PUSH RECOVERY THROUGH WALKING PHASE MODIFICATION FOR BIPEDAL LOCOMOTION

Albertus Hendrawan Adiwahono

Supervised by Assoc. Prof. Chew Chee Meng

National University of Singapore

2011

PUSH RECOVERY THROUGH WALKING PHASE MODIFICATION FOR BIPEDAL LOCOMOTION

Albertus Hendrawan Adiwahono

(B. Eng, M. Eng) ITB

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgements

I praise the LORD of all creation, whom I know through my lord and living savior Jesus Christ. Researching robotics has made me appreciate wonderful things that He had designed, such as my robust ability to walk and my excellent sensory system to percept my surroundings. More astonishingly, instead of creating me as a mere bipedal robot, the LORD made me with the capability to reason, to love, and to have a relationship with Him and other people. Such great art is amazing to think about.

My gratitude and respect to my supervisor Assoc. Prof. Chew Chee Meng, who has given me tremendous trust and freedom to develop my research. Thank you for patiently guiding me towards a better and mature researcher.

I thank my fellows in joy and pain of developing robots: Billy Saputra, Huang Weiwei, Tomasz Mareck lubecki, Renjun, Bingquan, Wen hao, and Mr. Soo the engineer. Also to the fellow students in my lab: Dau Van Huan, Wu Ning, Chanaka, XiaoBing, Hui, Boon Hwa, Chee Tek, Zhaoyan and many others that has been a friend.

Thank you to my parents and family, to my love Stephanie, and to my fellowship brothers and sisters who has been such a blessing to me. Thank you for the constant prayers, concern, and encouragement during my study. I thank my God every time I remember you.

Soli Deo Gloria!

Albertus, July 2011

Author’s Publication Related to the Thesis

- A. H. Adiwahono, Chee-Meng Chew, “Bipedal Robot Arbitrary Push Recovery through walking phase modification,” under review in ROBOTICA. 2011.
- A. H. Adiwahono, Chee-Meng Chew, Weiwei Huang and Van Huan Dau, “Humanoid Robot Push Recovery through Walking Phase Modification,” CIS-RAM, Singapore, 2010.

Table of Contents

Summary

Push recovery capability is an important aspect that a biped must have to be able to safely maneuver in a real dynamic environment. In this thesis, a generalized push recovery scheme to handle pushes from any direction that may occur at any walking phase is developed. Using the concept of walking phase modification and depending on the severity of the push, a series of intuitive and systematic push recovery decision choices is presented. The result is a biped that could adapt according to the magnitude of disturbance to determine the best course of action. Numerous push recovery experiments at different walking phases and push directions have been tested using a 12 DoF realistic biped model in Webots dynamic simulation. Afterwards, the performance evaluation and insights from our work are presented. Based on the performance analysis during our experiments, an additional controller is introduced to further improve the overall scheme. The versatility and potential of the overall scheme is also shown through a demonstration of the biped balancing on an accelerating and decelerating cart.

KEYWORDS: Bipedal robot, biped, bipedal walking, push recovery, walking phase.

List of Tables

List of Figures

List of Symbols

F (N)	Force
m (Kg)	Mass of LIPM
g (m/s^2)	Gravitational acceleration
θ (rad)	Angle between vertical axis and the LIPM leg
l (m)	LIPM leg length
τ_a (N/m)	Ankle torque in sagittal plane
τ_{ay} (N/m)	Ankle torque in lateral plane
$x_0^{(n)}$ (m)	Initial COM position with respect to the stance foot at step n in x axis
$x_T^{(n)}$ (m)	Final COM position with respect to the stance foot at step n in x axis
x_d (m)	Desired COM position with respect to the stance foot in x axis
$y_0^{(n)}$ (m)	Initial COM position with respect to the stance foot at step n in y axis
$y_T^{(n)}$ (m)	Final COM position with respect to the stance foot at step n in y axis
y_{dr} (m)	Desired COM position with respect to the stance foot during right foot swing phase in y axis
y_{dl} (m)	Desired COM position with respect to the stance foot during left foot swing phase in y axis
$\dot{x}_0^{(n)}$ (m/s)	Initial COM linear velocity at step n in y axis

$\dot{x}_T^{(n)}$ (m/s)	Final COM linear velocity at step n in y axis
$\dot{x}_d$ (m/s)	Desired COM linear velocity in y axis
$\dot{y}_0^{(n)}$ (m/s)	Initial COM linear velocity at step n in y axis
$\dot{y}_T^{(n)}$ (m/s)	Final COM linear velocity at step n in y axis
$\dot{y}_{dr}$ (m/s)	Desired COM linear velocity during right foot swing phase in y axis
$\dot{y}_{dl}$ (m/s)	Desired COM linear velocity during left foot swing phase in y axis
T (s)	Stepping time
T_n (s)	Normal walking stepping time
T_s (s)	Single support time
T_d (s)	Double support time
$x_{COM}(t)$ (m)	COM position with respect to origin at time t in x axis
$y_{COM}(t)$ (m)	COM position with respect to origin at time t in y axis
$x_{foot}(t)$ (m)	Foot position with respect to origin at time t in x axis
$y_{foot}(t)$ (m)	Foot position with respect to origin at time t in y axis
$z_{foot}(t)$ (m)	Foot position with respect to origin at time t in z axis
Z_f (m)	Desired step height
$E_x(t)$ (m^2/s^2)	Orbital energy at time t in x axis
$E_y(t)$ (m^2/s^2)	Orbital energy at time t in y axis

E_{dx} (m^2/s^2)	Desired orbital energy constant in x axis
E_{dy} (m^2/s^2)	Desired orbital energy constant in y axis
T_x (s)	Minimum time for the foot to travel $x^{+}_{\lim} - x^{-}_{\lim}$ in x axis
T_y (s)	Minimum time for the foot to travel $y^{+}_{\lim r} - y^{-}_{\lim r}$ or $y^{+}_{\lim l} - y^{-}_{\lim l}$ in y axis
T_z (s)	Minimum time for the foot to travel to lift up and land in z axis
T_k (s)	Stepping time that has been elapsed in a particular step
$\tau^{+}_{\lim}$ (N/m)	Maximum ankle torque output
$\tau^{-}_{\lim}$ (N/m)	Minimum ankle torque output
ϕ (rad)	Body posture in sagittal plane
ϕ_y (rad)	Body posture in lateral plane

Chapter 1

Introduction

Bipedal robot is a machine that uses two limbs to achieve locomotion. It is interesting that while bipedal locomotion seems easy and robust for humans and bipedal animals, it is very challenging for researchers to replicate the same level of robustness for bipedal robots. The goal of this dissertation is to develop control architecture for bipedal robots towards achieving a robust bipedal locomotion, especially in terms of the capability to recover from a push.

The subsequent sections provide an overview of bipedal robot development. A more detailed discussion of past and ongoing research of bipedal robot will be presented in chapter 2.

1.1. Background and motivation

As humans are bipedal, the idea to build bipedal robots is especially interesting. By studying bipedal locomotion we gain knowledge about human locomotion. In turn, this knowledge could be very useful in many areas beyond robotics itself. For example, the insights obtained from researching bipedal locomotion may contribute in developing devices and therapy methods to help people who lost their walking ability. Fig. 1 shows some of current and future ideas about bipedal robot applications.

There are many other reasons for developing bipedal robots. In general, legged robot has mobility advantages compared to wheeled robot in traversing terrains with gap and discontinuity such as terrains with pitfalls and stairs. Furthermore, a bipedal robot has the smallest foot print area compared to other types of robots, which allows it to maneuver effectively in a crowded urban area or to potentially step in a limited space such as small stepping stones.

It is our dream to build bipedal robots that can assist us in our dynamic and unpredicted environment. The idea is to have bipedal robots that can assist human to do the tedious tasks and replace human to do the dangerous tasks. However, it is very challenging to develop bipedal robots that have the mobility and robustness that are similar to human. The difficulties are mainly due to the limited understanding of bipedal locomotion, limited current hardware performance for a human-sized robotic system, non-linear dynamics, and limited capabilities of sensory systems to percept unpredicted environment interaction.

To be able to operate safely and successfully in a real life, outside of the research lab environment, a bipedal robot must have a certain level of robustness. This means a bipedal robot needs to have the ability to maintain its locomotion, such as walking, in the presence of unpredicted environment interaction.

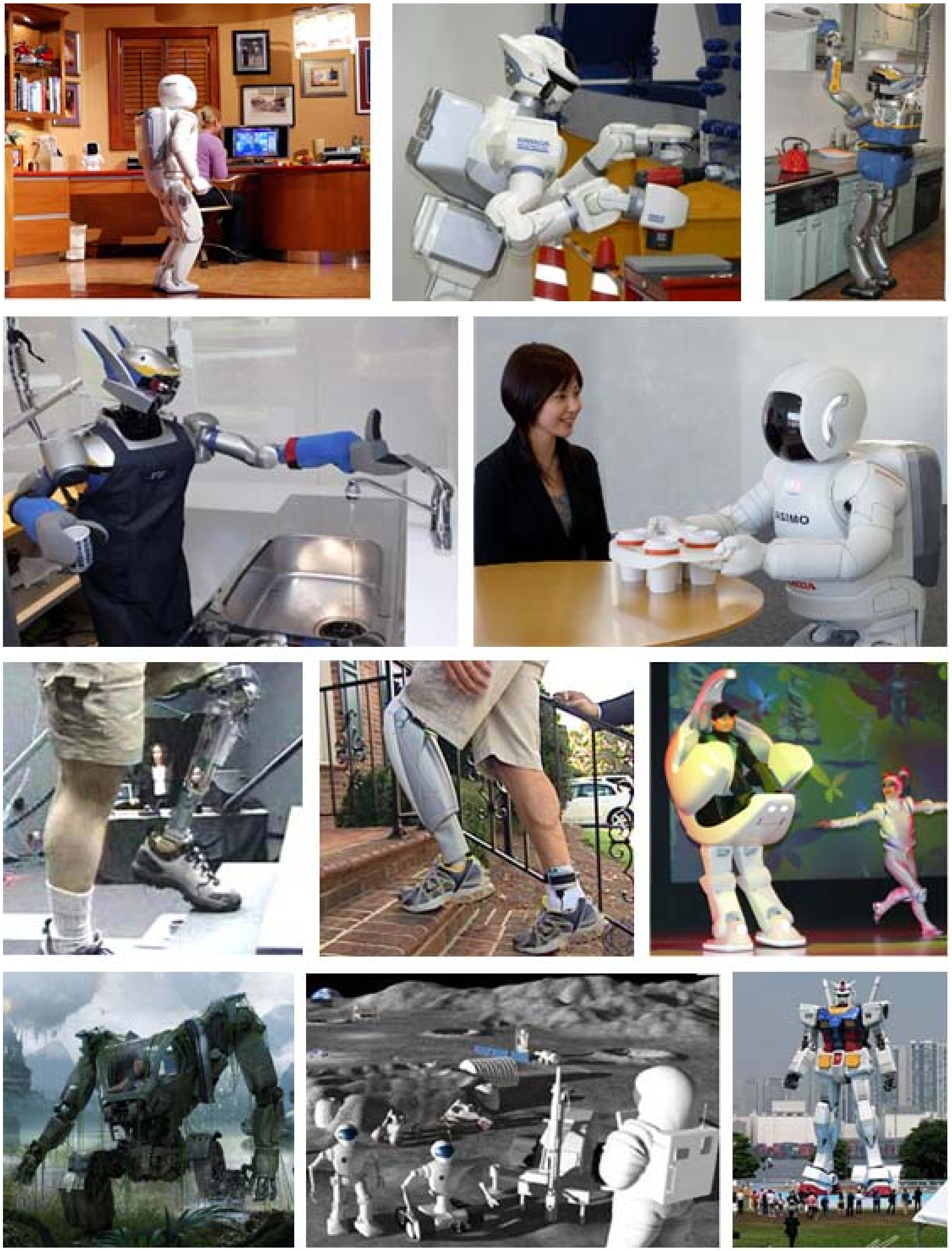


Figure 1: Bipedal robot research potential applications. First row: humanoid robot working in human environment. Second row: humanoid robot serving human. Third row: human locomotion assistive device. Fourth row: futuristic vision of bipedal robots.

Some of the most common forms of interaction in an environment are pushes. For example, in a crowded urban area some pushes (i.e. general force disturbance on the subject) and bumps will likely to occur occasionally. In sports such as soccer and football, violent pushes are almost inevitable. Fig. 2 shows the activities which may require push recovery capabilities

Several researchers have started to investigate the problem of push that occurs while the robot is standing. Given a disturbance, they try to investigate how the robot may maintain balance. For a small disturbance, simple ankle torque compensation may be enough to maintain balance. While for a larger forward disturbance, several steps ahead may be required to put the system back to equilibrium.

However, the problem of push recovery while the robot is walking is much more challenging and not much explored yet. Until now, the robust bipedal robot that can assist and replace human in a dynamic and unpredicted environment is yet to be seen. It is the goal of this research to find simple yet effective strategy to control the walking push recovery in humanoid robots.

Figure 2: Activities that may require push recovery capability

1.2. Objective and contribution

Research gaps for the current development of bipedal robot walking algorithm are summarized as follows:

- Most bipedal robot relied on a pre-planned (off-line) walking trajectory for its walking algorithm. Because the off-line algorithm is designed with little or no real time reactive ability, it does not have the robustness required to maintain the dynamic equilibrium of walking in the presence of strong unpredicted disturbance such as a push.

- Currently, there are very few studies on push recovery for bipedal robot walking. The current studies of bipedal robot push recovery have not systematically analyzed the different nature of pushes. Some works have claimed that the robot is able to maintain balance in the presence of “strong” disturbance, without defining clearly the magnitude, direction, and the timing of the push. A clear description of the push recovery problem is required.

- To comprehend the effectiveness of a particular push recovery strategy and to compare the performance between various proposed controllers, a more systematic performance benchmark in push recovery study is necessary.

The main aim of this thesis is to develop and propose a walking control architecture that has a push recovery capability for a bipedal robot. The push recovery capability will be demonstrated while the bipedal robot is stepping on the spot and walking forward. The magnitude of the push, the push duration, the line of action, and the walking phase when the push occurs will be considered in the general control architecture.

The specific objectives of the thesis are as follows:

- To introduce the walking phase modification as the main philosophy that could be used for push recovery.
- The bipedal robot could recover from an arbitrary push that is applied while the bipedal robot is stepping on the spot (i.e. walking with zero forward velocity).
- The bipedal robot could recover from an arbitrary push that is applied while the bipedal robot is walking forward.
- The performance of our push recovery controller could be used as a benchmark for future push recovery controllers or other push recovery schemes. To our knowledge, this thesis is the first to produce such benchmark.
- The proposed control architecture could be adjusted to maintain walking on an accelerating and decelerating cart.

The resulting control architecture should have the following specifications:

- It should be applicable for real time implementation. Hence it may be implemented in real bipedal robot hardware.
- It should be applicable for bipedal robots of different mass and size parameters.
- It should be applicable using current hardware technologies. The push detection sensor should use accelerometer, gyro, and pressure sensor, which are quite common in robotics. The actuator of the robot should be assumed using motor and harmonic drive system.

The result of this study may significantly contribute towards the development of robust bipedal robot locomotion control, especially in terms of push recovery capability.

The theoretical contributions of this thesis are:

- Systematic descriptions of the push problem, which helps to aim towards systematic push recovery study.
- Establishment of walking phase modification principle as a staple approach for push recovery during walking.
- Control policy that chooses the most energy efficient way of doing push recovery
- Iterative algorithm and the local joint modification as the strategies to compensate for the dynamics inaccuracies of a simple model. This thesis use LIPM (Linear Inverted Pendulum Mode) to model the actual biped with distributed mass and inertia.
- Synthesis of general control architecture for bipedal robot walking with push recovery capability.

The practical contributions of this thesis are:

- The practical consideration in the proposed method.
- Demonstration of the push recovery capability for bipedal robot walking in dynamic simulation.
- Application of the algorithm for balancing on an accelerating and decelerating cart.

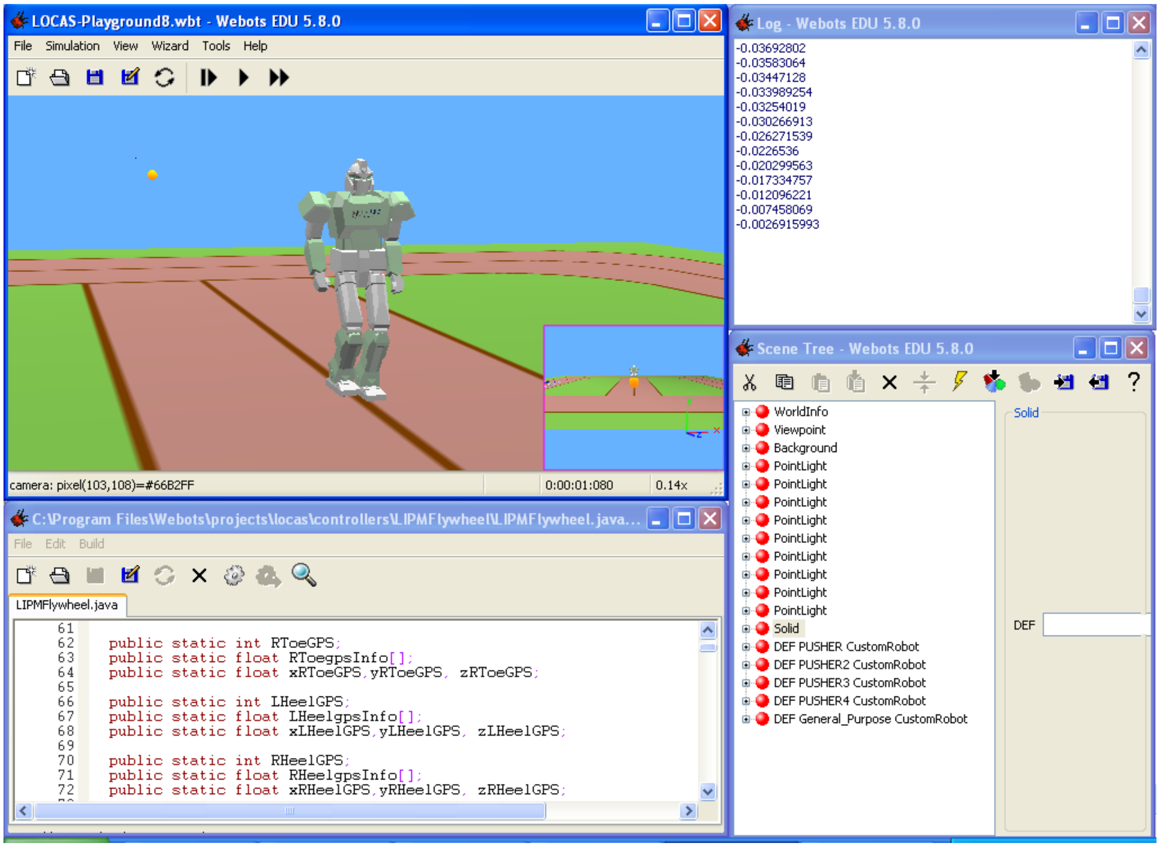
The scope of this research is restricted to push recovery for bipedal robot walking. The assumptions that are used in the algorithm will be explained in chapter 3.

1.3. Simulation tools

Webots is used as the main tool to develop and test the push recovery experiments in this thesis. Webots simulation software is developed by Cyberbotics. It is a development environment that can be used to model, program, and simulate mobile robots. The user could specify and construct one or more robot, in a shared environment. The properties of each object such as mass, moment of inertia, and friction are chosen by the user. Various simulated sensor and actuator is also available to be equipped for each robot.

We chose Webots as our simulation tool because it allows a bipedal robot to be tested in physically realistic simulation world. Webots is especially suitable for push recovery experiments because each object in Webots is defined by a surface (i.e. bounding box), which is an important feature to prevent two different objects from going through each other. Furthermore, Webots could be easily interfaced with CAD software, Java, C, and

MATLAB, which accommodate for the construction of a relatively complex humanoid robot and its control architecture. Fig. 3 shows the user interface of Webots.



```
-0.03692802
-0.03583064
-0.03447128
-0.033989254
-0.03254019
-0.030266913
-0.026271539
-0.0226536
-0.020299563
-0.017334757
-0.012096221
-0.007458069
-0.0026915993
```

```

    public static int RToeGPS;
    public static float RToegpsInfo[];
    public static float xRToeGPS,yRToeGPS, zRToeGPS;

    public static int LHeelGPS;
    public static float LHeelgpsInfo[];
    public static float xLHeelGPS,yLHeelGPS, zLHeelGPS;

    public static int RHeelGPS;
    public static float RHeelgpsInfo[];
    public static float xRHeelGPS,yRHeelGPS, zRHeelGPS;
```

Figure 3: Webots simulation user interface.

1.4. Thesis outline

The thesis is organized as follows:

Chapter 2 presents and discusses the literature review of bipedal robot research. The literature review focuses on the area that has influenced our thesis work, namely powered bipedal robot, model based approach, and push recovery study.

Chapter 3 proposes a generalized push recovery controller for bipedal walking. First, the problem of push recovery is described. Then, based on the problem and hardware consideration, a push recovery scheme is developed. The push detection, the walking phase modification scheme, and the control policy is presented and discussed. Then, the overall controller is implemented in a realistic 12 Degree of freedom (DOF) humanoid robot model. The push recovery performance is systematically tested and evaluated.

Chapter 4 proposes an additional strategy that further improves the performance of the push recovery. The considerations and implementation of the foot placement compensator is presented. Moreover, an additional implementation of the push recovery scheme for balancing on a moving cart is demonstrated.

Chapter 5 summarizes the contributions in this thesis and outlines directions for future research.

The appendixes present the details of the thesis. Although these details may not be significant for general readers, they could be valuable for readers that would like to closely study the proposed method or for engineers who would like to replicate the work done in this thesis.

Appendix I clarifies the derivation of the LIPM with ankle torque model.

Appendix II describes the LIPM in the lateral plane .

Appendix III describes the details of the normal walking implementation.

Appendix IV presents the details of all the algorithms in this thesis

Appendix V shows the simulated realistic humanoid robot model dimension, mass, and inertia properties.

Appendix VI presents the development of the biped robot NUSBIP-III ASLAN.

Chapter 2

Literature Review

This chapter presents a literature review on the development of bipedal robot, especially around the area that has an impact on our research work. In general to specific order, the areas are: powered bipedal robot, model based approach, and push recovery research.

2.1. Bipedal robot development overview

Developing humanoid bipedal robot has been the dream of many scientist, artist, and engineers. The earliest record of bipedal robot development perhaps dated around the year 1495, when Leonardo Da Vinci developed a humanoid automaton. In 1969, Dr. Ichiro Kato started the first humanoid robotics research team at Waseda and developed the WL robots series [1]. Around the same time, M.Vukobratovic [2] introduced the concept of zero-moment point (ZMP) for the analysis of bipedal locomotion which has been widely used by many researchers until now. In early 1980s, M. Raibert [3,4] developed the hopper robots to investigate active balance and dynamic stability in legged locomotion. His idea has been influencing today's advanced legged robots such as

Bigdog and petman. In early 1990, Mcgeer pioneered the study of passive walkers which emphasis the efficiency and naturalistic approach to achieve bipedal locomotion [5]. Fig. 4 shows some of the earliest bipedal robot works.

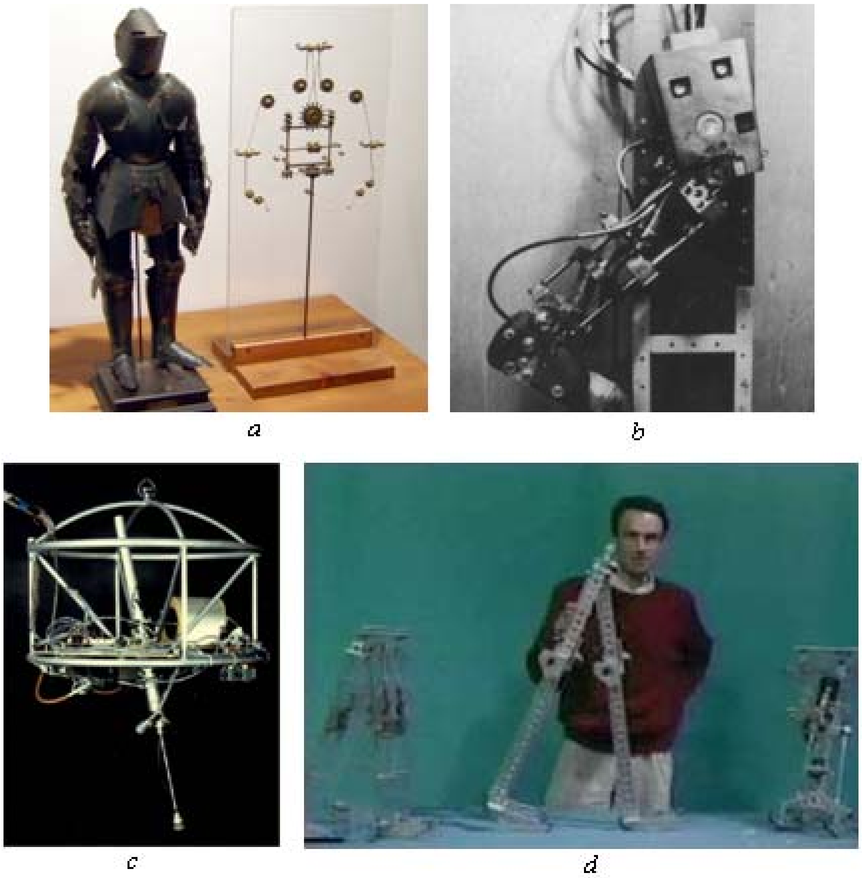


Figure 4: Some of the earliest legged robots. Fig 4a: Leonardo’s robot, Fig 4b: W-L1 by Kato, Fig 4c: The hopper robot by Raibert, Fig 4d: Early passive walkers.

Since bipedal robot is such a complex and broad problem, there have been many researches and approaches on bipedal locomotion. In present day, bipedal robot walking research could be divided into two main paradigms [6]. The first one is the powered walking approach. The second paradigm is the passive walking approach.

2.1.1. Powered bipedal robot

There are many kinds of control algorithms that have been used to control powered bipedal robots. While many approaches may not be mutually exclusive to each other, we could list them as: Model based, biologically inspired based, Imitation based approach, and heuristic based. The model based powered bipedal robot, which is the approach that we choose for the bipedal robot discussed in this thesis, will be discussed in more detail in section 2.2.

The biologically inspired approach builds the fundamentals of the control algorithm based on the observation and interpretation of how living creature works. Neurophysiological studies suggest that walking gait could be generated by a central pattern generator (CPG) in the spinal cord [7,8]. The CPG generates rhythmic excitation signals that control the actuators. CPG based approach is often used to control mobile robots that moves in a highly repetitive manner such as a swimming eel or snake robot. In the bipedal walking implementation, the CPG is usually used together with sensory feedback. Examples are Aoi and Tsuchiya [9], Endo, *et al.* [10], Nakanishi, *et al.* [11], and Shan, *et al.* [12]. Although the idea of CPG approach is very interesting, a rhythmic pattern itself is not a necessary condition to achieve bipedal walking. Furthermore, the general normal walking pattern will not be sufficient to maintain walking when the bipedal robot encounters hard disturbances.

The imitation based approach uses joint trajectories acquired from direct measurement of human subject as the main building block for the controller. It has been used since the early development of Honda Asimo’s predecessor; P2 and P3 [13]. By combining the prerecorded trajectories and several on-line compensators, P2 and P3 are able to walk relatively fast (about 1 m/s) and walk over 10-degree inclines. A common problem for the imitation based approach is the fact that the human subject and the humanoid robot have different mass distribution, moment of inertia, joints location, and degree of freedom. Despite the main issue, the motion of a humanoid robot that imitates human seems to be more graceful compared to the motion generated by trial and error.

The heuristic or algorithmic controller uses a set of intuitive precondition and action relations to generate walking. The state machine was simply built based on the conditions that occur during walking, such as the single support time, the swing foot touches the ground, and the double support time. Some early examples of robots built with the heuristic approach are Timmy by Eric Dunn and Robert Howe [14, 15] and Spring Flamingo by Pratt [16, 17].

Besides control algorithm, powered bipedal robot is also limited by the capability of its actuators. Until present day, harmonic drive system is arguably the most reliable and powerful drive system. It has been implemented in advanced powered bipedal robots such as ASIMO, Toyota robot, and HUBO. However, due to a large gear reduction ratio, the harmonic drive system is hardly back drivable and this limits the controller into a position tracking mode. Hence, the walking control algorithm must use local feedback control on the joints, where energy is used to track the desired joint position regardless of the workload. Consequently, today’s powered bipedal robots generally have poor energy efficiency and flexibility.

Another alternative to the powerful but stiff harmonic drives is the compliant artificial muscles. In contrast with the harmonic drives, the elastic actuator allows force control mode to be implemented. These kinds of actuators have been used in the Spring

Flamingo [16, 17] and the well-known SARCOS humanoid robot [51]. However, these robots have not yet achieved dynamic walking as shown by the robots that are using harmonic drives. Further research is much needed to developed artificial muscles that have similar performance with the human muscles.

Despite the challenges, some of the ongoing powered bipedal robot researches have shown promising results. Some of the leading research institutes around the world have shown their impressive bipedal robot namely Honda ASIMO [13], HRP [18], TOYOTA robot [19], and HUBO [20] (Fig. 5). Although these robots are not yet applicable for a practical daily implementations assisting humans, their achievements have become the milestones which motivate further researches in bipedal robotics.

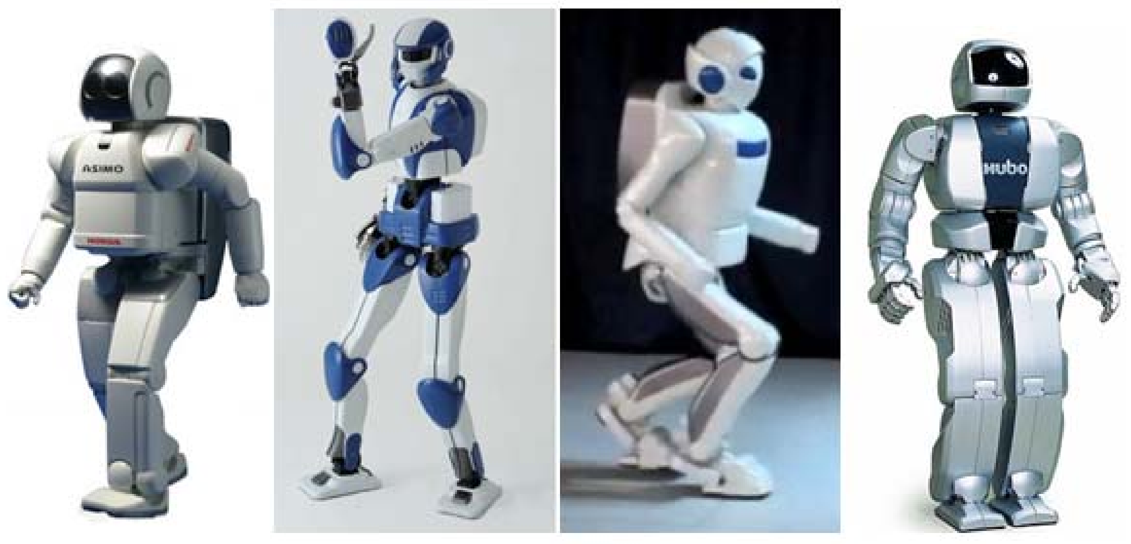


Figure 5: Today’s leading powered bipedal robots. From left to right: ASIMO by Honda, HRP-4 by AIST, Toyota humanoid robot by Toyota, and HUBO by KAIST.

2.1.2. Passive bipedal robot

The Passive bipedal walking approach relies on the dynamics of the legs and body to produce walking. This approach does not use position control as in the previous method but focuses on producing a stable cyclic gait. McGeer [5, 21] shows an underactuated robot descending a slope powered with only gravity.

Today, passive walkers are able to walk in a level ground by subsequently producing active power on the hip or the ankle. The power is used to compensate energy losses due to impacts and frictions during walking. In contrast, this power is meant more to shape and fine tune the natural dynamics rather than to impose prescribed kinematic motions as in the powered bipedal walking approach. Several robots developed by TU delft such as Denise [22] and its successor Flame have shown some promising natural efficient walking. Fig. 6 shows some of the most well developed passive bipedal walkers.

Although passive walking robots has a great advantage in terms of energy efficiency, generally it suffers from a poor versatility. Because most of the joints are not actuated and underpowered, present day passive walking robot often could not do other task besides walking.

The development of semi-powered robots like Flame and Denise starts to blur the line between a powered bipeds and passive bipeds. It has been suggested that an ideal future bipedal robot should have the positive traits of both powered and passive bipeds. It should have the versatility and strength of a powered biped to manage different tasks. On the other hand, it should also be able to walk efficiently and gracefully like a human does.

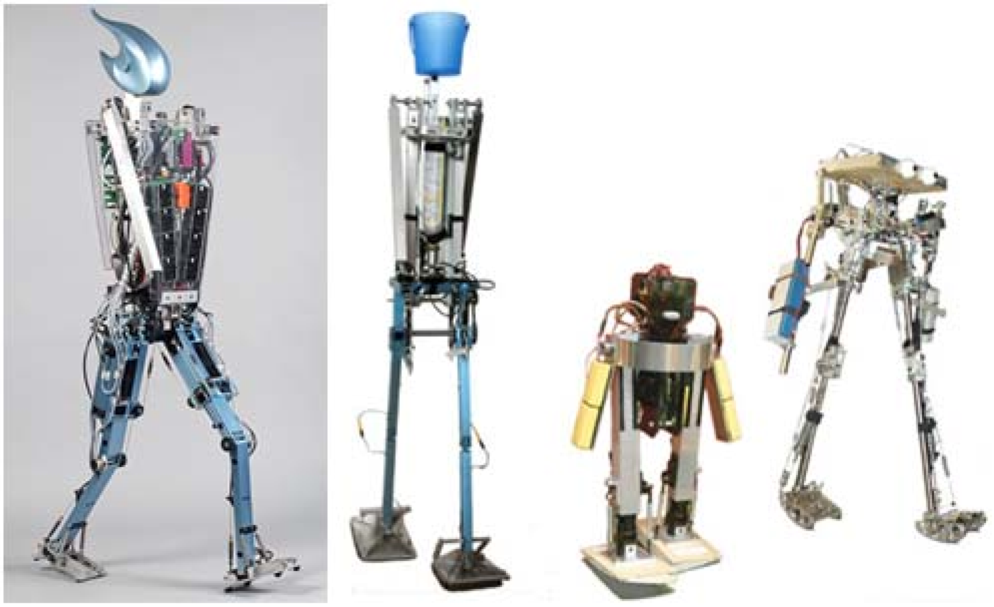


Figure 6: Passive bipedal robots. From left to right: Flame and Denise by TU Delft, Toddler by Massachusetts Institute of Technology, and The Cornell Biped by Cornell University.

2.2. Model based approach for powered bipedal robot

Since a bipedal robot may consists of a large number of mechanical and electrical parts, links, joints, and actuators, it is often considered impractical to calculate the exact physical properties of the system. To further complicate the matter, bipedal walking has a highly non-linear dynamics that could not be solved easily with traditional control techniques. Because of these difficulties, researchers have recognized the usefulness of using a model as a tool of analysis.

In the model based approach, a physical representation of the robot along with its mathematical derivation is used to estimate the dynamics of the robot. The model may vary from a very complex model, with many points of mass connected in a “tree branch” configuration with its respective moments of inertia, to a very simple model with a single point of mass and massless legs.

A complex model will have a better estimation of the robot dynamics, provided that the model itself is accurate to the actual robot. For example, the Acrobot model [23] included the inertia and dynamics of the leg. In another example of more complex model by Kajita, *et al*. [24], the inertia of every link in the robot is incorporated in planning a motion. However, a complex model often suffers from a high computation burden which limits its real-time implementation. Furthermore, the overall dynamics are highly complex and nonlinear which often required further linearization.

On the other hand, a simple model has its own advantages compared to the complex model. It is ideal for real time implementation due to its low computation burden and complexity. The dynamics equation of a simple model can often be solved analytically with relative ease. The inaccuracy in the estimation of the robot dynamics can often be compensated with some fine hand tuning [e.g. 25, 26] or machine learning [e.g. 27, 28]. Kajita *et al.* [29] proposed the Linear Inverted Pendulum Mode (LIPM), which is an effective simple model that has been heavily influencing modern bipedal locomotion research. In LIPM model, the robot’s body is assumed to be an inverted pendulum with a point mass that moves linearly with a constant height. Many bipedal robot walking controllers have been developed and implemented using this model [e.g. 30, 31, and 32].

A useful tool to analyze the LIPM trajectory is the concept of orbital energy [29]. The orbital energy is a kind of energy that describes the class of trajectory based on LIPM dynamics equation. Based on the magnitude of the orbital energy we could determine the behavior of the LIPM motion. For example when orbital energy is positive, the COM (center of mass) will swing from the minus side to the positive side of horizontal axis or vice versa. When the orbital energy is zero, the COM will stop at the equilibrium point. When the orbital energy is negative, the COM never passes the equilibrium point. Orbital energy is constant if:

- The COM moves horizontally with constant height and the leg is massless, which is the properties of LIPM.
- There is no disturbance.
- There is no energy loss during stepping.
- There is no energy added, or zero ankle torque input.

The relatively simple LIPM model is also often used together with the Zero Moment Point (ZMP) criterion to develop a dynamic stable walking motion. The Zero Moment Point is a point in the ground where the total influence of all forces acting on that point is zero [2]. In the case where the foot is stationary the regular ZMP coincide with the Center of Pressure (COP). In the case where the foot is experiencing rotation with respect to horizontal axis, the COP is on the edge of the foot. But this point would not be the regular ZMP anymore since it is not the point where the moment about two horizontal axes is zero [63]. In this case, the theoretical ZMP is outside the support polygon, which is a point on the ground where the ground reaction forces would have to act, in case of infinite foot size, to keep the foot stationary. The term fictitious ZMP (FZMP) [63] or Foot Rotation Indicator (FRI) [68] has been suggested to refer to this virtual point. In the real case, the foot would have rotated and the whole mechanism would have collapse if no action is taken.

The ZMP criterion is as follows: As long as ZMP lies strictly inside the support polygon of the foot, then the desired trajectories are dynamically feasible. If the ZMP lies on the edge of the support polygon, then the trajectories may not be dynamically feasible. When combined with joint trajectory control, the concept is sufficient to realize a dynamically stable biped trajectory. Numerous stable dynamic walking motion schemes have been realized using this criterion [e.g. 33, 34, 35, and 36].

The main limitation of this criterion is that the concept is not necessary for a dynamically stable walking. For example, during toe-off in human walking the ZMP criterion is violated and yet the human does not necessarily fall. Similarly, during push recovery the criterion may be violated but the biped may still be able to recover from the push.

Another limitation of the trajectories calculated based on ZMP is that the calculation usually relies on previewing several steps ahead of the robot (i.e. in the preview control [36]). Hence, there is a need to plan the walking trajectory, at least two steps ahead. This may work well in a controlled and predicted environment, such as the lab or on a performance stage, but it will not be adequate to handle large disturbances. It also involves discrete optimal control, which has relatively heavy computation load for real-time application.

2.3. Bipedal robot push recovery

2.3.1. Push recovery while the bipedal robot is standing

While balancing seems easy for humans, it has been an intriguing problem for bipedal robot. This section presents past research works on balancing or push recovery while the bipedal robot is standing still on the ground.

Hofmann [37] presented three basic strategies to balance a standing biped. He pointed that the key for balancing is to regulate the horizontal motion of the center of mass (COM). For small push, the first strategy is to simply shift the center of pressure (COP) by modifying the ankle torque. When this strategy is not sufficient to recover the biped, the second strategy is needed. The second strategy is to create a moment about the COM that would affect the COM motion. Finally, if the second strategy is not enough, the biped needs to take a step to the recover the balance.

To avoid unnecessary dynamic complexity, most researchers have chosen simplified models as the tools to analyze balancing problem. The LIPM model, which has been widely used to model bipedal walking, is very useful to analyze the motion of the COM during a push recovery [38]. In several studies, the COM of the LIPM is modified by adding a rotational moment of inertia (flywheel) [39, 40, 41, 42, 43, and 44]. The additional flywheel models the angular momentum, which could be used for push recovery as described in Hofmann's second strategy.

Pratt, *et al*. [41] introduced the concept of "capture point", which is a stepping point to determine where the biped should step after being pushed in order to return to its equilibrium standing position. Although the concept is appealing, modeling errors made the capture point could not be exactly determined solely from the LIPM. To solve this issue, Rebula, *et al*. [45] proposed to use machine learning, which could amend the estimated capture point from the LIPM model. Wight, *et al*. [46] also presents the foot placement estimator to predict the location of the capture point.

Besides the LIPM model, some researchers used inverted pendulum and double inverted pendulum to analyze balance [47]. Various control techniques has been proposed to control the pendulums, such as optimization [48], integral control [49], and linear feedback [50].

Researchers have also tried to integrate various approaches towards a more thorough push recovery strategy. Stephens [51] combined the ankle and hip strategies in the proposed balancing controller. Later on, Stephens, *et al*. [52] also implemented the model predictive control on the SARCOS biped, and demonstrated a push recovery. Hyon, *et al*. [53] presented a multi level postural balancing for humanoid robot, and demonstrated some push recovery, while the SARCOS biped is pushed from behind. Fig. 7 shows the SARCOS robot being disturbed during one of their experiments.

Yi, *et al.* [70] also implemented the ankle, hip, and stepping in their small humanoid robot Darwin-HP. In the approach, reinforcement learning is used to determine the parameter in dynamic simulation. Then, the result is implemented onto the Darwin-HP. This work implemented and combined the approach proposed by Stephens [51], Pratt [41], and Rebula [45]. However, because modeling inaccuracy between the simulated robot and the real robot, the effectiveness of the learning is limited.

All of the above examples are the work done for push recovery while the biped is stationary. The next subsection will present the frontier of bipedal research, which is push recovery while the biped is walking.

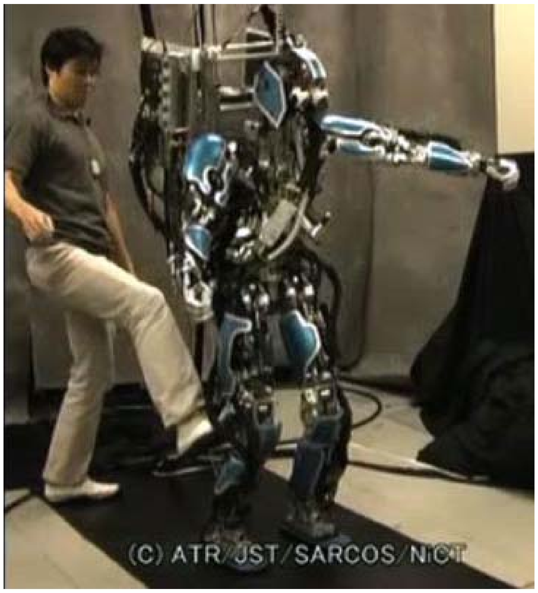


Figure 7: SARCOS robot being disturbed in a push recovery experiment.

2.3.2. Push recovery while the bipedal robot is walking

In traditional bipedal walking literature, there have been lots of methods proposed to stabilize a walking biped. However, most of these methods are designed with the assumption that the perturbation is small. For example, the perturbation could be due to the biped is walking on a rugged terrain, uneven terrain, or due to dynamic inaccuracies. Huang, *et al.* [54] proposed a feedback control system based on ZMP criterion and landing time regulation. Kajita, *et al.* [29] proposed a method using the concept of orbital energy to stabilize a biped while it is walking on rugged terrain. In this approach, the solution depends on the assumptions that the biped will always have enough time to step on the fixed stepping location. While these approaches are sufficient to enable the biped to walk on rugged terrain, it is not sufficient to handle big disturbances. A more general push recovery approach is needed.

Very few researchers have explored the problem of large push recovery during walking, in which a biped must withstand the push and maintain walking at the same time. Komura, *et al*. [55, 56] proposed a theoretical feedback controller scheme for bipedal walking that could recover a biped from a push in sagittal plane. The strategy applied the hip strategy and modified the stepping location in order to reduce the excessive angular momentum of the LIPM. The result was presented in 2D numerical animation. This hip strategy is similar to the flywheel strategy that has been used for push recovery when the biped is standing. Unfortunately, although the title of the paper mentions a large perturbation is inflicted to the biped, there is no data the walking phase at which the push is inflicted and the magnitude of the push. Furthermore, there is also no verification whether the biped could maintain walking. It seems that the study was meant to animate the reactive motion of human.

Wieber, *et al*. [57, 58, and 59] developed an online walking motion generation based on the model predictive control approach. In their latest result shown in [59], the algorithm minimized the jerk, COM velocity, and Zero Moment Point (ZMP) errors in order to improve the disturbance rejection capability during walking. A numerical simulation result using the LIPM is presented to verify the result. In the simulation, a LIPM with flat feet is pushed at the beginning and middle of walking phases. To verify that the biped does not fall, the center of pressure (COP) is verified to be inside the support polygon. However, the method required relatively large computation load and the ability to track ZMP error, which is not easy to be realized in a biped that has a fast walking motion.

Despite the many challenges in developing such a robust push recovery controller for bipedal robot walking, some researchers have shown that is indeed possible to be realized. Recently, a real bipedal robot that could maintain the gait after receiving a push has been demonstrated. In 2009, a real Toyota humanoid robot by Tajima, *et al*. [60, 19] showed an impressive push recovery capability while the robot is running on place. The robot, which is also modeled with by a point mass, has been developed to achieve jumping and running. The balance controller consists of a compliance controller and a feedback controller to the motion generation. The compliance controller is used to absorb the shock from the impact, and then the feedback controller recalculated the COM trajectory and foot placement. Fig. 8 shows the snapshots of Toyota humanoid doing a push recovery [19]. Around the same time, a real biped by Boston Dynamics named PETMAN [61] has also shown a push recovery from the lateral side while walking (Fig. 9). Unfortunately, because these robots are funded and developed by large corporations, much of their experimental data, algorithms, and hardware specification are classified. Furthermore, the results are shown in lab environment and these robots have not shown that they are ready for dynamic environment.

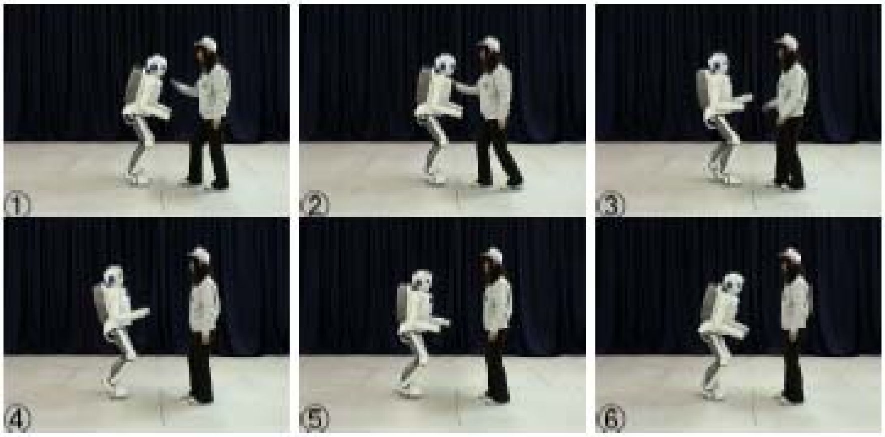


Figure 8: Toyota robot doing a push recovery while running on the spot [58].

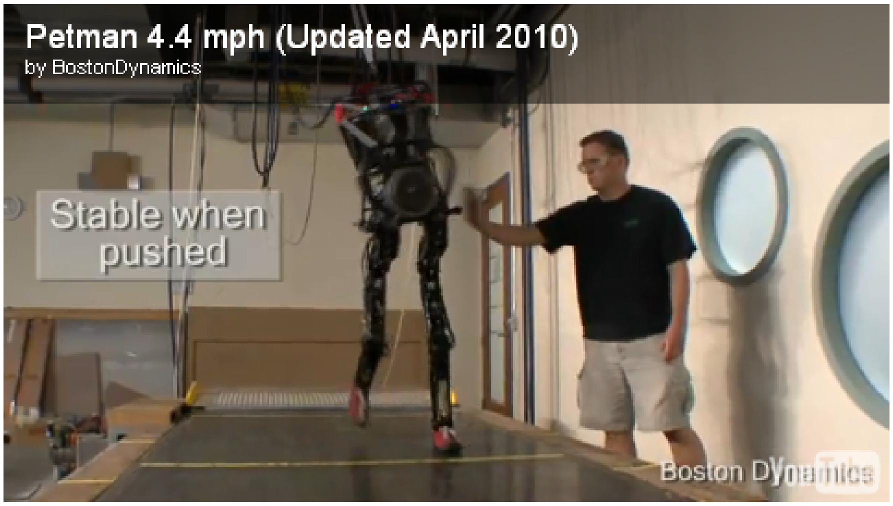


Figure 9: PETMAN doing a push recovery while walking [60].

Yi, *et al* [71], integrated the push recovery approach described in [70] with a swing foot compliance scheme to recover the biped from disturbance caused by uneven ground. In [72], Yi, *et al*. continued to develop the scheme for omnidirectional walking. In this work, the reinforcement learning is done directly in the physical robot instead of in the simulated robot. However, the main limitation to the scheme is that the biped could only aim to stop walking when it detects there is a big disturbance.

2.4. Summary

In this chapter, we present and discussed the literature review of current bipedal robot development, especially around the area of powered bipedal robot, the models that has been used to represents a biped, and the push recovery research. As described earlier, the development of powered bipedal robotics today has a significant dependence on position tracking and playback of recorded trajectories, which has low robustness in terms of push recovery. Very few researchers have started the work on push recovery for bipedal robot walking. Even fewer researchers have investigated the push recovery problem thoroughly and none have presented the push recovery performance systematically.

Chapter 3

Proposed Control Architecture

3.1. Background

As mentioned earlier, most powered biped robot today walks with an offline predefined trajectory with a rather weak adaptive walking behavior. Consequently, these bipeds could not survive maneuvering in a dynamic human environment where strong unpredicted disturbances may occur at any time. To be able to safely walk in an uncontrollable environment, further systematic research is needed to improve a biped's capability to maintain walking after receiving an unforeseen push.

In this chapter, we propose a generalized push recovery scheme that could help a biped maintain its walking after being pushed from any direction, at any walking phase, up to a certain magnitude. The proposed algorithm is designed with practical hardware considerations, sensory feedback possibilities, and a relatively low computation cost that enables the scheme to be realized online. We would also like to introduce the concept of walking phase modifications, which is the main philosophy for our push recovery scheme. The idea is to modify the acceleration and deceleration phase of the biped in order to recover it to the normal walking state. Combined with the scheme to conserve energy, we develop a control policy that would be used as a guideline to recover from a push. The performance evaluation presented in this thesis could serve as a benchmark to compare the effectiveness of our proposed method with future methods or other schemes.

3.2. Problem of push recovery for bipedal robot walking

This section discusses the characteristics of the push recovery problem and some of the points that are considered when we design the push recovery strategy in the subsequent sections.

A push is a loose term for a general force disturbance on a subject. In bipedal robot research, a push is sometimes used to verify the robustness of a particular control strategy. However, often there is very little explanation about what kind of push is applied in the experiment. For clarity of the push recovery problem, let us examine more closely on the nature of the push event itself.

In the scope of this thesis, the attributes of push that will be thoroughly considered are:

- Magnitude of the push: The magnitude of the push could vary from a soft touch up to a certain level of magnitude, which will be discussed in the performance evaluation.
- Line of action: The push could come from any direction in the transverse plane.
- Walking phase when the push occurs: The push could occur at any walking phases in the biped’s walking cycle.

To focus our study on the above attributes, some issues will be constrained or simplified with the following assumptions: The duration of force acting on the biped is short, or instantaneous, as compared to the stepping time of the biped. As the force is large enough to produce a definite change in momentum during a very short time, we shall refer this force as an impulsive force [62]. In the transverse plane, the impulsive force is assumed to be acting on the biped’s center of mass (COM), which is a necessary assumption in order to use the LIPM model. It is also assumed that the ground is level and the leg motion is not hindered.

We will use the Linear Inverted Pendulum Mode (LIPM) [29] as a tool to analyze the push recovery problem. The advantages of using LIPM for push recovery approach are as follows:

- Solution of the linear differential equation could be obtained with relative ease. Hence it is especially helpful to analyze difficult and novel research area such as push recovery from various magnitude, direction, and timing during walking.
- Low computation cost makes the LIPM attractive for real-time application, where fast decision making is required. This feature is especially important during hard push recovery where a fraction of a second could differentiate between a success and a failure.

The LIPM also has its limitations, such as it is not an accurate representation of the whole distributed mass dynamics and the angular momentum of the biped. This is the tradeoff between modeling accuracy and simplicity. A local joint compensator and an iterative push recovery steps is used to compensate for these limitations.

Next, we would like to address the problem of push detection for bipedal robot. The detection problem is important because this is one of the key factors in choosing the course of our push recovery strategy. Unlike human, current robotic technologies do not allow a real bipedal robot to directly sense a force exerted to an arbitrary spot on its body yet. Therefore, it is currently impractical to use the force of the push as a detection method. Another possibility to detect a push is to sense the ground reaction force. However, precise ground reaction force measurements usually require a fairly complex filtering and high computation time, which is unsuitable for dynamic walking application.

In our scheme, orbital energy [29] is used as the key information to detect whether or not the biped needs to do push recovery. Orbital energy remains constant during normal bipedal walking, assuming there is no energy loss during stepping and the biped walks with constant average velocity. Therefore, it can be used to evaluate the walking energy of a biped at any time. For LIPM, orbital energy is a function of the horizontal linear velocity and the COM displacement with respect to the stance foot. The linear velocity information could be obtained relatively fast with low computation cost using current robotic inertial sensors and the displacement could be derived from LIPM equation of motion.

3.2.1. Dynamic balance of bipedal robot walking

Before the main strategy is presented, we would first discuss about the 'dynamic balance' [63], as some authors may prefer to refer it as 'stability', of bipedal walking. The main purpose of designing a bipedal walking control is to achieve a dynamic balance, or to maintain walking. In bipedal robot walking research, there are many criterion proposed to evaluate whether a bipedal walking could be classified as 'stable'.

In the powered bipedal robot research area, the notion of ZMP dominates as the tool to analyze bipedal walking. If the ZMP is always within the support polygon of the bipedal robot during walking, the bipedal walking is considered to be stable. This criterion is often referred as stability margin criterion. Combined with some other schemes, this criterion has been applied in some of the most successful bipedal robots walking to date [13, 18, and 19]. However, although these robots could perform dynamic walking, most of these robots are not yet ready for unknown strong disturbances in the real dynamic environment.

However, in relation to push recovery, ZMP criterion is not necessary conditions to maintain walking. Although the criterion could be violated, there is still a possibility that the bipedal robot could maintain walking. The push recovery strategy is especially designed to deal with strong disturbances that will almost immediately violate the undisturbed normal walking condition. For example, the strong magnitude of the impulsive force may cause the theoretical ZMP to be shifted outside of the support polygon before the biped could even complete a recovery step. The subsequent steps after the push also do not necessarily have to comply with the normal walking cyclic pattern.

Therefore, we adopt a more general definition of dynamic balance similar to the one proposed by Pratt, *et al*. [38] which states that a biped is stable if and only if the robot does not fall after some reasonable finite amount of time, where fall is defined as any part of the body other than the feet is touching the ground. In this sense, a successful push recovery simply means that the biped does not fall within some reasonable finite amount of time after a push has occurred. All bipeds, including human, will fall if the push exceeds some magnitude. Thus, push recovery scheme does not guarantee that a biped will always maintain walking when disturbed. Instead, the key consideration is what scheme can be used to increase the chance of a successful push recovery.

3.3. Push recovery strategy

Based on the characteristics of the problem, a push recovery scheme is designed. As mentioned earlier, a LIPM will be used as the tool of analysis for the push recovery strategy. The analysis results and the overall strategy will then be applied to a simulated humanoid robot.

3.3.1. Overview of push recovery strategy

Suppose the normal walking of the biped has been established as described in Appendix III. The overall flow of the push recovery scheme is as follows (Fig. 10): First, the push detection will monitor the biped's state at all time during walking. Once a push is detected, the walking phase modification scheme will be active, and the biped will be in a *pushed state*. Generally, 'walking phase' could refer to different phases of walking such as single support, double support, swing time, stance foot transition, etc. However, in our term the 'walking phase' refers to the dynamics point of view of bipedal walking, where a bipedal walking cycle could be divided into an acceleration phase and a deceleration phase. The scheme will modify the walking phase of the biped so as to enable the biped to recover to its normal walking state. The modification of the walking phase can be seen as a set of decisions that adjust the walking gait. The walking phase modification is also more than simple feedback adjustments that only modify certain joint angles as it is commonly implemented in today's powered bipeds.

In essence, the approach tries to bring the LIPM closer to its desired states for every step during a *pushed state*. The walking phase modifications consist of two main parts:

The first part is the modification in the acceleration phase, which regulates how much the COM travels through the determination of the stepping time and the ankle torque. A control policy will be used as the guidance for the regulation. The second part is the modification in the deceleration phase, which regulates the foot placement determination. The walking phase modification is subject to the constraints on the bipedal robot itself. In the walking phase modification, the acceleration phase will be modified before the deceleration phase modification. The reason is that the controller must ensure that the biped state does not violate the constraints because of the push. Once the acceleration phase has been safely modified, the deceleration phase is modified to bring the LIPM states converge to the desired states.

When applied to the humanoid robot with distributed mass, the walking phase modifications will be executed together with some subtler local joint compensator in order to maintain the biped's dynamic balance. Although the local joint compensator is not strictly necessary, it is useful to compensate for dynamic inaccuracies and small deviations between the LIPM and the realistic humanoid robot model. With this scheme, the biped is able to recover from the push. Each section will be presented in more details in the subsequent sections.

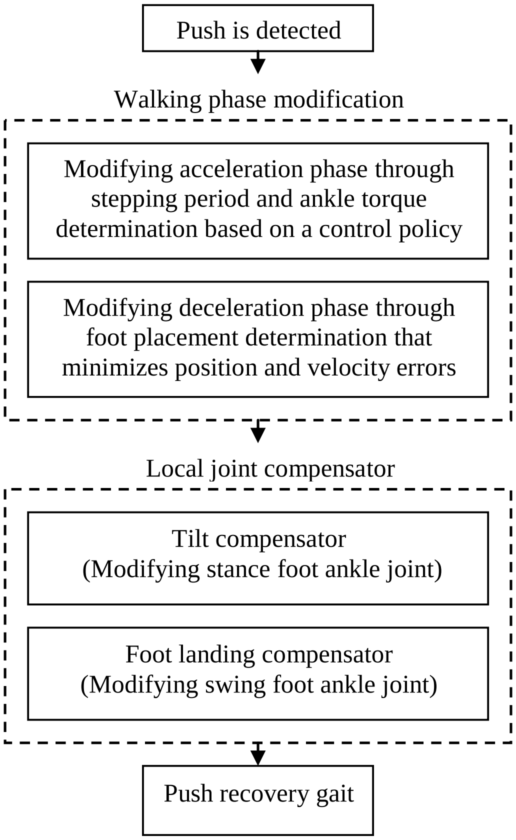

Push is detected
Walking phase modification
Modifying acceleration phase through stepping period and ankle torque determination based on a control policy
Modifying deceleration phase through foot placement determination that minimizes position and velocity errors
Local joint compensator
Tilt compensator (Modifying stance foot ankle joint)
Foot landing compensator (Modifying swing foot ankle joint)
Push recovery gait

Figure 10: Overview of push recovery strategy

3.3.2. Push detection

This section will describe the push detection portion of the algorithm. To detect a push, the states of the LIPM and its orbital energy needs to be monitored. The LIPM dynamics is described in the following subsection.

3.3.2.1. LIPM dynamics

A 2D LIPM with a joint at the point foot that represents ankle torque input is used to model the biped. The LIPM for sagittal and lateral plane are assumed to be decoupled [64]. The LIPM itself consists of a point mass with a massless leg. The point mass is constrained to move straight with a constant height. The sagittal plane LIPM free body analysis can be seen in Fig. 11.

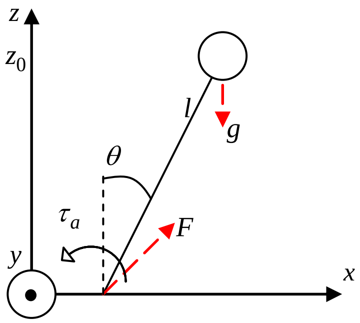


Figure 11: LIPM with ankle torque.

In Fig. 11, z axis is pointing upward, x axis is pointing forward, y axis is pointing inward, F is the ground reaction force, m is the mass of the model, g is the gravitational acceleration, l is the length of the leg (the length of the leg varies over time to support the COM trajectory at constant height), θ is the angle between vertical axis and the line joining the ankle joint and the COM , and τ_a is the ankle torque. The constraints for LIPM: $z = z_0$, $\dot{z} = 0$, and $\ddot{z} = 0$ where z_0 is the constant COM height. The dynamics equation of the LIPM [30]:

$$\ddot{x} = \frac{g}{z_0}x - \frac{\tau_a}{mz_0} \tag{1}$$

Suppose τ_a is a constant (step input sagittal torque). Then, we have the solutions of Eq. (1) as follows:

$$x(t) = \cosh(wt)x_0 + \frac{1}{w}\sinh(wt)\dot{x}_0 - \frac{\tau_a}{mz_0w^2}(\cosh(wt) - 1) \tag{2}$$

$$\dot{x}(t) = w\sinh(wt)x_0 + \cosh(wt)\dot{x}_0 - \frac{\tau_a}{mz_0w}\sinh(wt) \tag{3}$$

x_0 is the initial x coordinate position of the COM with respect to the stance foot, $\dot{x}_0$ is the initial velocity along x coordinate, $w \equiv \sqrt{g/z_0}$. By setting $t = T$ (T is the stepping time), the sagittal LIPM states at the end of step n : $(x_T^{(n)}, \dot{x}_T^{(n)})$, can be written as follows:

$$x_T^{(n)} = \cosh(wT)x_0^{(n)} + \frac{1}{w}\sinh(wT)\dot{x}_0^{(n)} - \frac{\tau_a}{mz_0 w^2}(\cosh(wT) - 1) \tag{4}$$

$$\dot{x}_T^{(n)} = w\sinh(wT)x_0^{(n)} + \cosh(wT)\dot{x}_0^{(n)} - \frac{\tau_a}{mz_0 w}\sinh(wT) \tag{5}$$

where $x_0^{(n)}$ and $x_T^{(n)}$ are the initial and final x coordinate positions of the COM at step n with respect to the stance foot, respectively; $\dot{x}_0^{(n)}$ and $\dot{x}_T^{(n)}$ are the initial and final linear velocities of the COM at step n , respectively.

The lateral plane has an equivalent LIPM as that of sagittal plane, but with the horizontal y axis pointing to the left side of the robot. The lateral LIPM states at the end of step n $(y_T^{(n)}, \dot{y}_T^{(n)})$, can be written:

$$y_T^{(n)} = \cosh(wT)y_0^{(n)} + \frac{1}{w}\sinh(wT)\dot{y}_0^{(n)} - \frac{\tau_{ay}}{mz_0 w^2}(\cosh(wT) - 1) \tag{6}$$

$$\dot{y}_T^{(n)} = w\sinh(wT)y_0^{(n)} + \cosh(wT)\dot{y}_0^{(n)} - \frac{\tau_{ay}}{mz_0 w}\sinh(wT) \tag{7}$$

where $y_0^{(n)}$ and $y_T^{(n)}$ are the initial and final positions of the COM at step n with respect to the stance foot, respectively; $\dot{y}_0^{(n)}$ and $\dot{y}_0^{(n)}$ are the initial and final linear velocities of the COM at step n , respectively.

3.3.2.2. Orbital energy as push detection

As mentioned earlier, orbital energy is chosen as the reliable parameter to detect a push. The orbital energy is obtained by multiplying the LIPM dynamics (with zero ankle torque) with $\dot{x}$ and integrating it:

$$\int \ddot{x} dx - \int \frac{g}{z_0} x dx = 0$$

$$\frac{1}{2}\dot{x}^2 - \frac{g}{2z_0}x^2 = \text{constant} \equiv E$$

Since the biped is represented by two decoupled LIPM, we have an orbital energy value for each plane in real-time. The orbital energies in the sagittal and lateral planes are calculated as follows [29]:

$$E_x(t) = \frac{1}{2}\dot{x}(t)^2 - \frac{1}{2}w^2 x(t)^2$$

$$E_y(t) = \frac{1}{2}\dot{y}(t)^2 - \frac{1}{2}w^2 y(t)^2 \qquad (8)$$

where $E_x(t)$ and $E_y(t)$ are the orbital energies for the sagittal and lateral plane, respectively. $\dot{x}(t)$ and $\dot{y}(t)$ are the sensory readings of the COM linear velocities at time t in the sagittal and lateral planes, respectively. $x(t)$ and $y(t)$ are obtained from the LIPM differential equation solutions (Eq. (2)).

In the push recovery problem formulation, we have mentioned that the push may come from any direction in the transverse plane. In our scheme that uses two decoupled LIPM, we need to determine the priority plane (either sagittal plane or lateral plane), which is the plane at which the effect of the disturbance on the LIPM is the most severe. This will affect the policy taken by the main controller to recover from the push. The decisions of the push recovery scheme will be aimed towards satisfying the constraints in the priority plane first, before it considers the non-priority plane.

One of the main reasons we choose orbital energy for the push detection is that it theoretically remains constant during normal walking with a constant desired LIPM states. However, due to dynamic complexity, sensory noise, and actual energy losses during impact, the orbital energy may fluctuate to some extent in a real biped during walking. Hence, for practical usage, we define orbital energy deviation error for each plane as follows:

$$E_x error(t) = \left| E_{dx} - E_x(t) \right|$$

$$E_y error(t) = \left| E_{dy} - E_y(t) \right| \tag{9}$$

where $E_x error(t)$ and $E_y error(t)$ are the orbital energy deviations from the desired orbital energy in the sagittal and lateral plane, respectively. The desired orbital energies are defined as:

$$E_{dx} = \frac{1}{2}\dot{x}_d{}^2 - \frac{1}{2}w^2 x_d{}^2$$

$$E_{dy} = \frac{1}{2}\dot{y}_{dr}{}^2 - \frac{1}{2}w^2 y_{dr}{}^2 \text{ (or } \frac{1}{2}\dot{y}_{dl}{}^2 - \frac{1}{2}w^2 y_{dl}{}^2 \text{)}$$

where E_{dx} and E_{dy} are the desired orbital energies in the sagittal and lateral planes, respectively. x_d and $\dot{y}_d$ are the desired states in the sagittal plane at support exchange, which will determine the normal walking step length and walking speed. y_{dr} , $\dot{y}_{dr}$ (for right foot swing phase); and y_{dl} , $\dot{y}_{dl}$ (for left foot swing phase) are the desired states in the lateral plane at support exchange. These states determined the step width and sway speed at the support exchange in the lateral plane. All of these desired state constants should be chosen based on the physical dimension of the biped. The orbital energy deviations will then be compared to threshold constants $E_x threshold$ and $E_y threshold$ for sagittal and lateral planes, respectively.

We proposed the push detection algorithm as follows:

Push Detection Algorithm

Input: sensory feedback $\dot{x}(t)$ and $\dot{y}(t)$, COM position $x(t)$ and $y(t)$ from Eq.(2).

Output: determine whether the biped is in *pushed state*, if yes, which plane is the priority plane

Step 1: Calculate the orbital energy in sagittal and lateral plane, based on Eq. (8).

Step 2: Calculate the orbital energy error in sagittal and lateral plane, based on Eq. (9).

Step 3: If $E_x error(t) > E_x threshold$ or $E_y error(t) > E_y threshold$, then go to step 4. Otherwise the biped is not pushed, go to step 1.

Step 4: The biped is in *pushed state*. If $E_x error(t) > E_y error(t)$, then the priority plane is sagittal plane. Otherwise priority plane is lateral plane.

This algorithm will run continuously during normal walking. When the biped enters a *pushed state*, a new stepping time will begin. The current biped state will be used as the initial condition for the *pushed state*.

3.3.3. Walking phase modification

Walking phase modification, which is the main and staple approach for the proposed push recovery strategy during walking, is presented as follows:

3.3.3.1. Concept

This section will discuss the concept of walking phase modification. For push recovery, it is important to understand the dynamics of bipedal walking. In terms of dynamics, bipedal walking cycle can be classified into two major phases: the acceleration phase and the deceleration phase. The deceleration phase occurs whenever the COM is approaching the stance foot, while the acceleration phase occurs whenever the COM is moving away from the stance foot.

When a biped is pushed, the orbital energy suddenly changed. Because of the unusual orbital energy level, the biped could not walk properly and it may fall if no action is taken. Fig. 12 shows a case example of a biped being pushed from behind. In Fig. 12, E_1 is the orbital energy at step n after the push, E_2 is the orbital energy at the step $n+1$ after the push, x_{acc} is the accelerating phase distance, and x_{dec} is the deceleration phase distance. Because the push is from behind, the orbital energy is suddenly increased (Fig. 12a). At the end of the normal stepping time, the biped has an excessive velocity and acceleration phase distance, which may cause it to fall (Fig. 12b). Therefore, it is the goal of push recovery to recover the orbital energy to the desired normal walking level (Fig. 12c).

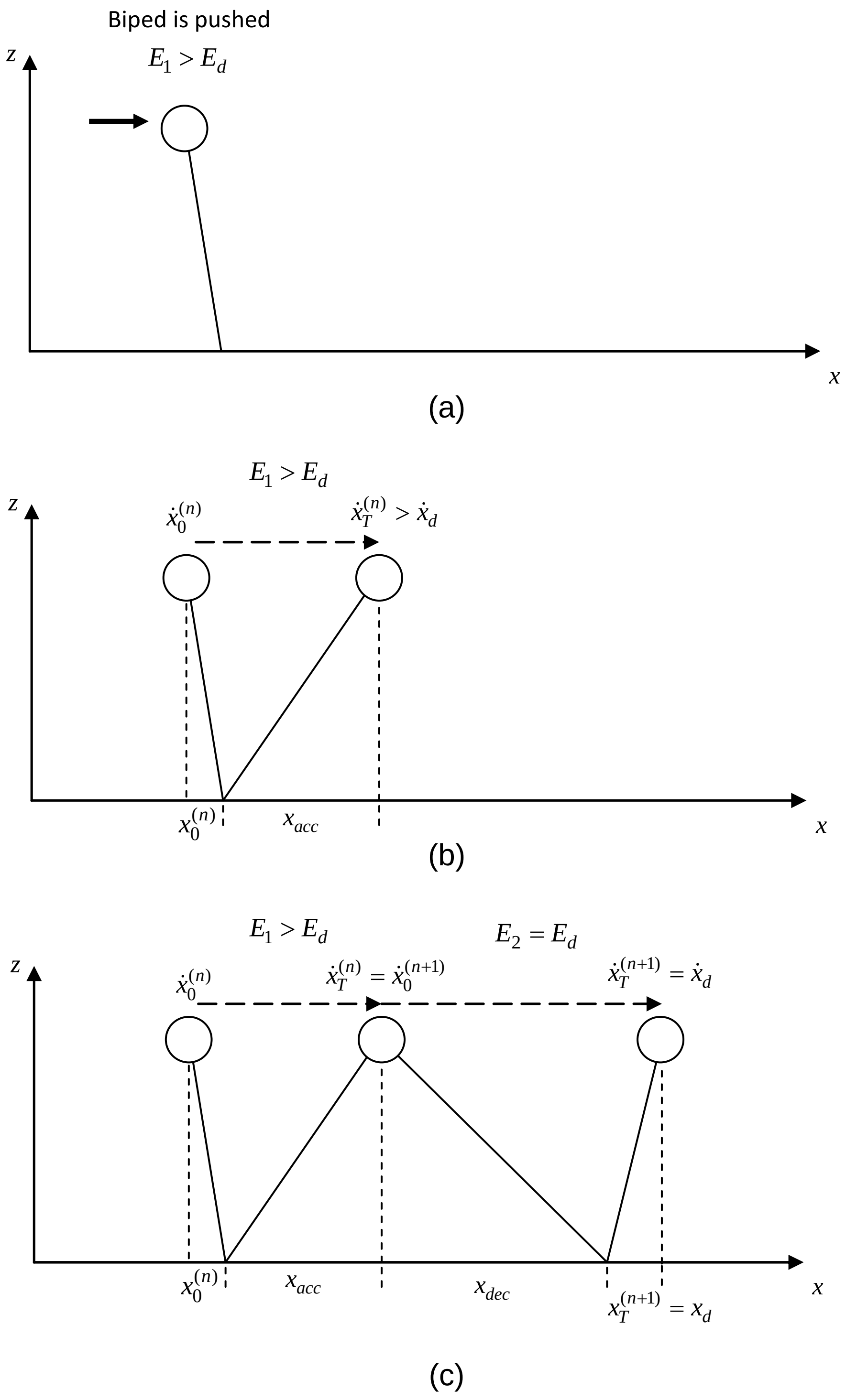


Figure 12: (a) A biped modeled with LIPM is pushed, the orbital energy is suddenly increased. (b) Because of the push, the biped may fall if no recovery action is taken. (c) To return to the desired orbital energy level, the biped needs to do push recovery.

In Fig. 12c, the orbital energy E_1 and E_2 are defined as follows:

$$E_1 = \frac{1}{2}(\dot{x}_T^{(n)})^2 - \frac{g}{z_0}x_{acc}{}^2$$

$$E_2 = \frac{1}{2}(\dot{x}_0^{(n+1)})^2 - \frac{g}{z_0}x_{dec}{}^2 \qquad (10)$$

Suppose there is no energy loss during stepping, we have $\dot{x}_T^{(n)} = \dot{x}_0^{(n+1)}$. Hence, the relation between E_1 and E_2 are as follows:

$$E_2 = E_1 + \frac{g}{z_0}x_{acc}{}^2 - \frac{g}{z_0}x_{dec}{}^2 \qquad (10)$$

It could be seen that the orbital energy E_2 depends on the orbital energy of the previous step E_1, the acceleration phase distance, and the deceleration phase distance. Therefore, the orbital energy could be recovered through modifying the proportion between the acceleration phase and the deceleration phase. We shall refer this scheme as the walking phase modification.

The acceleration phase could be modified through regulating the stepping time and ankle torque (Fig. 13). Then, based on the stepping time and ankle torque that has been determined, the foot placement will determine the most appropriate deceleration phase distance to bring the biped closer to its desired states. The next subsections will explain the implementation of the walking phase modification in detail.

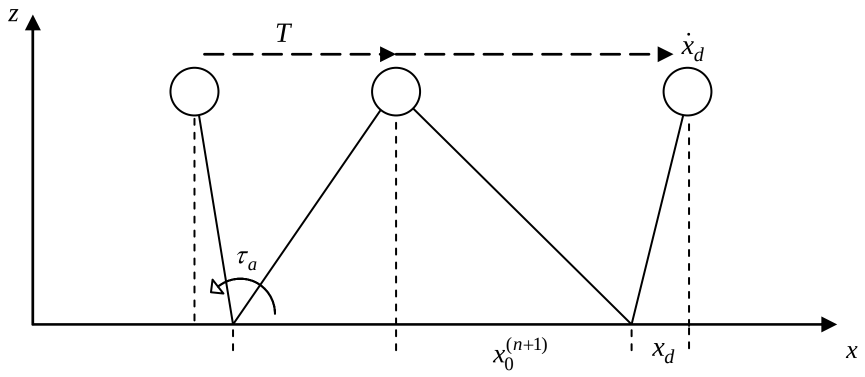


Figure 13: Stepping time T, ankle torque τ_a, and foot placement $x_0^{(n+1)}$ are the parameters used to modify the walking phase of the biped.

3.3.3.2. Modifying acceleration phase through control policy

In section 3.2, we have discussed the characteristics of the push recovery problem that a bipedal robot may be in. Among them, the magnitude of the push is the most critical aspect. Therefore, the capability of the biped to respond appropriately to wide range of push magnitude is an important factor for a push recovery strategy.

A bipedal robot, being a machine, requires a control policy in deciding its action. This control policy will serve as an intuitive guide in deciding the key actions taken to regulate the acceleration phase. The actions may not be unique, as a car driver could chose between decelerating his car smoothly or abruptly. However, as the push magnitude increases, there will be fewer alternatives to chose from. These alternatives are the consequence of the fact that a humanoid robot has some physical constraints. There will also be conditions that the biped has no other alternative but to prepare to fall, because the magnitude of the push is simply too great, and the biped has reach its constraint limits.

The constraints that are being considered are as follows: Consider a realistic humanoid robot whose legs are powered with servos. Each servo has a local position feedback that enables it to drive a particular joint with a limited velocity and torque. Several constraints that need to be considered:

- Constraint on the stepping reach

The first constraint is related to the limited reach of the biped's swing foot. The value of the constraint will depend on the biped's leg geometry and the constant COM height setting. In the implementation, it may be chosen to be lower than the maximum reach of the robot to avoid singularity and for practical safety considerations.

The swing foot reach in the sagittal plane has a constraint in forward and backward direction with respect to the COM. Suppose the biped is pushed from behind, the forward stepping constraint will be used as the stepping constraint. On the other hand, suppose the biped is pushed from front, the backward stepping constraint will be used as the stepping constraint. Similarly, the lateral plane also has its own stepping constraints with respect to the COM: the inward stepping constraint and the outward stepping constraint. Note that the inward stepping constraint is extremely limited due to the fact that the biped could not cross its legs.

- Constraint on the stepping time

Each servo that drives the swing leg joints has a velocity limit. Therefore, there exist a theoretical minimum stepping time for the swing foot to move from its current position to reach a particular constraint on the stepping reach. However, the swing foot does not necessarily move in the same trajectory for each step and it may not move with a constant velocity. Because of this difficulty, the minimum stepping time need to be estimated. There could be many ways to estimate the minimum stepping time. In this thesis, the minimum stepping time is estimated by comparing the approximate distance that must be travelled by the swing foot and the swing foot's maximum velocity.

- Constraint on the ankle torque

The last constraint is on the ankle torque, which is the maximum and minimum torque that can be delivered by the ankle joint system. The constraints value will depend on the servo and gearbox of the ankle joint.

The above constraints are considered in the formulation of the control policy. The motivation in formulating different actions for different initial LIPM states in a *pushed state* is to spend as little as energy as possible (i.e. battery energy). For example, using stronger ankle torque will consume more battery energy, and vice versa. Therefore, ankle torque will be used only when necessary, with the appropriate magnitude and direction. This feature is an advantage of the proposed approach. In a sense, the biped considers its options, and chooses the most efficient (least battery consumption) solution to recover itself. The proposed control policy and its various actions are shown in Fig. 14.

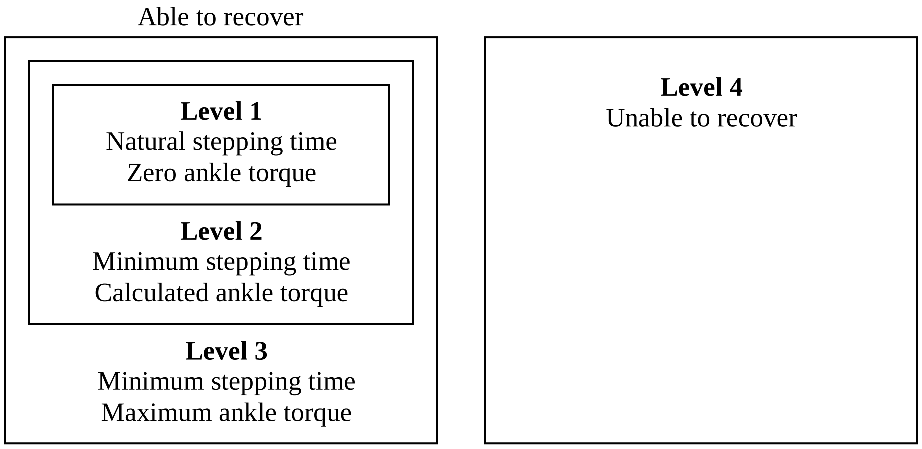


Figure 14: Venn diagram of the control policy. The sets correspond to the initial LIPM states in a *pushed state*. The initial states that lie within level 1-3 theoretically could be recovered with the corresponding actions. The initial states that lie within level 4 are exclusive from the other cases. Level 4 could not be recovered because the push magnitude is too great.

To be able to execute a particular action set, there are some particular conditions that must be met. Fig. 15 shows the conditions for each cases (while the control policy is applied to all conditions, an illustration in the sagittal plane when the push is from behind is given as a case example. In Fig. 15, T is the natural stepping time (i.e. the time required for the COM of an unactuated (zero torque) LIPM to reach a particular position in x axis). $T_{\lim}$ is the estimated minimum stepping time, T_n is the default normal walking stepping time (i.e. the walking step time when the biped is not pushed), τ_a is the calculated ankle torque, $\tau_{\lim}^{+}$ and $\tau_{\lim}^{-}$ are the maximum and minimum constraints on the ankle torques, and $x_{forward}$ is the forward constraint on the stepping reach).

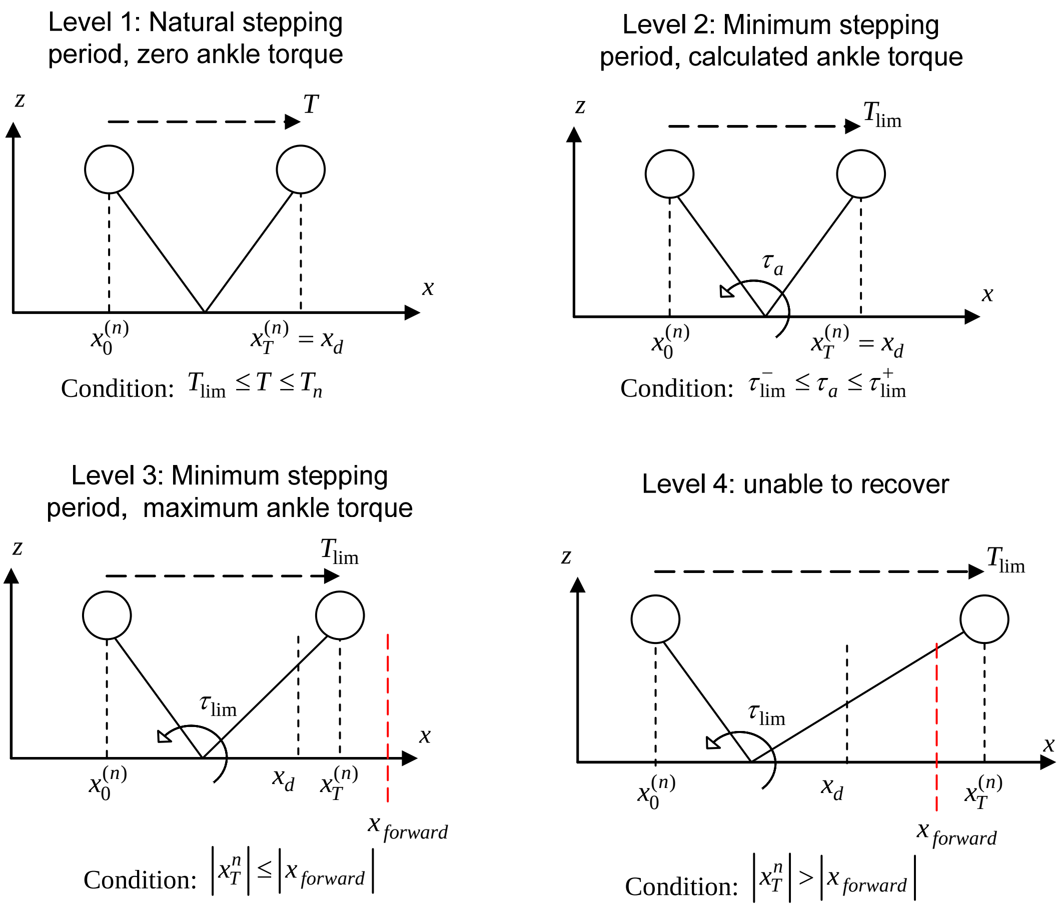

Level 1: Natural stepping period, zero ankle torque
z
T
x
$x_0^{(n)}$
$x_T^{(n)} = x_d$
Condition: $T_{\text{lim}} \le T \le T_n$
Level 2: Minimum stepping period, calculated ankle torque
z
T_{lim}
τ_a
x
$x_0^{(n)}$
$x_T^{(n)} = x_d$
Condition: $\tau_{\text{lim}}^- \le \tau_a \le \tau_{\text{lim}}^+$
Level 3: Minimum stepping period, maximum ankle torque
z
T_{lim}
τ_{lim}
x
$x_0^{(n)}$
x_d
$x_T^{(n)}$
$x_{forward}$
Condition: $\left|x_T^n\right| \le \left|x_{forward}\right|$
Level 4: unable to recover
z
T_{lim}
τ_{lim}
x
$x_0^{(n)}$
x_d
$x_{forward}$
$x_T^{(n)}$
Condition: $\left|x_T^n\right| > \left|x_{forward}\right|$

Figure 15(a-d): Control policy. The LIPM states at the beginning of a push state are used as a criterion to choose the stepping time and ankle torque.

The conditions are constructed intuitively as follows:

- Level 1: Natural stepping time, zero ankle torque (Fig. 15a)

  Condition: The unactuated LIPM will reach the desired COM position and the natural time for the unactuated LIPM to reach the desired COM position is within the allowed stepping time. This is the simplest of all the cases, in which the push will drive the biped to reach its desired COM position with a smaller stepping time compared to the normal stepping time. In this case, the COM will move to the desired position based on the initial states naturally.

- Level 2: Minimum stepping time, calculated ankle torque (Fig. 15b)

  Condition: The actuated LIPM with a calculated ankle torque could reach the desired COM position. In this case, the LIPM uses some ankle torque to modify its motion to reach the desired COM position. Minimum stepping time is chosen to minimize the torque requirement.

- Level 3: Minimum stepping time, maximum ankle torque (Fig. 15c)

  Condition: The actuated LIPM with maximum ankle torque could reach within constraint on the COM position. In this case, the biped uses the maximum ankle torque and minimum stepping time to bring the COM to the nearest position with the desired position. Although the COM may not end up in its desired position, the biped still has a chance to maintain walking. This is because as long as the COM is moving within the constraint on COM position, the deceleration phase distance could still be made more than the acceleration phase (Fig. 16).

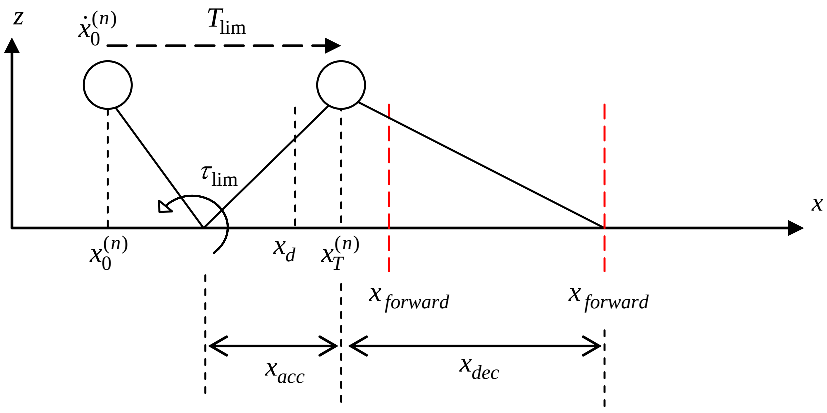


Figure 16: Level 3 is recoverable because as long as the COM is within the constraint of stepping reach, the deceleration phase distance could be made more than the acceleration phase. Hence, the biped still has a chance to recover from the push.

- Level 4: Unable to recover (Fig. 15d)

  Condition: The actuated LIPM with maximum torque exceeds constraint on the COM position. Note that the COM position is constrained using the same constraints with the stepping reach, as we assumed the biped must be able to decelerate more than it accelerates in order to recover from a push. This is the threshold where the magnitude of the push is considered to be too large to be recovered.

Suppose the biped has just entered *pushed state* and the sagittal plane is the priority plane, the control policy decides the actions as follows:

Control Policy Algorithm

Input: estimated minimum stepping time $T_{\lim}$, the LIPM state in sagittal plane $(x(t), \dot{x}(t))$.

Output: determine stepping time T and the necessary ankle torque in the sagittal plane τ_a .

Step 1: Use the LIPM state as the *pushed state* variables:

$$(x_p, \dot{x}_p) \leftarrow (x(t), \dot{x}(t))$$

Step 2: Estimate the natural stepping time for the COM to reach the desired position x_d . Calculated based on LIPM dynamics, by substituting the *pushed state* parameters and zero ankle torque to Eq. (4):

$$T \leftarrow \frac{1}{w}\ln\left(\frac{x_d + \sqrt{{x_d}^2 - \frac{1}{2}\left(x_p + \frac{\dot{x}_p}{w}\right)\left(x_p - \frac{\dot{x}_p}{w}\right)}}{\left(x_p + \frac{\dot{x}_p}{w}\right)}\right) \tag{12}$$

Step 3: Check whether the natural stepping time is within the allowed stepping time ($T_{\lim} \le T \le T_n$). If this condition is satisfied, then it is a level 1 push and the actions are: use natural stepping time, zero ankle torque. Otherwise use the minimum stepping time ($T \leftarrow T_{\lim}$) and go to step 4.

Step 4: Calculate the necessary ankle torque to bring the COM to reach the desired COM position x_d. Calculated by substituting $x_T^{(n)} = x_d$ to Eq. (4):

$$\tau_a \leftarrow (\cosh(wT)x_p + \frac{1}{w}\sinh(wT)\dot{x}_p - x_d)mz_0w^2\frac{1}{(\cosh(wT)-1)} \tag{13}$$

Step 5: Check whether the ankle torque is within the constraints on the ankle torque ($\tau_{\lim}^- \le \tau_a \le \tau_{\lim}^+$). If this condition is satisfied, then it is a level 2 push and the actions are: use minimum stepping time, calculated ankle torque. Otherwise, use the constraint on the ankle torque that has been exceeded as the ankle torque and go to step 6.

Step 6: Calculate the COM position at the end of the stepping time:

$$x_T^{(n)} \leftarrow \cosh(wT)x_p + \frac{1}{w}\sinh(wT)\dot{x}_p - \frac{\tau_{\lim}^-}{mz_0w^2}(\cosh(wT)-1) \tag{14}$$

Step 7: Check whether the COM position at the end of the minimum stepping time is within the constraint on the stepping reach ($\left|x_T^{(n)}\right| \le \left|x_{forward}\right|$). If this condition is satisfied, then it is a level 3 push and the actions are: use minimum stepping time, maximum ankle torque. Otherwise, prepare to fall because the biped could not decelerate more than it accelerates (it is a level 4 push).

Afterwards, the stepping time that has been decided in the priority plane will be used to calculate the necessary ankle torque in the non-priority plane. The next section will describe how the stepping time and ankle torques will be used as part of the input to modify the deceleration phase.

3.3.3.3. Modifying deceleration phase through foot placement determination

For the foot placement determination, it is important to understand the relation between foot placement and the LIPM states. With the LIPM initial states at step n, the stepping time, and the ankle torque as the input, the controller determines the foot placement, such that the LIPM final state at step $n+1$ can be closer towards the desired state. Fig. 17 shows the two successive steps of the walking pattern based on the LIPM approach:

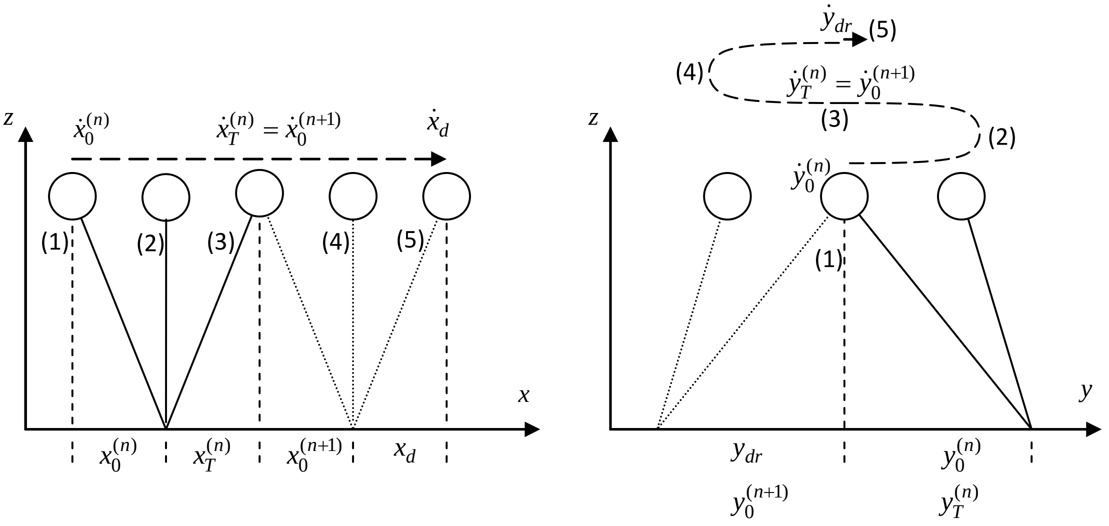


Figure 17: Two successive walking pattern based on the LIPM approach are considered. Suppose the biped starts from a right foot swing phase. Left figure shows the COM

motion in the sagittal plane (swing legs are not shown in the figure). Right figure shows the COM motion in the lateral plane (dotted line indicates right leg). The number (1)-(5) indicates the motion sequence. The dashed arrows indicate the COM motion trajectory in the horizontal axis.

In the sagittal plane, the desired state for each support exchange is $(x_d, \dot{x}_d)$ (for walking forward $x_d > 0, \dot{x}_d > 0$). Suppose the LIPM is at the beginning of step n, x_d is the desired COM position at the support exchange at the end of step $n+1$. In the sagittal plane, x_d will be half of the desired step length. $\dot{x}_d$ is the desired velocity that should be achieved at every support exchange instance, during a normal walking with a constant normal walking stepping time.

The foot placement in the sagittal plane is determined as follows: Let the error index at the end of step $n+1$ be defined as [32]:

$$N \equiv (x_d - x_T^{(n+1)})^2 + (\dot{x}_d - \dot{x}_T^{(n+1)})^2 \tag{15}$$

Assuming that the LIPM could return to the desired states at step $n+1$; and moves with zero ankle torque and default stepping time T_n:

$$x_T^{(n+1)} = \cosh(wT_n)x_0^{(n+1)} + \frac{1}{w}\sinh(wT_n)\dot{x}_0^{(n+1)} \tag{16}$$

$$\dot{x}_T^{(n+1)} = w\sinh(wT_n)x_0^{(n+1)} + \cosh(wT_n)\dot{x}_0^{(n+1)} \tag{17}$$

Substituting Eq. (16) and Eq. (17) into Eq. (15), the foot placement $x_0^{(n+1)}$ that minimizes the error index can be obtained by solving

$$\frac{dN}{dx_0^{(n+1)}} = 0$$

Hence, the foot placement:

$$x_0^{(n+1)} = \frac{\cosh(wT_n)(x_d - {}^1\!/_w \sinh(wT_n)\dot{x}_0^{(n+1)}) + \sinh(wT_n)w(\dot{x}_d - \cosh(wT_n)\dot{x}_0^{(n+1)})}{\cosh^2(wT_n) + w^2\sinh^2(wT_n)} \qquad (18)$$

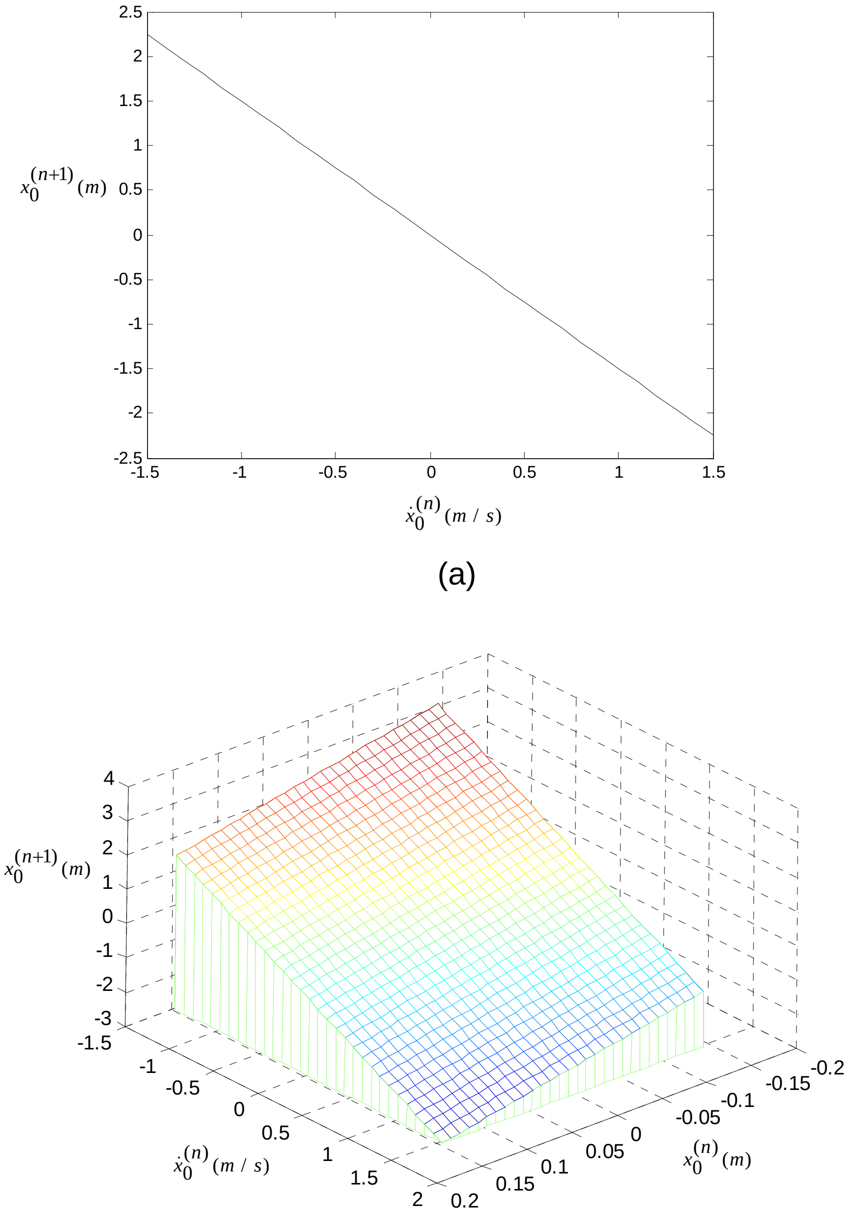
With the assumption of no energy loss during stepping, we have $\dot{x}_0^{(n+1)} = \dot{x}_T^{(n)}$. $\dot{x}_T^{(n)}$ can be calculated from the initial states of the LIPM as the inputs (Eq. (5)).

Similar with the sagittal plane, the foot placement in the lateral plane during right foot swing phase is as follows:

$$y_0^{(n+1)} = \frac{\cosh(wT_n)(y_{dr} - {}^1\!/_w \sinh(wT_n)\dot{y}_0^{(n+1)}) + \sinh(wT_n)w(\dot{y}_{dr} - \cosh(wT_n)\dot{y}_0^{(n+1)})}{\cosh^2(wT_n) + w^2\sinh^2(wT_n)} \qquad (19)$$

The desired states during right foot swing phase $(y_{dr}, \dot{y}_{dr})$ $(y_{dr} > 0, \dot{y}_{dr} > 0)$ should be changed to $(y_{dl}, \dot{y}_{dl})$ $(y_{dl} < 0, \dot{y}_{dl} < 0)$ during right foot support phase.

To comprehend how the foot placement determination works, we present a simple LIPM simulation (no constraint on the stepping reach, zero torque, and constant stepping time) in the sagittal plane as follows: Suppose a push occurs to the LIPM in sagittal plane during step n. Depending on the initial state, Eq. (18) will decide the foot placement such that at the end of step $n+1$ the LIPM states are restored to its desired states. Fig. 18a shows the foot placement (based on Eq. (18)) when the biped is stepping on the spot (the desired states and the initial positions are zeros: $x_d = 0m$, $\dot{x}_d = 0m/s$, $x_0^{(n)} = 0m$). It is observed that the LIPM will decide to take a longer step forward when the initial velocity is larger, and takes a longer step backward when the initial velocity is smaller. Suppose a push occurs while the biped is walking forward (let the desired states: $x_d = 0.145m$ and $\dot{x}_d = 0.651\text{m/s}$), the initial condition of the LIPM at the moment of push will also affect the decision of the foot placement (Fig. 18b). This simple experiment demonstrates the basic foot placement determination in order to restore the LIPM states to its desired states. An advantage of this approach is that it has a lower computation load compared to controllers that needs to preview several stepping reference ahead to determine the COM trajectory [e.g. 36, 65].



(a)

(b)

Figure 18: (a) Relation between foot placement decision $x_0^{(n+1)}$ (m) and initial velocity $\dot{x}_0^{(n)}$ (m/s) when $x_0^{(n)} = 0$ (m). Note that negative sign of $x_0^{(n+1)}$ means the LIPM is stepping forward ($x_0 \equiv x_{COM} - x_{foot}$). (b) A surface depicting the relation between the foot placement $x_0^{(n+1)}$ with the LIPM initial states $(x_0^{(n)}, \dot{x}_0^{(n)})$ during walking forward.

In the simple simulation above, the LIPM is always able to return to its desired states after taking a single step because we made an impractical assumption that the LIPM has an infinite leg length that can reach to any point. For the real implementation, the constraint on the stepping reach will limit the foot placement decision. Hence, the algorithm to determine foot placement in the sagittal plane is as follows:

Foot Placement Algorithm

Input: stepping time T, ankle torque (τ_a) from *Control Policy Algorithm*, *pushed state* variables $(x_p, \dot{x}_p)$.

Output: determine the foot placement in sagittal plane $x_0^{(n+1)}$

Step 1: Calculate the COM velocity at the end of step n using Eq. (5), using stepping time and ankle torque from Control Policy Algorithm and *push state* parameters.

$$\dot{x}_T^{(n)} \leftarrow w\sinh(wT)x_p^{(n)} + \cosh(wT)\dot{x}_p^{(n)} - \frac{\tau_a}{mz_0 w}\sinh(wT) \tag{20}$$

Step 2: Calculate the foot placement using Eq. (18):

$$x_0^{(n+1)} \leftarrow \frac{\cosh(wT_n)(x_d - {}^{1}\!/_{w}\sinh(wT_n)\dot{x}_T^{(n)}) + \sinh(wT_n)w(\dot{x}_d - \cosh(wT_n)\dot{x}_T^{(n)})}{\cosh^2(wT_n) + w^2\sinh^2(wT_n)} \tag{21}$$

Step 3: Verify the foot placement is within the constraints on the stepping reach. If this condition is satisfied, use that foot placement. Otherwise, use the constraint on the stepping reach that has been exceeded as the foot placement.

Similarly, the foot placement in the lateral plane is found by using the lateral plane parameters. The Foot Placement Algorithm assumes that at the end of step $n+1$ the biped's COM could return to its desired states. However, if the foot placement is beyond the constraint on the stepping reach, the biped will have to step to the furthest stepping position, and then reexamine the states at the beginning of the next step. The constraint on the stepping reach is the largest possible deceleration distance to be had. This is the reason why in *Control Policy Algorithm*, the COM must not have an acceleration distance more than the constraint on the stepping reach.

In summary: the stepping time selection and ankle torque chosen by the control policy has modified the acceleration phase. The foot placement determination has modified the deceleration phase. All of these modification are done at the moment the biped enters *pushed state*, which forms the push recovery gait. The next section presents the compensators for this gait.

3.3.4. Local joint compensator

The local joint compensator is an additional controller to improve the overall robustness of the bipedal walking. The purpose of the local joint compensator is to compensate small disturbances that do not trigger the biped into a *pushed state*, and to bring the biped to a steady normal walking once the *pushed state* has been recovered. It will slightly modify the ankle joint according to sensory feedback information. The sensory information used is the body posture angle in sagittal and lateral plane, and a simple force feedback to sense whether a foot has landed on the ground.

- Tilt compensator

The lateral dynamic balance is crucial to maintain walking. Fig. 19 shows the two types of the tilt over cases that may occur due to the disturbance. The lateral stabilizer will modify the ankle joint of the stance foot in a series of repeated short duration modifications to effectively counter the tilting motion. The modifications will make the ankle to rotate in the opposite direction of the error in order to restore upright posture. The amount of modification will depend on the posture deviation angle at the lateral plane.

$\phi_y(t)=0$ means that the body is upright, and $\phi_y(t)>0$ indicates an inward tilt over case. The inward-tilt-over case is treated in a more sensitive manner since it may lead to the swing foot landing on the ground prematurely. The modification value is determined as follows:

$$M_{lar}(t)=\frac{1}{2}\sigma\left(1+\sin\left(\frac{2\pi t}{T_R}-\frac{\pi}{2}\right)\right) \qquad (0<t<T_R) \qquad (22)$$

where $M_{lar}(t)$ is the left ankle roll angle modification, which will be added to the stance foot ankle roll angle. σ is the modification value obtained from experimentation (Table 1), and T_R is the ankle roll compensation duration.

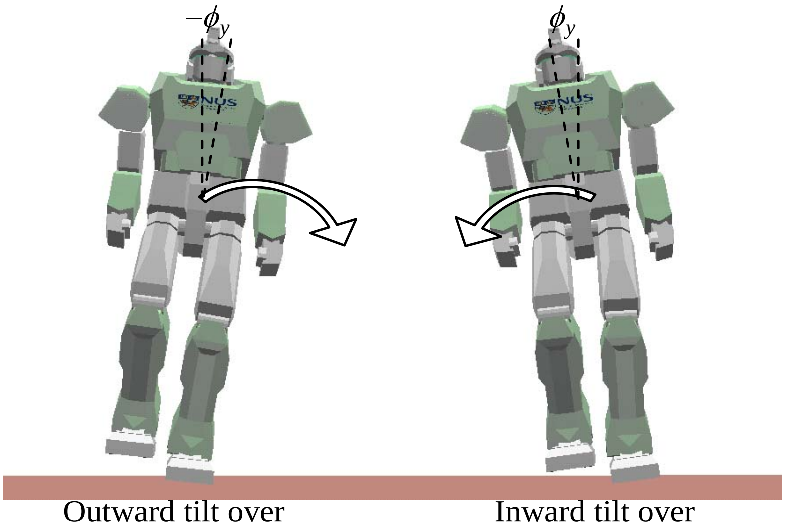


Figure 19: Lateral plane tilt over cases.

Table 1**:** Lateral tilt compensation value

Posture angle $\phi_y(t)$ (rad)	Modification value R (rad)
$\phi_y(t) < -0.08$	$\sigma = -0.09$
$-0.08 < \phi_y(t) < -0.04$	$\sigma = -0.07$
$-0.04 < \phi_y(t) < -0.02$	$\sigma = -0.05$
$-0.02 < \phi_y(t) < 0.01$	$\sigma = 0$
$0.01 < \phi_y(t) < 0.02$	$\sigma = 0.03$
$0.02 < \phi_y(t) < 0.03$	$\sigma = 0.05$
$\phi_y(t) > 0.04$	$\sigma = 0.07$

- Foot landing compensator

In the case of a strong push, the biped may be tilted towards the direction of the push. In the sagittal plane, where the distance between the toes and ankle is relatively large, even a small forward tilt may cause the toes to hit the ground prematurely. To compensate this, a swing leg landing compensation is developed.

The compensator works as follows: when the swing foot has not touch the ground, the ankle of the swing foot is modified to make the swing foot always parallel with respect to the horizontal axis.

$$q_{rap}(t) = -\phi(t) \qquad (23)$$

where $q_{rap}(t)$ is the right ankle pitch angle, $\phi(t)$ is the body posture angle in sagittal plane, $\phi(t) = 0$ means that the body is upright, and $\phi(t) > 0$ indicates a forward tilt case. Then, when the foot has landed relatively flat with the ground, a counter motion of the ankle is applied for a short duration to return the ankle joint to be perpendicular to the trunk.

$$q_{rap}(t) = -\phi_m + \frac{t_p}{T_P}\phi_m \qquad (0 < t_p < T_P) \quad (24)$$

where T_P is the duration for the ankle counter motion. ϕ_m is the body posture angle when the swing foot touches the ground. This counter motion effectively helps the biped’s posture to be upright again. Similar strategy is also applied to the ankle roll angle of the swing foot.

3.3.5. Overall strategy

Based on the considerations and specific controllers that we have described in the previous sections, the complete overall strategy is built. The overall strategy will be used both on a normal walking and *pushed state*. Fig. 20 shows the schematic of the flowcharts of the overall strategy. At every sampling time, a sensory feedback provides the input for the push detection. If the push detection does not detect any push, the normal walking will proceed with the default stepping time value and the biped walks as an unactuated (zero torque) LIPM.

If a push is detected at any sampling time, the biped enters *push state*, and the LIPM states at that sampling time will be the *pushed state* parameters. Next, the minimum stepping time is estimated and the control policy determined the stepping time and the ankle torques. Afterwards, the swing foot placement is decided to bring the LIPM closer to its desired states. All of these gait determination is executed at the same sampling time. Then, while the gait is executed during the rest of the stepping time, the local joint compensator works to maintain walking by keeping the swing leg to land appropriately and restoring the biped's posture to be upright.

Once the biped completed a step, the whole algorithm is iterated again. If the orbital energy error is still above the threshold, the biped is still in a *pushed state*. The *push state* parameters are updated with the state of the LIPM at the beginning of this new step. Hence, the biped will continue to be in a *push state*, until the orbital energy error has been decreased to an acceptable level.

The iteration of the proposed method is a key strategy to compensate for the dynamic inaccuracy and energy losses during stepping. The *pushed state* parameters, which are updated at the beginning of a new step, are the resultant of all the actual dynamics that had occurred in the previous step. In a sense, the algorithm has effectively taken into account the previous step dynamics complexities to formulate a push recovery gait, without actually modeling it in the LIPM.

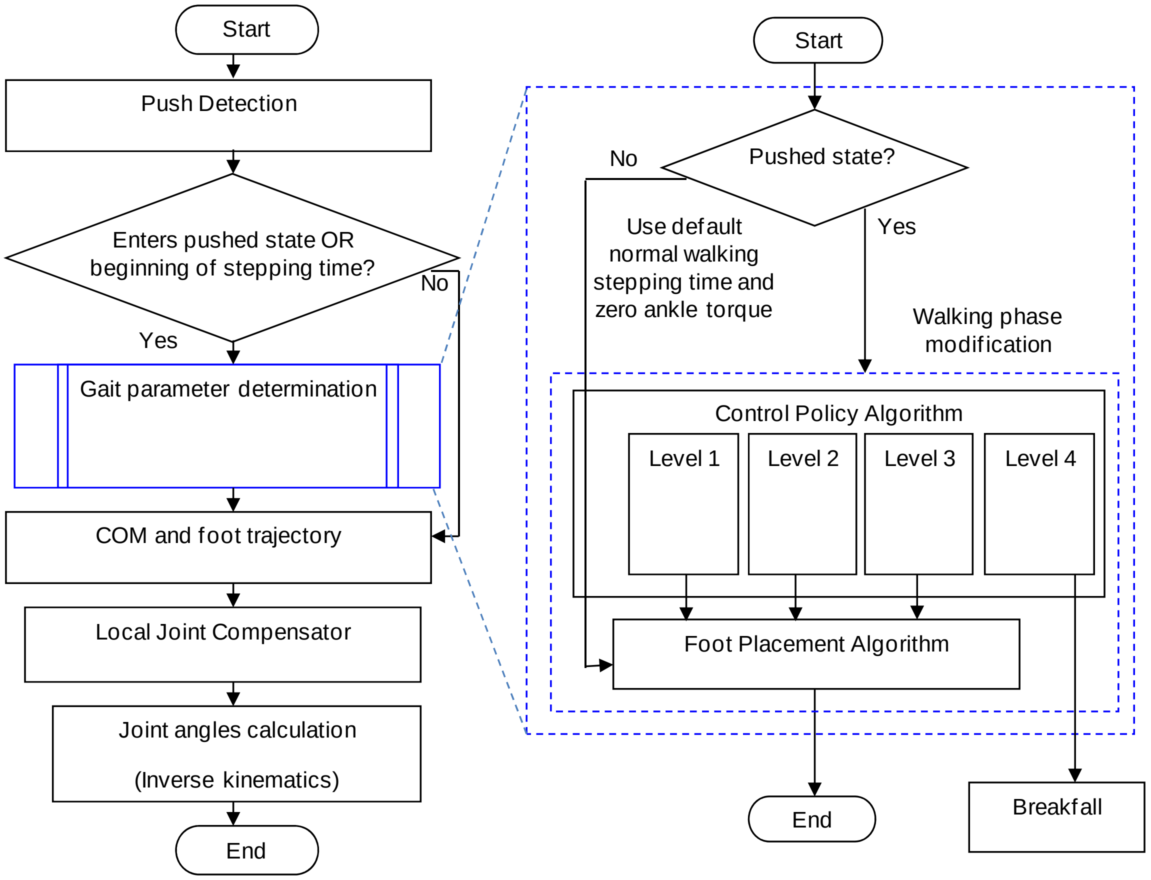


Figure 20: Overall strategy. The flowchart on the left is iterated at every sampling time. The gait parameter determination process (flowchart on the right) is conducted at the moment the biped enters *pushed state* or at the beginning of any stepping time.

3.4. Push recovery experiments with realistic humanoid robot model in dynamic simulation

To verify and evaluate the effectiveness of the proposed push recovery controller, a thorough push recovery during walking experiments in dynamic simulation is conducted.

The push recovery controller is applied into a realistic humanoid robot model with distributed mass and inertia. Then, the push recovery during walking capability of the humanoid robot is evaluated.

3.4.1. Humanoid robot model

The LIPM are used as a tool to represent a simulated humanoid robot model developed in Webots (Fig. 21). Webots is simulator software that could simulate the dynamics of a physical humanoind robot and its environments. The robot model stands 1.7, weighs 86.6 Kg (distributed mass with inertias), and has 6 DoF in each limb. In the simulation, it is equipped with velocity sensor at the pelvis. Table 2 lists some of the key parameters of the humanoid robot.

Table 2**:** Simulated humanoid robot parameters

Total Mass	86.6 Kg
Pelvis + Torso + Head Mass	4.53 Kg
Each Arm Mass	3.44 Kg
Each Leg Mass	18.6 Kg
Leg Length	0.95 m
Standing Center of Mass Height	0.85 m
Foot Length	0.34 m
Foot Width	0.14 m

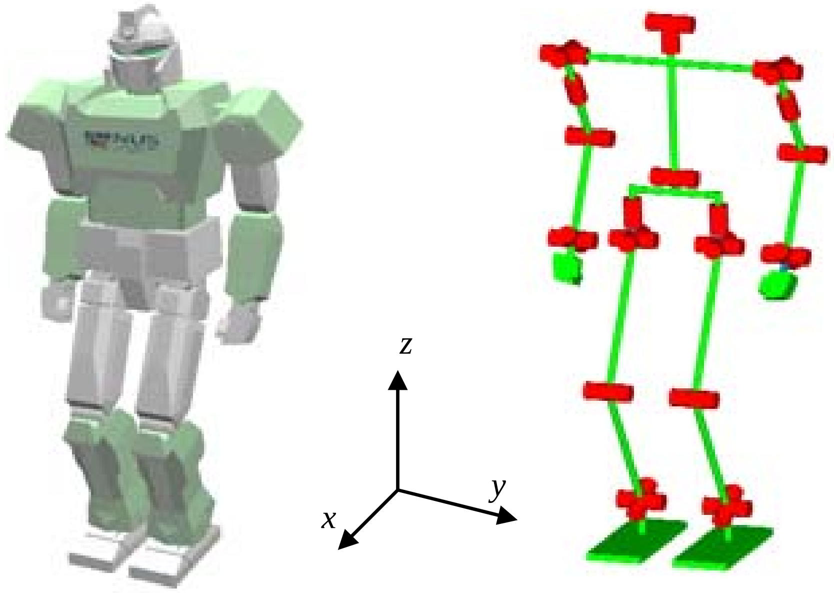


Figure 21: The simulated humanoid robot model developed in Webots and its joint configuration.

3.4.2. Push recovery experiments

In this section, a series of push recovery experiments is presented. First, we will test the performance of the overall strategy to handle pushes coming from the sagittal and lateral planes independently. Then, we will demonstrate that our overall strategy could also work for pushes that comes from arbitrary directions.

- Performance Evaluation

The performance tests will be evaluated while the biped is stepping on the spot and walking forward. The maximum impulse that can be withstood (the biped does not fall after some reasonable observation time) without any feedback will be compared to the maximum impulse that can be recovered by the push recovery scheme, at a particular walking phase.

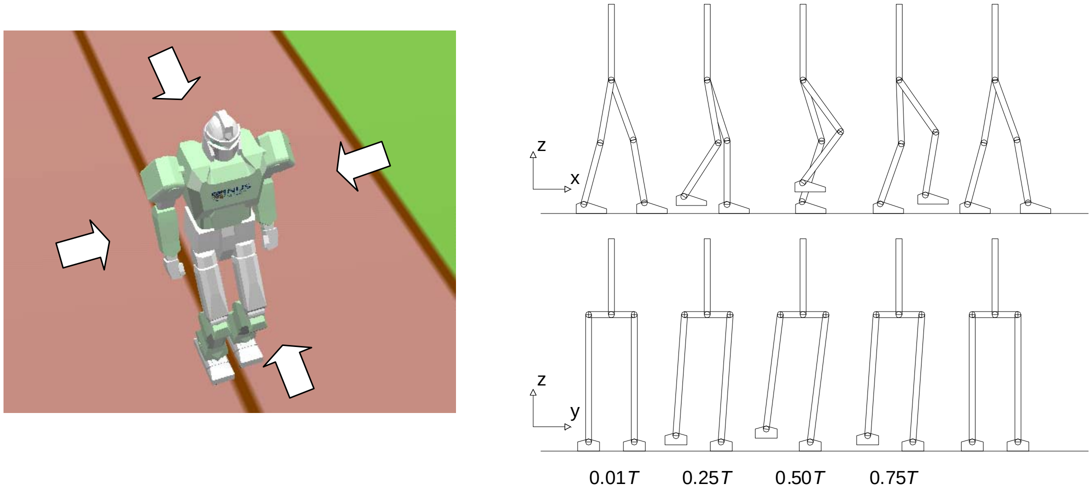
The impulse is calculated from a force sensor attached at the tip of the pusher ball. The impulse is calculated as:

$$imp_{1\to 2} = \int_{t_1}^{t_2} F dt$$

where F is the measured force at sampling time $t_1 < t < t_2$. The impulse information is displayed for our observation purpose only; it is not used in any way in the push recovery strategy.

Some key simulation parameters and constraints that we used are as follows: Sampling time: $0.008s$; desired states in the lateral plane: $y_{dr} = 0.095$ m, $\dot{y}_{dr} = 0.296$ m, $y_{dl} = -0.095$ m, $\dot{y}_{dl} = -0.296$ m; desired orbital energies: $E_{dx} = 0.065$ m^2/s^2, $E_{dy} = -0.019$ m^2/s^2; normal walking stepping time $T_n = 0.64$ s. desired states in sagittal plane for walking forward: $x_d = 0.145$ m, $\dot{x}_d = 0.651$ m/s, for stepping on the spot: $x_d = 0$ m, $\dot{x}_d = 0$ m/s. The constraint values are as follows: stepping reach $x_{forward} = 0.2$ m, $x_{backward} = -0.2$ m, $y_{outward} = 0.18$ m, $y_{inward} = 0.094$ m; estimated minimum stepping time $T_{\lim} = 0 - 0.4$ s (depending on swing foot position); ankle torque $\tau_{\lim}^{+} = 30$ Nm, $\tau_{\lim}^{-} = -30$ N/m. The biped's linear momentum during walking forward is about 27 - 44 Ns, with forward linear velocity of 0.3-0.5 m/s. These parameters and constraints will be kept constant throughout the tests.

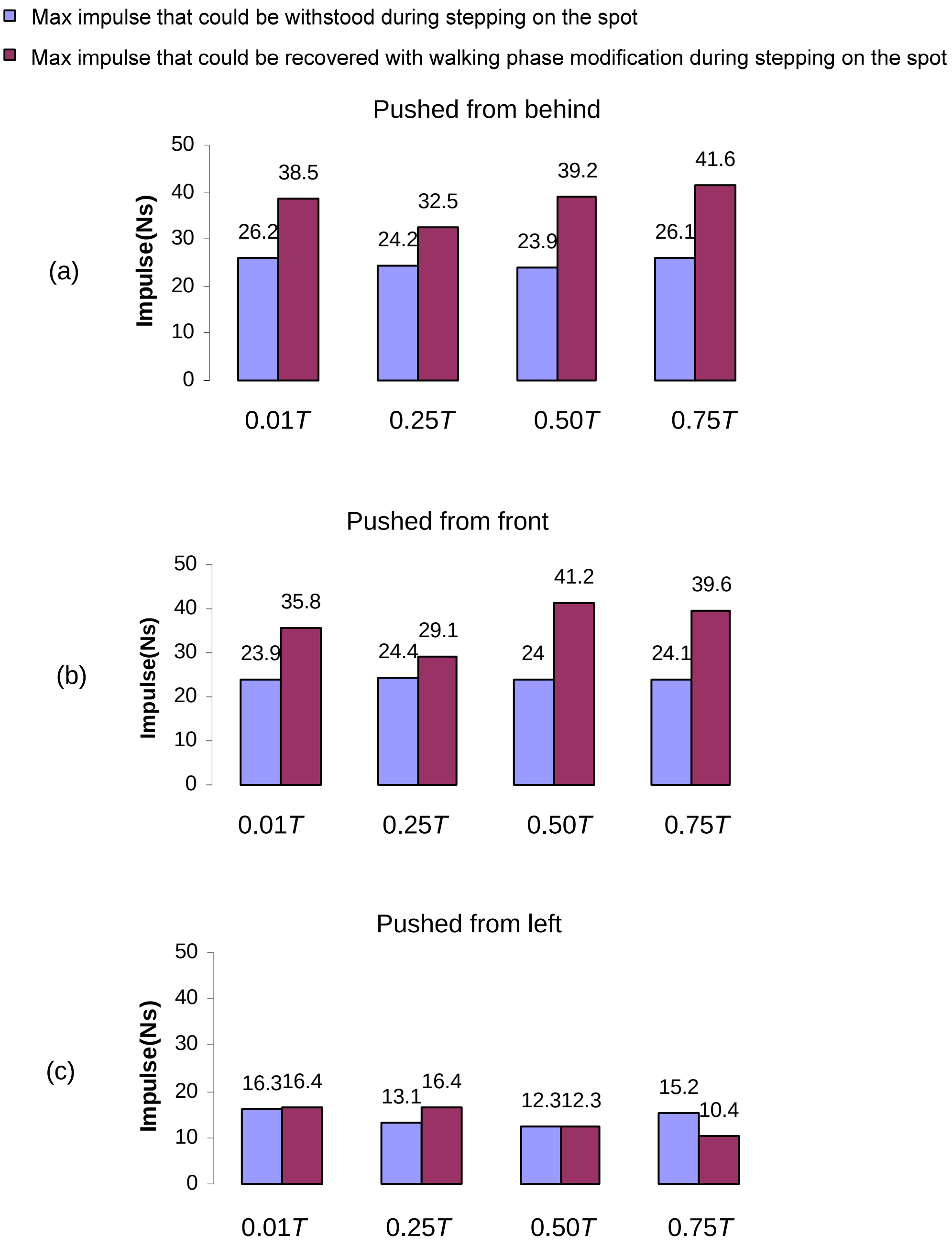
The biped will be pushed from four directions: behind, front, left, and right (Fig. 22a). To evaluate the performance in each direction without exhaustively testing it at every possible sampling time, we choose four test points which represent the nearby sampling points in bipedal robot walking cycle. Fig. 22b shows a stick walking figure representing a walking biped in the right foot swing phase and the corresponding test points. The test points are at $T_k = 0.01T_n$, $T_k = 0.25T_n$, $T_k = 0.5T_n$, and $T_k = 0.75T_n$. These test points represents acceleration phase ($T_k = 0.5T_n$, and $T_k = 0.75T_n$) , deceleration phase ($T_k = 0.01T_n$, and $T_k = 0.25T_n$) , double support phase ($T_k = 0.01T_n$), and single support phase ($T_k = 0.25T_n$, $T_k = 0.5T_n$, and $T_k = 0.75T_n$).



(a) (b)

Figure 22(a-b): Performance evaluation. Four push directions are applied: behind, front, left, and right. For each direction, the pushes occur at four different timings during right foot swing phase.

The performance test results can be seen in Fig. 23 and Fig. 24.


Max impulse that could be withstood during stepping on the spot
Max impulse that could be recovered with walking phase modification during stepping on the spot
Pushed from behind
(a)
Impulse(Ns)
50
40
30
20
10
0
26.2
38.5
24.2
32.5
23.9
39.2
26.1
41.6
0.01*T*
0.25*T*
0.50*T*
0.75*T*
Pushed from front
(b)
Impulse(Ns)
50
40
30
20
10
0
23.9
35.8
24.4
29.1
24
41.2
24.1
39.6
0.01*T*
0.25*T*
0.50*T*
0.75*T*
Pushed from left
(c)
Impulse(Ns)
50
40
30
20
10
0
16.3
16.4
13.1
16.4
12.3
12.3
15.2
10.4
0.01*T*
0.25*T*
0.50*T*
0.75*T*

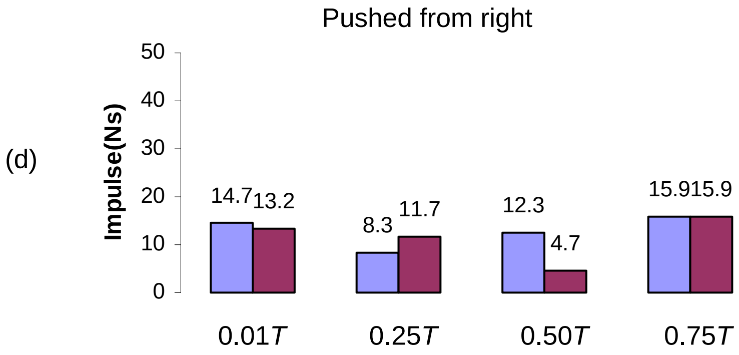
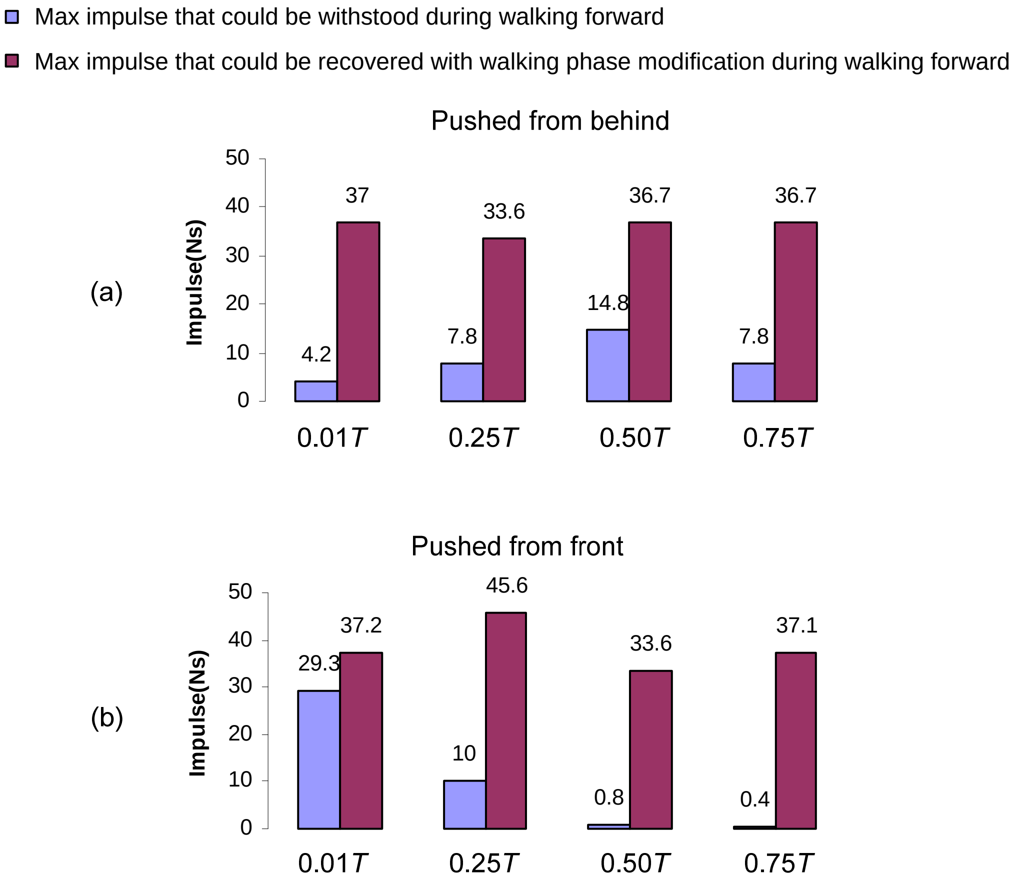


Figure 23(a-d): Performance evaluation when the biped is stepping on the spot, at the right foot swing phase.



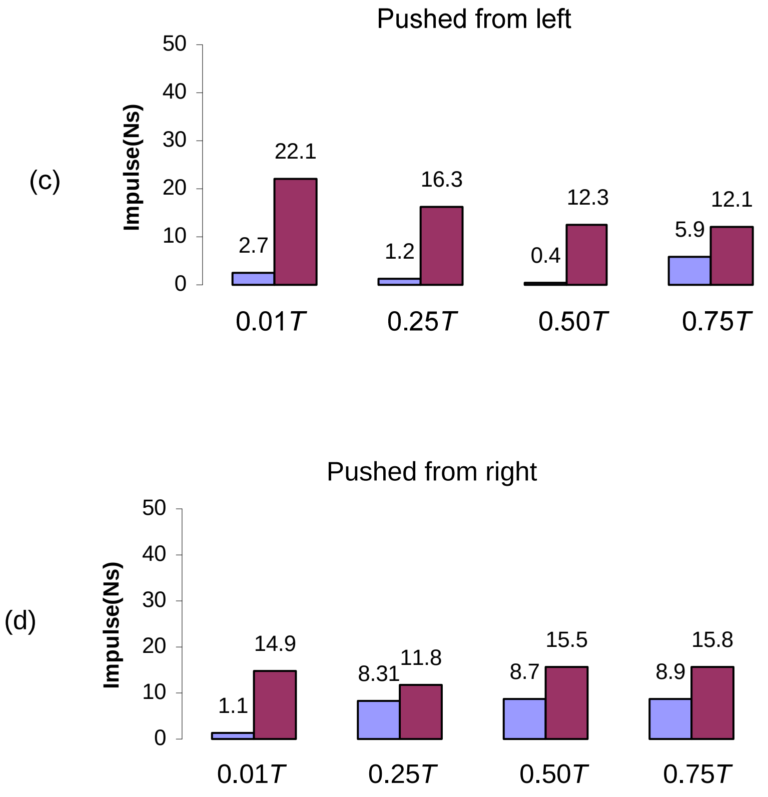


Figure 24(a-d): Performance evaluation when the biped is walking forward, at the right foot swing phase.

The results that show the peak magnitude that can be withstood without any push recovery scheme could give us insights regarding the most robust cases and the critical cases in bipedal walking disturbance rejection. As can be seen in Fig. 23 and Fig. 24, the biped is more able to withstand disturbance when it is stepping on the spot compared to when it is walking forward. Note that as the disturbance magnitude is closer to the peak magnitude, we found that the biped's foot has started to rotate with respect to the ground towards one side for a while, before swaying to the other side. Although the biped sways from one side to the other quite heavily, it does not fall at these peak magnitudes. This confirms the argument that maintaining the ZMP inside the support polygon is not a necessary condition to maintain walking. Since our experiment is conducted during right foot swing phase, the biped is most prone to disturbances from the right side.

The results that indicate the peak magnitude that can be recovered by our push recovery scheme shows that the biped is generally more able to handle disturbance in the sagittal plane. This is because the constraint on the stepping reach in the sagittal plane is considerably larger than that of lateral stepping. Another reason is because the foot length is also considerably longer compared to the foot width, therefore the frontal area has more stability margin.

Overall, compared to the results without the push recovery scheme, our approach seems to be highly effective to handle disturbance when the biped is walking forward. An interesting limitation of our push recovery scheme is observed. While the biped is stepping on the spot, there are some cases of push from the lateral side where the performance is close or lower than the results without push recovery scheme (e.g when pushed from the right). This is due to the fact that the biped could not cross its legs. The foot obstruction cases occurred in the lateral plane, where the biped’s feet are obstructing each other every couple of steps, which limit the biped’s capability to catch itself. Fast stepping times during these cases may worsen the performance because the biped topples when it tries to quickly lift up the stance foot.

- Push recovery from arbitrary pushes

To verify the general effectiveness of the overall push recovery strategy, we will show some push recovery from several pushes with various magnitude and direction. In the first experiment, four subsequent pushes are applied when the biped is stepping on the spot.

Table 3 shows the directions, magnitude, and the timing of the pushes. In table 3, α is the direction of the push in the transverse plane, with respect to x axis. imp is the impulse, and T_k is the walking phase when the push occurred. The linear velocity profile of the biped in x and y axis recorded from the experiment can be seen in Fig. 25. Similarly, four subsequent pushes are applied while the biped is walking forward. Table 4 shows the specifications of the pushes. The recorded velocity profile can be seen in Fig. 26. Note the sudden velocity jumped at the moment of pushes. Then, the biped manages to recover the velocity to normal level.

Table 3**:** Push specifications, applied when the biped is stepping on the spot

First	Second	Third	Fourth
$\alpha = -3$ rad $imp = 35.9$ Ns $T_k = 0.03T_n$	$\alpha = 1.1$ rad $imp = 29$ Ns $T_k = 0.46T_n$	$\alpha = -2.5$ rad $imp = 21$ Ns $T_k = 0.81T_n$	$\alpha = 2.9$ rad $imp = 34.6$ Ns $T_k = 0.06T_n$

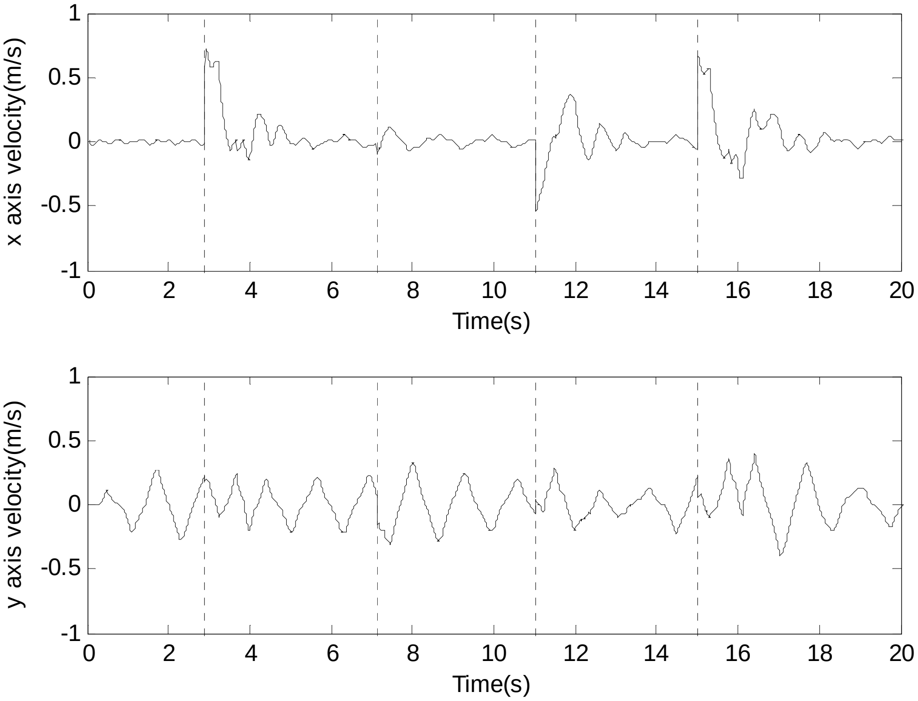


Figure 25: The velocity profile of the biped recorded from the experiment, where 4 subsequent pushes are applied while biped is stepping on the spot. The dotted vertical lines are the moment of the pushes.

Table 4**:** Push specifications, applied when the biped is walking forward

First	Second	Third	Fourth
$\alpha = 2.9$ rad $imp = 31.8$ Ns $T_k = 0.48T_n$	$\alpha = -1.2$ rad $imp = 11.2$ Ns $T_k = 0.72T_n$	$\alpha = 2.4$ rad $imp = 13$ Ns $T_k = 0.65T_n$	$\alpha = -0.13$ rad $imp = 35$ Ns $T_k = 0.81T_n$

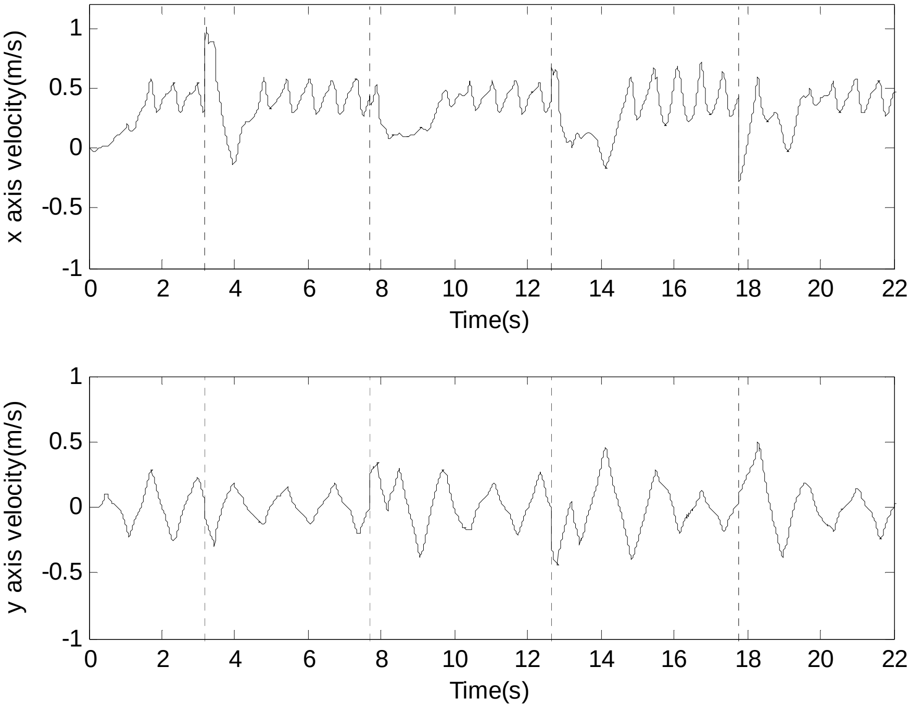


Figure 26: The velocity profile of the biped where 4 subsequent pushes are applied while biped is walking forward. The dotted vertical lines are the moment of the pushes.

From these experiments, we have demonstrated the ability of the overall controller that is used for pushes that comes from various directions. Overall, the local joint compensator and our approach to reexamine the biped’s state at the beginning of every step work well to compensate for energy losses and some of the discrepancies that may occur because of the difference between the LIPM and the realistic biped.

3.5. Discussion

In this section, we will share the insights that we learnt in developing the push recovery controller. In the performance evaluation that is presented in the previous section, we should not focus at the exact peak impulse magnitude of the push because it is relative to the parameters of our biped and subjective to the numerical inaccuracy of the Webots dynamic simulation.

From our observation during experimentation with a simulated realistic biped, the peak performance of the push recovery scheme is not only determined by the overall algorithm, but also by other factors not included in the LIPM. When the stepping position limit is large, the biped is theoretically more able to recover from a hard push before the control policy (*Control Policy Algorithm*) decides that the biped has no chance to recover from the push. However, as the step length becomes larger, the discrepancy between the LIPM and the realistic biped becomes more prominent as all the approximations become less accurate. For example, the actual energy loss when the swing foot landed becomes larger as the step length increases. Since we are using LIPM model that has massless legs, we could not estimate the energy loss from the LIPM model. This introduces some inaccuracy in the foot placement estimation, as the biped sometimes overestimates or underestimates the foot placement. In consequence, for a push of moderate magnitude, our biped may sometimes walk back and forth for a few steps in attempt to recover from the push. On the other hand, for a very hard push, the uncalculated energy loss may actually help to reduce the biped’s excessive momentum at a faster rate than the theoretical LIPM. Moreover, we assume theoretically that when a hard push occurred, the deceleration phase distance must be more than the acceleration phase so that the biped could recover from the push. However, because there are energy losses in the actual stepping, we have seen some cases where the biped could maintain walking although this condition is not satisfied.

For the constraint on the stepping time, we would like to confirm that generally the performance is better when the stepping time limit is smaller (except for the foot obstruction cases). For the ankle torque, generally a larger torque allows the biped to recover from a larger push magnitude. However, when the ankle torque calculation is not exactly accurate as needed by the realistic biped, the biped's ZMP could be shifted outside of the support polygon and the foot may start to rotate with respect to the ground. This observation affirms our preference in the control policy that it is generally safer to have fast and small steps for a push recovery scheme.

Our approach still has a prominent limitation, which are to be resolved in the next chapter. A push with a large magnitude causes the biped to rotate quite significantly, which causes the swing foot to land prematurely and not at the spot as intended in the theoretical LIPM.

3.6. Summary

In this chapter, we have presented a general push recovery scheme to handle arbitrary pushes when the biped is walking. We have demonstrated the effectiveness of our strategy through dynamic simulation, in which the biped could recover from pushes with various magnitudes and directions that may occur at any walking phases. Although we have demonstrated that our overall strategy could handle arbitrary push, it should be noted that our push recovery scheme is not a unique solution. In the event of a push, there could be other sets of solution, and perhaps better ones, to achieve a successful push recovery. More research on push recovery is required to explore the effectiveness of different strategies and approaches, before bipeds are ready to operate in our real dynamic environment.

Chapter 4

Additional Strategy and Application

4.1. Background

In the previous chapter, a push recovery algorithm for bipedal walking has been described and implemented in a simulated realistic biped. The peak performance of the simulated realistic biped in recovering from pushes with varying direction and magnitude during walking has been evaluated.

In this chapter, an additional strategy which could further improve the performance is proposed. This additional strategy is formulated to solve some part of the limitations that has been observed during previous evaluation test. Furthermore, an additional application of the overall strategy is shown. Although the overall strategy is designed for push recovery, it could also be adapted for general balancing purpose during bipedal walking. An interesting demonstration of the simulated biped balancing on an accelerating and decelerating cart is presented.

4.2. An additional strategy: foot placement compensator

In most traditional powered biped literature, it is generally assumed that a biped's foot should never rotate with respect to the ground (in x and y axis) in order to maintain walking. Since generally there is not much disturbance during normal walking, this assumption is reasonable. In fact, most ZMP based strategy focus on the preservation of large stability margin to maintain walking [e.g. 65, 66, and 67]. The idea is that if the ZMP can be kept inside the support polygon, then the foot will never rotate with respect to the ground (rotation in x and y axis). If the theoretical ZMP is shifted outside of the support polygon, then it is assumed that a fall is imminent because the foot started to rotate with respect to the ground. Goswami [69] has presented an in depth analysis about the foot rotation point and its relation to bipedal walking balance.

In this section, we shall present a scheme to handle the foot rotation and its consequences. Instead of trying to guarantee that a foot rotation will not occur, we will predict the effect of the foot rotation and make adjustments to the biped's foot placement. The foot placement compensator strategy is most needed when the magnitude of a push inflicted to the walking biped is very large; in which foot rotation is a common phenomena that could not be avoided.

4.2.1. The foot rotation problem

In the previous evaluation tests, a common foot rotation problem that limits the performance has been acknowledged. This problem seems to occur across all test points that we observed, especially when the push magnitude is very large. We will describe the problem for a push that comes from behind the biped, as follows:

Suppose a LIPM, which represents a biped, is set to achieve a posture as shown in Fig. 27 at the end of step n. In the LIPM, $x_T^{(n)}$ is length of the accelerating phase, $x_0^{(n+1)}$ is the length of the decelerating phase. The COM is designed to move linearly with a constant height ($z = z_0$).

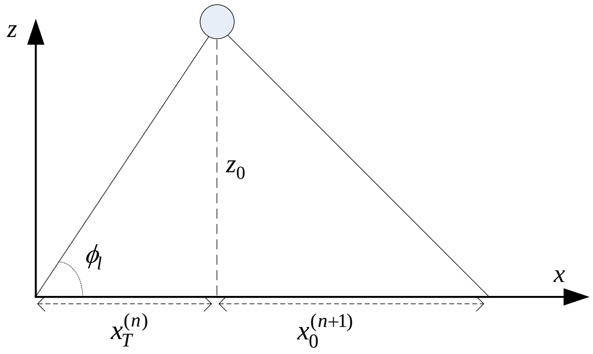


Figure 27: LIPM at support exchange at the end of step n.

However, when a very hard push is acting upon a biped at a particular time during a stepping time, the theoretical ZMP could be immediately shifted outside the support polygon and the biped will start to rotate with respect to the stance foot. Consequently, at the end of the stepping time the biped may end up in a tilted position with a deviation angle. This deviation angle is negligible when the magnitude of the push is small, but becomes more prominent as the magnitude of the push is larger. A relatively large deviation angle will cause the swing foot to hit the ground prematurely with a large force (Fig. 28). Furthermore, the actual deceleration distance becomes considerably smaller than the intended distance $x_0^{(n+1)}$. Thus, the biped does not decelerate as much as intended. All of these consequences may contribute to the failure of the push recovery attempt.

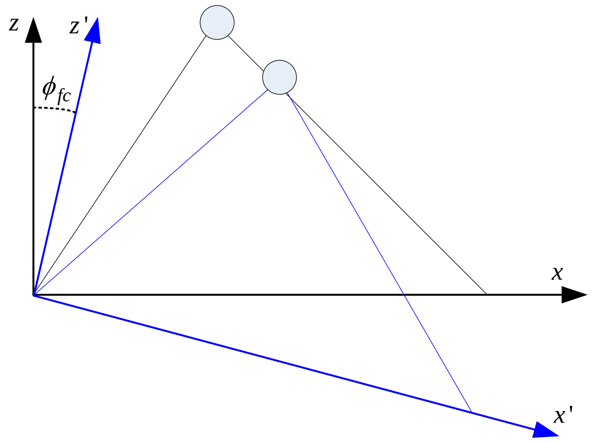


Figure 28: Because of the impulse received from a very hard push, a biped could be tilted heavily. This relatively large tilt will cause the actual swing foot of the biped to hit the ground prematurely with an abrupt impact force, and at an improper location.

Similar problem also occurred when the push comes from arbitrary directions. For some other cases, the push may cause the swing foot to land too late, which could also cause the biped to fall. But the root of the problem remains the same: the push causes the stance foot to rotate with respect to the ground, which causes the swing foot to land at the wrong location and time.

4.2.2. The concept of foot placement compensator

To accommodate for the foot rotation problem that has been described in the previous section, we propose a controller called the foot placement compensator. The main philosophy of the compensator is to redirect the swing foot position, such that the swing foot will land at the right time and at the intended location relative to COM. Hence, the timing of the landing and the intended deceleration distance of the LIPM could be achieved.

Fig. 29 shows the overview of the overall push recovery scheme. The foot placement compensator is implemented after the walking phase modification plans the gait. Then, the initial foot placement will be modified accordingly. The local joint compensator (described in section 3.4) is also a key component to make the overall strategy works, as it will keep the swing foot orientation to be parallel with ground at landing. Thus, enabling the foot to land relatively flat on the ground.

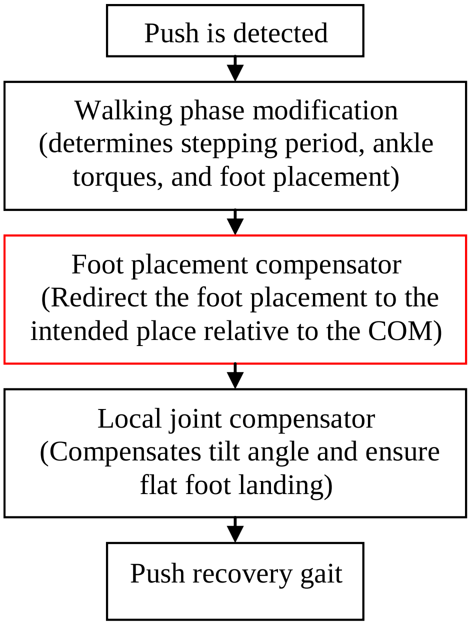

Push is detected
Walking phase modification (determines stepping period, ankle torques, and foot placement)
Foot placement compensator (Redirect the foot placement to the intended place relative to the COM)
Local joint compensator (Compensates tilt angle and ensure flat foot landing)
Push recovery gait

Figure 29: Overview of push recovery strategy. The additional strategy is placed after the walking phase modification is done.

4.2.3. Implementation of the foot placement compensator

To redirect the swing foot to a new foot placement, the foot placement compensator will calculate a set of compensation values at the moment the biped enters *pushed state*. As mentioned earlier, by redirecting the foot the deceleration distance of the rotated COM could be made as if no rotation has occurred. Hence, the LIPM could decelerate as planned by the walking phase modification.

The compensation values are found using geometric approach. We will once again describe the foot placement compensator using the case where the push comes from behind the biped (Fig. 30):

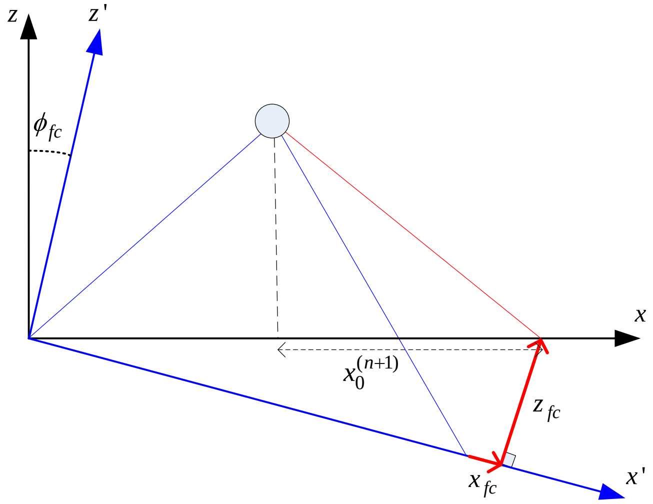


Figure 30: The foot placement compensator redirects the foot placement such that the biped could land the foot at the intended location relative to the tilted COM in x axis.

For simplicity, we will assume the biped will rotate with a constant angular velocity. The deviation angle at the end of the stepping time is estimated based on the initial tilt angle, the stepping time, and the angular velocity obtained from sensory reading as follows:

$$\phi_{fc} = \phi_0 + \omega T \tag{5}$$

where ϕ_{fc} is the estimated tilt at the end of the stepping time, T is the stepping time, ϕ_0 and ω is the initial tilt and angular velocity when the biped enters push state or at the beginning of a new stepping time, respectively. If necessary, more accurate approximation of the biped rotation could be made, provided that a more accurate sensory system and computation power is available.

Then, the first compensation value z_{fc} is made as follows:

$$z_{fc} = \left(\frac{x_T^{(n)}}{\cos\phi_l} \cos\left(\phi_l - \phi_{fc}\right) - x_0^{(n+1)} \right) \sin\theta_{fc} \tag{26}$$

where

$$\phi_l = \tan^{-1}\left(\frac{z_0}{x_T^{(n)}} \right)$$

Then, the second compensation value x_{fc} is calculated as follows:

$$x_{fc} = \frac{z_{fc}}{\tan\phi_{fc}} - \left(x_T^{(n)} - x_0^{(n+1)} \right) \tag{27}$$

Once the foot placement compensation values have been obtained, it will be added to the original foot placement. By adding these compensation values, the swing foot could land closer to the proper place, with the deceleration distance $x_0^{(n+1)}$, as planned by the walking phase modification.

The same approach is applied to find the compensation values for pushes from any other directions. Detailed Foot Placement Compensator Algorithm and its implementation could be found in Appendix IV.

4.2.4. Overall strategy

Fig. 31 shows the overall strategy flow chart. As described earlier, in the overall strategy the foot placement compensator is added in the gait determination process, after the stepping time and the original foot placement has been decided by the walking phase modification. All of these gait determination is executed at the moment the biped enters pushed state. Then, the local joint compensator works to balance the biped during its stepping time.

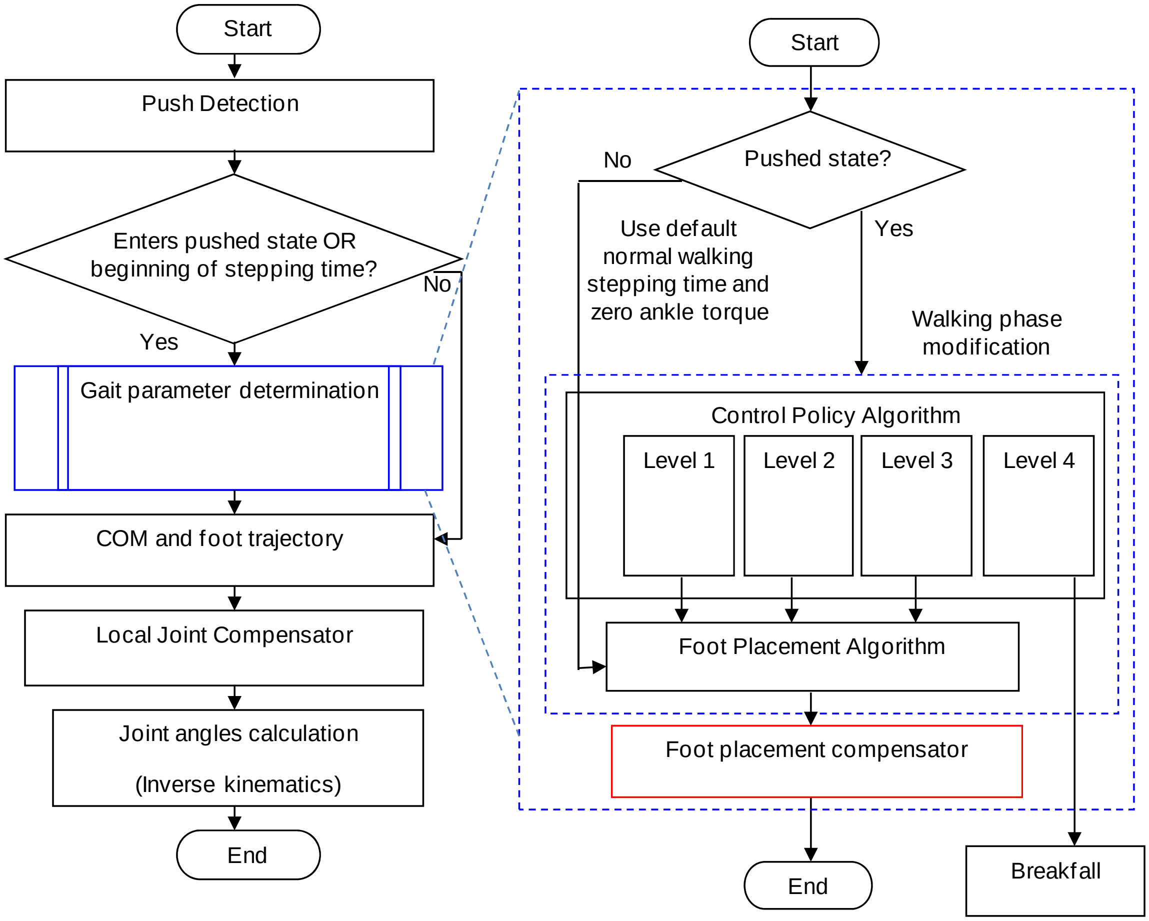


Figure 31: Overall Strategy. The flowchart on the left is iterated at every sampling time. The gait parameter determination process (flowchart on the right) is conducted at the moment the biped enters push state or at the beginning of a stepping time.

4.2.5. Push recovery experiments with realistic humanoid robot model in dynamic simulation

In this section, a series of push recovery experiments to examine the effectiveness of adding foot placement compensator in our overall strategy is presented. First, we will examine the performance of the overall strategy to handle pushes coming from the sagittal and lateral planes exclusively. Then, we will demonstrate that our overall strategy could work for pushes that comes from arbitrary directions.

- Performance Evaluation

The performance of the biped while using both walking phase modification and foot placement compensator will be compared with the performance of the biped without any feedback and with walking phase modification only (results obtained in previous chapter). In our test, the performance indicates the maximum impulse that does not cause the biped to fall after a reasonable observation time.

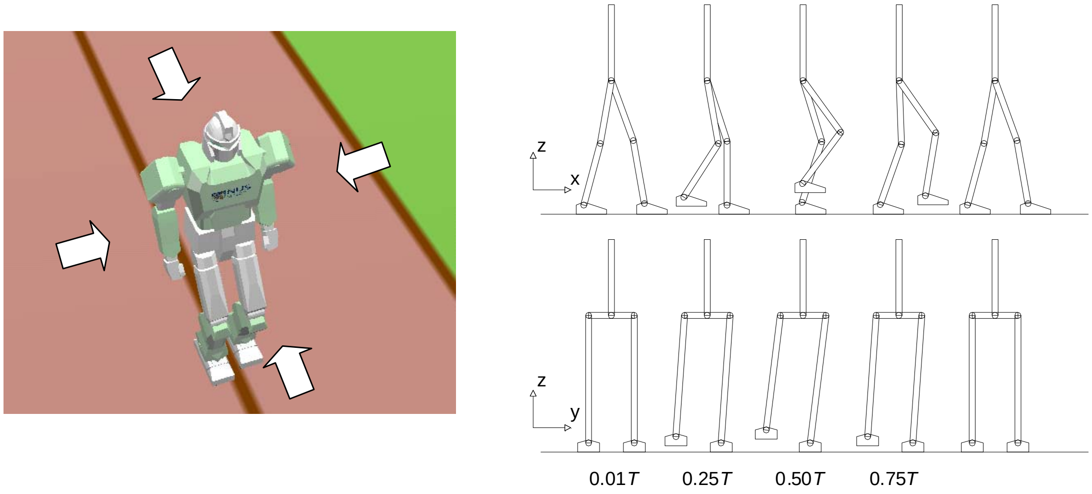
The impulse is calculated from a force sensor attached at the tip of the pusher ball. The impulse is calculated as:

$$imp_{1\to 2} = \int_{t_1}^{t_2} F dt$$

where F is the measured force at sampling time $t_1 < t < t_2$. The impulse information is displayed for our observation purpose only; it is not used in any way in the push recovery strategy.

Some key simulation parameters and constraints that we used are as follows: Sampling time: $0.008s$; desired states in the lateral plane: $y_{dr} = 0.095$ m, $\dot{y}_{dr} = 0.296$ m, $y_{dl} = -0.095$ m, $\dot{y}_{dl} = -0.296$ m; desired orbital energies: $E_{dx} = 0.065$ m^2/s^2, $E_{dy} = -0.019$ m^2/s^2; normal walking stepping time $T_n = 0.64$ s. desired states in sagittal plane for walking forward: $x_d = 0.145$ m, $\dot{x}_d = 0.651$ m/s, for stepping on the spot: $x_d = 0$ m, $\dot{x}_d = 0$ m/s. The constraint values are as follows: stepping reach $x_{forward} = 0.2$ m, $x_{backward} = -0.2$ m, $y_{outward} = 0.18$ m, $y_{inward} = 0.094$ m; estimated minimum stepping time $T_{\lim} = 0 - 0.4$ s (depending on swing foot position); ankle torque $\tau_{\lim}^{+} = 30$ Nm, $\tau_{\lim}^{-} = -30$ N/m. The biped's linear momentum during walking forward is about 27 - 44 Ns, with forward linear velocity of 0.3-0.5 m/s. These parameters and constraints will be kept constant throughout the tests.

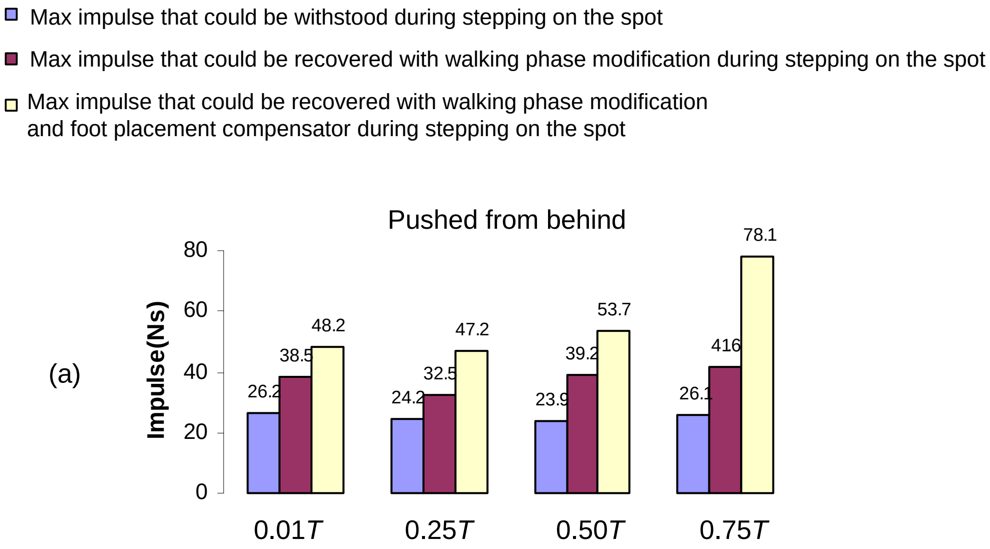
Similar with the experiments in the previous chapter, the performance tests will be evaluated while the biped is stepping on the spot and walking forward at left support phase. The push will come from four directions (Fig. 32a). For each direction, the pushes occur at four test points. The test points are at $T_k = 0.01T_n$, $T_k = 0.25T_n$, $T_k = 0.5T_n$, and $T_k = 0.75T_n$ (Fig. 32b). These test points represents acceleration phase ($T_k = 0.5T_n$, and $T_k = 0.75T_n$) , deceleration phase ($T_k = 0.01T_n$, and $T_k = 0.25T_n$) , double support phase ($T_k = 0.01T_n$), and single support phase ($T_k = 0.25T_n$, $T_k = 0.5T_n$, and $T_k = 0.75T_n$). Fig. 33 and Fig. 34 shows the performance test results.



(a) (b)

Figure 32(a-b): Performance evaluation. Four push directions are applied: behind, front, left, and right. For each direction, the pushes occur at four different timings during right foot swing phase.

(a)



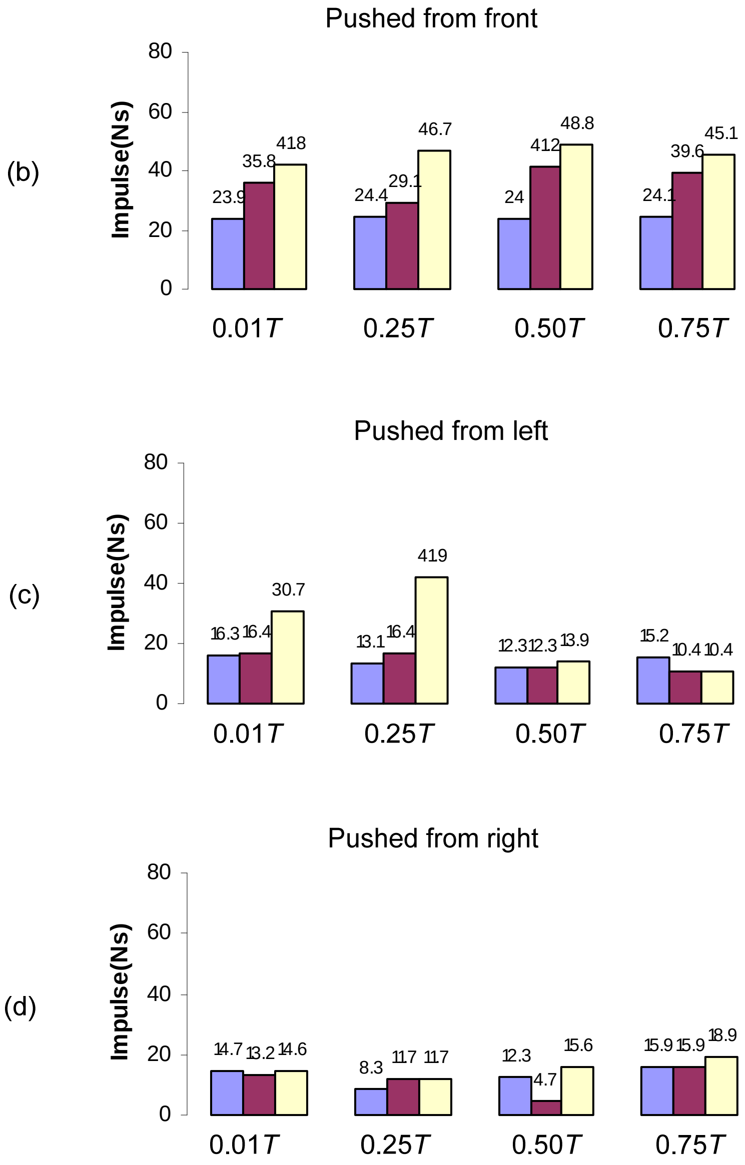
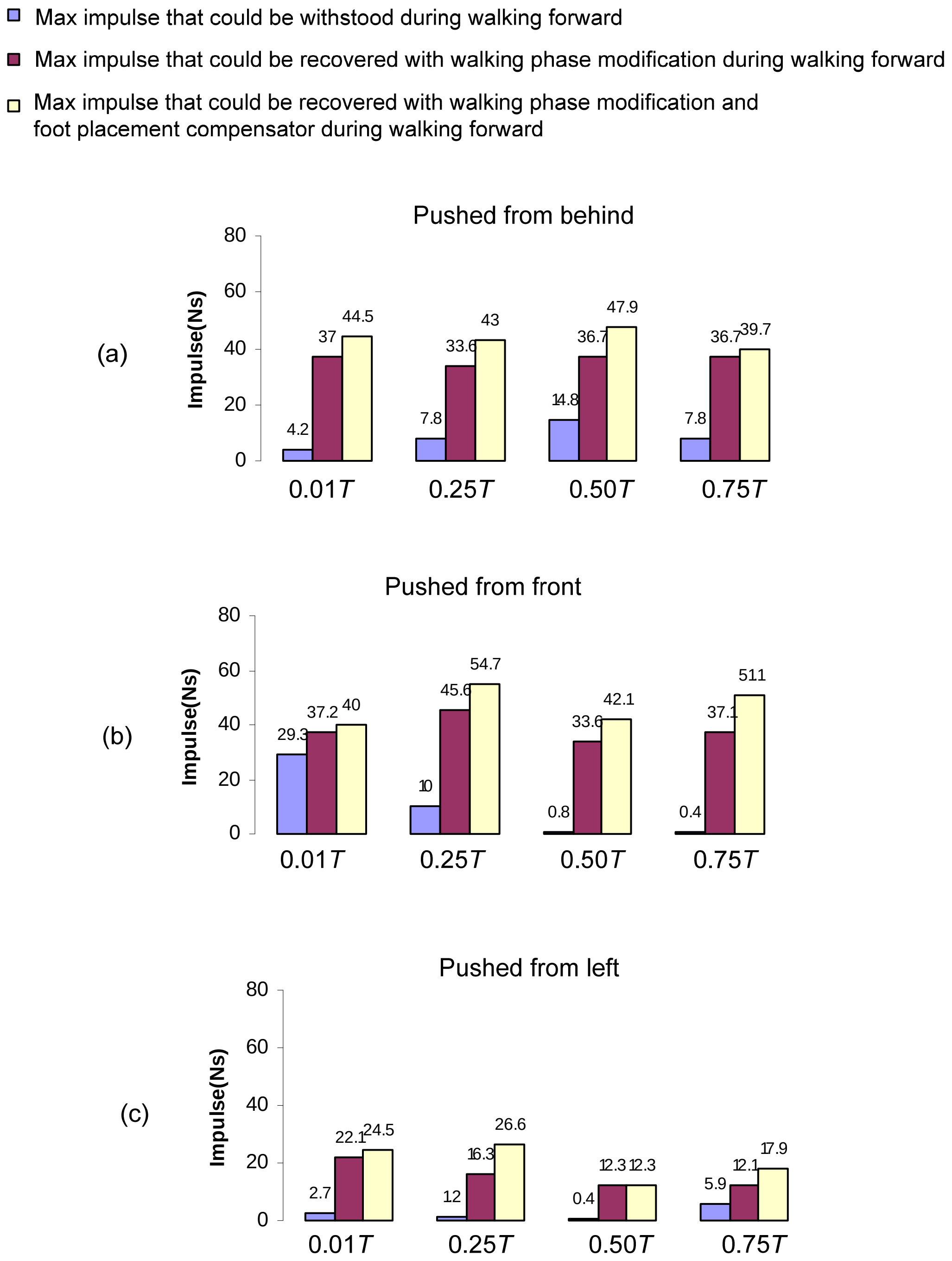


Figure 33(a-d): Performance evaluation when the biped is stepping on the spot, at the right foot swing phase.



Max impulse that could be withstood during walking forward
Max impulse that could be recovered with walking phase modification during walking forward
Max impulse that could be recovered with walking phase modification and foot placement compensator during walking forward
(a)
Pushed from behind
Impulse(Ns)
80
60
40
20
0
4.2
37
44.5
7.8
33.6
43
14.8
36.7
47.9
7.8
36.7
39.7
0.01T
0.25T
0.50T
0.75T
(b)
Pushed from front
Impulse(Ns)
80
60
40
20
0
29.3
37.2
40
10
45.6
54.7
0.8
33.6
42.1
0.4
37.1
511
0.01T
0.25T
0.50T
0.75T
(c)
Pushed from left
Impulse(Ns)
80
60
40
20
0
2.7
22.1
24.5
12
16.3
26.6
0.4
12.3
12.3
5.9
12.1
17.9
0.01T
0.25T
0.50T
0.75T

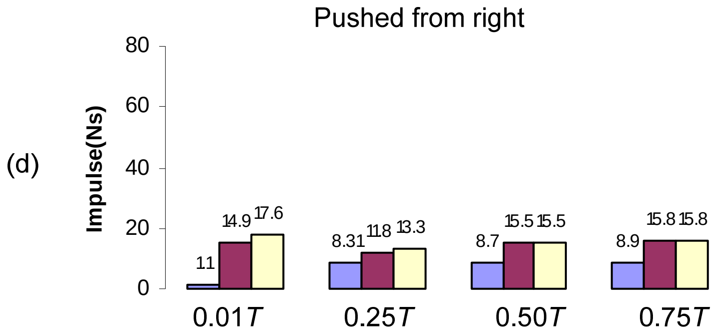


Figure 34(a-d): Performance evaluation when the biped is walking forward, at the right foot swing phase.

As can be seen in the results, the addition of foot placement compensator increases the performance considerably in most test points. Most significant increase is noted in cases where the biped is stepping on the spot, especially when the push occurred from behind and front of the biped. The foot rotation problem that occurred at these cases is compensated quite well, and the biped has higher capability to maintain walking. The addition of foot placement compensator is also effective to increase the performance when push comes from the left sides, especially during the early phases. This result verifies the usefulness of the proposed foot placement compensator method to increase the success rate of push recovery.

However, we observed that the addition of foot placement compensator seems to be not so effective to increase the performance for some side pushes. Several factors may be related to this issue. The first factor is related to the stability margin. In our biped model, the foot width, the constraint on the stepping reach during a side step is much smaller compared to the foot length and the forward/backward stepping. This condition makes side toppling over much easier than forward/backward toppling over. The second factor is related to the limitation that the biped could not cross its legs. During push recovery, a biped may take several steps to balance itself and return to its desired states. For side stepping, the swing leg is obstructed by the stance leg at every couple of steps, which limits the chance of the biped to step at the necessary location. These issues became even more prominent at the later phases of walking, where the COM acceleration magnifies the effect of the push. All of these coupled factors limit the success rate to maintain walking after receiving side pushes.

A plausible and simplistic solution to these problems is to have a different level of push detection threshold. While the biped is operating near the ineffective points, the threshold of the push detection could be increased. Thus, the biped does not use the push recovery scheme at all. By doing this, the biped will only perform the push recovery scheme when it is effective. Future researches should explore more solutions which could solve this problem. For example, the biped should be able to cross its legs. However, this solution required a sleek biped leg design with a high dexterity and degree of freedom. Another interesting potential solution could also be observed from humans. From our simple observations, humans may try to hop sideways when the swing leg is obstructed by the stance leg. Raibert [3] and Tajima, *et al.* [60] has also shown that hopping could be used to regain balance.

- Push recovery from arbitrary pushes

In this part, the bipedal robot will demonstrate the combined walking phase modification and foot placement compensator scheme to recover from several pushes with various magnitude and direction. In the first experiment, four subsequent pushes is applied when the biped is stepping on the spot. Table 1 shows the directions, magnitude, and the moment of the pushes. In table 5, α is the angle of the pusher in the transverse plane, with respect to x axis. imp is the impulse, and T_k is the walking phase when the push occurred. The linear velocity profile of the biped in x and y axis recorded from the experiment can be seen in Fig. 35.

Table 5: Push specifications, applied when the biped is stepping on the spot

First	Second	Third	Fourth
$\alpha = -2.9$ rad	$\alpha = 1$ rad	$\alpha = 0.1$ rad	$\alpha = -2.8$ rad
$imp = 51.1$ Ns	$imp = 22.1$ Ns	$imp = 38.3$ Ns	$imp = 30.1$ Ns
$T_k = 0.23T_n$	$T_k = 0.13T_n$	$T_k = 0.48T_n$	$T_k = 0.58T_n$

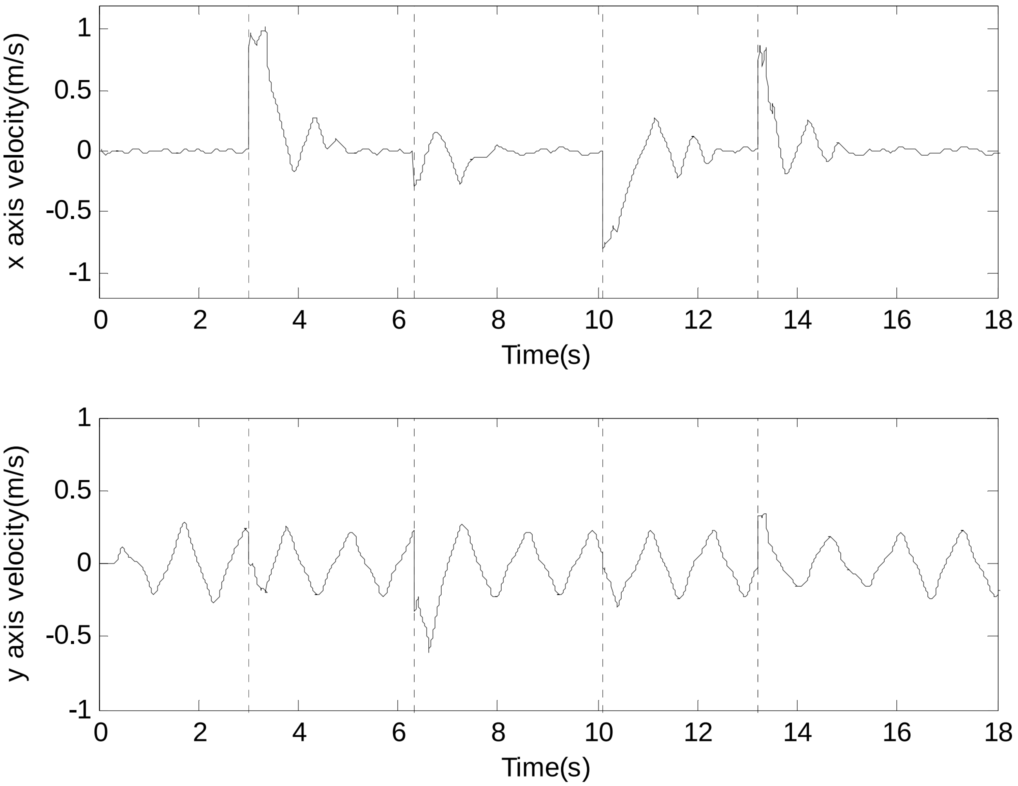


Figure 35: The velocity profile of the biped recorded form the experiment, where 4 subsequent arbitrary push is applied while biped is stepping on the spot. The dotted vertical lines are the moment of the pushes.

In the next experiment, four subsequent pushes are applied while the biped is walking forward. Table 6 shows the specifications of the pushes. The linear velocity profile of the biped in x and y axis can be seen in Fig. 37. Note the sudden velocity change at the moment of pushes. Then, the biped is able to recover the velocity to normal level and maintain walking.

Table 6: Push specifications, applied when the biped is walking forward

First	Second	Third	Fourth
$\alpha = -3\,\mathrm{rad}$	$\alpha = 1.9\,\mathrm{rad}$	$\alpha = -0.3\,\mathrm{rad}$	$\alpha = 2.8\,\mathrm{rad}$
$imp = 36.7$ Ns	$imp = 22.5$ Ns	$imp = 32$ Ns	$imp = 10.7$ Ns
$T_k = 0.33T_n$	$T_k = 0.35T_n$	$T_k = 0.66T_n$	$T_k = 0.6T_n$

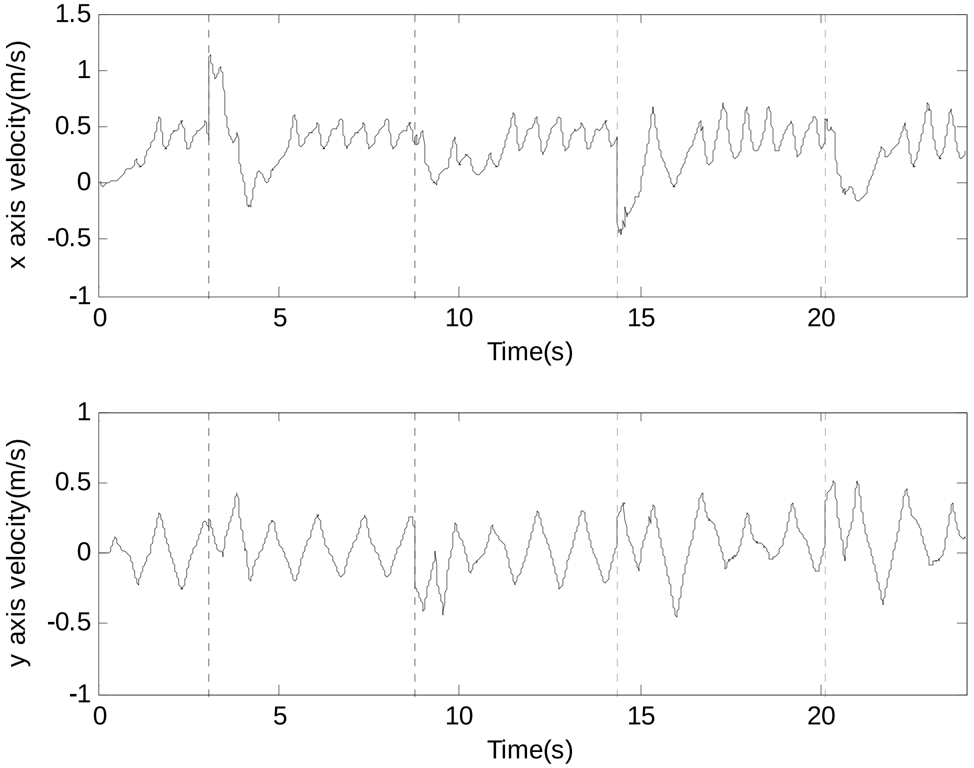


Figure 36: The velocity profile of the biped where 4 subsequent arbitrary push is applied while biped is walking forward. The dotted vertical lines are the moment of the pushes.

In the arbitrary push recovery experiments, the biped has demonstrated the capability to maintain walking after receiving pushes that comes from various directions and magnitudes. Overall, the performance of the biped to maintain walking after receiving arbitrary pushes is similar with the results observed from the performance evaluation. The biped is generally more capable to withstand pushes that comes from the frontal and rear area, compared to pushes that comes from the sides.

4.3. An additional application: balancing on accelerating cart

Generally, a push recovery scheme is used to maintain dynamic balance after receiving a push. But because our overall scheme essentially monitors the change of orbital energy at all times during walking, it could also be used to detect bipedal walking instability in a more general sense. This section presents an adaptation of the push recovery scheme for the purpose of maintaining bipedal walking on an accelerating or decelerating cart.

Consider a biped is stepping on the spot on top of a moving cart (Fig. 37). The cart's velocity changes over time with some magnitude of acceleration in forward and backward direction. The velocity and the acceleration rate of the cart are unavailable for the biped. This scenario resembles a situation where a person is walking on an accelerating or decelerating train.

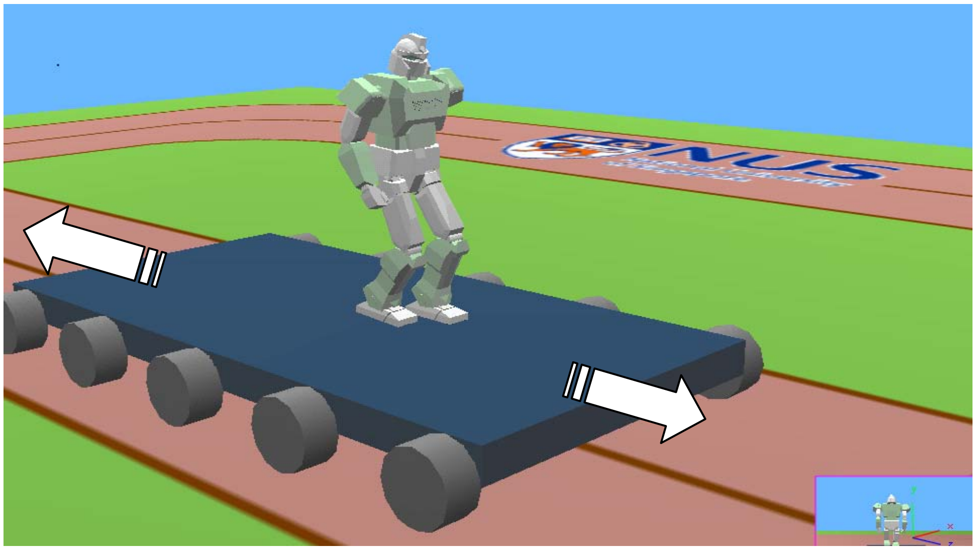


Figure 37: A bipedal robot is walking on an accelerating or decelerating cart. The biped tries to maintain walking while the dynamics of the cart is unknown to the biped.

4.3.1. The Problem of balancing on accelerating cart

When the cart accelerates or decelerates, the biped's moment of inertia will resist the motion. Hence, the biped will start to rotate about its foot (Fig. 38). If no action is taken to recover the biped's balance, the biped may topple and fall. To our knowledge, until now there is no ongoing bipedal walking research that explores this problem.

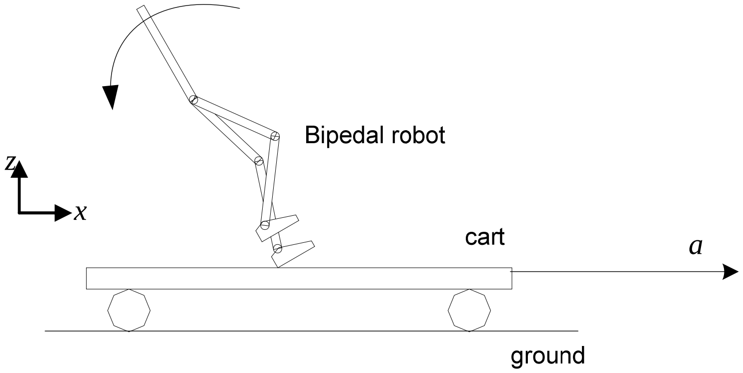


Figure 38: A bipedal robot is walking on an accelerating or decelerating cart. The acceleration of the cart will cause the biped to rotate, which may cause a fall.

Interestingly, there are many similarities and few differences between balancing on accelerating cart problem and push recovery problem. In the balancing on cart problem, the biped is experiencing a moment that has similar effect as a push force on the upper body. In this thesis, the moment is assumed to be acting on the biped's center of mass (COM). The magnitude is of the moment is proportional to the acceleration of the cart. The biped tends to rotate as long as the cart accelerates. Thus the acceleration time of the cart has similar effect with the push duration. In our study, the cart's acceleration time is constrained to be short, or considered instantaneous, as compared to the stepping time of the biped. The acceleration could also occur at any walking phases in the biped's walking cycle. It is also assumed that the cart is level and the leg motion is not hindered.

The main issue on the balancing on cart problem is that the biped does not know the acceleration and velocity value of the cart. This is a realistic human daily life scenario, where a person who does not drive the train/cart could not know the exact velocity or acceleration value of the train/cart. As with the push recovery problem, the biped could only use its own sensors to detect its own instability. The concept of orbital energy as an instability detection is still relevant because it remains constant during normal bipedal walking. Moreover, the sensory feedbacks could be obtained relatively fast with low computation cost from inertial measurement unit (IMU).

In sensing the biped's orbital energy, there is an important difference between conventional push recovery problem and the balancing on accelerating cart problem. In the conventional push recovery problem, the orbital energy is obtained from the value of COM linear velocity. The COM linear velocity is obtained directly from linear velocities measurement of the IMU. However, this information could not be used to solve the balancing on accelerating cart problem because the IMU linear velocity measurement is a mix between the cart velocity and the biped's velocity relative to the cart. Since the cart velocity is unknown, the biped's velocity could not be deducted from this information. Therefore, other means to detect the biped's linear velocity relative to the cart is needed.

4.3.2. Strategy for balancing on accelerating cart

We will use the IMU angular velocity information to approximate the biped's linear velocity. This adaptation is the key strategy to solve the balancing on accelerating cart problem. The linear velocity value will be derived from the sensory reading of the angular velocity as follows:

$$\dot{x}_{av}(t) = \omega(t) z_s \tag{28}$$

where $\dot{x}_{av}(t)$ is the derived velocity value, $\omega(t)$ is the angular velocity reading and z_s is a constant representing the height of the sensor from the biped's foot. Because the value of $\dot{x}_{av}(t)$ depends on the biped's own angular velocity, it is related to the biped's dynamics and therefore it could represent the linear velocity of the biped relative to the cart (Fig. 39).

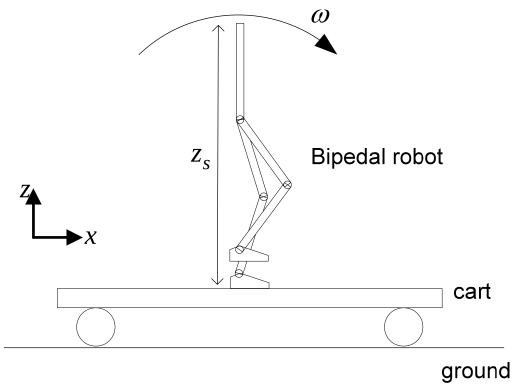


Figure 39: A bipedal robot is walking on an accelerating or decelerating cart. The biped will try to maintain walking on a moving cart, while the velocity of the cart is unknown to the biped.

Subsequently, all linear velocity values in the controller will be replaced by $\dot{x}_{av}(t)$. This approach is applied to the lateral plane as well. After the instability is detected through the change in the orbital energy level, the controller continues with the same overall strategy as the push recovery scheme (Fig. 31). The balancing on accelerating cart problem can now be treated in the same way as the conventional push recovery problem.

4.3.3. Balancing on accelerating cart experiment with realistic humanoid robot model in dynamic simulation

This subsection presents a dynamic simulation of the realistic biped stepping on the spot while the cart is accelerating and decelerating. Fig. 40 shows the recorded velocity profile of the cart and the velocity of biped relative to the ground ($\dot{x}(t)$) in x axis. As can be seen, the velocity of the biped fluctuates as the biped is doing push recovery and tries to maintain walking while the cart accelerates and decelerates.

Fig. 41 shows the velocity value derived from the angular velocity ($\dot{x}_{av}(t)$) recorded from the simulation. As can be seen, the value of $\dot{x}_{av}(t)$ fluctuates around zero, which is relative to the stance foot rotation point on the cart. Although $\dot{x}_{av}(t)$ fluctuates quite heavily, the overall strategy still manages to maintain walking. The addition of the foot placement compensator proved very useful for balancing on accelerating cart problem as well, since most of the time the biped tilted quite significantly due to reaction forces. This result verifies that the overall push recovery scheme could be adapted to for the balancing on accelerating cart problem.

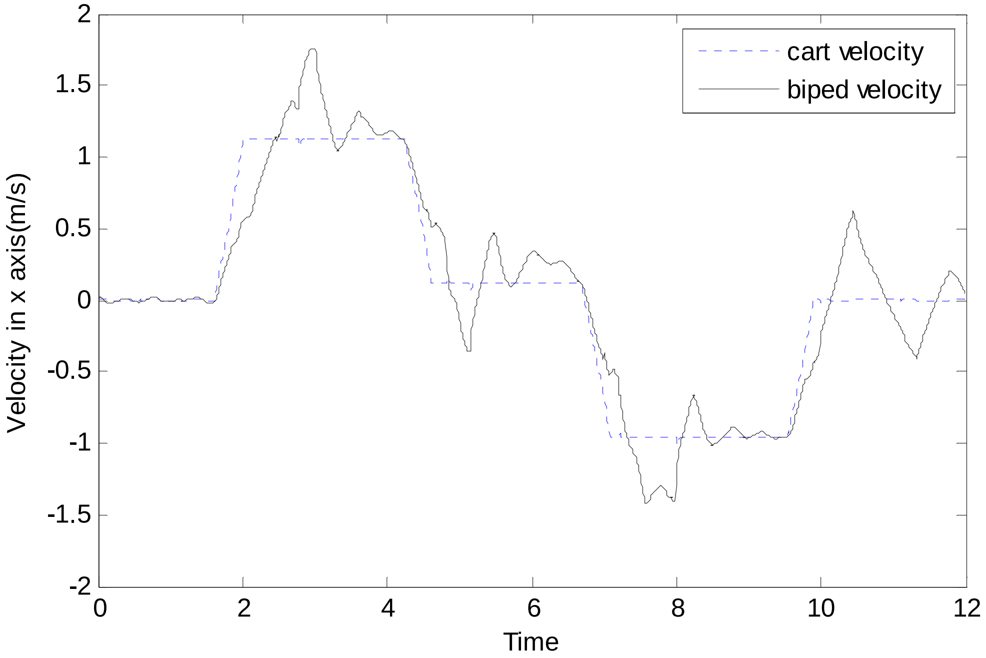


Figure 40: The velocity of the cart and the velocity of the biped $\dot{x}(t)$ obtained directly from IMU linear velocity measurement. In this figure, the biped's velocity is relative to the ground, which is a mix between the cart velocity and the biped's velocity relative to the cart.

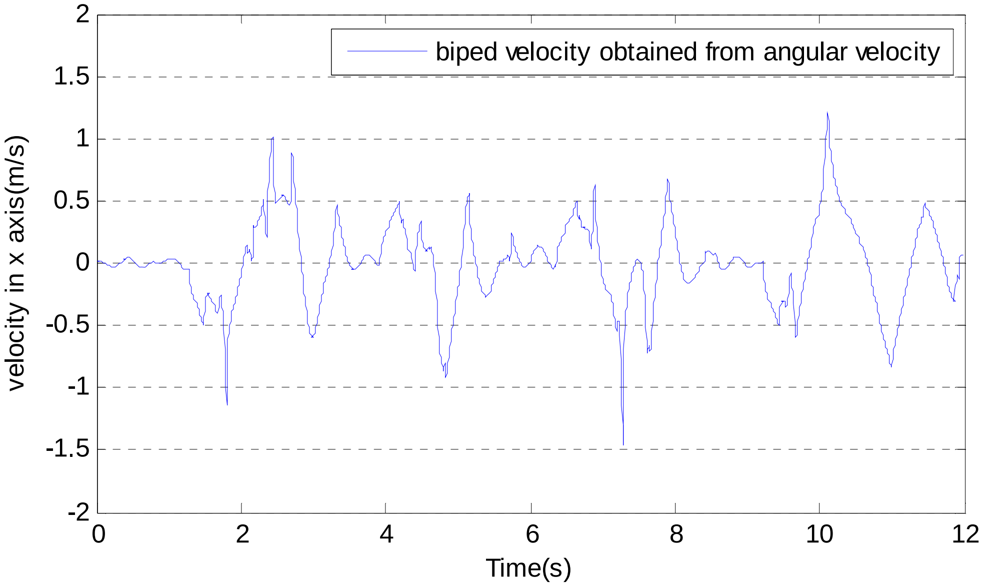


Figure 41: The derived linear velocity of the biped $\dot{x}_{av}(t)$. The angular velocity value is relative to the cart, and therefore $\dot{x}_{av}(t)$ could be used to approximate the biped's velocity relative to the cart.

4.3.4. Discussion on balancing experiment on accelerating cart

From our observation during experimentation, the derived velocity value $\dot{x}_{av}(t)$ works well to approximate the linear velocity and the orbital energy of the biped, up to a certain magnitude of the acceleration or deceleration of the cart. However the approach starts to lose its effectiveness as the acceleration or deceleration of the cart becomes very large. This is due to some issues and limitations as follows:

The realistic biped is modeled with a LIPM. Being a simple model, the LIPM also has its own drawback such as the simplified linear dynamic that do not exactly match the realistic biped's nonlinear dynamics. The discrepancy between the LIPM and the realistic biped becomes more prominent as the step length is getting larger because of a larger push magnitude. As in the conventional push recovery, the discrepancy may cause the biped to undershoot or overshoot the necessary stepping location. The necessary stepping location is also determined by other factors not included in the LIPM such as energy loss during stepping. As consequence, the biped sometimes walks back and forth for a few steps until the biped's state returns to the normal states.

While this is not so much of an issue for conventional push recovery problem, it may cause instability for the balancing on accelerating cart problem. This is related to the nature of the derived velocity value $\dot{x}_{av}(t)$, which fluctuates heavily around zero. As the biped overshoot the necessary stepping location, the new $\dot{x}_{av}(t)$ value may turn to the opposite sign with an increase in absolute magnitude. This problem may cycle itself with greater magnitude on the next step, which leads to system instability. If the value of $\dot{x}_{av}(t)$ did not decrease to an acceptable level, the biped will walk back and forth endlessly. A simple solution to this problem is to implement a watchdog that monitors the state of the biped. If the watchdog detects that the biped orbital energy is not decreased after several steps, the controller may declare that the push recovery has failed.

4.4. Summary

In this chapter, an additional foot placement compensator strategy that compensates the foot rotation effect due to a very hard push has been presented. A considerable improvement on the performance has been achieved. Moreover, an additional application of the push recovery strategy has been presented. By slightly altering the push detection method, the overall strategy could be used as a strategy to maintain walking on an accelerating and decelerating cart. This result demonstrates the versatility of our push recovery strategy to be adapted to different situations in a dynamic environment. Issues and limitation of our proposed scheme has been discussed as well.

After exploring push recovery and balancing on cart problem, we would like to reaffirm that these problems may also be approached with other schemes and solutions. Further observation and studies are needed to solve the performance limitation found in our cases.

Chapter 5

Conclusion

The primary objective of this study was to develop a bipedal robot walking control architecture that has a push recovery capability and to evaluate its effectiveness. The magnitude of the push, the push duration, the line of action, the walking phase when the push occurs, and the physical constraints had been considered in the proposed control architecture.

5.1. Summary of results

In Chapter 3, a generalized push recovery algorithm was developed. The nature of the push problem and the considerations that shaped the structure of the controllers was described. The walking phase modification as the main philosophy of the controller was introduced and presented. The push detection method, the physical consideration, and the control policy were also presented. The performance evaluation that shows the push recovery capability was shown and analyzed. Furthermore, the simulation results verified the effectiveness of the algorithm to maintain bipedal walking in the presence of arbitrary pushes.

In Chapter 4, a common problem that became prominent during a push recovery from a very hard push was acknowledged and analyzed. A foot placement compensator was developed to handle the problem. The performance of the combined strategies was presented, which showed significant performance improvement in most cases. The versatility of the overall strategy is also demonstrated through an experiment of balancing on an accelerating and decelerating cart.

5.2. Final Remarks

The simulation and experimental results confirms that it is possible to use simple model to regulate a complex system such as humanoid robot. Moreover, the same basic strategy could be applied to different bipedal robots with different mass properties, as long as they still share similar position control mode.

Our push recovery controller could be compared with the work of Komura, *et al.* [56], Wieber, *et al.* [59], and Tajima, *et al.* [19] who also developed push recovery capability during bipedal gait. Komura, *et al.* proposed a physical model to handle push from behind the robot and presented the result in 2D animation, Wieber, *et al.* proposed an online preview controller that could handle disturbance from the side, and Tajima, *et al.* presented an experimental push recovery while the robot was marching on place. However, none of them discussed the practical implementation and the performance evaluation of their push recovery approach.

The results of this thesis suggest that balancing can be unified as an integrated strategy. By little adjustment, our push recovery controller could be modified into a scheme that could maintain walking on an accelerating and decelerating cart. To our knowledge, no other researcher has presented such demonstration yet.

Besides controlling an actual bipedal robot, the algorithm could also be useful to generate human push recovery motion. Similar to the work of Komura, *et al*., this may potentially contribute to broad application such as movies, special effects, and interactive games that demand a realistic human motion based on physical properties.

5.3. Significance of the study

The result in this thesis is the first to thoroughly discuss push recovery problem for bipedal walking. Therefore, it could significantly contribute as a step towards the development of fully robust bipedal robot locomotion control, especially in the area of push recovery capability. The extension of the algorithm to cope balancing on a cart could also contribute towards a unified balancing strategy for bipedal robot that could handle various kinds of disturbance at once.

This thesis is also the first that proposed a systematic performance evaluation of the push recovery, which may help as a general benchmark for comparing various push recovery controllers.

5.4. Limitation and recommendation for future research

For our push recovery scheme, the parameters that needs to be determined are the LIPM parameters, constraint on the stepping (kinematic, time, and torque constraints), and the practical orbital energy error tolerances. These set of parameters could be determined intuitively without much effort, with good performance. However, it is challenging to obtain the most optimum set of parameters, which could withstand the largest range of disturbance across all test points. Truly, a comprehensive understanding on the dynamics of bipedal walking is much needed in optimizing these parameter values. Optimization method using reinforced learning as used by Chew, *et al*. [28], Rebula, *et al*. [45], or neural network [69] could be considered for the future.

Our approach also has several other limitations, which are to be resolved in future. First, the massless legs of the LIPM could not model the impact dynamics, which cause the foot placement to overshoot the ideal position occasionally. Second, we have not take into account angular momentum or hip strategy approach, which could further improve the performance.

The principles that have been established in this thesis, such as the walking phase modification and the control architecture, could also be applied to more complicated models, as long as the acceleration and deceleration phase of that model are clearly distinguishable. However, the dynamics solution of the complicated model will not be as straightforward as the LIPM.

Future study should also relax the assumptions that we have made:

- Currently, our proposed control architecture will only consider the initial LIPM state when the biped enters push state and at stance foot exchange instances. To accommodate properly to a continuous push, the controller will needs to be compliance and sensitive to the push magnitude and direction at all time. The capability to be guided by the push has not been realized in today's bipedal robot, and it will be an exciting topic to be discussed in future research.
- The assumption that push is acting on the COM is necessary to justify the usage of LIPM. If the push is not acting on the COM, a moment about the vertical axis will act on the biped. If the moment is large enough, it may cause the biped to rotate significantly about the vertical axis. To properly compensate for this moment, more complicated models are required. Whole body motion control similar to the resolved momentum control [24] may need to be implemented as well.
- The assumption that the ground is level is necessary to focus our effort on the push recovery only. If the ground is not level, an unconventional stepping reference, depending on the push recovery algorithm and the ground condition, may need to be incorporated to the control architecture.
- The assumption that the leg motion is not hindered is necessary for the biped to do a proper stepping. Otherwise, it could be considered that the biped is tripped. Until now, trip recovery problem has not been studied in bipedal robot research.

Potential future work is to implement our approach to the NUSBIP-III ASLAN (see appendix VI). Although our overall strategy is designed with practical consideration in mind, much work will be required to resolve the hardware limitations. First is the sensors limitation. Unlike the sensor in the simulation that can measure accurately the robot states, the actual inertial sensor has noises and the readings were far from ideal. This would cause some approximation error of the robot states. Thus, the subsequent decision based on these states will also have some error. To better approximate the robot states, a more advanced sensor system such as the IMU (Inertial Measurement Unit) and its filtering techniques should be considered for future hardware implementations.

To be able to perform a good push recovery performance, a high maximum joint speed for fast motions is needed. Future version of the hardware should improve the joint speed limit. However, it has been observed that servos became less accurate during high-speeds motions. Thus, the inaccuracy of the servos would cause some error in the stepping motion, which may cause the robot to fall. The strategy of Toyota robot by Tajima, *et al.* [19] to recalculate the COM trajectory and foot placement based on these errors should be considered for hardware implementation.

Overall, the proposed controller was generally effective to recover a bipedal robot to its equilibrium balance of walking. The algorithm presented in this thesis could contribute towards a robust bipedal walking controller that would enable a bipedal robot to operate safely and successfully in our dynamic environment.

Bibliography

[1] I. Kato, H. Tsuiki, “Hydraulically powered Biped Walking Machine with a High Carrying Capacity,” *Proceedings of the 4th International Symposium on External Control of Human Extremities*, pp. 410-421, 1972.

[2] M. Vukobratovic, A. A. Frank, D. Juricic, “On the stability of biped locomotion,” *Biomedical Engineering, IEEE Transactions on*, BME-17(1):25–36, 1970.

[3] M. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, 1986.

[4] M. Raibert and J. Hodgins, “Animation of dynamic legged locomotion.” *Computer Graphics*, 25(4):349–358, 1991.

[5] T. McGeer, “Passive dynamic walking.” *International Journal of Robotics Research*, 9(2), pp. 62–82, 1990.

[6] Kuo, A.D, “Choosing your steps carefully” *IEEE Robotics & Automation Magazine*, v 14, pp 18-29, June 2007.

[7] S. Grillner. ,”Control of locomotion in bipeds, tetrapods and fish,” American Physiological Society, Bethesda, 1981.

[8] S.Grillner and P. Zangger, ”How detail is the central pattern generation for locomotion,” *Brain Research*, 88:367–371, 1975.

[9] S. Aoi and K. Tsuchiya, “Locomotion control of a biped robot using nonlinear oscillators,” *Autonomous Robots*, 19:219–232, 2005.

[10] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng. Experimental studies of a neural oscillator for biped locomotion with qrio. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 596–602, 2005.

[11] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. A, “framework for learning biped locomotion with dynamical movement primitives,” In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots,*pages 925–940, 2004.

[12] J. Shan and F. Nagashima, “Neural locomotion controller design and implementation,” for humanoid robot hoap-1. In *Proceedings of The 20th Annual Conference of the Robotics Society of Japan*, 2002.

[13] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of Honda humanoid robot,” *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 1998,* Belgium, pp. 1321–1326. 1998.

[14] E. Dunn, R. Howe, “Towards smooth Bipedal Walking,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2489-2494, 1994.

[15] E. Dunn, R. Howe, “Foot Placement and Velocity Control In Smooth Bipedal Walking,” *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 578-583, 1996.

[16] J. E. Pratt, G. A, Pratt, “Exploiting Natural Dynamics in the Control of Planar Bipedal Walking Robot,” *Proceedings of the Thirty Sixth Annual Allerton Conference on Communication, Control, and Computing,* pp.739-748, 1998.

[17] J. E. Pratt, G. A, Pratt, “Intuitive Control of Planar Bipedal Walking Robot,” *Proceedings of the IEEE International Conference on Robotics and Automation,* 1998.

[18] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi, "Design of Prototype Humanoid Robotics Platform for HRP," *IEEE/RSJ international conference on intelligent robots and systems,* 2002.

[19] R. Tajima, D. Honda, K. Suga, "Fast running experiments involving a humanoid robot," *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 09,* Kobe, pp.1571-1576, 2009.

[20] J. Lee, J.-Y. Kim, I.-W. Park, B.-K. Cho, M.-S. Kim, I. Kim and J. H. Oh, "Development of a Human-Riding Humanoid Robot HUBO FX-1", SICE-ICCAS 2006, 2006.

[21] T. McGeer, "Passive dynamic walking with Knees." *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1640-1645, 1990.

[22] M. Wisse, G. Feliksdal, J. Van Frankkenhuyzen, B. Moyer, "Passive Walking Bipedal Robot," *IEEE Robotics and Automation Magazine*, pp. 52-62, 2007.

[23] M. W. Spong, "The swing up control problem for the acrobot," *IEEE Control Systems Magazine*, pages 49-55, Feb 1995.

[24] S. Kajita. F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada 2003.

[25] I.-W. Park, J.-Y. Kim and J.-H. Oh, "Online Walking Pattern Generation and Its Application to a Biped Humanoid Robot-KHR-3(HUBO)," *Journal of Advanced Robotics,* 2007.

[26] J.-Y. Kim, I.-W. Park, and J.-H. Oh, “Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor,” *Journal of Intelligent and Robotic Systems*, 2006.

[27] D. Katic and M. Vukobratovic, “Control Algorithm for Biped Walking Using Reinforcement Learning,” *2nd Serbian-Hungarian Joint Symposium on Intelligent Systems*, 2004.

[28] C.-M. Chew and G. A. Pratt, “Dynamic Bipedal Walking Assisted by Learning,” *Robotica*, Volume 20, p477-p491, 2002.

[29] S. kajita, A. Kobayashi, and T. Yamaura, “Dynamic Walking Control of a Biped Robot along a Potential Conserving Orbit,*” IEEE Trans. On R&A, vol RA-8, no.4,* pp.431-438, 1992.

[30] S. Kajita, and K. Tani, “Study of Dynamic Biped Locomotion on Rugged Terrain,” *Proceedings of the IEEE International Conference on Robotics and Automation,*, pp. 1405–1410, 1991.

[31] S. Kajita, O. Matsumoto, and M. Saigo, “Realtime 3D walking pattern generation for a biped robot with telescopic legs,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2299–2308, 2001.

[32] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, “A real-time pattern generator for biped walking,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 31–37. 2002

[33] T. Sugihara, Y. Nakamura, and H. Inoue, “Real time humanoid motion generation through ZMP manipulation based on inverted pendulum control,” in *IEEE International Conference on Robotics and Automation*, Washington, D.C., 2002.

[34] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, H. Inoue., ”Online generation of humanoid walking motion based on fast generation method of motion pattern that follows desired ZMP,” *IEEE/RSJ international conference on intelligent robots and systems*, Lausanne, Switzerland, 2002

[35] J. Y. Kim, I. W. Park, J. H. Oh., “Experimental realization of dynamic walking of the biped humanoid robot KHR-2 using zero moment point feedback and inertial measurement,” *Advanced Robotics*. 2006.

[36] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” *in Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 1620–1626, 2003.

[37] A. Hofmann, “Robust execution of Bipedal walking Tasks from Biomechanical Principles,” Ph.D. thesis. 2006.

[38] J. Pratt and R. Tedrake, “Velocity based stability margins for fast bipedal walking,” *in First Ruperto Carola Symposium in the International Science Forum of the University of Heidelberg entitled “Fast Motions in Biomechanics and Robots”*, Heidelberg Germany, 2005.

[39] A. Goswami and V. Kallem, “Rate of change of angular momentum and balance maintenance of biped robots,” *inIEEE International Conference on Robotics and Automation (ICRA)*, pp. 3785– 3790, 2004.

[40] T. Komura, A. Nagano, H. Leung, and Y. Shinagawa. “Simulating pathological gait using the enhanced linear inverted pendulum model. *IEEE transactions on bio-medical engineering*, 2005.

[41] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," *in 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 200–207, 2006.

[42] S. H. Lee, A. Goswami, "Reaction Mass Pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," *in IEEE International Conference on Robotics and Automation (ICRA),* pp. 4667-4672, 2007.

[43] D. N. Nenchev, A. Nishio, "Ankle and hip strategies for balance recovery of a biped subjected to an impact," Robotica 26(5): 643-653, 2008.

[44] B. Stephens, "Humanoid Push Recovery," *in IEEE-RAS 2007 International Conference on Humanoid Robots*, Pittsburgh, PA, 2007.

[45] J. Rebula, J.Pratt, F Canas and A. Goswami, "Learning Capture Point for Improved Humanoid Push Recovery," *in IEEE-RAS 2007 International Conference on Humanoid Robots*, Pittsburgh, PA, 2007.

[46] D. L. Wight, E. G. Kubica, and D. W. L. Wang. "Introduction of the Foot Placement Estimator: a Dynamic Measure of Balance for Bipedal Robotics," *Journal of Computational and Nonlinear Dynamics*, 2008.

[47] H. Hermami, F. Weimer, and S. Koozekanani, "Some aspects of the inverted pendulum problem for modeling of locomotion systems," *IEEE transactions on Automatic Control*, 1973.

[48] C. Atkeson and B. Stephens, "Multiple Balance Strategies from One Optimization Criterion," *in IEEE-RAS 2007 International Conference on Humanoid Robots,* Pittsburgh, PA, 2007.

[49] K. Kondak and G. Hommel, “Control and Online Computation of Stable Movement for Biped Robot,” *IEEE/RSJ international conference on intelligent robots and systems,* pp. 874-879, 2003.

[50] C. Golliday and H. Hemami, “Postural Stability of the two-degree-of-freedom Biped by General Linear Feedback,” *IEEE Transactions on Automatic Control*, 1976.

[51] B. Stephens, ”Integral control of humanoid balance,” *in IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems*, 2007.

[52] B. Stephen and C.G. Atkenson, “Push Recovery by Stepping For Humanoid Robot with Force Controlled Joints,” *in IEEE-RAS 2007 International Conference on Humanoid Robots,* Nashville, TN, USA, 2010.

[53] S.-H. Hyon, R. Osu, Y. Otaka, “Integration of Multi-Level Postural Balancing on Humanoid Robots,” *in IEEE International Conference on Robotics and Automation (ICRA),* Hyogo, Japan, 2009.

[54] Q. Huang, Y. Nakamura, and T. Inamura, "Humanoids Walk with Feedforward Dynamic Pattern and Feedback Sensory Reflection," *in IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 4220-4225.

[55] T. Komura, H. Leung, J. Kuffner, **“**Animating Reactive Motions for Biped Locomotion” *in Proceedings of the ACM symposium on Virtual reality software and technology,* USA, 2004.

[56] T. Komura, H. Leung, S. Kudoh, J. Kuffner, **“**A Feedback Controller for Biped Humanoids that Can Counteract Large Perturbations During Gait,” *in IEEE International Conference on Robotics and Automation (ICRA),* 18-22 April 2005 Page(s): 1989 - 1995.

[57] P.-B. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations.” *In IEEE International Conference on Humanoid Robotics*, 2006.

[58] A. Herdt, H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, “Online walking gait generation with adaptive foot positioning through linear model predictive control,” *In IEEE/RSJ International Conference on Intelligent Robots & Systems*, Nice, 2008.

[59] A. Herdt, H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, “Online Walking Motion Generation with Automatic Foot Step Placement,” *Advanced Robotics*, vol 24, pp. 719-737, 2010.

[60] R. Tajima, K. Suga, “Motion having a Flight Phase: Experiment Involving a One-Legged Robot”, *In IEEE/RSJ International Conference on Intelligent Robots & Systems*, China, 2006.

[61] Boston Dynamics, “PETMAN”, *in http://www.bostondynamics.com /robot_petman.html*, 2010.

[62] F. P. Beer, E. R Johnston, “Vector Mechanics for Engineers: Dynamics,” 3rd SI metric edition, Mcgraw-Hill, 2003.

[63] M. Vukobratovic, B. Borovac, “Zero Moment Point- Thirty five years of its life,” *in International Journal of Humanoid Robotics*, Vol. 1, No. 1 (2004) 157–173, 2004.

[64] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, 2001,” The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation,” *In IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2001.

[65] Q. Huang, K. yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, “Planning Walking Pattern for a Biped Robot,” *IEEE Trans on Robotics and Automation,* Vol. 17, No. 3, 2001.

[66] H. Lim and A. Takanishi,” Waseda biped humanoid robots realizing human-like motion,” *IEEE International Workshop on Advanced Motion Control*, 2000.

[67] J. H. Park, H. Chung, “ZMP Compensation by On-line Trajectory Generation for Biped Robots,” *in IEEE International Conference on Systems, Man, and Cybernetics,* Tokyo, Japan, 1999.

[68] A. Goswami, “Postural Stability of Biped Robots and the Foot Rotation Indicator (FRI) Point,” *International Journal of Robotic Research*, 1999.

[69] A. Kun, W. T. Miller, “Adaptive Dynamic Balance of a Biped Robot Using Neural Networks,” in *IEEE International Conference on Robotics and Automation (ICRA),* Minneapolis, USA, 1996.

[70] S. –J. Yi, B.-T. Zhang, and D. D. Lee, “Learning Full Body Push Recovery Control for Small Humanoid Robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[71] S. –J. Yi, B.-T. Zhang, and D. D. Lee, “Practical Bipedal Walking Control on Uneven Terrain Using Surface Learning and Push Recovery,” *In IEEE/RSJ International Conference on Intelligent Robots & Systems*, San Fransisco, USA, 2011.

[72] S. –J. Yi, B.-T. Zhang, and D. D. Lee, “Online Learning of a Full Body Push Recovery Controller for Omnidirectional Walking,” *in IEEE/RAS International Conference on Humanoid Robots,* Bled, Slovenia, 2011.

Appendix I: Derivation of LIPM with Ankle Torque

The LIPM in the sagittal plane and its free body analysis can be seen in Fig. A-1.

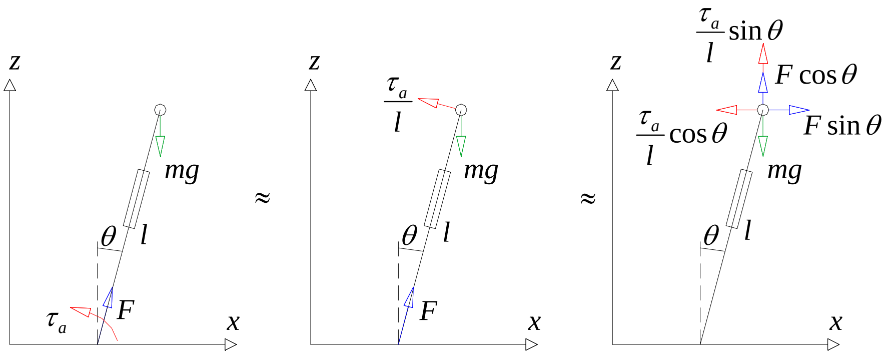


Figure 42: LIPM with ankle torque. The forces acting on the LIPM (left figure) can be analyzed as in the middle and right figure.

In Fig. 42, z axis is pointing upward, x axis is pointing forward, F is the ground reaction force, m is the mass of the model, g is the gravitational acceleration, l is the length of the leg, θ is the angle between vertical axis and the leg, and τ_a is the ankle torque. The dynamics of this model is derived as follows:

$$m\ddot{x} = F\sin\theta - \frac{\tau_a}{l}\cos\theta \qquad \text{(A.1)}$$

$$m\ddot{z} = F\cos\theta + \frac{\tau_a}{l}\sin\theta - mg \qquad \text{(A.2)}$$

The constraints for LIPM: $z = z_0$ and $\dot{z} = 0$ where z_0 is the constant COM height. From geometry, $\cos\theta = z_0/l$ and $\sin\theta = x/l$. Substituting these constraints to Eq. (A.2) and Eq. (A.1) yields:

$$\ddot{x} = \frac{g}{z_0}x - \frac{\tau_a}{mz_0} \qquad \text{(A.3)}$$

Appendix II: LIPM in lateral plane

The lateral plane has an equivalent LIPM with the sagittal plane, but with the horizontal y axis pointing to the left of the robot. The LIPM dynamics in the lateral plane are as follows:

$$y(t) = \cosh(wt)y_0 + \frac{1}{w}\sinh(wt)\dot{y}_0 - \frac{\tau_{ay}}{mz_0w^2}(\cosh(wt)-1) \tag{A.4}$$

$$v_y(t) = w\sinh(wt)y_0 + \cosh(wt)\dot{y}_0 - \frac{\tau_{ay}}{mz_0w}\sinh(wt) \tag{A.5}$$

where y_0 is the initial position of the LIPM with respect to the stance foot, $\dot{y}_0$ is the initial velocity, and τ_{ay} is a constant step input lateral torque. The lateral LIPM states at the end of step n $(y_T^{(n)}, v_{Ty}^{(n)})$, can be written:

$$y_T^{(n)} = \cosh(wT)y_0^{(n)} + \frac{1}{w}\sinh(wT)\dot{y}_0^{(n)} - \frac{\tau_{ay}}{mz_0w^2}(\cosh(wT)-1) \tag{A.6}$$

$$v_{Ty}^{(n)} = w\sinh(wT)y_0^{(n)} + \cosh(wT)\dot{y}_0^{(n)} - \frac{\tau_{ay}}{mz_0w}\sinh(wT) \tag{A.7}$$

where $y_0^{(n)}$ and $y_T^{(n)}$ are the initial and final positions of the COM at step n with respect to the stance foot, respectively; $v_{0y}^{(n)}$ and $\dot{y}_T^{(n)}$ are the initial and final linear velocities of the COM at step n, respectively.

Appendix III: Normal walking controller details

Normal walking implementation in lateral plane

Fig. 43 shows the two successive steps of the LIPM in lateral plane that is being considered.

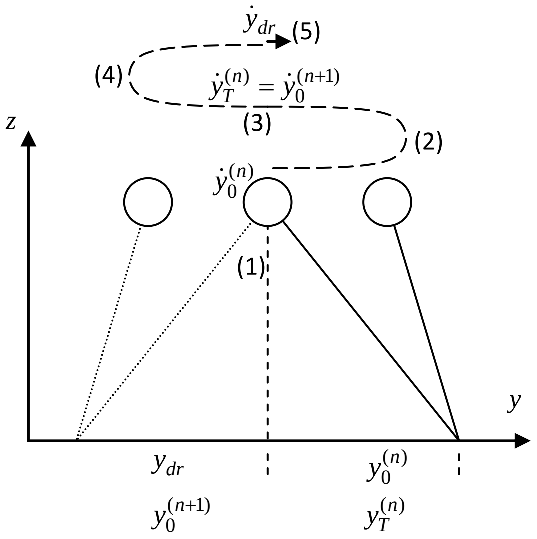


Figure 43: Two successive LIPM steps are considered in the normal walking controller. The figure shows the COM motion considered in the lateral plane at the right foot swing phase (dotted line indicates right leg). The number (1)-(5) indicates the motion sequence. The right foot support phase COM motion counterpart is similar with the right side but with opposite directions. The dashed arrows indicate the COM motion trajectory in the horizontal axis.

In the lateral plane there are two set of desired states for the support exchanges, one for the right foot swing phase $(y_{dr}, \dot{y}_{dr})$ $(y_{dr} > 0, \dot{y}_{dr} > 0)$ and another one for the right foot support phase $(y_{dl}, \dot{y}_{dl})$ $(y_{dl} < 0, \dot{y}_{dl} < 0)$. The desired state ($y_{dr} \equiv -y_{dl}$) is the desired COM position at support exchange at the end of step $n+1$ in the lateral plane. The desired COM velocity can be found using Eq. (A.6), while putting zero ankle torque:

$$\dot{y}_{dr} = -y_{dr}(1-\cosh(wT_n))w/\sinh(wT_n)$$

$$\dot{y}_{dl} = -y_{dl}(1-\cosh(wT_n))w/\sinh(wT_n)$$

Similar with the sagittal plane, the foot placement in the lateral plane during right foot swing phase is as follows:

$$y_0^{(n+1)} = \frac{\cosh(wT_n)(y_{dr} - {}^1\!/_w \sinh(wT_n)\dot{y}_0^{(n+1)}) + \sinh(wT_n)w(\dot{y}_{dr} - \cosh(wT_n)\dot{y}_0^{(n+1)})}{\cosh^2(wT_n) + w^2\sinh^2(wT_n)} \quad \text{(A.8)}$$

The desired states during right foot swing phase $(y_{dr}, \dot{y}_{dr})$ should be changed to $(y_{dl}, \dot{y}_{dl})$ during left foot swing phase. These states determined the step width and sway speed in the lateral plane.

Online foot and COM trajectory

In the implementation, the normal walking stepping time T_n consists of a single support and a double support time:

$$T_n = T_s + T_d$$

where T_s is the single support time and T_d is the double support time. The swing foot is arranged to lift up and swing at the single support time T_s only. The swing foot motion in sagittal and lateral plane is determined as follows:

$$x_{foot}(t) = x_{foot}{}^{(n-1)} + \frac{t}{T_s}(x_T^{(n-1)} - x_0^{(n)} + x_T^{(n)} - x_0^{(n+1)})$$

$$(\frac{T_d}{2} < t < \frac{T_d}{2} + T_s) \quad \text{(A.9)}$$

$$y_{foot}(t) = y_{foot}{}^{(n-1)} + \frac{t}{T_s}(y_T^{(n-1)} - y_0^{(n)} + y_T^{(n)} - y_0^{(n+1)})$$

$$(\frac{T_d}{2} < t < \frac{T_d}{2} + T_s) \quad \text{(A.10)}$$

where $x_{foot}{}^{(n-1)}$ and $y_{foot}{}^{(n-1)}$ is the position of the stance foot with respect to origin at step $n-1$. The foot motion in the vertical direction is shaped by a sinusoidal wave in order to have a smooth lift off and landing:

$$z_{foot}(t) = \frac{1}{2}Z_f(1+\sin(\pi\frac{t}{T_s}) - \frac{\pi}{2}) \qquad (\frac{T_d}{2} < t < \frac{T_d}{2} + T_s) \quad \text{(A.11)}$$

where Z_f is the stepping height constant.

The COM position at sagittal and lateral plane with respect to origin is calculated based on the LIPM equation of motion, as follows:

$$x_{COM}(t) = x_{foot}{}^{(n)} + \cosh(wt)x_0^{(n)} + \frac{1}{w}\sinh(wt)v_0^{(n)} - \frac{\tau_a}{mz_0w^2}(\cosh(wt)-1)$$

$$(0 < t < T_n) \quad \text{(A.12)}$$

$$y_{COM}(t) = y_{foot}{}^{(n)} + \cosh(wt)y_0^{(n)} + \frac{1}{w}\sinh(wt)v_{0y}^{(n)} - \frac{\tau_{ay}}{mz_0w}\sinh(wt)$$

$$(0 < t < T_n) \quad \text{(A.13)}$$

The overall flow of the normal walking controller can be seen in Fig. 44. At the beginning of stepping time, the foot placement is decided based on Eq. (18 and Eq. (A.8), using zero ankle torque and the default stepping time. Then the foot and COM trajectory are calculated for each sampling time using Eq. (A9-A13). The iteration of these process forms the normal walking gait.

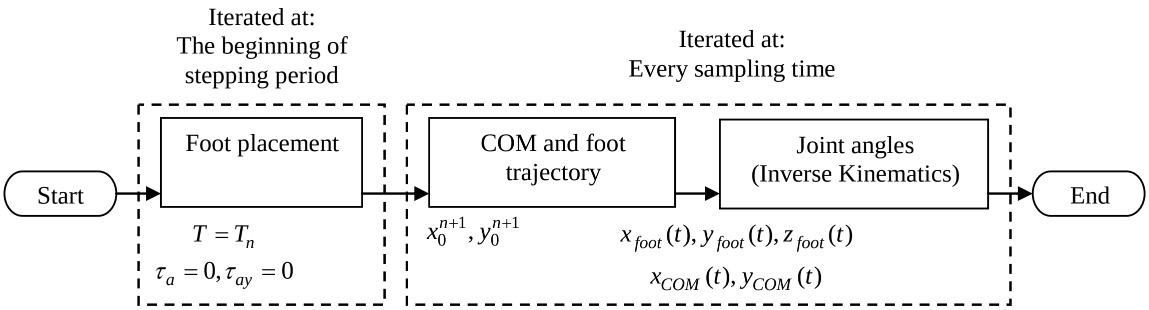


Figure 44: Normal walking controller

Appendix IV: Algorithm details

- ***Push Detection Algorithm***

Input: sensory feedback $\dot{x}(t)$ and $\dot{y}(t)$

Output: Determination of pushed state and priority plane

```
calculate $E_x(t)$ and $E_y(t)$ with Eq. (10)
calculate $E_x error(t)$ and $E_y error(t)$ with Eq. (11)
if $E_x error(t) > Ethreshold$ or $E_y error(t) > E_y threshold$ then
    return the biped is pushed
    if $E_x error(t) > E_y error(t)$ then
        return priority plane is sagittal plane
    else
        return priority plane is lateral plane
    endif
else
        return the biped is not pushed
endif
```

- ***Estimating Minimum Stepping time***

Input: sensory feedback $\dot{x}(t)$, $\dot{y}(t)$, and priority plane from Push Detection Algorithm

Output: Determine $T_{\lim}$

```
if priority plane is sagittal plane then
```

```
    **if** $v(t) \geq v_d$ **then**
        $T_{\lim} = \frac{T_n - T_k}{T_n} T_z + \frac{x_{COM}(t) - x_{foot}(t) - x_{forward}}{x_{backward} - x_{forward}} T_x$
    **else if** $v(t) < v_d$
        $T_{\lim} = \frac{T_n - T_k}{T_n} T_z + \frac{x_{forward} - (x_{COM}(t) - x_{foot}(t))}{x_{forward} - x_{backward}} T_x$
    **endif**
**else if** priority plane is lateral plane **then**
    **if** $\dot{y}(t) \geq \dot{y}_{dr}$ **then**
        $T_{\lim} = \frac{T_n - T_k}{T_n} T_z + \frac{y_{COM}(t) - y_{foot}(t) - y_{inwardr}}{y_{outwardr} - y_{inwardr}} T_y$
    **else if** $\dot{y}(t) < \dot{y}_{dr}$
        $T_{\lim} = \frac{T_n - T_k}{T_n} T_z + \frac{y_{outwardr} - (y_{COM}(t) - y_{foot}(t))}{y_{outwardr} - y_{inwardr}} T_y$
    **endif**
**endif**
```

Clarification for the above algorithm is as follows: $x_{forward}$ and $x_{backward}$ is the constraint on the forward and backward stepping in x coordinate with respect to the hip, respectively. $y_{outwardr}$ and $y_{inwardr}$ is the right foot outward and inward constraint on the stepping in y coordinate with respect to the hip, respectively. $y_{outwardl}$ and $y_{inwardl}$ is the left foot outward and inward stepping constraint in y coordinate with respect to the hip, respectively. T_n is the stepping time, T_k is the time that has elapsed in a stepping time when the biped enters pushed state, T_z is the minimum time required to lift up and land the swing foot in vertical motion. T_x and T_y are the minimum time required for the swing foot to travel the distances $x_{forward} - x_{backward}$ and $y_{outwardr} - y_{inwardr}$, respectively. T_z, T_x, and T_y are determined by hardware capability. This algorithm assumes that when a priority plane has been decided, the swing foot motion in the other plane is considered negligible. For the right foot support phase, $\dot{y}_{dr}$ and $y_{outwardr} - y_{inwardr}$ is replaced by $\dot{y}_{dl}$ and $y_{inwardl} - y_{outwardl}$, respectively.

- ***Control Policy Algorithm***

Input: priority plane from Push Detection Algorithm, minimum stepping time $T_{\lim}$, the biped state $(x(t), \dot{y}(t), y(t), \dot{y}(t))$

Output: Determine stepping time T and ankle torques $(\tau_{a,}\tau_{ay})$

```
$(x_p, \dot{x}_p, y_p, \dot{y}_p) \leftarrow (x(t), \dot{x}(t), y(t), \dot{y}(t))$
**if** priority plane is sagittal plane **then**
    $T \leftarrow \frac{1}{w} \ln \left( \frac{x_d + \sqrt{{x_d}^2 - \frac{1}{2}\left(x_p + \frac{\dot{x}_p}{w}\right)\left(x_p - \frac{\dot{x}_p}{w}\right)}}{\left(x_p + \frac{\dot{x}_p}{w}\right)} \right)$
    $\tau_a \leftarrow 0$
    **if** $T > T_{\lim}$ **and** $T < T_n$ **then**
        **return** case 1 push
    **else**
        $T \leftarrow T_{\lim}$
    $\tau_a \leftarrow (\cosh(wT)x_p + \frac{1}{w}\sinh(wT)\dot{x}_p - x_d)mz_0w^2 \frac{1}{(\cosh(wT)-1)}$
    **if** $\tau_a \geq \tau_{\lim}^{-}$ **and** $\tau_a \leq \tau_{\lim}^{+}$ **then**
        **return** case 2 push
    **else**
        **if** $\tau_a < \tau_{\lim}^{-}$
            $\tau_a \leftarrow \tau_{\lim}^{-}$
        **else if** $\tau_a > \tau_{\lim}^{+}$
            $\tau_a \leftarrow \tau_{\lim}^{+}$
        **endif**
        $x_T^{(n)} \leftarrow \cosh(wT)x_p + \frac{1}{w}\sinh(wT)\dot{x}_p - \frac{\tau_a}{mz_0w^2}(\cosh(wT)-1)$
            **if** $x_T^{(n)} > x_{\lim}^{-}$ **and** $x_T^{(n)} < x_{\lim}^{+}$ **then**
                **return** case 3 push
            **else**
                **return** case 4 push
            **endif**
    **endif**
**endif**
$\tau_{ay} \leftarrow (\cosh(wT)y_p + \frac{1}{w}\sinh(wT)\dot{y}_p - y_{dr})mz_0w^2 \frac{1}{(\cosh(wT)-1)}$
**if** $\tau_{ay} < \tau_{\lim}^{-}$
    $\tau_{ay} \leftarrow \tau_{\lim}^{-}$
    **else if** $\tau_{ay} > \tau_{\lim}^{+}$
        $\tau_{ay} \leftarrow \tau_{\lim}^{+}$
    **endif**
**else if** priority plane is lateral plane **then**
        $T \leftarrow \frac{1}{w}\ln\left(\frac{y_{dr} + \sqrt{{y_{dr}}^{2} - \frac{1}{2}\left(y_p + \frac{\dot{y}_p}{w}\right)\left(y_p - \frac{\dot{y}_p}{w}\right)}}{\left(y_p + \frac{\dot{y}_p}{w}\right)}\right)$
    $\tau_{ay} \leftarrow 0$
        **if** $T > T_{\lim}$ **and** $T < T_n$ **then**
        **return** case 1 push
    **else**
        $T \leftarrow T_{\lim}$
        $\tau_{ay} \leftarrow (\cosh(wT) y_p + \frac{1}{w}\sinh(wT)\dot{y}_p - y_{dr}) m z_0 w^2 \frac{1}{(\cosh(wT) - 1)}$
        **if** $\tau_{ay} \geq \tau_{\lim}^{-}$ **and** $\tau_{ay} \leq \tau_{\lim}^{+}$ **then**
            **return** case 2 push
        **else**
            **if** $\tau_{ay} < \tau_{\lim}^{-}$
                $\tau_{ay} \leftarrow \tau_{\lim}^{-}$
            **else if** $\tau_{ay} > \tau_{\lim}^{+}$
                $\tau_{ay} \leftarrow \tau_{\lim}^{+}$
            **endif**
```

```
49:             $y_T^{(n)} \leftarrow \cosh(wT)y_p + \frac{1}{w}\sinh(wT)\dot{y}_p - \frac{\tau_{ay}}{mz_0 w^2}(\cosh(wT)-1)$
50:             if $y_T^{(n)} > y_{\lim r}^{-}$ and $y_T^{(n)} < y_{\lim r}^{+}$ then
51:                 return case 3 push
52:             else
53:                 return case 4 push
54:             endif
55:         endif
56:     endif
57:     $\tau_a \leftarrow (\cosh(wT)x_p + \frac{1}{w}\sinh(wT)\dot{x}_p - x_d)mz_0 w^2 \frac{1}{(\cosh(wT)-1)}$
58:     if $\tau_a < \tau_{\lim}^{-}$
59:         $\tau_a \leftarrow \tau_{\lim}^{-}$
60:     else if $\tau_a > \tau_{\lim}^{+}$
61:         $\tau_a \leftarrow \tau_{\lim}^{+}$
62:     endif
63: endif
```

Some clarifications for the above algorithm are as follows: the time to reach the desired COM position in line 3 and line 34 can be solved from Eq. (4) and Eq. (A.6), with zero ankle torque, respectively. The ankle torque τ_a in line 9 and 57 is obtained by substituting $x_T^{(n)} = x_d$ to Eq. (4). The lateral ankle torque τ_{ay} in line 27 and 40 is obtained by substituting $y_T^{(n)} = y_{dr}$ to Eq. (A.6). The algorithm for the right foot support phase is done by replacing all y_{dr} with y_{dl} and replacing the right foot constraints into the respective left foot constraints.

- ***Foot Placement Algorithm***

Input: stepping time T, ankle torques $(\tau_{a,}\tau_{ay})$ from *Control Policy Algorithm*, the biped state $(x(t), v(t), y(t), v_y(t))$

Output: Determine the foot placement in sagittal plane $x_0^{(n+1)}$ and in lateral plane $y_0^{(n+1)}$

```
$(x_p, \dot{x}_p, y_p, \dot{y}_p) \leftarrow (x(t), \dot{x}(t), y(t), \dot{y}(t))$
$v_T^{(n)} \leftarrow w\sinh(wT)x_p^{(n)} + \cosh(wT)\dot{x}_p^{(n)} - \frac{\tau_a}{mz_0 w}\sinh(wT)$
$x_0^{(n+1)} \leftarrow \frac{\cosh(wT_n)(x_d - {}^1\!/_w \sinh(wT_n)\dot{x}_T^{(n)}) + \sinh(wT_n)w(\dot{x}_d - \cosh(wT_n)\dot{x}_T^{(n)})}{\cosh^2(wT_n) + w^2\sinh^2(wT_n)}$
**if** $x_0^{(n+1)} < x_{\lim}^{-}$
    $x_0^{(n+1)} \leftarrow x_{\lim}^{-}$
**else if** $x_0^{(n+1)} > x_{\lim}^{+}$
    $x_0^{(n+1)} \leftarrow x_{\lim}^{+}$
**endif**
$v_{Ty}^{(n)} \leftarrow w\sinh(wT)y_p + \cosh(wT)\dot{y}_p - \frac{\tau_{ay}}{mz_0 w}\sinh(wT)$
$y_0^{(n+1)} \leftarrow \frac{\cosh(wT_n)(y_{dr} - {}^1\!/_w \sinh(wT_n)\dot{y}_T^{(n)}) + \sinh(wT_n)w(\dot{y}_{dr} - \cosh(wT_n)\dot{y}_T^{(n)})}{\cosh^2(wT_n) + w^2\sinh^2(wT_n)}$
**if** $y_0^{(n+1)} < y_{\lim r}^{-}$
    $y_0^{(n+1)} \leftarrow y_{\lim r}^{-}$
```

```
else if $y_0^{(n+1)} > y_{\lim r}^{+}$
    $y_0^{(n+1)} \leftarrow y_{\lim r}^{+}$
endif
```

- ***Foot Placement Compensator Algorithm***

Input: priority plane from Push Detection Algorithm, stepping time T from Control Policy Algorithm, the foot placement in sagittal plane $x_0^{(n+1)}$ and in lateral plane $y_0^{(n+1)}$ from Foot Placement Algorithm, the angular velocity in sagittal plane $\omega(t)$ and in lateral plane $\omega_y(t)$, the posture angle in sagittal plane $\phi(t)$ and in lateral plane $\phi_y(t)$.

Output: Determine the foot placement compensation values x_{fc}, y_{fc}, z_{fc}

```
$(\phi_0, \phi_{0y}, \omega, \omega_y) \leftarrow (\phi(t), \phi_y(t), \omega(t), \omega_y(t))$
if priority plane is sagittal plane then
    $\phi_{fc} \leftarrow \phi_0 + \omega T$
    $\phi_l \leftarrow \tan^{-1}\left(\frac{z_0}{x_T^{(n)}}\right)$
    $z_{fc} \leftarrow \left(\frac{x_T^{(n)}}{\cos\phi_l}\cos\left(\phi_l - \phi_{fc}\right) - x_0^{(n+1)}\right)\sin\phi_{fc}$
    $x_{fc} = \frac{z_{fc}}{\tan\phi_{fc}} - \left(x_T^{(n)} - x_0^{(n+1)}\right)$
else if priority plane is lateral plane then
    $\phi_{fc} \leftarrow \phi_{0y} + \omega_y T$
    $\phi_l \leftarrow \tan^{-1}\left(\frac{z_0}{y_T^{(n)}}\right)$
    $z_{fc} \leftarrow \left( \frac{y_T^{(n)}}{\cos\phi_l} \cos\left(\phi_l - \phi_{fc}\right) - y_0^{(n+1)} \right) \sin\phi_{fc}$
    $y_{fc} \leftarrow \frac{z_{fc}}{\tan\phi_{fc}} - \left( y_T^{(n)} - y_0^{(n+1)} \right)$
endif
```

In the implementation, the compensation values are also constrained to some constants to keep them within the kinematic operation limit of the legs. The implementation details are as follows:

- ***Foot placement compensator implementation***

Input: foot placement compensation values x_{fc}, y_{fc}, z_{fc}

Output: updated swing foot position

```
if ($T_k > \frac{T_d}{2}$ and $T_k \le \frac{T_d}{2} + T_s$) then
    $z_{foot} \leftarrow \frac{1}{2} Z_f (1 + \sin(\pi \frac{t}{T_s}) - \frac{\pi}{2})$
    if ($T_k > \frac{T_d}{2} + \frac{1}{2} T_s$ and $z_{foot} < z_{fc}$) then
        $z_{foot} \leftarrow z_{fc}$
    endif
    $x_{foot} \leftarrow x_{foot}^{(n-1)} + \frac{T_k}{T_s} (x_T^{(n-1)} - x_0^{(n)} + x_T^{(n)} - x_0^{(n+1)} + x_{fc})$
    $y_{foot} \leftarrow y_{foot}^{(n-1)} + \frac{T_k}{T_s} (y_T^{(n-1)} - y_0^{(n)} + y_T^{(n)} - y_0^{(n+1)} + y_{fc})$
endif
if ($T_k > \frac{T_d}{2} + T_s$ and $T_k \le T$) then
```

```
10:        $z_{foot} \leftarrow z_{fc} - z_{fc}\left(\frac{T_k - 0.5T_d - T_s}{0.5T_d}\right)$
11:    $x_{foot} \leftarrow x_{foot}^{(n-1)} + (x_T^{(n-1)} - x_0^{(n)} + x_T^{(n)} - x_0^{(n+1)} + x_{fc}) - x_{fc}\left(\frac{T_k - 0.5T_d - T_s}{0.5T_d}\right)$
12:    $y_{foot} \leftarrow y_{foot}^{(n-1)} + (y_T^{(n-1)} - y_0^{(n)} + y_T^{(n)} - y_0^{(n+1)} + y_{fc}) - y_{fc}\left(\frac{T_k - 0.5T_d - T_s}{0.5T_d}\right)$
```

In this algorithm, T_k is the time that has elapsed in a stepping time, T_s is the single support time and T_d is the double support time. In line 1 to 7, which is the single support time, the swing foot compensation is applied. With the compensation, the biped should be able to land at the proper place relative to the COM. Then in line 9 to 12, which is the second half of the double support time, the foot is returned to its original placement. This approach effectively restores the biped's hip height and step length to its default value.

Appendix V: Realistic humanoid robot model details

Dimensions

Fig. 45 shows the dimensions of the realistic bipedal model.

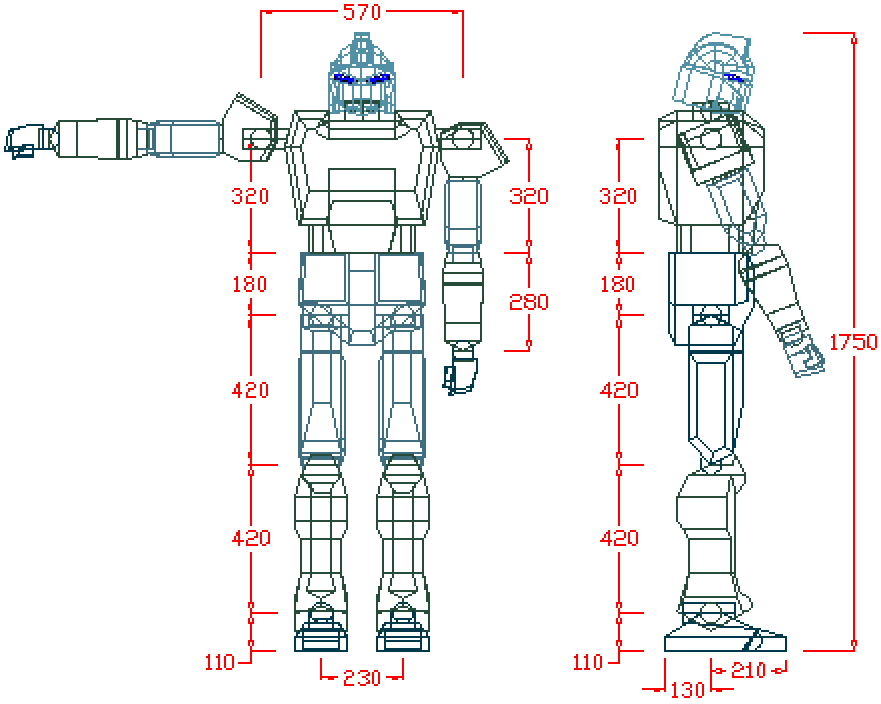
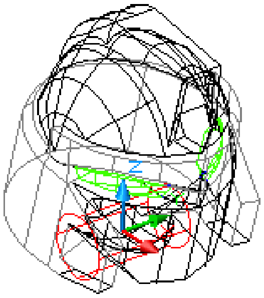
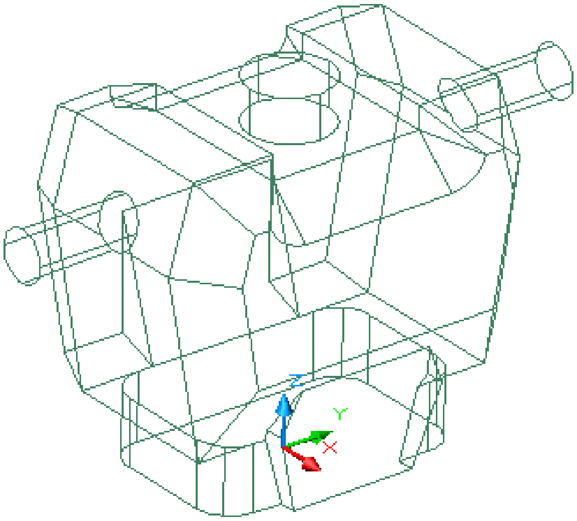
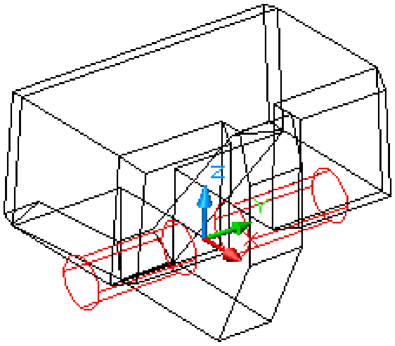
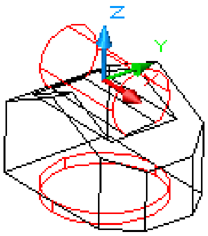


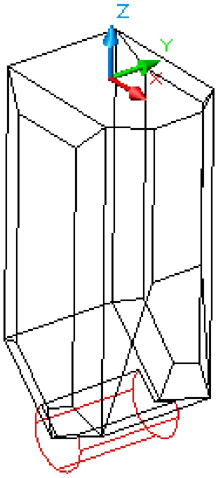
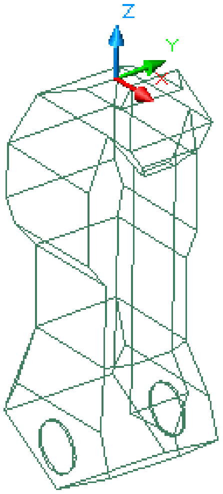
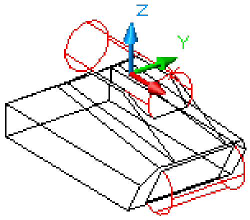
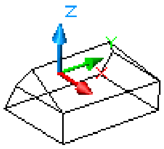
Figure 45: Simulated bipedal robot dimensions (in mm)

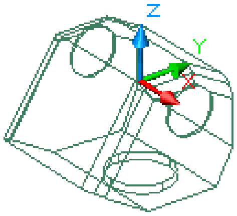
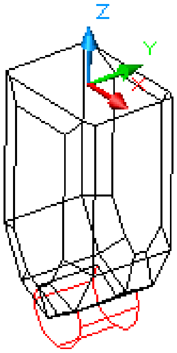
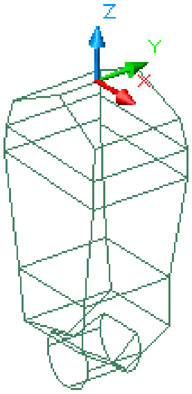
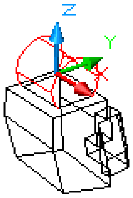
Mass, COM location, and Inertia

Table 7 shows the parts of the model and its COM location ($\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix}$) and inertia matrices ($I$). The COM locations are located with respect to the origin of the part (the origin of the coordinate system on the right figure).

Table 7: Simulated bipedal robot model center of mass and inertia matrices

Head $mass = 3.09Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0.01 & 0 & 0.13 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.0178 & 0 & 0 \\ 0 & 0.0179 & 0 \\ 0 & 0 & 0.0012 \end{bmatrix} (Kgm^2)$	
Torso $mass = 13Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.23 \end{bmatrix} (m)$ $I = \begin{bmatrix} 1.25 & 0 & 0 \\ 0 & 0.89 & 0 \\ 0 & 0 & 0.46 \end{bmatrix} (Kgm^2)$	
Pelvis $mass = 26.44Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} -0.01 & 0 & 0.08 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.7 & 0 & 0 \\ 0 & 0.53 & 0 \\ 0 & 0 & 0.54 \end{bmatrix} (Kgm^2)$	
Hip $mass = 2.54Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -0.01 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.02 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} (Kgm^2)$	

Thigh $mass = 4.69Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0 & 0.01 & -0.17 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.19 & 0 & 0 \\ 0 & 0.19 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} (Kgm^2)$	
Shank $mass = 8.63Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0.01 & 0 & -0.31 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.95 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.03 \end{bmatrix} (Kgm^2)$	
Foot $mass = 2.2Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0.01 & 0 & -0.06 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.02 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} (Kgm^2)$	
Toe $mass = 0.53Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0.013 & 0 & 0.012 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.0014 & 0 & 0 \\ 0 & 0.0009 & 0 \\ 0 & 0 & 0.002 \end{bmatrix} (Kgm^2)$	

Shoulder $mass = 1.09Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0.002 & 0 & -0.113 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.0178 & 0 & 0 \\ 0 & 0.0179 & 0 \\ 0 & 0 & 0.0012 \end{bmatrix} (Kgm^2)$	
Upper Arm $mass = 0.73Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} 0.0002 & 0 & -0.0066 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.0053 & 0 & 0 \\ 0 & 0.0049 & 0 \\ 0 & 0 & 0.0011 \end{bmatrix} (Kgm^2)$	
Lower Arm $mass = 1.19Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} -0.012 & 0 & -0.165 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.0044 & 0 & 0 \\ 0 & 0.0439 & 0 \\ 0 & 0 & 0.0011 \end{bmatrix} (Kgm^2)$	
Hand $mass = 0.43Kg$ $\begin{bmatrix} x_{com} & y_{com} & z_{com} \end{bmatrix} = \begin{bmatrix} -0.061 & 0 & -0.0749 \end{bmatrix} (m)$ $I = \begin{bmatrix} 0.0036 & 0 & 0 \\ 0 & 0.0032 & 0 \\ 0 & 0 & 0.0005 \end{bmatrix} (Kgm^2)$	

Appendix VI: Description of NUSBIP-III ASLAN

Brief History

There has been numerous bipedal robot in different sizes developed as the platforms of researches by the Legged locomotion Group (LLG) of National University of Singapore (NUS). Among the smaller platforms are the RO-PE I-VI series, which has been participating in Robocup kid size. Besides this smaller platform, LLG also has been developing the human-sized bipedal series, called NUSBIP.

The NUSBIP-III ASLAN is the latest, third generation of NUSBIP series. It has been developed since early 2008. It is developed mainly as a general platform for bipedal walking research.

Current Development

ASLAN significantly improves the existing physical bipedal robot, NUSBIP-II, especially in the physical structure and the actuator subsystem. The structure of the legs has been improved and the joints are upgraded using the harmonic drives system, which gives excellent power and accuracy with zero backlash. The servos are controlled by ELMO motor drivers, connected to the main PC 104 microprocessor via CAN bus system. By using these systems, ASLAN has achieved stable dynamic walking motions. Next, two arms and one waist joint have been added on the body, and new sensors have been added into the system. Fig. 46 shows the mechanical design and the early realization of ASLAN.

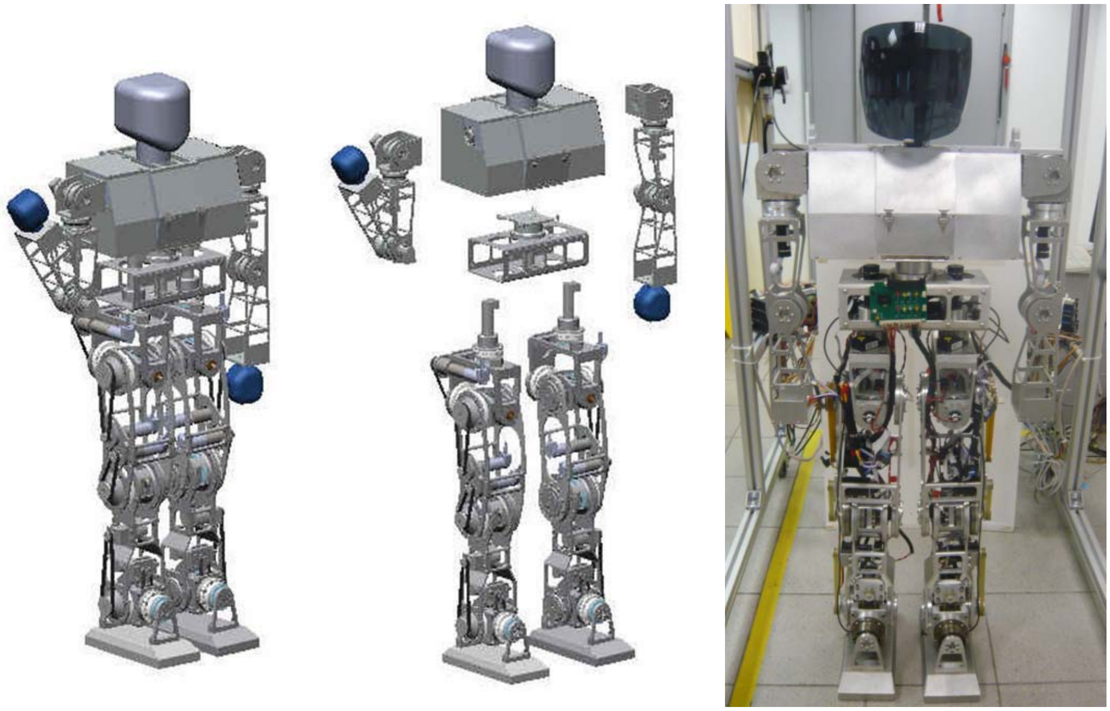


Figure 46: Mechanical drawing and realization of NUSBIP-III ASLAN

ASLAN is a humanoid robot modeled after a teenager. It has a trunk with two legs, two arms and one waist joint. Its weight is approximately 60kg and hip height is around 0.7m when the robot is standing. The general specifications of ASLAN are shown in Table 1.

Similar with the realistic biped descript in appendix III, ASLAN has six DOFs on each leg: three at the hip, one at the knee, and two at the ankle; four degrees of freedom on each arm: three at the shoulder, one at elbow. The DOFs at the hip allow the leg to twist and adduct/abduct, as well as swing forward and backward. The DOF at the knee allows the leg to flex. The DOFs at the ankle allow the foot to pitch and roll. Fig. 47 shows the leg configuration.

Table 8: Specification of NUSBIP-III ASLAN

Height	1350mm
Width	550mm
Weight	60Kg
Walk speed	0.3m/s
Actuator	servomotor + harmonic gear + drive unit
Control Unit	PC/104 + ELMO + CAN bus system
Operation system	Windows XP RTX

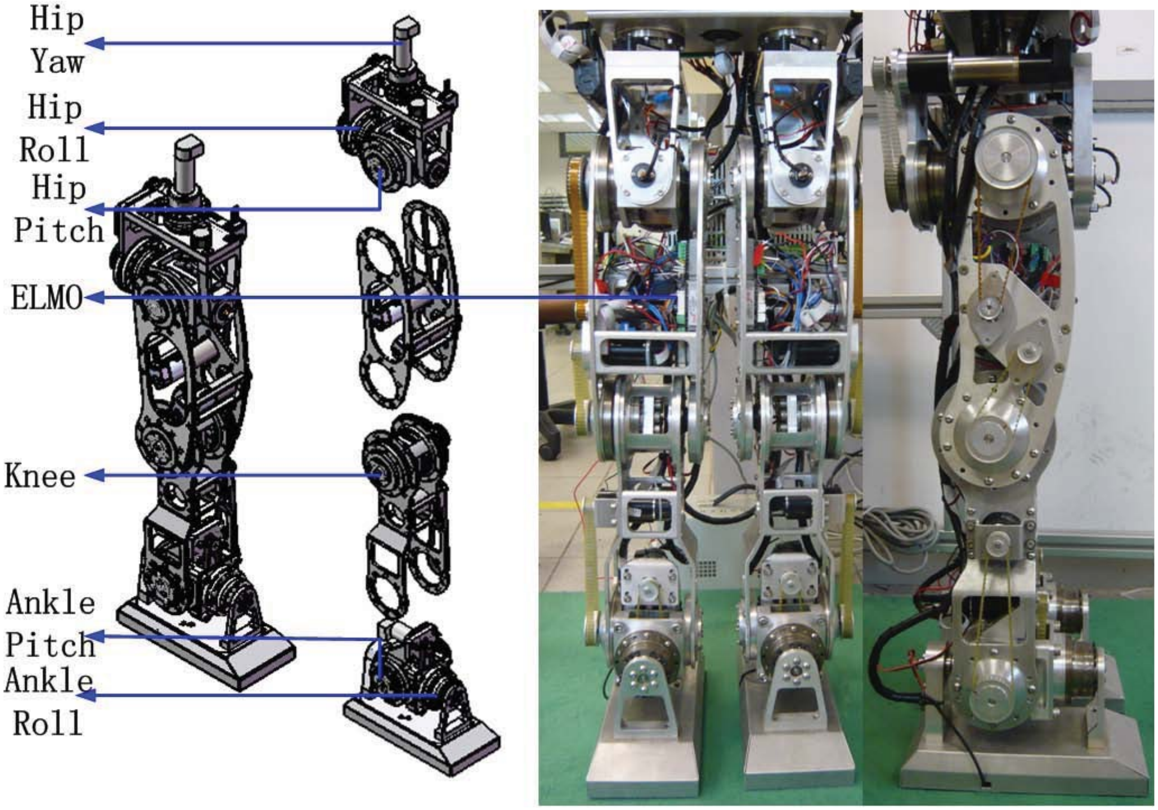


Figure 47: NUSBIP-III ASLAN legs.

The torso is designed with strategic sensory system, battery, and main processors placement in mind. The main processor is located at the top center section of the chest, providing ventilation from above the torso. The inertial sensory system such as gyros and accelerometers are designed to be placed in the middle chest section as well, above the COM. The battery is placed in the belly, very near to the COM, with a hatch in front of the chest for easy access. The side areas of the chest are used to storage other hardware and ELMO motor drivers. Figure 48 shows the torso design.

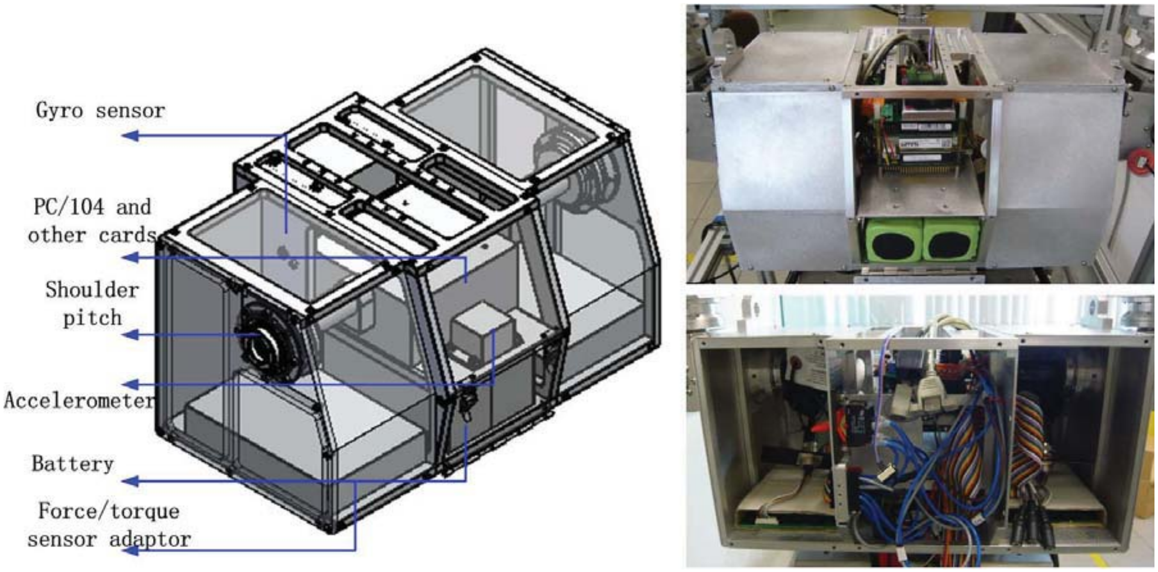


Figure 48: NUSBIP-III ASLAN torso design.

Several off-line walking algorithms have been tested on ASLAN, such as the ZMP preview control by Kajita *et al.* [36]. Several task such as walking, turning, climbing a known slope and stair has been realized. However, an off-line walking algorithm is not ideal for long term robust walking development.

In 2010, the normal walking strategy presented in this thesis (Appendix III) was implemented for ASLAN, albeit without the inertial sensory feedback system. Some basic behavior has been successfully developed. It is able to do forward walking, backward walking, turning, side stepping, and kicking. In June 2010, ASLAN participated in the ROBOCUP humanoid adult size category, where our team, team RO-PE, manage to won the first prize for the adult size soccer competition and the adult size technical challenge. Figure 49 shows ASLAN in a soccer match against other bipedal robot during ROBOCUP 2010.

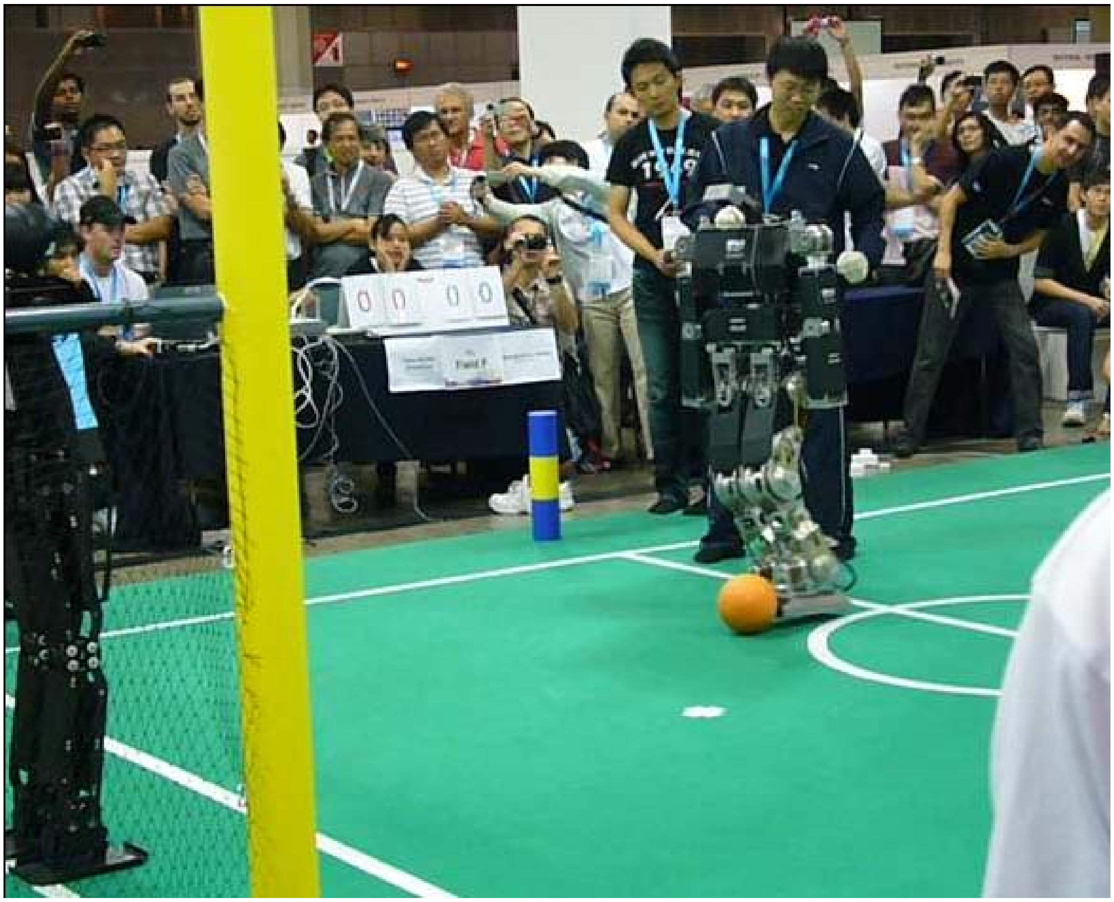


Figure 49: NUSBIP-III ASLAN kicking for goal in ROBOCUP 2010 finale.

Potential future plans

Several improvements are required in order to realize the robust walking and the push recovery capability presented in this thesis. First, is the implementation of a reliable sensory system, which is crucial for the push detection and the decision making of the algorithm. Second, a fast walking behavior needs to be realized. Currently, ASLAN is walking with 0.64s stepping time, which is very close to its minimum stepping time. As discussed in section 3, a fast stepping time is important for the performance of the push recovery. A possible solution would be to implement the brushless motors for the knees and ankles, which could improve the maximum joint speed and acceleration. Third, the weight of the legs needs to be reduced. Currently ASLAN’s COM is too low, which makes fast dynamic walking with big steps very difficult. Mechanical modifications are currently in progress.