6-2020

Push Recovery for Humanoid Robots using Linearized Double Inverted Pendulum

Saurav Singh
ss3337@rit.edu

Push Recovery for Humanoid Robots using Linearized Double Inverted Pendulum

by

Saurav Singh

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science
in Electrical and Microelectronic Engineering

Supervised by

Professor Dr. Ferat Sahin
Department of Electrical and Microelectronic Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
June 2020

Approved by:

Dr. Ferat Sahin, Professor
Thesis Advisor, Department of Electrical and Microelectronic Engineering

---

Dr. Jamison Heard, Assistant Professor
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Gill Tsouri, Professor
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Ferat Sahin, Professor
Department Head, Department of Electrical and Microelectronic Engineering

Dedication

I dedicate this work to my parents, Ravindra Singh and Indu Singh, for their immeasurable and undying love and support.

Acknowledgments

Dr. Ferat Sahin, my thesis advisor, has been an inspiration and motivator for majority of my thesis work. This work would not have been possible without his invaluable guidance and support. He always encouraged me to solve a complex problem with the simplest solution. I am extremely grateful to him for giving me this opportunity to work with him.

I am also grateful to all the members of Multi-Agent Biorobotics Laboratory for all the crucial discussions and fun brainstorming sessions that shaped my research. A special thanks to Shitij P. Kumar for guiding me at every step of this work. Spending time with him really taught me what it means to do research.

I am extremely grateful to my friends Lipisha, Rashmi and Srinath for standing beside me through thick and thin during the course of my master's.

I thank Rochester Institute of Technology for providing me with resources, labs, and materials for the progress of the research.

Lastly, I would like to thank my parents for always being there for me. They may not understand what I do but they never fail to show their unwavering support to me.

Abstract

Biped robots have come a long way in imitating a human being's anatomy and posture. Standing balance and push recovery are some of the biggest challenges for such robots. This work presents a novel simplified model for a humanoid robot to recover from external disturbances. The proposed Linearized Double Inverted Pendulum, models the dynamics of a complex humanoid robot that can use ankle and hip recovery strategies while taking full advantage of the advances in controls theory research. To support this, an LQR based control architecture is also presented in this work. The joint torque signals are generated along with ankle torque constraints to ensure the Center of Pressure stays within the support polygon. Simulation results show that the presented model can successfully recover from external disturbances while using minimal effort when compared to other widely used simplified models. It optimally uses the the torso weight to generate angular momentum about the pelvis of the robot to counter-balance the effects of external disturbances. The proposed method was validated on simulated 'TigerBot-VII', a humanoid robot.

List of Contributions

- Proposed a novel simplified model, *Linearized Double inverted Pendulum (LDIP)* for standing balance and push recovery of humanoid robots that can use ankle and hip recovery strategies while taking full advantage of the advances in optimal controls theory research. To support this, an LQR based control architecture is also presented in this work.

- Presented an approximate relationship between maximum ankle torque and maximum balance force at the hip to comply with Center of Pressure stability Criteria.

- The LDIP model is designed for TigerBot-VII, a humanoid robot engineered and built at Rochester Institute of Technology, New York. The proposed model can help the robot to use hip and ankle strategies for standing balance and push recovery.

- Worked on the hardware of TigerBot-VII, analysed some of the design flaws and proposed future modifications and improvements.

Publication Submission:

S. Singh, F. Sahin, "Push Recovery for Humanoid Robots based on Linearized Double Inverted Pendulum", *in IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humanoid robots are largely devised for applications with mundane and routine tasks that can be automated using the existing infrastructure specifically meant for humans. These robots may need to walk or stand on uneven terrain or even bump into potential obstacles in the environment. For these reasons, standing balance and push recovery in humanoid robots is an important field of research, with many questions yet to be answered. They are highly susceptible to external disturbances and tend to fall over. The controls problem becomes more complex as these robots are generally high degree of freedom, non-linear dynamics, small support polygon, under-actuated systems with strong coupling between joints. Representing the robot using a simplified model enables us to facilitate real-time control of the biped robot.

1.1 Push Recovery Strategies

Achieving stability of the robot is a challenging task and implementation to recover from say a push or an external disturbance is done using three strategies: ankle, hip and stepping. When small disturbances are present, the robot

can use its ankle joints only to balance itself and recover from the push. For larger disturbances, the robot can use ankle and hip joints to deal with the external disturbances like humans do. For even larger disturbances, it may not be possible to recover from external disturbances just by using ankle and hip strategies and the robot will then need to take a step to avoid falling. Thus, ankle and hip strategies can be used to maintain balance for smaller disturbances, while stepping strategies are used for larger disturbances, shown in Fig. 1.1.

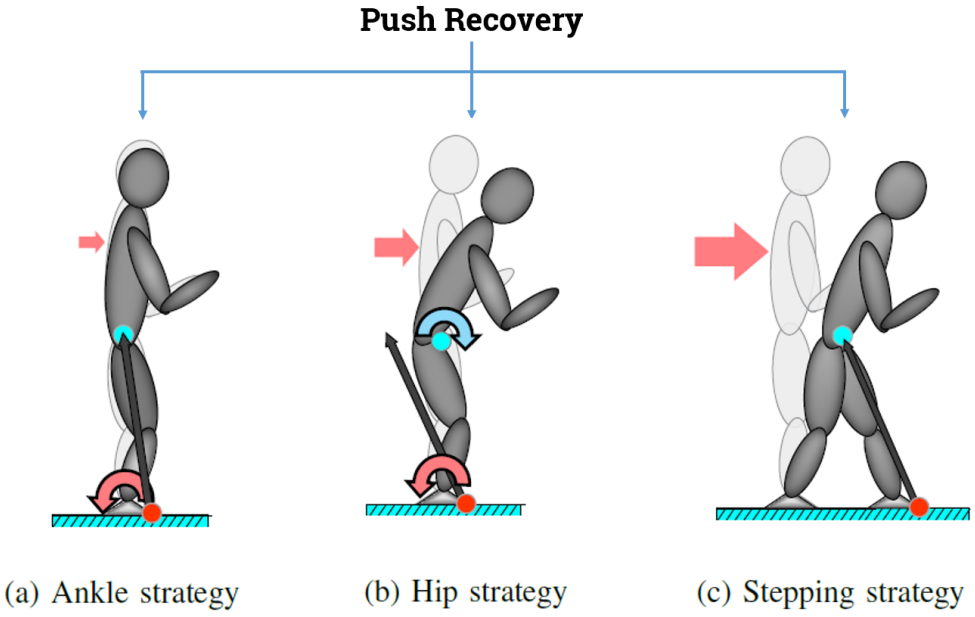


Figure 1.1: Push recovery strategies for humanoid robots.

This work proposes a *Linearized Double Inverted Pendulum Model (LDIP)* that incorporates both ankle and hip strategies for standing balance and push recovery.

1.2 Simplified Models for Humanoid Robots

A simplified model is essential to understand the dynamics of the real robot. It is easier to then control this simplified model rather than the high Degree of Freedom (DoF) robot with complex dynamics. The research proposed in this work is motivated by the in-house robot TigerBot-VII [1][1]. TigerBot-VII is a complex 14 DoF humanoid robot with IMU sensors on the torso and six-axis force/torque sensors on the ankle joints. The model proposed in this work is to achieve balance and push recovery on this robot.

One of the most commonly used simplified model is the *Linear Inverted Pendulum Model (LIPM)* [2] [3] and has been further discussed in Chapter 2. In this model the entire mass of the robot is concentrated at its Center of Mass (CoM) and the legs are assumed to have zero mass. This model also imposes a constraint to maintain the CoM at a constant height and only allows motion in forward/backward direction. When humans try to maintain their balance, they generally rotate their arms and lunge their torso forward/backward in an attempt to balance themselves. To capture this behaviour, an extension of LIPM was later proposed called *Linear Inverted Pendulum Model with flywheel (LIPM-FW)* [4] where the centroid angular momentum is explicitly modeled. Linearized Double Inverted Pendulum model (LDIP) is built upon the concept of LIPM with flywheel but the angular momentum is an implicit design of the model, described in Chapter 2.

The proposed model is built upon the idea to redistribute the mass of the robot's upper and lower body to optimally control the torso and generate an-

[1]It is a humanoid robot designed and built at Rochester Institute of Technology, New York https://youtu.be/lZBe5EtFKaQ

gular moment about its pelvis. This angular momentum mimics the human behavior of lunging their torso forward/backward while balancing. In essence, the angular momentum is implicitly modeled in LDIP and the angular position and velocity of the torso are a part of the system state variables. This further enables us to control and use the torso optimally using modern control techniques which takes into consideration the internal states of the system. In order to control and actuate the robot in real-time, it is important to not only have a simplified model representation but also one that is more realistically close to the actual robot. The LDIP model considers this trade-off and successfully balances the robot using minimal effort at the joints.

The rest of the thesis report is organized as follows: Chapter 2 presents related work in the field of humanoid balancing and push recovery, Chapter 3 proposes the model definition and equations for LDIP. Chapter 4 presents a control architecture based on LDIP model for push recovery and standing balance. Chapter 5 gives an overview of TigerBot-VII used for validation of the proposed model. Chapter 6 presents the specifications of TigerBot-VII which is used as a test platform, comparison between LDIP and other proposed models in simulation and discusses the findings. Chapter 7 concludes this work and presents the advantages of the proposed model. Chapter 8 lays out the future work for this research and discusses issues and recommended modifications for TigerBot-VII robot hardware.

Chapter 2

Literature Survey

This Chapter covers some of the many simplified models proposed by various researchers and their implementations in the humanoid robot community. Stability criteria and standing balance for bipedal robots is a vast field of research. Based on the various stability criteria defined, a lot of research has been conducted proposing several simplified models that could capture a humanoid robot dynamics. Some of these criteria include Zero Moment Point (ZMP) [2] [5] [6] [7] [8], Center of Pressure (CoP) [7] [8], Capture Point (CP) [4], and Foot Rotation Indicator (FRI) [7].

2.1 Stability Criteria

In humanoid literature, many researchers have proposed different stability criteria's for these robots which can be used to identify whether the robot is stable and maintaining its balance or not. When a humanoid robot is moving, it needs to make sure that there is always a contact between it's

sole and the ground. This means, given a motion of a humanoid robot, we need to determine whether or not there is a contact between the sole and the ground and plan the motions to ensure this contact is maintained. Generally, a humanoid robot is made to comply to one of the stability criteria for these kind of purposes. For this, we first need to understand what is a support polygon of a humanoid robot.

Support Polygon is a convex hull, which is the smallest convex set including all contact points of the robot with ground. To get an intuition, if we take an elastic cord and enclose all the contact points between the robot and the ground, the polygon then formed will be called the support polygon, shown in Figure 2.1.

Figure 2.1: Support Polygon of a humanoid robot.

2.1.1 Zero Moment Point

Zero Moment Point or ZMP is a point on the ground at which the horizontal moment generated by the ground reaction force/torque equals zero [2].

In Figure 2.2, the sole of the robot and the ground are in contact and the arrows below represent the ground reaction force distribution. The point on the ground where the resultant moment of all the ground reaction forces is

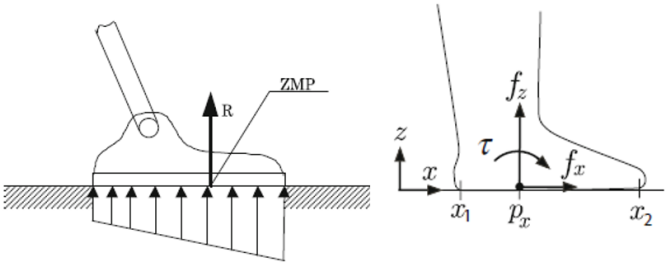


Figure 2.2: Zero Moment Point (ZMP).

zero, will be called the zero-moment point. For the robot to be stable, the Zero moment point must always stay inside the support polygon. If this is not true, the robot will have a tendency to topple over.

2.1.2 Center of Pressure

The field of pressure forces (normal to the sole) is equivalent to a single resultant force, exerted at the point where the resultant moment is zero [8]. This point is called Center of Pressure or CoP. The main difference between ZMP and CoP is that CoP is linked to forces exerted by contact while ZMP is linked to forces transmitted without contact.

2.1.3 Foot Rotation Indicator

The Foot Rotation Indicator or FRI point is a point on the foot–ground contact surface, within or outside the support polygon, where the net ground reaction force would have to act to achieve a zero moment condition about the foot with respect to the FRI point itself [7]. The FRI point coincides with the ZMP and CoP when the foot is stationary, and diverges from the ZMP for non-zero rotational foot accelerations. This can be observed in Figure 2.3

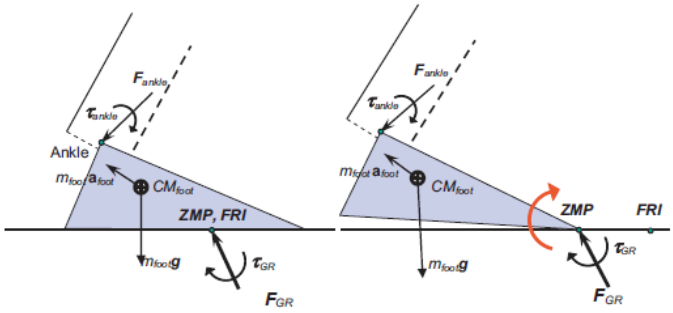


Figure 2.3: Foot Rotation Indicator (FRI).

2.2 Existing Simplified Models for Humanoid Robots

2.2.1 Linear Inverted Pendulum Model

The most commonly used model in humanoid literature is *Linear Inverted Pendulum Model (LIPM)* [2] [3], shown in Figure 2.4. This model makes three assumptions: (i) all the mass of the robot is concentrated at its center of mass (CoM); (ii) the robot has zero mass legs, whose tips contact the ground at single rotating joints; (iii) only the forward/backward and the up/down motions of the robot is considered, neglecting lateral motion. LIPM also constraints the CoM to a constant height. This model is popular among the humanoid research community because the model is inherently linear which makes it easy to control.

The dynamics of the model is described by the following equation:

$$\ddot{x} = \frac{g}{z_0} x \tag{2.1}$$

where the z_0 is the constant height of the CoM and x is the displacement of the CoM in horizontal direction. S. Faraji et al. [9] used LIPM with a stepping

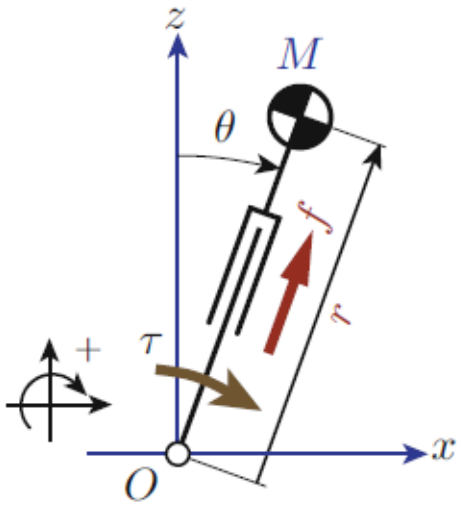


Figure 2.4: Linear Inverted Pendulum Model.

strategy to recovery from large external disturbances which was based on time projection control.

2.2.2 Spherical Inverted Pendulum Model

E. Ahmed et al. [10] proposed a *Spherical Inverted Pendulum Model (SIP)* that does not have to follow the constraint to maintain a constant height for CoM, shown in Figure 2.5. This helps their model to control and generate a more natural motion.

The dynamics of the model is described by the following equation:

$$\ddot{\theta} = \frac{g}{l}\theta \tag{2.2}$$

where the l is the length of fixed pendulum arm and θ is the angular displacement of the CoM. The authors also proposed an energy based controller

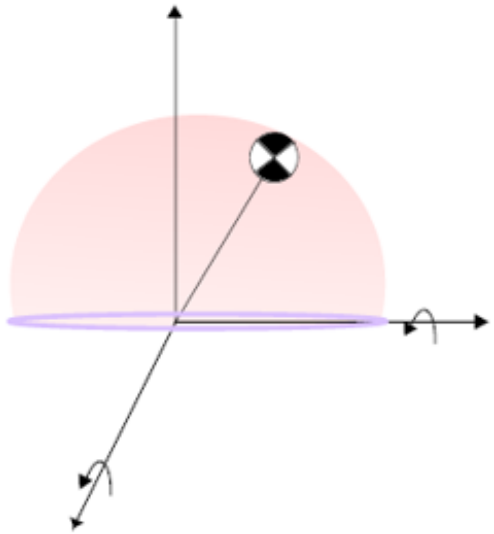


Figure 2.5: Spherical Inverted Pendulum Model.

based on SIP model which has a critically damp response, achieving the fast stabilization time with the least amount of energy consumption possible. The proposed control law is as follows:

$$\ddot{\theta} = (1 - k_p)\,\omega^2\theta - k_p\omega\dot{\theta} \tag{2.3}$$

Both LIPM and SIP use only the ankle joints for balancing and thus only consists of ankle based push recovery strategies when dealing with small disturbances and stepping strategy for larger disturbances.

2.2.3 Linear Inverted Pendulum with Flywheel

As the push force increases, we need to incorporate hip strategy in conjunction with ankle strategy to be able to maintain balance. J. Pratt et al. [4] proposed that angular momentum about the CoM must also be considered to capture

the behaviour of human beings rotating their arms rapidly or lunging forward in an attempt to balance themselves. They explicitly model the angular momentum by replacing the point mass in the LIPM with a Flywheel, thus calling it *Linear Inverted Pendulum Model with Flywheel (LIPM-FW)*. They also proposed the concept of Capture Points and Capture Regions where a Capture Point is a point on the ground where the robot can step to in order to bring itself to a complete stop and Capture Region is the collection of all Capture Points. For the Linear Inverted Pendulum Model, there is a unique Capture Point corresponding to each state and the ability to accelerate the CoM by changing the angular momentum can extend the unique Capture Point to a set of adjacent points, which is called Capture Region. If the Capture Region overlaps with the support polygon, the robot will be able to stabilize itself without moving, i.e., just using ankle and hip strategies. If they do not overlap, the robot will need to take a step to avoid falling. If the capture region is out of the kinematically reachable region, then the robot will not be able to avoid falling by taking just one step.

The dynamics of the model is described by the following equations:

$$\ddot{x} = \frac{g}{z_0}x - \frac{1}{mz_0}\tau_h \tag{2.4}$$

$$\ddot{\theta}_b = \frac{1}{J}\tau_h \tag{2.5}$$

where the z_0 is the constant height of the CoM, x is the displacement of the CoM in horizontal direction, θ_b is the angular displacement of the flywheel and τ_b is the angular moment of inertia generated by the flywheel.

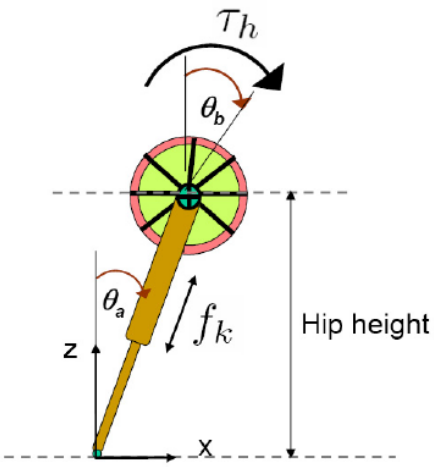


Figure 2.6: Linear Inverted Pendulum with Flywheel Model.

The LIPM-FW model uses both, ankle and hip based push recovery strategies for small disturbances and stepping strategy for larger disturbances. M. Shafiee-Ashtiani et al. [11] and A. Elhasairi et al. [12] used LIPM-FW model for push recovery and standing balance for position controlled robots but used different control approaches.

2.2.4 Double Inverted Pendulum

B. Stephens et al. [13] [14] [15] [16] proposed an integral controller for humanoid push recovery based on a *Double Inverted Pendulum Model*. They presented a Center of Pressure Regulator (CoPR) with a model tracking control that allows the robot to behave like a double inverted pendulum with the controller. Their proposed approach performed better than other standard controllers like the constrained LQR controllers.

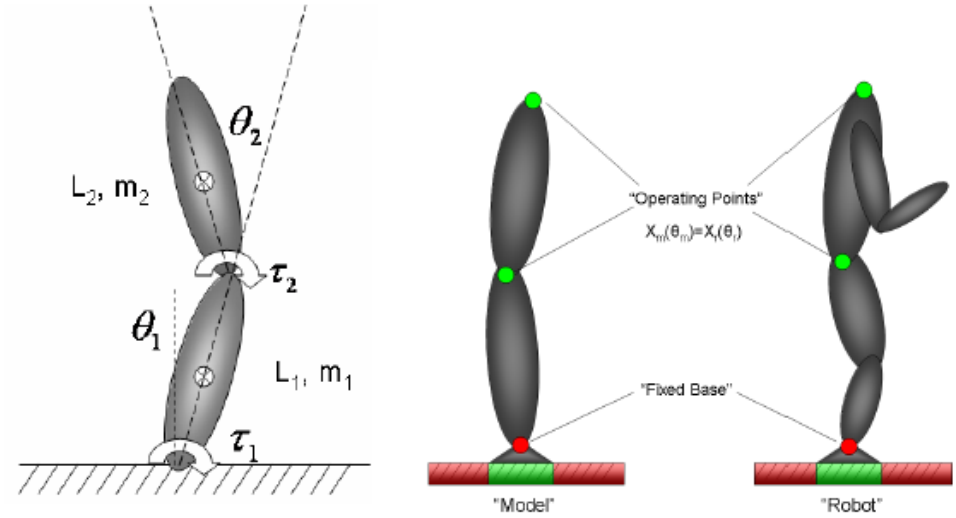


Figure 2.7: Double Inverted Pendulum Model.

2.3 Active versus Passive Dynamics

2.3.1 Active Dynamics Robots

Humanoid robots tends to be built as a research platform for general control and manipulation. These robots generally have a rigid actuators and follow a precisely calculated motion trajectory. With enough actuators in their arms, legs and body and a high acceleration potential, they can follow undisturbed trajectories with precision while remaining statically stable and interact with environment. Such humanoid robots have active dynamics and they are generally fully actuated and controllable. Simplified models discussed in Section 2.2 Even though, such active dynamics robots have a much more flexible range of motion, they also tend to have a delicate stability and must have a detailed maps of the environment to avoid collisions which cause large force spikes and potentially damage their rigid gearing. Large impulsive disturbances tend

to make these humanoid robots sensitive to external disturbances. Figure 2.8 shows some examples of active dynamics robots which also include RIT's TigerBot-VII.

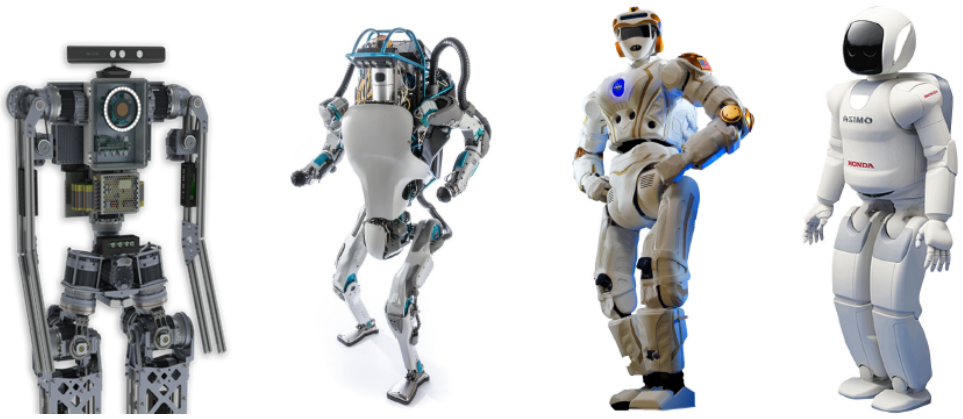


Figure 2.8: RIT's TigerBot-VII, Boston Dynamics' ATLAS, NASA's Valkyrie, and Honda's ASIMO.

One very success outlier amongst these robots is ATLAS. With the help of high powered hydraulic actuators, ATLAS from Boston Dynamics have managed to achieve a robust locomotion. Even though, no research regarding ATLAS have been made public by Boston Dynamics, the robot simulation model is made available online. ATLAS simulation model is a popular research platform amongst humanoid research community.

2.3.2 Passive Dynamics Robots

Static stability is limited to stable terrain and the robots need to rely on dynamic stability while moving or working in the real world. Passive dynamics refers to the dynamical behavior of the humanoid robot when not drawing energy from a supply. Many hoppers and biped robots are able to move

robustly with the combination of passive compliance and intuitive control laws by using techniques like placing the next step farther forward for higher speeds and injecting energy with each stride. Figure 2.9 shows some examples of active dynamics robots.

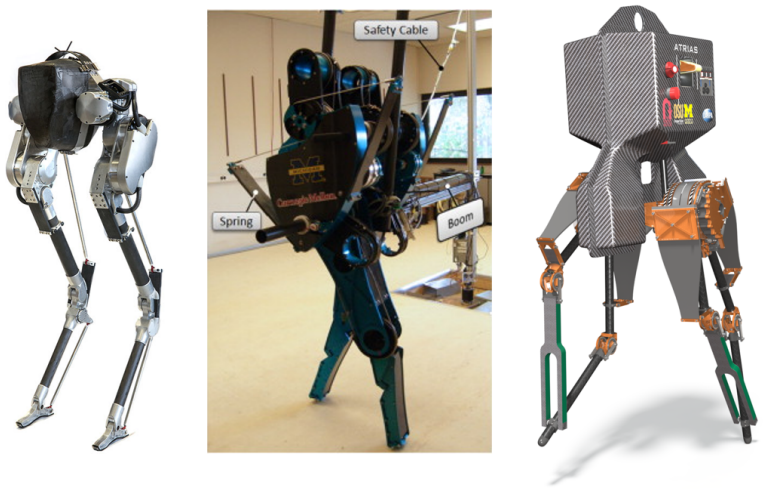


Figure 2.9: Oregon State University's Cassie, MABEL, and ATRIAS.

One very successful semi-active dynamics biped robots is Cassie [17]. It is designed with a low leg-to-torso inertia ratio and leg angle axes near the CoM for simplified dynamics. As such, it is amenable to intuitive control schemes and uses the spring-loaded inverted pendulum (SLIP) model as the simplified model [18] [19].

Spring Loaded Inverted Pendulum Model

The SLIP can be seen as a simple, lumped-parameter representation of animal limbs. The SLIP comprises a single point-mass body and a massless linear spring for each leg. Being composed of only a point mass and springs, the

SLIP is conservative and has no rotational inertia or angular momentum, and the only control signal for the unactuated SLIP is the leg angle at touchdown.

Figure 2.10 shows the SLIP model for running gaits [20]. A point mass m rebounds on a massless spring with stiffness k and rest length r_0 in stance while moving under the sole influence of gravity (g) in flight. Stance begins when the point mass satisfies a landing condition in flight. Stance ends when the spring has rebounded to its rest length.

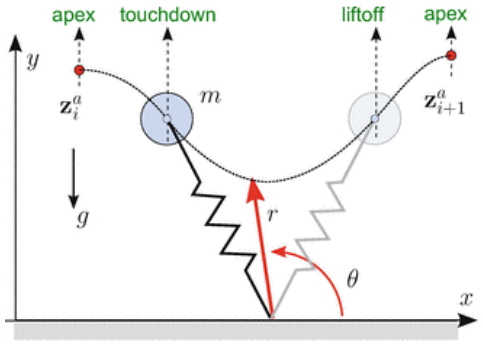


Figure 2.10: Spring Loaded Inverted Pendulum Model (SLIP).

The next Chapter introduces the Linearized Double Inverted Pendulum Model with all the assumptions and formulations for the model.

Chapter 3

Linearized Double Inverted Pendulum

3.1 Model Definition and Assumptions

The robot models have historically been inspired by human stance and generally represented as an inverted pendulum. The proposed LDIP is modeled as a double inverted pendulum with the following characteristics and assumptions as illustrated in Figure 3.1:

1. The mass of the legs is concentrated at mass m_1 and that of the torso is concentrated at mass m_2.

2. The distance between the hip joint and m_2 is fixed, and denoted by l_2. Similarly, the distance between the ankle and the hip is variable, denoted by l_1.

3. m_1 is fixed at the center of the length l_1, equidistant from ankle and

hip while maintaining a constant height of $z_0/2$. This implies that hip is always maintained at a constant height of z_0.

4. Only forward/backward and up/down motion is considered with negligible lateral motion.

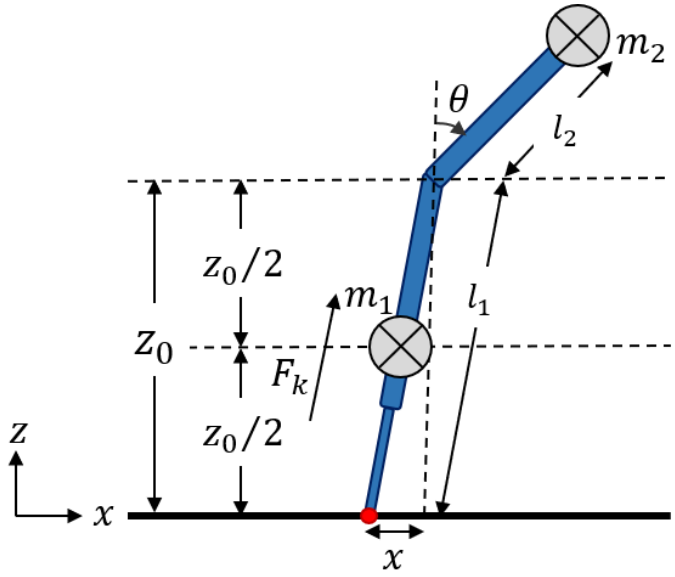


Figure 3.1: Linearized Double Inverted Pendulum Model.

3.2 Model Dynamics - Equations of Motion

The equations of motion have been derived using Lagrange's equations of the form:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \tag{3.1}$$

where L is the Lagrangian which is defined as the difference between the

kinetic energy, K, and the potential energy, P, of the system, i.e. $L = K - P$. The system state variables are $q = [x \ \theta]^T$, where x is the position of hip with respect to CoP in horizontal direction, θ is the angular position of torso with respect to the ground normal.

We start with the positions of mass m_1 in xz-plane and differentiate it with respect to time to get its velocity.

$$x_1 = \frac{x}{2} \qquad z_1 = \frac{z_0}{2}$$
$$\dot{x}_1 = \frac{\dot{x}}{2} \qquad \dot{z}_1 = 0$$

Similarly, for mass m_2,

$$x_2 = x + l_2 S\theta \qquad z_2 = z_0 + l_2 C\theta$$
$$\dot{x}_2 = \dot{x} + l_2\dot{\theta}C\theta \qquad \dot{z}_2 = -l_2\dot{\theta}S\theta$$

Next, we compute the square of total velocities for m_1 and m_2.

$$\begin{aligned} \dot{r}_1^2 &= \dot{x}_1^2 + \dot{z}_1^2 \\ &= \left(\frac{\dot{x}}{2}\right)^2 + (0)^2 \\ &= \frac{\dot{x}^2}{4} \end{aligned} \qquad \begin{aligned} \dot{r}_2^2 &= \dot{x}_2^2 + \dot{z}_2^2 \\ &= \left(\dot{x} + l_2\dot{\theta}C\theta\right)^2 + \left(-l_2\dot{\theta}S\theta\right)^2 \\ &= \dot{x}^2 + l_2^2\dot{\theta}^2 + 2\dot{x}\dot{\theta}l_2C\theta \end{aligned}$$

The total kinetic energy, K, and potential energy, P, is as follows,

$$\begin{aligned} K &= \frac{1}{2}m_1\dot{r}_1^2 + \frac{1}{2}m_2\dot{r}_2^2 \\ &= \frac{1}{2}m_1\left(\frac{\dot{x}^2}{4}\right) + \frac{1}{2}m_2\left(\dot{x}^2 + l_2^2\dot{\theta}^2 + 2\dot{x}\dot{\theta}l_2C\theta\right) \\ P &= m_1gz_1 + m_2gz_2 \\ &= m_1g\left(\frac{z_0}{2}\right) + m_2g\left(z_0 + l_2C\theta\right) \end{aligned}$$

The Lagrangian for the system is:

$$\begin{aligned} L &= K - P \\ &= \frac{1}{8}m_1\dot{x}^2 + \frac{1}{2}m_2\left(\dot{x}^2 + l_2^2\dot{\theta}^2 + 2\dot{x}\dot{\theta}l_2C\theta\right) - m_1g\left(\frac{z_0}{2}\right) - m_2g\left(z_0 + l_2C\theta\right) \end{aligned}$$

Using equation (3.1), we can compute the equations of motion for the system with $q_1 = x$ and $q_2 = \theta$.

$$\left(\frac{m_1}{4} + m_2\right)\ddot{x} + m_2l_2\ddot{\theta}C\theta - m_2l_2\dot{\theta}^2S\theta = F_x \tag{3.2}$$

$$m_2l_2\ddot{x}C\theta + m_2l_2^2\ddot{\theta} - m_2gl_2S\theta = \tau_\theta \tag{3.3}$$

where F_x is the force in x-direction and τ_θ is the torque applied by the hip joint. The force F_x will also be dependent on the force F_k as shown in Figure 3.1. Consider the free body diagrams for mass m_1 and m_2 shown in Figure 3.2.

For mass m_1 to maintain a constant height, the forces on m_1 in z-direction

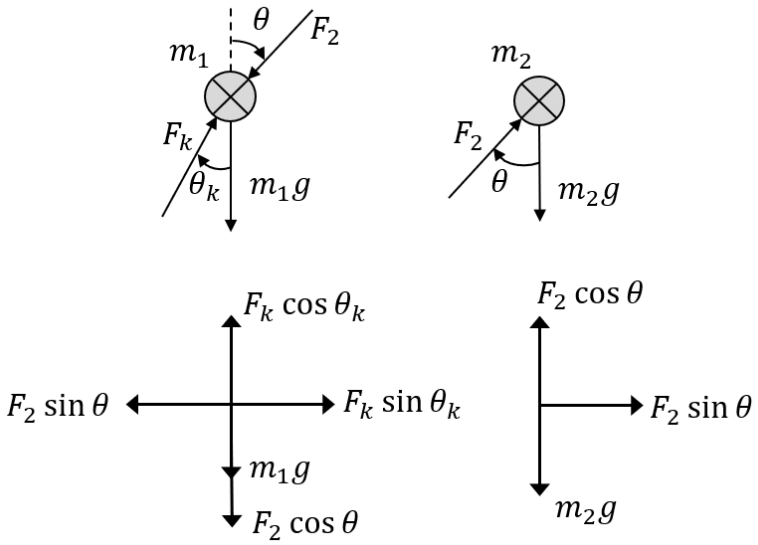


Figure 3.2: Free Body Diagram for mass m_1 and m_2.

must be zero. In Figure 3.2, free body diagram of m_2 shows that $F_2C\theta = m_2g$ and $F_2S\theta = -u_x$ where u_x is the control force in x-direction discussed later in this chapter.

$$\begin{aligned} F_kC\theta_k &= m_1g + F_2C\theta = m_1g + m_2g \\ F_k &= \frac{(m_1 + m_2)\, g}{C\theta_k} \end{aligned} \tag{3.4}$$

$$F_x = F_kS\theta_k + u_x \tag{3.5}$$

Putting equation (3.4) in equation (3.5) we get,

$$\begin{aligned} F_x &= \left[\frac{(m_1+m_2)\,g}{C\theta_k}\right] S\theta_k + u_x \\ &= (m_1+m_2)\,g\tan\theta_k + u_x \\ &= (m_1+m_2)\,g\frac{x}{z_0} + u_x \end{aligned}$$

$$F_x = \frac{(m_1+m_2)\,g}{z_0}x + u_x \tag{3.6}$$

Finally, by substituting F_x from equation (3.6) in equation (3.2) and taking τ_θ as control torque u_θ, we get the equations of motion for the systems.

$$\left(\frac{m_1}{4}+m_2\right)\ddot{x} + m_2 l_2 \ddot{\theta} C\theta - m_2 l_2 \dot{\theta}^2 S\theta = \frac{(m_1+m_2)\,g}{z_0}x + u_x \tag{3.7}$$

$$m_2 l_2 \ddot{x} C\theta + m_2 l_2^2 \ddot{\theta} - m_2 g l_2 S\theta = u_\theta \tag{3.8}$$

The following assumptions have been made in order to linearize them. θ and $\dot{\theta}$ are much smaller than 1, i.e. $\theta << 1$ and $\dot{\theta} << 1$. It should be noted that this assumption does not put any constraints on system variable x unlike in a Double Inverted Pendulum Model [13] where linearization is needed on all the system variables.

The linearized equations of motion are written in the state space representation as follows:

$$\dot{\overline{x}} = A\overline{x} + B\overline{u} \tag{3.9}$$

where $\overline{x}$ is the state variable vector, $\overline{u}$ is input variable vector, A is system

matrix and B is the control matrix. They have been defined as:

$$\overline{x} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \qquad \overline{u} = \begin{bmatrix} u_x \\ u_\theta \end{bmatrix} \tag{3.10}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{4g}{z_0}\left(1+\frac{m_2}{m_1}\right) & 0 & -4g\frac{m_2}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-4g}{l_2 z_0}\left(1+\frac{m_2}{m_1}\right) & 0 & \frac{g}{l_2}\left(1+\frac{4m_2}{m_1}\right) & 0 \end{bmatrix} \tag{3.11}$$

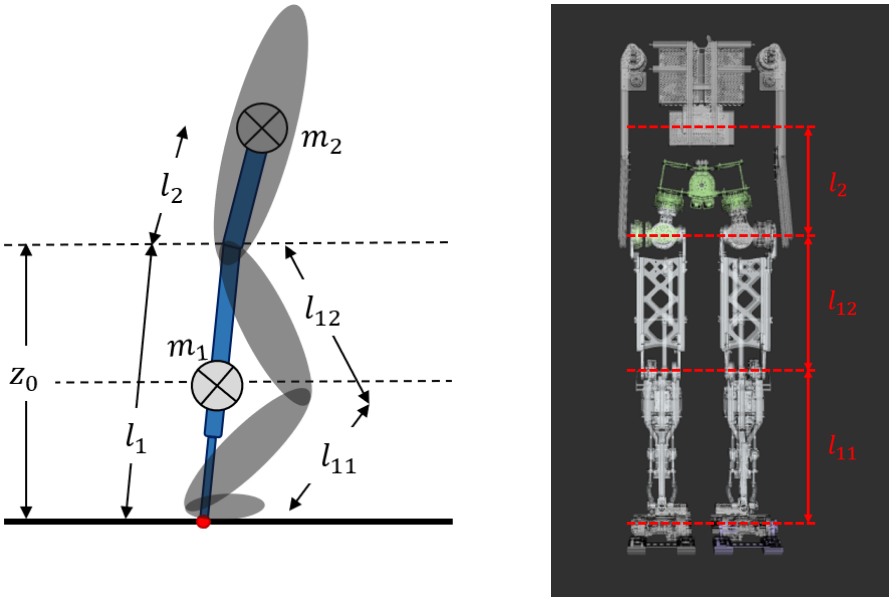
$$B = \begin{bmatrix} 0 & 0 \\ \frac{4}{m_1} & \frac{-4}{m_1 l_2} \\ 0 & 0 \\ \frac{-4}{m_1 l_2} & \frac{1}{l_2^2}\left(\frac{4}{m_1}+\frac{1}{m_2}\right) \end{bmatrix} \tag{3.12}$$

Adhering to the above mentioned constraints of $\theta << 1$ and $\dot{\theta} << 1$, the model can describe the dynamics of the robot assuming the required position is kinetically reachable given by,

$$|x| \leq \sqrt{(l_{11}+l_{12})^2 - z_0^2} \tag{3.13}$$

where l_{11} is the length of the tibia, l_{12} is the length of the femur, and the overlap between the robot parameters and the LDIP model has been illustrated in Figure 3.3.

The system variables defined for LDIP model will be used in balancing



(a) (b)

Figure 3.3: (a) Overlap between the robot and the LDIP Model (b) Corresponding lengths for Tigerbot-VII.

the robot utilizing modern control techniques. In the next Chapter, we will discuss about the LDIP balance control strategy implementation proposed in this work.

Chapter 4

LDIP Standing Balance Control

The Linear Quadratic Regulator (LQR) has been used in conjunction with Proportional Derivative (PD) controller for standing balance of the robot. The complete control framework, shown in Figure 4.1 for LDIP consists of an LQR block for push recovery and standing balance, along with a PD controller to maintain a constant height z_0 of the hip. These two controllers work together to generate the joint control torques for the humanoid robot.

Current states of the system, x, $\dot{x}$, θ, $\dot{\theta}$ and z_{hip} are determined using Forward Kinematics (FK). The system state vector $\left[x\ \dot{x}\ \theta\ \dot{\theta}\right]^T$ is fed to the LQR controller block which generates the required force inputs $[u_x\ u_\theta]^T$ to balance the robot using the feedback control law given by,

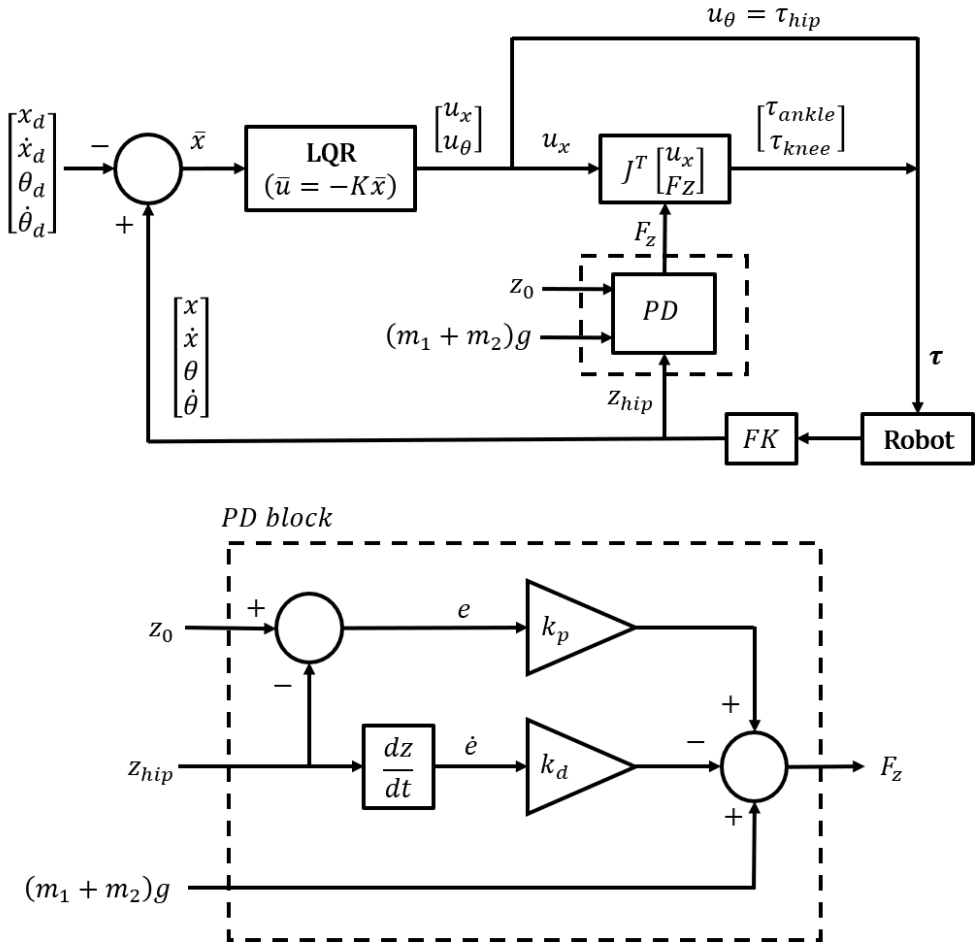


Figure 4.1: Control architecture for push recovery using LDIP model.

$$\begin{bmatrix} u_x \\ u_\theta \end{bmatrix} = -K_{LQR} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{4.1}$$

Additionally, the PD controller shown in Figure 4.1 computes the force required to compensate the deviation in height. The current height of the hip, z_{hip}, is compared with the fixed desired height, z_0. PD controller uses this error with its derivative to calculate the control force. This control force is then added to the constant force applied on the robot due to gravity, $(m_1 + m_2)\, g$, and computes the total force required in z-direction, F_z.

$$F_z = (m_1 + m_2)\, g + k_p\, (z_0 - z_{hip}) - k_d \dot{z_{hip}} \tag{4.2}$$

The required force inputs from LQR and PD controller are used to generate the joint torques for the robot. Jacobian of the robot is used to calculate the ankle and knee joint torques as shown in equation (4.3). Required input torque u_θ computed using LQR is used to generate torque at the hip joint.

$$\begin{bmatrix} \tau_{ankle} \\ \tau_{knee} \end{bmatrix} = J^T \begin{bmatrix} u_x \\ F_z \end{bmatrix} \tag{4.3}$$

$$\tau_{hip} = u_\theta$$

Next Chapter will give an overview of TigerBot-VII used for the validation of LDIP model.

Chapter 5

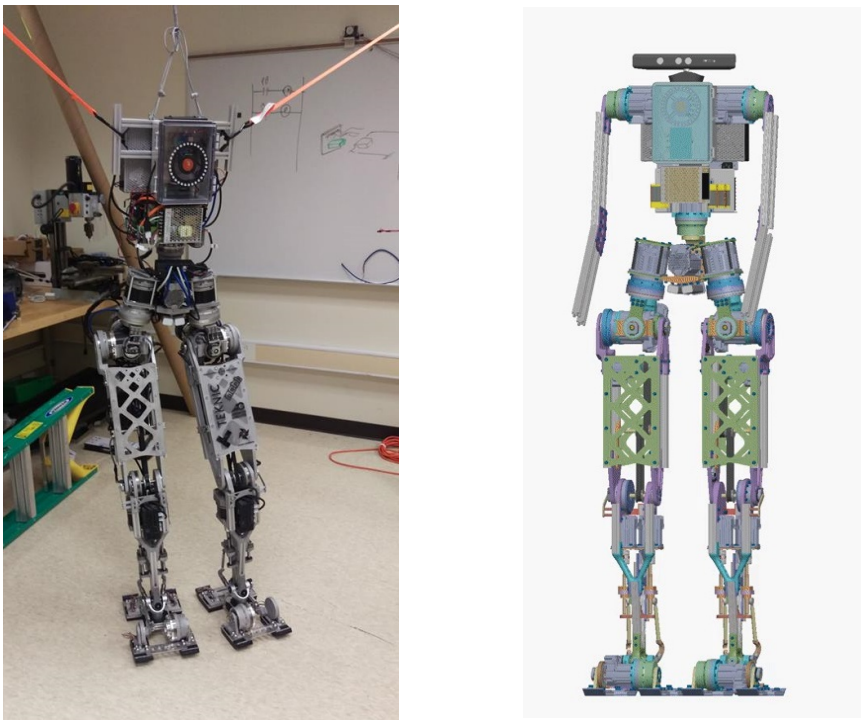
Overview of TigerBot-VII

5.1 Introduction

Tigerbot-VII is a humanoid robot designed and built at Rochester Institute of Technology, New York as a part of multi-disciplinary senior design project[1] [1]. It is a 14 DoF humanoid robot that is about the size of a human being. It is the 7th iteration of RIT's TigerBot humanoid series, hence called TigerBot-VII. The robot was originally intended to be a 22-DOF robot including arms and head rotation, however, the current version consists of each 7-DOF legs and a static torso. Figure 5.1 shows the CAD model and the real robot.

TigerBot-VII has an IMU sensor in the chest to detect the torso orientation and a Six Axis Force/Torque (SAFT) sensors in each foot to get the force/torque feedback from the feet. Each joint has an external absolute encoder to read the current joint states of the robot.

[1]Most of the information in this section is based on the work of student members who designed and built the robot.



(a) (b)

Figure 5.1: TigerBot-VII (a) Real Robot (b) CAD Model.

5.2 TigerBot-VII's Anatomy

TigerBot-VII is a human sized 14 DoF humanoid robot that has 7 independently controlled joints in each leg and is a fully active dynamics robot. Figure 5.2 shows the axis of rotation for each joint in the robot with respect to the universal coordinate frame. Each leg in TigerBot-VII features a three-axis hip joint, a knee joint, two-axis ankle joint and an active toe with a passive heel. Each of the joint except the ankle joints is driven by a 1:100 gear ratio harmonic drive to increase the torque output from each joint.

Figure 5.3 shows the hip and knee joints of TigerBot-VII. As shown in Fig-

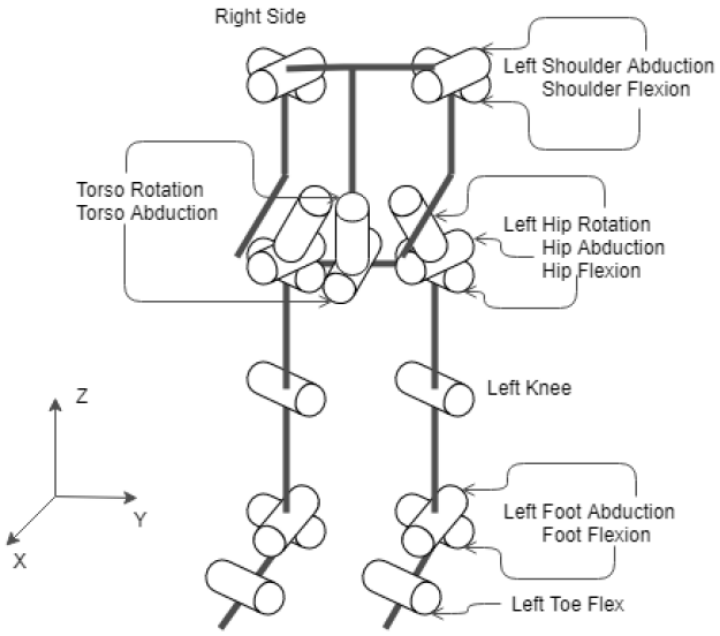
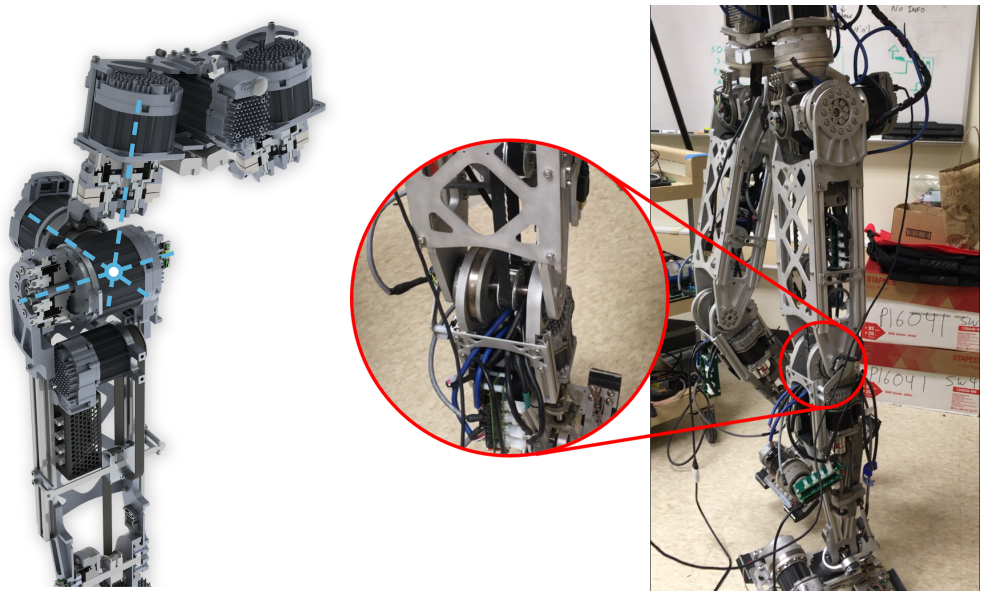


Figure 5.2: Joint axis of TigerBot-VII.

ure 5.3(a), the joint is designed such that the axis of rotation of each actuator within the hip joint intersects each other. This allows a 3 degree of freedom motion for the hip joint with a roll, pitch and yaw motion. Collectively, the hip joint resembles a ball and socket joint just like the hip and shoulder joints in humans. Figure 5.3(b) shows the one axis knee joint which is driven by a belt and pulley mechanism, which in turn is driven by an actuator attached close to hip joint in the upper femur. The placement of the actuator makes the joint compact and keeps the weight of the heavy motors close to the torso.

The two axis ankle joint is controlled through two linear actuators on the sides of the tibia as shown in Figure 5.4. The linear actuators drive the fibula up/down to move the ankle, shown with yellow dashed arrow lines. Furthermore, it also depicts the range of motion of the ankle joint if only one



(a) (b)

Figure 5.3: (a) Three axis hip joint with intersecting axis of rotation (b) Single axis knee joint.

actuator is moved. Ankle's pitch is controlled by driving the actuators at the same speed in the same direction, while the roll is controlled by driving the actuators at the same speed in opposite direction. Thus, by controlling the speed and direction of the two linear actuators appropriately, we can control the roll and pitch of the ankle joint simultaneously. The measure of velocity to travel one degree per second for each axis is used to calculate the actuator velocities to move the desired change in angle with the desired time for each axis independently. Adding the desired actuator velocities for the two axis give the required velocities to move the joint in both axes simultaneously. Note that the ankle joint actuators do not have the harmonic drive.

The feet of the robot consists of two separate sections, one is the active

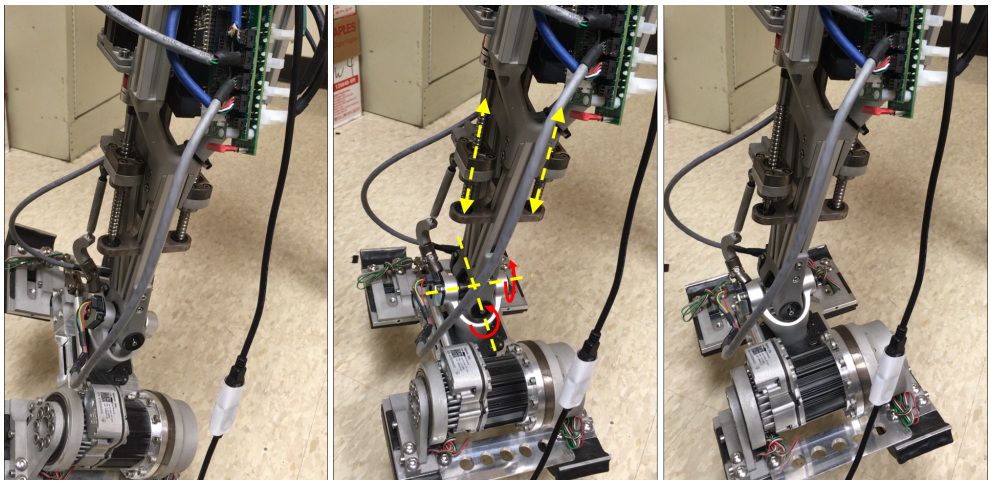


Figure 5.4: Two axis ankle joint.

Table 5.1: Physical Robot Parameters for Tigerbot-VII.

Parameter	Value	Parameter	Value
Height (m)	1.689	Mass (kg)	51.51071
Leg length (m)	0.9798	Foot Length (m)	0.33921
Femur Length (m)	0.44288	Tibia Length (m)	0.43513

powered toe which is driven by a motor through a harmonic drive and the other is a passive heel which helps to absorb the shocks while walking by passively damping them. The feet of the robot also houses a custom Six Axis Force/Torque (SAFT) sensor which can be used to get force/torque feedback for standing balance, push recovery and walking. The soles of the feet have been lined with a foam cushioning to absorb shocks while stepping and load cells are attached between the foam cushion and the feet on all four segments of the feet that touches the ground. Table 5.1 summarizes the real robot's parameters.

5.3 Hardware Specifications

The robot uses Teknic Clearpath SDSK series motors for all joints. These motors are powerful actuators with multiple operation modes. Tecknic Clearpath SDSK [21] is a Step and Direction series of motors which are essentially industry grade stepper motors with added functionality. These motors have an enable input pin, direction input pin and a step input pin, all these pin are digital input pins. The direction pin is used to set the direction of the motion and the step pin is fed a train of pulses to control the speed of the motion. Each pulse rotates the motor by a certain degree based on a customizable resolution set for the motor. These motors also have a Regressive Auto Spline (RAS) feature which is a vibration and resonance suppression features. It is a jerk limiting, and jerk-derivative limiting feature. It uses advanced algorithms to analyze each commanded move and rapidly calculate and "fit" a forth-order polynomial spline to it. This converts the sharp transitions between constant velocity and acceleration with more gradual, rounded corners.

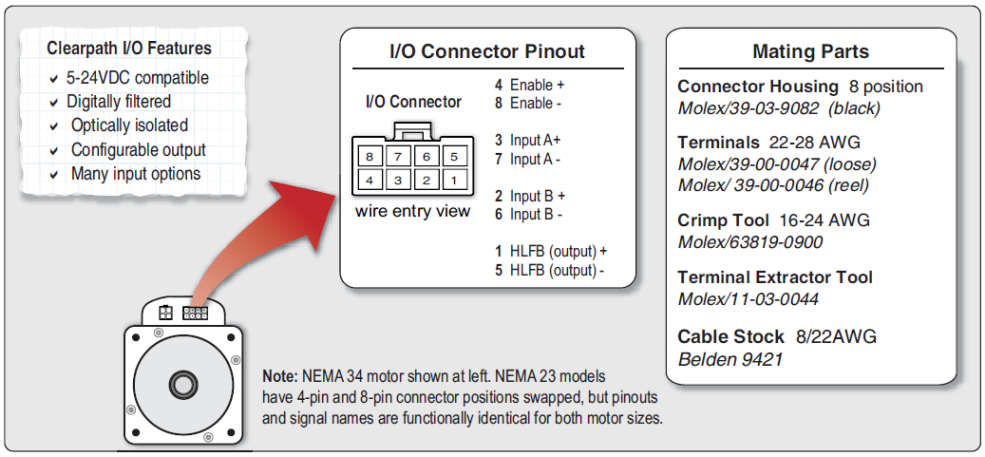


Figure 5.5: I/O Connector pinout for Teknic Clearpath motors.

These motors operate on 75V DC power supply. Due to the surge current drawn by these high power motors, it is essential that the Teknic power supply is used to power them up which is especially designed to suppress these surges. A Teknic power supply is mounted as an integral part of the torso of the robot with a built in emergency stop. All the power lines to each motor and the robot runs through the Teknic power supply.

Since the Clearpath motors does not give access to their internal encoders, external encoders have been used for each joint in TigerBot-VII. AMT203 modular absolute encoder from CUI Inc. is used for every joint except the hip rotation joint where IncOders from Zettlex is used. AMT203 modular absolute encoder is a 12 bit resolution encoder which communicates over SPI. This encoder could not be used in the hip rotation joint as the encoder placement was not possible due to design and space constraints around that joint. IncOders from Zettlex are inductive absolute encoders that consist of two rings that must be mounted parallel to each other but not touching each other. It measures the angle difference between the two rings through inductance. Although, this encoder also uses SPI for communication, it does not follow some aspects of SPI in the sense that it does not have a slave select line (SS). Communicating with a mix of AMT203 modular absolute encoder and IncOders on the same SPI channel is thus handled by using a multiplexer to add a virtual slave select line in IncOders.

A custom three-layer PCB stack is used as an interface to control the joints where each PCB stack can handle up to three joints. The main processing unit in each PCB stack is a Teensy 3.2 that handles all the communication with the motors and encoders. The first two layers house the Teensy 3.2 and

have connectors for communication with motors and encoders while having an option of adding a multiplexer by using a jumper between two pins. The third layer takes in the 75V DC power line and distributes the power between the 3 motors connected to the PCB stack through a current measuring circuit. There are 3 current measuring circuits on the third layer of the PCB stack for each motor. This could be used to measure the current drawn by each motor, hence indirectly measuring the torque applied by each motor. Note that the encoder and the other electronics on the PCB stack is powered separately and are not connected to the 75V DC power line. There are also 12V rails for each PCB stack, which gets converted locally to 5V for sensors, and 5V rails for the ODroid XU4 and the USB hubs. The Teensys can be powered from the PCB stack or from the USB hub.

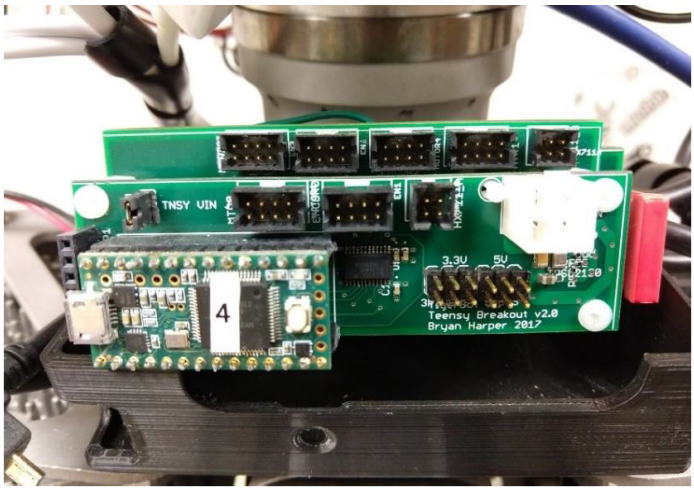


Figure 5.6: Three layered PCB stack to control 3 joints per PCB stack.

All the Teensy 3.2 in each PCB stack are connected to an Odroid XU4 which acts as the master CPU for the robot. An IMU from Variense is also

connected to the Odroid XU4 which is placed at the center of the torso. It has a 3-axis gyroscope, a 3-axis accelerometers, and a magnetometer. It is a plug-and-play sensor which can be directly connected to a single board computer through a USB. It outputs readings from the accelerometer, gyroscopes, and magnetometer, and can be configured to output data as Quaternions, Euler angles, and the heading. The sensor has ROS libraries available online and can be interfaced directly as a separate ROS-node.

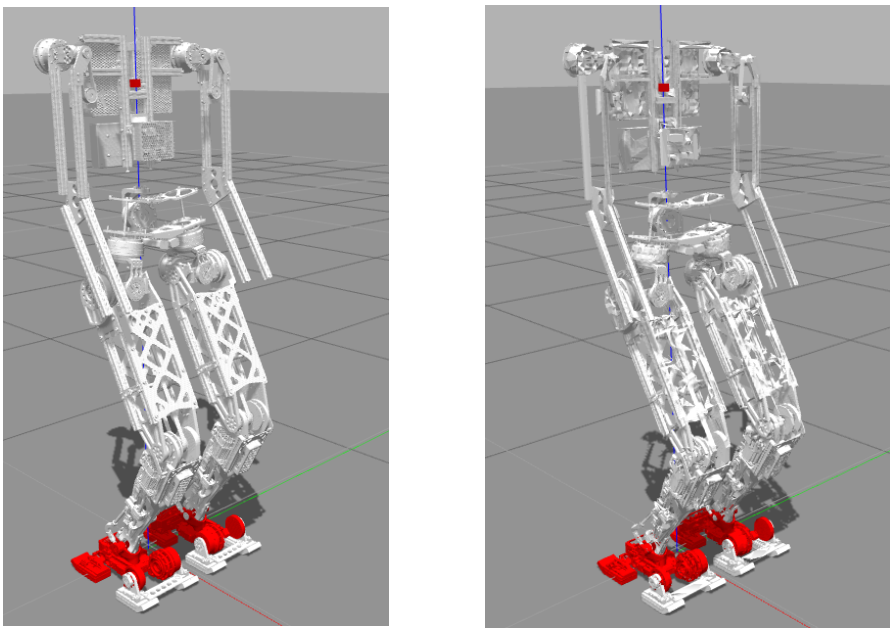
The robot's feet have 4 load cells that measure how the robot's weight is distributed and if the feet are in contact with the ground. TigerBot-VII also has a custom Six-Axis Force/Torque (SAFT) sensor module in each foot for precise force and torque feedback. This sensor also has a Teensy 3.2 inside it that can be directly connected to the Odroid XU4 and made into a ROS-node through rosserial for effective communication.

5.4 Gazebo Simulation

Gazebo simulation, which is a physics engine has been setup to validate the LDIP model to closely represent the real robot's complex dynamics based on the information given in the URDF of the robot model. The initial framework for TigerBot's model in Gazebo was setup in 2018 [1] and that framework has been modified to work more effectively.

The simulation model previously had an issue where it was using very detailed meshes for each link and rendering those detailed meshes took a lot of computation power. The simulation time lagged much behind as compared to the real time and the system slows down considerably. To overcome this

problem, the number of vertices in the mesh were reduced by running a cluster decimation on each mesh in MeshLab[2]. Figure 5.7 shows the comparison of TigerBot-VII model before and after the simplification of the meshes.



(a) (b)

Figure 5.7: TigerBot-VII Gazebo model with (a) High mesh vertices (b) Low mesh vertices.

Furthermore, sensors were added to the Tigerbot-VII Gazebo model. IMU sensors were added at the torso and feet of the robot, marked with red in Figure 5.7. This helped in finding the orientation of the torso and the feet at any given time. Force/Torque sensors and contact sensors were added at the feet to mimic the SAFT sensor and contact sensors in the feet of the robot as discussed in section 5.2. A separate Gazebo plugin was written to create a

[2]MeshLab is an open source system for processing and editing 3D triangular meshes. http://www.meshlab.net/

transport node and push information from gazebo to ROS topics.

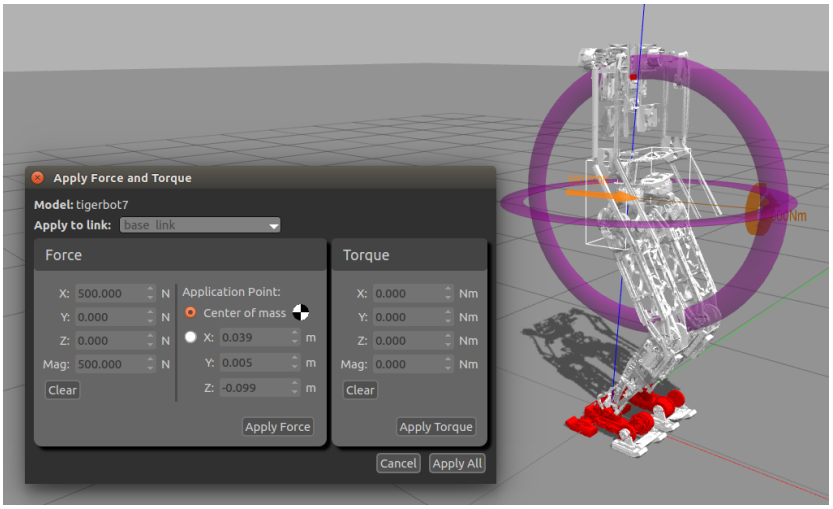
To validate the LDIP model, we needed a way to emulate external disturbances in Gazebo. Two methods were used for testing and validation of the model.

1. The first method was using Gazebo's in-built tools and utilities. Specifically, the *apply_body_wrench* service in Gazebo that applies a wrench (force/torque) on a link of the robot. The Gazebo interface is shown in Figure 5.8(a).

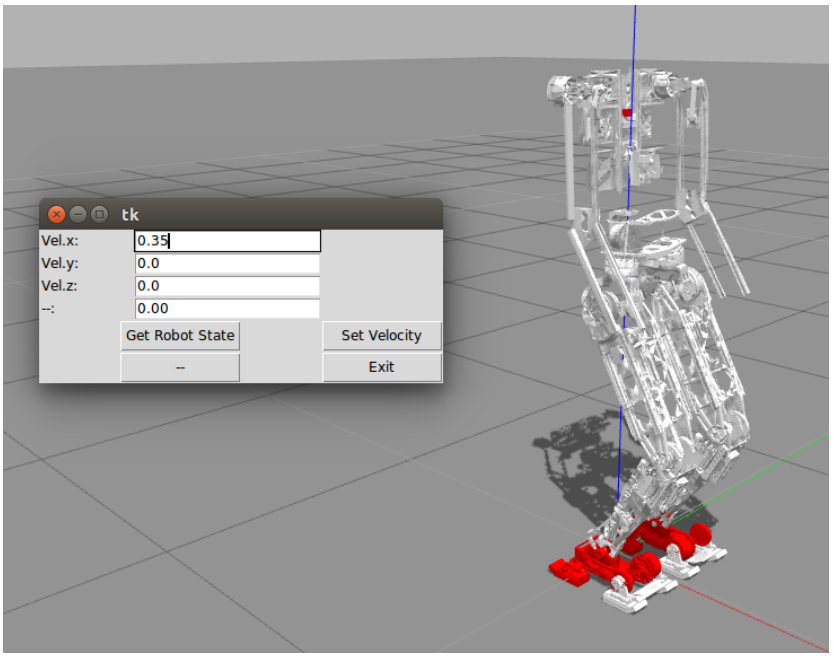
2. The second method was creating a GUI using python tkinter library to give an initial velocity to the pelvis. This emulates an impulsive external disturbance that would change the velocity from the state of rest to the test case velocity. The python tkinter based interface uses *GetModelState* and *SetModelState* Gazebo services to get the current state of the robot model and set the velocities to the test case velocity for the model. The interface is shown in Figure 5.8(b).

The robot was shifted from position controlled implementation to effort controlled implementation in Gazebo for validation of this work.

The model performance and its ability to balance the robot in the presence of disturbance has been simulated and tested on MATLAB. The model functionalities and capabilities were validated using Gazebo physics simulation with TigerBot-VII robot model. The calculated joint torque signals have been used to balance the robot in Gazebo while emulating external disturbances. In the next Chapter, the simulation setup has been described and the recorded results have been discussed.



(a)



(b)

Figure 5.8: Emulating external disturbance by (a) applying force at the pelvis (b) giving an initial velocity to the pelvis.

Chapter 6

Simulation Results

To verify the proposed model, simulation experiments were conducted in MATLAB and LQR has been used as the controller for the model. Further validation was done on Gazebo using TigerBot-VII simulation model.

6.1 Torque constraints for balance criteria

It is assumed there is no slippage between the foot of the robot and the ground. The ankle torque must be constrained to ensure that the CoP does not leave the support polygon of the robot. CoP can be calculated using equation (6.1) [7].

$$x_{cop} = -\frac{\tau}{f_z} \tag{6.1}$$

where f_z is the normal force equivalent to the force due to gravity, $f_z = (m_1 + m_2)g$. When CoP is at the edge of the support polygon, the maximum

constrained torque that can be generated at the ankle is given by,

$$\tau_{max} = -x_{cop\_max}(m_1 + m_2)g \tag{6.2}$$

The input state variable u_x, shown in equation (4.1) is restricted to approximate the required constraints on ankle torque, τ. The maximum constrained force u_{x_max} at the hip in x direction is computed by approximating its relation with the ankle joint torque τ_{max} given $x_{max} << z_0$, shown in Figure 6.1,

$$u_{x\_max} \approx \frac{\tau_{max}\cos\theta_k}{\sqrt{x^2+z_0^2}} = \frac{\tau_{max}z_0}{x^2+z_0^2} \approx \frac{\tau_{max}z_0}{z_0^2} = \frac{\tau_{max}}{z_0} \tag{6.3}$$

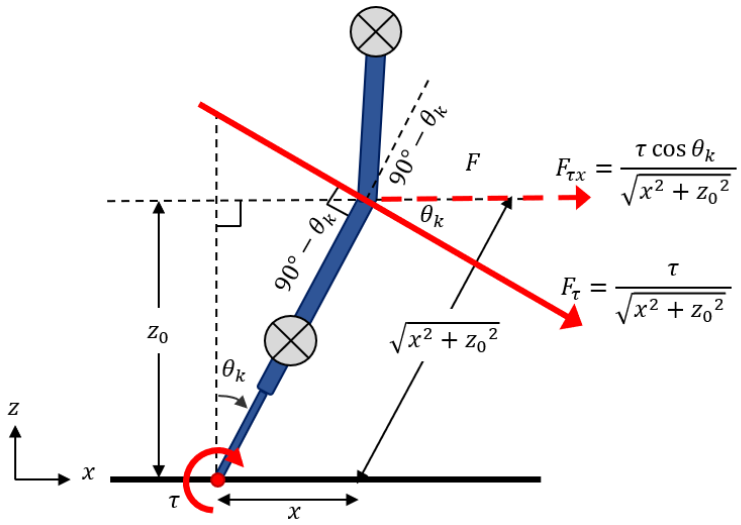


Figure 6.1: Relation between ankle torque and u_x.

Table 6.1: Robot Parameters for Tigerbot-VII used for Standing Balance.

Parameter	Value
m_1	8.781401 kg
m_2	15.510074 kg
l_{11}	0.431687 m
l_{12}	0.439994 m
l_2	0.383356 m
g	9.81 m/s^2
z_0	0.75 m

6.2 Experimental Setup and Results

As discussed in Chapter 5, Tigerbot-VII robot model is used for the simulation experiments. It is a 14 Degree of freedom humanoid robot IMU sensor on the torso and six-axis force/torque sensors on the ankle joints. For simulation, Tigerbot's URDF parameters are used as summarized in Table 6.1. The LQR weight matrices Q and R are manually tuned to get the desired response given by,

$$Q = diag\left[1e^2, 1e^3, 0.1, 0.1\right] \quad R = diag\left[200, 100\right]$$

The computed state-feedback matrix K_{LQR} is

$$K_{LQR} = \begin{bmatrix} 309.718 & 74.309 & 41.231 & 22.626 \\ -449.198 & -74.492 & 56.857 & -14.693 \end{bmatrix}$$

The balancing capabilities of the proposed model is inspected by giving an initial velocity at the pelvis of the robot. This simulates an external impulsive disturbance that changes the velocity from state of rest to the given test case velocity. Figure 6.2 shows the change in system states and the torque generated

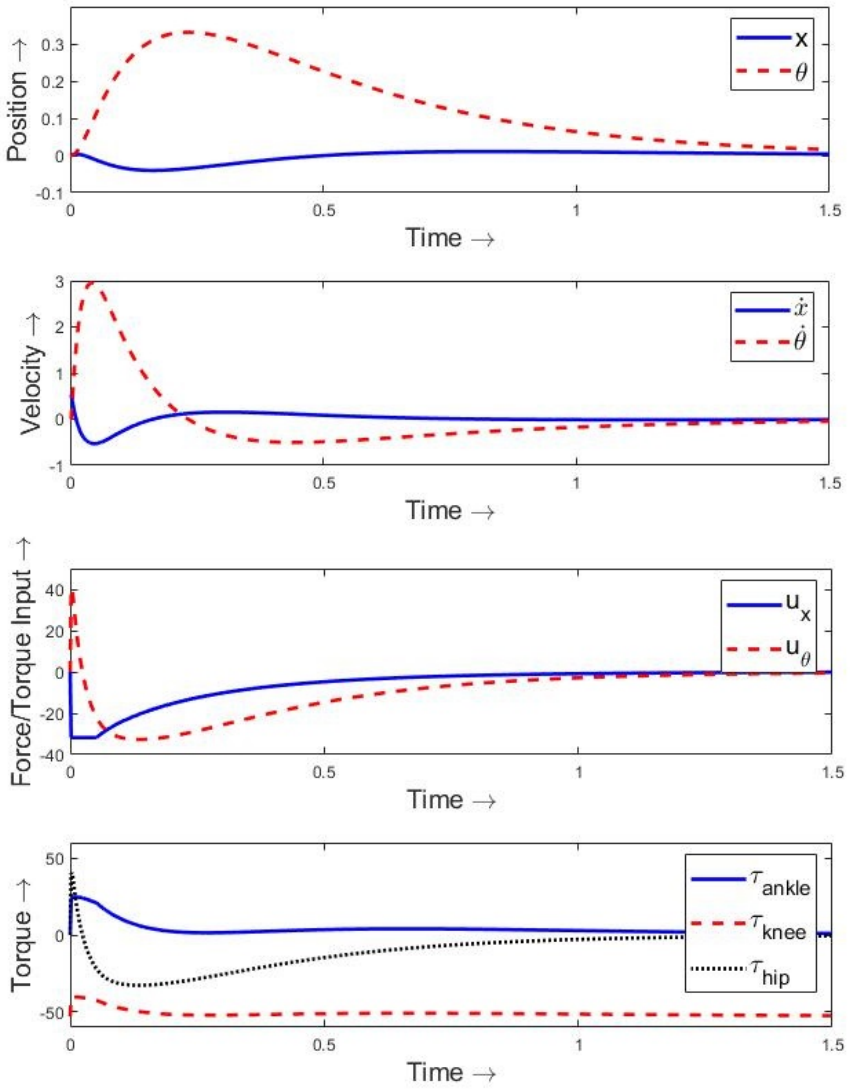
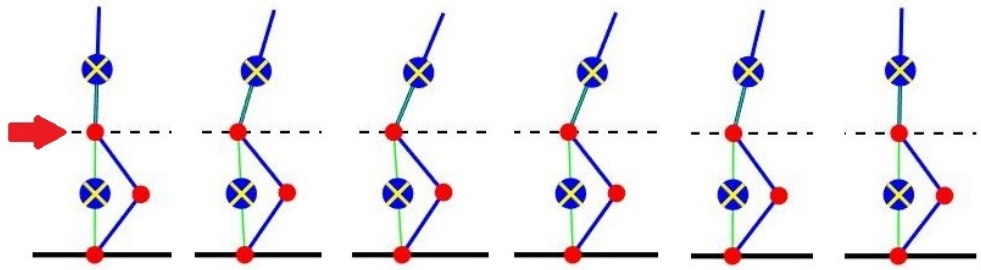
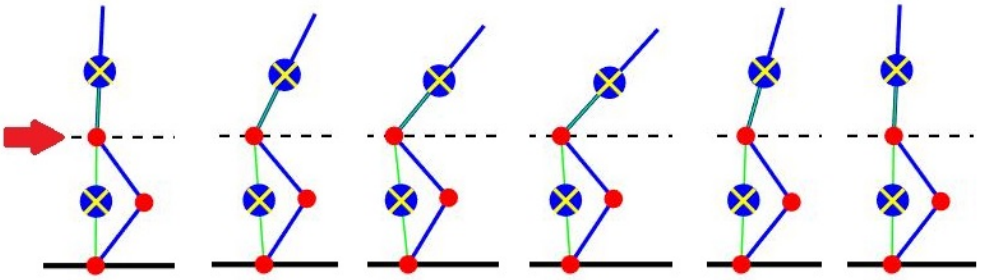


Figure 6.2: Response to an impulsive disturbance that would change the velocity from 0.54 m/s.



(a)



(b)

Figure 6.3: Animation of the LDIP Model response to an impulsive disturbance that changes the velocity from rest to (a) 0.54 m/s and (b) 0.9 m/s using robot parameters mentioned in Table 6.1. Video Link : https://youtu.be/89pCQ5W1myM

as a function of time for an initial velocity of 0.54 m/s. The input force u_x is limited to ± 31.7732N to restrict $x_{cop} \in (-0.1, 0.1)$ which in-turn constraints the ankle torque τ_{ankle} to ± 23.8299N-m. The hip torque τ_{hip} is also limited to ± 90N-m for these experiments. Figure 6.3(a) shows the animation of the LDIP model in response to the external impulsive disturbance corresponding to Figure 6.2.

From Figure 6.2 and 6.3, it can be observed that by lunging the torso in the direction of disturbance, the robot can generate enough moment of inertia about the pelvis to effectively recover from the external disturbance. The restrictions imposed on u_x successfully constraints the maximum torque on ankle τ_{ankle} between ± 23.8299N-m which ensures that the CoP is inside the support polygon. As the magnitude of external disturbance is increased, the swing of the torso increases to generate more angular moment of inertia while having minimal changes in the system state variable x, position of pelvis.

When compared with other similar models like LIPM and LIPM-FW, it can be observed that LDIP model has a lower force requirement to recover from an external disturbance, given constraints over the forces and torques that can be generated by the robot. Furthermore, the angular moment of inertia generated by lunging the torso forward/backward is an implicit part of the model and the torso position is a system state variable. Due to this, applying modern control techniques on this model does not require explicit control over the torso.

Figure 6.4 shows the trajectories and applied force/torque to balance the humanoid robot using LQR from a disturbance that can change the velocity of pelvis from state of rest to 0.34m/s. The LIPM-FW model behaviour

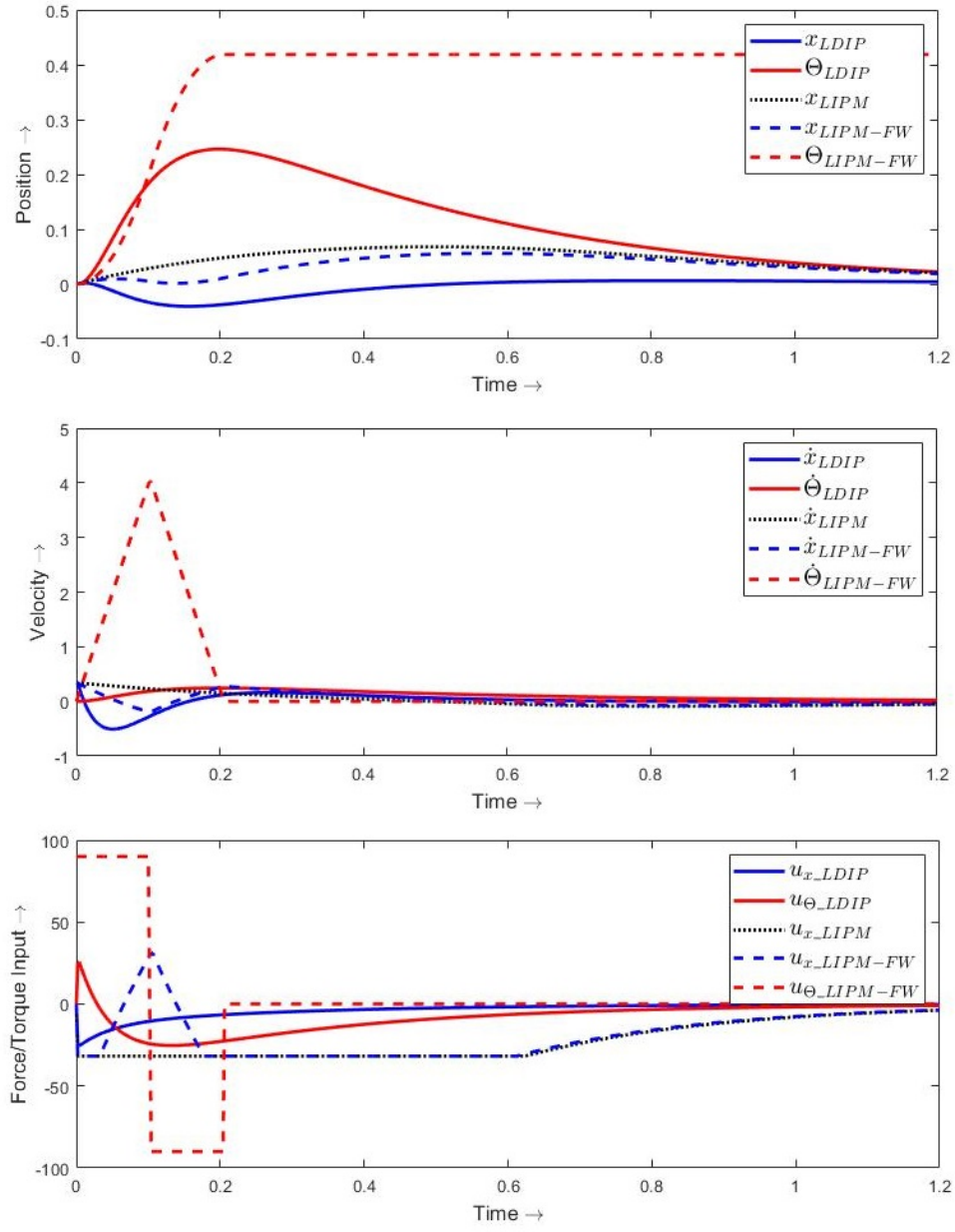


Figure 6.4: Comparison between LIPM, LIPM-FW and LDIP to an impulsive disturbances that changes velocity of pelvis from rest to 0.34m/s.

Table 6.2: Performance comparison between different models when subjected to different external impulsive disturbances.

Model	Maximum initial velocity before falling
LIPM	0.35 m/s
LIPM-FW	0.41 m/s
LDIP*	0.8 m/s

simulation for comparison is similar to the implementation in [4]. For LIPM, the input forces and torques needed to balance the robot are large as it only uses ankle strategy to recover from the disturbances. The LIPM-FW shows a similar performance when small disturbances are applied, however it is able to balance against larger disturbances as compared to LIPM. This is because of the added effects of the angular moment generated about the CoM, which aids in increasing the capture region. In LDIP model, the torque signals to generate the angular moment of inertia about the pelvis of the robot is directly calculated by the LQR controller. This enables us to optimally move the torso of the robot and create effects similar to a human lunging forward/backward or rotating the arms for balance Table 6.2 shows a performance comparison between different models by showing the maximum initial velocity at the pelvis from which each model can recover before falling or deviating too much from the constraints.

To validate the proposed model, LDIP has been implemented for push recovery scheme using Tigerbot-VII on Gazebo physics simulation environment, shown in Figure 6.5. PD controller parameters used for this implementation are $k_p = 2500$ and $k_d = 100$. A small constant bias force was also added to u_x in the Gazebo implementation. This was done in order to offset the effects

of $x = 0$ point being the center of ankle which is not always the Zero Moment Point or ZMP of the robot. State variables and motion torque trajectories are shown in Figure 6.6.

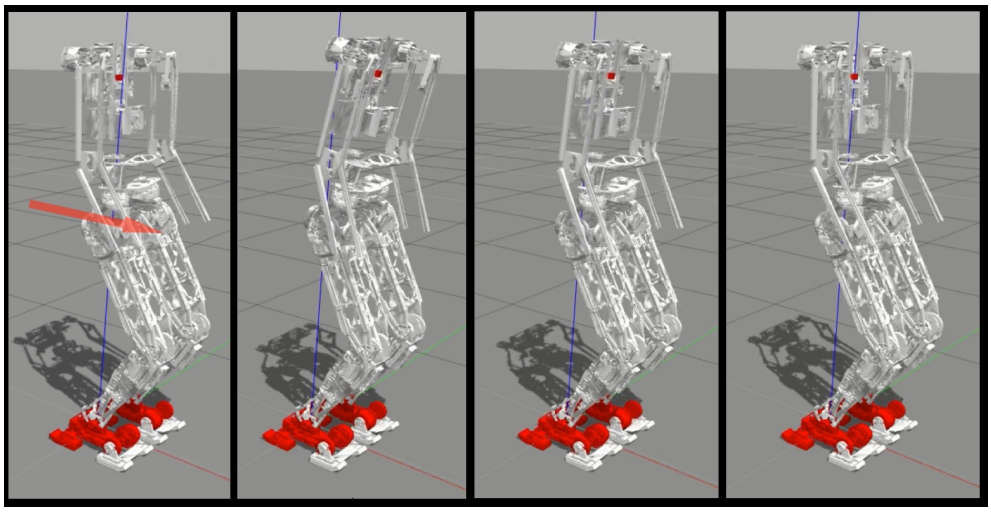


Figure 6.5: Gazebo simulation response with constrained LQR on LDIP model to an impulsive disturbances that changes velocity of pelvis from rest to 0.35m/s. Video Link : https://youtu.be/89pCQ5W1myM

If the external disturbance is applied at an angle, velocity component in x direction is handled by the LQR controller and the velocity component in z direction is handled by the PD controller as it is responsible to maintain a constant height z_0 of the hip and compensate for any deviations in the height.

Figure 6.7 and 6.8 shows the gazebo simulation and the system states and motion torque trajectories respectively in response to an external disturbance that changes the velocity of the pelvis from rest to 0.4243 m/s diagonally upwards, i.e., $v = (0.3\hat{i} + 0\hat{j} + 0.3\hat{k})m/s$. It was observed that if the velocity in positive z direction is significantly large, the robot's feet will no longer stay on the ground and will become unstable.

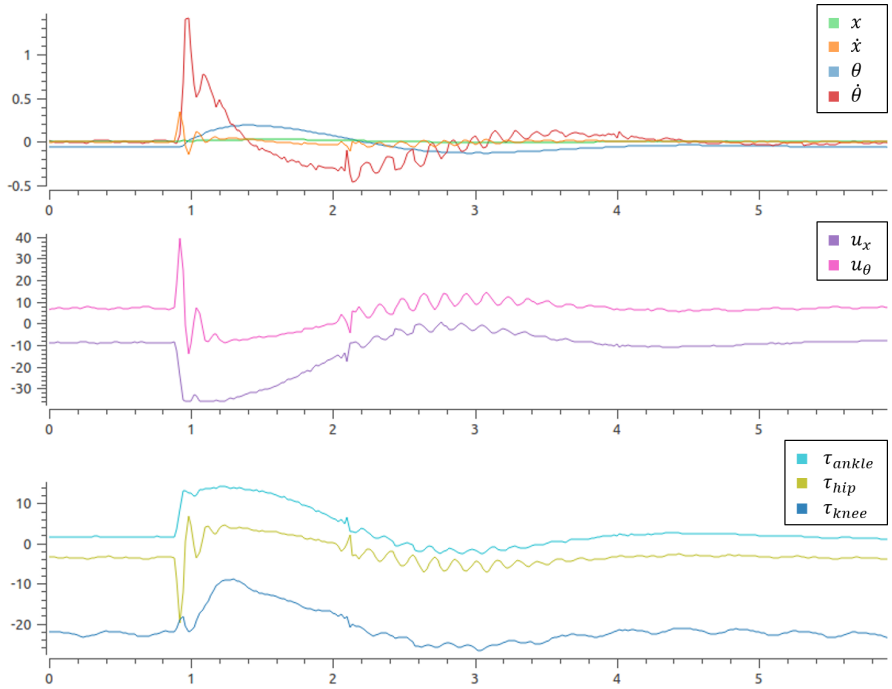


Figure 6.6: State variables and motion torque trajectories in response to an initial velocity of 0.35 m/s, $v = (0.35\hat{i} + 0\hat{j} + 0\hat{k})m/s$

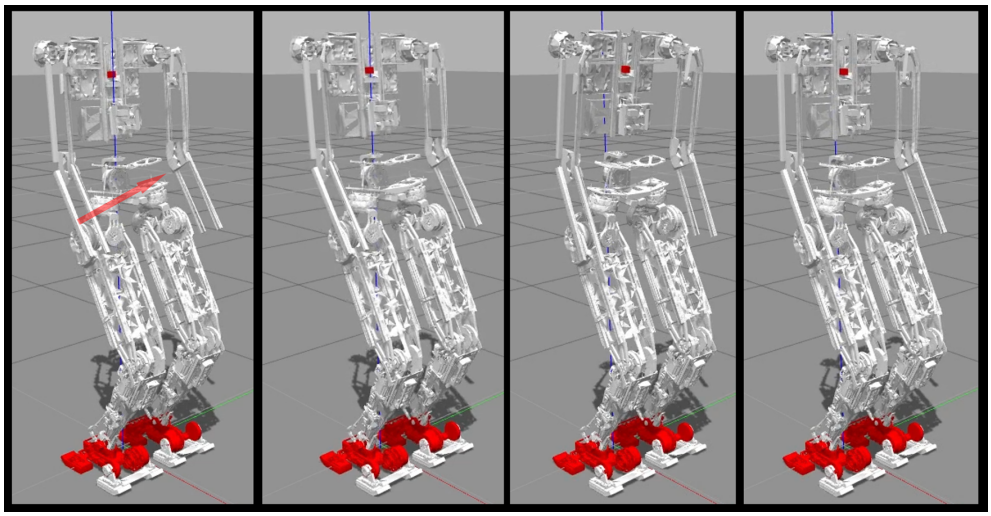


Figure 6.7: Gazebo simulation response with constrained LQR on LDIP model to an impulsive disturbances that changes velocity of pelvis from rest to 0.4243 m/s in the diagonally upward direction, $v = (0.3\hat{i} + 0\hat{j} + 0.3\hat{k})m/s$.

Figure 6.9 and 6.10 shows the gazebo simulation and the system states and motion torque trajectories respectively in response to an external disturbance that changes the velocity of the pelvis from rest to 0.4243 m/s diagonally downwards, i.e., $v = (0.3\hat{i} + 0\hat{j} - 0.3\hat{k})m/s$. It was observed that the PD controller is able to handle larger velocities in the negative z direction as compared to positive z direction. If the controller fails to balance the robot, the robot will need to switch to stepping strategy and take a step to avoid falling.

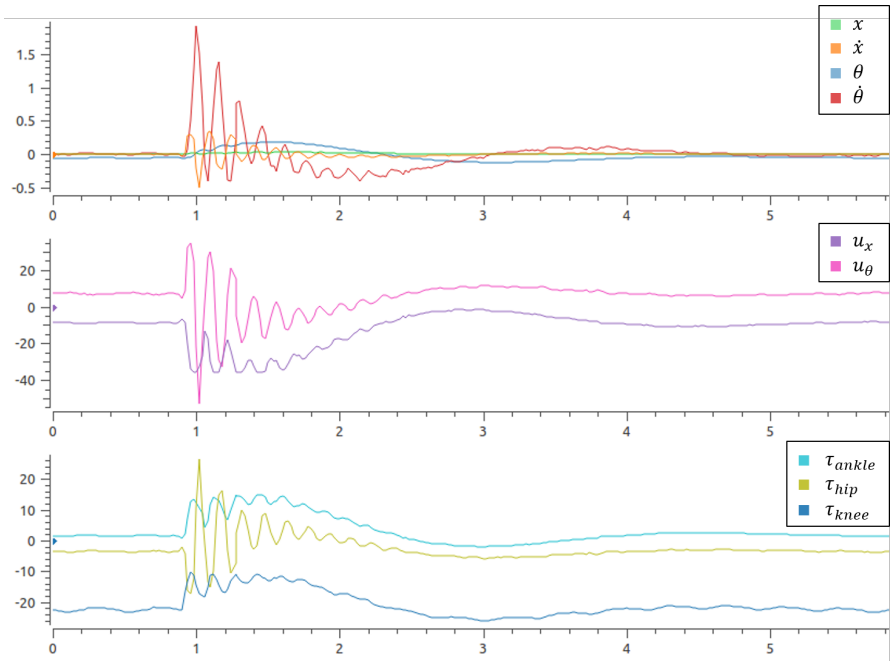


Figure 6.8: State variables and motion torque trajectories in response to an initial velocity of 0.4243 m/s in the diagonally upward direction, $v = (0.3\hat{i} + 0\hat{j} + 0.3\hat{k})m/s$

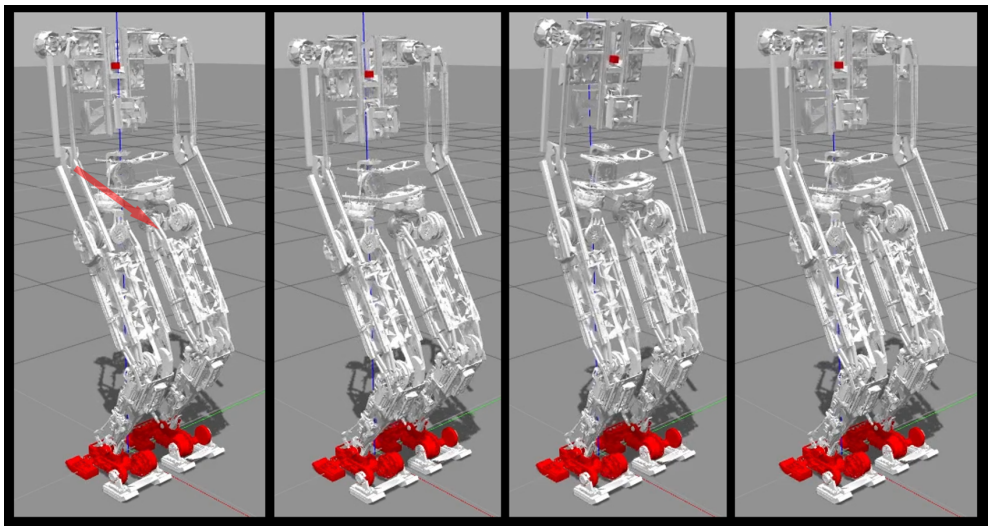


Figure 6.9: Gazebo simulation response with constrained LQR on LDIP model to an impulsive disturbances that changes velocity of pelvis from rest to 0.4243 m/s in the diagonally downward direction, $v = (0.3\hat{i} + 0\hat{j} - 0.3\hat{k})m/s$.

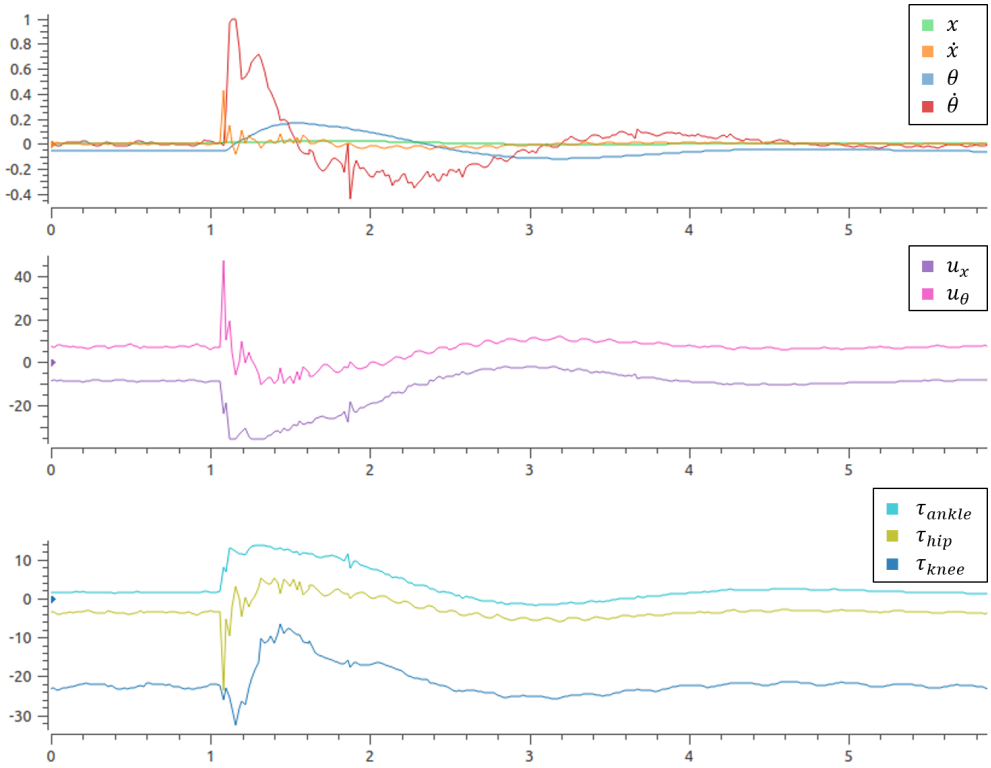


Figure 6.10: State variables and motion torque trajectories in response to an initial velocity of 0.4243 m/s in the diagonally downwards direction, $v = (0.3\hat{i} + 0\hat{j} - 0.3\hat{k})m/s$

In addition to using the feedback matrix K_{LQR} computed by LQR in MATLAB, intuitive manual tuning of the K matrix was tested out in this implementation. In the process, we stumbled upon another interesting finding, the robot moved its torso in the opposite direction of the external disturbance in an attempt to maintain its CoM within the support polygon rather than trying to lunge its torso to generate angular moment of inertia as discussed earlier. This implementation was also able to balance the robot from smaller external disturbances. The intuition behind this attempt was that the robot should move the torso further back based on how much the pelvis moved in x-direction. Figure 6.11 shows the robot recovering from an impulsive disturbances that changes velocity of pelvis from rest to 0.33m/s using this implementation. The manually tuned K matrix used for this implementation is

$$K = \begin{bmatrix} 120 & 25.695 & 0 & 0 \\ 800 & 0 & 800 & 30.3 \end{bmatrix}$$

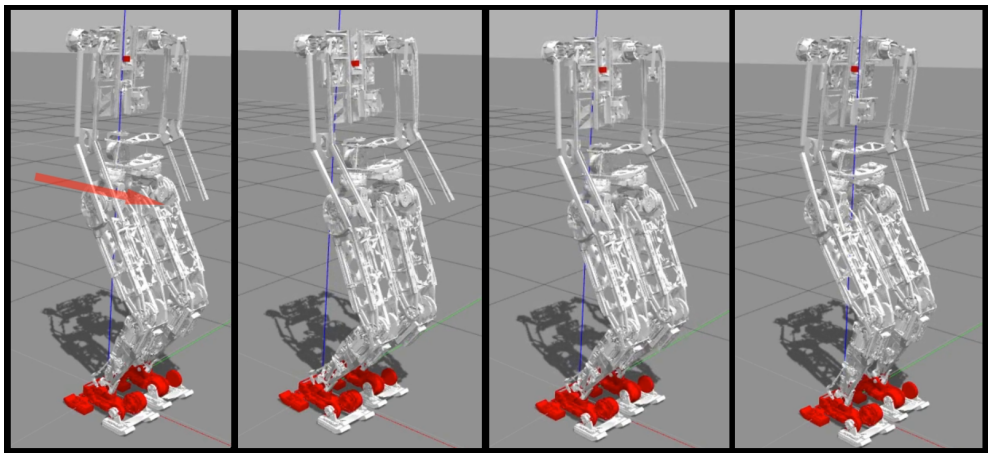


Figure 6.11: Gazebo simulation response using constrained LQR on LDIP model with modified K to an impulsive disturbances that changes velocity of pelvis from rest to 0.33m/s. Video Link : https://youtu.be/QFVxtds2hbI

Chapter 7

Conclusion

7.1 Conclusion

This work presents the Linearized Double Inverted Pendulum model for humanoid robots' standing balance and push recovery. The simplified model efficiently describes the system with linearization of angular position and velocity of the torso to recover from external disturbances. The implicit modeling of the angular moment of inertia about the pelvis of the robot helps in using modern control techniques with ease. The LDIP model can use ankle and hip recovery strategies while taking full advantage of the advances in optimal controls theory research.

To support this, an LQR based control architecture is also presented to verify the model performance. A comparison between LDIP and two other models show that the presented control architecture achieves the task at hand with minimal effort and force requirements. This enables the robot to recover from a wider range of disturbance without the need for taking a step.

The model is ideal for biped robots with a torso that have a significant mass. This model will not be a good fit for robots with small and light torso like Cassie. Another drawback of this model is that it would be difficult to model take into account the angular momentum generated by rotating the arms of the robot rapidly. But on the other hand, this model does a very good job of taking into account the angular momentum generated by lunging the torso. This is why, it would be an ideal fit a humanoid robot like TigerBot-VII.

Chapter 8

Future Work

8.1 Future Research

The current implementation of LDIP model majorly addresses push recovery with ankle and hip strategies. For future work, we must conduct a study on how to come up with an empirical formulation of orbital energy for LDIP model that will allow us to use concepts like capture point and capture regions for LDIP model as well. This would further allow us to identify when the controller will fail to recover from the push and the robot will need to take a step to avoid falling. Further studies on LDIP model can be conducted to analyse its effectiveness for walking gait generation methodologies as well.

It would be interesting to see the algorithm performance when the state variable $x = 0$ point is considered as the CoP in the Gazebo simulation using the force/torque sensor added in the simulation model. This should theoretically increase the recovery capabilities of the robot according to theory of ZMP and robot dynamics [2].

8.2 Issues and suggestions for TigerBot-VII

The proposed Linearized Double Inverted Pendulum model has currently been tested in simulation environment. The model needs to be tested on the real robot, but there were a couple of issues identified in the robot which need to be addressed, before this or any other research can be conducted using the existing hardware. Following are the issues identified in the robot and recommended modifications to tackle these issues.

1. Since TigerBot-VII is built out of separate machined parts bolted together, the body parts at some places are under stress due to machining inaccuracies. This is especially noticeable at the knee joints and has an adverse effect on harmonic drive-pulley-bearing mechanism in the knee joints, shown in Figure 8.1. Due to the stress, the axis of rotation of

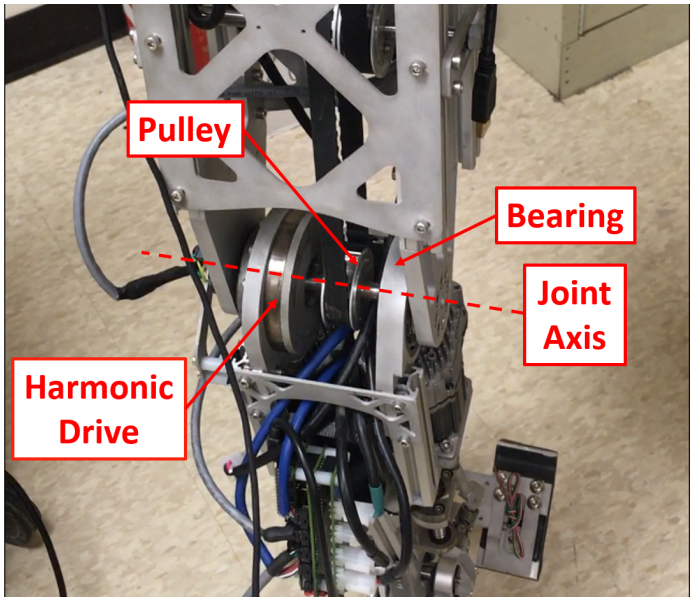


Figure 8.1: Harmonic drive-pulley-bearing mechanism in knee joint.

the bearing and the harmonic drive do not coincide which gives a jerky motion in the knee joint. Adding a beam coupler in the shaft should fix this issue by relieving the stress from the shaft and gaining some tolerance for misalignment of the bearing and the harmonic drive.

2. The actuators used in the robot are not suited for this application. The SDSK series motors by Teknic used in the robot is a stepper motor and cannot be used in torque controlled mode like its MCxx series variants, which is a pre-requisite for this application. Furthermore, even though these motors have a position encoder and output torque measurements computed internally, the user cannot access this information during operation. The user must use an external current measurement circuitry or external absolute encoders to get this information.

3. The inaccurate placement of external encoders can result in a slippage in the encoder disc which is very dangerous. Not knowing if the leg is moving or not while giving it a move command can result in the robot colliding into itself and damaging itself. An issue of similar nature was observed in the knee joint encoder which has been temporarily fixed but needs a better and a more permanent solution, shown in Figure 8.2. Such issues also warrant for usage of hard limit switches to stop the robot before it collides with itself rather than relying on the software limits.

4. The current firmware implementation for each Teensy node that controls the joint motors and encoders is not optimal, shown in Figure 8.3. Each Teensy acts as a separate ROS node and is responsible for controlling

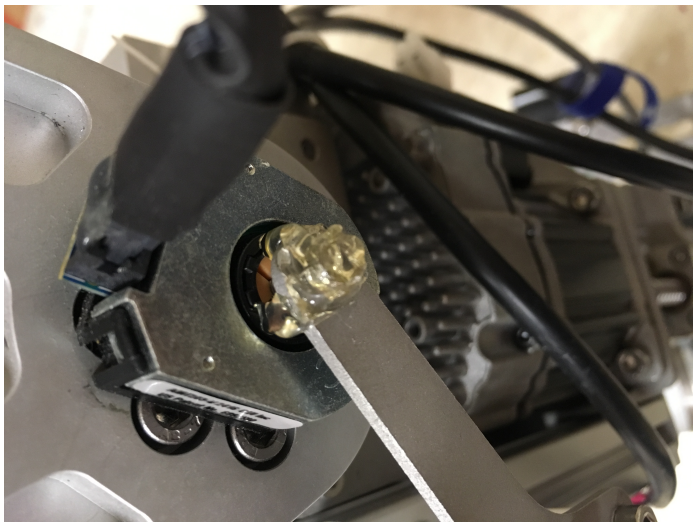


Figure 8.2: Temporary fix for the slippage in the external encoder of the knee joint.

position of motors, reading the encoders and reporting back joint states for up to 3 joints. This works fine as long as the rate of message transfer is low. To optimise communication rate and reduce ROS message overhead, making separate Teensy ROS nodes should be avoided. Instead, a script in the Odroid should command the motors and read the encoder information through serial communication and not use rosserial. The separate script in the Odroid can then act as a single ROS node.

5. There is a lot of discrepancy between the URDF of TigerBot-VII and the real robot. Specifically, the URDF and the mesh models of each link does not include the actuators as part of the link. Therefore, their masses and inertia do not contribute to the dynamics of the simulation model which makes it different from the real robot. Tables 6.1 and 5.1 shows that the total mass in the URDF excluding feet mass is 24.291475 kg, whereas

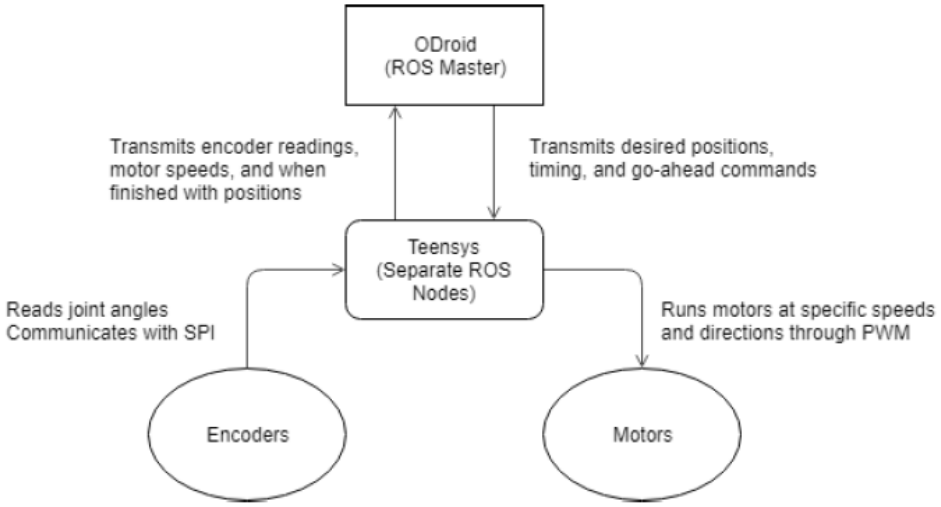


Figure 8.3: Current ROS implementation for TigerBot-VII Hardware communication.

the total mass of the real robot is 51.5101 kg. The actuator models is also missing visually in the simulation model as shown in Figure 5.7. These need to added in the CAD model and generate the meshes and URDF again from Solidworks.

6. The ankle joint of TigerBot-VII is designed to provide 2 DOF motion at the ankle, shown in Figure 8.4. The design intends to put most of the weight of the robot at the ankle joint and uses the Fibula for stability and ankle joint motion. The design is intended to put minimal weight on the Fibula, but it was discovered that although the fibula itself is capable of handling that weight, the tibiofibular joint is too weak and we may run into a problem where that joint can break and pop out of its housing. This issue needs to be addressed by reinforcing the tibiofibular joint which is the weakest point in the joint.

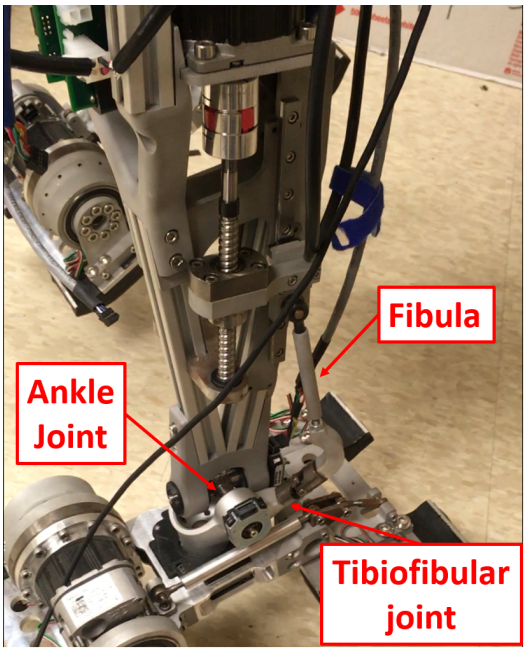


Figure 8.4: Tibiofibular joint in the ankle.

7. TigerBot-VII has a Six-Axis Force/Torque sensor in each foot under the ankle joint for precise force/torque feedback, shown in Figure 8.5. The sensor housing block contains strain gauges, Wheatstone half-bridges to amplify the readings, and analog-to-digital converters for translating the readings. Load cells built into the foot are also connected to the AD converters. The sensor also houses a Teensy 3.2 that is used to take sensor readings and can be made into a separate ROS node and connected to Odroid which acts as the ROS master in the system. The sensor is missing a code base for Teensy. To make the sensor functioning, the electrical schematics and sensor formulation must be understood first. This is a challenge due to the fact that clear instructions and formulations are not available in the sensor documentation.

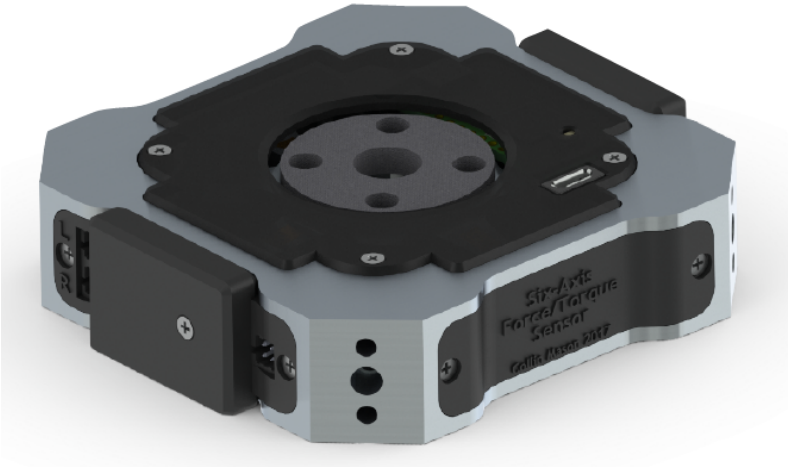


Figure 8.5: Modular Six-Axis Force/Torque sensor in the ankle.

Bibliography

[1] F. Sze, "Simulation and framework for the humanoid robot tigerbot," 2018.

[2] K. Harada, H. Hirukawa, S. Kajita, and K. Yokoi, *Introduction to humanoid robotics*. Springer, 2014.

[3] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 1405–1411 vol.2, 1991.

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

[5] M. Vukobratovic and B. Borovac, "Zero-moment point — thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, 2004.

[6] J. Kim, "Multi-axis force-torque sensors for measuring zero-moment point in humanoid robots: A review," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1126–1141, 2020.

[7] M. B. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *The International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, 2005.

[8] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure-zero moment point," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.

[9] S. Faraji, H. Razavi, and A. J. Ijspeert, "Bipedal walking and push recovery with a stepping strategy based on time-projection control," *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 587–611, 2019.

[10] A. Elhasairi and A. Pechev, "Humanoid robot balance control using the spherical inverted pendulum mode," *Frontiers in Robotics and AI*, vol. 2, p. 21, 2015.

[11] M. Shafiee-Ashtiani, A. Yousefi-Koma, R. Mirjalili, H. Maleki, and M. Karimi, "Push recovery of a position-controlled humanoid robot based on capture point feedback control," *CoRR*, vol. abs/1710.10598, 2017.

[12] Y. Wang, Q. Zhu, R. Xiong, and J. Chu, "Standing balance control for position control-based humanoid robot," *IFAC Proceedings Volumes*, vol. 46,

no. 20, pp. 429 – 436, 2013. 3rd IFAC Conference on Intelligent Control and Automation Science ICONS 2013.

[13] B. Stephens, "Integral control of humanoid balance," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4020–4027, 2007.

[14] B. Stephens, "Humanoid push recovery," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pp. 589–595, 2007.

[15] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1248–1255, 2010.

[16] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 52–59, 2010.

[17] A. Abate, *Mechanical Design for Robot Locomotion*. PhD dissertation, Oregon State University, 2018.

[18] T. Apgar, P. Clary, K. Green, A. Fern, and J. Hurst, "Fast online trajectory optimization for the bipedal robot cassie," 06 2018.

[19] J. Reher, W. Ma, and A. D. Ames, "Dynamic walking with compliance on a cassie bipedal robot," in *2019 18th European Control Conference (ECC)*, pp. 2589–2595, 2019.

[20] H. Geyer and U. Saranli, *Gait Based on the Spring-Loaded Inverted Pendulum*, pp. 1–25. Dordrecht: Springer Netherlands, 2018.

[21] Teknic, Inc., *Clearpath User Manual*.

Chapter 9

Appendix

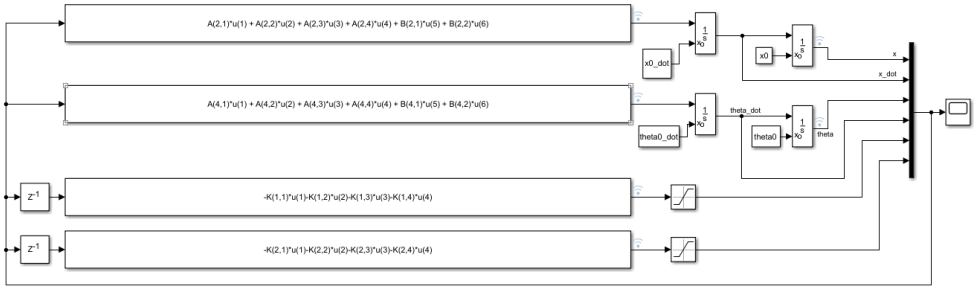


Figure 9.1: MATLAB implementation of LDIP Model.