MOMENTUM-AWARE ROBUST LOCOMOTION CONTROL OF A QUADRUPED ROBOT

A Thesis

by

İbrahim Burak Özkaynak

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

Master of Science

in the
Department of Mechanical Engineering

Özyeğin University
September 2024

MOMENTUM-AWARE ROBUST LOCOMOTION CONTROL OF A QUADRUPED ROBOT

Approved by:

Assoc. Prof. Barkan Uğurlu, Advisor
Dept. of Mechanical Eng.
Özyeğin University

Assoc. Prof. Özkan Bebek
Dept. of Mechanical Eng.
Özyeğin University

Assoc. Prof. Kemalettin Erbatur
Dept. of Mechatronics Eng.
Sabancı University

Date Approved: August 13, 2024

I dedicate this thesis to my dear family.

ABSTRACT

This thesis presented a novel locomotion framework for quadruped robots to maintain dynamically balanced locomotion behavior. The proposed framework consists of two parts: trajectory generation and locomotion controller. Firstly, the trajectory generation provides a vertical center of mass (CoM) trajectory generation based on preview control to accomplish agile locomotion tasks such as jumping and jogging. Conventionally, preview control is implemented for walking planning with constant CoM height since this condition renders the pendulum equation linear. We show that the exact implementation can be done for varying CoM heights while the pendulum frequency is constant. To this end, we compactly derived the infinite horizon and finite horizon time-invariant linear quadratic regulator solutions for potential real-time applicability. Secondly, the proposed locomotion controller provides momentum-aware robust behavior to overcome complicated blind locomotion tasks such as walking on slippery terrain, high-speed trot walking, walking under perturbation, and rough terrain walking. To generate the locomotion controller, we utilize the centroidal momentum concept. To reach this point, we proposed the Centroidal Momentum Observer (CMO)-based controller, which consists of Spatial Force Control with CMO. This controller maintains the robot's operation by working under a nominal wrench and estimating the disturbance wrench. Additionally, We introduced the procedure using the auxiliary variable-based observer design approach to implement CMO. Moreover, the Virtual Model Control (VMC) was implemented to map the wrench obtained from the CMO-based controller as contact force references, which were utilized in computed torque control. The proposed trajectory generation approach and the locomotion controller were tested and compared with conventional approaches via a set

of simulation experiments, where we used a realistic model of the quadruped robot Kara. In addition, the preliminary experiment was conducted on the real quadruped robot to test the proposed locomotion controller.

ÖZETÇE

Bu tez, dinamik olarak dengeli bacaklı hareket davranışını sürdürmek için dört ayaklı robotlar için yeni bir lokomosyon çerçevesi sunmuştur. Önerilen çerçeve iki bölümden oluşmaktadır: yörünge üretimi ve hareket denetleyicisi. İlk olarak, yörünge üretimi, zıplama ve koşma gibi çevik hareket görevlerini yerine getirmek için önizleme denetleyicisine dayalı ağırlık merkezi (CoM) yörüngesi üretimi sağlar. Geleneksel olarak, önizleme denetleyicisi sabit ağırlık merkezi yüksekliği ile yürüme planlaması için uygulanmaktadır, çünkü bu koşul sarkaç denklemini doğrusal hale getirmektedir. Sarkaç frekansı sabitken, değişen CoM yükseklikleri için yörünge üretiminin yapılabileceğini gösteriyoruz. Bu amaçla, sonsuz ve sonlu ufuk zamanlarına sahip, zamanla değişmeyen, doğrusal kuadratik düzenleyicinin kompakt ve gerçek zamanlı uygulanabilir bir çözümünü sunduk. İkinci olarak, önerilen lokomosyon deneyleycisi kaygan arazide yürüme, yüksek hızlı tırıs yürüyüşü, bozucu etki altında yürüme ve engebeli arazide yürüme gibi karmaşık kör hareket görevlerinin üstesinden gelmek için momentuma duyarlı sağlam davranış sağlar. Bacaklı hareket denetleyicisini oluşturmak için merkezsel momentum kavramını kullanıyoruz. Bu noktaya ulaşmak için, Merkezsel Momentum Gözlemcisi (CMO) ile Uzamsal Kuvvet Denetleyecisi'nden (SFC) oluşan CMO tabanlı denetleyiciyi önerdik. Bu denetleyici, nominal bir kuvvet-moment vektörü altında çalışmasını sağlayarak ve bozucu kuvvet-moment vektörünü tahmin ederek robotun yürümesini iyileştirir. Ek olarak, CMO'yu uygulamak için yardımcı değişken tabanlı gözlemci tasarım yaklaşımını kullanan yöntemi tanıttık. Ayrıca, CMO tabanlı denetleyicisinden elde edilen kuvvet-moment vektörünü, hesaplanan tork denetleyecisinde kullanılan yer tepki kuvveti referansları olarak haritalamak için Sanal Model Denetleyecisi (VMC) uygulanmıştır. Önerilen yörünge oluşturma yaklaşımı ve hareket

denetleyicisi, dört ayaklı robot Kara'nın gerçekçi bir modelini kullandığımız bir dizi simülasyon deneyi aracılığıyla test edilmiş ve geleneksel yaklaşımlarla karşılaştırılmıştır. Ayrıca, önerilen lokomosyon denetleyecisini test etmek için gerçek bir dört ayaklı robot üzerinde bir ön deney gerçekleştirilmiştir.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to the people who shared their time, knowledge, patience, and motivation:

My thesis supervisor; Barkan Uğurlu,

My professors; Özkan Bebek, Emre Sarıyıldız, Ramazan Ünal, Ahmet Fatih Tabak, Kenan Kaya, Yalçın Çekiç,

The quadruped robot team: Erim Can Özçınar, Burhan Kara, Barış Balcı, Sinan Emre,

My lab mates: Ahmed Adel Fahmy Soliman, Altun Rzayev, Barış Baysal, Deniz Yılmaz, Ege Gediksiz, Elshan Sarkarfarshi, Giti Tammari, Omid Arfaei, Süleyman Can Çevik, Sinan Coruk, Tuğçe Ersoy, Mahsa Sarrafi, and Nour Amin,

My close friends; Hasan Tahsin Kokonalıoğlu, Erhan Ekmen, Berk Akgüneş, Mert Karakurt, Yunus Emre Koş, Damla Ayyıldız, Ece Melis Erkoçak, Mahshad Azima, Sümeye Vatansever, Seray Afacan, Pelin Ülker, Su Demirkol, Vahide Taşdelen, Doğa Cengiz,

My family; Münevver Özkaynak, Murat Özkaynak, Mehmet Taha Özkaynak, Sevde Özkaynak, Yusuf Fatih Erol,

The electronics technician; Mehmet Karakoç.

I would also like to thank Özyeğin University for its support and opportunities during the master's program.

ACRONYMS

CoM Center of Mass

CMO Centroidal Momentum Observer

SFC Spatial Force Control

TO Trajectory Optimization

RHC Receding Horizon Control

LIP Linear Inverted Pendulum

LIPM Linear Inverted Pendulum Mode

ZMP Zero Moment Point

MPC Model Predictive Control

CPG Central Pattern Generation

GM Generalized Momentum

GMO Generalized Momentum Observer

QP Quadratic Programming

CPU Central Process Unit

RAM Random Access Memory

GRF Ground Reaction Force

F/E Flexion Extension

A/A Flexion Extension

AHRS Attitude and Reference Heading System

ICS Inertial Coordinate System

CMM Centroidal Momentum Matrix

LQR Linear Quadratic Regulator

DLQR Discrete Linear Quadratic Regulator

DARE Discrete Algebraic Riccati Equation

RNEA Recursive Newton Euler Algorithm

RMSE Root Mean Square Error

RMS Root Mean Square

SYMBOLS

Symbol	Description
$\mathbb{R}$	Euclidean Space
$\mathbb{H}$	Hilbert Space
$\mathbf{r}_f, \mathbf{p}_f$	Foot Position Vector
$\mathbf{r}_c$	CoM Position Vector
$\mathbf{r}_{ch}$	CoM to Hip Position Vector
$\mathbf{r}_{hf}$	Hip to Foot Position Vector
$\mathbf{R}_t$	Rotation Matrix of the Torso
p_{fx}	Robot Foot position in x axis
p_{fy}	Robot Foot position in y axis
p_{fz}	Robot Foot position in z axis
l_i	Robot link lengths
$\dot{\mathbf{p}}_{c,i}$	Foot contact velocity vector
$\mathbf{J}_{c,i}$	Contact Jacobian matrix
$\bar{\mathbf{J}}_c$	Simplified contact Jacobian matrix
$\bar{\mathbf{J}}_{cf}$	Simplified contact Jacobian matrix corresponding to Floating-Base Velocity
$\bar{\mathbf{J}}_{cr}$	Simplified contact Jacobian matrix corresponding to robot joint velocities
$\mathbf{S}$	Selection matrix
$\mathbf{q}$	Generalized coordinates vector
$\dot{\mathbf{q}}$	Generalized velocities Vector

$\dot{\mathbf{q}}_f$	Generalized velocities vector corresponding to Floating-Base
$\dot{\mathbf{q}}_r$	Generalized velocities vector corresponding to robot joints
$\mathbf{J}_t$	Task Jacobian matrix
$\dot{\mathbf{p}}_t$	Task space position vector
$\dagger$	Moore-Penrose pseudo inverse
$\mathbf{M}(\mathbf{q}), M(q)$	Joint-space inertia matrix
$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), C(q, \dot{q})$	Coriolis matrix
$\mathbf{G}(\mathbf{q}), G(q)$	Gravity vector
$\mathbf{F}_{con}$	Contact force vector
$\tau, \mathcal{T}_{cmd}$	Command torque vector
$\mathbf{h}_G$	Spatial momentum vector
$\mathbf{I}_i$	Spatial momentum of a body i
$\mathbf{l}_i$	Linear momentum of a body i
$\mathbf{k}_i$	Angular momentum of a body i
$\mathbf{m}_i$	Mass of a body i
$\mathbf{c}_i$	CoM position of the body i
$\mathbf{h}_G$	Centroidal momentum vector
$\mathbf{X}_G$	Spatial transformation matrix
$\mathbf{A}_G(\mathbf{q})$	Centroidal momentum matrix
p_x	ZMP position
c_x	CoM position along x-axis
c_z	CoM position along z-axis
$\ddot{c}_x$	CoM acceleration along x-axis

$\ddot{c}_z$	CoM acceleration along z-axis
$\dddot{c}_z$	CoM jerk along z-axis
$\dot{L}_y$	Moment of the pendulum along y-axis
w	Natural frequency of the pendulum
x, x^*	State vector
y	Output vector
u	Control input
A	System matrix
B	Input matrix
C	Output matrix
T	Time step
x_d	Desired state vector
y^{ref}	Reference output
y	Measured output
e	Output error
Δx	State variation vector
Δu	Control input variation
Q_e	Weight for the error
Q_x	Weight for the state variation
Q	Weight matrix
R	Weight for the control input
$\tilde{A}$	Augmented system matrix
$\tilde{B}$	Augmented input matrix
$\tilde{x}$	Augmented state vector
K_i	Integral gain
K_d	Delay gain
K_s	State feedback gain

$\mathcal{H}$	Hamiltonian matrix
λ^*	Costate vector
V	Eigenvector matrix
V_x	Stable eigenvector block for state
V_λ	Stable eigenvector block for costate
D	Diagonal eigenvalue matrix
Λ	Diagonal stable eigenvalue matrix
S	Solution to the DARE
$\boldsymbol{K}$	Augmented gain vector obtained from the DARE
$\tilde{A}_c$	Closed-loop system matrix
$\tilde{X}$	Closed-loop system state vector
N_L	Horizon length of the DARE
T_{fl}	Flight phase
T_s	Support phase
T_d	Double support phase
T_{tr}	Transfer phase
$K_p, \mathbf{K_p}$	Joint controller proportional gain matrix
K_v	Joint controller derivative gain matrix
K_{ff}	Feed-forward gain matrix
K_{fb}	Feedback gain matrix
N_{cont}	Number of contacts
F^{ref}	Contact force reference
F_{err}	Contact force error
$\ddot{\mathbf{p}}_{cmd}$	Operational-Space acceleration command vector
$\ddot{\mathbf{p}}_{com}^{ref}$	Cartesian-space reference acceleration vector
$\dot{\mathbf{p}}_{com}^{ref}$	Cartesian-space reference velocity vector

$\ddot{\mathbf{p}}_{com}$ Cartesian-space measured acceleration vector

$\dot{\mathbf{p}}_{com}$ Cartesian-space measured velocity vector

$\mathbf{K}_a$ Feedback gain vector for the Cartesian-Space acceleration

$\mathbf{K}_v$ Feedback gain vector for the Cartesian-Space velocity

$\mathbf{K}_f$ Feed-forward gain vector for the Cartesian-Space acceleration

$\mathcal{F}_{\text{cal}}$ Calculated force vector

$\mathbf{Q}_{\text{torso}}$ Quaternion vector of the robot torso

η Scalar part of the quaternion vector

$\boldsymbol{\epsilon}$ Vector part of the quaternion vector

η^{ref} Scalar part of the reference quaternion vector

$\boldsymbol{\epsilon}^{\text{ref}}$ Vector part of the reference quaternion vector

$\boldsymbol{\omega}^{\text{ref}}$ Reference angular velocity vector

$\boldsymbol{\omega}$ Angular velocity vector

$\boldsymbol{\omega}_{\text{cmd}}$ Angular velocity command vector

$\dot{\boldsymbol{\omega}}^{\text{ref}}$ Reference angular acceleration vector

$\dot{\boldsymbol{\omega}}_{\text{cmd}}$ Angular acceleration command vector

$\boldsymbol{e}_o$ Orientation error vector

$\mathbf{K}_o$ Angular position gain matrix

$\mathbf{K}_\omega$ Angular velocity control gain matrix

$\mathbf{I}_{\text{torso}}$ Torso inertia tensor

$\mathcal{M}_{\text{cal}}$ Calculated moment vector

$\mathbf{W}_{\text{cal}}$ Calculated wrench vector

$\mathbf{W}_{\text{des}}$ Desired wrench vector

$\mathbf{W}_{\text{n}}$ Nominal wrench vector

$\mathbf{W}_{\text{grav}}$ Gravity wrench vector

$\mathbf{W}_{\text{cmd}}$ Wrench command vector

$\mathbf{A}_{\mathrm{Gn}}(\mathbf{q})$	Nominal centroidal momentum matrix
$\dot{\mathbf{h}}, \mathbf{f}_G$	Net external wrench vector in CoM
$\Delta\mathbf{h}$	Centroidal momentum disturbance vector
$\Delta\dot{\mathbf{h}}$	Disturbance centroidal momentum vector
$\Delta\ddot{\mathbf{h}}$	Rate change of the disturbance wrench vector
$\mathbf{z}$	Auxiliary variable vector
$\dot{\mathbf{z}}$	First order auxiliary variable vector
$\hat{\dot{\mathbf{z}}}$	Estimated first order auxiliary variable vector
$\mathbf{L}, \mathbf{U}$	Observer gain matrices
$\mathbf{f}_i$	Contact force command vector for the foot i
$\mathbf{F}_{\mathrm{con}}^{\mathrm{cmd}}$	Contact force command vector
${}^{i}\mathbf{r}_G$	Foot contact position with respect to CoM
$\mathbf{M}_f$	Joint-space inertia matrix corresponding to floating-base
$\mathbf{M}_r$	Joint-space inertia matrix corresponding to robot joints
$\mathbf{M}_{rf}, \mathbf{M}_{fr}$	Coupling joint-space inertia matrices
$\mathbf{h}_f, \mathbf{h}_r$	Nonlinearity vectors corresponding to the floating-base and robot joints
$\mathbf{F}_{\mathrm{con}}$	Measured contact force vector
$\dot{\mathbf{h}}_n$	Measured nominal wrench vector
$\mathbf{A}_{vmc}$	VMC matrix
$\dot{\mathbf{h}}_n^{\mathrm{cmd}}$	Nominal wrench command vector
$\dot{\mathbf{h}}_n^{\mathrm{des}}$	Nominal desired wrench vector
$\mathbf{e}_r^{\mathrm{cmd}}$	Joint position error vector
$\dot{\mathbf{e}}_r^{\mathrm{des}}$	Joint velocity error vector
$\mathbf{C}_1, \mathbf{C}_2$	Stability condition matrices
τ_f	Joint friction torque

χ Friction compensation activator boolean

α Friction compensation amplitude parameter

ϵ Scaling factor parameter of the friction compensation

γ Velocity coefficient of the friction compensation

β Offset parameter of the friction compensation

κ Velocity threshold of the friction compensation

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Literature Review

Recent advancements in legged robotics technology provide the development of various robots that are capable of executing challenging tasks, such as fast walking, jumping, jogging, walking on rough terrain, and maintaining balance under disturbances. Quadruped robots hold an essential place among legged robots owing to their ability to exhibit these locomotion scenarios. Concerning these scenarios, agility and robustness are one of the keys locomotion characteristics. There were developed quadruped robots exhibiting agile and robust locomotion behavior with various actuation strategies. For instance, Boston Dynamics developed the hydraulically driven Big Dog and, it represents running locomotion [2]. MIT Mini Cheetah executed 360° back-flips and it is electrically actuated [3].

Agility and robustness are not only attributed to their actuation strategy but also to the performance of their locomotion frameworks. The locomotion framework can be defined as two main parts: Trajectory generation and gait planning, and full state feedback control. In this sense, optimal control problem solutions are the essential tools for the trajectory optimization (TO) and the model-based dynamic locomotion approaches are becoming more prevalent. For the trajectory optimization there are diverse approaches exist. Such as, Li et al. proposed reduced antero-posterior sequence gait parametrization technique that provides diverse gaits [4]. Similarly, MIT Cheetah 3 has exhibited impressive jumping with an efficient TO approach, tracking controller, and landing with a robust stabilization controller [5]. Cebe et al. brought about trajectory optimization algorithm provides dynamically feasible contact location definition and base motion generation over obstacled terrain [6].

Even though recent advances in algorithms, e.g., receding horizon controllers (RHC), yield successful results, they generally work at a low sampling rate, especially when detailed models are used. Simplified models, on the other hand, lead to less complex formulations, and efficient RHC-based trajectory strategies can be developed. To that end, we defend the idea that preview control-based trajectory generation has potential. Kajita et al. used the preview control strategy to generate a center of mass (CoM) trajectory for a biped robot walking with zero moment point (ZMP) control. They formulated the algorithm employing a servo problem and utilized the cart-table model based on the 3D linear inverted pendulum (LIP) mode with horizontal plane constraint and ZMP equations [7]. Preview control-based trajectory generation was also realized using three mass models in [8], indicating that it is efficient with more complex models.

Recently, preview control-based trajectory generation approaches have gained popularity. Murooka et al. showed that preview control-based online trajectory generation that uses centroidal dynamics and stabilization control for humanoid dynamic motion is more computationally efficient than conventional MPC [9]. Additionally, Ryu et al. introduced a preview control-based walking pattern generation for biped robots containing vertical CoM motion through a time-varying ZMP equation [10]. While some of these studies provide suitable solutions for the case of varying CoM height, their computational complexity makes their applicability challenging. Furthermore, implementing these approaches to generate agile locomotion tasks like jumping and jogging may not be straightforward.

To execute the generated trajectory, the sensory or state feedback controllers are applicable to maintain dynamically coherent locomotion. Righetti et al. proposed Central Pattern Generation (CPG), also called a system of coupled oscillators with a sensory feedback technique to locomote the quadrupedal robot, and the robot could walk on uneven terrain without an extended control pipeline [11, 12]. Fahmi et al. introduced the terrain compliance estimator combined with the whole-body controller, which provides online adaptation in varying terrains for contact consistency [13]. Through this, they provides robust blind locomotion for quadruped robots. In this context, feasible contact force distribution computation is one of the key elements to reach whole-body behavior on varying terrain blind locomotion.

The optimal control-based approach has gained popularity in blind locomotion and provides feasible contact force distribution that regularizes posture. For instance, Model Predictive Control (MPC) used a single rigid body model (SRBM) to determine feasible ground reaction forces, and this approach was implemented on MIT Cheetah 3 [14]. Utilizing a simplified model can bring uncertainties and model errors. Addressing these issues, Bledt and Kim presented a Regularized Predictive Control framework that regulates heuristics for legged robot locomotion. In light of this framework, the robot's parametric uncertainties and complex dynamics can be approximated; thus, the robot's dynamically feasible states can be achieved with optimal foot positions and contact forces as input [15]. The model errors and uncertainties can also be tackled with the disturbance observers. For instance, Generalized Momentum Observer (GMO) is a common concept in blind locomotion that provides disturbance and contact force estimation to regulate posture and maintain the robot's balance on varying terrain [16, 17]. Bledt et al. proposed the discrete-time GM-based disturbance observer to enrich the accuracy of the force control [18]. This observer can be utilized for various control frameworks, and it can be combined with QP formulation to calculate dynamically feasible contact force distribution that contains interaction force [19]. Moreover, the learning-based approaches were utilized for the ground reaction for estimation. An and Lee proposed the neural network approach to estimate the ground reaction force [20].

1.2 Contribution

The literature review presented the various locomotion frameworks for legged robots, such as optimal control-based, model, and learning-based approaches. The main idea of these approaches is maintaining the agile, robust, and dynamically coherent locomotion behavior.

In this thesis, we presented the locomotion control framework that provides the jumping and spot jogging trajectory generation and momentum-aware controller to execute dynamically balanced locomotion on varying terrain. The five main contributions are indicated via this thesis that are:

1. **Jumping and Jogging Trajectory:** We demonstrate that with constant pendulum frequency, we may utilize preview control for vertical CoM trajectory generation; modulation of this the parameter enables us to yield jumping and spot-jogging behavior.

2. **Preview Control Gain Calculation:** A compact derivation of the infinite horizon and finite horizon time-invariant linear quadratic regulator solutions were yielded for potential real-time applicability

3. **Centroidal Momentum Observer-based Control:** A control framework that utilizes the centroidal momentum concept provides robust locomotion on all terrain types and posture regulation under perturbation. Additionally, the implementation of the CMO with virtual model control is designated as a contact force reference for a computed torque control presented.

4. **Centroidal Momentum Observer Synthesis:** A novel CMO derivation approach presented by utilizing the auxiliary variable-based observer design approach.

5. **Stability Analysis of the Control Framework:** A stability analysis of the

controller framework is presented. The stability analysis utilizes the operational space control, the joint level controller, and the robot dynamics, and it defines the generalized stability conditions.

6. **Friction Compensation:** A friction compensation approach is presented to enhance joint tracking of the robot's actuator.

The locomotion control framework was validated through the simulations. The proposed trajectory generation approach was compared with the naive approaches via simulations. Moreover, we implemented a preview control-based pattern generation approach for the forward walking simulations. Additionally, the simulation benchmark conducted the proposed force controller framework with the SFC and GMO-based controllers.

CHAPTER II

QUADRUPED ROBOT KINODYNAMICS

2.1 Robot Model

The quadruped robot Kara in Fig. 1 has 12 custom electrical actuators with high payload capabilities. Three degrees of freedom (DoF) exist for each leg of the robot: knee hip flexion extension (F/E), hip F/E, and hip abduction/adduction (A/A). The robot specifications are also given in Table 1. Torso orientation and acceleration are obtained using the Attitude and Reference Heading System (AHRS) unit, and absolute encoders obtain joint-related metrics. The robot's main controller is the Jetson Xavier AGX, equipped with an 8-core ARM v8.2 64-bit CPU and 32 GB RAM. There are four force sensor units for four legs, and they are utilized to measure GRF. Moreover, these force sensors are employed for contact detection to execute inverse dynamics-based controllers accurately. To verify the proposed controller, we constructed the realistic simulation model of Kara on Raisim v1.1.7, as seen in Fig. 1 [21].

Table 1: Kara Specifications

Torso Size (L×H×W):	$0.576 \times 0.358 \times 0.202$
Total Weight:	40 kg
Thigh-Knee:	0.38 m
Knee-Foot:	0.32 m
Total actuated DoF:	3 in each leg 3 x 4 =12 DoF
Encoder Resolution:	0.0014 °/count
Rated Output Torque:	48 Nm
Gear Ratio:	30
Sampling Frequency:	500 Hz

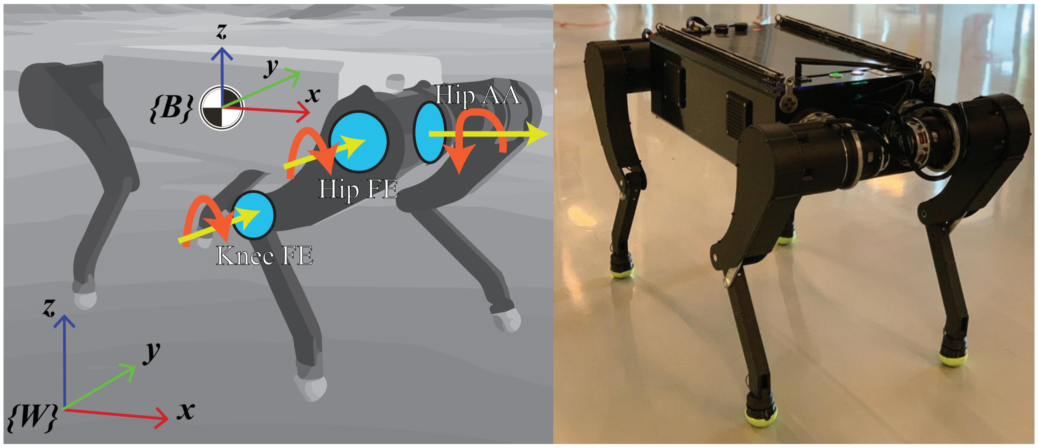

Figure 1: Left: Quadruped Robot Kara Simulation Model. Right: Quadruped Robot Kara

2.2 Robot Kinematics

The robot kinematics are derived as in [22, 23] to relate joint space to task space. The robot kinematics includes floating base kinematics and leg kinematics for motion generation. Moreover, constraint and operational space kinematics are presented for the stability analysis of the whole-body controller.

2.2.1 Floating-base Kinematics

The floating-base kinematics was obtained through a position vector relation as seen in Fig. 2. The foot position $\mathbf{r}_f \in \mathbb{R}^3$ in the world frame can be yielded as follows:

$$\mathbf{r}_f = \mathbf{r}_c + \mathbf{R}_t(\mathbf{r}_{ch} + \mathbf{r}_{hf}) \tag{1}$$

where $\mathbf{r}_c \in \mathbb{R}^3$ is the CoM position vector. The vectors $\mathbf{r}_{ch} \in \mathbb{R}^3$ and $\mathbf{r}_{hf} \in \mathbb{R}^3$ represent the position of the foot relative to the hip A/A and the CoM, respectively. $\mathbf{r}_{ch} \in \mathbb{R}^3$ is the hip A/A joint position vector relative to CoM. $\mathbf{R}_t \in \mathbb{R}^{3\times3}$ is the rotation matrix of the torso in the Euler angles domain. The foot position vector relative to hip A/A was obtained as follows:

$$\mathbf{r}_{hf} = \mathbf{R}_t^\intercal(\mathbf{r}_f - \mathbf{r}_c) - \mathbf{r}_{ch} \tag{2}$$

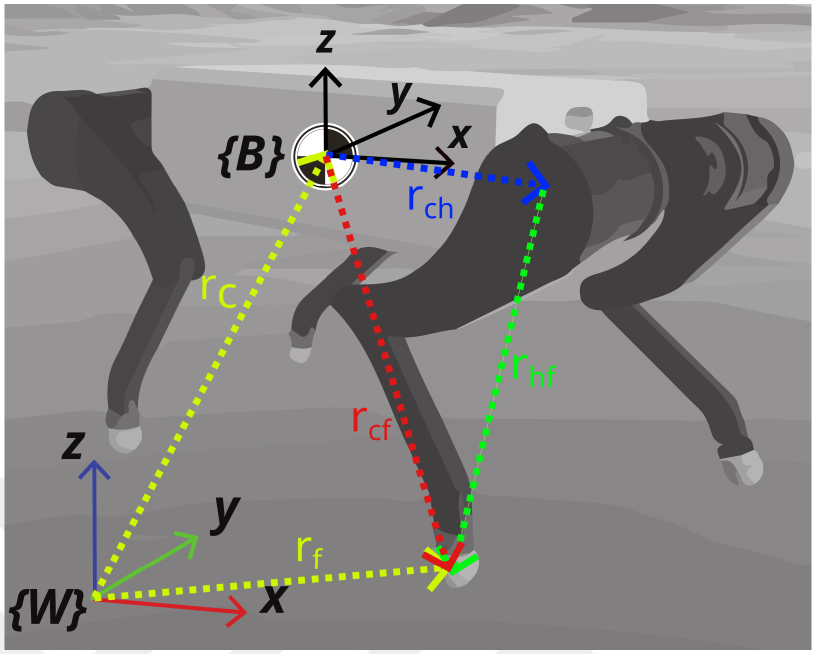

Figure 2: Kinematic Diagram of the Robot

where $\mathbf{R}_t$, $\mathbf{r}_f$ and $\mathbf{r}_c$ can be acquired from the trajectory generator. Finally, $\mathbf{r}_{hf}$ was obtained to compute joint angles [22].

2.2.2 Single Leg Kinematics

In Fig. 3, the single-leg kinematic diagram of the robot is indicated. The blue rectangular box represents the robot's torso. While black lines represent the robot links (l_0, l_1, l_2, l_3, l_4 and l_5), purple circles represent the robot's actuated joints (q_0, q_1, and q_2). Furthermore, the green circle is the contact position of the foot. The black circle represents the robot's fixed paw angle (q_3). The foot position can be obtained as follows:

$$\mathbf{p}_f = \begin{bmatrix} p_{fx} & p_{fy} & p_{fz} \end{bmatrix}^\intercal \tag{3}$$

$$p_{fx} - l_0 = l_4 \sin(q_1 + q_2) + l_3 \sin(q_1) + l_5 \sin(q_1 + q_2 + q_3) \tag{4}$$

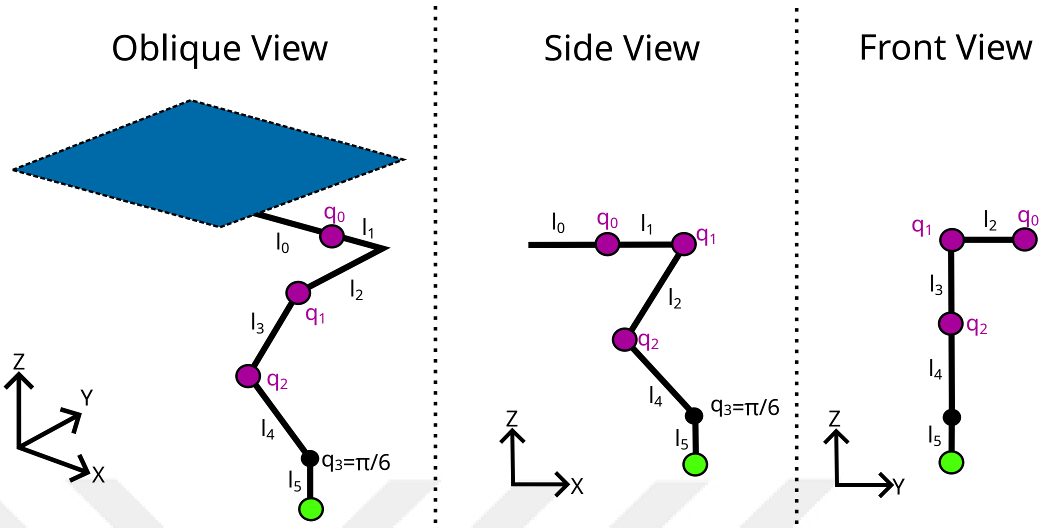

Figure 3: Leg Kinematics Diagram of the Robot

$$
\begin{aligned}
p_{fy} &= l_1\cos(q_0) + l_2\cos(q_0) - l_4\cos(q_1+q_2)\sin(q_0) \\
&- l_3\cos(q_1)\sin(q_0) - l_5\cos(q_1+q_2+q_3)\sin(q_0)
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
p_{fz} &= l_1\sin(q_0) + l_2\sin(q_0) + l_4\cos(q_1+q_2)\cos(q_0) \\
&+ l_3\cos(q_1)\cos(q_0) + l_5\cos(q_1+q_2+q_3)\cos(q_0)
\end{aligned} \tag{6}
$$

where p_{fx}, p_{fy}, and p_{fz} are the contact positions of the foot in x, y and z, respectively. $\|\mathbf{p}_f\|^2$ yields the following relations:

$$
\begin{aligned}
C_2 &= A_2\cos(q_2) + B_2\sin(q_2) \\
A_2 &= 2l_4 + 2l_5\cos(q_3) \\
B_2 &= -2l_5\cos(q_3)
\end{aligned} \tag{7}
$$

$$
C_2 = \frac{1}{l_3}[(p_{fx}-l_0)^2 + p_{fy}^2 + p_{fz}^2 - (l_1+l_2)^2 - l_3^2 - l_4^2 - l_5^2 - 2l_4l_5\cos(q_3)] \tag{8}
$$

Utilizing (7), we can calculate q_2. Additionally, by rearranging (4), the following relation is obtained:

$$\begin{aligned} P_{fx} - l_0 &= A_1 \cos(q_1) + B_1 \sin(q_1) \\ A_1 &= l_4 \sin(l_2) + l_5 \sin(q_2 + q_3) \\ B_2 &= l_3 + l_4 \cos(l_2) + l_5 \cos(l_2 + l_3) \end{aligned} \tag{9}$$

The eq. (9) is utilized to determine q_1. Furthermore, equation (5) is reformulated as follows:

$$\begin{aligned} p_{fy} &= A_0 \cos(q_0) + B_0 \sin(q_0) \\ A_0 &= l_1 + l_2 \\ B_0 &= l_4 \cos(q_1 + q_2) + l_3 \cos(q_1) + l_5 \cos(q_1 + q_2 + q_3) \end{aligned} \tag{10}$$

The equations (7), (9), and (10) can be addressed using the analytical solution for $A\cos(x) + b\sin(x) = c$. By analytically solving (7), (9), and (10), we can determine q_2, q_1, and q_0, respectively.

2.2.3 Constraint Kinematics

The constraint kinematics was derived based on the assumption that the one or more feet are in contact $\dot{\mathbf{p}}_{\mathbf{c}} = \mathbf{0} \in \mathbb{R}^k$. For each stance leg, there are three constraints along x-y-z axes. In this sense, if all 4 feet are on the ground, the dimension of this constraint is $k = 12$. The velocity of the feet can be defined as below:

$$\dot{\mathbf{p}}_{c,i} = \mathbf{J}_{c,i}\dot{\mathbf{q}} \qquad \text{where} \quad \dot{\mathbf{q}} = [\dot{\mathbf{q}}_f \ \dot{\mathbf{q}}_r]^\intercal \tag{11}$$

$\dot{\mathbf{p}}_{c,i} \in \mathbb{R}^3$ and $\mathbf{J}_{c,i} \in \mathbb{R}^{3\times 18}$ represents the i^{th} foot velocity and Jacobian, respectively. $\dot{\mathbf{q}} \in \mathbb{R}^{18}$ is the generalized velocity and it consists of the floating base generalized velocity $\dot{\mathbf{q}}_f \in \mathbb{R}^6$ and joint generalized velocity $\dot{\mathbf{q}}_r \in \mathbb{R}^6$. The foot contact velocity vector for all 4 feet can written as follows:

$$\dot{\mathbf{p}}_c = \begin{bmatrix} \dot{\mathbf{p}}_{c,1} \\ \dot{\mathbf{p}}_{c,2} \\ \dot{\mathbf{p}}_{c,3} \\ \dot{\mathbf{p}}_{c,4} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{c,1} \\ \mathbf{J}_{c,2} \\ \mathbf{J}_{c,3} \\ \mathbf{J}_{c,4} \end{bmatrix} \dot{\mathbf{q}} \tag{12}$$

$\mathbf{J}_{c,1} \in \mathbb{R}^{3\times18}$, $\mathbf{J}_{c,2} \in \mathbb{R}^{3\times18}$, $\mathbf{J}_{c,3} \in \mathbb{R}^{3\times18}$, and $\mathbf{J}_{c,4} \in \mathbb{R}^{3\times18}$, are the contact Jacobian matrices for left front leg (LF), right front leg (RF), right hind leg (RH), and left hing leg (LH), respectively. Suppose one or more legs are swinging $k \neq 12$. In this case, we can assign selection matrix $\mathbf{S} \in \mathbb{R}^{k\times12}$ with an appropriate size. The following relation provides the foot velocity as below:

$$\dot{\mathbf{p}}_c = \mathbf{S}\mathbf{J}_c\dot{\mathbf{q}} \qquad \text{For simplicity:} \qquad \bar{\mathbf{J}}_c = \mathbf{S}\mathbf{J}_c \tag{13}$$

We can assign the constraint kinematics condition $\dot{\mathbf{p}}_{\mathbf{c}} = \mathbf{0}$ to obtain floating base generalized velocity as below:

$$\dot{\mathbf{p}}_c = \begin{bmatrix} \bar{\mathbf{J}}_{cf} & \bar{\mathbf{J}}_{cr} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_f \\ \dot{\mathbf{q}}_r \end{bmatrix} = \bar{\mathbf{J}}_{cf}\dot{\mathbf{q}}_f + \bar{\mathbf{J}}_{cr}\dot{\mathbf{q}}_r = \mathbf{0} \tag{14}$$

$$\dot{\mathbf{q}}_f = -\bar{\mathbf{J}}_{cf}^{\dagger}\bar{\mathbf{J}}_{cr}\dot{\mathbf{q}}_r \tag{15}$$

where $\dagger$ represents the Moore-Penrose Pseudo Inverse. $\bar{\mathbf{J}}_{cf} \in \mathbb{R}^{k\times6}$ and $\bar{\mathbf{J}}_{cr} \in \mathbb{R}^{k\times12}$ are the Jacobian matrices that correspond to floating-base and robot joint velocities, respectively.

2.2.4 Operational Space Kinematics

Operational space kinematics were yielded utilizing the constraint and task velocity kinematics. The task velocity can be defined as below:

$$\dot{\mathbf{p}}_t = \mathbf{J}_t\dot{\mathbf{q}} \tag{16}$$

where $\dot{\mathbf{p}}_t \in \mathbb{R}^6$ includes CoM velocity and the torso angular velocity. $\mathbf{J}_t \in \mathbb{R}^{6\times18}$ is the task Jacobian matrix. We can build the operational space kinematics with the augmented form, including the constraint kinematics and the task kinematics as follows:

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{\mathbf{p}}_c \\ \dot{\mathbf{p}}_t \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{J}}_c \\ \mathbf{J}_t \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \bar{\mathbf{J}}_{cf} & \bar{\mathbf{J}}_{cr} \\ \mathbf{J}_{tf} & \mathbf{J}_{tr} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_f \\ \dot{\mathbf{q}}_r \end{bmatrix} \tag{17}$$

where $\mathbf{J}_{tf} \in \mathbb{R}^{l\times6}$ and $\mathbf{J}_{tr} \in \mathbb{R}^{l\times12}$ are the task Jacobian matrices that correspond the floating-base and robot joint velocities, respectively. The generalized velocity can be obtained as follows:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{p}} \tag{18}$$

The Eq. (18) allows us to achieve mapping from the operational space to joint space when the robot satisfies constraint forces. If the $\dot{\mathbf{q}}$ is not fully controllable, the controllable states can obtained as follows:

$$\dot{\mathbf{q}}_r = [\mathbf{0}_{6\times1} \quad \mathbf{I}_{12\times1}]\mathbf{J}^{\dagger}\dot{\mathbf{p}} \tag{19}$$

Utilizing the constructed operational space kinematics in Eqs. (18) and (19), we can obtain the generalized inverse kinematic expression as follows:

$$\dot{\mathbf{q}} = \mathbf{J}^{\intercal}\dot{\mathbf{p}} \quad + \quad \underbrace{(\mathbf{I} - \mathbf{J}^{\intercal}\mathbf{J})\dot{\mathbf{p}}_0}_{\text{Null Space Movement}} \tag{20}$$

Moreover, we can designate the controller by utilizing the constraint kinematics in Eq. (15) as below:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_f \\ \dot{\mathbf{q}}_r \end{bmatrix} = \begin{bmatrix} -\bar{\mathbf{J}}_{cf}\bar{\mathbf{J}}_{cr}\dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_r \end{bmatrix} = \begin{bmatrix} -\bar{\mathbf{J}}_{cf}\bar{\mathbf{J}}_{cr} \\ \mathbf{I} \end{bmatrix} \dot{\mathbf{q}}_r \tag{21}$$

2.3 Floating-base Dynamics

The floating-base dynamics is crucial for generating controllers that provide dynamically coherent locomotion. The inverse dynamics-based controller and CMM-based controllers can be obtained through the floating-base dynamics. The floating base dynamics can be defined as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{S}^{\intercal}\boldsymbol{\tau} + \bar{\mathbf{J}}_c^{\intercal}\mathbf{F}_{con} \tag{22}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{18\times18}$ is the joint-space inertia matrix. $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{18\times18}$ and $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{18}$ stand for the Coriolis matrix and gravity vector, respectively. $\mathbf{S} = [\mathbf{0}_{12\times6} \quad \mathbf{I}_{12\times6}]$ is the selection matrix to acquire torque from the actuated joints. $\mathbf{F}_{con} \in \mathbb{R}^{3nc}$ represents the contact forces.

2.3.1 Recursive Newton-Euler Algorithm

We implemented the Recursive Newton-Euler algorithm to construct an inverse dynamics controller [24, 25]. This algorithm iteratively computes Newton-Euler's equations of motion for each robot link. This recursive algorithm is separated into two phases: outward and inward iterations. In the outward iterations, the algorithm recursively computes the linear and angular accelerations, along with the angular velocity of each CoM of the link. The recursion starts from link 0 up to link n by computing angular velocity and linear and angular accelerations for each link. For joint $i + 1$, the angular velocity and acceleration are computed based on the corresponding values from the previous link.

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}\mathbf{R}\,^{i}\omega_i + \dot{\mathbf{q}}_{i+1} \tag{23}$$

$^{i+1}_{i}\mathbf{R}$ shows the rotation matrix from the i-th frame to the $i + 1$-th frame. $^{i+1}\omega_{i+1}$ represents the angular velocity of link $i + 1$. $^{i}\omega_i$ represents the angular velocity of the previous link (i). The joint velocity, which is the parent of the link ($i + 1$), is represented by $\dot{\mathbf{q}}_{i+1}$. The angular acceleration $^{i}\dot{\omega}_i$ of the link can be calculated as below:

$$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}\mathbf{R}\,^{i}\dot{\omega}_i + {}^{i+1}_{i}\mathbf{R}\,^{i}\omega_i \times \dot{\mathbf{q}}_{i+1} + \ddot{\mathbf{q}}_{i+1} \tag{24}$$

where $\ddot{\mathbf{q}}_{i+1}$ is the parent link joint velocity. Moreover, To compute the linear acceleration $^{i+1}\dot{\mathbf{v}}_{i+1}$ for the frame at the parent joint of link $i+1$, we can utilize the following expression.

$$ {}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}_{i}\mathbf{R}[{}^{i}\omega_i \times {}^{i}\mathbf{p}_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}\mathbf{p}_{i+1}) + {}^{i}\dot{\mathbf{v}}_i] \tag{25} $$

In (25) ${}^{i}\dot{\mathbf{v}}_i$ and $\mathbf{p}_{i+1}$ are the linear acceleration of the previous link and the joint position vector of the frame $i+1$ with respect to i. The algorithm also requires the computation of the CoM acceleration ${}^{i}\dot{\mathbf{v}}_{c_i}$ and it can computed as below:

$$ {}^{i}\dot{\mathbf{v}}_{c_i} = {}^{i}\dot{\omega}_i \times {}^{i}\mathbf{p}_{c_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}\mathbf{p}_{c_i}) + {}^{i}\dot{\mathbf{v}}_i \tag{26} $$

where ${}^{i}\mathbf{p}_{c_i}$ is the CoM position vector of the link i with respect to joint frame i. Through the obtained kinematic variables from the Eqs. (23), (24), (25), and (26), we can obtain the force and torques on each link via Newton-Euler equations as follows:

$$ \mathbf{F}_i = m_i\,{}^{i}\dot{\mathbf{v}}_{c_i} \tag{27} $$

$$ \mathbf{N}_i = {}^{c_i}\mathbf{I}\,{}^{i}\dot{\omega}_i + {}^{i}w_i \times {}^{c_i}\mathbf{I}\,{}^{i}\omega_i \tag{28} $$

The forces acting on link i's CoM are denoted by $\mathbf{F}_i$, and the total mass of the i^{th} link is indicated as m_i. The torques acting on the i^{th} link are denoted by $\mathbf{N}_i$, whereas the inertia tensor of link i is denoted by ${}^{c_i}\mathbf{I}$. The joint torques are calculated using inward iterations once the forces and torques operating on each link have been calculated. Force-balance and moment-balance equations are obtained for a typical link as follows in order to determine the joint torques that will result in the net forces and torques on each link:

$$ {}^{i}\mathbf{f}_i = {}^{i}\mathbf{F}_i + {}^{i}_{i+1}\mathbf{R}\,{}^{i+1}\mathbf{f}_i \tag{29} $$

$$ {}^{i}\mathbf{n}_i = {}^{i}\mathbf{N}_i + {}^{i}_{i+1}\mathbf{R}\,{}^{i+1}\mathbf{n}_{i+1} + {}^{i}\mathbf{p}_{c_i} \times {}^{i}\mathbf{F}_i + {}^{i}\mathbf{p}_{i+1} \times ({}^{i}_{i+1}\mathbf{R}\,{}^{i+1}\mathbf{f}_{i+1}) \tag{30} $$

Joint forces and torques are represented as ${}^{i}\mathbf{F}i$ and ${}^{i}\mathbf{N}_i$, respectively. The Newton-Euler algorithm is formulated for a single leg of the quadruped robot and is then applied to each leg individually. The RNEA for one leg is given in Table 2.

Table 2: The Recursive Newton-Euler algorithm for the one leg of the quadruped robot

RNE Algorithm
`// Initialize floating-base`
$\omega_0 = \mathbf{R}_{fb}^{\intercal} \omega_{fb}$
$\dot{\omega}_0 = \mathbf{R}_{fb}^{\intercal} \dot{\omega}_{fb}$
$\dot{\mathbf{v}}_0 = \mathbf{R}_{fb}^{\intercal} (\dot{\mathbf{v}}_{fb} - \mathbf{g})$
`// Outward Iteration`
for $i = 0$ **to** 4 **do**
$^{i+1}\omega_{i+1} = {}^{i+1}_{i}\mathbf{R}\, {}^{i}\omega_i + \dot{\mathbf{q}}_{i+1}$
$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}\mathbf{R}\, {}^{i}\dot{\omega}_i + {}^{i+1}_{i}\mathbf{R}\, {}^{i}\omega_i \times \dot{\mathbf{q}}_{i+1} + \ddot{\mathbf{q}}_{i+1}$
$^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}_{i}\mathbf{R}[{}^{i}\omega_i \times {}^{i}\mathbf{p}_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}\mathbf{p}_{i+1}) + {}^{i}\dot{\mathbf{v}}_i]$
$^{i}\dot{\mathbf{v}}_{c_i} = {}^{i}\dot{\omega}_i \times {}^{i}\mathbf{p}_{c_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}\mathbf{p}_{c_i}) + {}^{i}\dot{\mathbf{v}}_i$
$\mathbf{F}_i = m_i\, {}^{i}\dot{\mathbf{v}}_{c_i}$
$\mathbf{N}_i = {}^{c_i}\mathbf{I}\, {}^{i}\dot{\omega}_i + {}^{i}w_i \times {}^{c_i}\mathbf{I}\, {}^{i}\omega_i$
end
`// Initialize Contact Force`
$\mathbf{f}_5 = -{}^{5}_{0}\mathbf{R}\, \mathbf{f}_{con}$
`// Inward Iteration`
for $i = 4$ **to** 1 **do**
$^{i}\mathbf{f}_i = {}^{i}\mathbf{F}_i + {}_{i+1}^{i}\mathbf{R}\, {}^{i+1}\mathbf{f}_i$
$^{i}\mathbf{n}_i = {}^{i}\mathbf{N}_i + {}_{i+1}^{i}\mathbf{R}\, {}^{i+1}\mathbf{n}_{i+1} + {}^{i}\mathbf{p}_{c_i} \times {}^{i}\mathbf{F}_i + {}^{i}\mathbf{p}_{i+1} \times ({}_{i+1}^{i}\mathbf{R}\, {}^{i+1}\mathbf{f}_{i+1})$
end
`// Select Torques`
for $i = 1$ **to** 3 **do**
$\tau_{\mathbf{i}} = \mathbf{n}_i$
end

2.4 Centroidal Momentum

Centroidal momentum in a rigid-body system refers to the aggregated momentum of all its bodies [26]. Linear and angular momenta of each body can be represented using the 6D spatial vector notations to provide compact notation as follows:

$$\mathbf{h}_i = \begin{bmatrix} \mathbf{l}_i \\ \mathbf{k}_i \end{bmatrix} = \mathbf{I}_i \mathbf{v}_i = \mathbf{I}_i \begin{bmatrix} \mathbf{v}_i \\ \omega_i \end{bmatrix} \tag{31}$$

In (31) $\mathbf{h}_i \in \mathbb{R}^6$ is the spatial momentum of each body, $\mathbf{l}_i \in \mathbb{R}^3$ and $\mathbf{k}_i \in \mathbb{R}^3$ are its angular and linear momentum relative to a local coordinate origin, respectively.

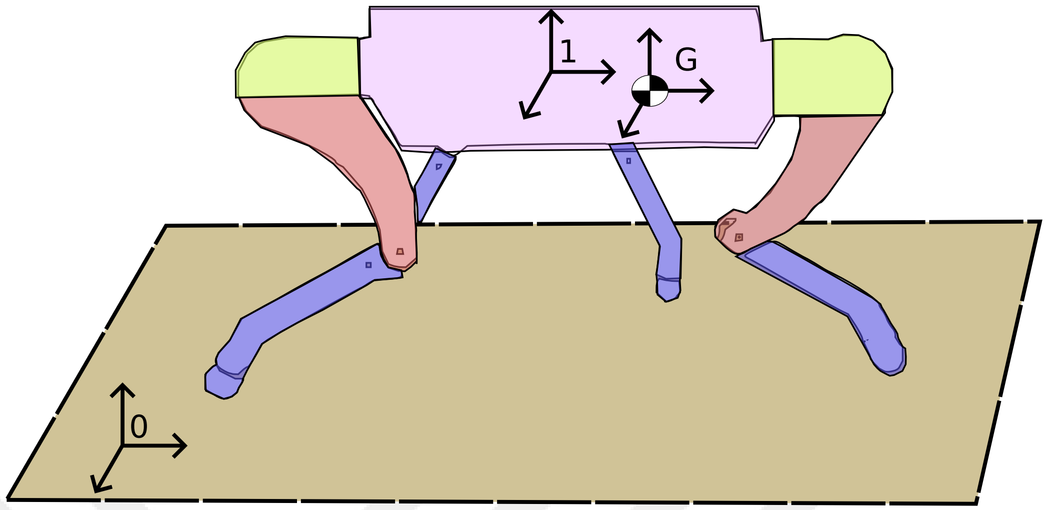

Figure 4: Coordinate System Definitions

Spatial inertia of the each body indicated as $\mathbf{I}_i \in \mathbb{R}^{6\times6}$ and its form as follows:

$$\mathbf{I}_i = \begin{bmatrix} m_i \mathbf{1}_{3\times3} & m_i \mathbf{c}_i \times \\ m_i \mathbf{c}_i \times & \bar{\mathbf{I}}_i \end{bmatrix} \tag{32}$$

where $\mathbf{c}_i \times \in \mathbb{R}^{3\times3}$ is skew symmetric form of the CoM for body i. $\bar{\mathbf{I}}_i \in \mathbb{R}^{3\times3}$ is the Cartesian inertia tensor of the body i.

The Fig. 4 shows the coordinate system definitions for the following derivations. The inertial coordinate system (ICS) 0 is fixed to the Earth. The floating-base coordinate system 1 is attached to the torso, and the AHRS is placed here. G represents the CoM coordinate system and matches the orientation of the ICS. The rigid bodies are the periwinkle, medium salmon, light-yellow green, and lilac colors.

The centroidal momentum is the superposition of the individual link's spatial momenta as given below:

$$\mathbf{h}_G = \sum_{i=1}^{N_b} {}^{i}\mathbf{X}_G^{\intercal} \mathbf{h}_i \tag{33}$$

where ${}^{i}\mathbf{X}_G^{\intercal}$ is the spatial transformation matrix that provides spatial momentum transformation from the body frame to the CoM frame (G). The spatial coordinate transformation provides various orientations of the links and relates the CoM position offsets of the links as following form:

$$ {}^{i}\mathbf{X}_{G}^{\intercal} = \begin{bmatrix} {}^{G}\mathbf{R}_{i} & \mathbf{0} \\ {}^{G}\mathbf{R}_{i}{}^{i}\mathbf{p}_{G}\times^{\intercal} & {}^{G}\mathbf{R}_{i} \end{bmatrix} \tag{34} $$

where the ${}^{G}\mathbf{R}_{i} \in \mathbb{R}^{3\times 3}$ and ${}^{i}\mathbf{p}_{G}\times \in \mathbb{R}^{3\times 3}$ are the rotation matrix from i frame to G frame and the skew-symmetric form of the position vector. The centroidal momentum computation can be achieved by utilizing the floating base system CMM as follows:

$$ \mathbf{h}_{G} = \mathbf{A}_{G}(\mathbf{q})\dot{\mathbf{q}} \tag{35} $$

where $\mathbf{A}_{G} \in \mathbb{R}^{6\times 18}$ and $\dot{\mathbf{q}} \in \mathbb{R}^{18}$ are the CMM and generalized velocity, respectively. In the CMM model, it is assumed that forces are projected from the joint space, and the effect of gravity is not included. These conditions indicate that external forces are the sole effect on the system. In this sense, The projected forces can be calculated as follows:

$$ \tau_{1} = \mathbf{\Phi}_{1}^{\intercal}\mathbf{f}_{1} = \mathbb{U}_{1}(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}) \tag{36} $$

where $\mathbf{\Phi}_{1}^{\intercal} \in \mathbb{R}^{6\times 6}$ is a matrix that provides a transformation from a world frame to a floating-base. $\mathbf{f}_{1} \in \mathbb{R}^{6}$ is the net external wrench. $\mathbb{U}_{1} = [\mathbf{1}_{6\times 6}\ \mathbf{0}_{6\times 12}]$ is the selection matrix. We can define $\mathbf{\Psi}_{1} = \mathbf{\Phi}_{1}^{-1}$ to extract the net external wrench on the floating-base frame as follows:

$$ \mathbf{f}_{1} = \mathbf{\Psi}_{1}^{\intercal}\mathbb{U}_{1}(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}) \tag{37} $$

The net external wrench in the CoM frame can be obtained with spatial transform ${}^{1}\mathbf{X}_{G}^{\intercal}$ as below:

$$ \begin{aligned} \mathbf{f}_{G} &= {}^{1}\mathbf{X}_{G}^{\intercal}\mathbf{f}_{1} \\ &= {}^{1}\mathbf{X}_{G}^{\intercal}\mathbf{\Psi}_{1}^{\intercal}\mathbb{U}_{1}(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}) \\ &= \mathbf{A}_{G}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{A}}_{G}(\mathbf{q})\dot{\mathbf{q}} \end{aligned} \tag{38} $$

The eq. (38) reveals the computation recipe for the $\mathbf{A}_G$ and bias force $\dot{\mathbf{A}}_G(\mathbf{q})\dot{\mathbf{q}}$. The computations can be defined as below.

$$\begin{aligned} \mathbf{A}_G(\mathbf{q}) &= {}^1\mathbf{X}_G^{\intercal}\mathbf{\Psi}_1^{\intercal}\mathbb{U}_1\mathbf{M}(\mathbf{q}) \\ \dot{\mathbf{A}}_G(\mathbf{q})\dot{\mathbf{q}} &= {}^1\mathbf{X}_G^{\intercal}\mathbf{\Psi}_1^{\intercal}\mathbb{U}_1\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \end{aligned} \tag{39}$$

CHAPTER III

TRAJECTORY GENERATION

3.1 Preview Control-Based Trajectory Generation

This section explains a preview control based trajectory generation approach of the quadruped robot in detail. Preview control is a type of servomechanism to tackle the output tracking problem [27]. It includes LQR, LQI, and preview action based that controllers can be defined as a performance index to regulate and provide optimal behavior for dynamic systems. In this sense, Preview control is utilized for ZMP-based CoM trajectory generation of quadruped robots.

3.1.1 Walking Motion Generation

The CoM trajectory base model is 3D LIP and ZMP concept as in [1, 28, 29]. The equation for the ZMP along the x-axis is as follows:

$$p_x = c_x - \frac{\ddot{c}_x}{\ddot{c}_z + g} c_z - \cancelto{\approx 0}{\frac{\dot{L}_y}{m(\ddot{c}_z + g)}} \tag{40}$$

where, c_x and c_z represent the position of CoM, $\dot{L}_y$ denotes the moment around the y-axis, m is the total mass, p_x signifies ZMP on the x-axis, and g stands for the acceleration due to gravity. To illustrate these simplified dynamics, we can employ a cart-bowl model that signifies c_z as in Fig. 5 [1]. Since all the actuators are positioned around the torso, the robot's legs are relatively lightweight. Consequently, the rate change in angular momentum ($\dot{L}_y$) is negligible during motions where the torso's orientation is approaching zero. As a result of this negligible effect, we express the natural frequency ω and the acceleration of the CoM ($\ddot{c}_x$) as follows:

$$\omega = \sqrt{\frac{\ddot{c}_z + g}{c_z}} \tag{41}$$

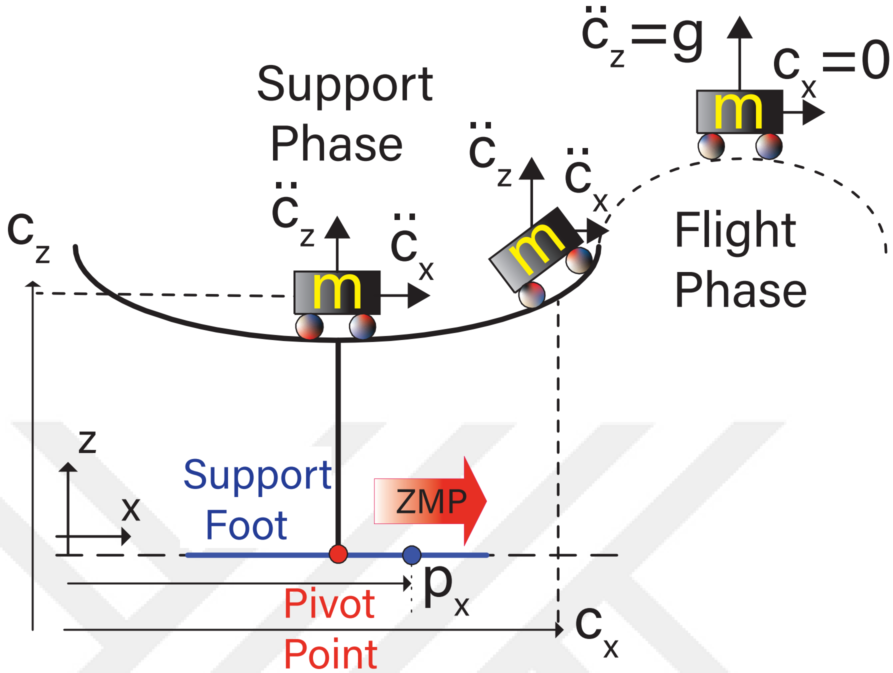

Figure 5: Cart-Bowl Model [1]

$$\ddot{c}_x = \omega^2(c_x - p_x) \tag{42}$$

The Eqs. (41) and (42) can be utilized to produce the trajectory of the CoM in the x-axis. For output tracking problem and constant CoM height, the (42) can be rewritten as below:

$$p_x = c_x - \ddot{c}_x \frac{c_z}{g}. \tag{43}$$

Let us define the ZMP servo problem with jerk control as in [7]. The following state-space model represents the ZMP dynamics for the horizontal motion of CoM in equation (43) with controlled jerk input.

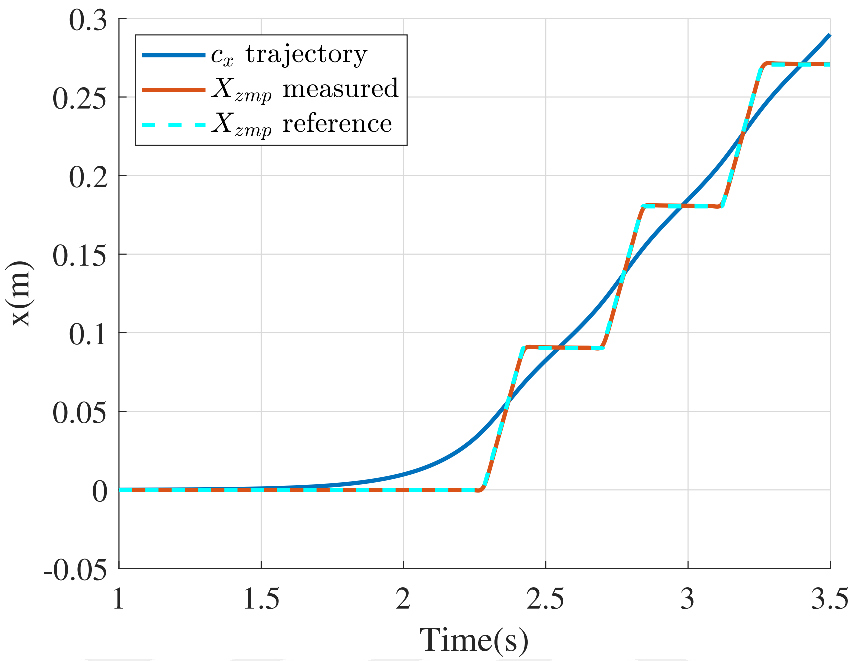

Figure 6: ZMP and CoM trajectory via Preview Control

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ \dot{c}_x \\ \ddot{c}_x \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{44a}$$

$$y = \begin{bmatrix} 1 & 0 & -\frac{c_z}{g} \end{bmatrix} \begin{bmatrix} c_x \\ \dot{c}_x \\ \ddot{c}_x \end{bmatrix} \tag{44b}$$

$$\text{where } x = \begin{bmatrix} c_x & \dot{c}_x & \ddot{c}_x \end{bmatrix}^\intercal \tag{44c}$$

$$u = \dddot{c}_x \tag{44d}$$

$$y = p_x \tag{44e}$$

The state vector x encompasses the linear aspects of CoM along the x-axis. At the same time, $\dddot{c}_x$ represents the jerk input and is the time derivative of the CoM's horizontal acceleration. When the state space components in equations (44) with reference ZMP trajectory utilized in preview control algorithm results in Fig. 6 can be obtained. In Fig. 44, blue, red, and cyan lines show the CoM trajectory c_x, measured ZMP in the x-axis and reference ZMP in the x-axis c_x, respectively. As seen in Fig 6 c_x reacts earlier than the reference ZMP trajectory thanks to the preview action mechanism of the preview control.

3.1.2 Jumping Motion Generation

The trajectory of the CoM for the quadruped robot in the z-axis is determined using 3D LIPM with varying height [30], employing the equations (41) and (42). The (42) can be reformulated to represent the z-axis acceleration of the CoM in the following manner:

$$\ddot{c}_z = \omega^2 c_z - g \tag{45}$$

The (41) can be reformulated as detailed in [7]. This method allows us to construct the output equation for (42). The resultant equation is provided below.

$$\frac{g}{\omega^2} = c_z - \frac{\ddot{c}_z}{\omega^2} \tag{46}$$

In (46), $\frac{g}{\omega^2}$ can be designated as the output, similar to the ZMP approach described in [7]. This output can be constructed to achieve the flight phase. The pendulum's natural frequency ω can be tuned to achieve a viable CoM trajectory for jumping motion. Typically, jumping is divided into two phases, as demonstrated in Fig. 5: the support and flight phases. The LIP dynamics during the support phase are described in (41), while the flight phase is generally described utilizing projectile motion dynamics. However, our trajectory generation technique uses only LIP dynamics to represent support and flight phases. The discrete state space representation of the vertical trajectory generator is given below:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] \end{aligned} \tag{47}$$

where

$$x[k] = \begin{bmatrix} c_z & \dot{c}_z & \ddot{c}_z \end{bmatrix}^{\intercal} \quad A = \begin{bmatrix} 0 & T & \frac{T^2}{2} \\ 0 & 0 & T \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & -\frac{1}{w^2} \end{bmatrix}$$
$$u[k] = u_z \quad y[k] = \frac{g}{w^2} \tag{48}$$

The preview control regards tracking the output. Thus, we use the desired output as a reference trajectory for the controller, which, in our case, $\frac{g}{\omega^2}$. Moreover, the discrete states and outputs of the discrete state space can be written as the objective function of the preview controller, as follows:

$$J = \sum_{k}^{\infty} \{Q_e e[k]^2 + \Delta x^T[k] Q_x \Delta x[k] + R\Delta u[k]^2\} \tag{49a}$$

$$\text{Subject to} : \bar{x}[k+1] = \bar{A}\bar{x}[k] + \bar{B}u[k] \tag{49b}$$

$$\text{where } \bar{x}[k] = \begin{bmatrix} e^{\intercal}[k] & \Delta x^{\intercal}[k] & x_d{}^{\intercal}[k] \end{bmatrix}^{\intercal} \tag{50}$$

The variation of the states is shown as $\Delta x[k] = x[k] - x[k-1]$, while the variation in control input is represented as $\Delta u[k] = u[k] - u[k-1]$. The output error is expressed as $e[k] = y[k] - y^{ref}[k]$. The positive semi-definite matrices Q_e and Q_x mean the servo error and state variation weights, respectively. The matrix R is positive-definite and defines the weight for the control input. To minimize equation (49a) under dynamic constraints, employing a subject-to-function incorporating the augmented state space model (49b) is necessary. This model contains future information regarding the desired trajectory $x_d[k]$, output error, and state variation [27]. To calculate the optimal gains of the preview controller subject, the minimizes the (49a) that utilizes the augmented state space, and we focused on the following form for simplicity as in [27].

The augmented state space components are

$$\tilde{A} = \begin{bmatrix} I & CA \\ 0 & A \end{bmatrix} \qquad \tilde{B} = \begin{bmatrix} CB \\ B \end{bmatrix} \tag{51a}$$

$$\text{where } \tilde{x}[k] = \begin{bmatrix} e^{\intercal}[k] & \Delta x^{\intercal}[k] \end{bmatrix}^{\intercal} \tag{51b}$$

$\tilde{A}$ denotes the augmented system matrix, while $\tilde{B}$ is the augmented input matrix utilized to compute optimal gains. A, B, and C designate the system matrix, input matrix, and output matrix of the trajectory generator as specified in equation (47), respectively. The vector $\tilde{x}[k]$ contains the state variables. The input of the optimal controller can be expressed as follows:

$$\Delta u[k] = -K_i \sum_{i=0}^{k} e[i] - K_s \Delta x[k] + \sum_{l=1}^{N_L} K_d[l] \Delta y^{ref}[k+l] \tag{52}$$

$y^{ref}[k+l]$ contains future references, where N_L means the number of steps into the future. K_i and K_s represent the integral and optimal state feedback gains, respectively, which can be determined by solving the Discrete Algebraic Riccati Equation (DARE) for infinite and finite horizon Linear Quadratic Regulator (LQR). $K_d[l]$ represents the set of delay gains employed for regulating the system with future references, and these gains are computed using the finite horizon Linear Quadratic Regulator (LQR) solution. K_i and K_d are obtained through the Schur algorithm and the Hamiltonian matrix, yielded from the costate equation. The optimal dynamics are denoted by $\lambda^*[k]$ and $x^*[k]$ according to references [31, 32]. The following expression was utilized to obtain the Inverse of the Hamiltonian matrix $\mathcal{H}^{-1}$.

$$\begin{bmatrix} x^*[k+1] \\ \lambda^*[k+1] \end{bmatrix} = \mathcal{H}^{-1} \begin{bmatrix} x^*[k] \\ \lambda^*[k] \end{bmatrix} \tag{53}$$

where

$$\mathcal{H}^{-1} = \begin{bmatrix} \tilde{A} + \tilde{B}R^{-1}\tilde{B}^T\tilde{A}^{-T}Q & -\tilde{B}R^{-1}\tilde{B}^T\tilde{A}^{-T} \\ -\tilde{A}^{-1}\tilde{B}R^{-1}\tilde{B}^T & \tilde{A}^{-1} \end{bmatrix} \tag{54}$$

The weight matrix Q includes Q_x and Q_e diagonally as in (55). When examining the diagonalized eigenstructure of the inverse of the Hamiltonian matrix, we prioritize ordering eigenvalues and vectors, beginning from the stable ones. The related equations are given below:

$$Q = \begin{bmatrix} Q & 0 & 0 & 0 \\ 0 & Q_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{55}$$

$$\hat{\mathcal{H}}^{-1} = VDV^{-1} \tag{56}$$

$$\text{where } V = \left[\begin{array}{c|c} V_x & * \\ \hline V_\lambda & * \end{array}\right] \qquad D = \left[\begin{array}{c|c} \Lambda & * \\ \hline * & * \end{array}\right] \tag{57}$$

V_x and V_λ include the stable eigenvector blocks for state and costate, respectively, while Λ includes the stable eigenvalues in diagonal form. K_i and K_s can be computed utilizing these matrices and components of the state space as

$$S = V_\lambda \Lambda^\intercal {V_x}^{-1} \qquad \boldsymbol{K} = \begin{bmatrix} K_i \\ K_s \end{bmatrix} = R^{-1}\tilde{B}^\intercal S^\intercal \tag{58}$$

S represents the positive semi-definite solution of the DARE. By iterating DARE with the closed-loop system for a finite horizon, we can derive the K_d gains. The closed-loop system is as follows:

$$\tilde{A}_c = \tilde{A} - \tilde{B}\boldsymbol{K} \tag{59}$$

Solving DARE requires initial conditions to find K_d. Initial conditions can be obtained as below:

$$\tilde{X}[1] = -\tilde{A}_c^T S\tilde{I} \qquad K_d[1] = -K_i \tag{60}$$

$\tilde{I}$ defined as $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\intercal$. Utilizing the (60) we can solve DARE for K_d as below:

$$K_d[l] = [R + \tilde{B}^T S\tilde{B}]\tilde{B}^T\tilde{X}(l-1),\ l = 2, ..., N_L \tag{61}$$

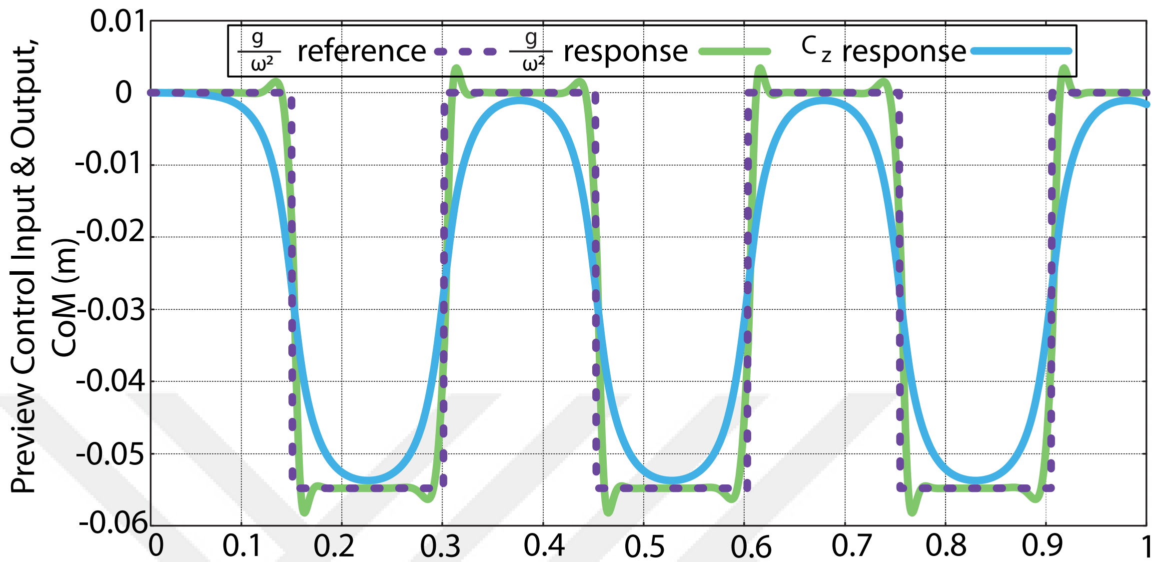

Figure 7: Numerical result of the trajectory generation algorithm

$$\tilde{X}[l] = \tilde{A}_c^T \tilde{X}[l-1] \tag{62}$$

K_d gain set can be computed by solving (61) and (62) recursively. If the given computation process for the proposed trajectory generation (47), numerical results in Fig.7 can be acquired. In Fig. 7, the purple dashed line indicates the reference of the preview control. The solid green line illustrates the output response of the preview control. In contrast, the solid turquoise line represents the resulting trajectory of the CoM in the vertical axis, denoted as c_z. As seen in Fig. 7, c_z responds before the reference $\frac{g}{\omega^2}$ trajectory thanks to the preview action mechanism of the preview control.

3.2 Locomotion Control Scheme

The Fig. 8 illustrates the locomotion control architecture of Kara for the vertical CoM trajectory generation approach. Within this framework, the g/w^2 reference block generates a reference signal, as illustrated in Figure 7, comprising three phases: i) flight phase, ii) transfer phase, and iii) support phase. These phases are characterized by associated durations: T_{fl}, T_{tr}, and T_{sup}, respectively, as shown in Figure 7. The preview control block uses the reference shown in Fig. 7 to generate the viable CoM (c_z) trajectory. The inverse kinematic block uses the preview control-based trajectory generation algorithm's computed vertical CoM trajectory to generate joint references (q_j^{ref}) for each leg. Leg numbers are indicated as the under script j. The superscript ref contains the reference values. The floating base orientation (R_t), generalized floating base accelerations ($\dot{q}_f$), and floating base angular velocities (w_t) are all included in the IMU. The PD (proportional-derivative) controller and inverse dynamics are involved in computed torque control. The inverse dynamics part of this control diagram used the Recursive Newton-Euler Algorithm (RNEA). RNEA requires IMU data, $\dot{q}_j$, and $\ddot{q}_j{}^{ref}$ to compute $\mathcal{T}_{ff}$. PD controller is the second part of the computed torque control that compensates for unmodelled dynamics in RNEA. The force constraints part uses the force sensor to measure the contact force (F_{con}) and force reference F^{ref} to compute the force command (F_{cmd}) for the RNEA. The diagonal gain matrices that represent feedforward and feedback are K_{ff} and K_{fb}, respectively. The force error between F^{ref} and F_{con} is represented by F_{err}. The control architecture's mathematical description can be expressed as follows.

$$\begin{aligned} \mathcal{T}_{cmd} = \;& M(q)\ddot{q} + C(q,\dot{q}) + G(q) + J^{\intercal}F_{cmd} + \\ & K_p q_{err} + K_v \dot{q}_{err} \end{aligned} \tag{63}$$

$$\text{where } F_{cmd} = K_{ff}F^{ref} + K_{fb}F_{err} \tag{64a}$$

$$F_{ref} = \begin{bmatrix} 0 & 0 & m(g+\ddot{c}_z)\frac{N_{cont}}{4} \end{bmatrix}^{\intercal} \tag{64b}$$

J and $M(q)$ are contact Jacobian and mass matrices respectively. The vectors for the Coriolis and gravity terms are $G(q)$ and $C(q,\dot{q})$, respectively. The gain matrices for PD controllers are K_p and K_v. N_{cont} indicates the number of contacts.

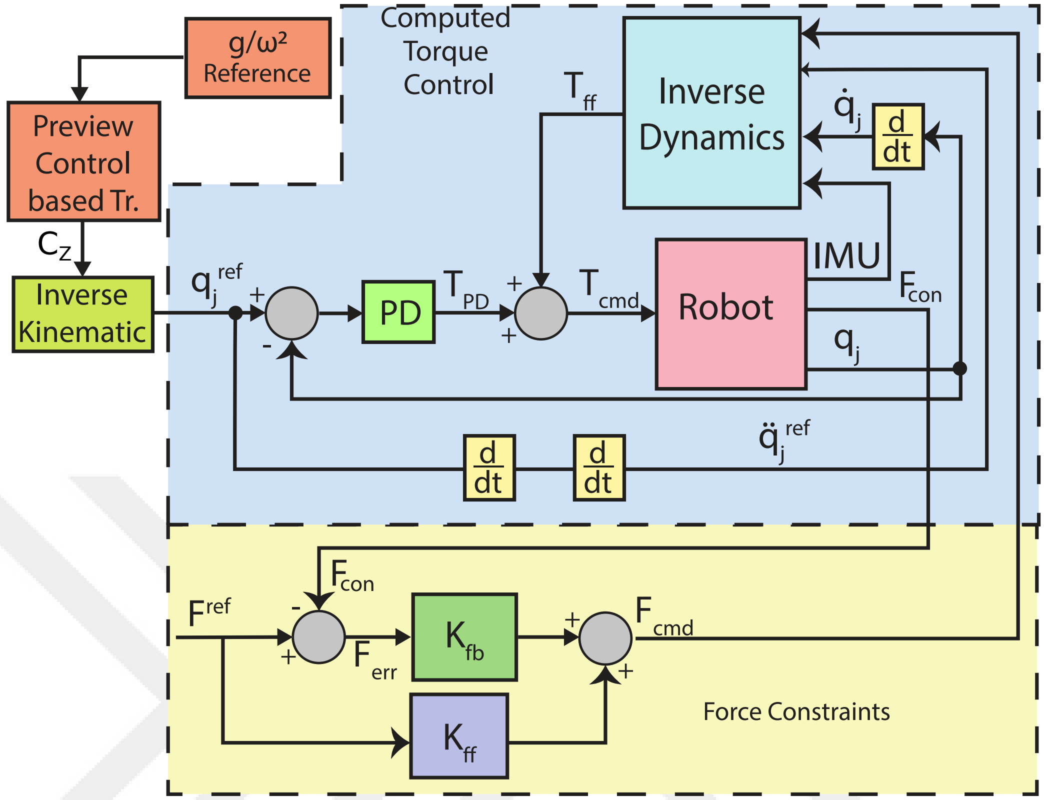

Figure 8: Overview of locomotion control architecture

3.3 Results

We conducted several simulations using the quadruped robot Kara dynamic model in the Raisim v1.1.7 simulation environment to figure out the performance of the presented trajectory generation algorithm. These simulations include two scenarios for the three-trajectory generation approach: spot-jogging and jumping. We compared our preview-control-based vertical CoM trajectory generation strategy with the polynomial-based trajectory [33] and the sinusoidal-based trajectory generation.

3.3.1 Forward Walking

In this scenario, the CoM trajectory is generated by the preview control approach as in [7]. The walking pattern was the trot-walking, and its phases were single support T_s and double support T_d as in [29]. T_s was 0.24 and T_d was 0.12 seconds. The robot was walked with 0.6 m/s CoM velocity. For trot walking, foot trajectories were generated by relating them to the generated CoM trajectory through the preview control. The trajectory in the x axis direction was a cubic spline, and foot clearance motion in the z-axis was generated with a sextic spline. In Fig. 9, the foot trajectories were represented with green and purple lines. The red dashed line represents the reference CoM trajectory, while the orange line indicates the actual CoM trajectory. The results show that the RMSE of the CoM trajectory is 0.77 [cm]. The local X_{zmp} and Y_{zmp} results were also given in Figs. 10 and 11. In light of these results, the peak-to-peak results are 23.2 [cm] for X_{zmp} and 7.63 [cm] for the Y_{zmp}.

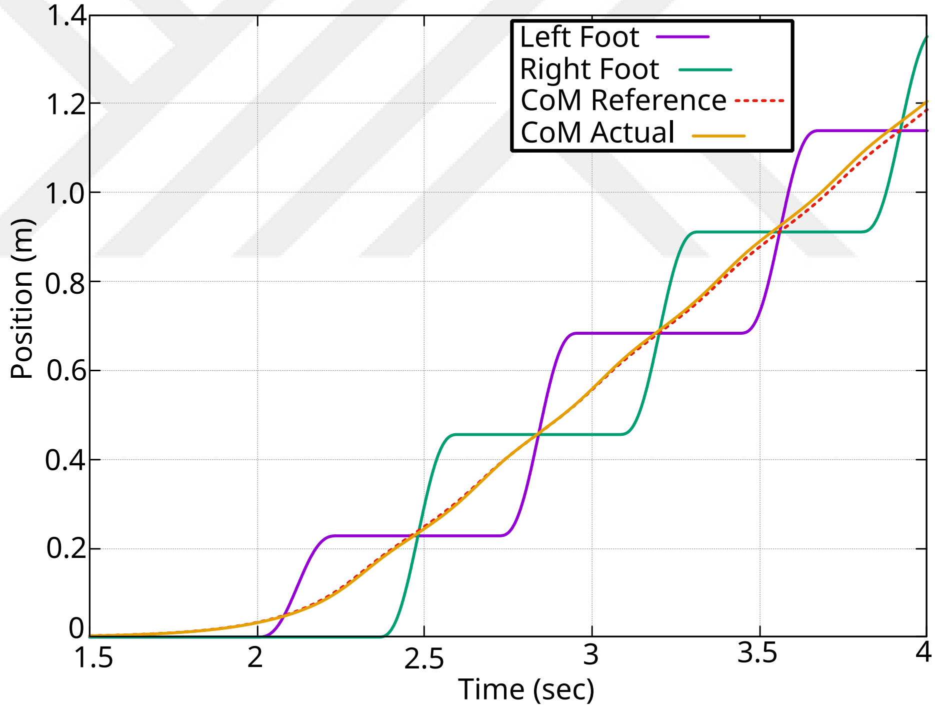

Figure 9: Walking simulation results: CoM and foot trajectories

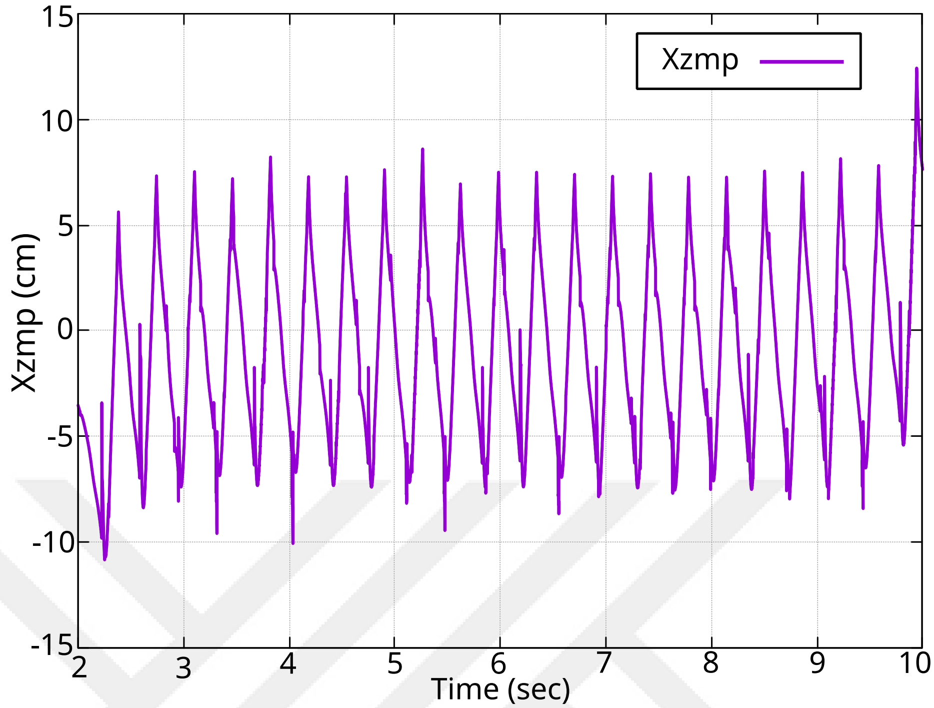

Figure 10: Walking simulation results: local X_{zmp}

3.3.2 Continuous Jumping

In this scenario, the quadruped robot Kara demonstrates continuous jumping, exhibiting trajectories based on preview control, polynomial, and sinusoidal. The frequency of the sinusoidal trajectory was manually adjusted to 6.5 Hz to provide the robot's jumping motion. The procedure of generating a polynomial trajectory is as follows [33]. Reference parameters for our preview control-based trajectory generation approach were T_{fl} and T_{sup} = 0.22 seconds, T_{tr} = 0.001 seconds, and ω = 28.5 Hz, g = 9.81 m/s^2, $R = 10^{-8}$, $Q_x = 1$, and $N_L = 1200$. Gains from force constraint: K_{ff} and K_{fb} were, respectively, 0.5 and 0.3. In Figs. 12, 13, 14, 15, the trajectory results based on sinusoidal, polynomial, and our preview control approaches are represented by the red, blue, and green colored lines, respectively. Moreover, ZMP is not defined throughout the flying phase; yellow blocks are placed over these data points. For each trajectory generator shown in Fig. 14, the root mean square (RMS) values of

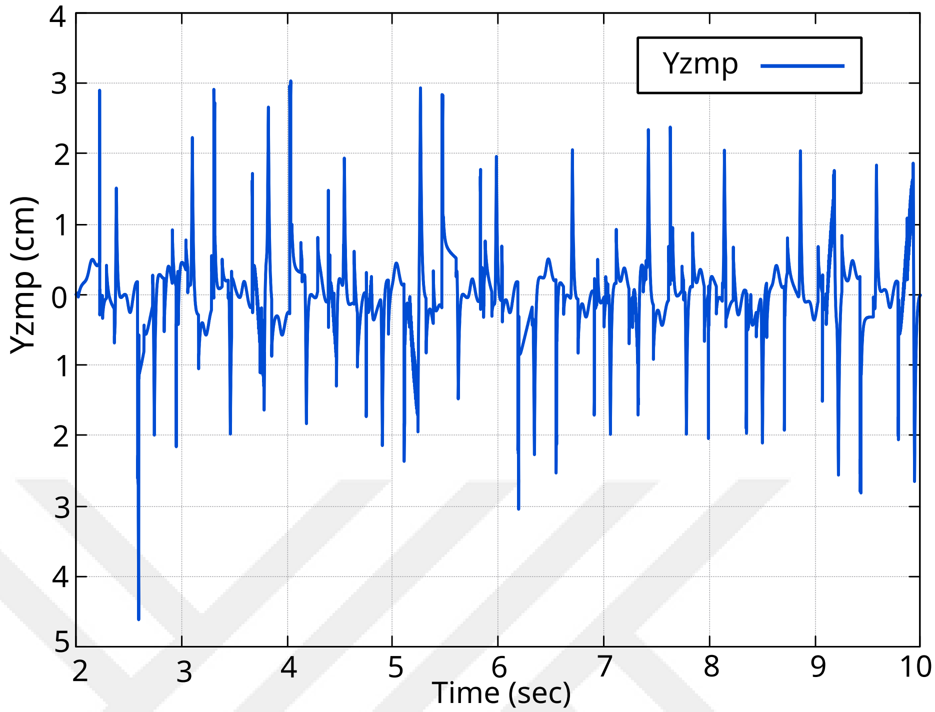

Figure 11: Walking simulation results: local Y_{zmp}

the torso pitch orientations are 0.2905 *deg* for sinusoidal, 0.1386 *deg* for polynomial, and 0.1008 *deg* for preview control. In Fig.15, The GRF peak-to-peak average for the polynomial trajectory is 7.51 [kN], the preview control trajectory is 2.81 [kN], and the sinusoidal trajectory is 3.0 [kN]. In Fig. 12 and Fig. 13, peak-to-peak average errors for the sinusoidal, polynomial, and proposed trajectory generators are, respectively, 2.96 [cm] and 1.70 m, 0.690 [cm] for X_{zmp}, and 0.31 [cm], 0.243 [cm] and 0.234 [cm] for Y_{zmp}.

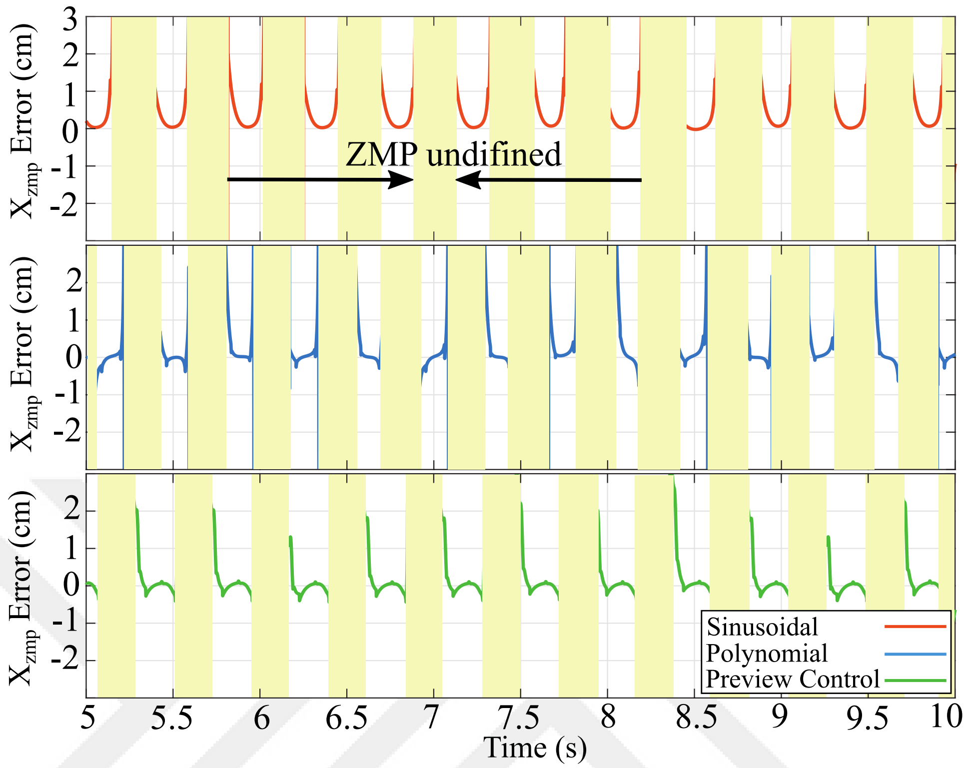

Figure 12: Continuous jumping simulation results: X_{zmp} errors

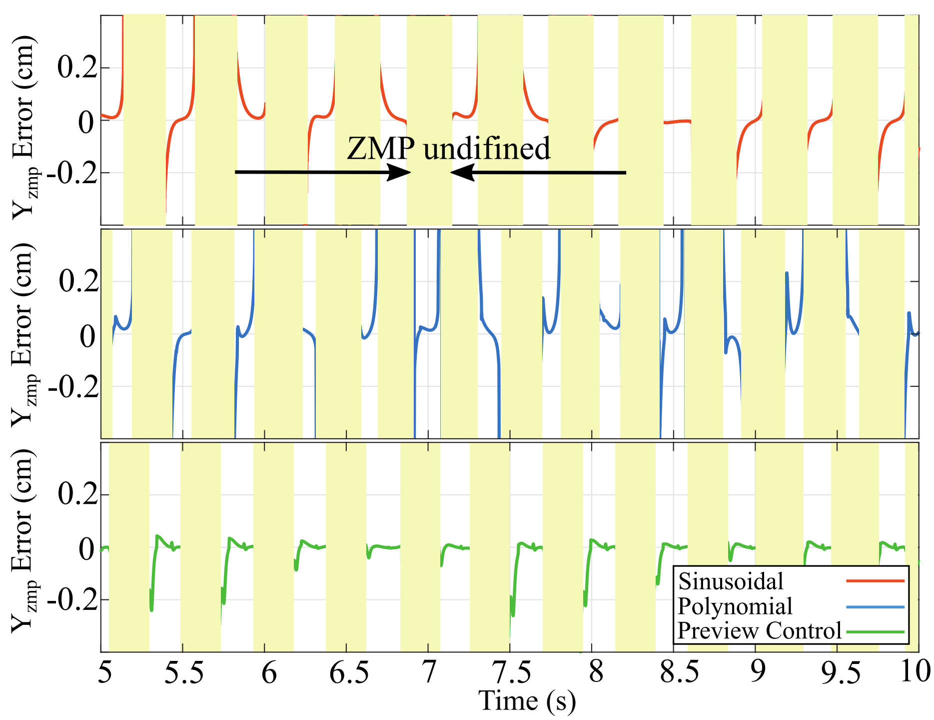

Figure 13: Continuous jumping simulation results: Y_{zmp} errors

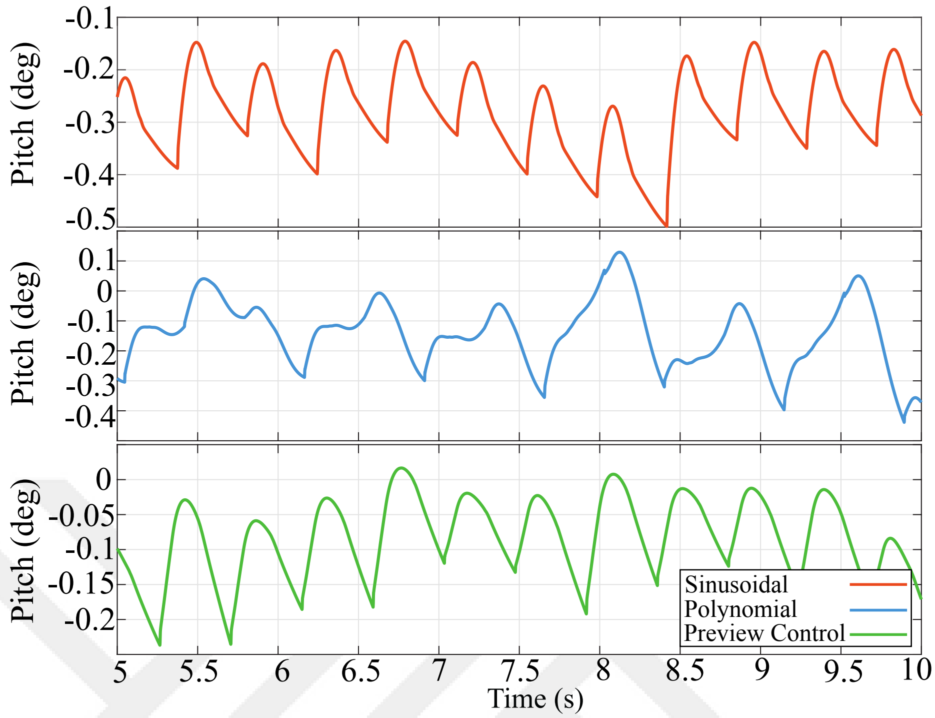

Figure 14: Continuous jumping simulation results: torso pitch orientations

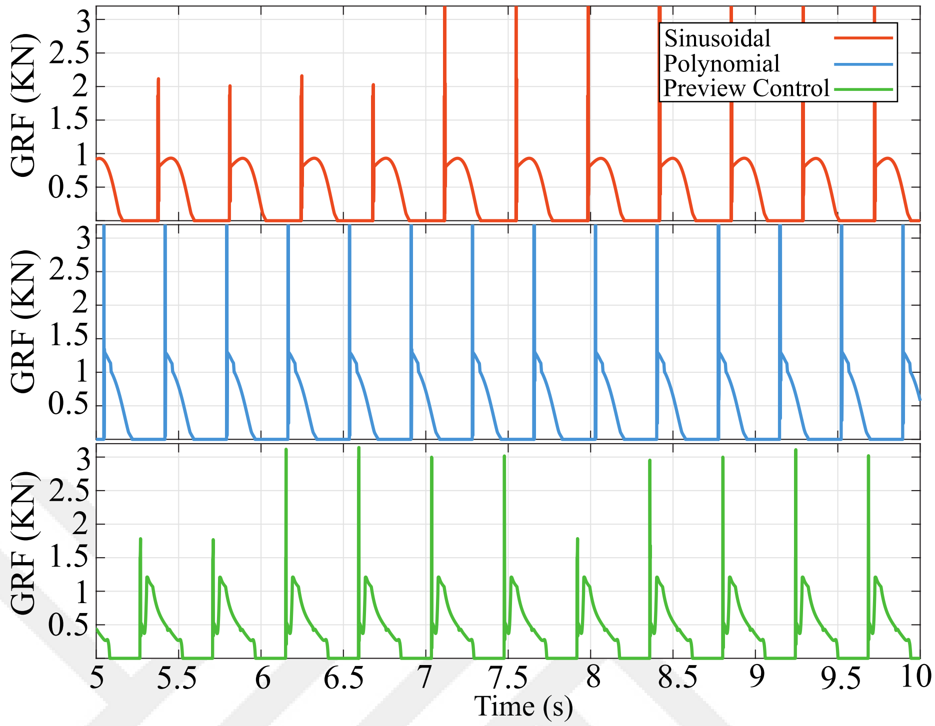

Figure 15: Continuous jumping simulation results: total ground reachtion forces for RF+LB foots

3.3.3 Spot Jogging

In this scenario, the quadruped robot has demonstrated spot-jogging using three different methods for generating trajectories: polynomial, sinusoidal, and preview control. Jogging requires retracting and switching diagonal legs during the flight phase, and it utilized CoM trajectories in the jumping case. In this scenario, the robot successfully retracted its feet and switched its diagonal feet using a quintic spline trajectory. The preview control-based trajectory generator's reference parameters were T_{fl} and T_{sup} at 0.15 seconds, T_{tr} at 0.001 seconds, and the control and system parameters were identical to those of scenario 1. The results of scenario 2 are displayed in Figures 16, 17, 18, and 19. The orientation for torso pitch angles for the sinusoidal, polynomial, and suggested trajectory generators, the RMS values are 7.8909 deg, 0.1393 deg, and 0.5393 deg, respectively, in Fig. 18. The robot nearly lost balance

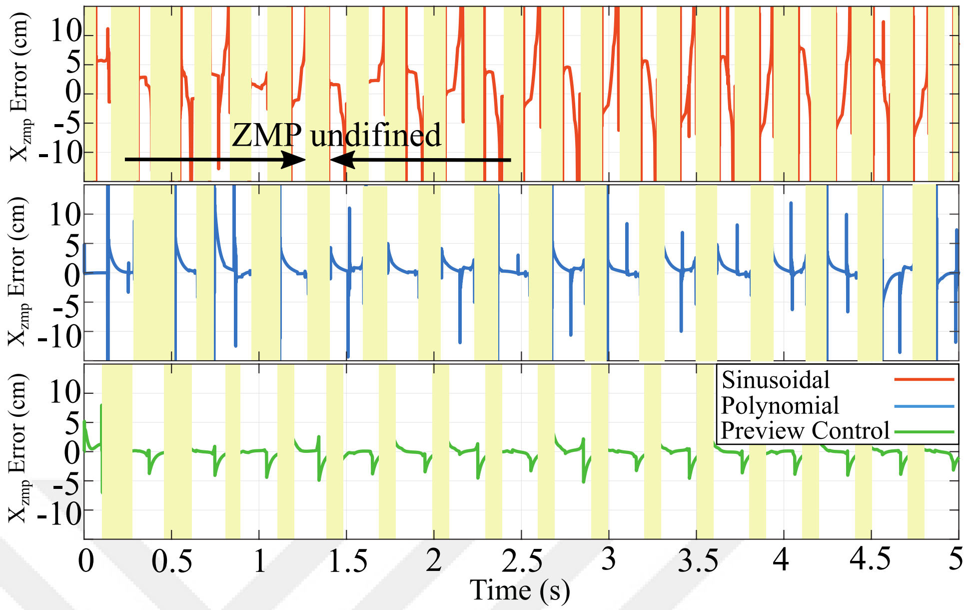

Figure 16: Spot jogging simulation results: X_{zmp} errors

and deviated from the reference point, which is why the sinusoidal trajectory RMS value is so high. The GRF peak-to-peak average for the sinusoidal, polynomial, and proposed trajectory generator, respectively, is 3.44 [kN], 2.65 [kN], and 2.1 [kN] for each trajectory. Peak-to-peak average errors of X_{zmp} are 9.32 [cm] and 5.74 [cm], 5.8 [cm] for sinusoidal, polynomial, and the proposed trajectory generator, respectively. Furthermore, Peak-to-peak average errors of Y_{zmp} are 1.81 [cm], 1.78 [cm], and 1.35 [cm] for sinusoidal, polynomial, and the proposed trajectory generator, respectively.

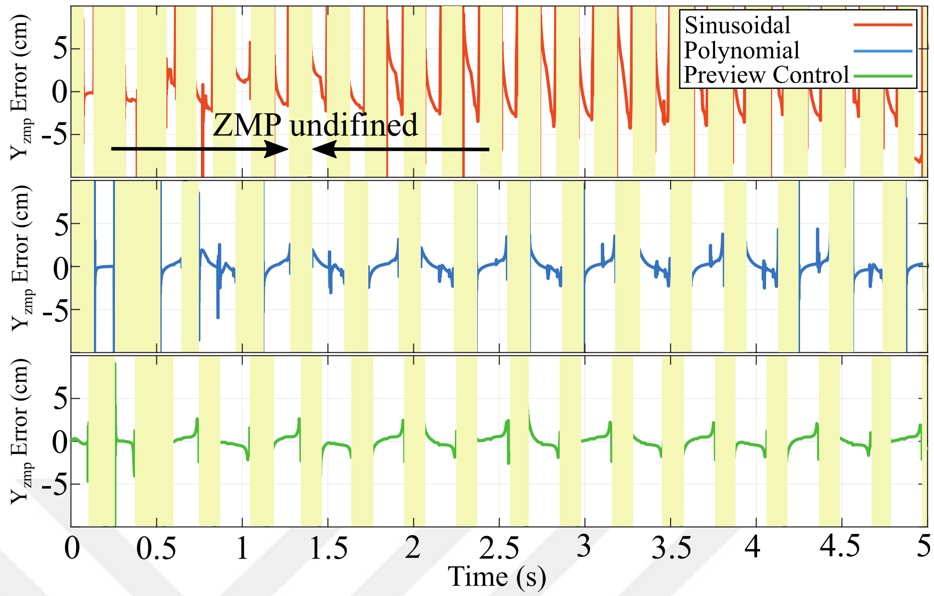

Figure 17: Spot jogging simulation results: Y_{zmp} errors

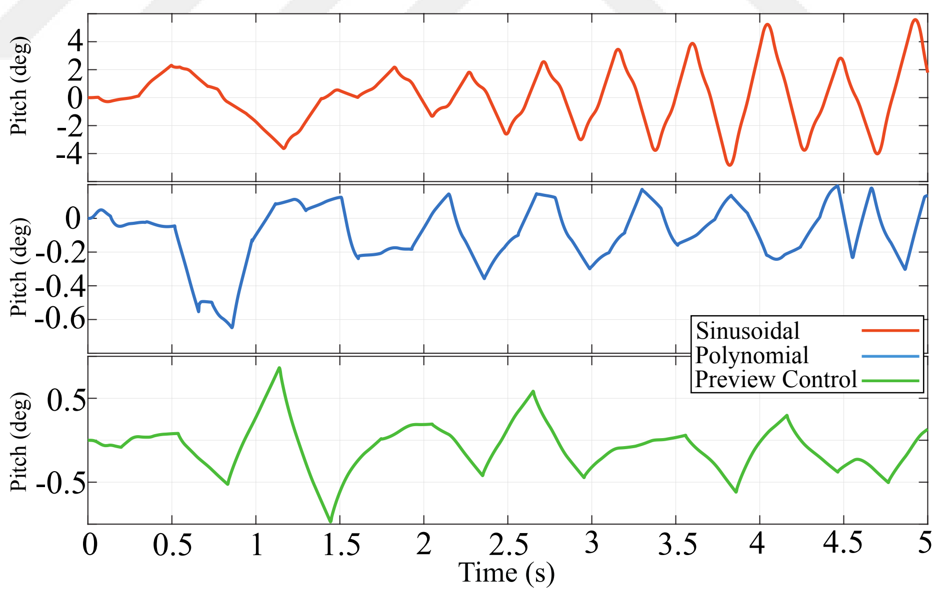

Figure 18: Spot jogging simulation results: torso pitch orientations

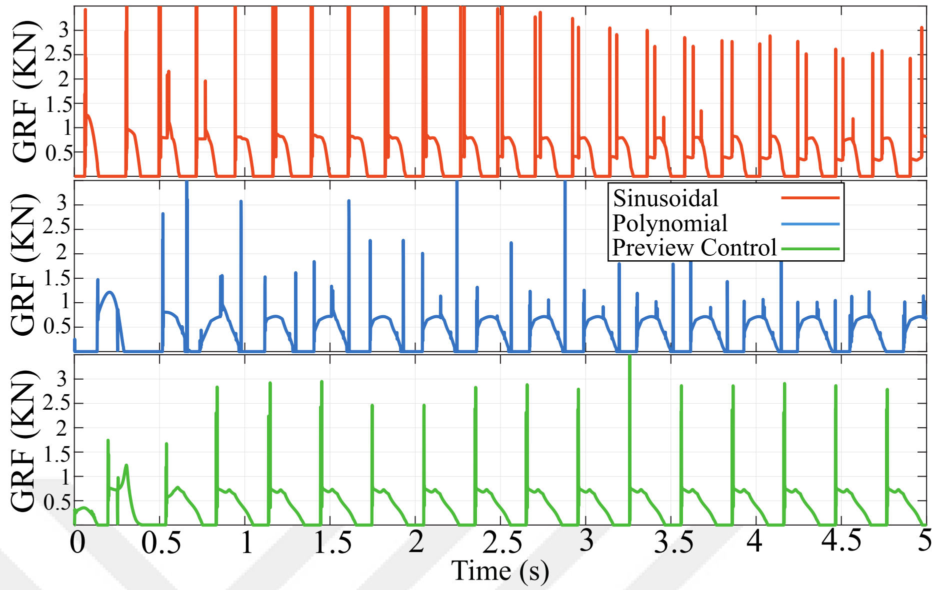

Figure 19: Spot jogging simulation results: total ground reaction forces for RF+LB foots

CHAPTER IV

MOMENTUM-AWARE ROBUST CONTROL

4.1 Centroidal Momentum Observer-Based Control

This section presents CMO and SFC approaches to provide robust locomotion for quadruped robots. Reacting to disturbances and eliminating uncertainties is crucial to satisfy dynamically coherent robot walking. Achieving these situations is possible by correctly calculating the contact forces for inverse dynamics-based control control. In this sense, SFC generates a wrench command by providing acceleration control. Moreover, the disturbances from the contact forces, external forces, and model errors can be observed through the CMO [34], helping the trunk to operate under a nominal wrench. We implemented the virtual model control algorithm using the nominal wrench to compute the contact forces for inverse dynamics-based control.

4.1.1 Spatial Force Control

The spatial force controller regulates the spatial force, including linear and angular acceleration and error of the defined task via the analytical cartesian trajectory planner [7, 23]. This approach controls linear and angular acceleration to reach admissible wrench commands. For linear components of the controller, the equation was utilized as given below.

$$\ddot{\mathbf{p}}_{\text{cmd}} = \mathbf{K}_a(\ddot{\mathbf{p}}^{\text{ref}}_{\text{com}} - \ddot{\mathbf{p}}_{\text{com}}) + \mathbf{K}_v(\dot{\mathbf{p}}^{\text{ref}}_{\text{com}} - \dot{\mathbf{p}}_{\text{com}}) + \mathbf{K}_f\ddot{\mathbf{p}}^{\text{ref}}_{\text{com}} \tag{65}$$

$\mathbf{K}_f \in \mathbb{R}^{3\times3}$, $\mathbf{K}_a \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_v \in \mathbb{R}^{3\times3}$ are the feed-forward and feedback gains. $\ddot{\mathbf{p}}_{\text{cmd}} \in \mathbb{R}^3$ is the command acceleration vector. The physically consistent force command can be obtained, including the total mass of the robot m for (65) as follows:

$$\mathcal{F}_{\text{cal}} = m\ddot{\mathbf{p}}_{\text{cmd}} \tag{66}$$

The angular components of the command accelerations are computed with the quaternion feedback [35]. This feedback mechanism provides orientation control. The quaternion vector is represented below.

$$\mathbf{Q}_{\text{torso}} = \begin{bmatrix} \eta & \boldsymbol{\epsilon} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^{\mathsf{T}} \tag{67}$$

η and $\boldsymbol{\epsilon}$ are the scalar and vector part of the quaternion of the torso. The orientation error can be defined as

$$\mathbf{e}_o = \eta^{\text{ref}}\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon}^{\text{ref}} + [\boldsymbol{\epsilon}^{\text{ref}}\times]\boldsymbol{\epsilon} \tag{68}$$

where $\boldsymbol{\epsilon}^{\text{ref}}\times$ is the skew-symmetric form of the reference quaternion vector part. The angular velocity and acceleration controller can be written as

$$\boldsymbol{\omega}_{\text{cmd}} = \boldsymbol{\omega}^{\text{ref}} - \mathbf{K}_o\mathbf{e}_o \tag{69}$$

$$\dot{\boldsymbol{\omega}}_{\text{cmd}} = \dot{\boldsymbol{\omega}}^{\text{ref}} + \mathbf{K}_\omega(\boldsymbol{\omega}^{\text{ref}} - \boldsymbol{\omega}) - \mathbf{K}_o\boldsymbol{e}_o \tag{70}$$

where $\mathbf{K}_\omega \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_o \in \mathbb{R}^{3\times3}$ are the angular velocity and angular position controller gains. $\boldsymbol{\omega}$, $\dot{\boldsymbol{\omega}}$, and $\dot{\boldsymbol{\omega}}_{\text{cmd}}$ are the angular velocity, angular acceleration, and angular acceleration commands, respectively. To obtain the desired angular moment for centroidal dynamics, angular velocity and acceleration commands were yielded into Euler's rotation equations. This relation is given as

$$\begin{aligned} \mathcal{M}_{\text{cal}} &= \mathbf{I}_{\text{torso}}\,\dot{\boldsymbol{\omega}}_{\text{cmd}} + \boldsymbol{\omega}_{\text{cmd}} \times (\mathbf{I}_{\text{torso}}\,\boldsymbol{\omega}_{\text{cmd}}) \\ &\approx \mathbf{I}_{\text{torso}}\,\dot{\boldsymbol{\omega}}_{\text{cmd}} \end{aligned} \tag{71}$$

where $\mathcal{M}_{\text{cal}} \in \mathbb{R}^3$ and $\mathbf{I}_{\text{torso}} \in \mathbb{R}^{3\times3}$ are calculated moment and inertia tensor of the torso respectively. The calculated force and moment can be formed as the calculated wrench vector $\mathbf{W}_{\text{cal}} \in \mathbb{R}^6$ as follows:

$$\mathbf{W}_{\text{cal}} = \begin{bmatrix} \mathcal{F}_{\text{cal}} \\ \mathcal{M}_{\text{cal}} \end{bmatrix} \tag{72}$$

Depending on the terrain topology, the subsequent open-loop relation adjusts the robot's compliance with the motion.

$$\mathbf{W}_{\text{des}} = \mathbf{K}_h \mathbf{W}_{\text{cal}} \tag{73}$$

$\mathbf{W}_{\text{des}} \in \mathbb{R}^6$ is the desired wrench, and $\mathbf{K}_h \in \mathbb{R}^{6\times6}$ is the wrench gain matrix that scales the calculated wrench.

4.1.2 Observer Synthesis

The centroidal dynamics and the centroidal momentum are the base concepts for observer synthesis. The centroidal momentum represents the total momenta of the system by adding up the momenta of each body [26]. The centroidal momentum was computed via the centroidal momentum matrix (CMM), the linear operator that correlates the floating-base system with its generalized velocities $\dot{\mathbf{q}} \in \mathbb{R}^{18}$ as

$$\mathbf{h}_G = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} \tag{74}$$

where $\mathbf{h}_G \in \mathbb{R}^6$ and $\mathbf{A}_G(\mathbf{q}) \in \mathbb{R}^{6\times18}$ are the centroidal momentum and the CMM respectively. Considering the (74) as state space, the CMO synthesized with high order to reach the disturbance wrench on the trunk of robot [36, 37, 38, 39]. Eq. (74) can be rewritten with disturbance dynamics as

$$\begin{aligned} \mathbf{h}_G &= \mathbf{A}_{\mathbf{Gn}}(\mathbf{q})\dot{\mathbf{q}} + (\mathbf{A}_G(\mathbf{q}) - \mathbf{A}_{\mathbf{Gn}}(\mathbf{q}))\dot{\mathbf{q}} \\ &= \mathbf{A}_{\mathbf{Gn}}(\mathbf{q})\dot{\mathbf{q}} + \Delta\mathbf{h} \end{aligned} \tag{75}$$

where $\mathbf{A}_{\mathbf{Gn}}(\mathbf{q}) \in \mathbb{R}^{6\times18}$ and $\Delta\mathbf{h} \in \mathbb{R}^6$ are the nominal CMM and centroidal momentum disturbance respectively. Differentiating (75) gives the external trunk wrench with disturbance effect $\mathbf{f}_G \in \mathbb{R}^6$ as

$$\dot{\mathbf{h}}_G = \mathbf{f}_G = \dot{\mathbf{A}}_{\mathbf{Gn}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{A}_{\mathbf{Gn}}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\dot{\mathbf{h}} \tag{76}$$

where $\Delta\dot{\mathbf{h}} \in \mathbb{R}^6$ is the disturbance wrench. The auxiliary variable method can be utilized [39] as follows:

$$\mathbf{z} = \Delta\dot{\mathbf{h}} + \mathbf{L}\dot{\mathbf{q}} \tag{77}$$

where $\mathbf{z} \in \mathbb{R}^6$ is the auxiliary variable vector, while $\mathbf{L} \in \mathbb{R}^{6\times18}$ is the observer gain matrix to be calculated. The auxiliary variable vector $\mathbf{z}$ can be obtained by integrating the following equation, which is acquired by arranging (75) and (76):

$$\begin{aligned} \dot{\mathbf{z}} &= \Delta\ddot{\mathbf{h}} + \mathbf{L}\ddot{\mathbf{q}} \\ &= \Delta\ddot{\mathbf{h}} - \mathbf{L}\mathbf{A}^{\dagger}_{\mathbf{Gn}}\mathbf{z} + \mathbf{L}\mathbf{A}^{\dagger}_{\mathbf{Gn}}(\mathbf{f}_G - \dot{\mathbf{A}}_{\mathbf{Gn}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{L}\dot{\mathbf{q}}) \\ &= \Delta\ddot{\mathbf{h}} - \mathbf{U}\mathbf{z} + \mathbf{U}(\mathbf{f}_G - \dot{\mathbf{A}}_{\mathbf{Gn}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{L}\dot{\mathbf{q}}) \end{aligned} \tag{78}$$

where $\dot{\mathbf{z}} \in \mathbb{R}^6$ and $\dagger$ represent the derivative of $\mathbf{z}$ and Pseudo-Inverse, respectively. The calculation of observer gain can be simplified with relation $\mathbf{L} = \mathbf{U}\mathbf{A}_{\mathbf{Gn}}$ where $\mathbf{U} \in \mathbb{R}^{6\times6}$ is the diagonal gain matrix to be tuned. $\Delta\ddot{\mathbf{h}} \in \mathbb{R}^6$ is the first order of disturbance vector which is omitted for estimation. The estimated derivative of $\dot{\mathbf{z}}$ can be rewritten as follows:

$$\dot{\hat{\mathbf{z}}} = -\mathbf{U}\mathbf{z} + \mathbf{U}(\mathbf{f}_G - \dot{\mathbf{A}}_{\mathbf{Gn}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{L}\dot{\mathbf{q}}) \tag{79}$$

Analogous to disturbance observers [40], $\mathbf{f}_G$ in (78) is designated as follows:

$$\mathbf{f}_G = \mathbf{W}_{\mathrm{n}} = \mathbf{W}_{\mathrm{des}} + \Delta\dot{\mathbf{h}} \tag{80}$$

$\mathbf{W}_{\mathrm{n}} \in \mathbb{R}^6$ is the nominal wrench that includes the desired wrench for the torso of the robot with the observed disturbance wrench. In addition, weight offset of the robot [41] added beside the $\mathbf{W}_{\mathrm{n}}$ as

$$\mathbf{W}_{\mathrm{cmd}} = \mathbf{W}_{\mathrm{n}} + \mathbf{W}_{\mathrm{grav}} \tag{81}$$

where $\mathbf{W}_{\mathrm{grav}} = [\mathbf{0}_{1\times2}\ mg\ \mathbf{0}_{1\times3}]^{\intercal}$ is the gravity wrench vector. To map $\mathbf{W}_{\mathrm{cmd}}$ as contact force distribution VMC was utilized [23, 42, 43, 44]. VMC and the proposed CMO-based controller relation can be expressed as below:

$$\mathbf{F}^{\mathrm{cmd}}_{\mathrm{con}} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} \\ {}^1\mathbf{r}_G\times & {}^2\mathbf{r}_G\times & {}^3\mathbf{r}_G\times & {}^4\mathbf{r}_G\times \end{bmatrix}^{\dagger} \mathbf{W}_{\mathrm{cmd}} \tag{82}$$

where $\mathbf{F}_{\text{con}}^{\text{cmd}} \in \mathbb{R}^{12}$, $\mathbf{f}_i \in \mathbb{R}^3$, ${}^{i}\mathbf{r}_G \in \mathbb{R}^3$ and $\mathbf{I}_{(3\times3)}$ are the calculated contact force vector includes all foot, the contact force vector for each foot, the contact position concerning CoM, and identity matrix respectively. The computed $\mathbf{F}_{\text{con}}^{\text{cmd}}$ can be utilized as in Fig. 20 for the computed torque control.

4.1.3 Stability

The stability analysis utilized the constraint and operational kinematics, floating-base dynamics in Sections 2.2.3, 2.2.4 and 2.3, respectively. Initially, the floating base robot dynamics can be represented as follows:

$$\mathbf{M}_f\ddot{\mathbf{q}}_f + \mathbf{h}_f(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{fr}\ddot{\mathbf{q}}_r = \bar{\mathbf{J}}_{cf}^{\intercal}\mathbf{F}_{con} \tag{83}$$

$$\mathbf{M}_r\ddot{\mathbf{q}}_r + \mathbf{h}_r(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{rf}\ddot{\mathbf{q}}_r = \boldsymbol{\tau} + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{F}_{con} \tag{84}$$

where $\mathbf{M}_f \in \mathbb{R}^{6\times6}$, $\mathbf{M}_r \in \mathbb{R}^{12\times12}$ are the mass matrices correspond to floating-base and robot joint, respectively. Additionally, $\mathbf{M}_{rf} \in \mathbb{R}^{6\times12}$ and $\mathbf{M}_{fr} \in \mathbb{R}^{6\times12}$ are the coupling inertia matrices between the floating base and the joints. In (83) and (84), $\mathbf{h}_f \in \mathbb{R}^6$ and $\mathbf{h}_r \in \mathbb{R}^6$ represent the nonlinearity vectors that correspond floating-base and robot joint, respectively. The measured contact force $\mathbf{F}_{con}$ can be written as follows:

$$\mathbf{F}_{con} = \mathbf{A}_{vmc}^{\dagger}(\dot{\mathbf{h}}_n + \Delta\dot{\mathbf{h}}) \tag{85}$$

where $\dot{\mathbf{h}}_n \in \mathbb{R}^6$ and $\mathbf{A}_{vmc}^{\dagger} \in \mathbb{R}^{6\times12}$ are the nominal measured wrench and VMC matrix in (82) provides force mapping from the torso to foot contact force. Moreover, the computed torque control can be written as below:

$$\begin{aligned}\boldsymbol{\tau} = \boldsymbol{\tau}_{pd} + \boldsymbol{\tau}_{ct} = \mathbf{K}_p(\mathbf{q}_r^{\text{ref}} - \mathbf{q}_r) + \mathbf{K}_d(\dot{\mathbf{q}}_r^{\text{ref}} - \dot{\mathbf{q}}_r) \\ + \mathbf{M}_r\ddot{\mathbf{q}}_r^{\text{ref}} + \mathbf{h}_r(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{fr}\ddot{\mathbf{q}}_f^{\text{ref}} - \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{F}_{con}^{\text{cmd}}\end{aligned} \tag{86}$$

where $\mathbf{F}_{con}^{\text{cmd}} \in \mathbb{R}^{3\times k}$ is the contact force command matrix obtained from the VMC and it can be defined as follows:

$$\mathbf{F}_{con}^{\text{cmd}} = \mathbf{A}_{vmc}^{\dagger}\dot{\mathbf{h}}_n^{\text{cmd}} \tag{87}$$

$$\dot{\mathbf{h}}_n^{\text{cmd}} = \dot{\mathbf{h}}_n^{\text{des}} + \Delta\hat{\dot{\mathbf{h}}} \tag{88}$$

where $\dot{\mathbf{h}}_n^{\text{cmd}} \in \mathbb{R}^6$ and $\dot{\mathbf{h}}_n^{\text{des}} \in \mathbb{R}^6$ represents the nominal and desired contact wrench command that correspond $\mathbf{F}_{con}^{\text{cmd}}$, respectively. We can obtain the following error dynamics by combining (84) and (86).

$$\begin{aligned}\mathbf{M}_r\ddot{\mathbf{q}}_r + \cancel{\mathbf{h}_r(\mathbf{q},\dot{\mathbf{q}})} + \cancel{\mathbf{M}_{rf}\ddot{\mathbf{q}}_f} &= \mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{q}}_r^{\text{ref}} \\ &\quad + \cancel{\mathbf{M}_{rf}\ddot{\mathbf{q}}_f} + \cancel{\mathbf{h}_r(\mathbf{q},\dot{\mathbf{q}})} + \mathbf{J}_{cr}^{\intercal}(\mathbf{F}_{con} - \mathbf{F}_{con}^{\text{cmd}})\end{aligned} \tag{89}$$

$$\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r + \bar{\mathbf{J}}_{cr}^{\intercal}(\mathbf{F}_{con} - \mathbf{F}_{con}^{\text{cmd}}) = \mathbf{0} \tag{90}$$

In (90), $\mathbf{e}_r \in \mathbb{R}^{12}$ and $\dot{\mathbf{e}}_r \in \mathbb{R}^{12}$ are the joint position error and join velocity error, respectively. We can redefine (90) by using (88) as below:

$$\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r + \mathbf{J}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}(\dot{\mathbf{h}}_n - \dot{\mathbf{h}}_n^{\text{des}} + \Delta\dot{\mathbf{h}} - \Delta\hat{\dot{\mathbf{h}}}) = \mathbf{0} \tag{91}$$

The controllers in (65) and (70) can be defined as follows:

$$\dot{\mathbf{h}}^{\text{des}} = \mathbf{K}_h\Big\{\ddot{\mathbf{q}}_f^{\text{ref}} + \mathbf{K}_a\ddot{\mathbf{e}}_f + \mathbf{K}_v\dot{\mathbf{e}}_f + \mathbf{K}_x\mathbf{e}_f\Big\} \tag{92}$$

We can yield $\mathbf{F}_{con}$ following expression utilizing (83).

$$\mathbf{F}_{con} = (\mathbf{J}_{cf}^{\intercal})^{\dagger}(\mathbf{M}_f\ddot{\mathbf{q}}_f + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{fr}\ddot{\mathbf{q}}) \tag{93}$$

By utilizing (85) and (92), the following relation can be obtained.

$$\dot{\mathbf{h}}_n + \Delta\dot{\mathbf{h}} = \mathbf{A}_{vmc}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}(\mathbf{M}_f\ddot{\mathbf{q}}_f + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{fr}\ddot{\mathbf{q}}) \tag{94}$$

We can plug the (94) and (92) into (91) and the following equation can be yielded.

$$\begin{aligned}&\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r + \bar{\mathbf{J}}_{cr}^{\intercal}\cancel{\mathbf{A}_{vmc}^{\dagger}\mathbf{A}_{vmc}}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}(\mathbf{M}_f\ddot{\mathbf{q}}_f + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{fr}\ddot{\mathbf{q}}) \\ &= \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\ddot{\mathbf{q}}_f^{\text{ref}} + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\mathbf{K}_a\ddot{\mathbf{e}}_f + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\mathbf{K}_v\dot{\mathbf{e}}_f + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\mathbf{K}_x\mathbf{e}_f \\ &+ \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\intercal}\Delta\hat{\dot{\mathbf{h}}}\end{aligned} \tag{95}$$

The (95) can be rearranged by collecting similar variables as below:

$$\begin{aligned}&\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r + \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}(\mathbf{M}_f\ddot{\mathbf{q}}_f + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})) = \bar{\mathbf{J}}_{cr}^{\intercal}(\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h - (\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\ddot{\mathbf{q}}_f) \\ &+ \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\mathbf{K}_a\ddot{\mathbf{e}}_f + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\mathbf{K}_v\dot{\mathbf{e}}_f + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\mathbf{K}_x\mathbf{e}_f + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\intercal}\Delta\hat{\dot{\mathbf{h}}}\end{aligned} \tag{96}$$

For stability condition, we can assume following expression to simplify (96) as follows:

$$\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h\cancelto{\approx \mathbf{0}}{\ddot{\mathbf{q}}_f^{\text{ref}}} - (\mathbf{J}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\cancelto{\approx \mathbf{0}}{\ddot{\mathbf{q}}_f} = \mathbf{0} \tag{97}$$

$$\mathbf{A}_{vmc}^{\dagger}\mathbf{K}_h = (\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f \tag{98}$$

The Eq. (98) can be plugged into the (96) and the following expression can be obtained.

$$\begin{aligned} &\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r + \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}(\mathbf{M}_{fr}\ddot{\mathbf{q}}_{\mathbf{r}} + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{A}_{vmc}^{\dagger}\Delta\dot{\hat{\mathbf{h}}})) \\ &= \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\Big\{(\mathbf{I}+\mathbf{K}_a)\ddot{\mathbf{e}}_f + \mathbf{K}_v\dot{\mathbf{e}}_f + \mathbf{K}_x\mathbf{e}_f\Big\} \end{aligned} \tag{99}$$

The Eq. (96) used to rewrite real momentum rate error as in following equation.

$$\Delta\dot{\mathbf{h}} = \mathbf{A}_{vmc}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}(\mathbf{M}_{fr}\ddot{\mathbf{q}}_r + \mathbf{h}_f(\mathbf{q},\dot{\mathbf{q}})) \tag{100}$$

We can plug the (100) into (99) to obtain more simplified relation as follows:

$$\begin{aligned} &\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r + \bar{\mathbf{J}}_{cr}^{\intercal}\mathbf{A}_{vmc}^{\dagger}(\Delta\dot{\mathbf{h}} - \Delta\dot{\hat{\mathbf{h}}}) \\ &\qquad = \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\Big\{(\mathbf{I}+\mathbf{K}_a)\ddot{\mathbf{e}}_f + \mathbf{K}_v\dot{\mathbf{e}}_f + \cancelto{\approx 0}{\mathbf{K}_x\mathbf{e}_f}\Big\} \end{aligned} \tag{101}$$

where $\Delta\dot{\mathbf{h}} - \Delta\dot{\hat{\mathbf{h}}} \approx \mathbf{0}$ when $\Delta\ddot{\mathbf{h}} = \mathbf{0}$ for stability condition. Moreover, we can utilize the following expression by recalling (15) to simplify (101).

$$\dot{\mathbf{q}}_f = -\underbrace{\bar{\mathbf{J}}_{cf}^{\dagger}\bar{\mathbf{J}}_{cr}}_{\bar{\mathbf{J}}_{fr}}\dot{\mathbf{q}}_r \tag{102}$$

$$\dot{\mathbf{e}}_f = \mathbf{J}_{fr}\dot{\mathbf{e}}_r \qquad \ddot{\mathbf{e}}_f = \dot{\bar{\mathbf{J}}}_{fr}\dot{\mathbf{e}}_r + \bar{\mathbf{J}}_{fr}\dot{\mathbf{e}}_r \tag{103}$$

The eq. (103) was provided to plug into (101) to relate floating-base errors with joint errors. The obtained relation is given below.

$$\mathbf{K}_p\mathbf{e}_r + \mathbf{K}_d\dot{\mathbf{e}}_r + \mathbf{M}_r\ddot{\mathbf{e}}_r = \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\Big\{(\mathbf{I}+\mathbf{K}_a)\dot{\bar{\mathbf{J}}}_{fr}\dot{\mathbf{e}}_r + (\mathbf{I}+\mathbf{K}_a)\dot{\bar{\mathbf{J}}}_{fr}\ddot{\mathbf{e}}_r + \mathbf{K}_v\bar{\mathbf{J}}_{fr}\dot{\mathbf{e}}_r\Big\} \tag{104}$$

We can finalize the simplification with the subsequent term to obtain the generalized stability conditions.

$$\begin{aligned} &\Big[\mathbf{K}_d - \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f(\mathbf{I}+\mathbf{K}_a)\dot{\bar{\mathbf{J}}}_{fr} - \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\mathbf{K}_v\bar{\mathbf{J}}_{fr})\Big]\dot{\mathbf{e}}_r \\ &+ \Big[\mathbf{M}_r - \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f(\mathbf{I}+\mathbf{K}_a)\Big]\ddot{\mathbf{e}}_r + \mathbf{K}_p\mathbf{e}_r = \mathbf{0} \end{aligned} \tag{105}$$

$$\mathbf{C}_1 = \left[\mathbf{K}_d - \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f(\mathbf{I}+\mathbf{K}_a)\dot{\mathbf{J}}_{fr} - \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f\mathbf{K}_v\bar{\mathbf{J}}_{fr})\right] > \mathbf{0} \tag{106}$$

$$\mathbf{C}_2 = \left[\mathbf{M}_r - \bar{\mathbf{J}}_{cr}^{\intercal}(\bar{\mathbf{J}}_{cf}^{\intercal})^{\dagger}\mathbf{M}_f(\mathbf{I}+\mathbf{K}_a)\right] > \mathbf{0} \tag{107}$$

In (106) and (107), $\mathbf{C}_1 \in \mathbb{R}^{12\times 12}$ and $\mathbf{C}_2 \in \mathbb{R}^{12\times 12}$ denote the stability condition matrices. We can tune the controllers for the designated stability condition matrices that should be positive definite.

Additionally, the CMO stability can be defined as follows:

$$\dot{\mathbf{z}} - \dot{\hat{\mathbf{z}}} = -\mathbf{U}(\mathbf{z} - \hat{\mathbf{z}}) + \Delta\ddot{\mathbf{h}} \tag{108}$$

where $\mathbf{z}_{err} \in \mathbb{R}^6$ represents the error of the auxiliary variable. When $\Delta\ddot{\mathbf{h}} \neq \mathbf{0}$ the estimated value $\hat{\mathbf{z}}$ asymptotically approaches $\mathbf{z} + \mathbf{z}_{err}$ asymptotically. Moreover, If $\Delta\ddot{\mathbf{h}} = \mathbf{0}$, the estimated value $\hat{\mathbf{z}}$ asymptotically approaches $\mathbf{z}$.

4.2 Overall Control Scheme

The general control block diagram is displayed in Fig. 20. In this figure, $\mathbf{p}_{\text{com}}$, $\dot{\mathbf{p}}_{\text{com}}$, $\mathbf{p}_{\text{f}} \in \mathbb{R}^3$, and $\boldsymbol{\omega}_{\text{torso}} \in \mathbb{R}^3$ denote CoM position, CoM velocity, foot position, and angular velocity, respectively. Measured quantities have no superscript, but the superscript ref denotes the reference value. Acceleration, angular location, and linear and angular velocity data are all included in the IMU. The task of evaluating operator commands and producing ZMP-based CoM trajectories, swing leg trajectories, and torso orientation belongs to the Cartesian trajectory planner in this block. These trajectories are evaluated by inverse kinematics block, which also generates related joint angles [23]. To achieve position control with compliance, e.g., lower PD gains, we used a computed torque control scheme in which floating-base dynamics utilized [45]. In addition, Spatial Force Control (SFC) was implemented to address force regularization/distribution [35, 46, 47]. The proposed observer was integrated for more robust locomotion control performance as it was key in handling various terrain conditions and perturbations. Additionally, the Virtual Model Control block provides to map the obtained wrench to the contact force reference.

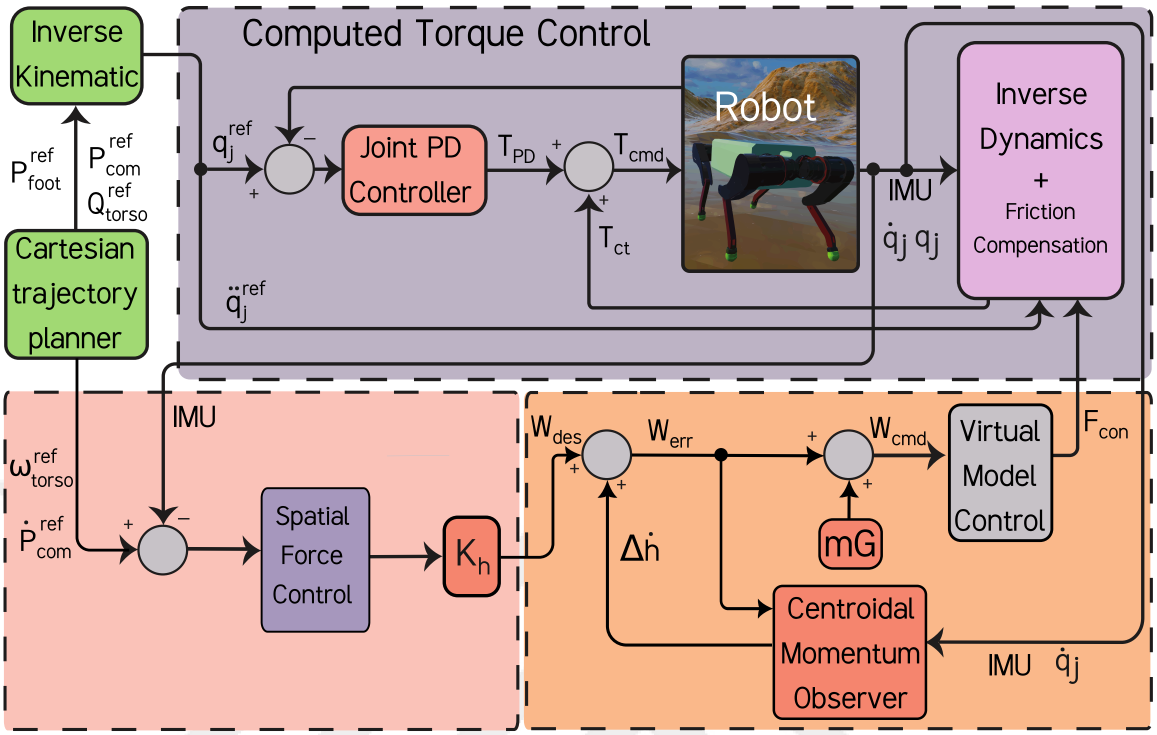

Figure 20: Block diagram of the proposed CMO controller implementation

4.2.1 Friction Compensation

To enhance the joint level control, we improved the friction model by identifying the friction model. To identify the friction model, we utilized the DOB concept in [48]. We collected the angular velocity and disturbance torque data. To model friction we utilized the *tanh*() function as follows:

$$\tau_f = \chi(\alpha\, tanh(\epsilon\,\dot{\theta}) + \gamma\,\dot{\theta} + \beta) \tag{109}$$

$$\chi = \begin{cases} 0, & \text{if } \dot{\theta} < \kappa \\ 1, & \text{if } \dot{\theta} > \kappa \end{cases} \tag{110}$$

where χ is the friction compensation activator boolean that prevent unwanted switching. α is the function's amplitude to adjust the friction range. ϵ is the scaling factor that affects the slope of the functions. γ is the velocity coefficient that affects the friction model's slope. β provides to add an offset for the model. κ is the velocity threshold value. Through the obtained data from the robot joint, we obtained the model in Fig. 21.

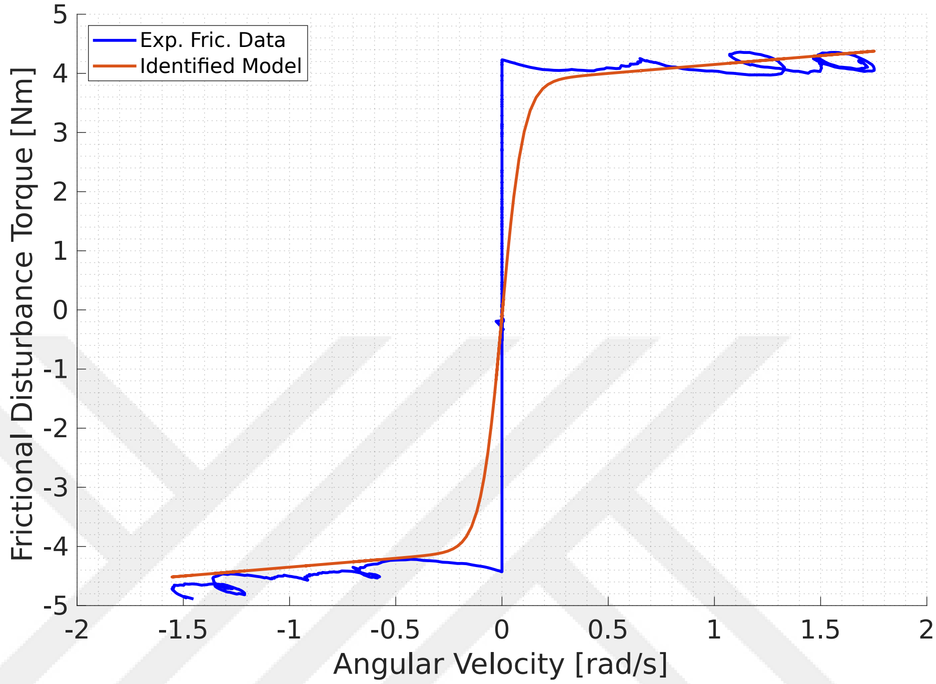

Figure 21: Friction compensation model

4.3 Results and Benchmarking

In order to test the functionality of the devised locomotion controller, a series of simulations was carried out on the Raisim simulator via the Kara quadruped robot model. The simulation was worked at 1 kHz on the workstation with an Intel Xeon 3.30 GHz CPU and 32GB RAM. The simulations' language was `C++`, and the linear algebra library `Eigen` [49] was used to generate control algorithms and computations. The proposed controller was compared with the two selected approaches in the literature: SFC and GMO. Each simulation scenario was conducted thrice with designated force controllers only SFC, SFC+GMO, and SFC+CMO, and they can be enumerated as:

- Simulation #0: A low friction surface was created in the Raisim simulation environment, and the walking test was conducted via Kara.

Table 3: Controller gains for simulation scenario

Joint PD Controller		Spatial Force Control			
$\mathbf{K_p}$	$diag(120, 40, 40)$	$\mathbf{K_f}$	Identity	$\mathbf{K_h}$	Identity
$\mathbf{K_d}$	$diag(12, 4, 4)$	$\mathbf{K_x}$, $\mathbf{K_o}$	$diag(20, 20, 20)$	$\mathbf{K_v}$, $\mathbf{K_\omega}$	$diag(20, 20, 20)$
Generalized Momentum Observer		**Centroidal Momentum Observer**			
$\mathbf{L}$	$diag(20, ..., 20)_{18\times18}$	$\mathbf{U}$	$diag(5, 5, 20, 100, 100, 100)$		

- Simulation #1: Kara performed forward walking at 1.2 m/s speed on flat surface to examine each controller performance.
- Simulation #2: Kara performed forward walking at 0.7 m/s speed on a flat surface, and 120N external force was applied twice to the robot's torso.
- Simulation #3: Kara walked on rough terrain to reach a speed of 0.6 [m/s].
- Simulation #4: Kara performed forward walking to reach 0.6 [m/s] speed on uneven terrain, and 150 [N.m] external moment was applied twice to the robot's torso.

The simulations' controller gains were set to constant values tabulated in 3. The cartesian trajectory planner parameters were adjusted for different sets of simulations. For simulations 1 and 2, T_s was set to 0.24 seconds, and the T_d was set to 0.12 seconds. For simulations 3 and 4, the T_s was set to 0.30 seconds, and the duration T_d was set to 0.15 seconds.

Additionally, we conducted an experiment on the real robot system and presented the preliminary experiment results. In experiments, we compare the SFC with SFC+CMO. Furthermore, we added a friction compensation model to the joint controllers.

4.3.1 Simulation # 0 : Forward Walking on Slippery Terrain

In this scenario, the low friction surface was generated in the Raisim simulation. The surface friction coefficient was set as 0.1. The robot's target mean velocity was 0.6 [m/s]. In Fig. 23, the phase portraits of the CoM were given. Moreover, the solid purple, orange, and green lines represent the SFC, SFC+GMO, and SFC+CMO performance, respectively. In this figure, the x axis indicates the local CoM position while the y axis represents the CoM velocity. These phase portraits provide a visual idea of the controller's regulation performance. From these results, phase portraits show that the SFC+CMO case exhibits a more consistent phase trajectory than the other cases. For example, SFC+GMO case phase trajectory variation is more significant than other cases.

Additionally, The Fig 22 includes the torso pitch rate. Through these results, we obtained the RMSE values of torso pitch rates are 6.25 [deg/s] for SFC, 11.28 [deg/s] for SFC+GMO, and 1.47 [deg/s] for SFC+CMO.

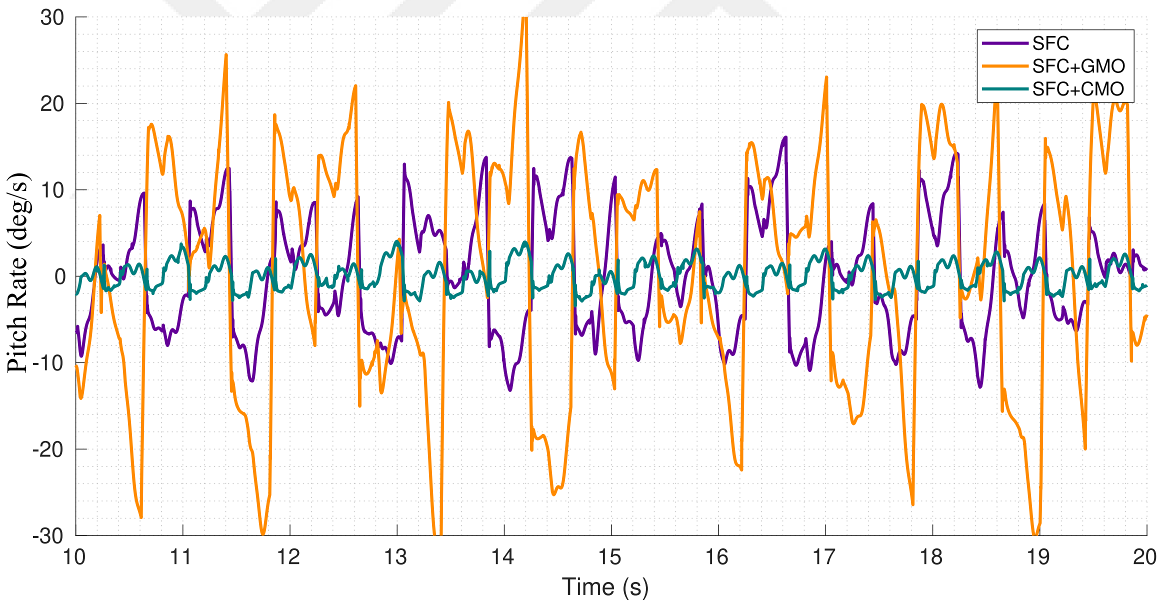

Figure 22: Simulation # 0: pitch rate results

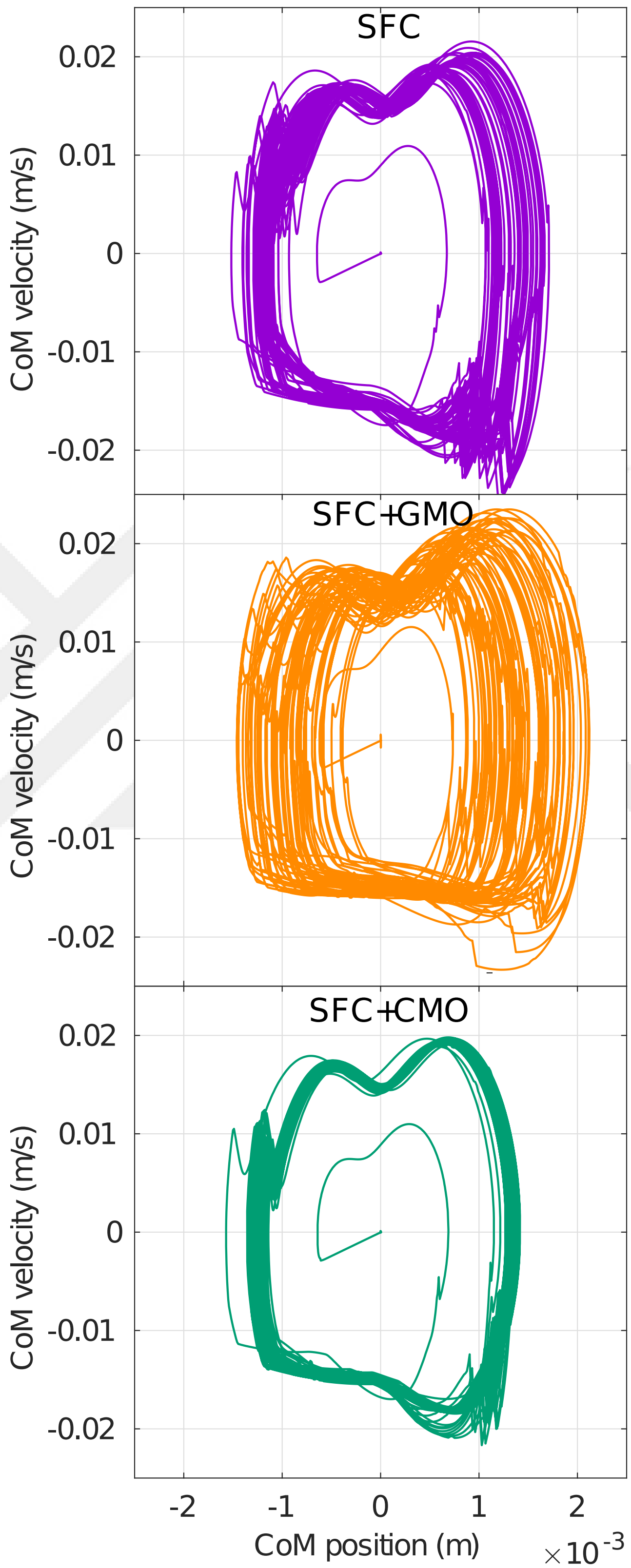

Figure 23: Simulation # 0: phase portraits

4.3.2 Simulation # 1 :High Speed Forward Walking on Flat Surface

In this scenario, the robot was walked on the flat terrain with a 1.2 [m/s] target mean velocity. All controllers exhibited dynamically balanced performance. To indicate each controller performance, the simulation results were given in Figs. 24, 26, 25, 27, 28, and 29. The solid purple, orange, and green lines represent the SFC, SFC+GMO, and SFC+CMO data in these results.

Firstly, the measured torso CoM velocity along the x axis for each controller was given in Fig 25. In this figure, the torso velocity for SFC at 9^{th} seconds between 12^{th} seconds deviated from the command velocity while SFC+GMO and SFC+CMO maintained the command velocity. Moreover, the RMSE of the torso velocity is 0.108 [m/s] for SFC, 0.131 [m/s] for SFC+GMO, and 0.113 for SFC+CMO. Secondly, The torso yaw rates were given in Fig. 24 for each controller. Through these results, RMSE were obtained, which are 6.64 [deg/s] for SFC, 4.18 [deg/s] for SFC+GMO, and 1.40 [deg/s] for SFC+CMO.

Additionally, the local Y_{zmp} peak-to-peaks in Fig. 26 are 22.12 [cm], 5.98 [cm], and 2.8 [cm] for SFC, SFC+, and SFC+CMO, respectively. Lastly, the GRF results were given in the Figs. 27, 28, and 29. From these results, we provided the GRFs mean±std are 2.40±0.60 [kN], 2.49±0.28 [kN], and 0.66±0.06 [kN] for SFC, SFC+GMO, and SFC+CMO, respectively.

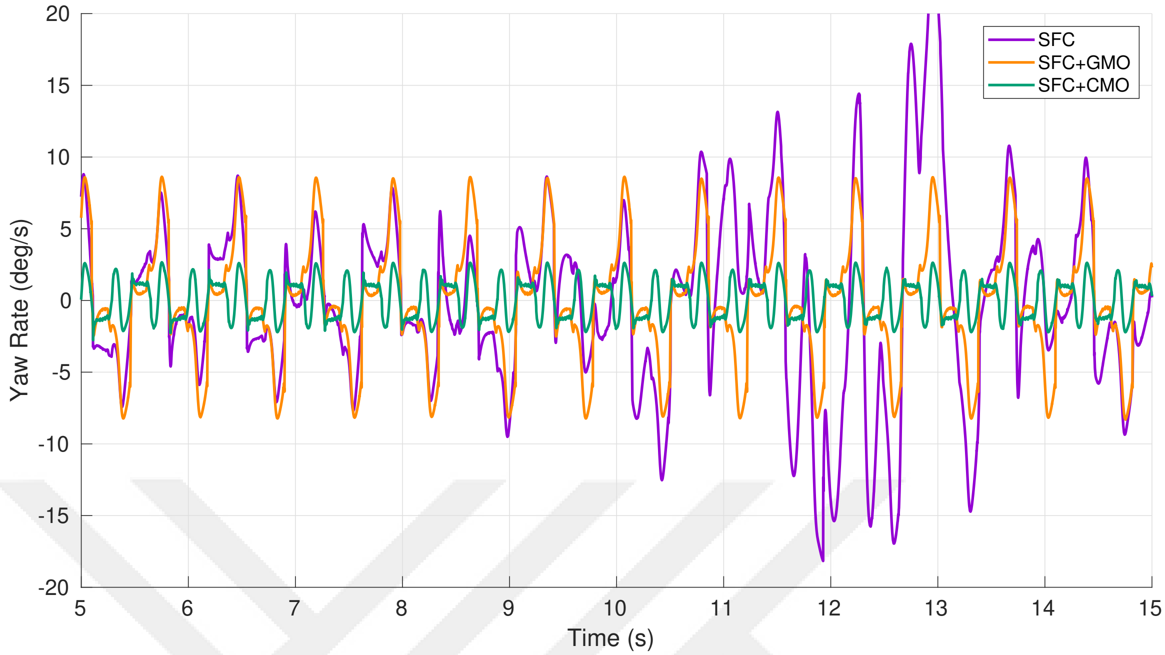

Figure 24: Simulation # 1: yaw rate results

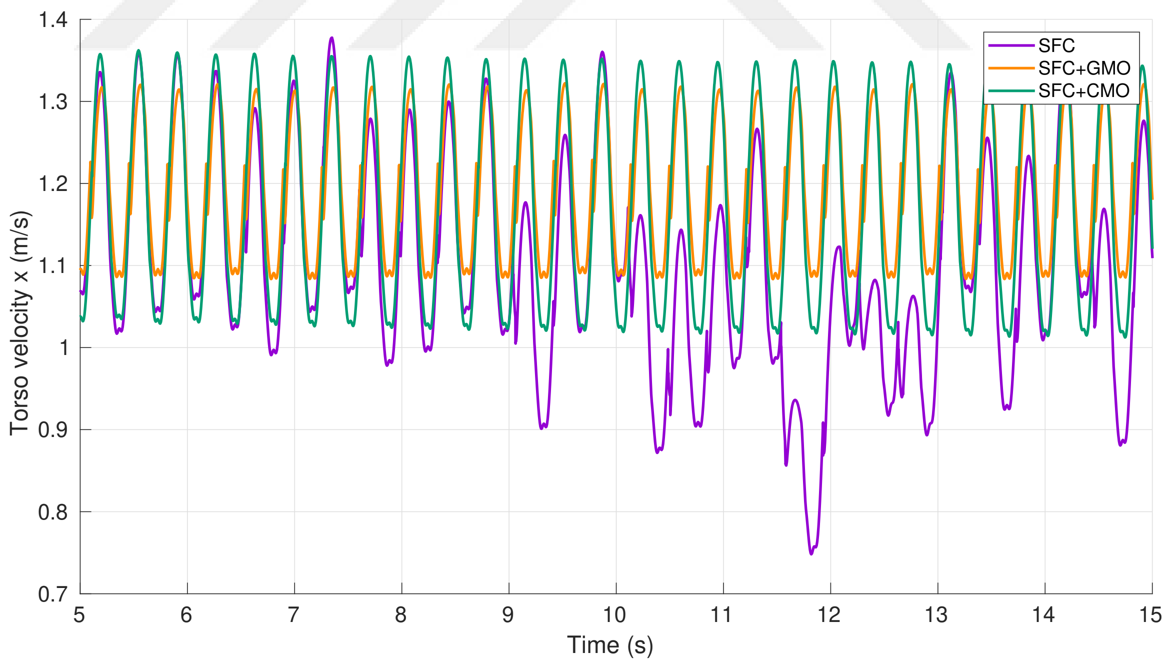

Figure 25: Simulation # 1 results: torso velocity in x axis

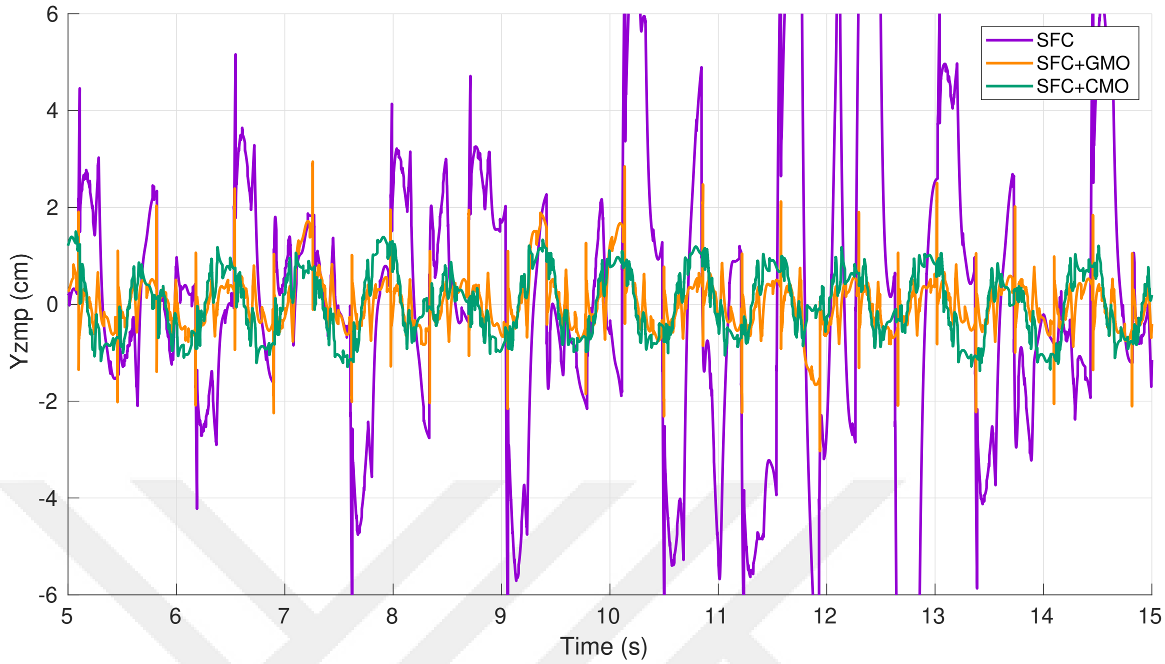

Figure 26: Simulation # 1 results: Y_{zmp} response

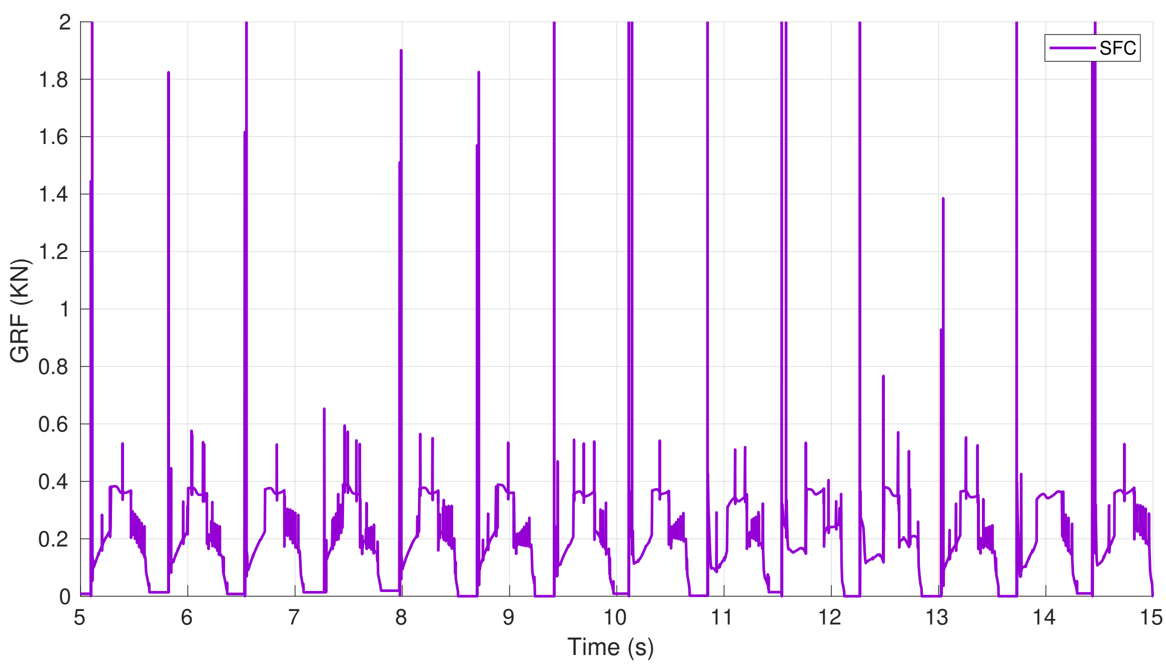

Figure 27: Simulation # 1 results: ground reaction forces for SFC

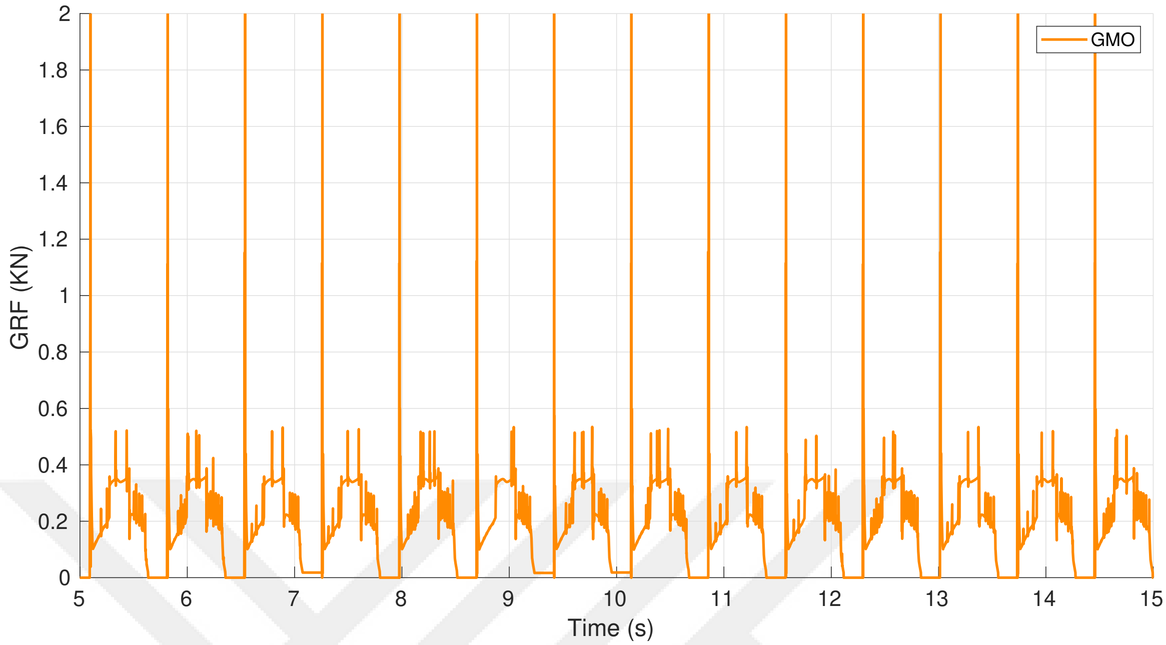

Figure 28: Simulation # 1 results: ground reaction forces for SFC+GMO

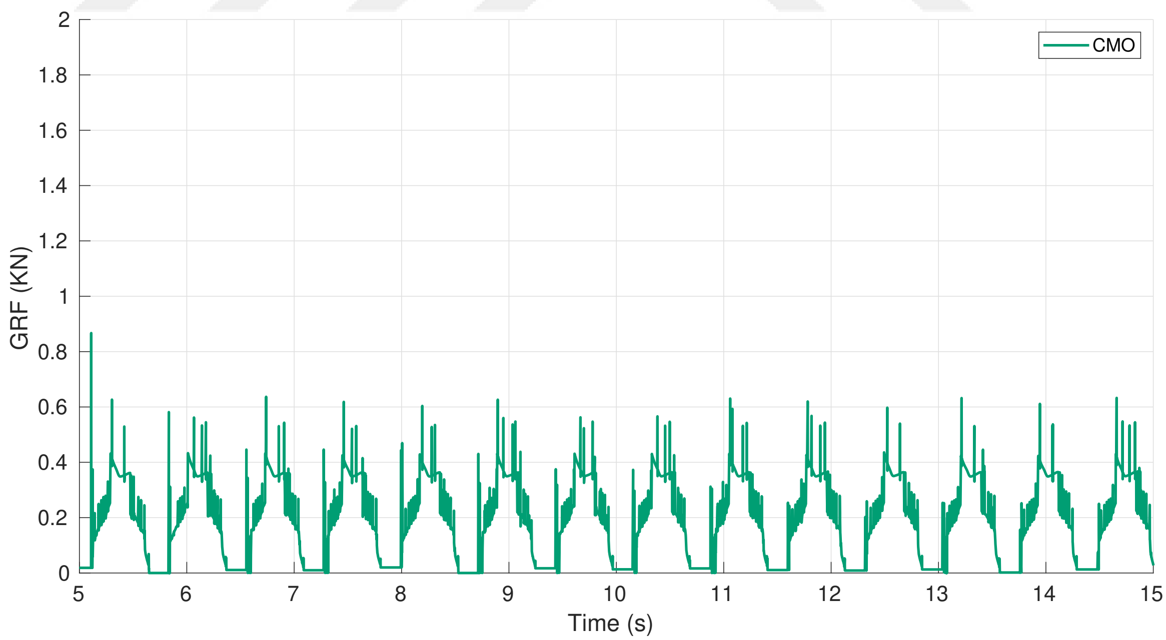

Figure 29: Simulation # 1 results: ground reaction forces for SFC+CMO

4.3.3 Simulation # 2 :Forward Walking on Flat Surface under Disturbance

To demonstrate posture regulation abilities, we applied a 150 [N.m] external moment around the x-axis twice on the robot's torso when the robot executed forward walking. The target mean velocity of the robot was 0.7 [m/s]. As a result of the simulation, all controllers presented a dynamically coherent locomotion behavior. Furthermore, the simulation results were presented in Figs. 30, 31, 32, 33, 34, and 35. The solid purple, orange, and green lines in these results depict the SFC, SFC+GMO, and SF+CMO data. Likewise, the yellow rectangular regions at 6 and 12 seconds represent the external moment.

In Fig. 31, the torso velocity along the x axis result was given. Through this result, we calculated the RMSE values as 6.53 [cm/s], 5.5 [cm/s], and 2.2 [cm/s] for SFC, SFC+GMO, and SFC+CMO, respectively. The torso yaw rates were also given in Fig. 30. The RMSE values are 6.64 [deg/s], 5.83 [deg/s], and 1.04 [deg/s] for SFC, SFC+GMO, and SFC+CMO, respectively. Moreover, Fig. 26 includes the Y_{zmp} measurements for each controller, and the Y_{zmp} peak-to-peak values are 22.35 [cm], 22.27 [cm], and 33.29 [cm] for SFC, SFC with GMO, SFC+CMO, respectively. Similarly, the GRF results were presented in the Figs 33, 34, and 35 for SFC, SFC with GMO, SFC+CMO, respectively. In light of these results, the mean±std values were computed that are 1.68±0.43 [kN], 2.08±0.2 [kN], 0.98±0.32 [kN] for SFC, SFC+GMO, and SFC+CMO, respectively.

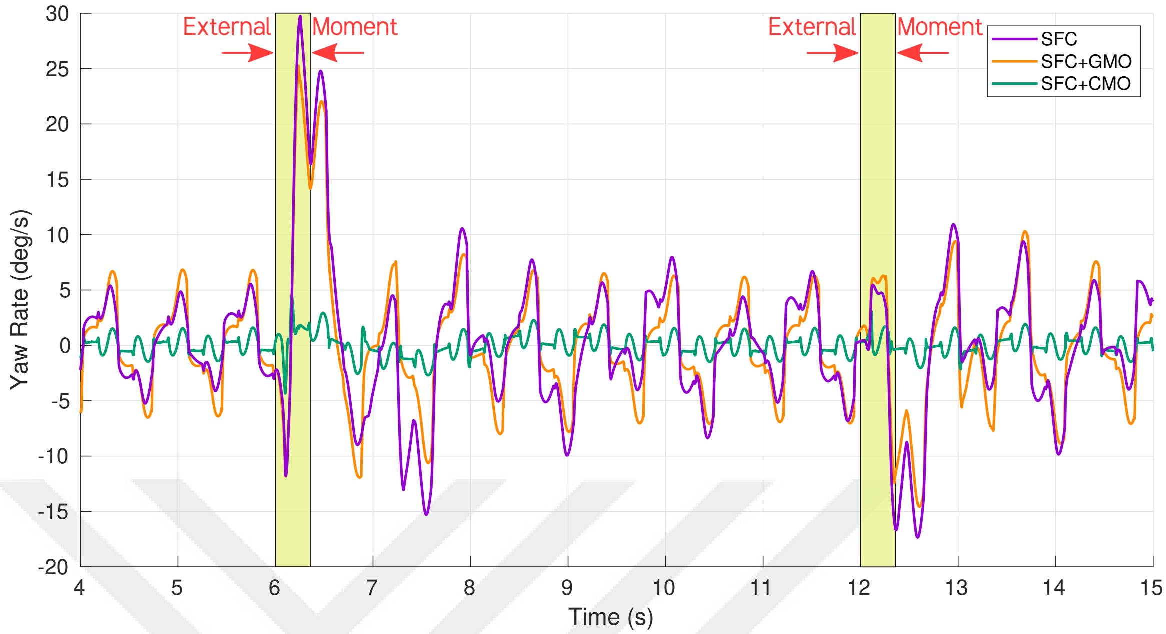

Figure 30: Simulation # 2: yaw rate results

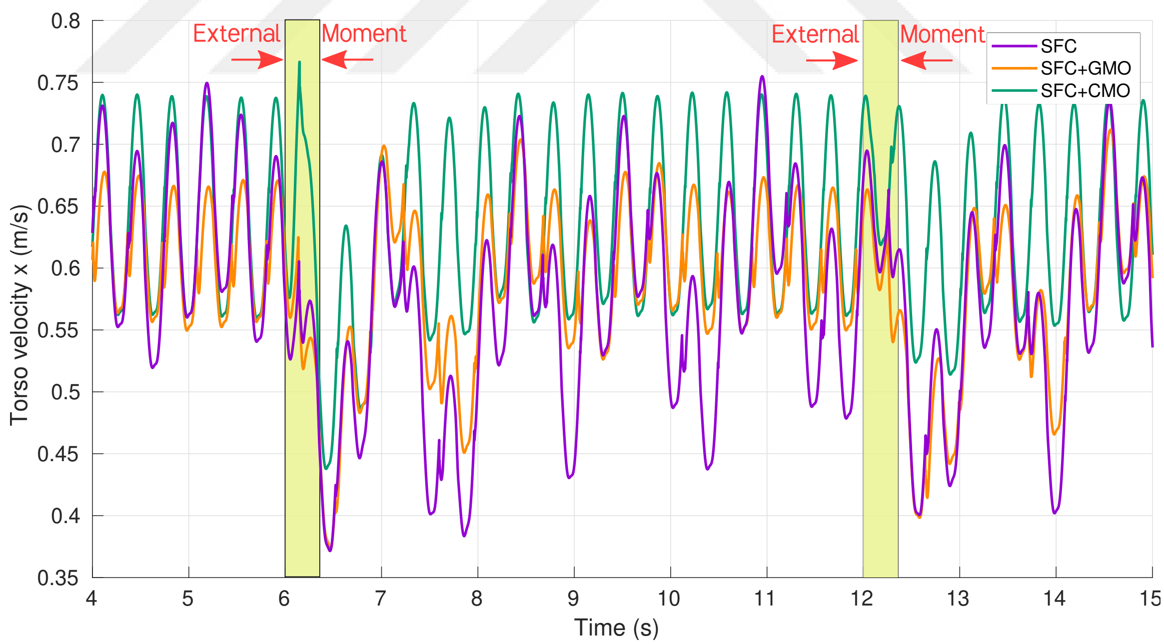

Figure 31: Simulation # 2 results: torso velocity in x axis

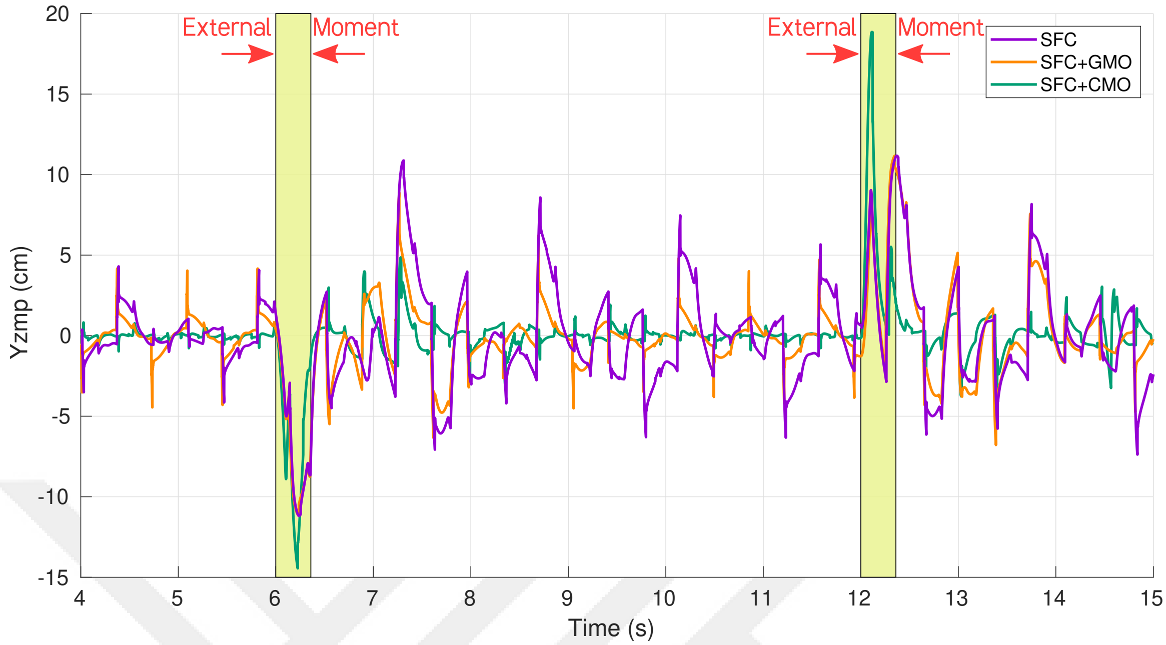

Figure 32: Simulation # 2 results: Y_{zmp} response

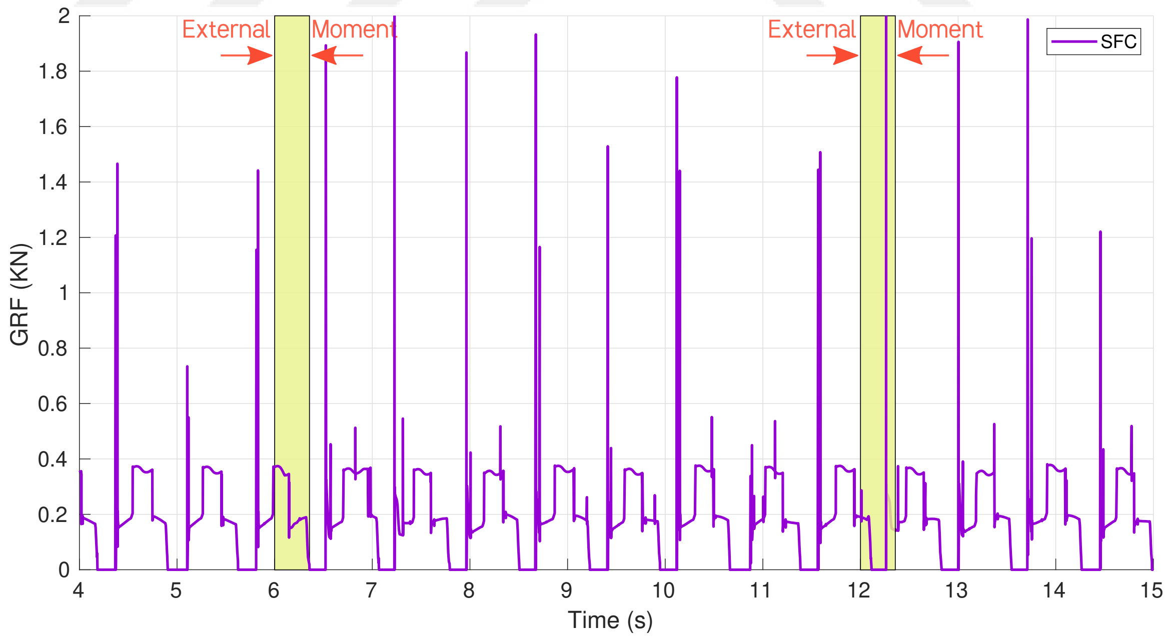

Figure 33: Simulation # 2 Results: ground reaction forces for SFC

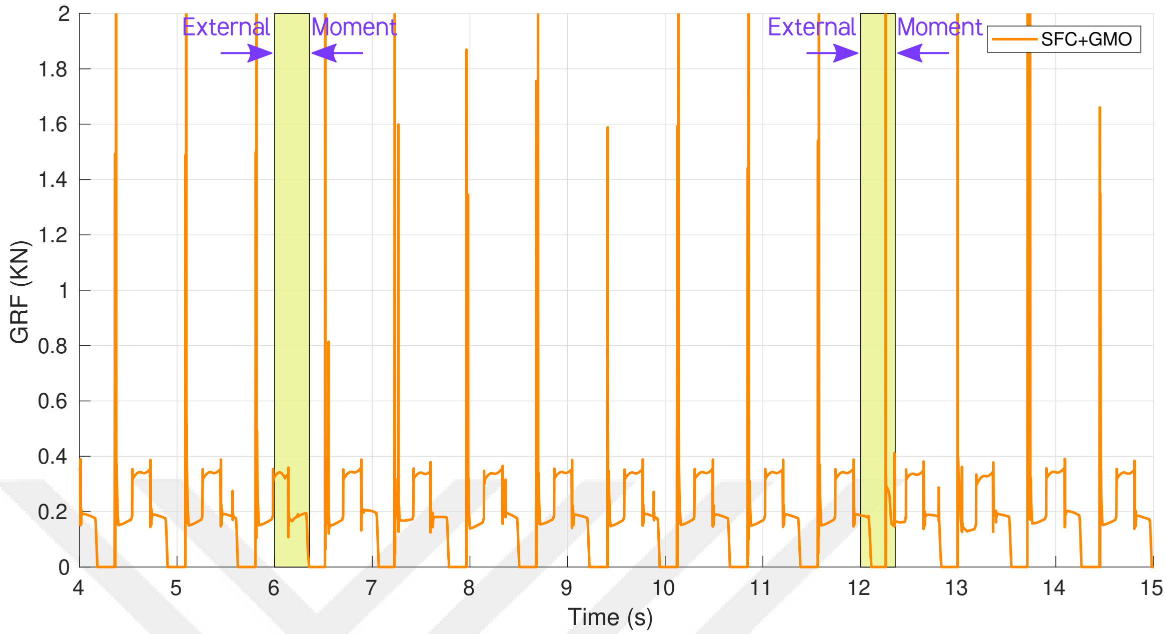

Figure 34: Simulation # 2 Results: ground reaction forces for SFC+GMO

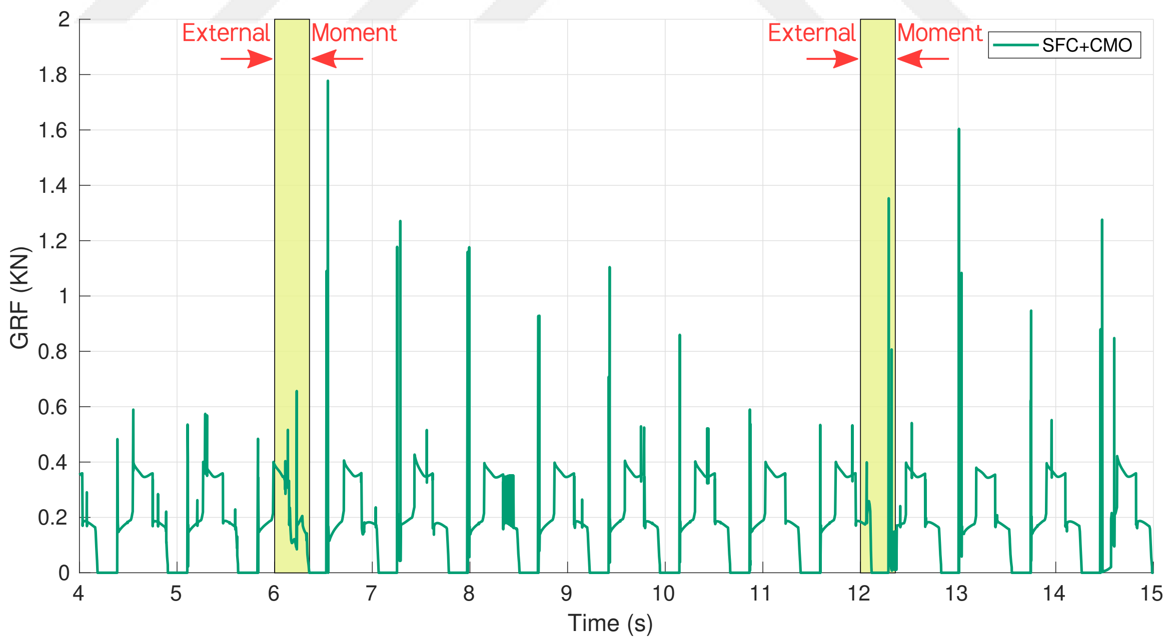

Figure 35: Simulation # 2 results: ground reaction forces for SFC+CMO

4.3.4 Simulation # 3 :Hill Climbing on Rough Terrain

In order to test the performance of the controllers on rough terrain, we generated the height map on the Raisim simulator as in Fig. 37. The height map can be generated on the Raisim simulator using a terrain generator or a height map image. The terrain generator utilizes the Perlin noise concept to generate a realistic height map [50]. The obtained terrain was given in Fig. 36.

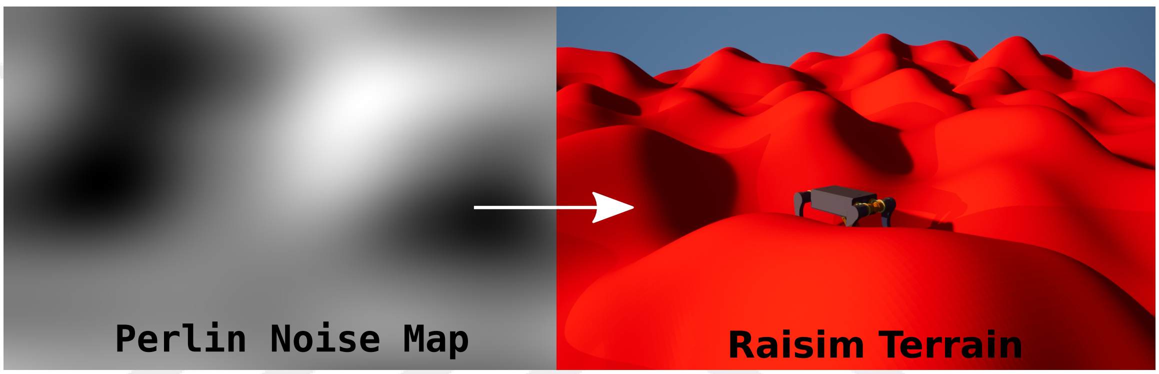

Figure 36: Terrain generation via Perlin Noise

The following parameters can adjust the terrain properties:

- **Frequency**: It scales the Perlin noise, and it provides to stretch or compress noise patterns on terrain. A higher frequency designates more significant broad hills, while a higher frequency provides more detailed terrain features.
- **zScale**: It scales the amplitude of the noise values. A higher value provides higher hills and valleys, while a lower value exhibits more flat terrain.
- **xSize & ySize**: These parameters define the resolution of the map. The larger values provide a large height map.
- **fractalOctaves**: This parameter controls the number of layers of Perlin noise to generate the final terrain.

- **fractalLacunarity**: It provides to shape frequency octave. For example, a lacunarity less than 1.0 means that each subsequent octave will have a lower frequency than the previous one, resulting in smoother details at higher octaves.

- **fractalGain**: This parameter determines the shape amplitude with each octave. A lower fractal gain resulted in the higher-frequency octaves affecting the overall terrain shape.

- **heightOffset**: This parameter shifts the entire terrain in world along the z axis.

The robot was placed randomly on the generated rough terrain map, and the robot command mean velocity was set to 0.6 [m/s]. As a result of the simulation, SFC+GMO and SFC+CMO represented dynamically coherent performance, while the SFC tipped over after some seconds. This means the SFC is insufficient for blind locomotion on rough terrain.

To display controllers performance, we presented the results in the figures 38, 39, 40, 41, 42 and, 43. In these figures, the solid purple, orange, and green lines in these results depict the SFC, SFC+GMO, and SFC+CMO data.

The torso velocity along the x axis results for each controller were given in Fig. 38. By utilizing these results, we obtained the RMSE values as 8.30 [deg/s], 5.95 [deg/s], and 1.35 [deg/s] for the SFC, SFC+GMO, and SFC+CMO, respectively. Additionally, the Y_{zmp} results were presented in Fig. 40 and the Y_{zmp} peak-to peaks are 7.92 [cm] 18.45 [cm], and 12.15 [cm] for the SFC, SFC+GMO, and SFC+CMO, respectively.

Moreover, the GRFs were provided in figures 41, 42, and 43. From these results, the mean±std of GRFs are 1.58±0.48 [kN], 1.63±0.349113 [kN], and 1.84±0.86 [kN] for the SFC, SFC+GMO and SFC+CMO, respectively.

Figure 37: The simulation model of the Kara on the rough terrain

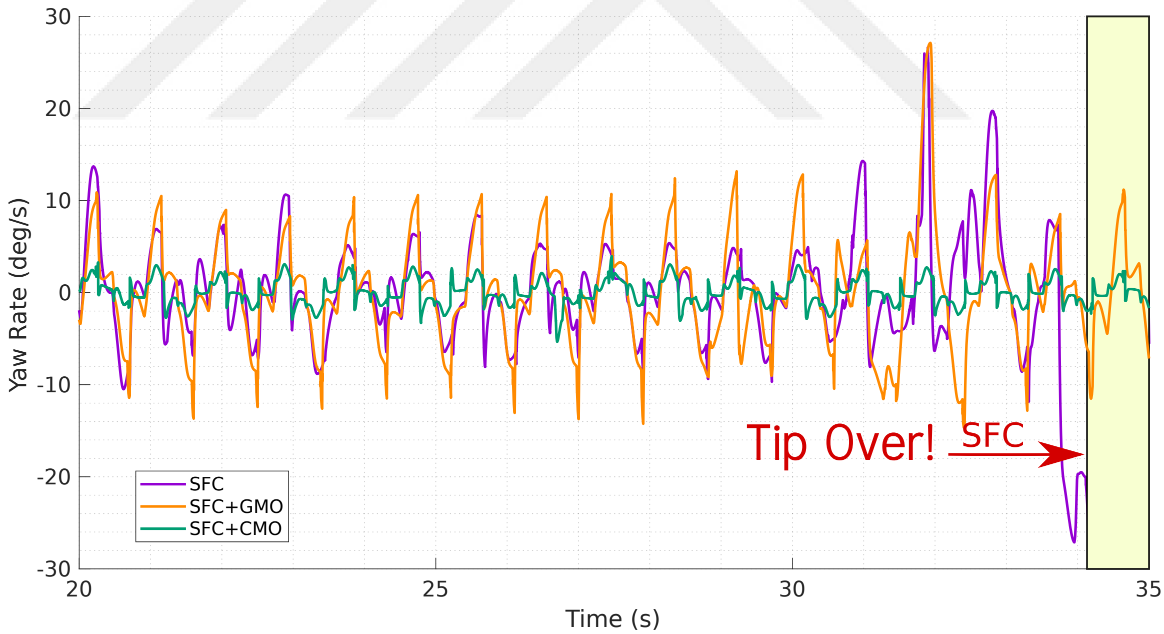

Figure 38: Simulation # 3 results: yaw rate

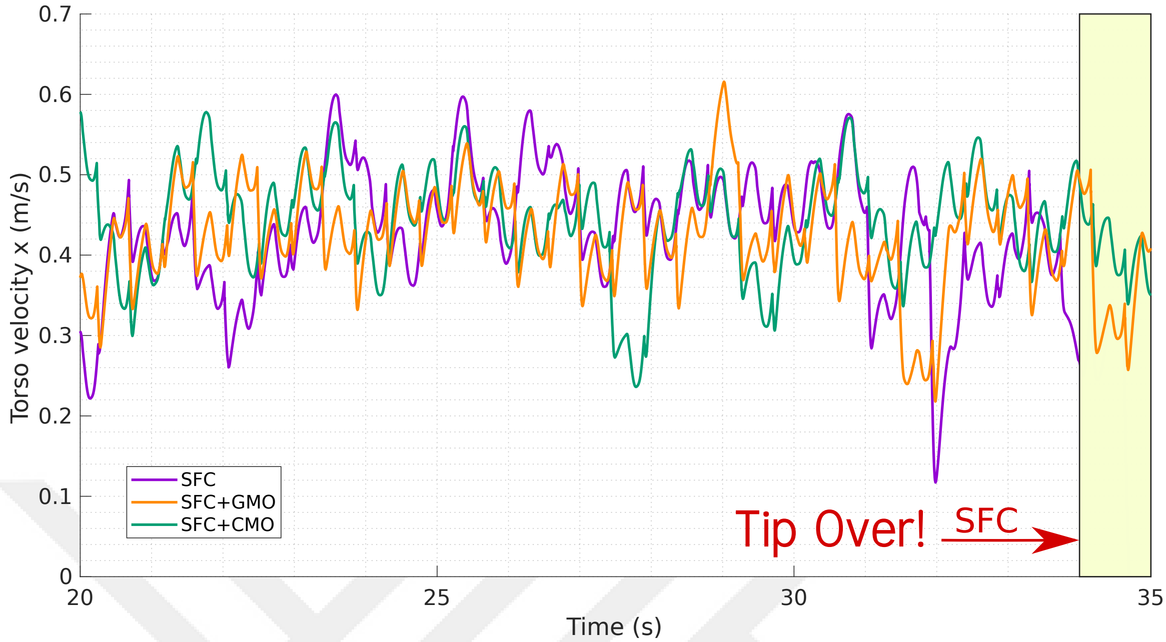

Figure 39: Simulation # 3 results: torso velocity in x axis

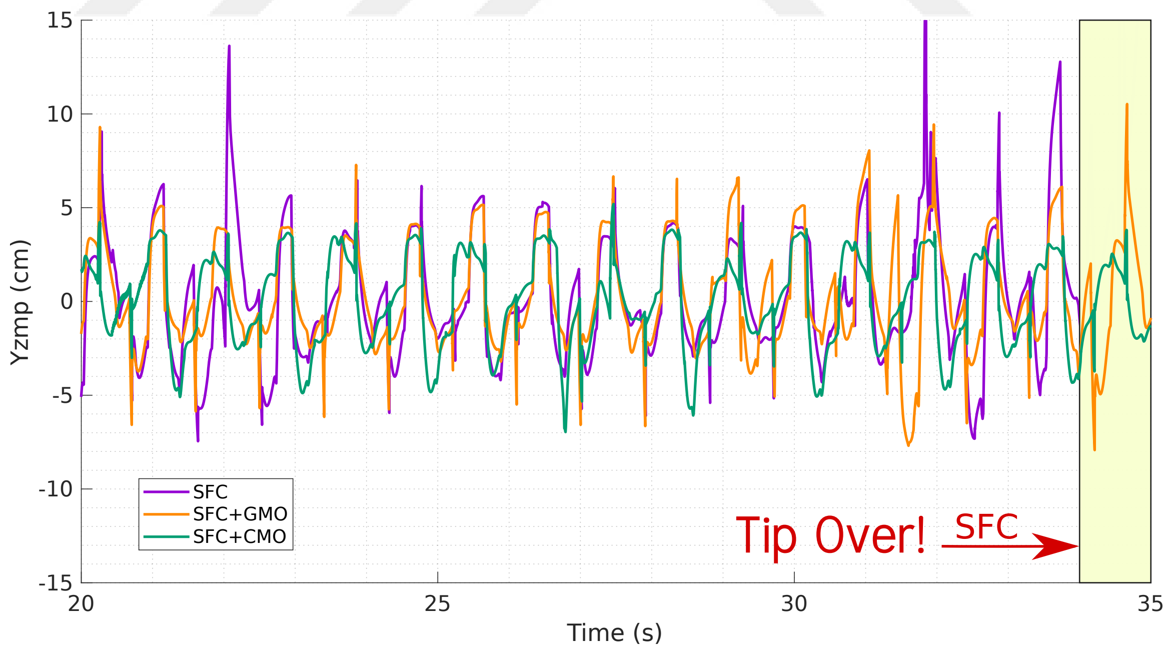

Figure 40: Simulation # 3 results: Y_{zmp} response

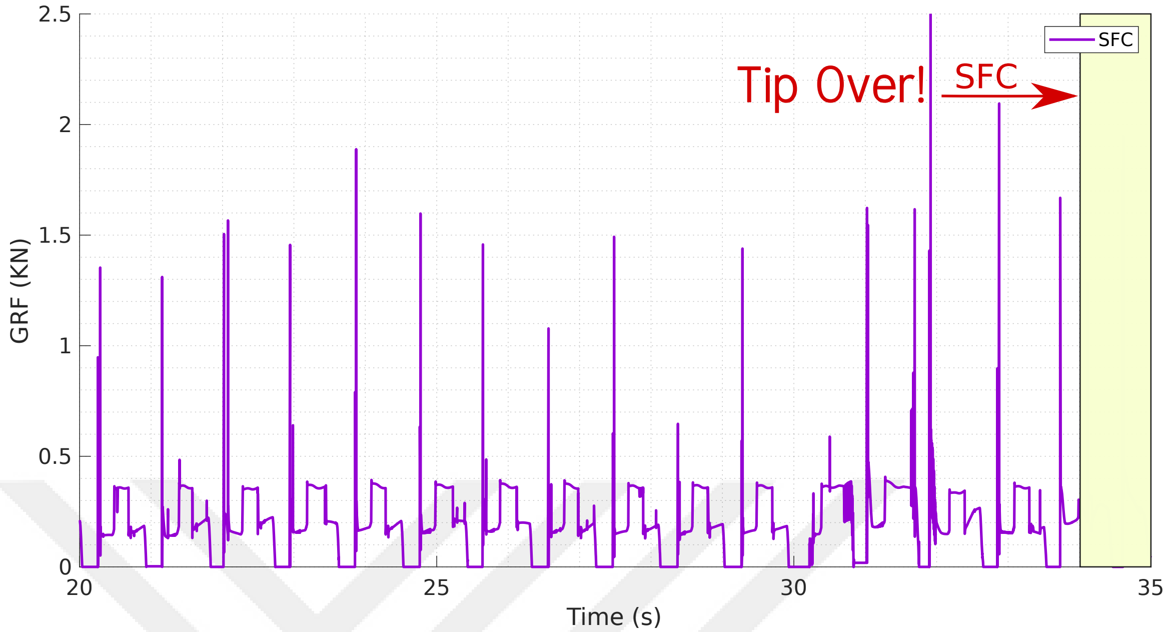

Figure 41: Simulation # 3 results: ground reaction forces for SFC

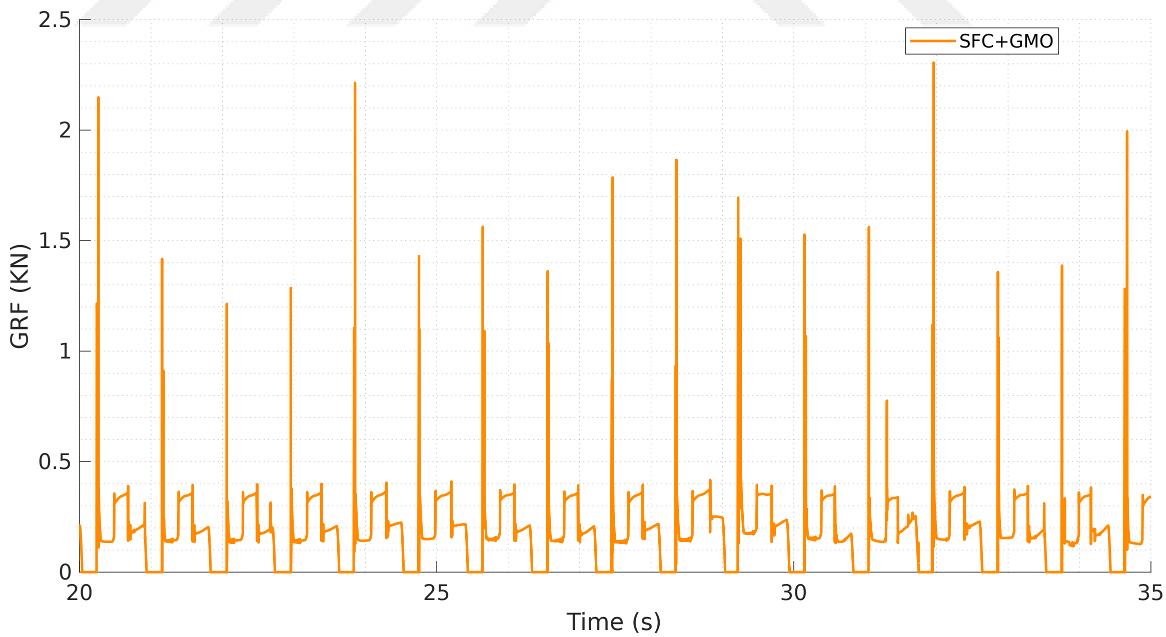

Figure 42: Simulation # 3 results: ground reaction forces for SFC+GMO

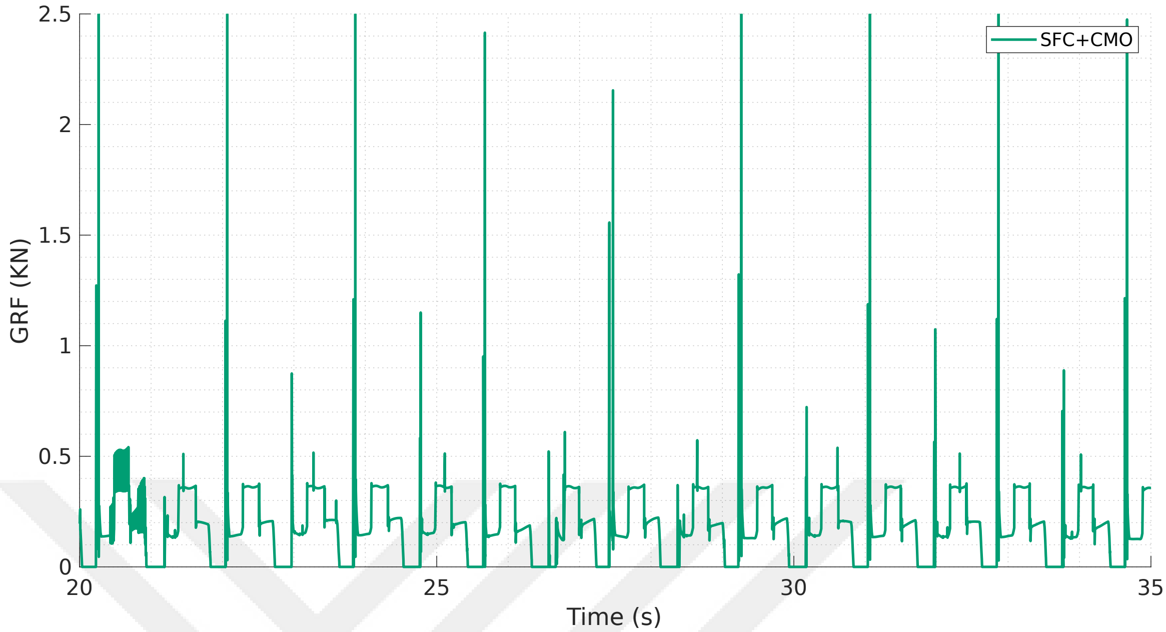

Figure 43: Simulation # 3 results: ground reaction forces for SFC+CMO

4.3.5 Simulation # 4 :Hill Climbing on a Rough Terrain under Disturbance

To observe the effect of the controllers, we conducted the walking simulation on the generated rough terrain map. The CoM mean velocity command of the robot was set as 0.6 [m/s], and during the walking, the 150 [kN] external moment was applied twice to the robot's torso around the x axis. As a result of experiments, all controllers exhibited dynamically coherent locomotion. The simulation results were placed in the Figs. 44, 45, 46, 41, 42 and 43. These results' solid purple, orange, and green lines depict the SFC, SFC+GMO, and CMO data.

Firstly, we presented the yaw rate results in Fig. 44. After perturbations at 6^{th} and 16^{th}seconds, the SFC+CMO controller's yaw rate change was less than the other controllers. Moreover, SFC+CMO's and SFC's yaw rate variations presented similar behavior and were less reactive to perturbations. The RMSE values were 5.11 [deg/s], 6.55[deg/s], and 1.66 [deg/s] for SFC, SFC+GMO, and SFC+CMO,respectively. This means that SFC+CMO is reactive to perturbations even on rough terrain and better

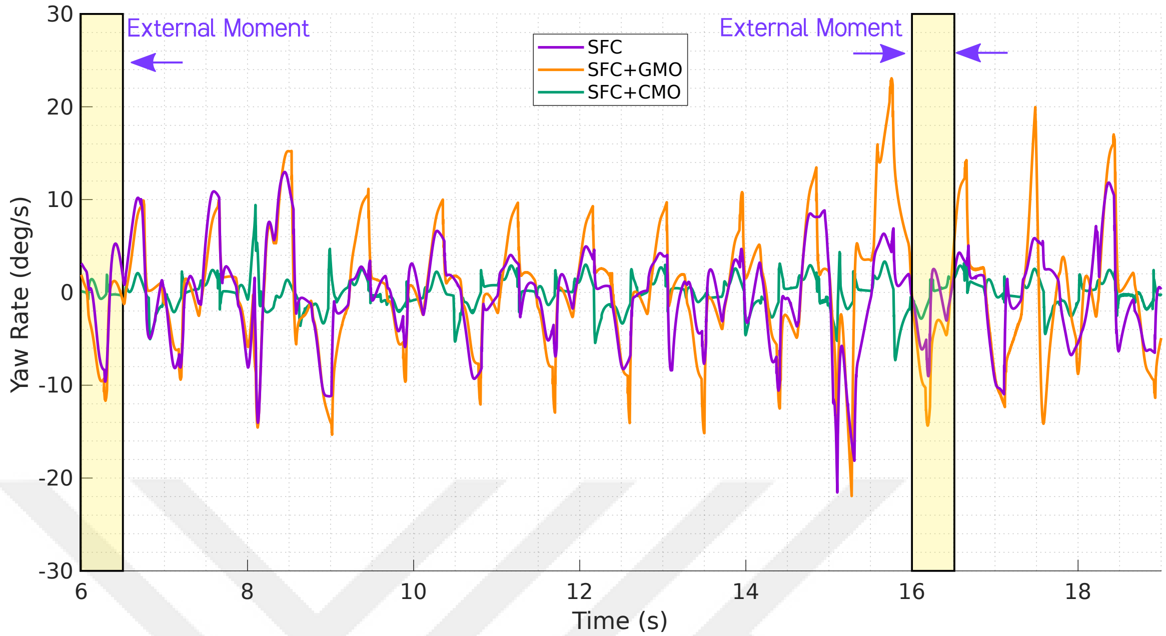

Figure 44: Simulation # 4 results: yaw rate

regulates the robot's posture.

Secondly. the torso velocity along x results were given in Fig 39. In this figure, SFC+CMO deviated less from the target velocity than the other controller. The velocity variation in the SFC+GMO controller case is much higher than in the other controller. Through these results, the RMSEs are 33.9 [cm/s], 31.5 [cm/s], and 13.4 [cm/s] for SFC, SFC+GMO, and SFC+CMO, respectively.

Furthermore, the Y_{zmp} results were given in Fig 46. In these results, the SFC+GMO represented more variation in Y_{zmp}. The Y_{yzm} variation of the SFC+CMO is less without perturbation, but when external moments were applied at 6^{th} and 16^{th} seconds, there were peaks in Y_{zmp} response. The peak-to peaks of Y_{zmp} are 27.92 [cm], 23.30 [cm], and 33.99 [cm] for SFC, SFC+GMO, and SFC+CMO, respectively.

Lastly, we provided the GRF's results in figures 27, 28, and 29 for SFC, SFC+GMO, and SFC+CMO respectively. Through these results, we computed the mean±std of the GRFs are 1.37 ±0.31[kN], 1.90± 0.46 [kN], and 1.60±0.89 [kN] for SFC, SFC+GMO and SFC+CMO, respectively.

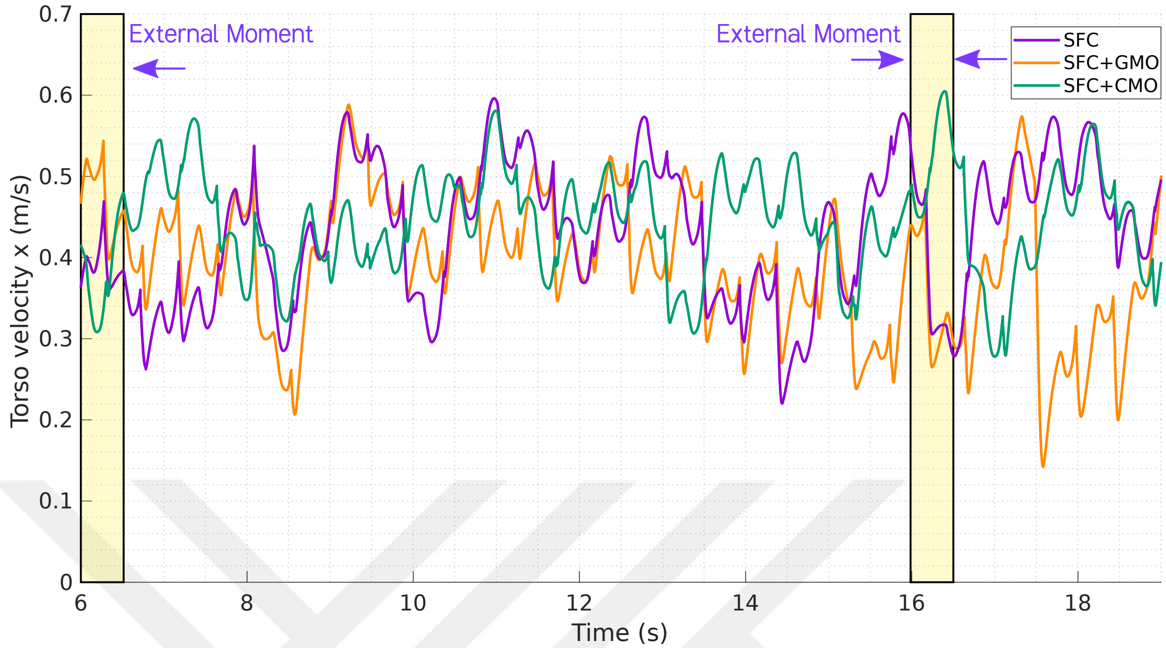

Figure 45: Simulation # 4 results: torso velocity in x axis

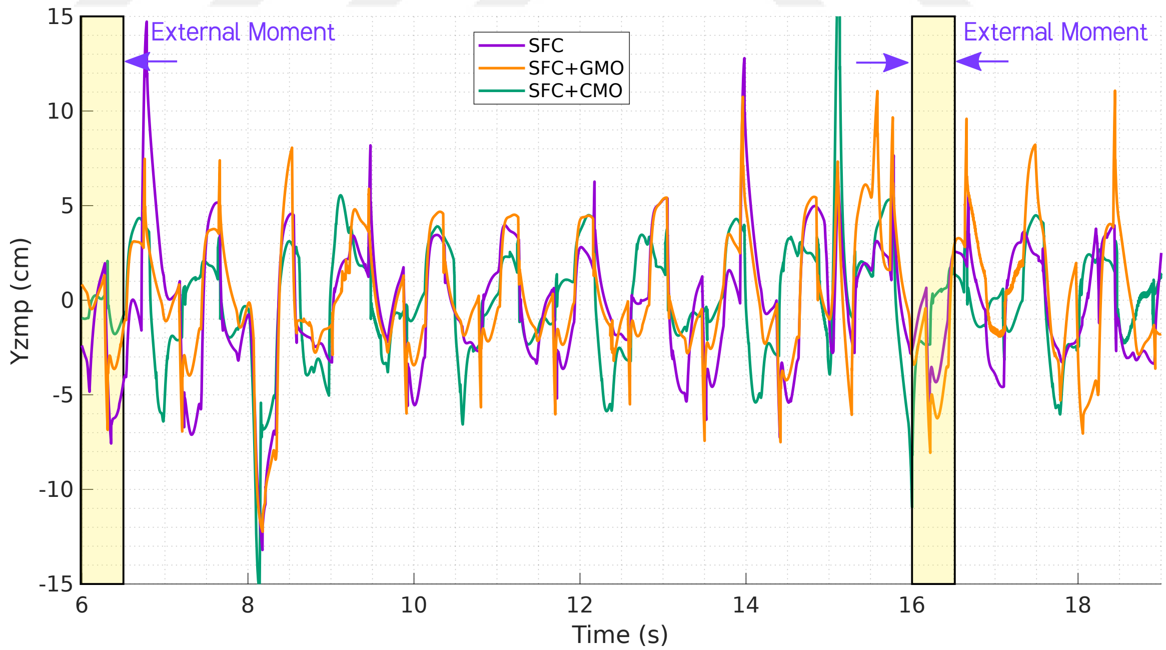

Figure 46: Simulation # 4 results: Y_{zmp} response

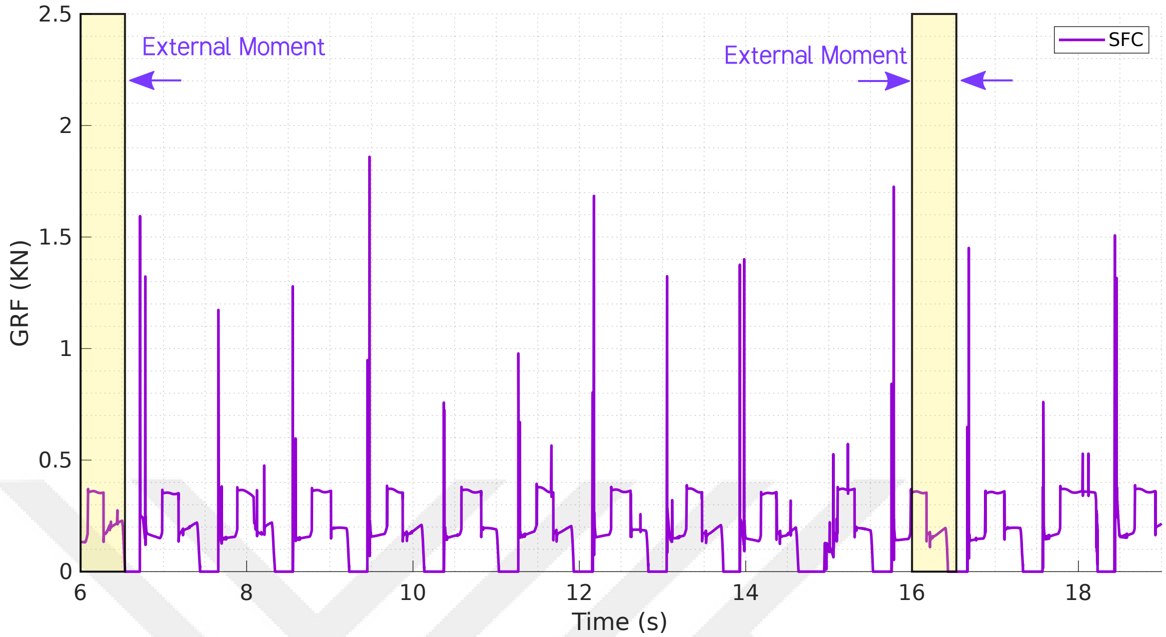

Figure 47: Simulation # 4 results: ground reaction forces for SFC

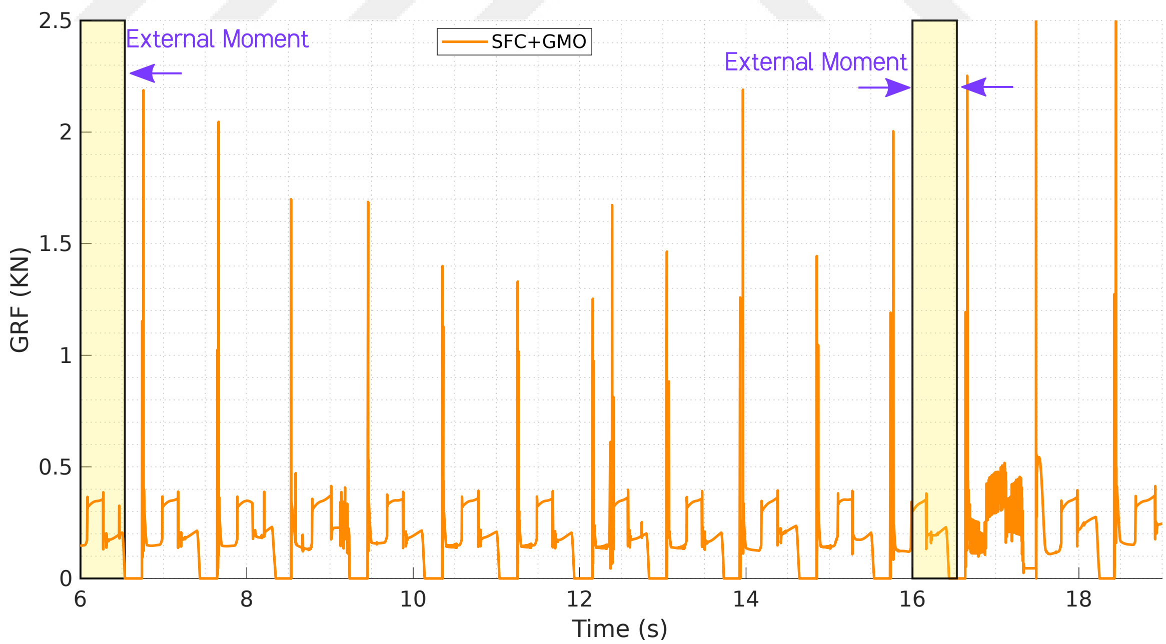

Figure 48: Simulation # 4 results: ground reaction forces for SFC+GMO

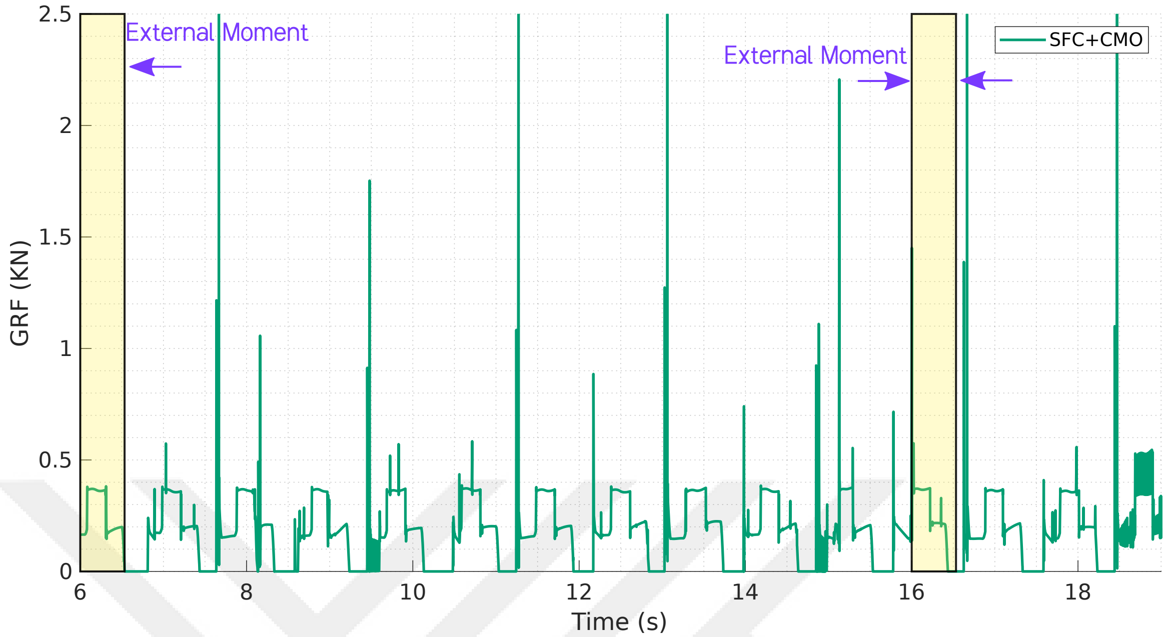

Figure 49: Simulation # 4 results: ground reaction forces for SFC+CMO

4.3.6 Statistical Measurements of the Simulations

To determine the most effective controller, the statistical measurements provided and the metrics are RMSE of yaw rate of the torso (Fig. 50) and torso velocity along x axis (Fig. 51), peak-to-peak values of the Y_{zmp} 52 and mean±std of the GRFs (Fig. 53). In Fig 50, yaw rates RMSE of all scenarios, and these results show that the SFC+CMO controller outperforms the other controllers for regulating yaw rates. In simulations, SFC+CMO yaw rates decreased the 66.5% in Simulation 1, 82% in Simulation 2, 77% in Simulation 3, and 67% in Simulation 4 for SFC+CMO compared to the second-best controller.

Additionally, the torso velocity RMSE values of all scenarios were visualized with the bar graph in Fig. 51. Similarly, SFC+CMO outperforms the other controllers again regarding the torso velocity along x axis tracking except simulation 1. In this scenario, the RMSE value of the SFC is the best, but the velocity variation was much higher than the other controller, as seen in Fig. 25. In simulations, the RMSE

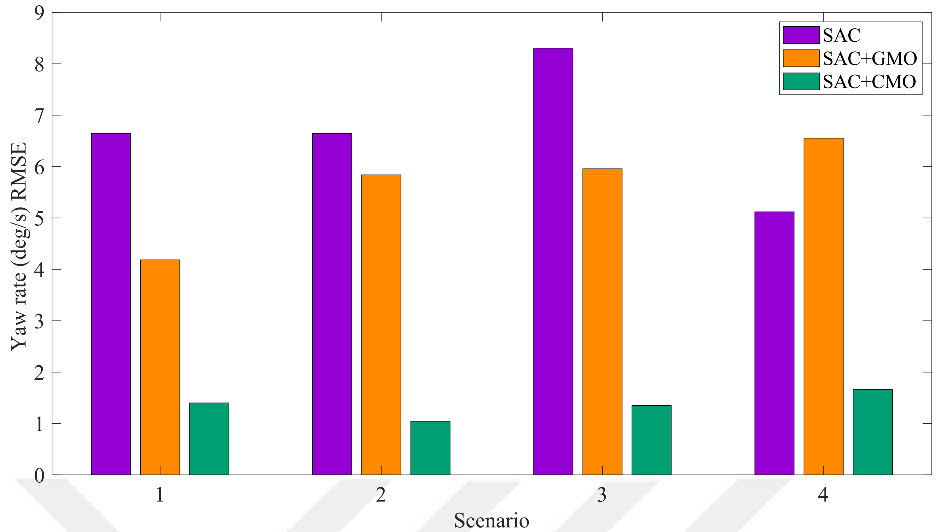

Figure 50: Bar graph of statistical measurements for each scenario: RMSE of the yaw rates

torso velocity along the x axis with the SFC+CMO controllers decreased by 55% in Simulation 2, 59% in Simulation 3, and 58% in Simulation 4 compared to the second-best controller.

Moreover, Y_{zmp} peak-to peaks were presented in Fig 52. Through these, the proposed controller outperforms other controllers in Simulation 1 and 3. In these simulations, Y_{zmp} peak-to-peaks with SFC+CMO decreased by 52% in Simulation 1 and 33% in Simulation 3. In simulations 2 and 4, Y_{zmp} peak-to-peaks increased for the proposed controller. We argue that the proposed controller's main objective is enhancing operational space force tracking. Due to this, the robot is reactive to disturbance, and Y_{zmp} peaks were revealed.

In Fig. 53, the GRFs mean±std of all scenarios were given. This figure shows that the proposed controller outperforms the other controllers except for simulations 3 and 4. The mean±std values of GRFs were decreased with 72%±76% [kN] in simulation 1 and 41%±24% [kN] in simulation 2 [kN]. These results show that the proposed controllers enhance operational space force tracking on rough terrain while increasing GRFs.

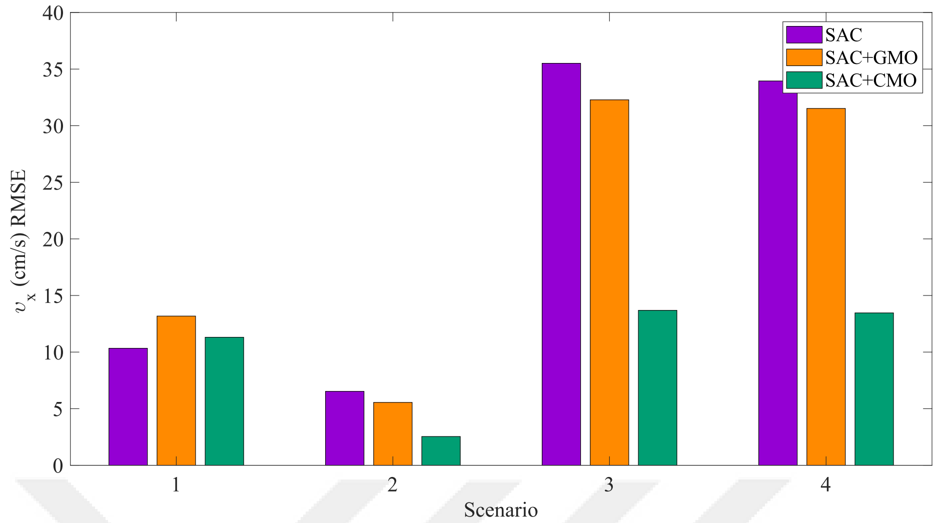

Figure 51: Bar graph of statistical measurements for each scenario: RMSE of the CoM velocity in x axis

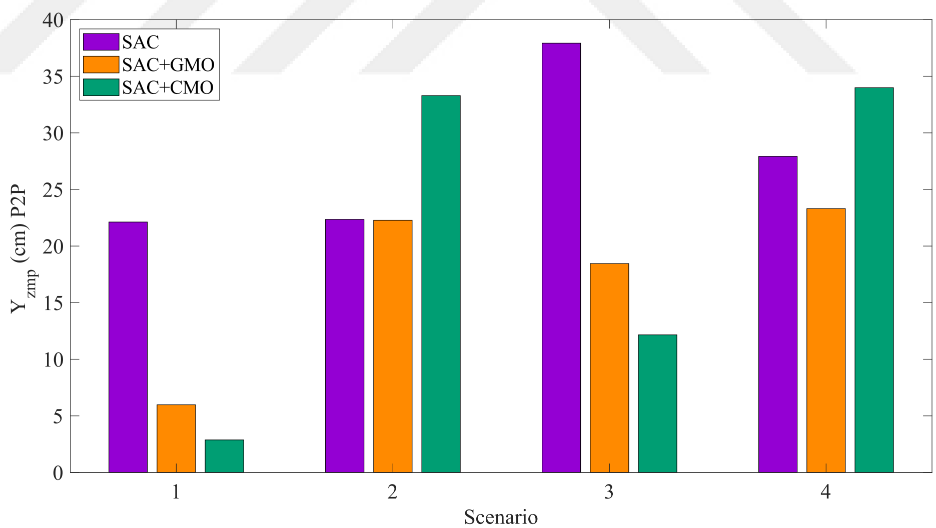

Figure 52: Bar graph of statistical measurements for each scenario: peak-to peak of the Y_{zmp}

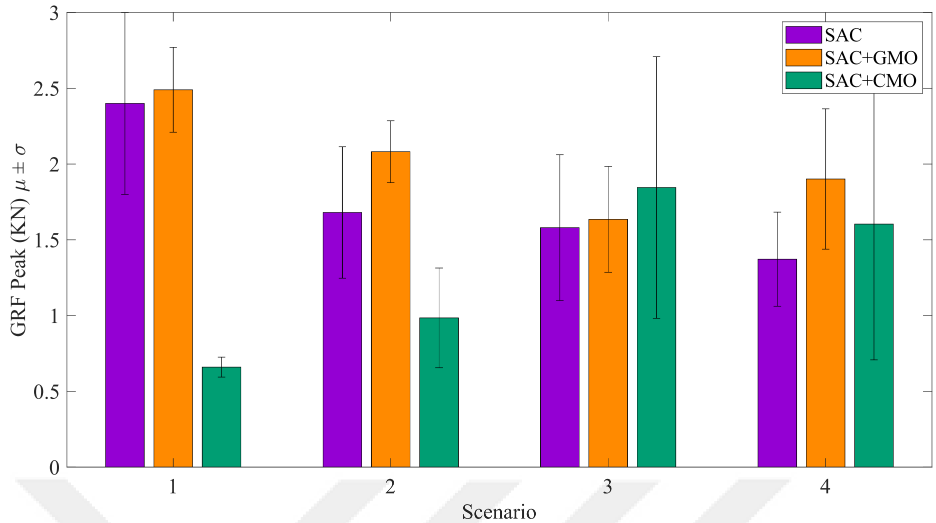

Figure 53: Bar graph of statistical measurements for each scenario: RMS of the GRFs

4.3.7 Experiment: Spot Walking on a Flat Terrain

In this experiment, we commanded the robot spot walking. The robot walked on a determined spot and its locomotion behavior observer. During the experiment, all controllers presented dynamically balanced behavior. The conducted experiments results were presented in Figs 54, 56 and 55. The solid purple and green lines in these figures represent SFC and SFC+CMO, respectively.

In Fig. 54, the torso yaw rate results were given. This figure shows that the torso yaw rate was reduced in the SFC+CMO controller case. Through these results, we computed the RMSE values 5.24 [deg/s] and 3.69 [deg/s]. The proposed controller reduced the RMSE by 29% from these values. This means that our proposed controller enhances posture regulation on flat terrain.

To understand balance behaviuour, we depicted the Y_{zmp} and X_{zmp} results Figs. in 55 and 56. Through these results, we examined X_{zmp} the peak-to-peaks are 93 [cm] and 70 [cm] for SFC and SFC+CMO, respectively. Additionally, the Y_{zmp} peak to peaks are 62 [cm] and 41 [cm] for SFC and SFC+CMO, respectively. For comparison,

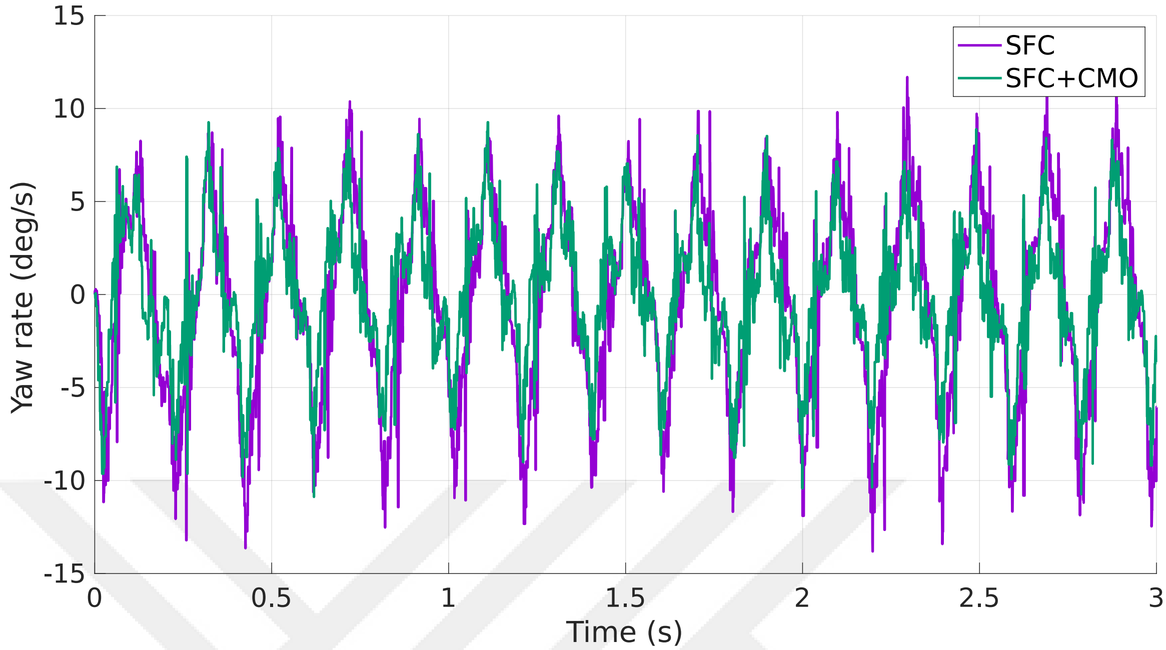

Figure 54: Experiment: Torso Yaw Rate Results

the X_{zmp} peak to peak was lowered by 24%, and the Y_{zmp} peak to peak was reduced by 33% with the proposed controller. This means that our proposed controller improves the balance of the robot.

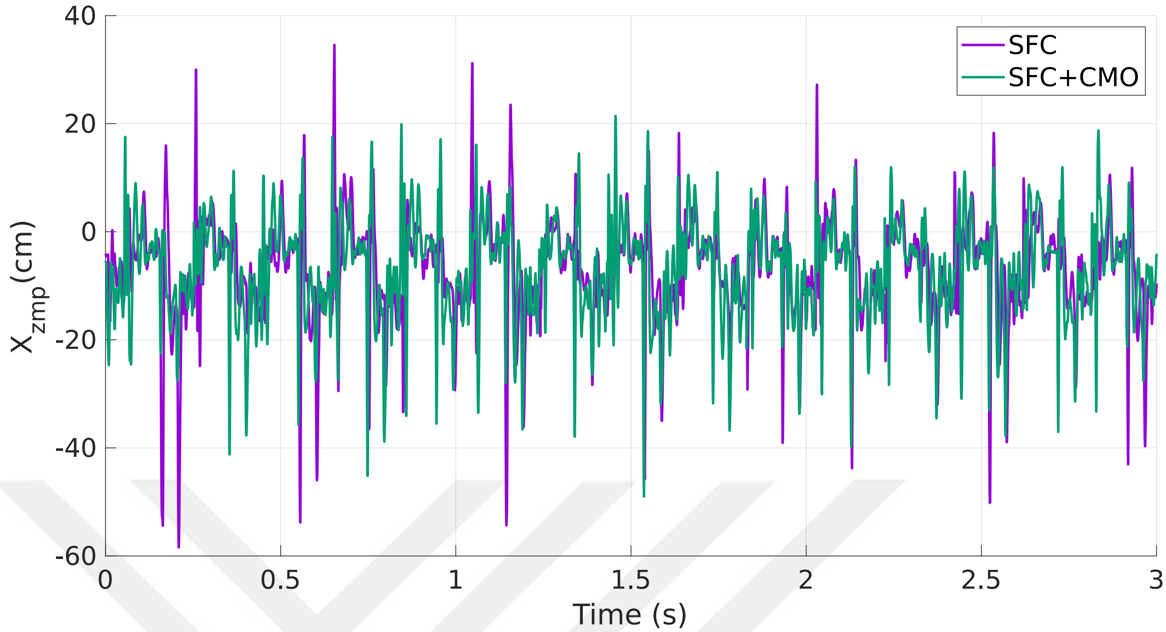

Figure 55: Experiment: X_{zmp} Results

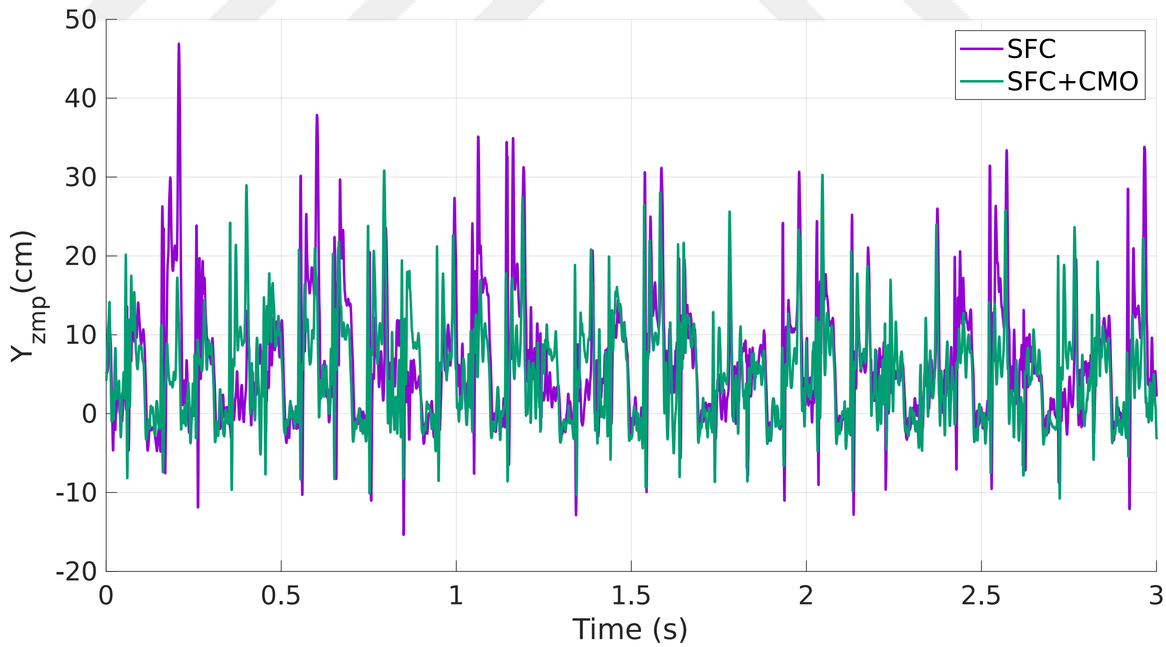

Figure 56: Experiment: Y_{zmp} Results

CHAPTER V

CONCLUSIONS

This thesis presented the locomotion framework for quadruped robots to maintain dynamically balanced locomotion, and it consisted of two parts: trajectory generation and locomotion controller. The simulation experiments were conducted on the Raisim simulator to validate the proposed approaches. Firstly, we presented a preview control-based trajectory generation approach of the quadruped robot for varying CoM height to accomplish agile locomotion tasks, namely, jumping and spot-jogging. This proposed approach utilized the preview control concept, and we provided the compact derivation of the gain calculation process. The proposed trajectory approach considerably diminishes all scenarios' ZMP errors and GRF. In scenario 1, torso pitch fluctuation is decreased compared to the other approaches, while in scenario 2, torso pitch fluctuation is less for polynomial-based trajectory. The sinusoidal trajectory operated well for jumping but performed poorly for spot-jogging because the trajectory parameter tuning process requires effort and uncertain contact phases.

Secondly, the CMO-based force controller proposed to enhance locomotion behavior on all terrain and under perturbations. Additionally, the controller includes naive SFC to enhance the operational space control. For comparison reasons, we implemented a GMO-based controller on the simulations. We tested the three selected controllers in simulations: SFC, SFC+GMO, and SFC+CMO. These controllers were tested in various cases: slippery terrain, high-speed trot walking, under perturbation, and rough terrain trot walking. From these cases, the simulations' statistical measurements were presented. From these results, the SFC+CMO outperforms other controllers in terms of posture regulation and operational space tracking performance.

This means that the GMO-based controllers presented unsatisfactory results compared to the CMO-based controllers. We argue that GMO-based controllers provide the ability to observe floating base momentum, but they do not provide the ability to exert floating base momentum as a joint torque. These require torque mapping from the joint space to the operational space, which is computationally redundant. In addition, the gain tunning process is complicated for GMO compared to CMO. Additionally, we derived the comprehensive stability analysis of the CMO with computed torque control. Furthermore, the CMO derivation process is presented by utilizing the auxiliary variable-based observer design approach.

Thirdly, the actual robot hardware experiment was conducted to test the devised controller. Initially, we implemented the presented friction compensation approach to enhance the joint position controller tracking. In experiment, The robot was executed spot walking on the flat terrain. This experiment provided the preliminary result of the proposed locomotion controller and SFC. Given these results, the proposed controller outperforms the SFC by looking at the statistical metrics: RMSE of X_{zmp}, Y_{zmp}, and yaw rates. We will conduct more experiments to test the proposed locomotion control framework in our future work.

REFERENCES

[1] B. Ugurlu and A. Kawamura, "Zmp-based online jumping pattern generation for a one-legged robot," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1701–1709, 2010.

[2] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10822–10825, 2008. 17th IFAC World Congress.

[3] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2019.

[4] Q. Li, L. Qian, S. Wang, P. Sun, and X. Luo, "Towards generation and transition of diverse gaits for quadrupedal robots based on trajectory optimization and whole-body impedance control," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2389–2396, 2023.

[5] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized jumping on the mit cheetah 3 robot," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7448–7454, 2019.

[6] O. Cebe, C. Tiseo, G. Xin, H.-c. Lin, J. Smith, and M. Mistry, "Online dynamic trajectory optimization and control for a quadruped robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12773–12779, 2021.

[7] 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 *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, IEEE.

[8] S. Shimmyo, T. Sato, and K. Ohnishi, "Biped walking pattern generation by using preview control based on three-mass model," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5137–5147, 2013.

[9] M. Murooka, M. Morisawa, and F. Kanehiro, "Centroidal trajectory generation and stabilization based on preview control for humanoid multi-contact motion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8225–8232, 2022.

[10] K. Ryu, J. Yoo, J. Back, and I.-W. Park, "Preview control-based online walking pattern generation for biped robots with vertical center-of-mass motion," *International Journal of Precision Engineering and Manufacturing*, vol. 21, pp. 1653–1661, July 2020.

[11] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *2008 IEEE International Conference on Robotics and Automation*, pp. 819–824, 2008.

[12] G. Bellegarda and A. Ijspeert, "CPG-RL: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.

[13] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, "Stance: Locomotion adaptation over soft terrain," *IEEE Transactions on Robotics*, vol. 36, p. 443–457, Apr. 2020.

[14] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[15] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252, 2018.

[16] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2006.

[17] C. Gong, F. Zhao, Z. Liao, T. Tao, X. Wang, and X. Mei, "Multi-contact cartesian null-space impedance control for the anthropomorphic manipulator without knowledge of force locations," *IEEE Robotics and Automation Letters*, p. 1–8, 2024.

[18] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018.

[19] Z. Zhu, G. Zhang, Z. Sun, T. Chen, X. Rong, A. Xie, and Y. Li, "Proprioceptive-based whole-body disturbance rejection control for dynamic motions in legged robots," *IEEE Robotics and Automation Letters*, vol. 8, p. 7703–7710, Nov. 2023.

[20] J. An and I. Lee, "Artificial neural network-based ground reaction force estimation and learning for dynamic-legged robot systems," *PeerJ Computer Science*, vol. 9, p. e1720, Dec. 2023.

[21] 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.

[22] E. C. Ozcinar, *Robust locomotion control for a quadruped robot with robust locomotion control for a quadruped robot with high payload capacity.* Master's thesis, Ozyegin University, Istanbul, Türkiye, June 2022.

[23] B. Ugurlu, I. Havoutis, C. Semini, K. Kayamori, D. Caldwell, and T. Narikiyo, "Pattern generation and compliant feedback control for quadrupedal dynamic trot-walking locomotion: experiments on robocat-1 and hyq," *Autonomous Robots*, vol. 38, 02 2015.

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

[25] J. J. Craig, *Introduction to Robotics: Mechanics and Control.* USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1989.

[26] 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.

[27] T. Katayama, T. Ohki, T. Inoue, and T. Kato, "Design of an optimal controller for a discrete-time system subject to previewable demand," *International Journal of Control*, vol. 41, pp. 677–699, 1985.

[28] B. Ugurlu, J. A. Saglia, N. G. Tsagarakis, and D. G. Caldwell, "Hopping at the resonance frequency: A trajectory generation technique for bipedal robots with elastic joints," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012.

[29] B. Ugurlu, J. A. Saglia, N. G. Tsagarakis, S. Morfey, and D. G. Caldwell, "Bipedal hopping pattern generation for passively compliant humanoids: Exploiting the resonance," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5431–5443, 2014.

[30] B. Ozkaynak and B. Ugurlu, "Preview control-based jumping and spot-jogging trajectory generation for quadruped robots," in *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, IEEE, July 2023.

[31] M. S. Fadali and A. Visioli, "Chapter 10 - optimal control," in *Digital Control Engineering (Second Edition)* (M. S. Fadali and A. Visioli, eds.), pp. 399–438, Boston: Academic Press, second edition ed., 2013.

[32] D. A. Bini, B. Iannazzo, and B. Meini, *Numerical Solution of Algebraic Riccati Equations.* Society for Industrial and Applied Mathematics, 2011.

[33] B. Ugurlu, K. Kotaka, and T. Narikiyo, "Actively-compliant locomotion control on rough terrain: Cyclic jumping and trotting experiments on a stiff-by-nature quadruped," in *2013 IEEE international conference on robotics and automation*, pp. 3313–3320, IEEE, 2013.

[34] D. Y. Oral, D. E. Barkana, and B. Ugurlu, "Centroidal momentum observer: Towards whole-body robust control of legged robots subject to uncertainties," in *2022 IEEE 17th International Conference on Advanced Motion Control (AMC)*, IEEE, Feb. 2022.

[35] J. Yuan, "Closed-loop manipulator control using quaternion feedback," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 434–440, 1988.

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

[37] E. Sariyildiz, H. Yu, T. Nozaki, and T. Murakami, "A robust state-space controller design for multi-mass resonant systems," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, Oct. 2016.

[38] E. Sariyildiz, R. Mutlu, and C. Zhang, "Active disturbance rejection based robust trajectory tracking controller design in state space," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, Mar. 2019.

[39] 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, p. 2042–2053, Mar. 2020.

[40] E. Sariyildiz and K. Ohnishi, "An adaptive reaction force observer design," *IEEE/ASME Transactions on Mechatronics*, vol. 20, p. 750–760, Apr. 2015.

[41] J. White, D. Swart, and C. Hubicki, "Force-based control of bipedal balancing on dynamic terrain with the "tallahassee cassie" robotic platform," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020.

[42] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, pp. 129–143, Feb. 2001.

[43] C. D. Remy, M. Hutter, M. Hoepflinger, M. Bloesch, C. Gehring, and R. Siegwart, "Quadrupedal robots with stiff and compliant actuation," *Automatisierungstechnik*, vol. 60, pp. 682–691, Nov. 2012.

[44] E. C. Ozcinar, O. Bebek, and B. Ugurlu, "Contact force distribution using centroidal momentum feedback for quadruped locomotion," in *2023 IEEE International Conference on Mechatronics (ICM)*, IEEE, Mar. 2023.

[45] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pp. 22–27, IEEE, 2008.

[46] J. Nakanishi, M. Mistry, J. Peters, and S. Schaal, “Towards compliant humanoids-an experimental assessment of suitable task space position/orientation controllers,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2007.

[47] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” *CoRR*, vol. abs/1909.06586, 2019.

[48] B. Ugurlu, M. Nishimura, K. Hyodo, M. Kawanishi, and T. Narikiyo, “Proof of concept for robot-aided upper limb rehabilitation using disturbance observers,” *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 110–118, 2015.

[49] G. Guennebaud, B. Jacob, *et al.*, “Eigen v3.” http://eigen.tuxfamily.org, 2010.

[50] K. Perlin, “An image synthesizer,” in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’85*, SIGGRAPH ’85, p. 287–296, ACM Press, 1985.

VITA

İbrahim Burak Özkaynak had graduated from the Dilnihat Ozyegin High School. He received a bachelor's Degree in Mechatronics Engineering at Bahçeşehir Üniversity in 2020. He is an M.Sc. student at Ozyegin University and working at the Biomechatronics Laboratory. His research interest is motion planning and locomotion control of legged systems.