ROBUST WHOLE-BODY CONTROL FOR LEGGED ROBOTS

A Thesis

by

Dilay Yeşildağ Oral

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master's Degree

in the
Department of Mechanical Engineering

Özyeğin University
August 2022

ROBUST WHOLE-BODY CONTROL FOR LEGGED ROBOTS

Advisor: Asst. Prof. R. Barkan Uğurlu
Coadvisor: Prof. Dr. Duygun Erol Barkana

Approved by:

Asst. Prof. R. Barkan Uğurlu, Advisor
Department of Mechanical Engineering
Özyeğin University

Asst. Prof. Yeşim Öniz
Department of Mechatronics Engineering
İstanbul Bilgi University

Asst. Prof. Ramazan Ünal
Department of Mechanical Engineering
Özyeğin University

Date Approved: 12 August 2022

To my family

ABSTRACT

This thesis aims to propose a robust whole-body locomotion controller for legged robots. To this end, it offers the centroidal momentum observer control algorithm, which could be a very useful tool for providing robust dynamic motion control in eliminating parameter uncertainty for legged locomotion. The control method based on centroidal momentum dynamics is essential for whole-body control. The method considers floating base dynamics when synthesizing controllers such that the base frame is not firmly connected to the ground; the base frame is freely floating. Therefore it can be applied to a wide range of mobile robotic systems. The method was tested using a simulated one-legged robot and whole-body humanoid models. As a result, we observe that the centroidal momentum observer control algorithm could be beneficial for whole-body robot robustness and stabilization.

ÖZETÇE

Bu tez, bacaklı robotlar için gürbüz bir tüm vücut hareket kontrolörü önermeyi amaçlamaktadır. Bu amaçla, bacaklı hareket için parametre belirsizliğini ortadan kaldırmada gürbüz dinamik hareket kontrolü sağlamak için çok yararlı bir araç olabilecek merkezi momentum gözlemci kontrol algoritmasını sunar. Merkezi momentum dinamiklerine dayalı kontrol yöntemi, tüm vücut kontrolü için esastır. Yöntem, temel koordinat sistemi zemine sıkıca bağlı olmayacak şekilde kontrolörleri sentezlerken seyyar koordinat sistemi parametrelerini dikkate alır; temel koordinat sistemi serbestçe yüzer. Bu nedenle çok çeşitli mobil robotik sistemlere uygulanabilir. Yöntem, simüle edilmiş tek bacaklı bir robot ve tüm vücut insansı modelleri kullanılarak test edildi. Sonuç olarak, merkezi momentum gözlemci kontrol algoritmasının tüm gövde bütüncül robotunun gürbüzlüğü ve stabilizasyonu için faydalı olabileceğini gözlemledik.

ACKNOWLEDGEMENTS

I would like to express my endless thanks to my thesis advisor, Asst. Prof. Barkan Uğurlu was always by my side and did not spare his support in all circumstances. My sincere thanks to my friends who are members of Biomechatronics Laboratory of Ozyegin University. Finally, I would like to thanks my husband and family for supporting and motivating me throughout my thesis work.

This work is supported by the Scientific and Technological Research Council of Turkey (TUBITAK), with the project 118E923.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The desire to wonder about the how humans and animals move, and to make devices that improves life quality of people accordingly, led to the development of various legged robots. Biological systems in nature show outstanding performance in load capacity, stability and dynamic behaviour. For this reason, it can often be considered a crucial source of inspiration in designing legged robotic systems for aspects ranging from mechatronics design to motion planning and gait. For example, skating athletes maintain orientation, balance, and speed in the face of the variety and complexity of oscillating, gliding, and moving their bodies through space on the ice. In another example, seeing that a child starts by crawling first, goes from crawling to walking, and from walking to running and jumping over time raises an exciting curiosity about people's legged control mechanisms [3]. They are great interest in animals for their locomotion mechanisms, as they move quickly and reliably from tree to tree in a forest, swamp or rocky environment. One of the best ways to learn about animal and human movement mechanisms is to build machines that move using legs.

In the early twentieth century, it was necessary to create kinematic connections that would produce the appropriate stepping motion. Design walking robots, it is called quest-static walking [4]. Nevertheless, it was realised that a link that provided constant movement may not accomplish the act of dynamic walking or running. Towards the end of the twentieth century, very efficient walking robots with good performances were designed, suitable for practical applications carrying significant loads with the corresponding flexibility and versatility [5].

The scientific study of legged movement began with Muybridge examining the movement of a horse in 1957 [6]. The first actively balanced machine was based on an automatically controlled inverted pendulum. Miura and Shimoyama built the first actively balancing walking machine [7] in 1984. Mark Raibert stated that interest in legged locomotion has two main aims in 1986. These are the high potential for mobility and the need to understand the movements of animals and humans. Since wheeled movement is primarily limited to flat surfaces, it is seen as unfavourable in terms of high mobility. Raibert demonstrated the advantage of legged robots over wheeled systems by showing that legs use isolated feet for support, whereas wheels or tracks require a continuous support path in 1986 [6]. Walking robots have been researched since the beginning of robot technology to overcome the limitations of wheeled systems by inspiring from walking solutions in nature. The legs provide better mobility over unstructured terrain than wheels or tracks because these robots can use isolated legs that optimize support and traction. This feature is a tremendous advantage for increasing speed and efficiency on rough terrain in a legged system.

Using a legged design; the tools used by humans and communication with the environment can be for functional purposes such as the development of bipedal movement. Some humanoid robots may be modelled only a part of the human body, the behaviour and control of a one-legged robot can be used as a basis for each piece of the multi-legged system. Legged robots can be used in industry, medicine, entertainment, agriculture, animal husbandry, transportation and education as well as many other fields. These robots are used in a wide range from aircraft construction to disaster scenarios, from robotic rehabilitation to active walking support [8–11].

Despite several studies, the robustness of legged systems is still an active research question. The dynamic model of legged robots is not as linear as predicted, and it is pretty challenging to model how the robot interacts with its environment with all uncertainties. For this reason, the control of legged robots can make the job a little more complicated. The controller is essential in realizing the stable periodic movement of the legged robots.

The primary purpose of the controller is to ensure the stability of the robot as it provides a dynamically balanced walk. A legged robot is statically balanced when controlled kinematically with negligible inertial forces.

Differently, if a robot is dynamically balanced, it can be balanced by the movement of the robot, taking into account inertial forces and manipulating them correctly. In line with these facts, it can be verified that dynamics play an essential role in the control of legged robots. Various topics related to dynamics, trajectory planning and control have been covered to enable legged robots to walk, run and balance.

One of these studies often uses the zero-moment point (ZMP) for model building and gait control. This method does not require precise knowledge of the robot dynamics, which includes the total mass, the aggregate position of the center of mass. Thus, the accuracy of the robot model may not be important [12, 13]. There are other that use abstracted models, and since the controller knows little about the system structure, this approach relies on feedback control [14, 15]. In this method, they often use an inverted pendulum model. In the second method, the researchers proposed a gait control and patterning method in which dynamic bipedal gait was successfully performed in simulations and experiments [16]. However, since the method creates a stable gait by changing the foot placements, the desired efficiency could not be obtained on flat surfaces. Therefore, in pendulum-based or ZMP-based methods [16–18], it is assumed that an abstracted model represents the robot dynamics, which may require dynamics filtering [19].

In response to that matter, optimization-based control methods emerged in recent years. In these methods, the controller is synthesized via solving a real-time optimal control problem in which an abstracted model is used to derive an approximate value function [20, 21]. Despite this, the method can still deal with the dynamic and contact constraints that arise due to full robot dynamics.

When regulating whole-body robot behaviour, angular momentum appears to be an important quantity [22, 23]. Kajita et al. proposed resolved momentum control to automatically coordinate upper and lower body movements [24]. Abdallah and Goswami [25] use a momentum controller to absorb disruptive effects. They suggest that people can absorb the impact of significant external disturbances by maintaining momentum for a certain period. After the effect is absorbed, the character regains his stance. Orin et al. derived the notion of centroidal momentum for a multibody floating-base system [26, 27] which led to the synthesis of synergistic whole-body humanoid control. [2]. Furthermore, recently developed optimization-based algorithms allow the centroidal momentum-based trajectory control in real-time [28, 29].

Note that the notion of centroidal momentum is sensitive to parametric uncertainties, as it is a model-based algorithm. This creates a particular disadvantage in yielding robust locomotion control behaviour, potentially limiting its applicability to settings with internal and external disturbances. Meanwhile, disturbance observer (DOB) is frequently applied and has been found to be effective in parameter uncertainties for motion control. It is a very useful robust control tool, owing to its efficacy and real-time applicability [30–32]. It can be intuitively adjusted to the desired bandwidth as it plays an essential role in robust control. It is also widely used in robotics, industrial automation, and automotive because of its effectiveness in compensating for distortion and the effect of model uncertainties [33, 34].

In light of these facts, we propose a method called the Centroidal Momentum Observer (CMD), in which a disturbance observer block is adapted to a centroidal momentum-based motion control scheme. This study aims to increase the legged systems' robustness to internal and external perturbations while addressing balancing control. Therefore, the main contribution of this work is to pave the way for robust whole-body control schemes for legged systems in which centroidal momentum control is synergistically combined with a disturbance observation scheme.

CHAPTER II

CENTROIDAL DYNAMICS

In this chapter, floating base dynamics, centroidal momentum and matrix calculations will be discussed.

2.1 Floating base dynamics

Legged robots are designed as robotic mechanisms with various configurations, e.g., one-legged, bipedal, quadrupedal, etc., depending on the application scenario, environment, and the speed of use. During this progress, it became essential to implement multi-body dynamics. A fixed-based robotic system is tightly coupled to an inertial frame and in case of no mechanical failure, the fixtures are supposed to ensure that the base is stationary despite robot motion. On the contrary, this is not the case for a floating-based robotic system and the motion of floating base is tightly coupled with robot motion.

Legged robots are could appear in various configurations depending how they are balanced statically and dynamically. They may have variable design parameters and could move with excessive joint accelerations. Therefore, there are times they may not be dynamically balanced if the center of pressure (CoP) reaches at one of the edges of the polygon that is formed by the support legs. For this reason, the control of legged robots is more challenging than expected, and dynamics play a significant role in obtaining stable motion.

Therefore, the first thing to do is to estimate the position and direction of this floating base before applying the desired control. Suppose that the floating base is represented with the underscript f. A legged robot's floating base, could be described with 6 coordinates

(q_f); three linear coordinates and three angular coordinates along the x-y-z axes, respectively.

$$q = \begin{bmatrix} q_f \\ q_r \end{bmatrix} \tag{1}$$

Currently, there is no sensor that allow direct measurement of the absolute position and orientation. To overcome this issue, one can use motion capture systems.

The dynamic model of a legged system can be acquired by a set of equations obtained using the physical laws of motion. It is easier for a legged system with fewer links and joints to obtain the equations of motion, while for systems with a complex legged with many links, finding the equations of motion is more complicated. Various techniques have been presented to facilitate the derivation of equations of motion. The derivation and propagation of the equations of motion have an essential place for dynamic analysis. A multi-body dynamics equation formulated for many applications in floating-based robots is shared below.

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau + J(q)^T F \tag{2}$$

In the above equation $M(q)$ is mass matrix, q,$\dot{q}$ and $\ddot{q}$ are generalized position velocity and acceleration vector, $b(q,\dot{q})$ are Coriolis and centrifugal terms, $g(q)$ is gravitational terms, τ is external generalized force. $J(q)$ is the Jacobian matrix of a contact, where F denotes the external wrench acting on the given contact.

While one can utilize Lagrangian and/or Hamiltonian approaches to derive dynamics symbolically, the expressions could gets larger and larger for sytem with multiple degree of freedom. To remedy this issue, one can use Newton-Euler algorithm which recursively computes the robot dynamics in a computationally efficient fashion.

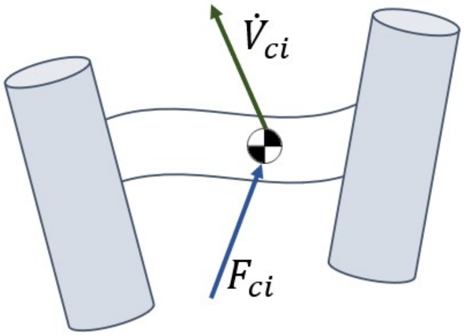


Figure 1: Force acting between links

2.1.1 Newton-Euler Method

The Newton-Euler algorithm recursively computes the velocities and forces that are propagating from one link to another. When dealing with the equations of motion of multi-body systems, it is an appropriate method to separate all bodies from joints and treat each body as a single unit while considering the propagation between the links. For this reason, the restraining forces at the joints should be considered external forces acting on each body when freely modelled. For all the bodies, Newton's 2nd law and Euler's equations of motion can be applied to all bodies that are subject to external forces and moments. Multi-body systems can exhibit movements compatible with the constraints imposed by the joints that limit the relative motion between the links. Considering the $i^t h$ body in Figure 1 and Figure 2, the following equations are obtained.

$$F_c = m\dot{V}_{c_i} \tag{3}$$

$$N = {}^{c_i}I\dot{w} + w \times {}^{c_i}Iw \tag{4}$$

In (3) and (4), N_i and F_{ci} are the resultant moment and force acting on the i^{th} link CoM. m is the mass of i^{th} link, $\dot{V}_{ci}$ is is linear acceleration of the i^{th} CoM, ${}^{c_i}I$ inertia tensor with

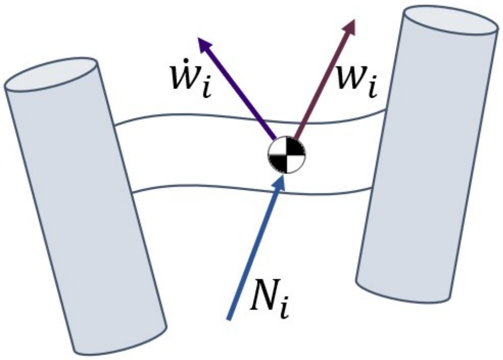


Figure 2: Momentum exerted between links

respect to the center of mass(CoM), w and $\dot{w}$ are the angular velocity and acceleration of the link. The c_i frame has its origin at the i^{th} link CoM and has an identical orientation with the i link frame. To derive the angular acceleration propagation from link to link;

$$ {}^{i}w_{i+1} = {}^{i}w_i + {}^{i}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{5} $$

In Equation (5), ${}^{i}w_{i+1}$ angular velocity of the $(i+1)^{th}$ link respect to i^{th} link, ${}^{i}w_i$ angular velocity of the i^{th} respect to i^{th} link, ${}^{i}_{i+1}R$ rotation matrix of the $(i+1)^{th}$ link respect to i^{th} link ,$\dot{Q}_{i+1}$ joint velocity vector of the $(i+1)^{th}$ link, S_{i+1} unit vector along the rotation axis. All differentiations need to take place for vectors represented in the inertial frame. If a vector is represented in a body frame, first perform a coordinate transformation to the inertial frame, then differentiate, and lastly perform a coordinate transformation again back to the body frame. Therefore, to compute ${}^{i+1}w_{i+1}$ from equation (5), multiplying both sides of the equation by ${}^{0}_{i+1}R$ before the differentiation;

$$ {}^{0}_{i+1}R^{i+1}w_{i+1} = {}^{0}_{i+1}R^{i+1}{}_{i}R^{i}w_i + {}^{0}_{i+1}R^{i}{}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{6} $$

This equation can also be represented as follows.

$$ {}^{0}w_{i+1} = {}^{0}w_i + {}^{0}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{7} $$

If we differentiate both sides of (7), the following is yielded.

$$ {}^{0}\dot{w}_{i+1} = {}^{0}\dot{w}_{i} + {}^{0}_{i+1}R\ddot{Q}_{i+1}S_{i+1} + {}^{0}w_{i+1} \times {}^{0}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{8} $$

If we multiply the both sides of (8) with ${}^{i+1}{}_{0}R$, the equations below can be derived sequentially.

$$ {}^{i+1}{}_{0}R^{0}\dot{w}_{i+1} = {}^{i+1}{}_{0}R^{0}\dot{w}_{i} + {}^{i+1}{}_{0}R^{0}{}_{i+1}R\ddot{Q}_{i+1}S_{i+1} + {}^{i+1}{}_{0}R^{0}w_{i+1} \times {}^{0}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{9} $$

$$ {}^{i+1}\dot{w}_{i+1} = {}^{i+1}\dot{w}_{i} + \ddot{Q}_{i+1}S_{i+1} + {}^{i+1}w_{i+1} \times {}^{0}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{10} $$

Observing (10), ${}^{i+1}w_{i+1}$ needs to be avoided, so when the ${}^{i+1}w_{i+1}$ expression is put back in its place;

$$ {}^{i+1}\dot{w}_{i+1} = {}^{i+1}\dot{w}_{i} + \ddot{Q}_{i+1}S_{i+1} + ({}^{i+1}{}_{i}R^{i}w_{i} + \dot{Q}_{i+1}S_{i+1}) \times {}^{0}_{i+1}R\dot{Q}_{i+1}S_{i+1} \tag{11} $$

This equation can also be represented as follows.

$$ {}^{i+1}\dot{w}_{i+1} = {}^{i+1}{}_{i}R^{i}\dot{w}_{i} + {}^{i+1}{}_{i}R^{i}w_{i} \times \dot{Q}_{i+1}S_{i+1} + \ddot{Q}_{i+1}S_{i+1} \tag{12} $$

Examining (12), angular acceleration is obtained. For translational acceleration, let us first start with the linear velocity expression as follows:

$$ {}^{i}V_{i+1} = {}^{i}Vi + {}^{i}w_{i} \times {}^{i}P_{i+1} \tag{13} $$

Equation (13), ${}^{i}V_{i+1}$ translational velocity of the $(i+1)^{th}$ link respect to i^{th} link, ${}^{i}V_{i}$ translational velocity of the i^{th} respect to i^{th} link, ${}^{i}_{i+1}P$ origin of the $(i+1)^{th}$ link as seen from the origin of i^{th} link ,${}^{i}w_{i}$ angular velocity of the i^{th} respect to i^{th} link. We multiply both sides of (13) with ${}^{0}{}_{i}R$ and then differentiate it to obtain the following equations, subsequently:

$${}^{0}{}_{i}R^{i}V_{i+1} = {}^{0}{}_{i}R^{i}Vi + {}^{0}{}_{i}R^{i}w_i \times {}^{0}{}_{i}R^{ii}P_{i+1} \tag{14}$$

$${}^{0}V_{i+1} = {}^{0}V_i + {}^{0}w_i \times {}^{0}{}_{i}R^{ii}P_{i+1} \tag{15}$$

Since translational acceleration is required, both sides of equation (15) must be differentiated,

$${}^{0}\dot{V}_{i+1} = {}^{0}\dot{V}_i + {}^{0}\dot{w}_i \times {}^{0}{}_{i}R^{i}P_{i+1} + {}^{0}w_i \times ({}^{0}w_i \times {}^{0}{}_{i}R^{i}P_{i+1}) \tag{16}$$

Once again, multiplying both sides ${}^{i}{}_{0}R$ to get the translational acceleration desired in the equation (16).

$${}^{i}{}_{0}R^{0}\dot{V}_{i+1} = {}^{i}{}_{0}R^{0}\dot{V}_i + {}^{i}{}_{0}R^{0}\dot{w}_i \times {}^{i}{}_{0}R^{0}{}_{i}R^{i}P_{i+1} + {}^{i}{}_{0}R^{0}w_i \times ({}^{i}{}_{0}R^{0}w_i \times {}^{i}{}_{0}R^{0}{}_{i}R^{i}P_{i+1}) \tag{17}$$

Thus, the following equation is obtained;

$${}^{i}\dot{V}_{i+1} = {}^{i}\dot{V}_i + {}^{i}\dot{w}_i \times {}^{i}P_{i+1} + {}^{i}w_i \times ({}^{i}w_i \times {}^{i}P_{i+1}) \tag{18}$$

Finally, multiply both sides of the (18) equation by ${}^{i}+1_{i}R$ to get the desired translational acceleration.

$${}^{i+1}{}_{i}R^{i}\dot{V}_{i+1} = {}^{i+1}{}_{i}R({}^{i}\dot{V}_i + {}^{i}\dot{w}_i \times {}^{i}P_{i+1} + {}^{i}w_i \times ({}^{i}w_i \times {}^{i}P_{i+1})) \tag{19}$$

The simplified version of the equation is as follows:

$${}^{i+1}\dot{V}_{i+1} = {}^{i+1}{}_{i}R({}^{i}\dot{V}_i + {}^{i}\dot{w}_i \times {}^{i}P_{i+1} + {}^{i}w_i \times ({}^{i}w_i \times {}^{i}P_{i+1})) \tag{20}$$

Likewise, translational acceleration of an individual link is computed as follows;

$${}^{i}\dot{V}_{ci} = {}^{i}\dot{w}_i \times {}^{i}P_{ci} + {}^{i}w_i \times ({}^{i}w_i \times {}^{i}P_{ci}) + {}^{i}\dot{V}_i \tag{21}$$

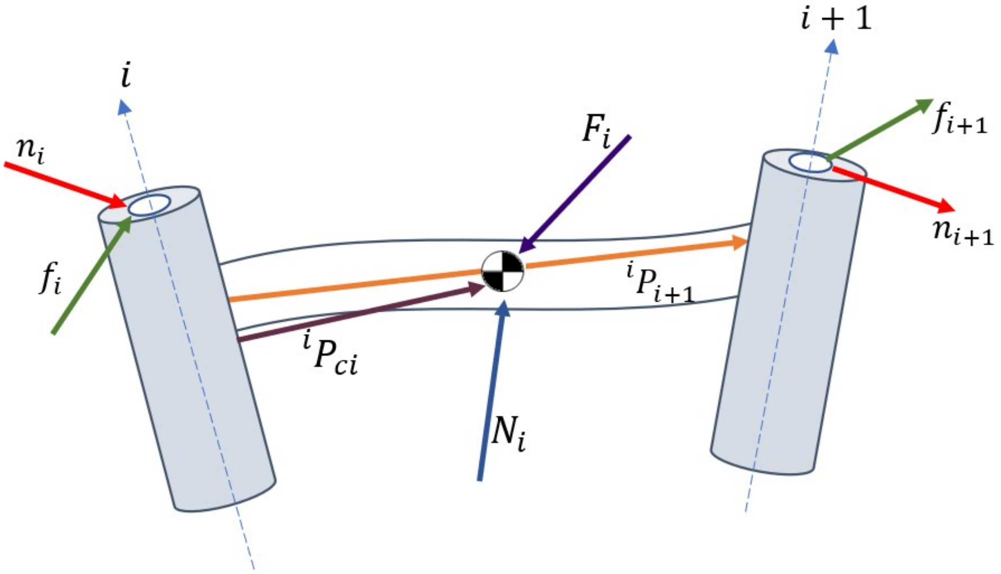


Figure 3: Torque and force exerted between links

When the equation (3) is considered again in the direction of the angular and translational accelerations obtained above, the following equation can be obtained from the figure 3.

$$ {}^iF_i = {}^if_i - {}^if_{i+1} \tag{22}$$

As can be seen in equation (22), ${}^if_i$ force exerted on the link i by link i and ${}^if_{i+1}$ force exerted on the link $i+1$ by link i. According to Newton's 3rd law, as link i exerts f_{i+1} on link $i+1$, link $i+1$ exerts $-f_{i+1}$ on link i. Thus, the equation (22) holds true. If the equation rewrite,

$$ {}^iF_i = {}^if_i - {}^iR_{i+1}{}^if_{i+1} \tag{23}$$

We follow a similar procedure for (4) to obtain joint torque.

$$ {}^iN_i = {}^in_i - {}^in_{i+1} + (-P_{ci} \times {}^if_i) - ({}^iP_{i+1} - {}^iP_{c_i}) \times {}^if_{i+1} \tag{24}$$

In the above equation ${}^in_i$ torque exerted on link i by link i, ${}^in_{i+1}$ torque exerted on link $i+1$ by link i, ${}^if_i$ force exerted on the link i by link i and ${}^if_{i+1}$ force exerted on the link $i+1$ by link i, ${}^i_{i+1}P$ origin of the $(i+1)^{th}$ link as seen from the origin of i^{th} link, ${}^iP_{c_i}$ origin of the CoM of i^{th} link as seen from the origin of i_{th} link. Since it is computed with respect to i, it can be used equation which is that $-{}^iP_{c_i} = {}^{c_i}P_i$. Thus, if equation (24) is rearranged,

$$
{}^iN_i = {}^in_i - {}^iR_{i+1}{}^i{}_{i+1} - P_{ci} \times {}^iF_i - {}^iP_{c_i} \times {}^i_{i+1}R^{i+1}f_{i+1} - {}^iP_{i+1} \times {}^i_{i+1}R^{i+1}f_{i+1} + {}^iP_{c_i} \times {}^i_{i+1}R^{i+1}f_{i+1} \tag{25}
$$

The simplified version of the equation is as follows:

$$
{}^iN_i = {}^in_i - {}^iR_{i+1}{}^i{}_{i+1} - P_{ci} \times {}^iF_i - {}^iP_{i+1} \times {}^i_{i+1}R^{i+1}f_{i+1} \tag{26}
$$

When the last equations (23) and (26), ${}^iF_i$ and ${}^iN_i$ are rewritten to yield ${}^if_i$ and ${}^in_i$;

$$
{}^if_i = {}^iF_i + {}^iR_{i+1}{}^if_{i+1} \tag{27}
$$

$$
{}^in_i = {}^iN_i + {}^iR_{i+1}{}^i{}_{i+1} + P_{ci} \times {}^iF_i + {}^iP_{i+1} \times {}^i_{i+1}R^{i+1}f_{i+1} \tag{28}
$$

2.2 Centroidal Momentum

The spatial momentum vector of a robot consists of linear and angular momentum. Following the derivation provided in [2], the spatial momentum concerning the i^{th} link is calculated as below:

$$
h_i = \begin{bmatrix} k_i \\ l_i \end{bmatrix} = I_i \nu_i = I_i \begin{bmatrix} w_i \\ \upsilon_i \end{bmatrix} \tag{29}
$$

In (29), h_i is the spatial momentum of the i^{th} link which consists of angular momentum k_i and linear momentum l_i. Likewise, ν_i stands for the spatial velocity concerning the i^{th} link, including angular velocity w_i and translational velocity υ_i . The spatial momentum can be obtained by multiplying v_i with the spatial inertia I_i which can be computed as below.

$$I_i = \begin{bmatrix} \bar{I}_i & m_i S(c_i) \\ m_i S(c_i)^\intercal & m_i \end{bmatrix} \tag{30}$$

As observed in (30), I_i is calculated using mass m_i, relative center of mass position c_i, and rotational inertia of the i^{th} link. Furthermore, $S(c_i)$ is the skew-symmetric matrix concerning the c_i vector. The centroidal momentum of a multi-dof system can be obtained through the sum of individual momenta of each link, provided that they are projected onto the CoM. To do so, we place a coordinate frame onto the CoM, named $\{G\}$ which is oriented in an identical manner to fixed frame of reference (FFR) $\{0\}$ [2]. Keeping this in mind, the sum of indiviual momenta can be computed as follows:

$$h_G = \sum_{i=1}^{N_b} {}^{i}X_G^\intercal h_i \tag{31}$$

In (31), ${}^{i}X_G^\intercal$ is the spatial transformation matrix defined between the i^{th} link and $\{G\}$. It is computed as in the following:

$${}^{i}X_G^\intercal = \begin{bmatrix} {}^{G}R_i & {}^{G}R_i S({}^{i}p_G)^\intercal \\ 0 & {}^{G}R_i \end{bmatrix} \tag{32}$$

where ${}^{i}R_G$ and ${}^{i}p_G$ are the rotation matrix and position vector defined between the i^{th} link and $\{G\}$, respectively.

The centroidal momentum can also be expressed as follows,

$$h_G = A_G \dot{q}, \tag{33}$$

where $A_G(q)$ is the centroidal momentum matrix, mapping h_G to $\dot{q}$. The vector $\dot{q} \in \mathbb{R}^{(n+6)}$ stands for the generalized velocities, including actuated joints and floating coordinates. One can also compute the net wrench acting on the CoM by differentiating (33):

$$f_G = \dot{h}_G = A_G \ddot{q} + \dot{A}_G \dot{q} \tag{34}$$

The main function of the CMM ($A_G(q)$) is to provide a linear mapping between the generalized velocities and the centroidal momentum. To derive the ($A_G(q)$) matrix, floating base dynamics can be used [35]. To this end, 6D vector approach is deemed to be computationally efficient [36]. For a general robotic system, the equation that governs rigid body dynamics is expressed as below,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{35}$$

where $q \in \mathbb{R}^{(n+6)}$ stands for the generalized coordinates and $M(q) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the joint-space inertia matrix, $C(q,\dot{q})\dot{q} \in \mathbb{R}^{(n+6)}$ is the vector of Coriolis and the centrifugal terms, $g(q) \in \mathbb{R}^{(n+6)}$ is the gravity term, $\tau \in \mathbb{R}^{(n+6)}$ is the generalized force. The eq. (35), can be re-written to separately express floating-base and actual joint components [36]:

$$\begin{bmatrix} M_{11} & M_{1*} \\ M_{*1} & M_{**} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_* \end{bmatrix} + \begin{bmatrix} C_1\dot{q} + g_1 \\ C_*\dot{q} + g_* \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_* \end{bmatrix} \tag{36}$$

In (36), the terms with the underscript $\{1\}$ are associated with the floating base coordinates. Likewise, τ_1 is associated with floating base, such that $\tau_1 = 0$ as they cannot produce force. Although the floating base is not actuated, the following equation is derived via the principle of virtual work [37]:

$$\Phi_1^{\intercal} f_1 = \tau_1 \tag{37}$$

In (37), Φ_1 can be derived via floating-base kinematics [2,38], and it maps τ_1 to the net wrench acting on the floating base, namely f_1. Considering the gravitational force as an external force, the net wrench acting on the floating base can be yielded as in the following:

$$f_1 = (\Phi_1^{\intercal})^{-1} U_1 \left(M(q)\ddot{q} + C(q,\dot{q})\,\dot{q} \right) \tag{38}$$

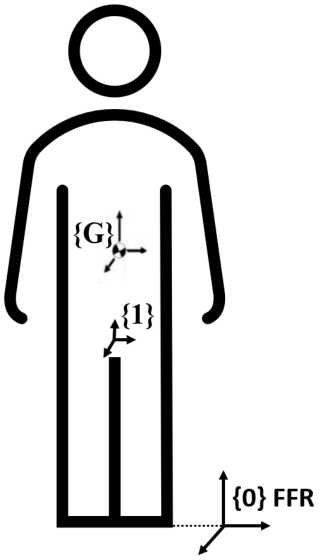


Figure 4: $\{0\}$ is the fixed frame of reference (FFR) , $\{1\}$ is the floating base coordinate system that is fixed under the foot and, the center of gravity $\{G\}$coordinate system is attached to the CoM. This figure is adapted from [2].

In (38), U_1 is defined as $U_1 = [1_{6x6}\ 0_{6xn}]$. Using the spatial transformation matrix between the frames $\{1\}$ and $\{G\}$, the net external wrench acting on the CoM can be obtained.

$$f_G = {}^1X_G^{\top} f_1 \tag{39}$$

Combining (38) with (39), the following equation is yielded.

$$\begin{aligned} f_G &= {}^1X_G^{\top}(\Phi_1^{\top})^{-1}U_1(M\ddot{q}+C\dot{q}) &(40)\\ &= A_G\ddot{q}+\dot{A}_G\dot{q} &(41)\end{aligned}$$

It is known that the eq. (41) holds true for an arbitrary choice of generalized acceleration, $\ddot{q}$. Therefore, the CMM A_G can be obtained as below.

$$A_G = {}^1X_G^T(\Phi_1^T)^{-1}U_1M(q) \tag{42}$$

This result imply that one can compute the centroidal momentum using the generalized coordinates and velocities, which consist of floating base and actuated joint coordinates.

CHAPTER III

CENTROIDAL MOMENTUM-BASED LOCOMOTION CONTROL

Legged robots are increasingly proficient in performing various tasks such as running, jumping, jumping, balancing against impacts, and climbing stairs. However, state-of-the-art controllers usually focus only certain subset of these tasks. Contrary to the situation, humans can multitask and respond instantly to any loss of balance. Based on this fact, whole-body control was developed to respond to the demand for human-like dexterity.

The control method based on centroidal momentum dynamics is essential for whole-body control. Due to the fact that centroidal dynamics govern the complete behaviour of any given system using such controllers. When performing actions such as walking, running, and squatting in humanoid and legged robots, it is possible to react and balance when an unexpected perturbation is received instantly.

3.1 Momentum-Feedback Control

The centroidal momentum feedback control can be seen in Figure 5. In the centroidal momentum feedback control, reference joint velocities can be obtained based on the desired linear and angular momenta. For this operation, the inverse of the $A_G(q)$ matrix must be used. The calculation of $A_G(q)$ was as described in the section *Centroidal Momentum*; see *Chapter 2*. Floating base dynamics are also used to calculate matrices. The floating base coordinates should be measured and inserted into the centroidal momentum matrix. In practice, this can be done using IMU sensors and state estimators [39].

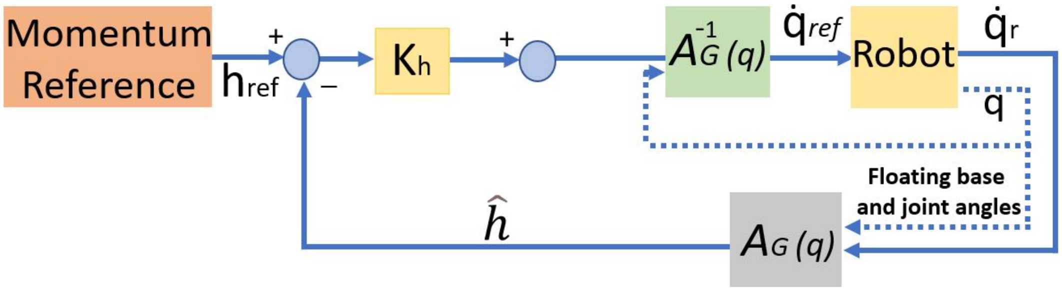


Figure 5: Controller Model of Centroidal Momentum Control. The momentum reference is determined by trajectory planning.

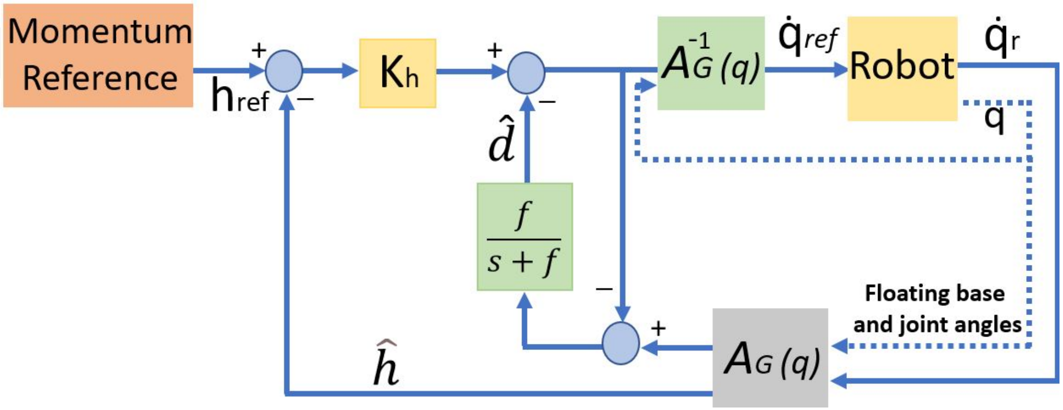
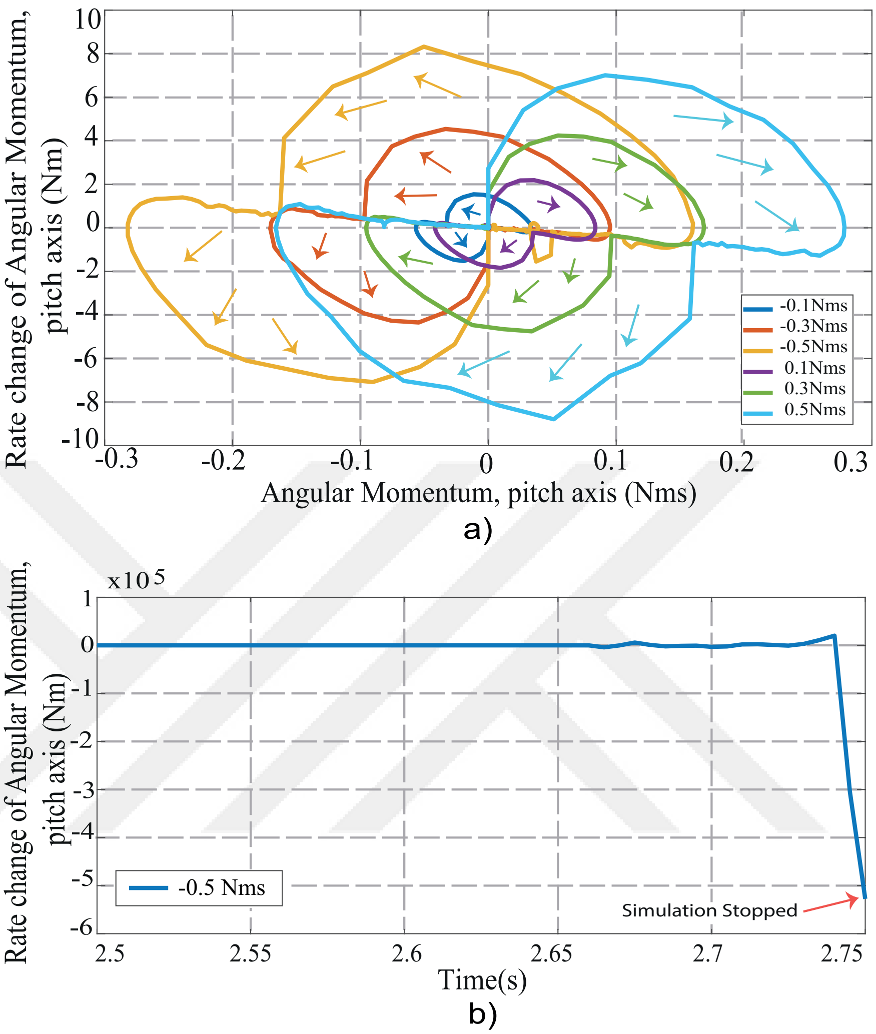


Figure 6: The controller schematic of the proposed Centroidal Momentum Observer. Capped parameters are obtained by estimation. The parameter f is low pass filter gain which is tuned empirically.The momentum reference is determined by trajectory planning.

3.2 Momentum Feedback and Centroidal Momenum Observer

A possible disadvantage with centroidal momentum-based control is model dependency, so it may not be able to deal with factors such as parameter uncertainty and internal disturbances. When parameter uncertainty and internal disturbances cannot be dealt with, the desired whole-body control may not be achieved. To resolve this issue, the disturbance observer can be used, as it is known to provide robustness even in the presence of modelling uncertainties [30, 40], so whole-body behaviour can be improved. Furthermore, reactive balance can be achieved in models such as legged robots, humanoid robots or exoskeletons, despite external factors.



a)

b)

Figure 7: Phases Portraits of Angular momentum

A simplified block diagram can be seen in Figure 6, where we combined a centroidal momentum feedback control and a disturbance observer. The forward calculation of the centroidal momentum matrix is provided as feedback to the momentum reference. K_h is the control parameter for the momentum feedback control.

In order to investigate the stabilization capability of the proposed centroidal momentum observer method, we use the one-legged robot model in Fig. 10. For this purpose, external angular momenta of -0.1 Nms, -0.3 Nms, -0.5 Nms, 0.1 Nms, 0.3 Nms and 0.5 Nms are applied to the robot CoM when DOB is in effect in accordance with the controller displayed in Fig. 6. Moreover, we repeat the same study when DOB is not in effect. The phase portraits concerning both cases are displayed in Fig. 7.

As may be examined in Fig. 7(a), the system maintains its balance upon being perturbed. Regardless of the amplitude of the perturbation within the ±0.5 Nms range, the rate change of angular momentum eventually converges and the robot recovers its balance. In our simulation studies, the proposed controller attenuated disturbances up to ±0.9 Nms. On the contrary, the system could not be stabilized when the DOB is not in effect. Its rate change of angular momentum grows exponentially; see Fig. 7(b). This intermediate result indicates the stabilization capability of the proposed centroidal momentum observer method.

3.3 Baseline Controller for Comparison: ZMP Feedback

The zero moment point (ZMP) indicates the point where the dynamic reaction force of the foot when it comes into contact with the ground does not produce any moment, that is, the point where the sum of the momenta due to inertial and gravitational forces is zero [17]. ZMP has an essential place in motion planning for legged and humanoid robots. Since it is desired to perform movements, such as walking and running, the trajectories must be planned by considering the dynamic balance of the whole body. This could make control very challenging, especially when the robot has greater mass and inertia distributed somewhat equally. For ZMP-based control, the support-polygon region must first be defined, and it is measured by the distance of the zero moment point from the boundaries of this region. The ZMP position is affected by the mass and inertia of the robot body and may require wide range of torques as they are intended to provide stability during movement. In order to eliminate the aforementioned situation, small body movements can be used to stabilize the robot's posture. New planning methods have been developed to define the trajectories of the legs so that the robot's body is oriented to reduce the ankle torque required to provide movement. When the trajectory planning is correct for the leg joints, the ZMP does not go out of the predefined support polygon, the robot's motion follows the desired

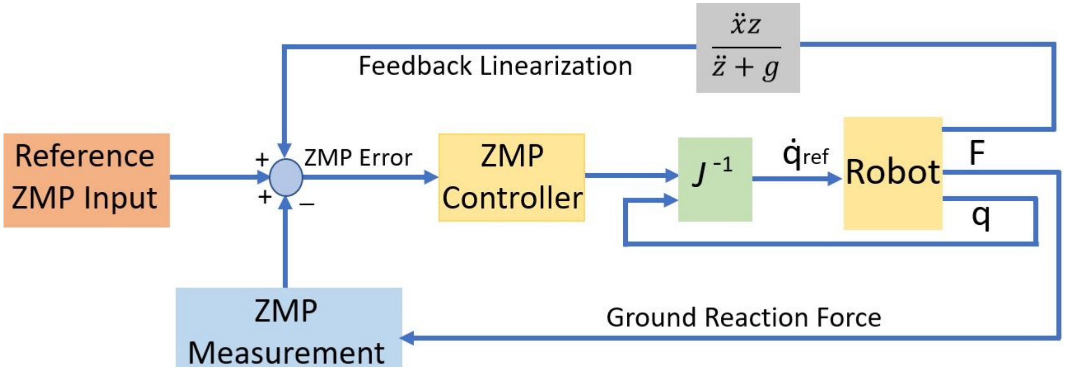


Figure 8: Feedback linearization Controller Model of ZMP Feedback Control

Table 1: The values of the parameters used in the X_{zmp} equation

A	0.008
w	2*pi*0.5
B	0.02
g	9.81 m/ s^2

trajectory, and the selected motion command is applied.

Following the derivation provided in [41], the ZMP equation is calculated in the x and y direction, assuming the rate of change of angular momentum around COM is negligible as below:

$$x_{zmp}(t) = x(t) - \frac{\ddot{x}(t)}{\ddot{z}_{com}+g} z_{com} \tag{43}$$

$$X_{zmpRef} = Asin(w(t-1)) + 0.01271(ReferenceValue) \tag{44}$$

$$\ddot{z}_{com} = -(B(w^2)sin(w(t-1))) \tag{45}$$

$$z_{com} = Bsin(w(t-1)) + 0.27787 \tag{46}$$

$$\ddot{x}_{com} = -(A(w^2)sin(w(t-1))) \tag{47}$$

$$F = \sum_{i=1}^{4} F_z \tag{48}$$

$$X_{zmp} = (x_1 F_{z1} + x_2 F_{z2} + x_3 F_{z3} + x_4 F_{z4})/F \tag{49}$$

In Table 1, the parameters used for the trajectory obtained with the help of this equations (44), (45), (46), (47), (48), (49) are given. In this equation (49), $x_1,...,x_4$ represents the point on all four sides of the foot, and $F_{z1},...,F_{z4}$ represents the force acting on the ground at those points. These values are found by measurement.

A simplified block diagram can be viewed in Fig. 8, where the ZMP feedback controller is constructed. The parts mentioned as *Reference ZMP Input, ZMP Measurument* and *Feedback Linerarizaiton* in the block diagram are given in Table (1). In principle, this method includes a PID controller and a feedback linearization block, and once the CoM is obtained, joint velocity references are computed using the inverse of Jacobian. When the size of the Jacobian matrix is large, the inverse algorithm will be difficult, so the joint velocity and angles vector is divided into two as measurable values depending on the floating base and joint angles and velocities. Then, in one of the iterative methods, as in *The Arnoldi Iteration*, the jacobian matrix corresponding to these velocities is divided into columns similarly. Then the inverse operation can be applied.

Since this method is one of the basic control methods and is a frequently used method, it is calculated to compare with newly designed controllers.

3.4 Centroidal Momentum Observer for a Whole Body Humanoid

In this section, we will talk about the application of Centroidal Momentum Observer Controller for multi body systems. A block diagram can be seen in Figure 6, where trajectory planning for the lower body is with the help then inverse kinematics and PD control. The disturbance observer (DOB) method is applied to the upper body of the robot model by using the centroidal momentum observer method, using the planned trajectories and the angular part of the centroidal momentum matrix. This controller yields the upper body joint velocities in a way to counteract the angular momentum, that is induced due to perturbations. The obtained velocities can also be passed through the high pass filter and integrated and feedback to the system as the position reference.

Calculating angular momentum in equation (70) is a physical quantity related to the inherent balancing capacity of the angular momentum system [35, 42, 43]. Even if the robot is not disturbed, the angular momentum oscillates due to difficulty setting the PD gains precisely or uncertainties in the robot model. In this case, the centroidal momentum observer (CMD) supports the system as it will regulate the system's overall angular momentum, thus making the robot more stable and robust.

3.5 Robot Models via Centroidal Momentum Computation

In this section, the robot models used during our simulation studies are explained.

3.5.1 One legged robot

The one-legged robot used in this study consists of 3 active joints, including a hip joint, a knee joint and an ankle joint, all along the pitch axis [1]. $q_i(i = 1,2,3)$ denote the relative angles between the links $(i-1)$ and the link (i). $\dot{q}_i(i = 1,2,3)$ represents joint velocities corresponding to each angle. Dynamic parameters of the one-legged robot are given in table 2. Since this robot is a planar one, table 2 includes parameters only x and z axes. CoM_x and CoM_z values in the table are the distances of the center of gravity between each joint and the joint relative to the joint frame. MSC Adams simulation model was created using the parameters while the actual robot and its MSC Adams model is displayed in Fig.10. Fig. 11 displays the joint configurations of the one-legged robot and the coordinate systems used for floating base dynamics. The floating base frame was placed at the foot center and the dynamic calculations were performed accordingly. As given in the eq.1, if it is desired to define the joint position and velocity vectors, one must consider floating base parameters, and real joint values robot. Thus, the generalized coordinates can be described

Table 2: Parameters of the one-legged robot [1]

$Mass(kg)$	$Length(m)$	$CoM_x(m)$	$CoM_z(m)$	$Moment of Inertia(kgm^2)$
m_0=5.376	l_0=0.175	c_{0x}=-0.0124	c_{0z}=0.1750	I_0=0.05763
m_1=1.76	l_1=0.150	c_{1x}=-0.0136	c_{1z}=-0.0327	I_1= 0.00469
m_2=1.88	l_2=0.150	c_{2x}=0.0068	c_{2z}=-0.0330	I_2= 0.00547
m_3=1.32	l_3=0.035	c_{3x}=0.0073	c_{3z}=0.0174	I_3=0.00361

as follows. The generalized velocities are the derivative of the joint position vector.

$$q^\top = \begin{bmatrix} q_{f1} & q_{f2} & q_{f3} & q_{f4} & q_{f5} & q_{f6} & q_{r1} & q_{r2} & q_{r3} \end{bmatrix}_{1x9} \tag{50}$$

$$\dot{q}^\top = \begin{bmatrix} \dot{q}_{f1} & \dot{q}_{f2} & \dot{q}_{f3} & \dot{q}_{f4} & \dot{q}_{f5} & \dot{q}_{f6} & \dot{q}_{r1} & \dot{q}_{r2} & \dot{q}_{r3} \end{bmatrix}_{1x9} \tag{51}$$

In the direction of the generalized coordinates and velocities, the mass matrix, the spatial transformation matrix and other matrices required for the A_g matrix are obtained as given in equation (42). Mass matrix calculation is as follows.

$$M(q) = \sum_{i=1}^{r} (m_i J_{vi}^\top J_{vi} + J_{wi}^\top R_i I_i R_i^\top J_{wi}) \tag{52}$$

In the above equation, m is the mass of the i^{th} link, J_v and J_w are the Jacobian matrix linear and rotational parts. R_i indicates the rotational matrix of the i^{th} link, while I_i indicates the rotational inertia of the i^{th} link. r shows the number of links, including the floating base parameters. Since the q vector has a size of $(9x1)$, the size of the matrix $M(q)$ is $(9x9)$. In addition, the matrix U_1 consists of a size of $(6,6)$ unit matrix and a size of $(6x3)$ zero matrix, as, it is expressed as $U_1 = [1_{6x6}\ 0_{6x3}]$. The remaining two matrices can be calculated as follows.

$$\Phi = \begin{bmatrix} {}^G R_1 & 0 \\ 0 & {}^G R_1 \end{bmatrix} \tag{53}$$

$$
{}^{1}X_G^{\intercal} = \begin{bmatrix} {}^{G}R_1 & {}^{G}R_1 S({}^{1}p_G)^{\intercal} \\ 0 & {}^{G}R_1 \end{bmatrix} \tag{54}
$$

In the above equations, ${}^{1}R_G$ and ${}^{1}p_G$ are the rotation matrix and position vector. Frame 0, frame 1 and frame G are shown in figure 11 . 0 is the fixed frame of reference (FFR), 1 is the floating base coordinate system that is fixed under the foot, and the center of gravity G coordinate system is attached to the CoM.The ${}^{1}p_G$ is calculated by taking the difference between the absolute base frame and the center of mass of the one-legged robot. Then, the skew symmetric form of the ${}^{1}p_G$ is computed and its transpose is taken; see eq.(54). Finally, the centroidal momentum matrix, A_G, can be obtained by using all these matrices in the order given in the eq.(42).

Since floating base parameters are used in the centroidal momentum matrix, the first six terms belong to floating base parameters. For the one-legged robot, the joint velocity vector (q) is a $(9x1)$ vector, and the centroidal momentum vector (h_G) is a $(6x1)$ vector. Therefore, when the angular and linear x, y, and z axes are written separately, a $(6x9)$ centroidal momentum matrix (A_G) matrix is obtained. It is also possible to calculate joint velocities using the centroidal momentum and the centroidal momentum matrix. The joint velocity vector is obtained by multiplying both sides by the inverse of the centroidal momentum matrix in the equation (33). However, since the centroidal momentum matrix is a $(6x9)$ matrix and the joint velocity vector is $(9x1)$, it is necessary to use different methods to get the inverse. Yet one can resolve the Ag matrix to express submatrices float are related to floating bae and actual joint parameters.

$$
h_G = A_G\dot{q} = \begin{bmatrix} A_{G1} & | & A_{G2} & | & \dots & | & A_{G9} \end{bmatrix} \begin{bmatrix} \dot{q}_f \\ \dot{q}_r \end{bmatrix} \tag{55}
$$

In (55), A_G denotes the submatrix related to the parameter $\dot{q}_i$. As seen in (55), the resolution of A_G enables us to isolate $\dot{q}_r$ as finite following.

$$h_G - \left[\begin{array}{c|c|c|c} A_{G1} & A_{G2} & \dots & A_{G6} \end{array}\right] \dot{q}_f = \left[\begin{array}{c|c|c} A_{G7} & A_{G8} & A_{G9} \end{array}\right] \dot{q}_r \tag{56}$$

$$\dot{q}_r = \left[\begin{array}{c|c|c} A_{G7} & A_{G8} & A_{G9} \end{array}\right]^{-1} (h_G - \left[\begin{array}{c|c|c|c} A_{G1} & A_{G2} & \dots & A_{G6} \end{array}\right] \dot{q}_f) \tag{57}$$

In (57), one can use Moore-Penrose inversion and Arnoldi Iteration to compute globally minimum $\dot{q}_r$. At the same time, all joint parameters ($\dot{q}_f$, $\dot{q}_r$) were tried to be computed with the complete Ag matrix, and the system would assign the optimal values to the floating coordinates, but this could not match their actual value.

3.5.1.1 Verification of Centroidal Momentum and Centroidal Momentum Matrix of One-Legged Robot

To verify our computations, we conducted a similar study. For this reason, since they are uncontrolled, the joint velocity vector for the Jacobian was calculated using velocity kinematics in line with the reference velocities given. In this calculation, the following equation was used. For this reason, since they are uncontrolled, the joint velocity vector for the Jacobian was calculated using velocity kinematics in line with the reference velocities given. In this calculation, the following equation was used.

$$V = J(q_r)\dot{q}_r \tag{58}$$

$V = [V_{Xe}V_{Ze}W_Y]^T$ is the cartesian velocities, and $\dot{q}_r = [\dot{q}_1\dot{q}_2\dot{q}_3]^T$ represents the joint velocities. V_{Xe} represents the linear velocity in the x-axis, V_{Ze} represents the linear velocity in the z-axis and W_Y represents the body angular velocity in the y-axis. $J(q_r)$ is the Jacobian matrix in eq.(58). In the following equation, the joint velocity vector is obtained by multiplying both sides by the inverse of the Jacobian matrix

$$\dot{q}_r = J(q_r)^{-1}V \tag{59}$$

Table 3: Reference Velocities

V_{Xe}	$0.02(2pi0.5\cos(w(t-3)))$
V_{Ze}	$0.044(2pi0.5\cos(w(t-3)))$
W_Y	0

Thus, thanks to eq.(57) and eq.(59) obtained by velocity kinematics, it was possible to compare joint velocities and to control the calculation of A_G matrix thanks to the velocities found with the well-known Jacobian method.

In addition, another control method is to compare the momentum obtained from the centroidal momentum and the momenta by going from the jacobian velocities obtained to the momenta.

A simulation model was created in MSC ADAMS for the one-legged model to apply the mentioned calculations. The primary purpose of the simulation is to compare the momenta obtained from the Jacobian and centroidal momentum. In the simulation, the squat motion was chosen for the ADAMS model. The main reason for selecting this movement is that if the robot completes the movement in line with the references without tipping over, it can form the basis for other movements, and the developed algorithm can be used for other movements such as walking, jumping, etc.

Firstly, a 5^{th} degree polynomial function was used in ADAMS to the squatting position, and the initial position was obtained after assigning $45,-90,45$ degrees to the hip, knee and ankle angles. In periodic motion, the velocities obtained using eq.(59) are used. Since the squatting motion was chosen, the angular velocity of the torso on the y-axis was set to zero. The linear velocity trajectories in the x and z axis are given in the table below.

Since the squatting motion is desired, the reference velocity in the z-axis was chosen larger. Then, using equation (59), the joint velocities found via velocity kinematics and they are multiplied by the total mass to obtain the momenta; that is, the linear momenta in the x and z axes, P_x, P_z. Because of the robot's motion, the linear momentum P_y in the y-plane is zero. Also, the angular momenta L_x, L_z in the x and z planes are zero. As for L_y, the angular momentum in the y plane, the calculation can be made using the formula below:

$$L_y = I_{CoM}\dot{Q}_{ref} \tag{60}$$

$\dot{Q}_{ref}$ is computed as follows;

$$\dot{Q}_{ref} = (z_{CoM}\dot{x} - x_{CoM}\dot{z})/(z_{CoM}^2 + x_{CoM}^2) \tag{61}$$

I_{CoM} is the inertia of the entire body of the robot. Joint velocities and momentums obtained with the help of these two methods can be compared in the simulation environment, and their accuracy can be shown; see figure 12. Also, z_{CoM} and x_{CoM} are the position of the center of mass of the robot in the x and z axes. $\dot{x}$ and $\dot{z}$ are velocities obtained with the help of Jacobian with the joint velocities measured from the robot.

As shown in figure 12, after obtaining the desired position in the model and measuring the joint angles as the output value in ADAMS, these joint angles obtained were used in the calculation of angular momentum, Jacobian matrix and centroidal momentum matrix. As given in eq.(60), the angular momentum L_y is calculated. By using equation (58), the obtained joint angles were differentiated and multiplied with the Jacobian matrix to obtain the reference velocities V_{XCoM},V_{ZCoM}. These obtained velocities were multiplied by the total mass of the one-legged robot and linear momenta P_x,P_z were calculated. The centroidal momentum was calculated using the equation (33) and the derivatives of the joint angles were used to obtain the centroidal momentum vector.

As a result of all these calculations, both the momenta obtained with the centroidal momentum and matrix and the momenta obtained with the help of velocity kinematics can be compared. When looking at Figures 13, 14 and 15, the error of P_x, P_z and, L_y can be observed respectively approximately. When the error graphs of the momenta are compared, the error is zero for linear momenta, P_x and P_z. There is a max error of $0.06Nms$ in the angular moment L_y; the main reason for this difference is that it cannot be obtained like linear momenta since $\dot{Q}_{ref}$ is computation includes approximations.

After comparing the momentum values and making sure flow the centroidal momentum, the next step was taken. The purpose of this step is to calculate the inverse of the centroidal momentum matrix and arrive at the joint velocity vector given in eq.(57). At the same time, by taking the derivative of the joint angles obtained from the ADAMS model, the joint velocities can be found and compared. The scheme of the simulation study is given in Figure 19.

When looking at Figures 16, 17 and 18, joint velocity of V_1, joint velocity of V_2 and joint velocity of V_3 can be seen respectively. When the graphs are compared, it is seen that the joint velocities obtained from both methods are almost exactly the same. Looking at the error graphs and joint velocities graphs, it can be ensured that the centroidal momentum algorithm works correctly.

Thus, the confirmation of the centroidal momentum and centroidal momentum matrix is provided via velocity kinematics, which is a well-known method. The reason why it is made in an one-legged robot is that the number of joints is relatively less, making the confirmation study easier. At the same time, when the number of joints increases, the computation time increases. Thus, using a one-legged robot model save time and makes comparisons easier.

3.5.2 Atlas Humanoid Robot

Designed by Boston Dynamics, Atlas Humanoid Robot is an advanced humanoid robot with state-of-the-art equipment and a compact mobile hydraulic system that has the strength and balance capability to demonstrate human-like endurance. The Atlas Humanoid Robot used in this study has 29 active joints and 1 fixed joint; see Fig.21. Its total mass is 174.25 kg. The real version of the Atlas robot can be seen in Figure 20. To create its robot model, RaiSim is used for both simulation and algorithm calculations [44]. Values such as CoM position, link inertia and mass are provided in RaiSim. $q_i(i = 1,2,3,...,29)$ denote the relative angles between the links $(i-1)$ and the link (i). $\dot{q}_i(i = 1,2,3,...,29)$ represents joint velocities corresponding to each angle. Since a joint is fixed, it does not participate in the calculations of the joint angle and velocity vector. Figure 21, displays the simulation model and joint configurations. Fig.22 displays the coordinate systems used to represent floating base coordinates. The floating base was placed at the pelvis, a common point when considering tree streets. As in the one legged robot, when the floating base parameters are added to the joint angles and joint velocity vectors, the size of the joint coordinates becomes $(36x1)$. The joint position and velocity vectors are represented as below:

$$q^{\top} = \begin{bmatrix} q_{f1} & q_{f2} & q_{f3} & 1.0 & q_{f4} & q_{f5} & q_{f6} & q_{r1} & \cdots & q_{r29} \end{bmatrix} (1x36) \tag{62}$$

$$\dot{q}^{\top} = \begin{bmatrix} \dot{q}_{f1} & \cdots & \dot{q}_{f6} & \dot{q}_{r1} & \cdots & \dot{q}_{r29} \end{bmatrix} (1x35) \tag{63}$$

To compute A_G matrix, firstly the mass matrix(35x35) is obtained. Afterwards, the processes continue in the same way and the size of the other matrices is obtained in accordance with the vectors q and $\dot{q}$. In addition, the matrix U_1 consists of the unit matrix size $(6x6)$ and the zero matrix with a size $(6x29)$. In this case, it is written as $U_1 = [1_{6x6}\ 0_{6x29}]$. For the remaining two matrices, the rotation matrix and position vectors are calculated, as in the case of one-legged robot. The position vector is computed by taking the difference between the linear parameters of the floating base and the center of mass of the atlas humanoid robot. Then the skew-symmetric form of their position vector is computed and transpose. The final form of the obtained rotation and position vector is replaced in the eq.(32). Thus, all the matrices necessary to obtain the centroidal momentum matrix are calculated. Finally, the centroidal momentum matrix can be obtained by multiplying all the matrices in the order given in the eq.(42). The size of our resulting A_G matrix is (6x35). Since the matrix is extensive and covers all body joints, it is necessary to divide it into submatrices to control robot links more easily. Accordingly, velocity and position vectors can be split-fed in a similar fashion.

$$A_G = \begin{bmatrix} A_{G_f} & A_{G_T} & A_{G_{LA}} & A_{G_{RA}} & A_{G_{LL}} & A_{G_{RL}} \end{bmatrix} \tag{64}$$

$$q^\top = \begin{bmatrix} q_f & q_T & q_{LA} & q_{RA} & q_{LL} & q_{RL} \end{bmatrix} \tag{65}$$

$$\dot{q}^\top = \begin{bmatrix} \dot{q}_f & \dot{q}_T & \dot{q}_{LA} & \dot{q}_{RA} & \dot{q}_{LL} & \dot{q}_{RL} \end{bmatrix} \tag{66}$$

In (64),A_{G_f}, A_{G_T}, $A_{G_{LA}}$, $A_{G_{RA}}$,$A_{G_{LL}}$ and $A_{G_{RL}}$ respectively denote the seepiers related to floating base, torso, left arm, right arm, left leg, right leg joint parameters. To enable whole-body control, figure 9 can be used see section the *3.4 Centroidal Momentum Observer for a Whole Body Humanoid*. In order to design the Centroidal Momentum Observer, firstly, the lower body trajectory can be given. To this end the robot comes to the squat position first, and then the fifth-degree polynomial function can be used for the continuity of the squat movement. Since centroidal momentum regulates the angular momentum of the system, it may be more appropriate to base the angular part of the calculated momentum. A low-pass filter provides the calculated and reference momentum as feedback to prevent phase difference. Since the feedback output comes as momentum, velocity and position data are needed to give it to the system. For this reason, the following equation is used.

$$-H_{dis} = A_{G_f}\dot{q}_f + A_{G_T}\dot{q}_T + A_{G_{LA}}\dot{q}_{LA} + A_{G_{RA}}\dot{q}_{RA} + A_{G_{LL}}\dot{q}_{LL} + A_{G_{RL}}\dot{q}_{RL} \tag{67}$$

When using eq.(67), one must note that $\dot{q}_{LL}$ and $\dot{q}_{RL}$ were predetermined. Moreover, $\dot{q}_f$ can be measured. Thus, (eq.67) takes the following from;

$$A_{G_T}\dot{q}_T + A_{G_{LA}}\dot{q}_{LA} + A_{G_{RA}}\dot{q}_{RA} = H_{dis} + A_{G_f}\dot{q}_f + A_{G_{LL}}\dot{q}_{LL} + A_{G_{RL}}\dot{q}_{RL} \tag{68}$$

The following equation can be expressed in the matrix form;

$$\begin{bmatrix} A_{G_T} & | & A_{G_{LA}} & | & A_{G_{RA}} \end{bmatrix} \begin{bmatrix} \dot{q}_T \\ \dot{q}_{LA} \\ \dot{q}_{RA} \end{bmatrix} = H_{dis} + A_{G_f}\dot{q}_f + A_{G_{LL}}\dot{q}_{LL} + A_{G_{RL}}\dot{q}_{RL} \tag{69}$$

With the help of (69), the velocity vector can be obtained by using pseudo inverse;

$$\begin{bmatrix} \dot{q}_T \\ \dot{q}_{LA} \\ \dot{q}_{RA} \end{bmatrix} = \begin{bmatrix} A_{G_T} & | & A_{G_{LA}} & | & A_{G_{RA}} \end{bmatrix}^{+} (H_{dis} + A_{G_f}\dot{q}_f + A_{G_{LL}}\dot{q}_{LL} + A_{G_{RL}}\dot{q}_{RL}) \tag{70}$$

The most important thing to note in eq.(70) is that its size is (6,17) for the Atlas Humanoid Robot, as the A_G matrix consists of the torso, left arm and right arm. In this case, the pseudo inverse can be used. Afterwards, joint position vector can be obtained by passing them through the high pass filter and integrating them.

$$\begin{bmatrix} q_T \\ q_{LA} \\ q_{RA} \end{bmatrix} = \int (\begin{bmatrix} A_{G_T} & | & A_{G_{LA}} & | & A_{G_{RA}} \end{bmatrix}^{+} (H_{dis} + A_{G_f}\dot{q}_f + A_{G_{LL}}\dot{q}_{LL} + A_{G_{RL}}\dot{q}_{RL})) \tag{71}$$

Thus, the upper body position vector is obtained to be used in simulations with the help of eq.(71).

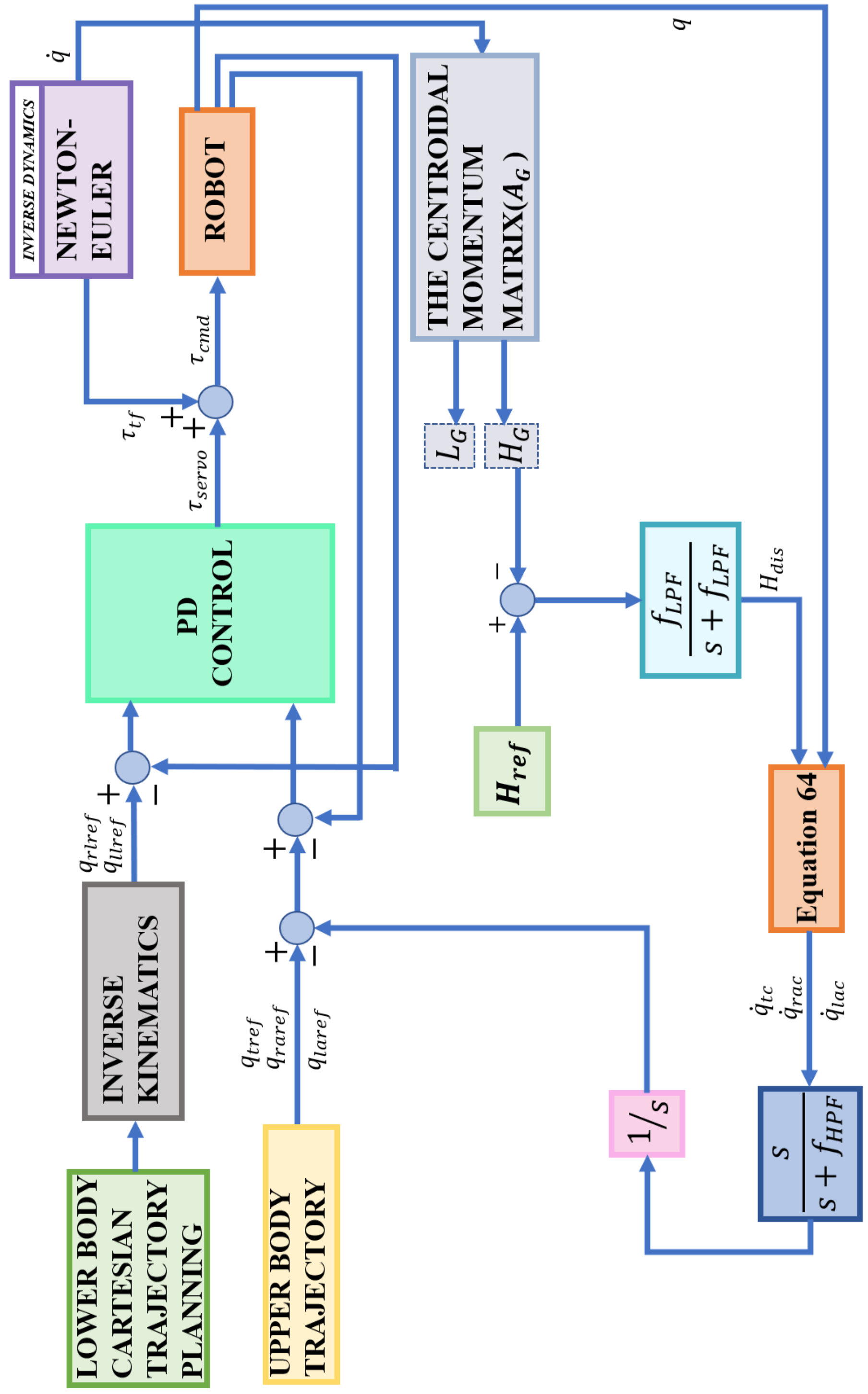


Figure 9: Controller scheme of the proposed Centroidal Momentum Observer for a Whole Body Humanoid. The parameters with hat are obtained through estimation. The parameter f_{LPF} is low pass filter gain and f_{HPF} is high pass filter gain empirically tuned.The momentum reference is determined by trajectory planning.

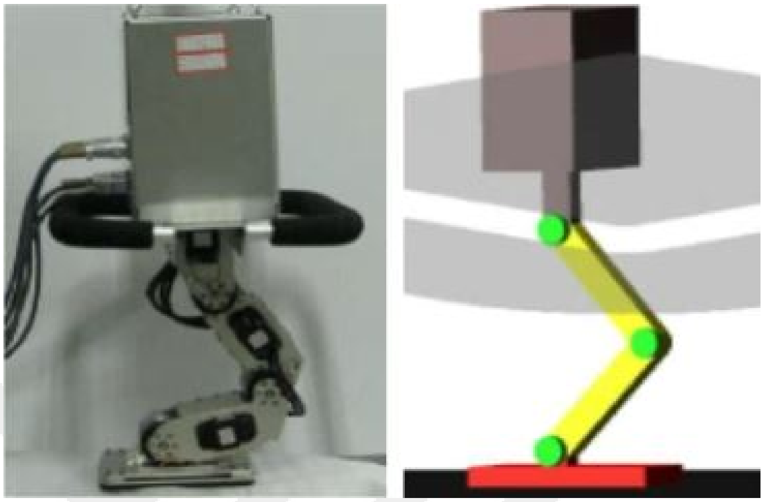


Figure 10: The one-legged robot [1], and its simulation.

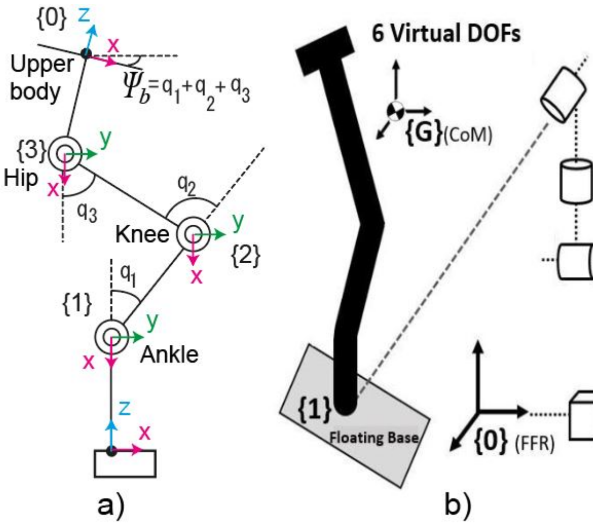


Figure 11: Floating base parameters of our one legged robot model. Adapted from [2].

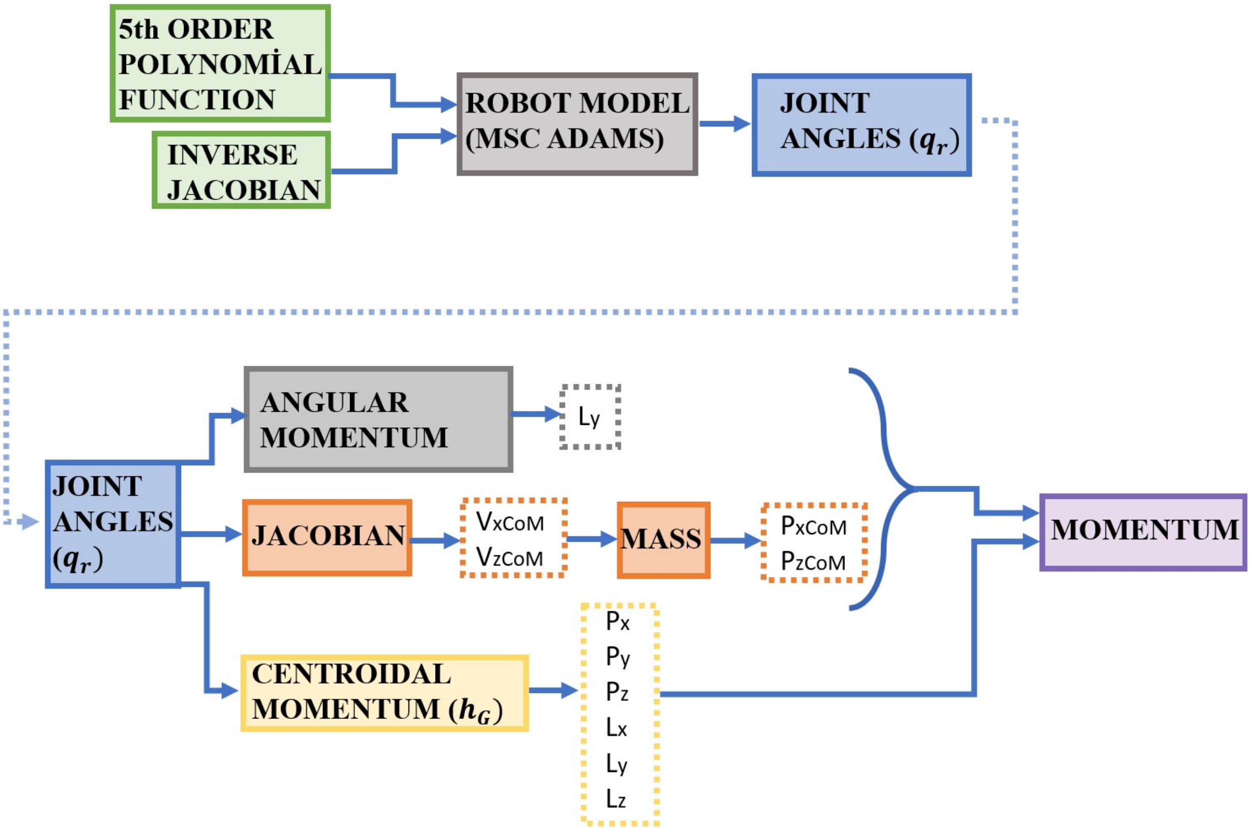


Figure 12: Momentum calculation scheme after calculating joint angles with the 5th order polynomial function and the inverse of the Jacobian matrix

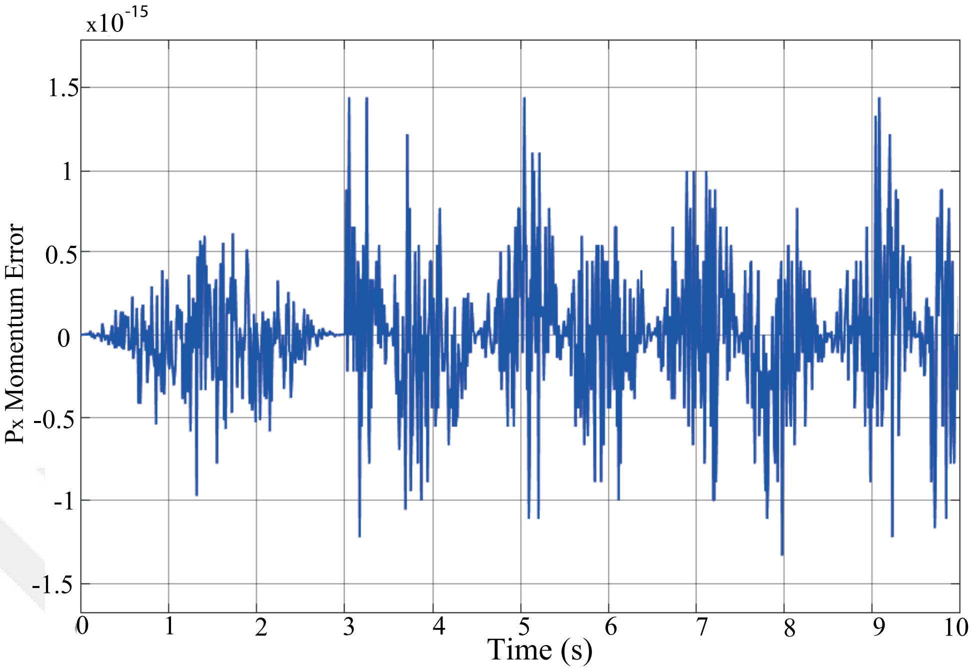


Figure 13: P_x Momentum Error *(While the robot comes to the squat position in the range of* $0-3$ *sn, it ensures the continuity of the movement in the range of* $3-10$ *sn.)*

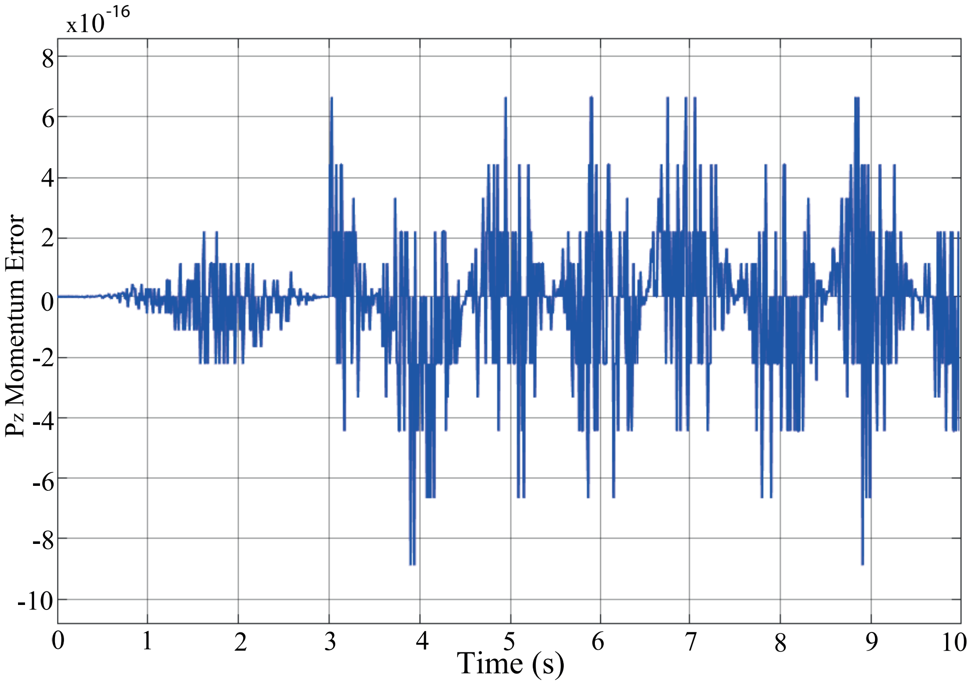


Figure 14: P_z Momentum Error *(While the robot comes to the squat position in the range of* $0-3$ *sn, it ensures the continuity of the movement in the range of* $3-10$ *sn.)*

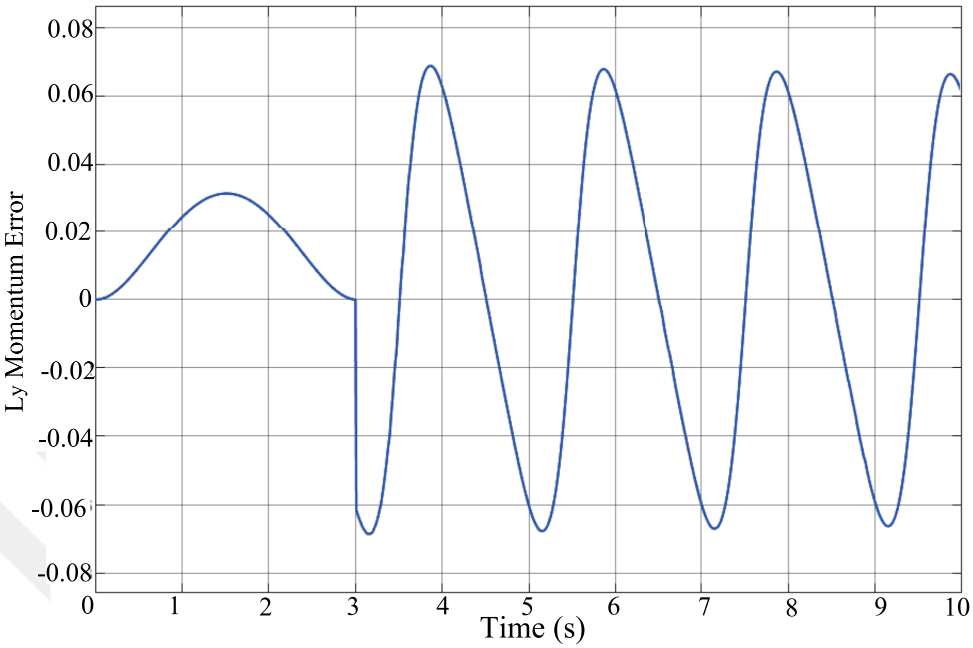


Figure 15: L_y Momentum Error *(While the robot comes to the squat position in the range of* $0-3$ *sn, it ensures the continuity of the movement in the range of* $3-10$ *sn.)*

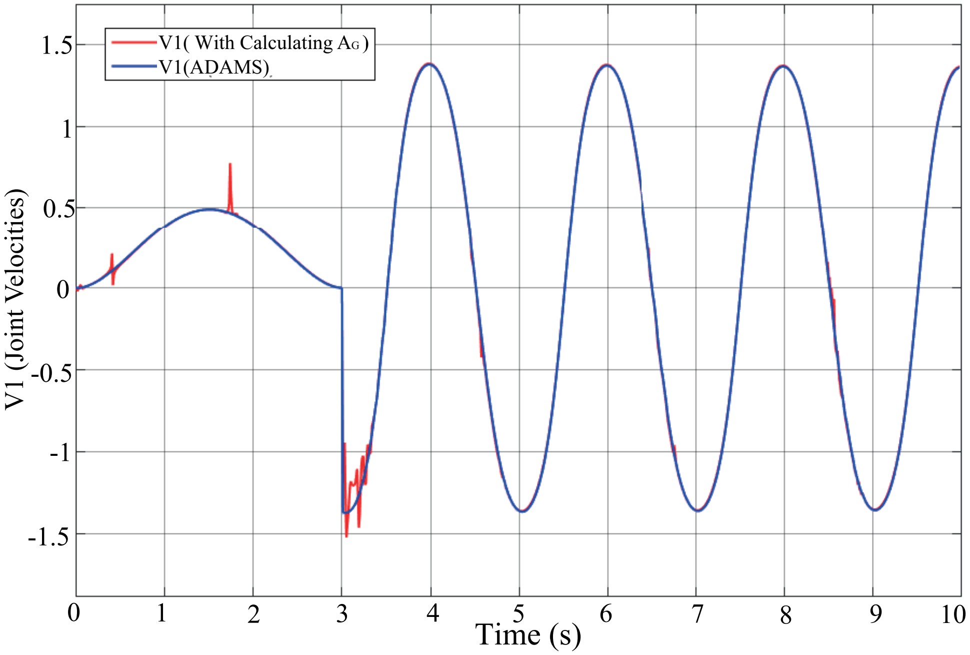


Figure 16: V_1 Hip Joint Velocity *(While the robot comes to the squat position in the range of* $0-3$ *sn, it ensures the continuity of the movement in the range of* $3-10$ *sn.)*

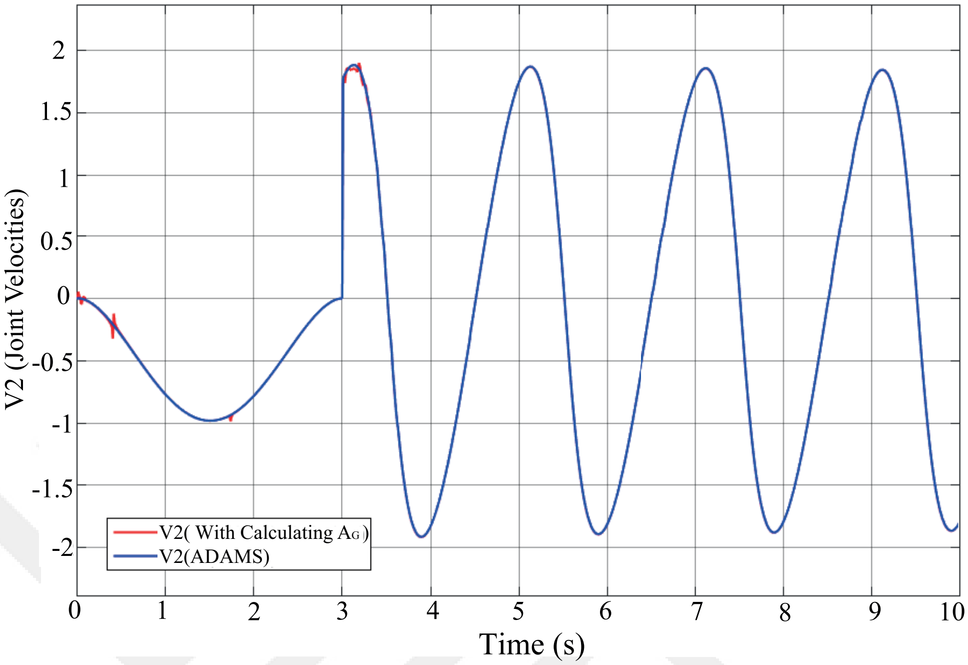


Figure 17: V_2 Knee Joint Velocity *(While the robot comes to the squat position in the range of* $0-3$ *sn, it ensures the continuity of the movement in the range of* $3-10$ *sn.)*

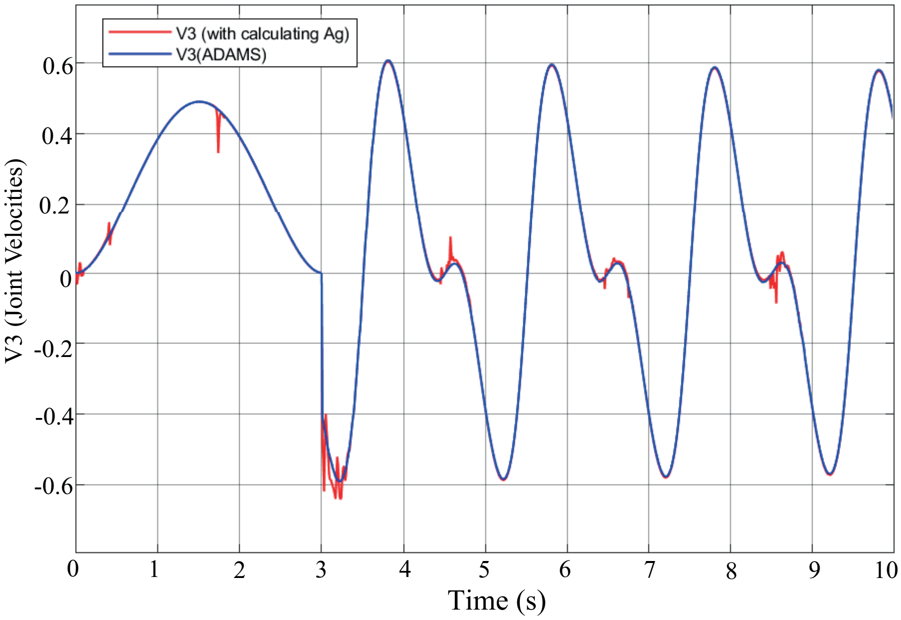


Figure 18: V_3 Ankle Joint Velocity *(While the robot comes to the squat position in the range of* $0-3$ *sn, it ensures the continuity of the movement in the range of* $3-10$ *sn.)*

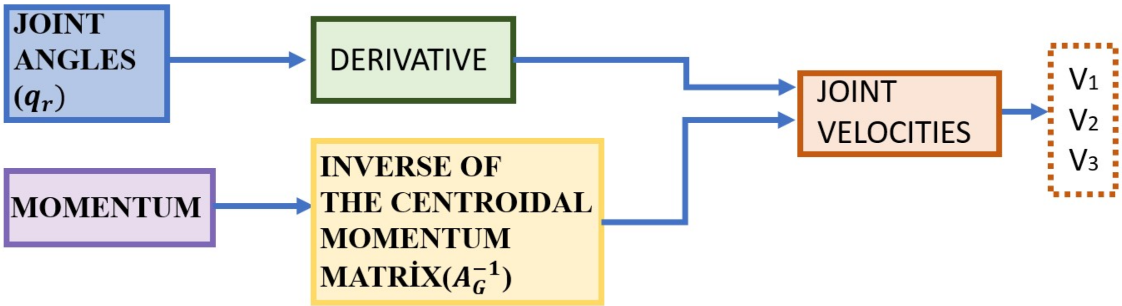


Figure 19: Calculation of joint velocities with the help of joint angles and centroidal momentum

Figure 20: The actual Atlas Humanoid Robot from Boston Dynamics.

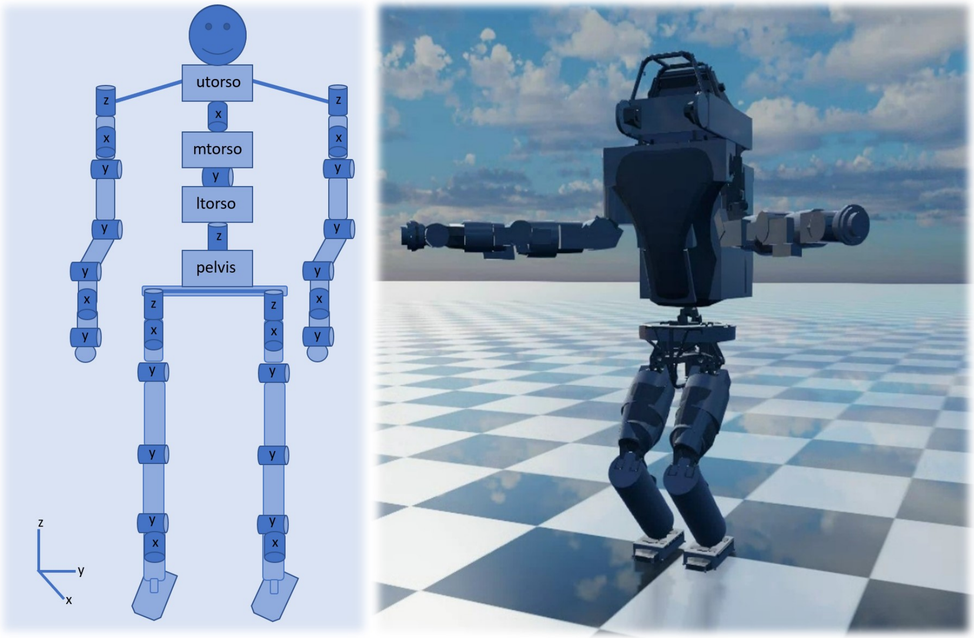


Figure 21: Simulation Model and Joint Configurations of Atlas Humanoid Robot(In the joint configuration of the robot, The torso consists of three parts: upper torso, middle torso and lower torso.)

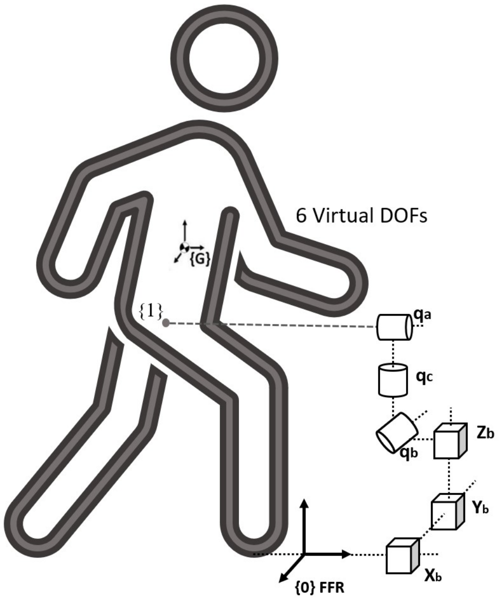


Figure 22: Floating base parameters of Atlas robot adapted from [2]

CHAPTER IV

SIMULATION EXPERIMENTS

In order to implement the proposed controller, namely, centroidal momentum observer, a series of simulations were performed using 3D dynamic simulators MSC ADAMS [45] and RaiSim [44]. Simulations were executed using a PC that possesses an Intel (R) Xeon (R) W-2155 processor (3.3 GHz) with 32.0 GB RAM, running on a 64-bit Windows operating system.

4.1 Simulation Scenarios

This section considers six different scenarios with different types of disturbances and situations.

4.1.1 Scenario #1: Unknown Load- One-Legged Robot

To check the robustness of a system, it must keep its balance despite parametric changes. For this reason, a 5kg of extra load was attached to the one-legged robot body, and the controllers were unaware of this load in this scenario. This load is not included in any dynamic and static calculations. In other words, this extra load is not considered in the calculations of the robot dynamics and centroidal momentum matrix. In this scenario, it is desired to test the disturbance rejection capacity of the controllers when there are parametric uncertainties.

4.1.2 Scenario #2: External Forces- One-Legged Robot

In this scenario, the question was what would the robot's reaction be when an indefinite amount of external force was applied and whether the controllers could maintain their robustness in this case. External forces were applied to the robot in the x-axis and z-axis

Table 4: The gain values of the controllers

		Scenario 1	**Scenario 2**	**Scenario 3**
	Parameter	**Value**		
CMC	K_h	1	1	1
CMD	K_h	0.3	0,3	0.3
	f	30	30	40
ZMP	k_p	0.6	0.6	0.6
	k_i	4.3	4.3	4.3

directions. When $t = 2$s, an external repulsive force of 30 N along the x direction and 100 N along the z direction were applied to the robot's body. In line with these applied forces, the controllers' responses were examined.

4.1.3 Scenario #3: External Momentum- One-Legged Robot

What is desired in this scenario is to see how long the controllers will maintain the system's stability when the one-legged robot is exposed to external momentum. An external angular momentum in the form of a 0.5 Ns amplitude step function along the y-direction and an external linear momentum in the x direction as a 0.2 Ns amplitude step function were applied over it when $t = 2$s.

For each scenario regarding the one-legged robot, three controllers were implemented. Their respective performances compared, i.e. Centroidal Momentum Control [2] (CMC, see Fig.5), the proposed controller Centroidal Momentum Observer (CMD, see Fig. Fig.6) and ZMP Feedback Control [41]; see Fig. 8. The ZMP feedback controller includes a feedback linearization loop with a PI controller. The gain values for the controllers are tabulated in Table 4.

4.1.4 Scenario # 4: Balancing while Perturbed from Lower Torso- Atlas Humanoid Robot

In this scenario, the Atlas humanoid robot was initialized while standing with bent knees. Then, by keeping it constant, an external force of 200N from the lower torso is given in the x direction, and the movement in the presence and absence of the controller in the

Table 5: Reference Cartesian Velocities and Filter Gains

V_{Xe}	$0.77-0.05\cos(2pi0.35t)m/s$
V_{Ze}	$0m/s$
W_Y	$0m/s$
f_{LPF}	$0.5Hz$
f_{HPF}	$40Hz$

direction of the external force is observed. This simulation scenario was also repeated with no perturbation.

4.1.5 Scenario # 5: Balancing while Perturbed from Shoulder- Atlas Humanoid Robot

In this scenario, similar to Scenario # 4, the Atlas humanoid robot was initialized while standing with bent knees. Then, keeping the CoM position constant, an external force of 100N in the x direction and 10N in the z direction is applied from the shoulder. In other words, movement is observed in the controller's presence and absence with the arm's external force. This simulation scenario was also repeated with no perturbation.

4.1.6 Scenario # 6: Squatting with Perturbation from Lower Torso- Atlas Humanoid Robot

In this scenario, the Atlas humanoid performed squatting motion; see Table 5 for cartesian velocity references and filter gains. While performing the squatting motion, the robot was perturbed by an external force of 100N from the lower torso and with this perturbation, the movement was observed in the presence and absence of the controller. This simulation scenario was also repeated with no perturbation.

The controller mentioned in *Chapter 3.4. Centroidal Momentum Observer for a Whole Body Humanoid* and shown in figure 9, is applied to the Atlas humanoid robot in all three cases. The results in the presence and absence of this controller were compared with and without disturbance. At the same time, the low pass gain and high pass gain values used in the controller are given in Table 5. In addition, the PD gains can be seen in Table 6 .

Table 6: PD gains

	Kp	Kd
Torso	5000	360
Arms	4000	355
Legs	3000	350

4.2 Results

The results are displayed in Figures 23-31, in which black dashed lines indicate reference variation, while solid blue, green, and red lines show response variations for the controllers CMD, CMC, and ZMP, respectively. In continuation, the results are displayed in Figures 32, 33, 39, 45, 46, where the blue lines show the response variations of the CMD controller. In contrast, the red lines show the response variations without the CMD controller. Also, the other results are displayed in Figures 34, 35, 36, 40, 41, 42, 47, 48, 49, where blue lines show the robot's response variations in perturbation. In contrast, the red lines show the robot's response variations without perturbation.

4.2.1 The Result of Scenario#1: Unknown Load- One-Legged Robot

In Figures 23, 24 and 25, the the x-axis CoM position (X_{com}), the z-axis CoM position (Z_{com}), and the x-axis ZMP position (X_{zmp}) are displayed for this scenario, respectively. As may be observed, the ZMP controller could not stabilize the robot in this scenario, the robot tips over. In particular, the x-axis ZMP reached to the rear end of the support polygon. On the contrary, both CMD and CMC could stabilize the robot. Among them, CMD outperformed CMC. The RMS (Root Mean Squares) values of CoM tracking error are 0.0267 cm and 0.193 cm for the x-axis, and 0.0922 cm and 0.473 cm for the z-axis, concerning CMD and CMC, respectively. Likewise, the RMS value of X_{zmp} variation in the case of CMD is 3.054 cm. It is measured as 4.765 cm, for the case of CMC. These results suggest that CMD is able to handle parametric uncertainties.

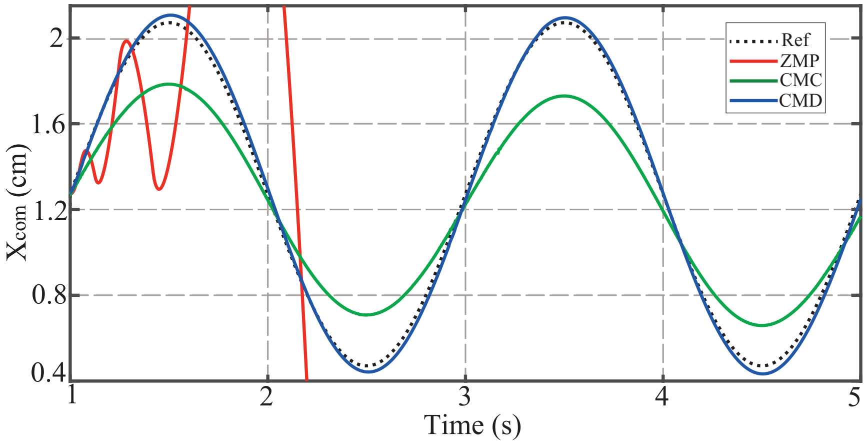


Figure 23: One legged robot of the center of mass(CoM) on the x-axis

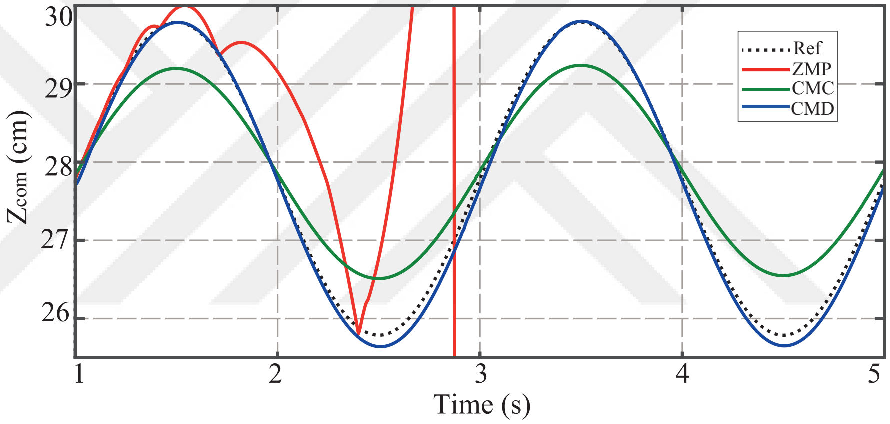


Figure 24: One legged robot of the center of mass(CoM) on the z-axis

4.2.2 The Result of Scenario#2: External Forces- One-Legged Robot

In Figures 26, 27 and 28, the the x-axis CoM position (X_{com}), the z-axis CoM position (Z_{com}), and the x-axis ZMP position (X_{zmp}) are displayed for this scenario, respectively. In this case, all three controllers achieved dynamic balancing. Similar to Scenario #1, CMD outperformed CMC and ZMP. CoM tracking errors are 0.02736, 0.187, and 5.137 for x-axis and 0.0562, 0.468, and 0.703 for z-axis, concerning CMD, CMC, and ZMP, respectively. The RMS values of concerning the actual X_{zmp} are 7.324, 7.717, and 3.756 for CMD, CMC, and ZMP, respectively. Judging from these results, CMD appears to provide more favorable balancing performance when the robot is subject to external forces, despite the variation in X_{zmp}.

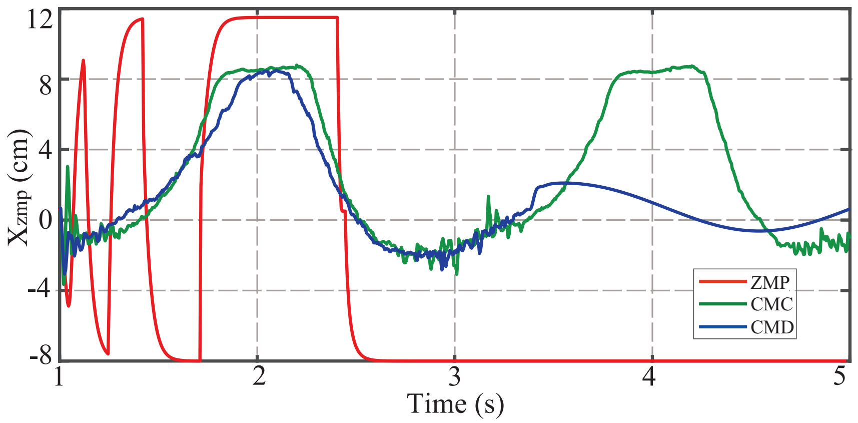


Figure 25: One legged robot of ZMP position on the x-axis

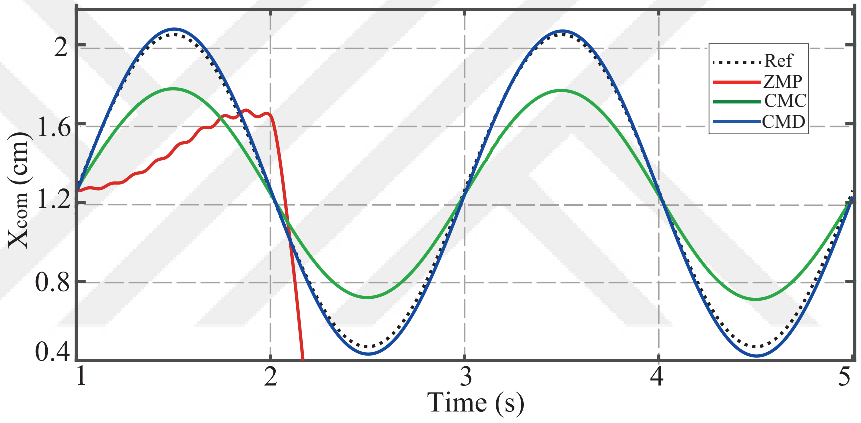


Figure 26: One legged robot of the center of mass(CoM) on the x-axis

4.2.3 The Result of Scenario#3: External Momentum- One-Legged Robot

In Figures 29, 30 and 31, the the x-axis CoM position (X_{com}), the z-axis CoM position (Z_{com}), and the x-axis ZMP position (X_{zmp}) are displayed for this scenario, respectively. In this scenario, the controllers CMC and ZMP failed to stabilize the robot and is tip over, whereas CMD was able to maintain dynamic balance. In order to counteract the external angular momentum applied to the CoM, the robot automatically bent his torso in the opposite direction and maintain the balancing. The RMS values of X_{com} and Z_{com} error are 0.0232 and 0.0564 when implementing CMD. In addition, the RMS of X_{zmp} was 3.957 for this controller.

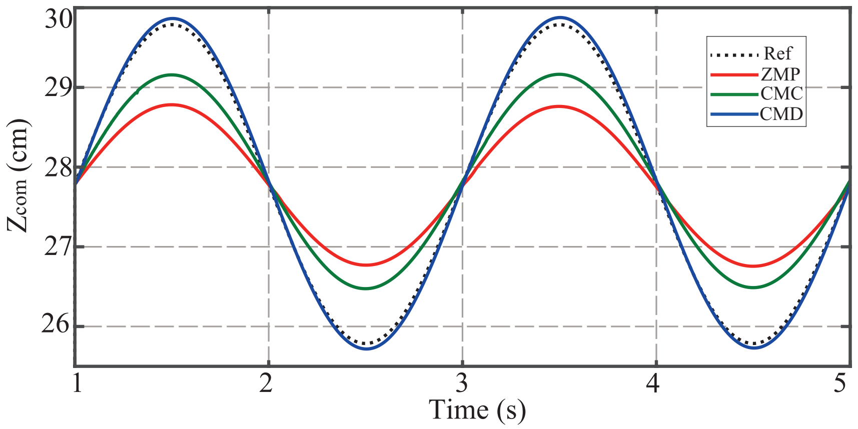


Figure 27: One legged robot of the center of mass(CoM) on the z-axis

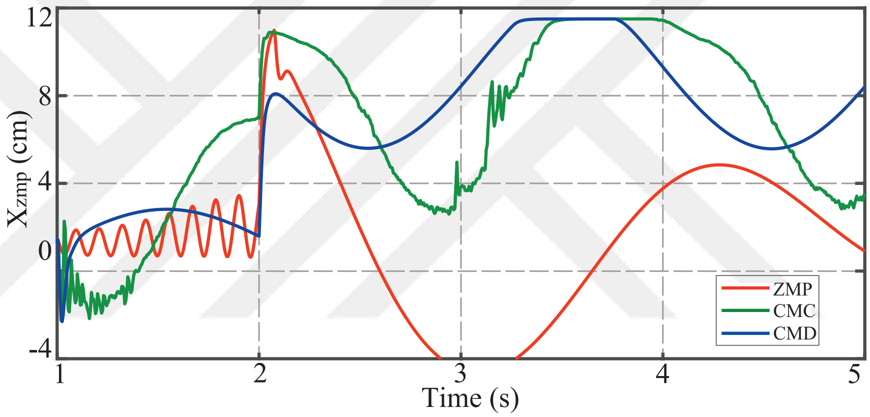


Figure 28: One legged robot of ZMP position on the x-axis

4.2.4 The Result of Scenario#4: Balancing while Perturbed from Lower Torso- Atlas Humanoid Robot

In Figures, the angular momentum changes of 32, 33, 34 are displayed for this scenario. Joint angle changes in the figures 35, 36 and joint velocity changes in the figures 37, 38 are displayed for this scenario, respectively.In this scenario, when there was the perturbation, in the absence of the CMD controller, the robot could not stabilize and overturned, while the CMD was able to maintain the dynamic balance. Also, when there is the disturbance, the body angle on the y-axis cannot be controlled because the robot overturns without CMD, and the angle increases to large values with its body weight. A larger angle is not a better result because the robot falls, the PD cannot exert enough torque due to its weight as it

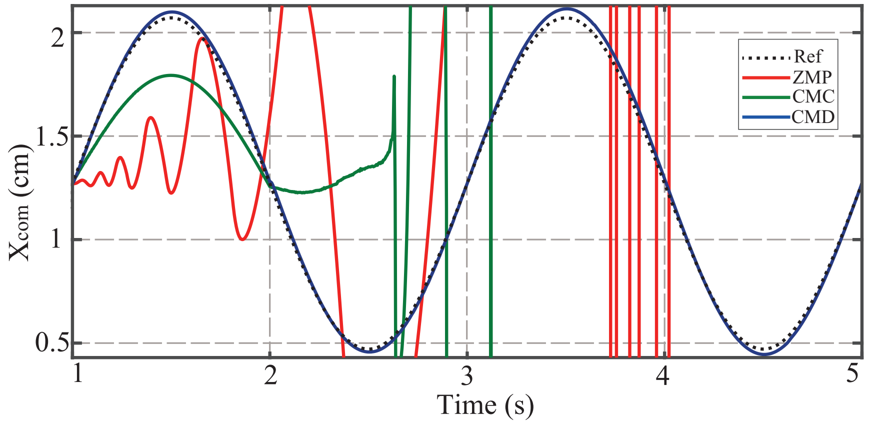


Figure 29: One legged robot of the center of mass(CoM) on the x-axis

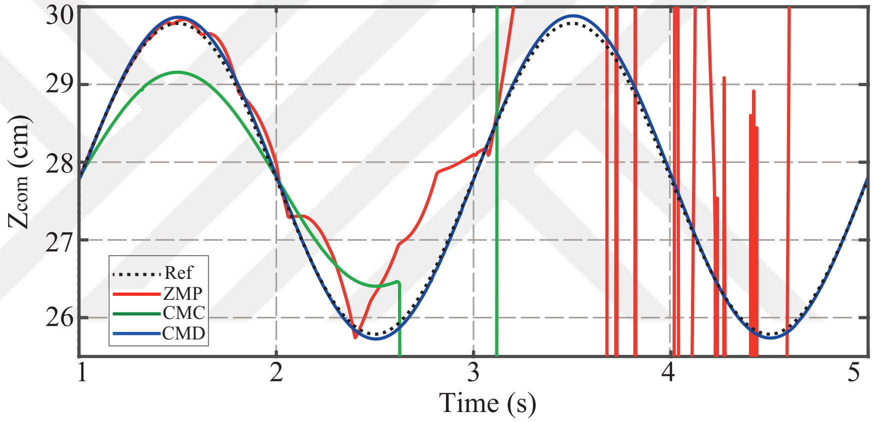


Figure 30: One legged robot of the center of mass(CoM) on the z-axis

falls, and the joint rotates randomly. In the absence of disturbance, it was observed that the robot stabilized faster in the presence of CMD when the initial angles were adjusted in the first seconds. Although the movement continued in the same position, there was a certain oscillation in the absence of CMD due to the weight of the arms. RMS values were 1.9 and 3.693 with and without CMD, respectively, when there is without perturbation. When these values are considered, a decrease of %48.5460 is observed when a CMD controller is found compared to the other situation. For this reason, the CMD controller is vital to ensure robustness and stabilization.

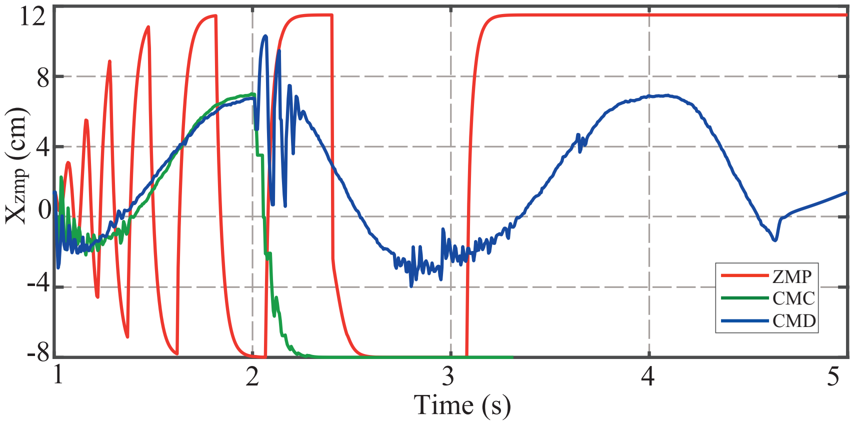


Figure 31: One legged robot of ZMP position on the x-axis

4.2.5 The Result of Scenario#5: Balancing while Perturbed from Shoulder- Atlas Humanoid Robot

In Figures, the angular momentum changes of 39, 33, 40 are displayed for this scenario. Joint angle changes in the figures 41, 42 and joint velocity changes in the figures 44, 43 are displayed for this scenario, respectively. In this scenario, while applying the distortion, the robot stabilized in the presence of CMD, but was unable to balance and fell in the absence of CMD. The CMD controller changes the body-x angle from about 1 to 2.5 degrees, the body-y angle from about 2.5to 1 degrees, and the body-z angle from about 0 to 6 degrees. Thus, it aims to maintain stabilization. It aims to maintain the stabilization of the robot in the presence of CMD and to ensure the continuity of motion when any disturbance is applied. Considering the CMD-free situation, when distortion is applied, the robot cannot maintain its stabilization as it is in the other state, i.e. the non-distortion state. Also, when distortion is applied, the CMD controller swings the shoulder-x angle between 0.3 and 0.7 and the shoulder-y angle between 0.15 and 0.45, causing distortion weakening. If there is no CMD controller, the robot will crash in 3 seconds and will not be able to maintain its stability. Thanks to the CMD controller, the robot creates a longer-term resistance mechanism against the reaction it receives from the shoulder. For the scenario in which external force is applied from the arm, the situation where there is no disturbance is the same as that from the lower torso. For these reasons, the CMD controller played an essential role

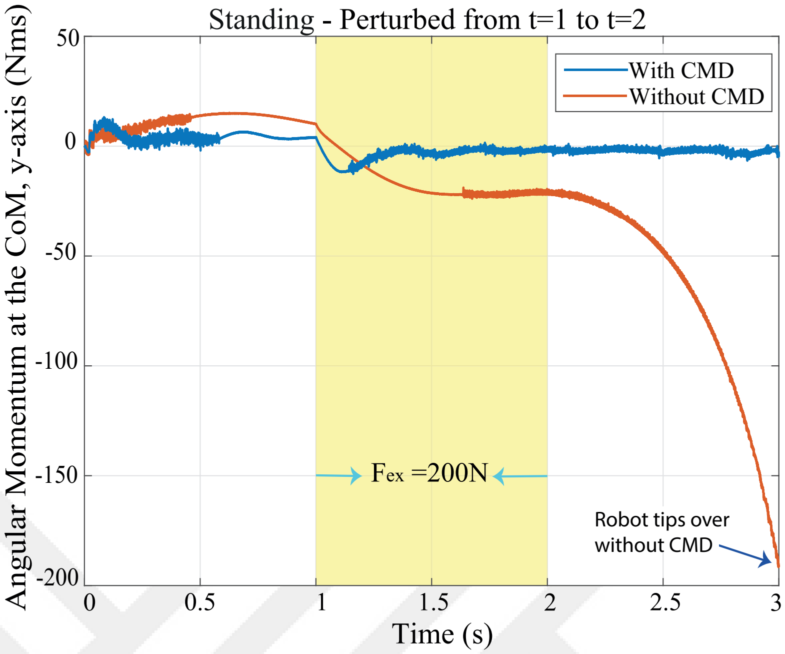


Figure 32: While external force is applied to the lower torso between t=1 and t=2 seconds, the angular momentum at the CoM, y-axis (Nms); Scenario #4.

in providing robustness and stabilization in this scenario.

4.2.6 The Result of Scenario#6: Squatting with Perturbation from Lower Torso-Atlas Humanoid Robot

In Figures, the angular momentum changes of 45, 46 and 47 are displayed for this scenario. Joint angle changes in the figures 48, 49 and joint velocity changes in the figures 50, 51 are displayed for this scenario, respectively. In this scenario, when there was the perturbation, in the absence of the CMD controller the robot could not stabilize and overturned, while the CMD was able to maintain the dynamic balance. CMD controller aimed to maintain balance by changing the torso angle on the y-axis between 1.5 and 3.5 degrees. Without CMD, the torso angle in the y-axis cannot be controlled, and the angle increases to large values with its own body weight. A larger angle is not a better result because the robot tips over; the system cannot provide enough torque due to its weight as it tips over, and the joint rotates randomly. For this scenario, in the absence of perturbation, there is a certain oscillation in the absence of CMD due to the weight of the arms. CMD tries to regulate

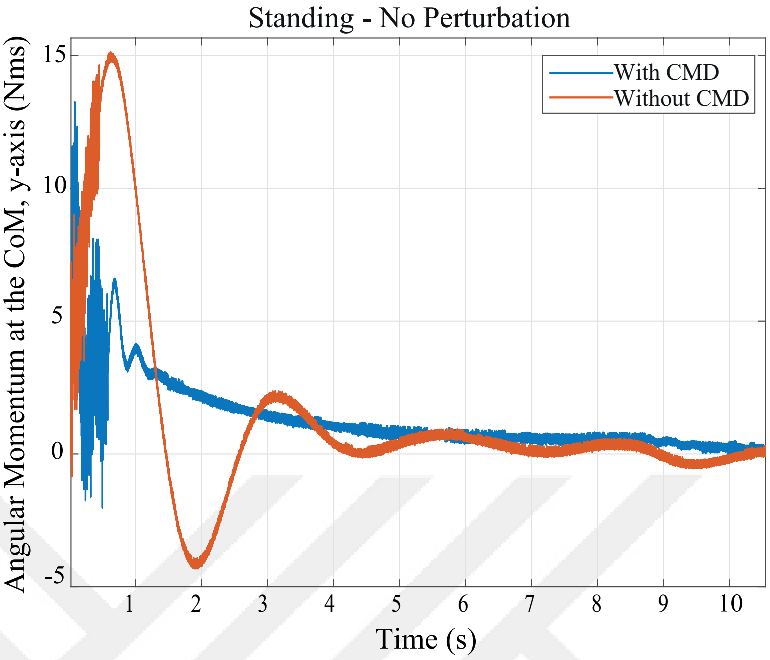


Figure 33: While external force is not applied to the lower torso, the angular momentum at the CoM, y-axis (Nms); Scenario #4.

the angular momentum so that it aims to keep the motion of the system constant in the absence of distortion. In this case, RMS values were 2.2681 and 7.7350 with and without CMD, respectively, when there is without perturbation. Considering these values, when a CMD controller is found compared to the other situation, a decrease in the RMS value of %70.6780 is observed. For this reason, the CMD controller is vital to ensure robustness and stabilization.

4.3 Discussion

Simulations were conducted with the proposed controller CMD in different scenarios for two different robots. The proposed centroidal momentum-based control algorithm provides robust dynamic motion control and eliminates parameter uncertainty. To test these features, the controller is not aware of disturbances. To this end, a disturbance observer (DOB), frequently used in motion control literature, ensures robustness despite external interference. In line with all this information, a one-legged robot model was simulated using

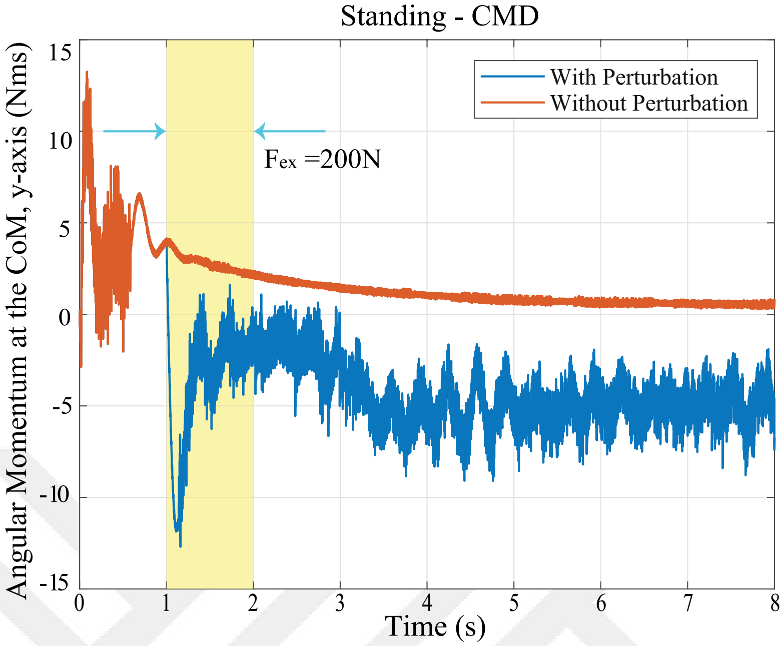


Figure 34: The angular momentum at the CoM in the presence of CMD controller, y-axis (Nms); Scenario #4.

MSC ADAMS. Then a centroidal momentum feedback controller and a Zero Moment Point (ZMP)-based control algorithm were developed [41]. The performances of the controllers were tested in the simulation environment following three different scenarios; i) unknown load, ii) external forces, and iii) external momentum distortions. The controllers were evaluated by comparing the center of mass (CoM) tracking along the x and z axes and the ZMP positions on the x-axis. Then, with the Atlas humanoid robot simulation, three different scenarios, fixed standing, lower torso and shoulder strength, and squat and strength from the lower torso, were tried. In these scenarios, the CMD controller, which was successful in the one-legged robot, was tried. The simulation results satisfactorily validated the proposed centroidal momentum observer as it performed well in all tested conditions.

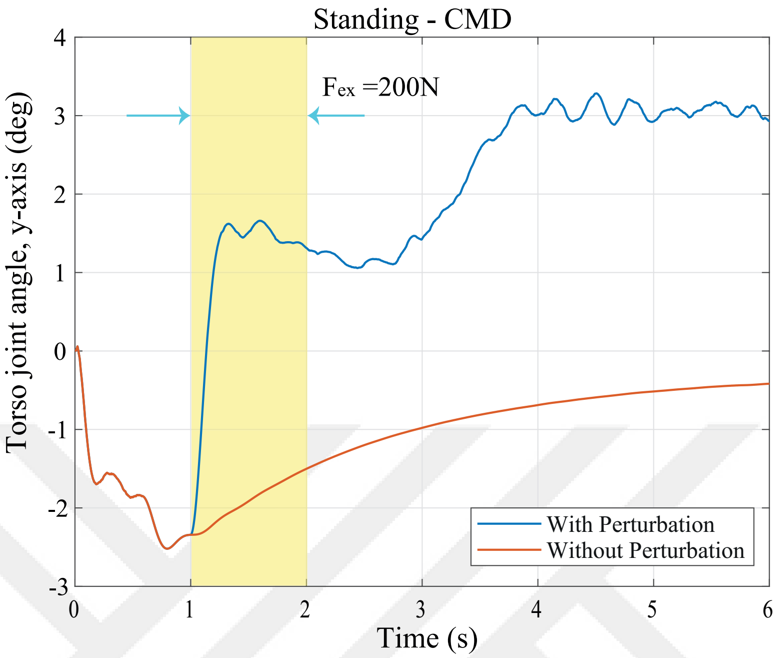


Figure 35: Torso joint angle in the presence of CMD controller, y-axis (deg); Scenario #4.

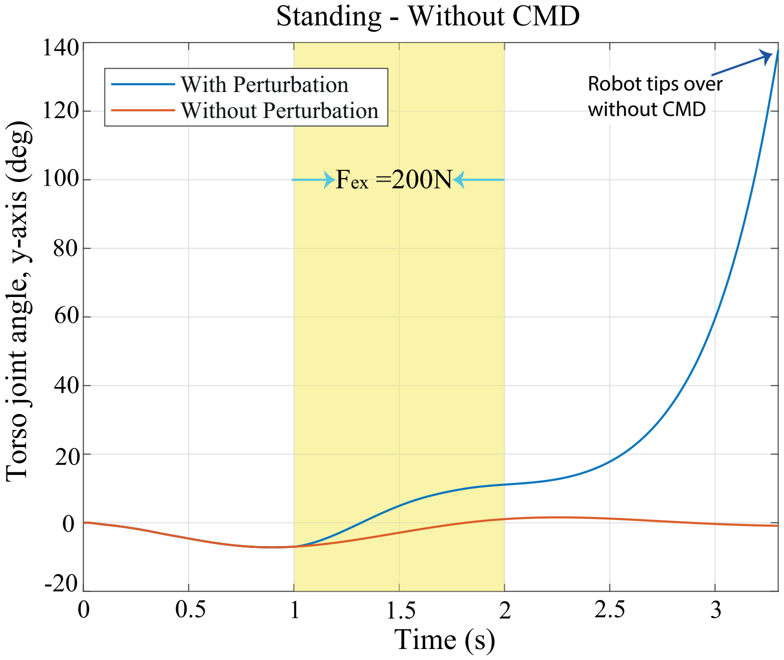


Figure 36: Torso joint angle in the absence of CMD controller, y-axis (deg); Scenario #4.

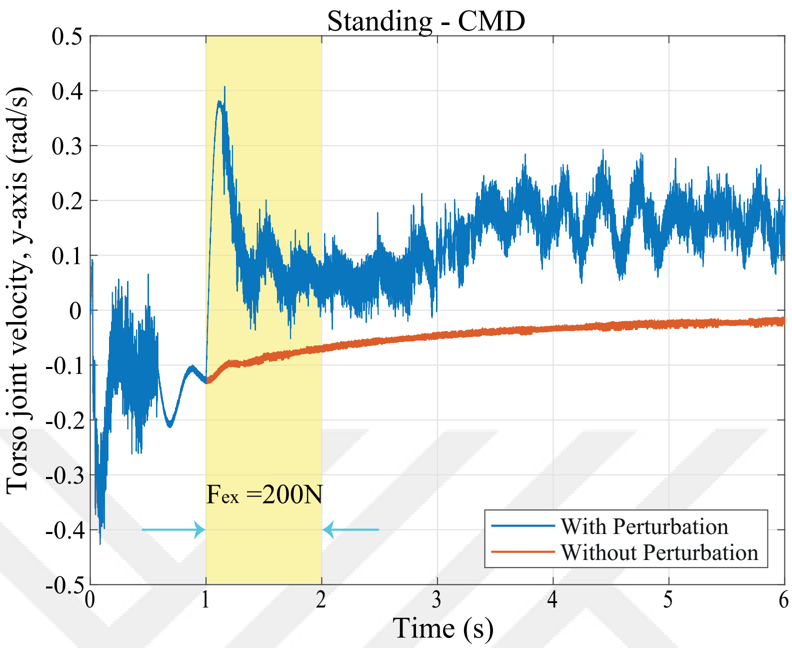


Figure 37: Torso joint velocity in the presence of CMD controller, y-axis (rad/s); Scenario #4.

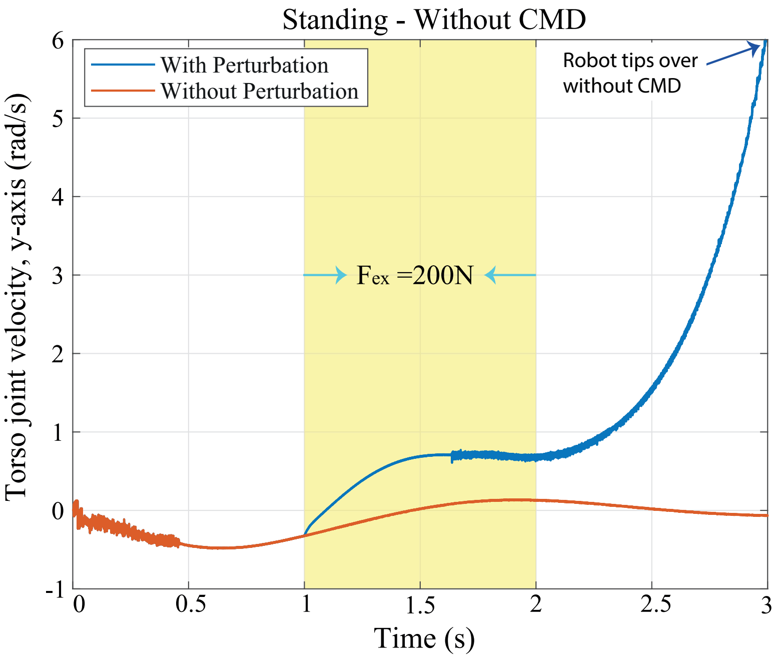


Figure 38: Torso joint velocity in the absence of CMD controller, y-axis (rad/s); Scenario #4.

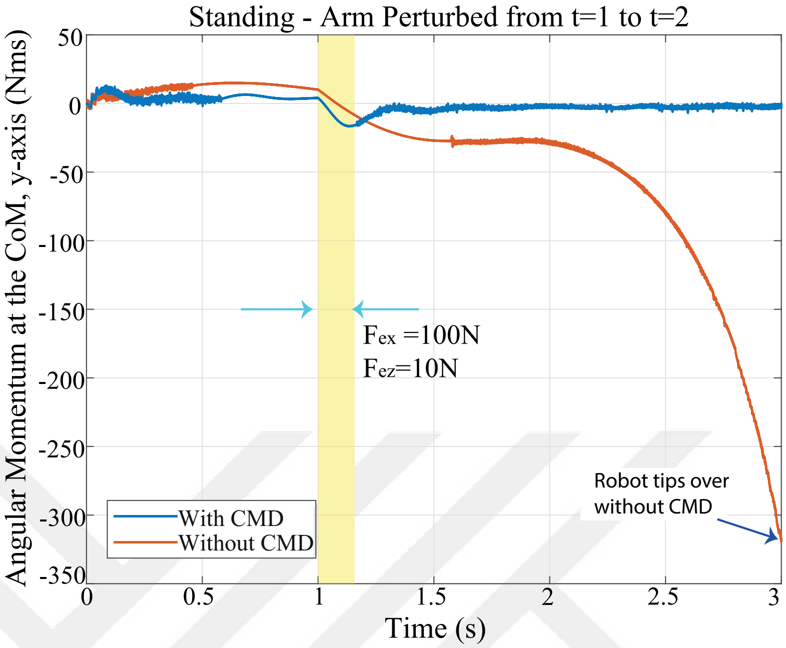


Figure 39: While external force is applied to the lower torso between t=1 and t=1.1 seconds, the angular momentum at the CoM, y-axis (Nms); Scenario #5.

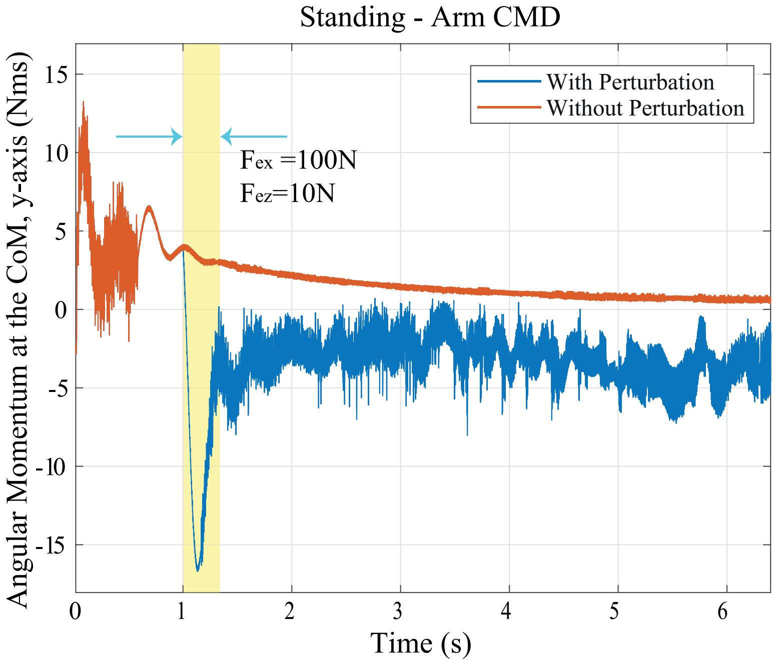


Figure 40: The angular momentum at the CoM in the presence of CMD controller, y-axis (Nms); Scenario #5.

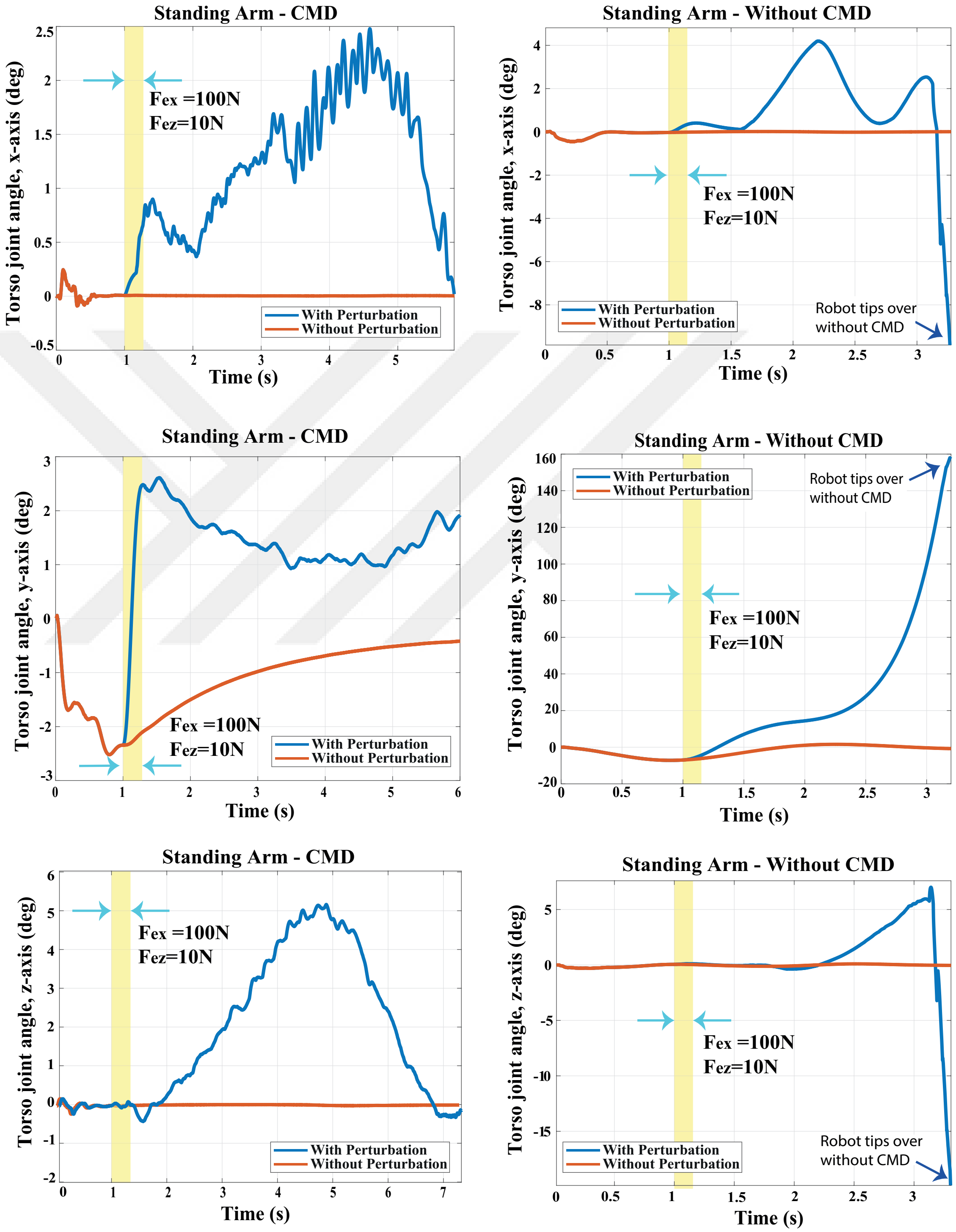


Figure 41: Torso joint angles with and without CMD controller, x-y-z axis (deg); Scenario #5.

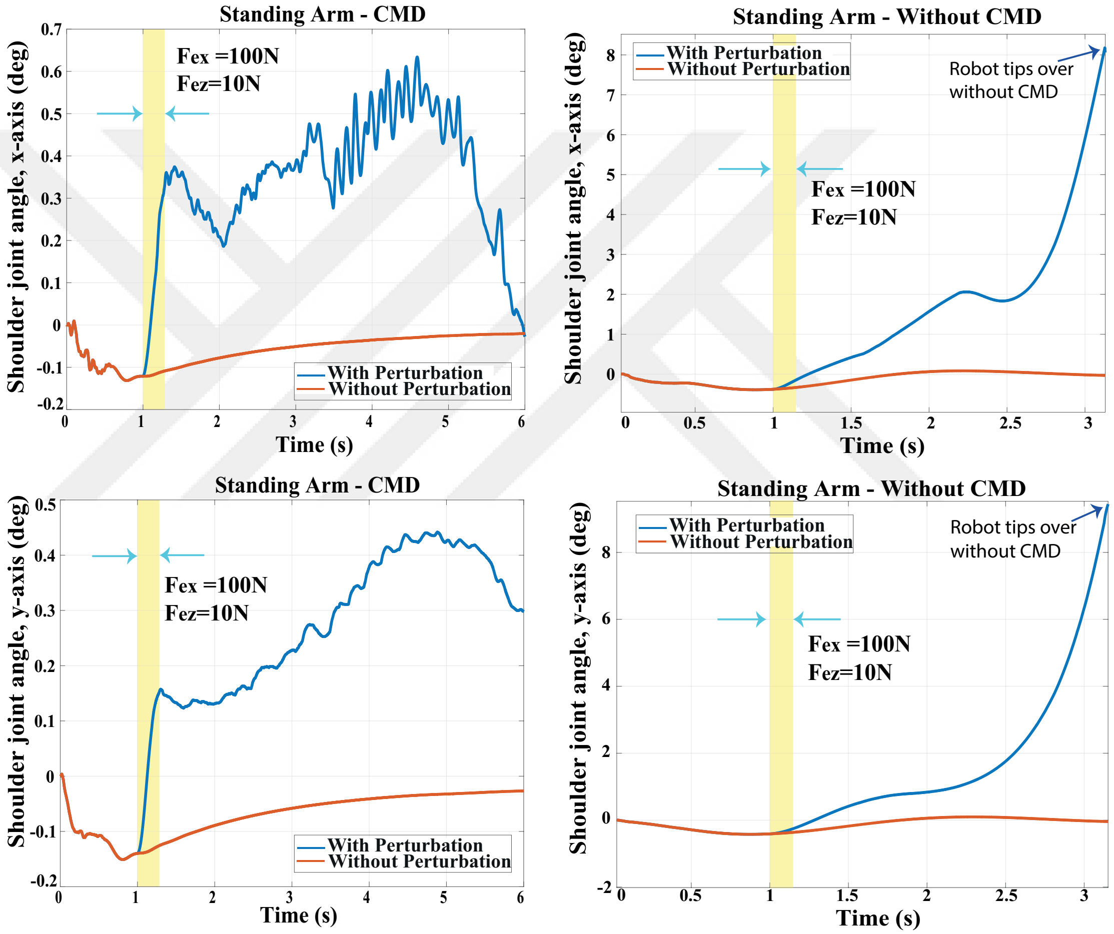


Figure 42: Shoulder joint angles with and without CMD controller, x-y axis (deg); Scenario #5.

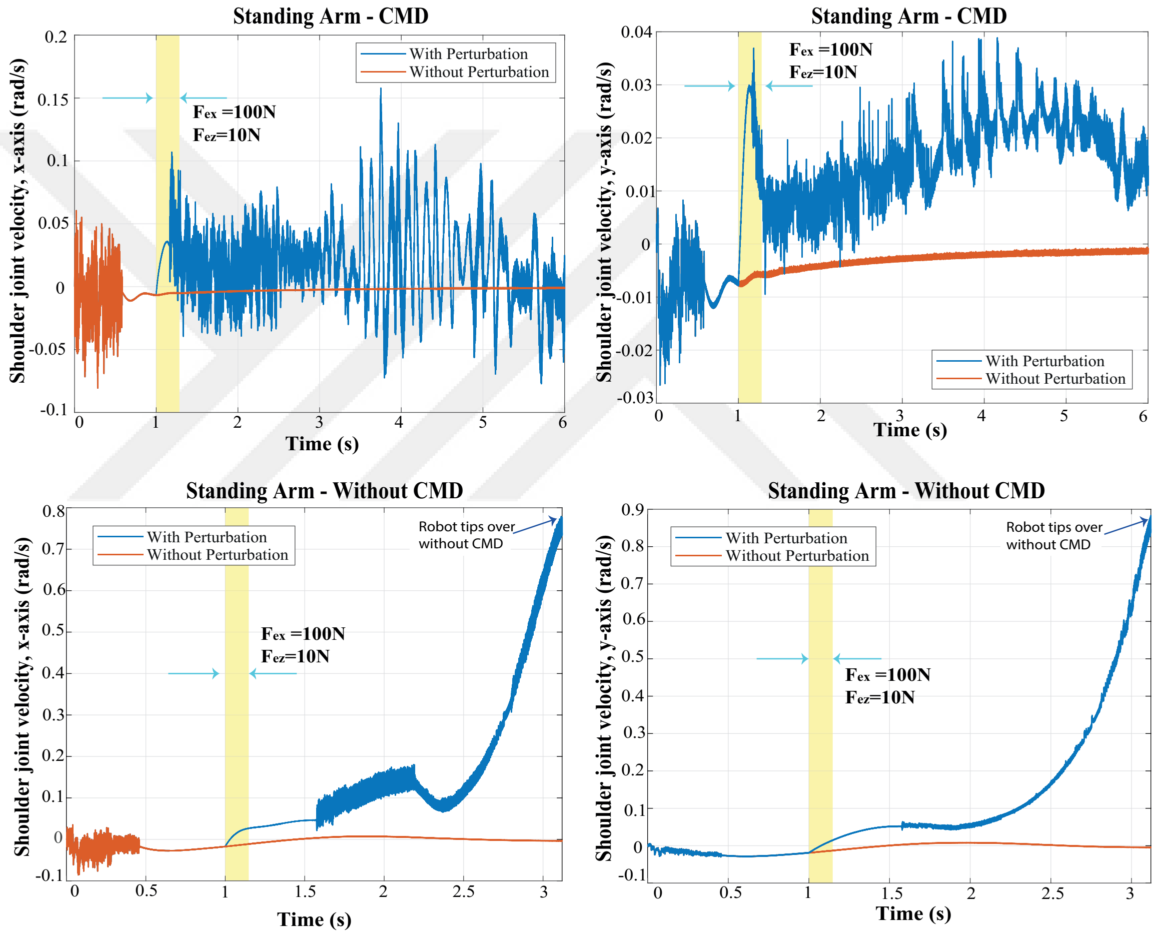


Figure 43: Shoulder joint velocities with and without CMD controller, x-y axis (rad/s); Scenario #5.

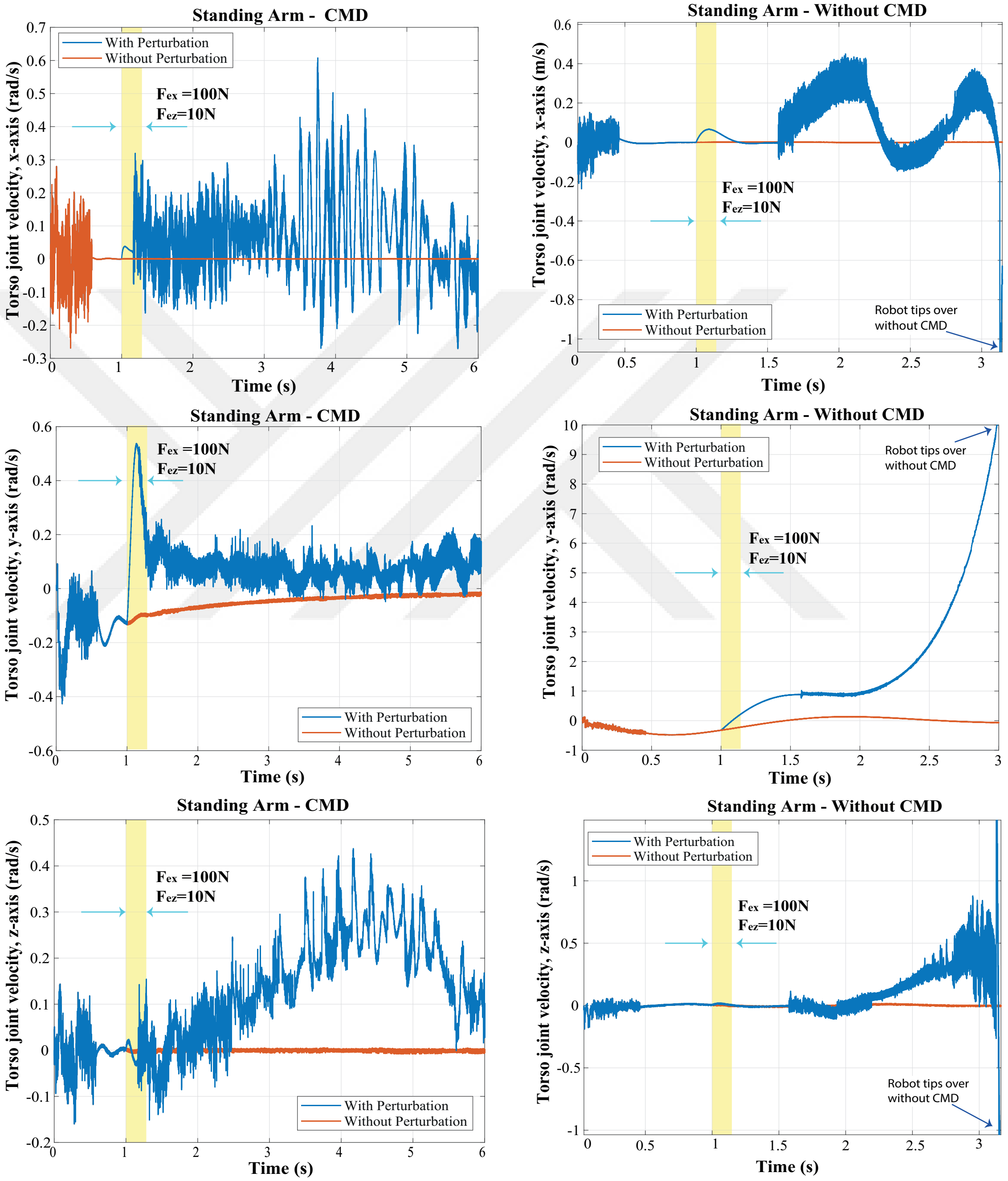


Figure 44: Torso joint velocities with and without CMD controller, x-y-z axis (rad/s); Scenario #5.

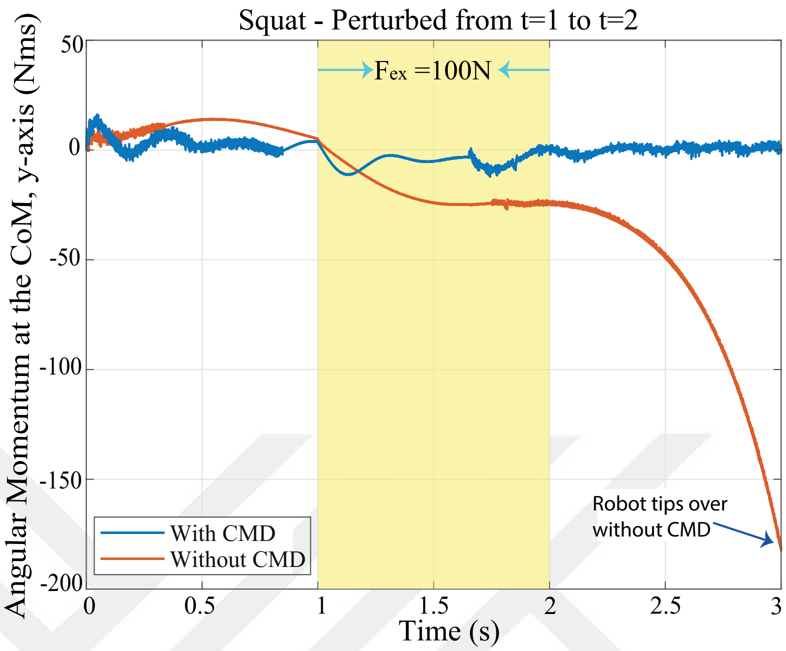


Figure 45: While external force is applied to the lower torso between t=1 and t=2 seconds, the angular momentum at the CoM, y-axis (Nms); Scenario #6.

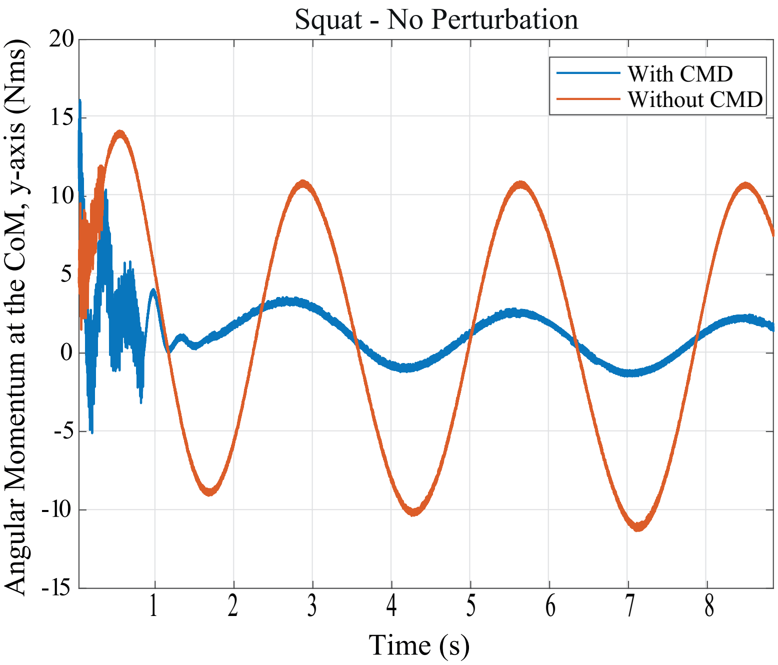


Figure 46: While external force is not applied to the lower torso, the angular momentum at the CoM, y-axis (Nms); Scenario #6.

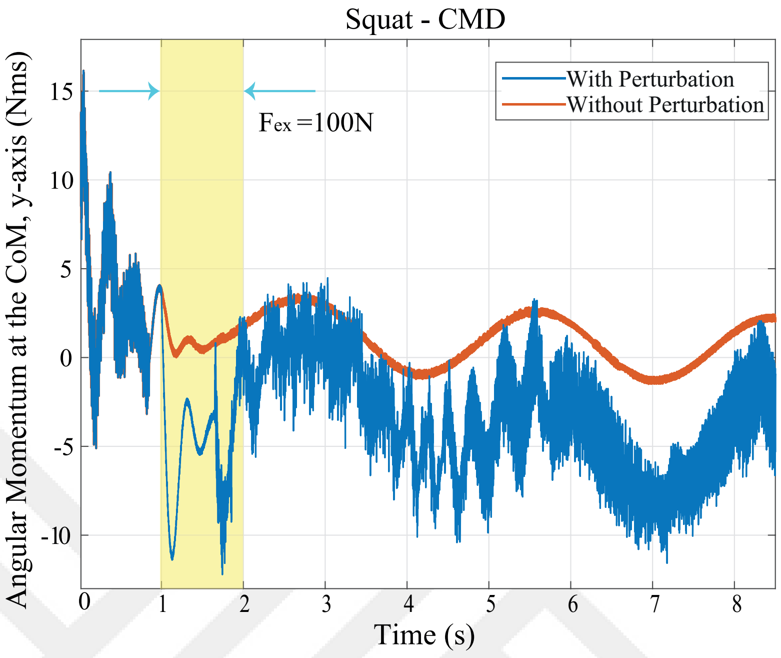


Figure 47: The angular momentum at the CoM in the presence of CMD controller, y-axis (Nms); Scenario #6.

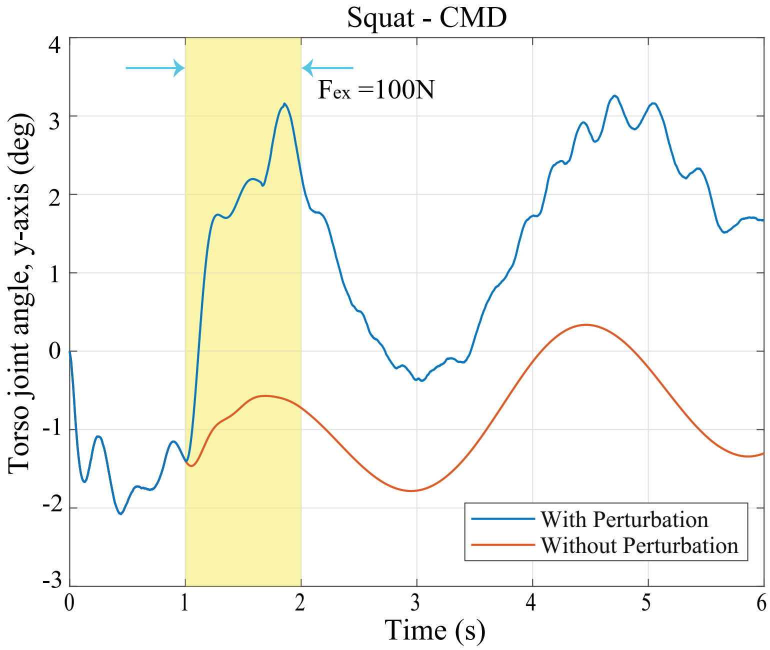


Figure 48: Torso joint angle in the presence of CMD controller, y-axis (deg); Scenario #6.

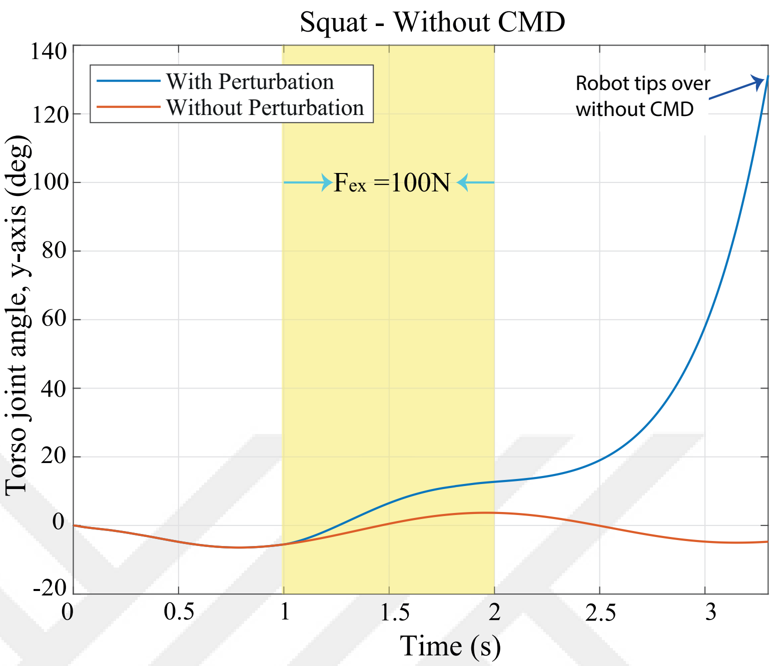


Figure 49: Torso joint angle in the presence of CMD controller, y-axis (deg); Scenario #6.

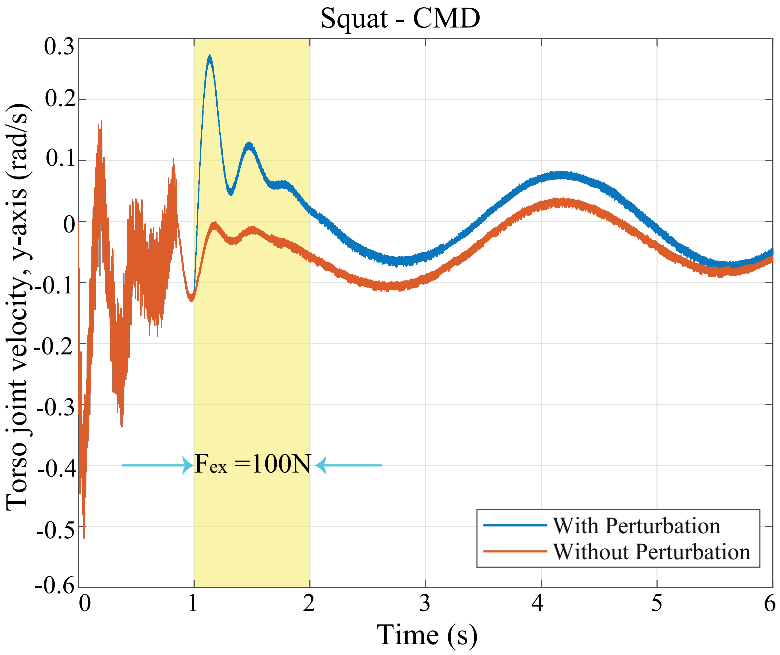


Figure 50: Torso joint velocity in the presence of CMD controller, y-axis (rad/s); Scenario #6.

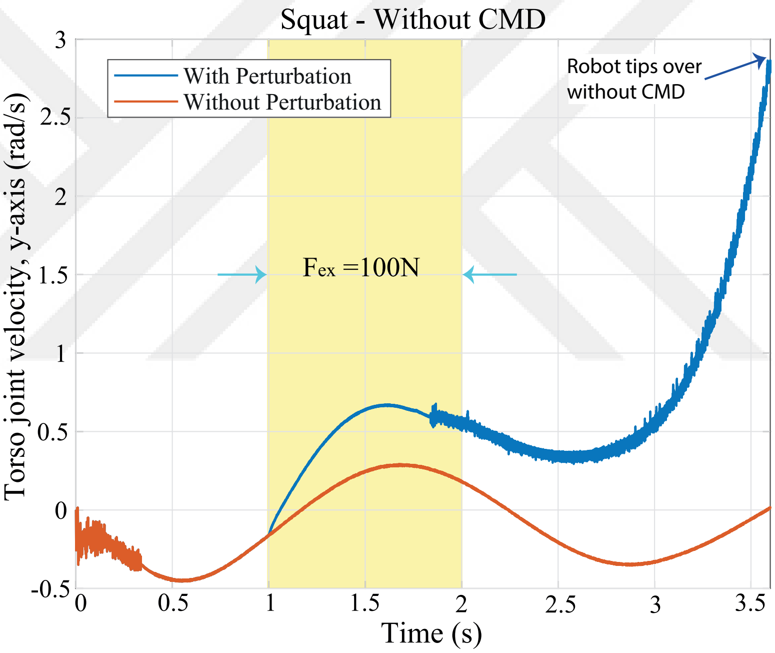


Figure 51: Torso joint velocity in the absence of CMD controller, y-axis (rad/s); Scenario #6.

CHAPTER V

CONCLUSIONS

This thesis presented a comparative study on robust control methods under various controllers. In line with this study, two robots' algorithms and simulations of control methods were examined. A controller based on DOB and centroidal momentum feedback have been proposed to ensure continuous and dynamically stable motion for these robots, and the performance of this controller is based on a widely used conventional centroidal momentum.

For the one-legged robot, the performance of this controller was compared with a classical centroidal momentum feedback controller and a ZMP feedback controller, which are used widely. The comparison study was realized considering three distinct scenarios in which different perturbations were applied to the one-legged robot. As the main result, the proposed CMD controller outperformed other controllers, providing relatively more robust and favourable performance.

As a result of the centroidal momentum observer being more successful comparing three different controllers in a one-legged robot, this controller was tested in Atlas Humanoid Robot, with the thought that the behaviour and control of a one-legged robot can be used as a basis for each part of the multi-legged system. For Atlas Humonaid Robot, a controller based on DOB and centroidal momentum feedback have been applied to provide whole-body control and to achieve continuous and dynamically balanced motion. This controller was compared in the presence and absence of the disturbance. As with the one-legged robot, the presence of the CMD controller was more favourable in three different scenarios than in the case without it.

Bibliography

[1] B. Ugurlu, E. Sariyildiz, T. Kawasaki, and T. Narikiyo, “Agile and stable running locomotion control for an untethered and one-legged hopping robot,” *Autonomous Robots*, Aug. 2021.

[2] P. M. Wensing and D. E. Orin, “Improved computation of the humanoid centroidal dynamics and application for whole-body control,” *International Journal of Humanoid Robotics*, vol. 13, p. 1550039, Mar. 2016.

[3] Y. Kusuda, “Toyota's violin-playing robot,” *Industrial Robot: An International Journal*, vol. 35, pp. 504–506, Oct. 2008.

[4] A. Takanishi, “In memoriam: Professor ichiro kato,” *Autonomous Robots*, vol. 2, no. 1, pp. 7–10, 1995.

[5] S. Shigemi, “ASIMO and humanoid robot research at honda,” in *Humanoid Robotics: A Reference*, pp. 55–90, Springer Netherlands, Oct. 2018.

[6] M. Raibert, *Legged Robots That Balance*. Artificial Intelligence, The MIT Press, first edition ed., 1986.

[7] H. Miura and I. Shimoyama, “Dynamic walk of a biped,” *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 60–74, 1984.

[8] A. Kheddar and et al., “Humanoid robots in aircraft manufacturing: The airbus use cases,” *IEEE Robotics & Automation Magazine*, vol. 26, pp. 30–45, Dec. 2019.

[9] N. G. Tsagarakis and et al., “WALK-MAN: A high-performance humanoid platform for realistic environments,” *Journal of Field Robotics*, vol. 34, pp. 1225–1259, June 2017.

[10] J. O. Brinker, T. Matsubara, T. Teramae, T. Noda, T. Ogasawarsa, T. Asfour, and J. Morimoto, “Walking pattern prediction with partial observation for partial walking assistance by using an exoskeleton system,” in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, IEEE, Aug. 2015.

[11] M. C. Yildirim, A. T. Kansizoglu, S. Emre, M. Derman, S. Coruk, A. F. Soliman, P. Sendur, and B. Ugurlu, “Co-ex: A torque-controllable lower body exoskeleton for dependable human-robot co-existence,” in *IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, IEEE, June 2019.

[12] S. Kagami, K. Nishiwaki, T. Kitagawa, T. Sugihiara, M. Inaba, and H. Inoue, “A fast generation method of a dynamically stable humanoid robot trajectory with enhanced zmp constra,” 2000.

[13] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking," *IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, pp. 368–374 vol.1, 1999.

[14] T. Zielinska, G. R. Rivera Coba, and W. Ge, "Variable inverted pendulum applied to humanoid motion design," *Robotica*, vol. 39, no. 8, p. 1368–1389, 2021.

[15] J. E. Pratt, P. Dilworth, and G. A. Pratt, "Virtual model control of a bipedal walking robot," *Proceedings of International Conference on Robotics and Automation*, vol. 1, pp. 193–198 vol.1, 1997.

[16] 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 *IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, IEEE, 2003.

[17] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life.," *I. J. Humanoid Robotics*, vol. 1, pp. 157–173, 03 2004.

[18] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: system overview and integration," in *IEEE/RSJ International Conference on Intelligent Robots and System*, IEEE.

[19] K. Yamane and Y. Nakamura, "Dynamics filter-concept and implementation of online motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 421–432, June 2003.

[20] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2014.

[21] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, pp. 429–455, July 2015.

[22] T. Buschmann, R. Wittmann, M. Schwienbacher, and H. Ulbrich, "A method for real-time kineto-dynamic trajectory generation," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, IEEE, Nov. 2012.

[23] J. Mayr, H. Gattringer, and H. Bremer, "A bipedal walking pattern generator that considers multi-body dynamics by angular momentum estimation," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, IEEE, Nov. 2012.

[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," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Cat. No.03CH37453)*, IEEE, 2003.

[25] M. E. Abdallah and A. Goswami, “A biomechanically motivated two-phase strategy for biped upright balance control,” *IEEE International Conference on Robotics and Automation*, pp. 1996–2001, 2005.

[26] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Autonomous Robots*, vol. 35, pp. 161–176, June 2013.

[27] D. Orin and A. Goswami, “Centroidal momentum matrix of a humanoid robot: Structure and properties,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Sept. 2008.

[28] C. Li, Y. Ding, and H.-W. Park, “Centroidal-momentum-based trajectory generation for legged locomotion,” *Mechatronics*, vol. 68, p. 102364, June 2020.

[29] B. Ponton, A. Herzog, A. D. Prete, S. Schaal, and L. Righetti, “On time optimization of centroidal momentum dynamics,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018.

[30] K. Ohnishi, M. Shibata, and T. Murakami, “Motion control for advanced mechatronics,” *IEEE/ASME Transactions on Mechatronics*, vol. 1, pp. 56–67, Mar. 1996.

[31] E. Schrijver and J. van Dijk, “Disturbance observers for rigid mechanical systems: Equivalence, stability, and design,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, pp. 539–548, Dec. 2002.

[32] E. Sariyildiz, R. Oboe, and K. Ohnishi, “Disturbance observer-based robust control and its applications: 35th anniversary overview,” *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 2042–2053, Mar. 2020.

[33] H.-W. Chow and N. C. Cheung, “Disturbance and response time improvement of submicrometer precision linear motion system by using modified disturbance compensator and internal model reference control,” *IEEE Transactions on Industrial Electronics*, vol. 60, pp. 139–150, 2013.

[34] Z. Yang, Y. Fukushima, and P. Qin, “Decentralized adaptive robust control of robot manipulators using disturbance observers,” *IEEE Transactions on Control Systems Technology*, vol. 20, pp. 1357–1365, 2012.

[35] S.-H. Lee and A. Goswami, “A momentum-based balance controller for humanoid robots on non-level and non-stationary ground,” *Autonomous Robots*, vol. 33, pp. 399–414, 2012.

[36] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer US, 2008.

[37] A. A. Shabana, *Dynamics of multibody systems*. Cambridge University Press, 2009.

[38] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, “Inverse kinematics with floating base and constraints for full body humanoid robot control,” in *IEEE International Conference on Humanoid Robots*, IEEE, 2008.

[39] T. Flayols, A. D. Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, “Experimental evaluation of simple estimators for humanoid robots,” in *IEEE International Conference on Humanoid Robots*, IEEE, Nov. 2017.

[40] E. Sariyildiz and K. Ohnishi, “Stability and robustness of disturbance-observer-based motion control systems,” vol. 62, pp. 414–422, Jan. 2015.

[41] A. C. Amran, B. Ugurlu, and A. Kawamura, “Energy and torque efficient zmp-based bipedal walking with varying center of mass height,” in *11th IEEE International Workshop on Advanced Motion Control (AMC)*, pp. 408–413, 2010.

[42] S.-H. Lee and A. Goswami, “Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots,” in *IEEE International Conference on Robotics and Automation*, pp. 4667–4672, 2007.

[43] A. Goswami and V. Kallem, “Rate of change of angular momentum and balance maintenance of biped robots,” in *IEEE International Conference on Robotics and Automation,ICRA*, vol. 4, pp. 3785–3790 Vol.4, 2004.

[44] J. Hwangbo, J. Lee, and M. Hutter, “Per-contact iteration method for solving contact dynamics,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[45] Adams, “Hexagon.” https://www.mscsoftware.com/product/adams, (accessed October 10, 2022).

VITA

Dilay Yeşildağ Oral received her BS degree in Mechatronics Engineering from İstanbul Bilgi University. During her undergraduate education, she took place in ”Biomechatronics & Haptics Research Group” with the advisory of Mehmet Ayyıldız and was awarded the second best design project in the department with her graduation project ”The tactile perception of 1D and 2D geometric shapes on the touch screen with electro vibration”. She was admitted to Özyeğin University in September 2019 and has been pursuing her Master's Degree in Mechanical Engineering under the supervision of Asst. Prof Barkan Uğurlu. She has worked as a teaching and research assistant during her education. At the same time, she still serves as an alumni member on the Advisory Board of Istanbul Bilgi University Mechatronics Engineering Department. She is interested in studying topics related to whole-body control systems, robust control for legged robots and robotics.