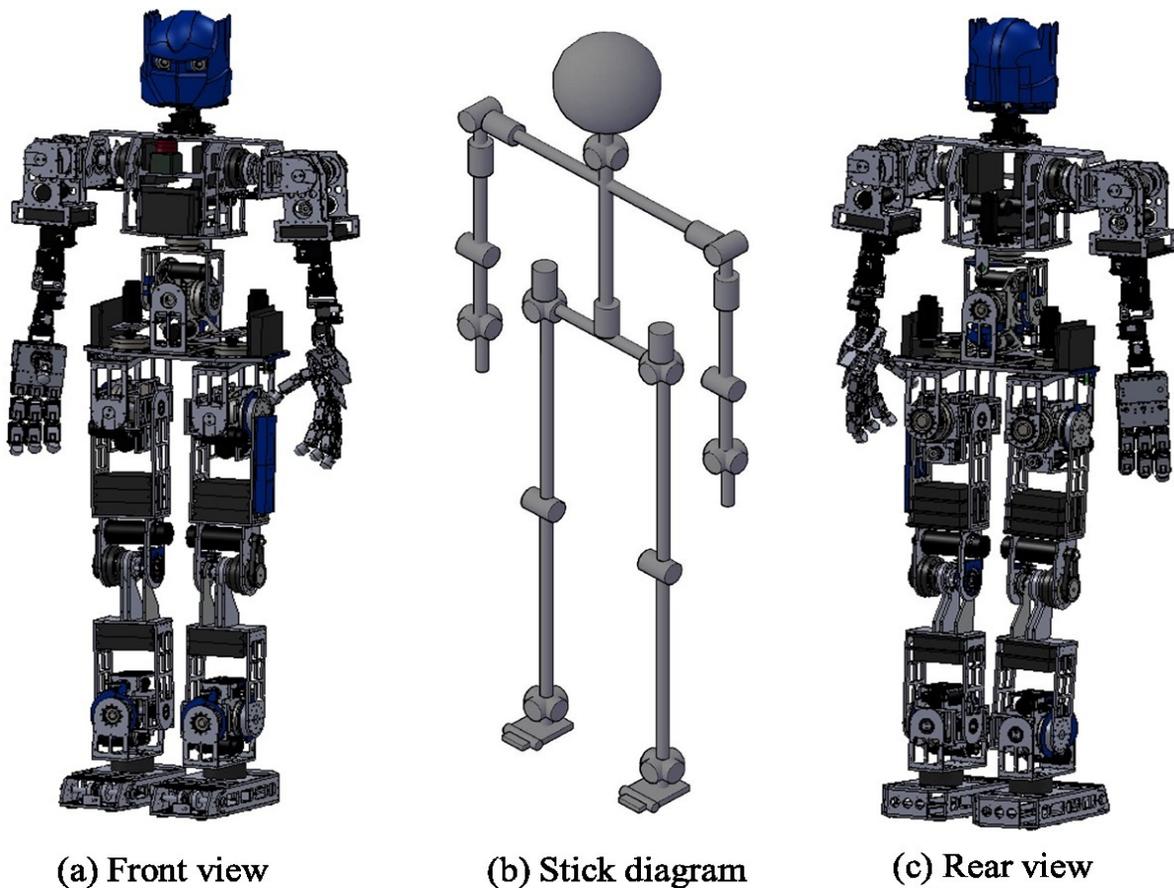
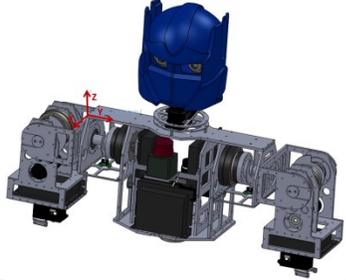
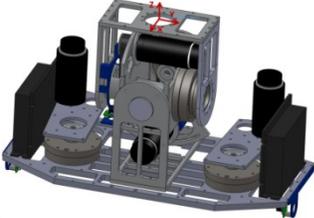
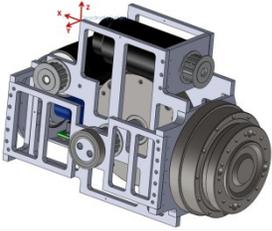
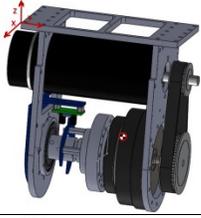
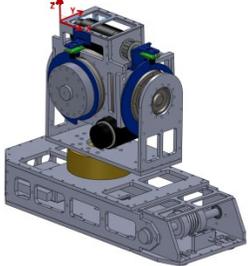


Figure A.1.1. TPinokio robot mechanism and CAD drawing

A.2 Distribution of Mass and CoM Location of the Modular Parts

The mass of the modular parts are listed in Table A.2.1, the values are calculated using SolidWorks software.

Table A.2.1. TPinokio Modular Parts.

Modular Part	CAD Drawing	CoM weight (kg)	CoM location (from reference coordinate)
Upper body assembly		13.13Kg	[-28.55 189.30 -66.24]
Pelvis assembly		6.8kg	[-23.83 3.22 -141.20]
Hip assembly		4.16kg	[-64.04 -39.13 -78.10]
Knee assembly		1.5kg	[-37.01 84.71 -93.36]
Ankle and feet assembly		4.5kg	[63.82 23.30 -111.17]

A.3 TPinokio Robot Power Consumption

Maxon DC motors [<http://www.maxonmotor.com>] are used in the joint actuator design due to its high performance and good power-to-weight ratio. Another reason for choosing DC motor is the ease to control, as the DC motor speed and torque is proportional to the input voltage and current respectively. Maxon motors are made with a wound field stator and a permanent magnet rotor, which has an efficiency of more than 90% . The torque and power requirements of the robot are summarized in Table A.3.1, The motors specifications are listed in Table A.3.2, and the actuators specification are listed in Table A.3.3.

Table A.3.1. TPinokio robot torque and power requirement.

Body segment	Joint	Movement	Max Torque (Nm)	Power (w)
Head	Neck	Pitch	2.217	6.96
		Yaw	2.217	6.96
Chest	Waist	Pitch	3.56	8.9
		Roll	24.69	2.52
		Yaw	3.56	5.48
Arm	Shoulder	Pitch	11.64	17.46
		Roll	11.64	17.46
		Yaw	11.64	17.46
	Elbow	Pitch	4.99	7.49
	Wrist	Pitch	2.205	3.3
		Yaw	2.205	3.3
Hand	Fingers	Pitch	0.98	1.54
Leg	Hip	Pitch	13.87	108.87
		Roll	31.78	13.03
		Yaw	15.68	24.62
	Knee	Pitch	23.63	164.96
	Ankle	Pitch	54.53	107.04
		Roll	61.17	25.07

Table A.3.2. Robot motors specifications.

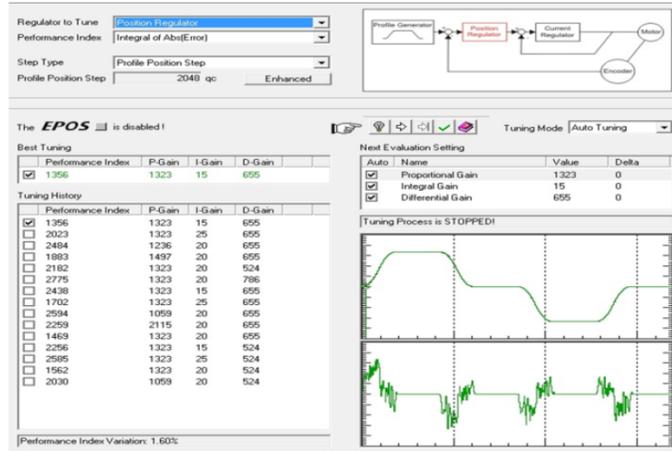
Motor data	Maxon RE40, 150W	Maxon RE35, 90W	
Nominal voltage	24v	24v	
Nominal speed	6930 rpm	6910	
Nominal torque	170 mNm	93.3 mNm	
Stall torque	2280 mNm	1160 mNm	
Max. permissible speed	12000 rpm	12000 rpm	

Table A.3.3. The robot actuator specifications and maximum operation range.

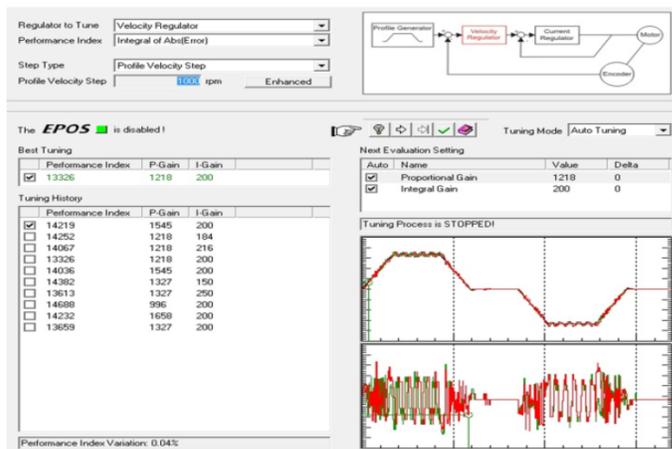
Joint	motion	Harmonic gear size /ratio/ max torque	Pulley & belt ratio	Motor power & max. speed	Angle Range (similar to Human)
Neck	Pitch	RX64, 200:1, 64kgf.cm	-	21w	-30° to 90°
	Yaw	RX64, 200:1, 64kgf.cm	-	21w	-80° to 80°
Shoulder	Yaw	Size 14 / 100 :1 54Nm	2:1	90w, 360°/s	-90° to 90°
	Pitch	Size 14 / 100 :1	2:1	90w, 360°/s	-180° to 120°
Elbow	Pitch	EX-106 ,164:1,10.4Nm	1:164	59w	0° to 130°
Waist	Roll	Size 14, 100:1, 54Nm	1:1	90w, 720°/s	-50° to 50°
	Pitch	Size 14, 100:1, 54Nm	1:1	90w, 720°/s	-20° to 70°
	Yaw	Size 14, 100:1, 54Nm	2:1	90w, 360°/s	-70° to 70°
Hip	Roll	Size 20, 120:1, 147Nm	2:1	150w, 300°/s	-60° to 45°
	Pitch	Size 20, 160:1, 147Nm	2:1	150w, 225°/s	-130° to 45°
	Yaw	Size 14, 100:1, 54Nm	2:1	90w, 360°/s	-90° to 60°
Knee	Pitch	Size 17, 120:1, 110Nm	2:1	200w, 300°/s	0° to 150°
Ankle	Roll	Size 14, 100:1, 54Nm	2:1	150w, 300°/s	-20° to 30°
	Pitch	Size 20, 160:1, 147Nm	2:1	150w, 225°/s	-30° to 80°

A.4 Motor Tuning

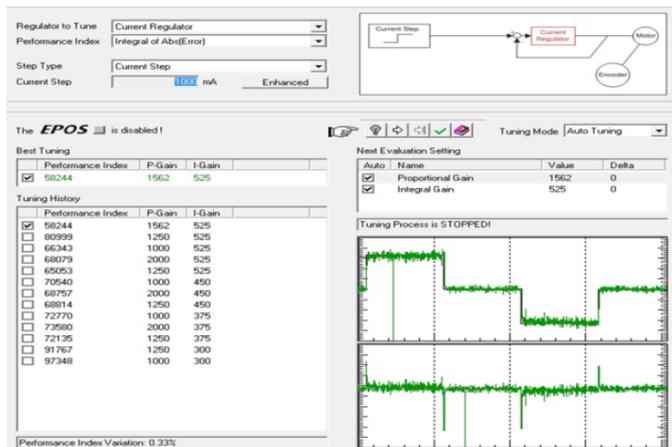
The optimal PID parameters of the motors are tuned with *EPOS studio*, the sample plots is as shown in Figure A.4.1.



(a) Position Tuning



(a) Velocity Tuning



(a) Current Tuning

Figure A.4.1. Motor PID parameters tuning.

A.5 Gear and Belt Selection

Selecting the right gears for the joints is not a trivial task. Suitable gears should be light weighted, high efficiency, low backlash and have good torque-to-volume ratio. Harmonic Drive gears (HDG) are used in the joints design due to its compact casing, high torque and its zero-backlash. Furthermore, the axes of the motors and gears can be positioned parallel coupled by pulley and belt instead of on the same axis.

A.5.1 Harmonic Drive Gear Design

The harmonic drive gear [http://www.harmonicdrive.net] as depicted in Figure A.5.1(a) is an assembly of three main parts; an outer ring named circular spline, a metal cup named flexspline and an elliptic steel disk named the wave generator. The basic operation is shown in Figure A.5.1(b). The wave generator is used as the input and is mounted inside the flexspline, which is used as the output. The wave generator is mounted in the flexspline. The flexspline has teeth machined into the outer surface of the steel cup, and the circular spline has teeth machined on the inner surface of the ring. The flexspline has two teeth less than the circular spline.

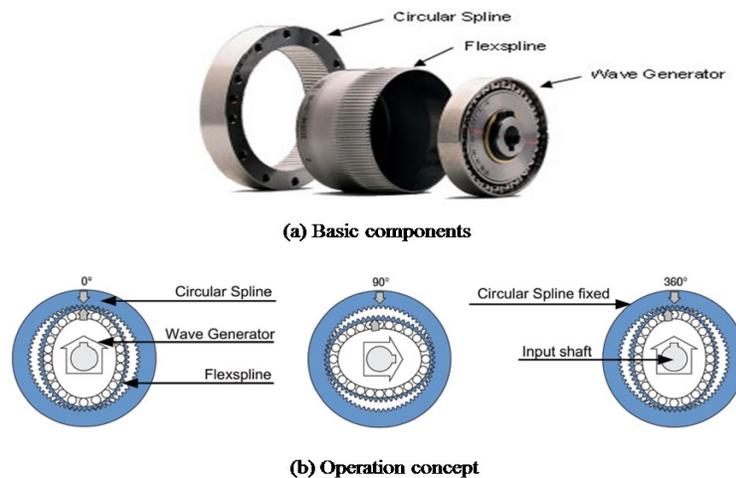


Figure A.5.1. Harmonic gear, (a) Basic components, (b) Operation concept.

In order to reduce the number of design parts, SHF units were selected for this project. This is due to its compactness, light weight and has an input shaft, which makes implementation of a

belt drive easy and it can be mounted directly on the joint assembly. The specifications are listed in Table A.5.1.

Table A.5.1. SHF series harmonic gear specification.

SHF-2UJ	Mass [kg]	Nominal torque [Nm]	Max repeated torque [Nm]	Max momentary torque[Nm]	Max input speed [rpm]
14	0.64	7.8	28	54	8500
17	0.95	24	54	110	7300
20	1.40	40	82	147	6500

A.5.2 Synchroflex Timing Belts

CONTI SYNCHROFLEX® timing belts [<http://www.contitech.de/>] is used on TPinokio gear system as shown in Figure A.5.2. It is able to deliver high outputs due to its high-grade components.

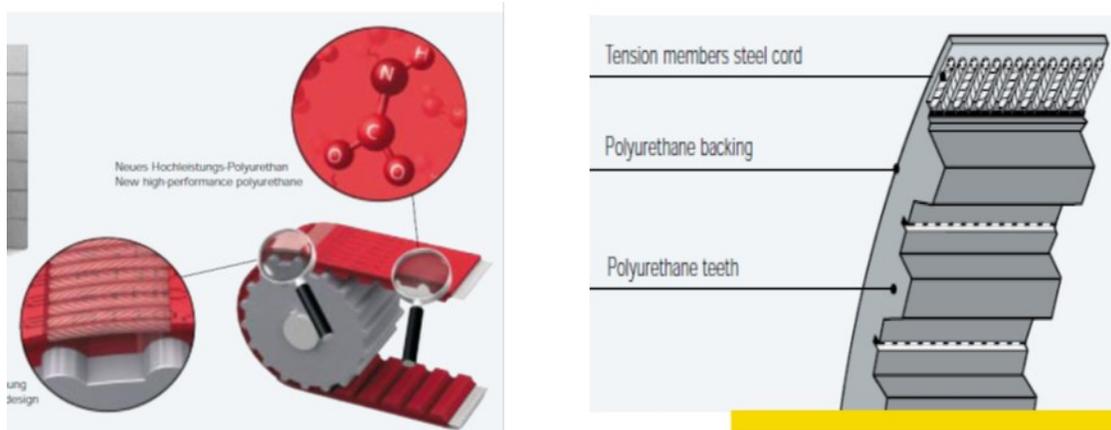


Figure A.5.2. Synchroflex timing belts.

The belt is made from hard-wearing polyurethane and constant-length galvanized steel and can produce high power output. The manufacturing process combines the advantages of high pitch and angular accuracy and is particularly suitable for high rotational speeds. The rotational speed of the belt can reach a range of up to 20,000rpm.

A.6 Kinematic (D-H parameters) model for TPinokio Robot.

The kinematic coordinate and link frames are as depicted in Figure A.6.1. The D-H link parameters are listed in Table A.6.1 to A.6.5.

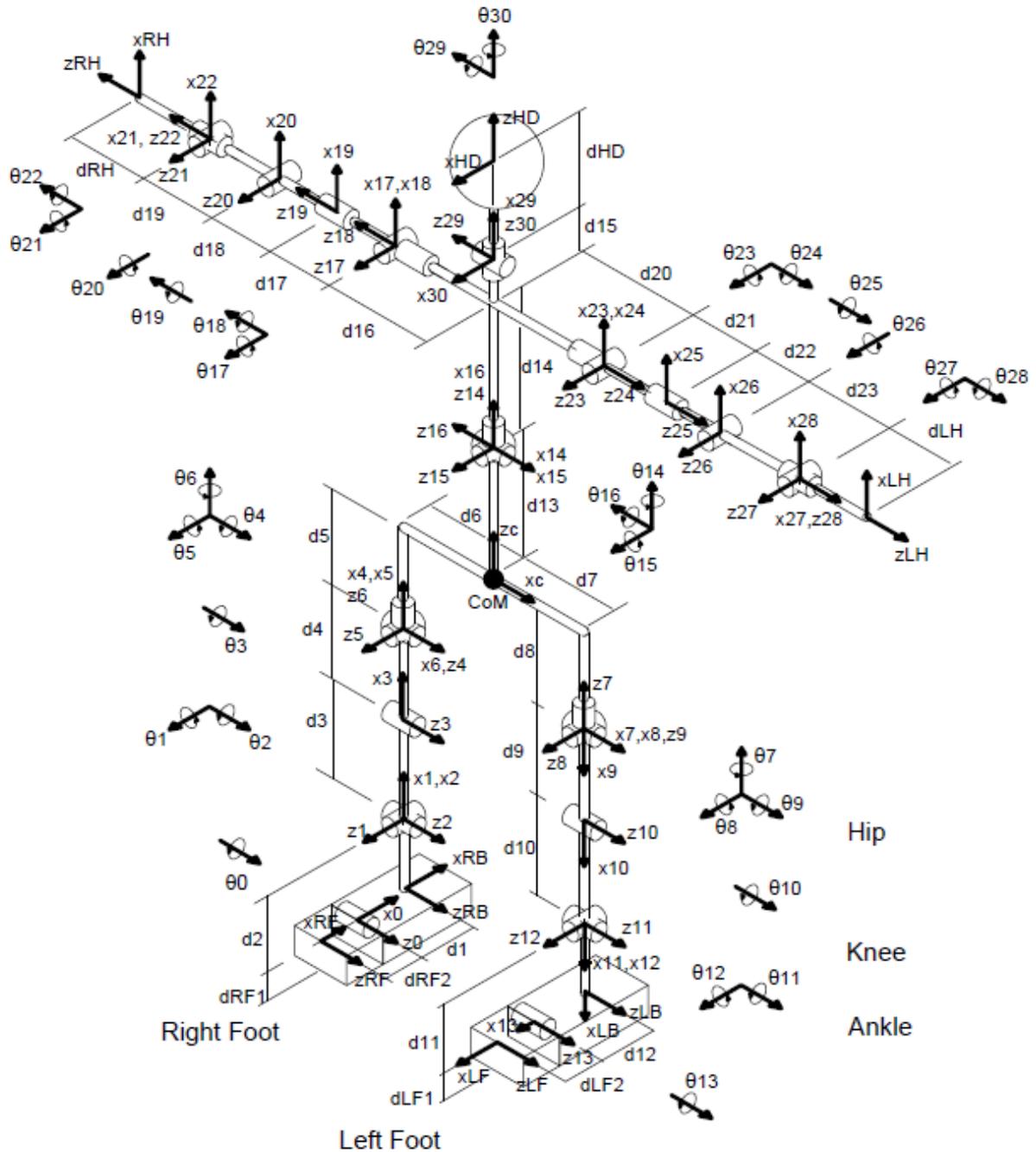


Figure A.6.1. TPinokio D-H model and link frames.

Table A.6.1. D-H parameters for Right leg toe to CoM.

Frame no. Joint axis j	Link Length a_j	Link twist α_j	Joint Distance d_j	Joint Angle θ_j
RF	*	*	*	*
0	d_{RF2}	0	0	0
RB	d_1	0	0	θ_0
1	d_2	-90	0	-90
2	0	90	0	θ_1
3	d_3	0	0	θ_2
4	d_4	0	0	θ_3
5	0	-90	0	θ_4
6	0	-90	0	$-90+\theta_5$
CoM	d_6	0	d_5	θ_6

Table A.6.2. D-H parameters for CoM to Left leg toe.

Joint axis j	a_j	α_j	d_j	θ_j
CoM	*	*	*	*
7	d_7	0	d_8	0
8	0	90	0	θ_7
9	0	-90	0	$-90+\theta_8$
10	d_9	0	0	θ_9
11	d_{10}	0	0	θ_{10}
12	0	90	0	θ_{11}
LB	d_{11}	-90	0	θ_{12}
13	d_{12}	0	0	-90
LLT	d_{LF2}	0	0	θ_{13}

Table A.6.3. D-H parameters for CoM to Right Hand finger tip.

Joint axis j	a_j	α_j	d_j	θ_j
CoM	*	*	*	*
14	0	0	d_{13}	0
15	0	90	0	θ_{14}
16	0	-90	0	$90+\theta_{15}$
17	d_{14}	90	d_{16}	θ_{16}
18	0	-90	0	θ_{17}
19	0	0	d_{17}	θ_{18}
20	0	90	d_{18}	θ_{19}
21	d_{19}	0	0	$90+\theta_{20}$
22	0	-90	0	$-90 + \theta_{21}$
RHT	0	0	d_{RH}	θ_{22}

Table A.6.4. D-H parameters for CoM to Left Hand finger tip.

Joint axis j	a_j	α_j	d_j	θ_j
CoM	*	*	*	*
14	0	0	d_{13}	0
15	0	90	0	θ_{14}
16	0	-90	0	$90+\theta_{15}$
23	d_{14}	90	d_{20}	θ_{16}
24	0	90	0	θ_{23}
25	0	0	d_{21}	θ_{24}
26	0	-90	d_{22}	θ_{25}
27	d_{23}	0	0	$-90+\theta_{26}$
28	0	90	0	$90+\theta_{27}$
LHT	0	0	d_{LH}	θ_{28}

Table A.6.5. D-H parameters for CoM to Head.

Joint axis j	a_j	α_j	d_j	θ_j
CoM	*	*	*	*
14	0	0	d_{13}	0
15	0	90	0	θ_{14}
16	0	-90	0	$90+\theta_{15}$
29	$d_{14}+d_{15}$	0	0	θ_{16}
30	0	-90	0	$-90+\theta_{29}$
HD	0	0	d_{HD}	θ_{30}

A.7 Inverse Kinematic

Referring to Figure A.7.1, the right leg close-form based on geometrical method and the set of equations are presented in this section without detailed explanations.

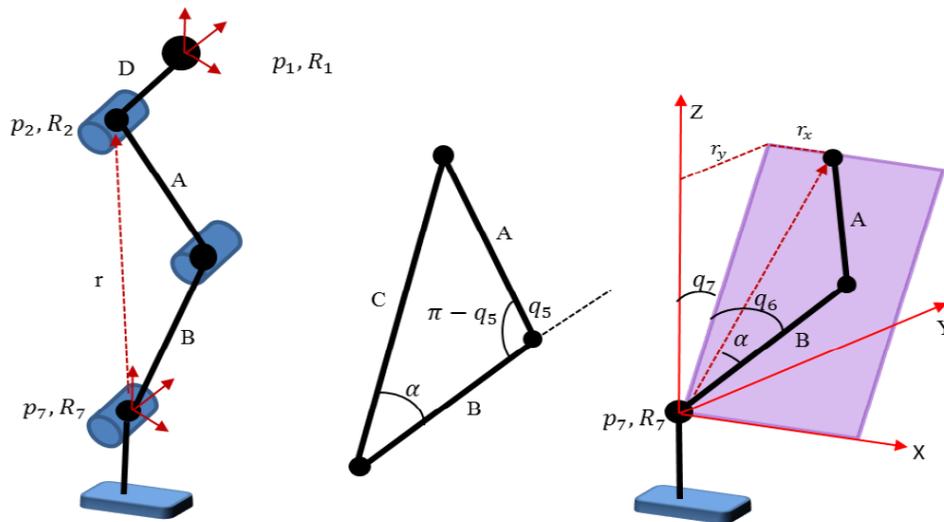


Figure A.7.1. TPinkio geometrical parameters.

The Cartesian coordinate of the right hip with reference to and expressed in the pelvis (p_1) coordinates is

$$p_2 = p_1 + R_1 .$$

The parameters between the ankle and hip (p_2, p_7) can be calculated as follows:

$$r = R_7^T(p_2 - p_7) \equiv [r_x \quad r_y \quad r_z]^T ,$$

$$C = \sqrt{r_x^2 + r_y^2 + r_z^2} ,$$

$$C^2 = A^2 + B^2 - 2AB \cos(\pi - q_5) .$$

A solution for the knee pitch angle q_5 is found immediately by solving the above equations.

$$q_5 = -\cos^{-1}\left(\frac{A^2 + B^2 - C^2}{2AB}\right) + \pi .$$

Applying the trigonometric relationship, the hip and ankle angle can be calculated as follows:

$$\frac{C}{\sin(\pi - q_5)} = \frac{A}{\sin \alpha} ,$$

$$\alpha = \sin^{-1}\left(\frac{A \sin(\pi - q_5)}{C}\right) ,$$

$$\text{ankle roll: } q_7 = \text{atan2}(r_y, r_z) ,$$

$$\text{ankle pitch: } q_6 = -\text{atan2}\left(r_x, \text{sign}(r_z) \sqrt{r_y^2 + r_z^2}\right) - \alpha ,$$

$$\begin{bmatrix} c_2 c_4 - c_2 s_3 s_4 & -s_2 c_3 & c_2 s_4 + s_2 s_3 c_4 \\ s_2 c_4 + c_2 s_3 s_4 & c_2 c_3 & s_2 s_4 - c_2 s_3 c_4 \\ -c_3 s_4 & s_3 & c_3 s_4 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

$$c_* \equiv \cos q_* \quad ; \quad s_* \equiv \sin q_*$$

$$\text{hip yaw: } q_2 = \text{atan2}(-R_{12}, R_{22}) ,$$

$$\text{hip roll: } q_3 = \text{atan2}(R_{32}, -R_{12}s_2 + R_{22}c_2) ,$$

$$\text{hip pitch: } q_4 = \text{atan2}(-R_{31}, R_{33}) .$$

(A.7.1)

Appendix B

Theory of Applied Robotics

B.1 Denavit-Hartenberg (D-H) Notation

A DH coordinate frame [Reza N. Jazar, 2010], as depicted in Figure B.1, is identified by four parameters:

- The variable a_i is the link length distance between z_{i-1} and z_i -axis along the x_i axis.
- The variable α_i is the link twist rotation of z_{i-1} about the x_i -axis to become parallel to the z_i -axis.
- The variable d_i is the joint distance between x_{i-1} and x_i -axis along the z_{i-1} -axis.
- The variable θ_i is the joint angle rotation of x_{i-1} -axis about the z_{i-1} axis to become parallel to the x_i -axis.

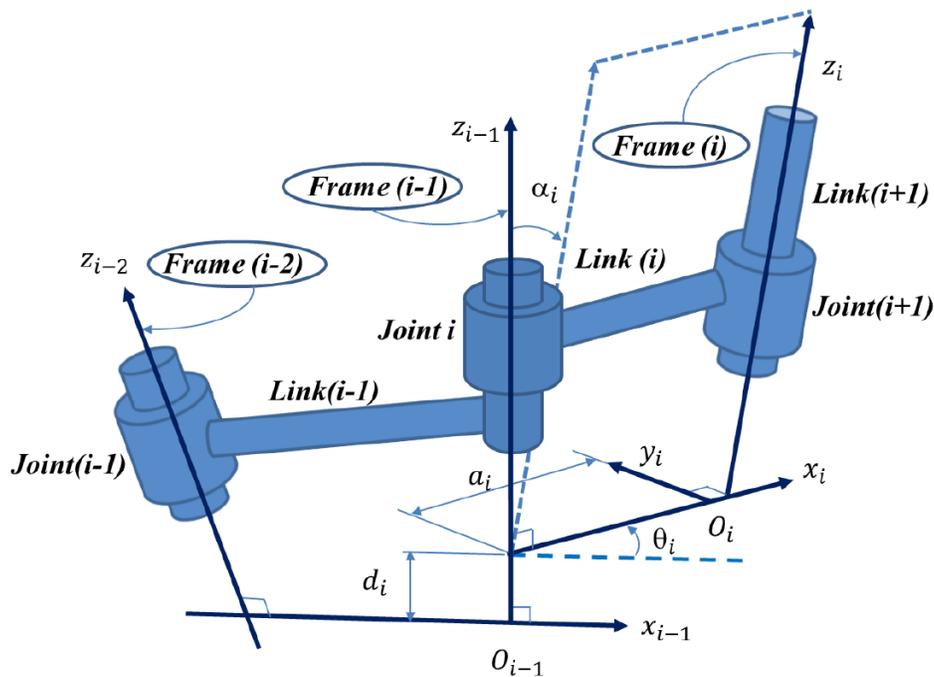


Figure B.1. Definition of D-H parameters.

A D-H parameter table as shown in Table B.1, helps to establish a systematic way to determine the parameters.

Table B.1 D-H parameter table of the link.

Frame No.	a_i	α_i	d_i	θ_i
1	a_1	α_1	d_1	θ_1
2	a_2	α_2	d_2	θ_2
...
j	a_j	α_j	d_j	θ_j
...
n	a_n	α_n	d_n	θ_n

B.2 Transformation Between Frames

B.2.1 Frame Coordinate

By using the Denavit-Hartenberg convention, to transform the coordinate of frames B_i to a new frame B_{i-1} , the transformation matrix ${}^{i-1}T_i$ can be expressed as follows :

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.2.1})$$

Where

$$R_{x_{i-1},\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_{x_{i-1},a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_{z_{i-1},\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_{z_{i-1},d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The transformation equation between $B_i (x_i, y_i, z_i)$ and $B_{i-1} (x_{i-1}, y_{i-1}, z_{i-1})$ can be written as

$$\begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \\ 1 \end{bmatrix} = {}^{i-1}T_i \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \quad (\text{B.2.2})$$

where

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.2.3})$$

This 4×4 matrix (B.2.3) can be further partitioned into rotational and translational parameters

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}d_i \\ 0 & 1 \end{bmatrix}, \quad (\text{B.2.4})$$

where

$${}^{i-1}R_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix}, \quad (\text{B.2.5})$$

and

$${}^{i-1}d_i = \begin{bmatrix} a_i \cos\theta_i \\ a_i \sin\theta_i \\ d_i \end{bmatrix}. \quad (\text{B.2.6})$$

Similarly, the inverse of the transformation matrix can be expressed as

$${}^i T_{i-1} = {}^{i-1}T_i^{-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -a_i \\ -\sin\theta_i \cos\alpha_i & \cos\theta_i \cos\alpha_i & \sin\alpha_i & -d_i \sin\alpha_i \\ \sin\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & \cos\alpha_i & -d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.2.7})$$

B.2.2 Rotational Matrixes

The rotational matrixes for the three x-y-z axes are

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.2.8})$$

B.3 Iterative Technique

Applying the inverse kinematics technique to search for the solution q_k from a set of nonlinear equation can be expressed as

$${}^0T_n = T(q) = {}^0T_{1(q_1)} {}^1T_{2(q_2)} {}^2T_{3(q_3)} \dots {}^{n-1}T_{n(q_n)} \dots$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.3.1})$$

$$\text{for } r_{ij} = r_{ij}(q_k), \quad k = 1, 2, \dots, n.$$

where n denote the DoF.

Equation (B.3.1) is used to determine the joint variable (q_k) by applying the *Newton-Raphson* iterative method to solve the following kinematic equation by setting

$$T(q) = 0. \quad (\text{B.3.2})$$

We begin with a calculated guess for the joint variables ,

$$q^* = q + \delta q. \quad (\text{B.3.3})$$

Applying the forward kinematics notation, the orientation parameters of the end-effector with some 'guessed' q variables is determined

$$T^* = T(q^*). \quad (\text{B.3.4})$$

Therefore, the 'guessed' error, must now be minimized with

$$\delta T = T - T^*. \quad (\text{B.3.5})$$

The Taylor expansion of equations (B.3.3) is

$$T = T(q^* + \delta q) = T(q^*) + \frac{\partial T}{\partial q} \delta q + O(\delta q^2). \quad (\text{B.3.6})$$

Assuming $\ll I$, we can write

$$\delta T = J \delta q, \quad (\text{B.3.7})$$

where J is the Jacobian matrix, can be repressed as

$$J(q) = \left[\frac{\partial T_i}{\partial q_j} \right], \quad (\text{B.3.8})$$

That implies

$$\delta q = J^{-1} \delta T. \quad (\text{B.3.9})$$

Therefore, the unknown variables q are

$$q = q^* + J^{-1} \delta T. \quad (\text{B.3.10})$$

The values obtained by (B.3.10) are then used as a new input value to repeat the iterative process for closer values and finally converge to the actual variables

$$q^{(i+1)} = q^{(i)} + J^{-1}(q^{(i)})\delta T(q^{(i)}) . \quad (\text{B.3.11})$$

The iteration method is illustrated in Figure B.3.

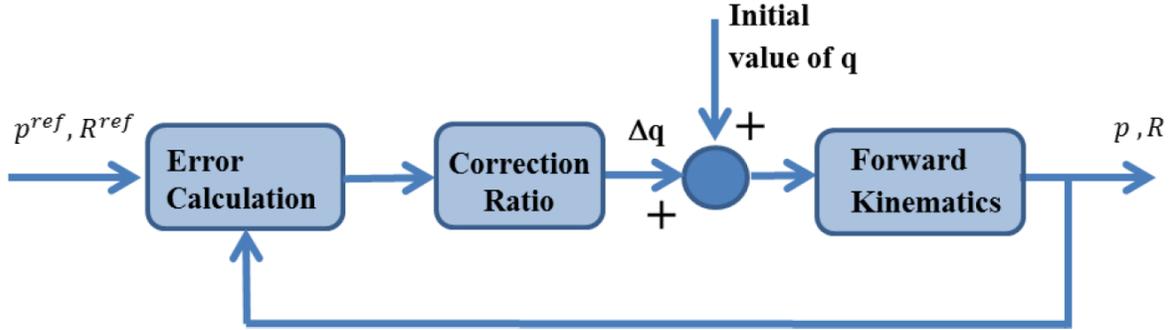


Figure B.3. Iteration block diagram.

Work example :Inverse kinematics for a two link planar manipulator.

The end-effector position of a two link planar manipulator can be described by

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} l_1 c\theta_1 + l_2 c(\theta_1 + \theta_2) \\ l_1 s\theta_1 + l_2 s(\theta_1 + \theta_2) \end{bmatrix} . \quad (\text{B.3.12})$$

We define $q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ and $\begin{bmatrix} X \\ Y \end{bmatrix}$, the Jacobian matrix is

$$J(q) = \begin{bmatrix} \frac{\partial T_i}{\partial q_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial X}{\partial \theta_1} & \frac{\partial X}{\partial \theta_2} \\ \frac{\partial Y}{\partial \theta_1} & \frac{\partial Y}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_1 s\theta_1 + l_2 s(\theta_1 + \theta_2) & -l_2 s(\theta_1 + \theta_2) \\ l_1 c\theta_1 + l_2 c(\theta_1 + \theta_2) & l_2 c(\theta_1 + \theta_2) \end{bmatrix} . \quad (\text{B.3.14})$$

The inverse is

$$J^{-1} = \frac{-1}{l_1 l_2 s\theta_2} \begin{bmatrix} -l_2 c(\theta_1 + \theta_2) & -l_2 s(\theta_1 + \theta_2) \\ l_1 c\theta_1 + l_2 c(\theta_1 + \theta_2) & l_1 s\theta_1 + l_2 s(\theta_1 + \theta_2) \end{bmatrix} , \quad (\text{B.3.15})$$

and the iteration is set up as

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^{(i+1)} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^{(i)} + J^{-1} \left(\begin{bmatrix} X \\ Y \end{bmatrix} - \begin{bmatrix} X \\ Y \end{bmatrix}^{(i)} \right) . \quad (\text{B.3.16})$$

Let assume the parameters $l_1 = l_2 = 1$, and end-effector position $T = [X \ Y]^T = [1 \ 1]^T$,

and from an initial guess position

$$q^{(0)} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^{(0)} = \begin{bmatrix} \pi/3 \\ -\pi/3 \end{bmatrix}, \quad (\text{B.3.17})$$

and

$$\delta T = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} c(\pi/3) + c(\pi/3 - \pi/3) \\ s(\pi/3) + s(\pi/3 - \pi/3) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 3/2 \\ \sqrt{3}/2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} + 1 \end{bmatrix}. \quad (\text{B.3.18})$$

The Jacobian and its inverse matrix are

$$J = \begin{bmatrix} -\frac{\sqrt{3}}{2} & 0 \\ \frac{3}{2} & 1 \end{bmatrix}, \quad J^{-1} = \begin{bmatrix} -\frac{2\sqrt{3}}{3} & 0 \\ \sqrt{3} & 1 \end{bmatrix}, \quad (\text{B.3.19})$$

and therefore,

$$q^{(1)} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^{(1)} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^{(0)} + J^{-1}(\delta T) = \begin{bmatrix} \frac{\pi}{3} \\ -\frac{\pi}{3} \end{bmatrix} + \begin{bmatrix} -\frac{2\sqrt{3}}{3} & 0 \\ \sqrt{3} & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} + 1 \end{bmatrix} = \begin{bmatrix} 1.62 \\ -1.78 \end{bmatrix}. \quad (\text{B.3.10})$$

Base on the iterative technique, the solution can be found in a few iterations as shown:

Iteration 2	$J = \begin{bmatrix} -0.844 & 0.154 \\ 0.934 & 0.988 \end{bmatrix}, \delta T = \begin{bmatrix} 0.065 \\ 0.155 \end{bmatrix}, q^{(2)} = \begin{bmatrix} 1.583 \\ -1.582 \end{bmatrix}$
Iteration 3	$J = \begin{bmatrix} -1 & 0.000433 \\ 0.988 & 0.999 \end{bmatrix}, \delta T = \begin{bmatrix} 0.0119 \\ -0.000362 \end{bmatrix}, q^{(3)} = \begin{bmatrix} 1.5707958 \\ -1.5708670 \end{bmatrix}$
Iteration 4	$J = \begin{bmatrix} -1 & 0.0 \\ 0.998 & 1.0 \end{bmatrix}, \delta T = \begin{bmatrix} -0.438 \times 10^{-6} \\ 0.711 \times 10^{-4} \end{bmatrix}, q^{(4)} = \begin{bmatrix} 1.570796329 \\ -1.570796329 \end{bmatrix}$

The result of $q^{(4)}$ is very close to the required value of $q = \begin{bmatrix} \pi/2 \\ -\pi/2 \end{bmatrix}$.

B.4 Singularities

Given a linear transformation relating a joint velocity $\dot{\theta}$ to Cartesian velocity v , if the matrix is nonsingular, we can express the relationship as

$$\dot{\theta} = J(\theta)^{-1}v. \quad (\text{B.4.1})$$

An in-depth study of the classification of singularities is beyond the scope of this thesis, for more information; please refer to [L.W. Tsai, 1999]. Singularities can be broadly classified into two categories:

- Workspace-boundary: this situation occurs when the manipulator is fully folded or extended in a manner that the end-effector is at the boundary limit of the manipulator.
- Workspace-interior: this situation is caused by a lining up of two or more joint axes.

A manipulator will lost one or more DoF if it is in singular configuration [J. Craig, 2005]. This means that there is some position/direction which it is impossible to reach, no matter what joint speed are selected.

B.5 Three-D Linear Inverted Pendulum Model

B.5.1 Dynamic of a 3D Inverted Pendulum

During single support phase (SSP), the dynamic of a bipedal can be represented by a single-mass inverted pendulum as depicted in Figure B.5 [S. Kajita, F. Kahehiro et al., 2003], the location of a point mass $p = (x, y, z)$ can be represented with state variables $q = (\theta_r, \theta_p, r)$ according to the right-hand rule (i.e. $\theta_r < 0$ and $\theta_p > 0$), and r is the length of the pendulum leg.

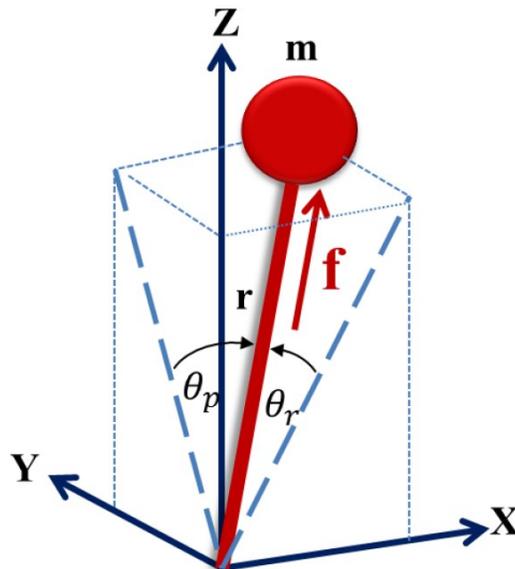


Figure B.5. Single Mass Inverted Pendulum Model.

Let (τ_r, τ_p) be the actuator torque, 3D inverted pendulum dynamic motion equations can be derived as

$$m(-z\ddot{y} + y\ddot{z}) = \frac{D}{C_r} \tau_r - mgy, \quad (\text{B.5.1})$$

$$m(z\ddot{x} - x\ddot{z}) = \frac{D}{C_p} \tau_p + mgx, \quad (\text{B.5.2})$$

where $C_r \equiv \cos \theta_r$, $C_p \equiv \cos \theta_p$, $D \equiv \sqrt{C_r^2 + C_p^2 - 1}$, g is the gravity acceleration and m is the object total mass. Note that these equations are non-linear.

B.5.2 3D Linear Inverted Pendulum Mode

To linearized the model, constraint is applied to limit the motion of the single mass in the z-plane with vector parameters $(k_x, k_y, -1)$ and intersect the z-axis at point at z_c with

$$z = k_x x + k_y y + z_c. \quad (\text{B.5.3})$$

The second derivative of equation (B.5.3) is

$$\ddot{z} = k_x \ddot{x} + k_y \ddot{y}, \quad (\text{B.5.4})$$

and the force f acting on the mass m can be separated into f_x , f_y , f_z Cartesian coordinates

$$f_x = \frac{x}{r} f, \quad f_y = \frac{y}{r} f, \quad f_z = \frac{z}{r} f. \quad (\text{B.5.5})$$

Base on Newton law, the forces equations for the three planes in equation can be written as

$$m\ddot{x} = \frac{x}{r} f, \quad m\ddot{y} = \frac{y}{r} f, \quad m\ddot{z} = \frac{z}{r} f - mg_z. \quad (\text{B.5.6})$$

Substitute the constraints (B.5.5) and (B.5.6) into (B.5.4), the matrix form is obtained

$$\begin{bmatrix} \frac{x}{r} f & \frac{y}{r} f & \frac{z}{r} f - mg_z \end{bmatrix} \begin{bmatrix} -k_x \\ -k_y \\ 1 \end{bmatrix} = 0. \quad (\text{B.5.7})$$

By setting the constraints to $(k_x = 0, k_y = 0)$ for walking on horizontal flat surface, from (B.5.7), the force f is determined

$$f = \frac{mg_z r}{z_c}. \quad (\text{B.5.8})$$

The natural trajectory of a 3D-LIPM can be determined by substituting (B.5.8) into (B.5.6):

$$\ddot{x} = \frac{g}{z_c} x, \quad (\text{B.5.9})$$

$$\ddot{y} = \frac{g}{z_c} y. \quad (\text{B.5.10})$$

Note that the above equations are now decoupled independent linear equations as compared to equations (B.5.1) and (B.5.2).

B.6 Moment of Inertial

Applying Newton's Second Law, the sum of the forces acting on a mass in linear motion equals the mass times the acceleration of the mass, is calculated as

$$\sum F = ma . \quad (\text{B.6.1})$$

For a rotating mass, Newton's Second Law is defined as

$$\sum M = I\ddot{\theta} , \quad (\text{B.6.2})$$

where $\sum M$ is the sum of the moments about the rotating axis, I is the moment of inertia of the mass about that axis, and $\ddot{\theta}$ is the mass angular acceleration. The moment of inertia about the axis can be written as

$$I = \int mr^2 dV, \quad (\text{B.6.3})$$

where r is the radial distance from the axis.

B.7 Zero-Moment Point (ZMP)

The Zero-Moment Point (ZMP) may also be regarded as the Center-of-Pressure of the reacting forces due to the robot body mass from the ground contact surface as depicted in Figure B.7.

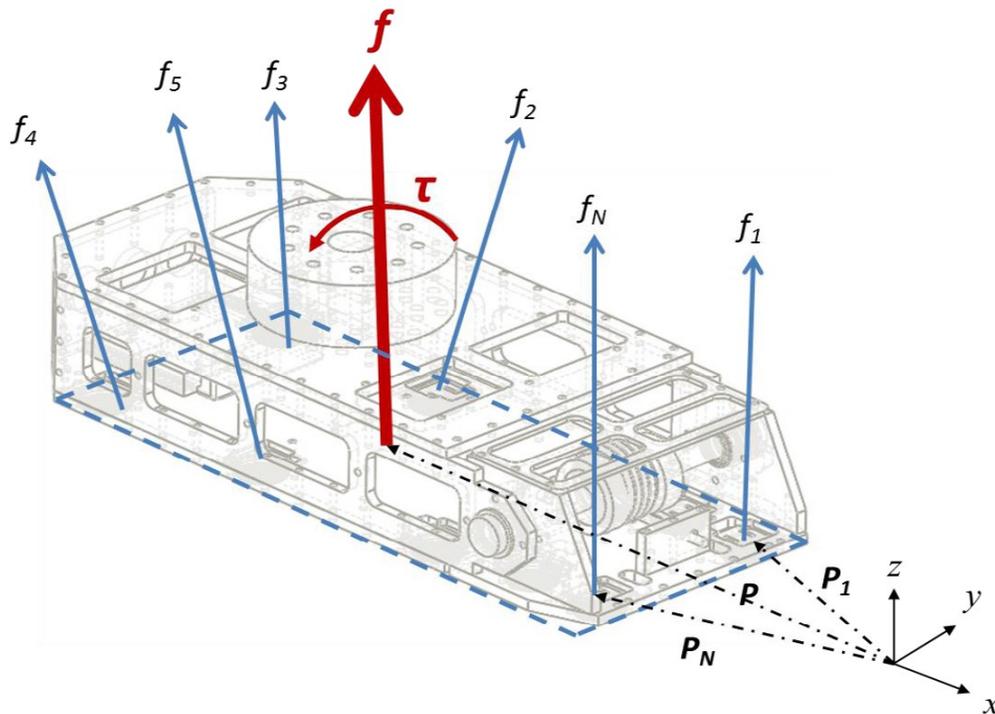


Figure B.7. ZMP and floor reaction forces.

The floor reaction forces f_i are formed up by a subset of points p_i and can be written in vector form:

$$f_i = [f_{ix} \quad f_{iy} \quad f_{iz}]^T, \quad \text{for } (i = 1, 2, \dots, N). \quad (\text{B.7.1})$$

Where f_{ix} , f_{iy} and f_{iz} are the forces in the x , y , z direction respectively. Therefore ZMP can be calculate as

$$ZMP = \frac{\sum_{i=1}^N p_i f_{iz}}{\sum_{i=1}^N f_{iz}}. \quad (\text{B.7.2})$$

The points p_i form the support convex polygon (SCP) foot-print. This can be concluded that the ZMP always exists in the SCP and the ZMP is undefined if the CoP is not inside the SCP.

The torque around the ZMP can be calculated as

$$\tau = \sum_{i=1}^N (p_i - p) \times f_i. \quad (\text{B.7.3})$$

The above equation can be represented in vector form for as follows:

$$\tau_x = \sum_{i=1}^N (p_{iy} - p_y) f_{iz} - \sum_{i=1}^N (p_{iz} - p_z) f_{iy}, \quad (\text{B.7.4})$$

$$\tau_y = \sum_{i=1}^N (p_{iz} - p_z) f_{ix} - \sum_{i=1}^N (p_{ix} - p_x) f_{iz}, \quad (\text{B.7.5})$$

$$\tau_z = \sum_{i=1}^N (p_{ix} - p_x) f_{iy} - \sum_{i=1}^N (p_{iy} - p_y) f_{ix}, \quad (\text{B.7.6})$$

where p_{ix} , p_{iy} , p_{iz} are the position vector and p_x , p_y , p_z are the ZMP components respectively.

For a flat horizontal surface, $p_{iz} = p_z$ for all i , the last term of (B.7.4) and the first term of (B.7.5) is zero, we obtained $\tau_x = \tau_y = 0$, and only $\tau_z \neq 0$. This is the definition of *zero-moment-point*.

B.8 Lagrangian Dynamic Model

The Lagrangian of a mechanical system is commonly defined as [C. Chevallereau, G. Bessonnet et al., 2009] the difference between the kinetic (K) and potential (V) energies

$$L(q, \dot{q}) = K(q, \dot{q}) - V(q) , \quad (\text{B.8.1})$$

$$T(q, \dot{q}) = \frac{1}{2} \dot{q}^T D(q) \dot{q} , \quad (\text{B.8.2})$$

$$V(q) = \sum_{i=1}^N V_i(q) , \quad (\text{B.8.3})$$

where D is the symmetric inertia matrix.

Applying the Lagrangian notation, the equation of motion can be derived

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i , \quad (\text{B.8.4})$$

where f_i are the joint torques and other forces acting on the i -th generalized coordinate $q \in Q$. The generalized coordinate represent the robot's position, orientation and shape in 3D space.