TECHNISCHE UNIVERSITÄT WIEN

ACIN
AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

DIPLOMA THESIS

Optimizing Footstep Placement for Robust and Stable Running of Humanoid Robots

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Univ.-Prof. Dr.-Ing. Christian Ott
Projektass. Tobias Egle MSc

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
David Demattio
Matriculation number 11713047

Vienna, October 2024

Robotic Systems Lab (RSL)
A-1040 Wien, Gußhausstr. 27–29, Internet: https://www.acin.tuwien.ac.at

Abstract

Bipedal legged locomotion offers improved accessibility and the ability to navigate complex, non-barrier-free environments more effectively than wheel-based systems. However, this increased versatility, particularly in humanoid robots, comes at the cost of slower movement. Based on existing methods for humanoid running, this work extends the analytical concept for biologically inspired center of mass (CoM) trajectory planning with optimal footstep and center of pressure adaptation. The analytical method allows for a planner design, where the desired footsteps are input parameters and the current footstep is adjusted to stabilize the motion. This forces the controller to select suboptimal footstep placements when kinematic constraints between the current and next previewed footsteps are reached. The existing trajectory generation algorithms, developed at the German Aerospace Center (DLR), were adapted and extended by optimizing all previewed footsteps to avoid kinematic limitations and increase robustness to disturbances. During stance, the center of pressure (CoP) is optimally adjusted within the contact area of the stance foot, the optimal adaptation of the footstep is executed in flight. These two strategies combined, enable high robustness to challenging starting robot configurations, allowing locomotion transitions, such as walking to running.
The trajectory planning algorithm is integrated into an inverse-dynamics based whole-body controller and validated in simulations with the humanoid robot Kangaroo.

Kurzfassung

Humanoide Roboter bieten im Vergleich zu radbasierten Robotersysteme Fortbewegungsmöglichkeiten in komplexeren und unwegsamen Umgebungen, jedoch mit einer geringerer Fortbewegungsgeschwindigkeit.

In dieser Arbeit wird auf Grundlage bestehender Methoden eine biologisch inspirierte Trajektorienplanung für den Körperschwerpunkt (Center of Mass, CoM) mittels polynomialer Splines angepasst und um optimierte Parameter erweitert. Die Methode basiert auf einem rein analytischen Reglerentwurf, bei dem die gewünschten Fußschrittpositionen als Eingabeparameter dienen. Dabei wird nur der aktuelle Fußschritt modifiziert, um die Bewegung zu stabilisieren. Dies zwingt den Regler dazu, suboptimale Schrittplatzierungen zu wählen, wenn kinematische Einschränkungen zwischen dem aktuellen und dem nächsten vorausberechneten Schritt auftreten.

Die am Deutschen Zentrum für Luft- und Raumfahrt (DLR) entwickelten Methoden bilden die Grundlage der Arbeit. Das Ziel der Arbeit ist es, den Planungsansatz sowhol durch die Optimierung einer endlichen Anzahl vorausberechneter Schritte als auch durch die optimale Positionierung des Druckmittelpunkts (Center of Pressure, CoP) im Standfuß zu erweitern. Dies soll helfen, kinematische Einschränkungen zu vermeiden und die Robustheit gegenüber Störungen zu erhöhen. Die Kombination dieser beiden Strategien ermöglicht die Konvergenz zu einer stabilen Laufbewegung aus einem größeren Bereich von Anfangsbedingungen, wie zum Beispiel den Übergang vom Stehen und Gehen zum Laufen.

Die Methode wird durch eine Ganzkörperregelung (Whole-Body Control, WBC) im Aufgabenraum mit dem humanoiden Roboter Kangaroo in einer Simulation validiert.

Contents

1 Introduction

In recent years, the rise of automation has significantly transformed various industries, with robots playing an increasingly crucial role in assisting humans across multiple fields. Robots are designed and tailored to perform specific tasks, ranging from industrial manufacturing [1] to healthcare [2], and their forms and functions often depend on the requirements of the tasks they are intended to accomplish. Often, mobile-based systems are used to cover a large workspace. In research significant progress has been made in replicating human-like gaits using robot locomotion. While wheeled robots are ideal for smooth and controlled environments, legged robots have captured interest due to their ability to navigate complex, human-centric terrains. Despite these advances, current robots are still a long way from achieving the capabilities of human movement. Humans can seamlessly adjust their gait to different speeds and surfaces and can maintain stability under challenging conditions. Such abilities remain difficult to replicate in robotic systems. For navigating complex terrains and achieving versatile movement, legged locomotion is generally favored over wheel-based systems.

Bipedal humanoids can overcome stairs, narrow passages, and uneven terrain with versatility and agility that more closely resemble human movement. This versatility often comes with the drawback that humanoid robots move very slowly. Additionally, the dynamic locomotion mode of running features short contact phases that are potentially underactuated and require substantial torque demands on the robot´s joints. Thus, running motions with bipedal humanoid robots are a challenging field where distinct mathematical models and control strategies based on the center of mass (CoM) dynamics are used to replicate human-like running gaits.

1.1 Contribution

This thesis aims to develop a robust planning approach for humanoid robot running. The work is based on planning approaches for running [3, 4], developed at the Institute of Mechatronics and Robotics at the German Aerospace Center (DLR), and their extension for walking to running transitions [5], developed at the Automation and Control Institute (ACIN) at the Vienna University of Technology. The goal of this thesis is to improve the robustness of the planning approach by optimizing several preview footsteps and the position of the center-of-pressure (CoP) in the stance foot. This enables the robot to avoid kinematic limitations and withstand strong perturbations. Furthermore, the proposed method enables the robot to recover from challenging initial conditions and achieve a stable running motion, e.g., in the transition from standing and walking to running.

1.2 Content outline

The thesis is organized as follows. Chapter 2 summarizes the related work in the field of legged humanoid locomotion, with a focus on running locomotion methods. In chapter 3 the base for this work is introduced. Furthermore, the mathematical model and planning algorithms are presented, separately for horizontal and vertical planning. Additionally, the online planning structure is introduced.

Chapter 4 extends the purely analytical planning method with optimal footstep placement and optimal CoP adaptation to increase the robustness of the planner while respecting kinematic limitations. In Chapter 5 the planning method is evaluated using a simulated point-mass model.

The whole-body control framework is derived in Chapter 6. Subsequently, the integration of the proposed planning method in the whole-body control framework is evaluated in Chapter 7 using the physics simulator MuJoCo [6] with the humanoid robot Kangaroo [7]. Chapter 8 provides the discussion and concludes the work.

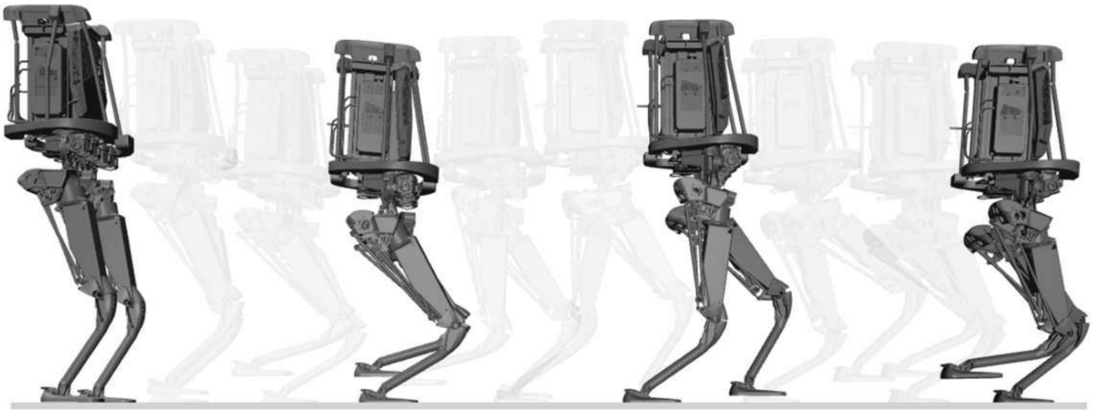


Figure 1.1: Timeseries of humanoid robot Kangaroo [7] while running.

2 Related work

In addition to the growing popularity of reinforcement learning (RL) approaches [8, 9], a common method for implementing robotic bipedal locomotion is to use the center of mass (CoM) dynamics to describe specific locomotion gaits. The CoM captures the most important features of a running gait without considering complex robot structures. Due to unilateral constraints, not every CoM trajectory is feasible. Robots with a mobile base are generally described with a free-floating model, thus the robot can only interact with its environment through its feet, assuming a stable motion. Without considering rotational inertia, all external forces are constrained to pass through the foot´s support polygon, spanned by the contact points on the ground. These conditions are met with various models for different gaits.

Many walking strategies are based on the Linear Inverted Pendulum (LIP) model [10, 11]. In this model, external forces are represented by a focal point that corresponds to the torque-free base joint of the LIP. The direction of the resulting external force is determined by the relationship between the CoM and the base joint of the LIP on the ground. Takenaka et al. [12] introduced the Divergent Component of Motion (DCM) as a new state variable, which separates the CoM dynamics into stable and unstable components. Unlike the LIP model, the DCM does not approximate the robot model. Englsberger et al. [13] extended this approach to 3D.

The generation of running motions is typically based on the Spring-Loaded Inverted Pendulum (SLIP) model, which extends the LIP model with a massless, compliant leg [14]. Given suitable parameters and starting conditions, the SLIP model can exhibit open-loop stability [15] or be controlled to achieve a stable limit cycle [16, 17]. The SLIP model's dynamics are nonlinear, which means they cannot be analytically solved to produce closed-form solutions. Consequently, this model is not ideal for producing running trajectories in aperiodic locomotion with predefined footstep placement and timing. Sovukluk et al. [18] extends the SLIP model with a foot, enabling center of pressure (CoP) adaptation.

Englsberger et al. [3] proposed a method to replicate human-like ground reaction forces using polynomial splines to create CoM trajectories. These trajectories are stabilized by a Biologically Inspired Deadbeat (BID) controller, which enables humanoid running motions in simulations [3]. The BID method shows promising running trajectories and good robustness to disturbances using footstep adaptation. Egle et al. [5] extended this approach with a purely analytical planning method for the transition between BID-based running and DCM-based walking. An alternative approach by Mesesan et al. [19] presents a 3D-DCM framework enabling walking, running, and jumping transitions.

3 Center-of-Mass Trajectory Planning

The running algorithm presented in this chapter was first introduced by Englsberger et al. [3] with the biologically inspired deadbeat (BID) control method, and then extended by Egle et al. [5] for the transition between walking and running. The BID method shows promising results in simulation for robust and stable online running given specific initial conditions, i.e. initial CoM velocities. Additionally, the robustness of the controller is limited by only adapting for the next footstep. Therefore, the BID is not feasible for challenging initial conditions, such as a standstill. This work adapts the existing biologically inspired running methods [3, 5] with an extension for optimal n-preview footstep adaptation and CoP adaptation. Both methods use a biologically inspired CoM trajectory generation for humanoid running locomotion.
Human running experiments [3] show specific ground reaction force (GRF) profiles while running on a force plate treadmill, that can be approximated using polynomial splines, as shown in Figure 3.1. The vertical and horizontal components of the GRF are approximated with a 2nd- and 3rd-order polynomial, respectively. Except for the impact dynamics, the approximation is accurate.

The total force $\boldsymbol{F}_{\text{com}} \in \mathbb{R}^3$ acting on the robot´s CoM $\boldsymbol{x} \in \mathbb{R}^3$ is computed using the leg-force $\boldsymbol{F}_{\text{leg}} \in \mathbb{R}^3$ (corresponding to the GRF profiles) and the gravitational force $\boldsymbol{F}_g \in \mathbb{R}^3$ as

$$\boldsymbol{F}_{\text{com}} = \boldsymbol{F}_{\text{leg}} + \boldsymbol{F}_g = \boldsymbol{F}_{\text{leg}} + m\boldsymbol{g}, \tag{3.1}$$

where m is the robot´s total mass and $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ is the gravitational acceleration. The acceleration of the CoM $\ddot{\boldsymbol{x}}$ is expressed using Newton´s second law and the total force

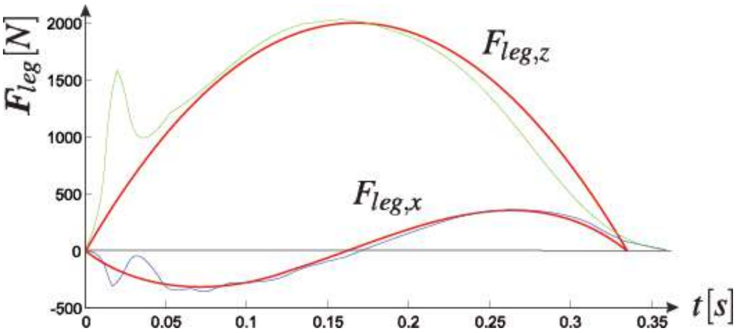


Figure 3.1: Human running experiments on a force plate treadmill [3]. The human leg forces are approximated with 2nd- and 3rd-order spline polynomials (red lines), for vertical (green line) and horizontal (blue line) directions.

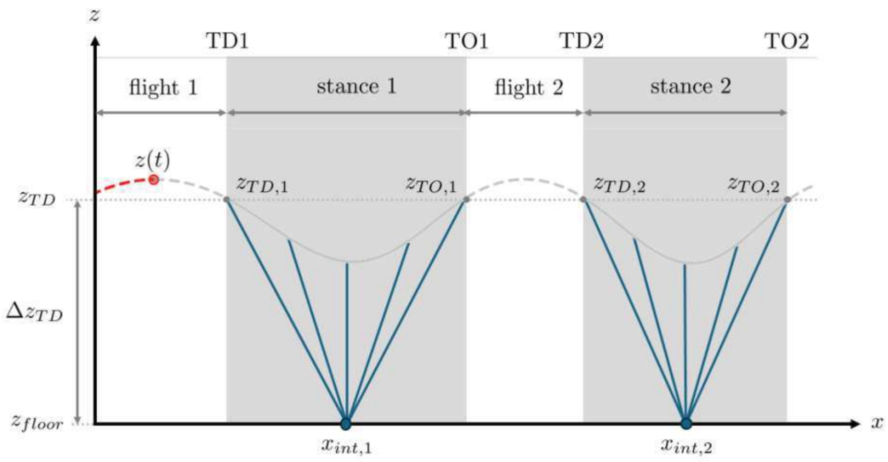


Figure 3.2: Vertical CoM trajectory planning for two preview phases. The CoM trajectory during stance is projected to the corresponding force focus point $x_{int,i}$ using leg forces.

acting on the CoM as

$$\ddot{\boldsymbol{x}} = \frac{\boldsymbol{F}_{\text{com}}}{m}. \tag{3.2}$$

The constant offset from $\boldsymbol{F}_{\text{leg}}$ to $\boldsymbol{F}_{\text{com}}$, allows us to express the CoM acceleration using the previously defined polynomial splines. Therefore the vertical and horizontal CoM positions $\boldsymbol{x}$ can be approximated using 4th- and 5th-order polynomial splines. The CoM velocities $\dot{\boldsymbol{x}}$ are approximated using 3rd- and 4th-order polynomials, accordingly.

Generally, one running motion can be separated into two phases, the stance phase and the flight phase. The planning method uses at least two preview footsteps, denoted as $n \geq 2$. Figure 3.2 shows the vertical position of the CoM for two preview phases ($n = 2$), where z_{TD} is the touch-down (TD) height and z_{TO} is the take-off (TO) height. In the stance phase, the robot is in single-leg contact with the floor, therefore the robot´s CoM is actuated via the contact force $\boldsymbol{F}_{\text{leg}}$ on the ground. The CoM trajectory is planned using polynomial splines with boundary conditions that ensure stable running.

On the other hand, in the flight phase, the robot is under-actuated and follows the natural dynamics of a parabolic flight curve. At a given time t, the robot´s CoM position $\boldsymbol{x}(t)$, velocity $\dot{\boldsymbol{x}}(t)$ and acceleration $\ddot{\boldsymbol{x}}(t)$ in flight are defined as

$$\begin{bmatrix} \boldsymbol{x}(t) \\ \dot{\boldsymbol{x}}(t) \\ \ddot{\boldsymbol{x}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_0 + \dot{\boldsymbol{x}}_0 t + \boldsymbol{g}\frac{t^2}{2} \\ \dot{\boldsymbol{x}}_0 + \boldsymbol{g}t \\ \boldsymbol{g} \end{bmatrix}, \tag{3.3}$$

where $\boldsymbol{x}_0$, $\dot{\boldsymbol{x}}_0$ and $\ddot{\boldsymbol{x}}_0$ are the initial ($t = 0$) CoM position, velocity and acceleration at TO, with $\boldsymbol{x}(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T$.

The following sections show the planning process of the CoM trajectories for the vertical and horizontal directions, separately. The vertical trajectory planning (in the z-axis) is

solved locally for each phase and aims to reach a desired TD height at the upcoming footstep, besides solving for continuity between stance and flight phase. The horizontal planning approach (in the x- and y-axis) aims to compute a CoM trajectory according to a sequence of n desired footstep positions and a final CoM TO velocity. Additionally, the distance of the leg force intersection point with the ground is minimized.
The following planning algorithms are adapted from [3, 5].

3.1 Vertical planning

The vertical CoM trajectories are planned using 4th-order polynomials for each preview phase i, defined as

$$\begin{bmatrix} z(t) \\ \dot{z}(t) \\ \ddot{z}(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 \\ 0 & 1 & 2t & 3t^2 & 4t^3 \\ 0 & 0 & 2 & 6t & 12t^2 \end{bmatrix} \boldsymbol{p}_z = \begin{bmatrix} \boldsymbol{t}_z^T(t) \\ \boldsymbol{t}_{\dot{z}}^T(t) \\ \boldsymbol{t}_{\ddot{z}}^T(t) \end{bmatrix} \boldsymbol{p}_z. \tag{3.4}$$

Here, the time-mapping row vectors $\boldsymbol{t}_z^T(t)$, $\boldsymbol{t}_{\dot{z}}^T(t)$ and $\boldsymbol{t}_{\ddot{z}}^T(t)$ map the parameter vector $\boldsymbol{p}_z \in \mathbb{R}^5$ to the position $z(t)$, velocity $\dot{z}(t)$ and acceleration $\ddot{z}(t)$ of the CoM.

Five linear boundary conditions for five polynomial parameters $\boldsymbol{p}_{z,i} \in \mathbb{R}^5$ are defined for each preview step i as

$$\underbrace{\begin{bmatrix} z_{TD,i} \\ \dot{z}_{TD,i} \\ \ddot{z}_{TD,i} \\ -g \\ z_{TD,i+1,des} + g\frac{T_{f,i+1}^2}{2} \end{bmatrix}}_{\boldsymbol{H}_{z,i}} = \underbrace{\begin{bmatrix} \boldsymbol{t}_z(0) \\ \boldsymbol{t}_{\dot{z}}(0) \\ \boldsymbol{t}_{\ddot{z}}(0) \\ \boldsymbol{t}_{\ddot{z}}(T_{s,i}) \\ \boldsymbol{t}_z(T_{s,i}) + \boldsymbol{t}_{\dot{z}}(T_{s,i})T_{f,i+1} \end{bmatrix}}_{\boldsymbol{B}_{z,i}} \boldsymbol{p}_{z,i}\,, \quad i \in [1, n]. \tag{3.5}$$

Here, the first three boundaries set the given CoM TD position, velocity, and acceleration. The fourth boundary sets the constant TO acceleration $\ddot{z}_{TO,i} = -g$. The fifth boundary sets a desired TD position for the next preview step, which is derived using the flight dynamics as

$$z_{TD,i+1,des} = z_{TO,i} + \dot{z}_{TO,i}T_{f,i+1} - g\frac{T_{f,i+1}^2}{2} = [\boldsymbol{t}_z(T_{s,i}) + \boldsymbol{t}_{\dot{z}}(T_{s,i})T_{f,i+1}]\boldsymbol{p}_{z,i} - g\frac{T_{f,i+1}^2}{2}. \tag{3.6}$$

The desired TD position $z_{TD,i,des}$, flight time $T_{f,i}$, and stance time $T_{s,i}$ are considered as design parameters, for each preview phase i separately. In [3, 5] the desired apex height is used as a fifth boundary condition enabling an adapted flight time, while also forcing the CoM trajectory to reach the desired apex height. We use the desired touch-down position to have a fixed flight time.
The solution for the parameter vector $\boldsymbol{p}_{z,i}$ is found, for each preview phase separately, as

$$\boldsymbol{p}_{z,i} = \boldsymbol{B}_{z,i}^{-1}\boldsymbol{H}_{z,i}. \tag{3.7}$$

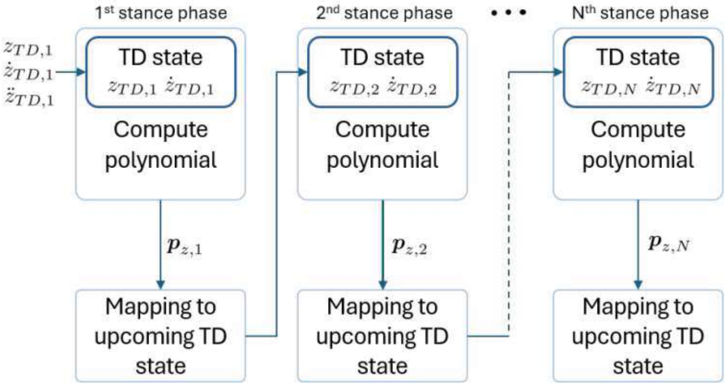


Figure 3.3: Computation flow of vertical planning

The vertical CoM trajectories are planned in a forward iterative manner, starting at the first preview phase i with given initial conditions (CoM TD position, velocity, and acceleration). The TD position, velocity, and acceleration for the next preview phases are evaluated using the CoM flight dynamics from (3.3) and the previous TO state as

$$\begin{bmatrix} z_{TD,i} \\ \dot{z}_{TD,i} \\ \ddot{z}_{TD,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_z(T_{s,i-1})\boldsymbol{p}_{z,i-1} + \boldsymbol{t}_{\dot{z}}(T_{s,i-1})\boldsymbol{p}_{z,i-1}T_{f,i} - g\frac{T_{f,i}^2}{2} \\ \boldsymbol{t}_{\dot{z}}(T_{s,i-1})\boldsymbol{p}_{z,i-1} - gT_{f,i} \\ -g \end{bmatrix}, \quad i > 1. \tag{3.8}$$

Figure 3.3 shows an outline of the vertical planning method.

3.2 Horizontal planning

The horizontal CoM trajectories are computed using 5th-order polynomials for each preview phase i. Here, the horizontal CoM position vector is defined as $\boldsymbol{\chi}(t) = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T$ combining both horizontal axes. Analogous to (3.4), the horizontal polynomials of fifth order are defined as

$$\begin{bmatrix} \boldsymbol{\chi}^T(t) \\ \dot{\boldsymbol{\chi}}^T(t) \\ \ddot{\boldsymbol{\chi}}^T(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix} \boldsymbol{p}_\chi = \begin{bmatrix} \boldsymbol{t}_\chi^T(t) \\ \boldsymbol{t}_{\dot{\chi}}^T(t) \\ \boldsymbol{t}_{\ddot{\chi}}^T(t) \end{bmatrix} \boldsymbol{p}_\chi. \tag{3.9}$$

Five linear boundary conditions for six polynomial parameters $\boldsymbol{p}_{\chi,i} = [\boldsymbol{p}_{x,i}, \boldsymbol{p}_{y,i}] \in \mathbb{R}^{6\text{x}2}$ are set for each preview step i as

$$\underbrace{\begin{bmatrix} \boldsymbol{\chi}_{TD,i}^T \\ \dot{\boldsymbol{\chi}}_{TD,i}^T \\ \ddot{\boldsymbol{\chi}}_{TD,i}^T \\ \ddot{\boldsymbol{\chi}}_{TO,i}^T \\ \boldsymbol{\chi}_{ft,i}^T \end{bmatrix}}_{\boldsymbol{H}_{\chi,i}} = \underbrace{\begin{bmatrix} \boldsymbol{t}_\chi(0) \\ \boldsymbol{t}_{\dot{\chi}}(0) \\ \boldsymbol{t}_{\ddot{\chi}}(0) \\ \boldsymbol{t}_{\ddot{\chi}}(T_{s,i}) \\ \boldsymbol{l}_{\chi,i}^T \end{bmatrix}}_{\boldsymbol{B}_{\chi,i}} \boldsymbol{p}_{\chi,i}\,, i \in [1, n]. \tag{3.10}$$

The first three boundaries set the initial CoM state to the CoM TD state. The fourth boundary sets the constant CoM TO state $\ddot{\boldsymbol{\chi}}_{TO,i} = [0,0]^T$ to zero for a continuous transition from stance to flight. The fifth boundary includes the step location during stance. The vector $\boldsymbol{l}_{\chi,i}$ maps the polynomial parameters to the corresponding step location $\boldsymbol{\chi}_{ft,i}$, as derived below. The step location for each preview step is commanded by the user. The general solution for (3.10) is found as

$$\boldsymbol{p}_{\chi,i} = \boldsymbol{B}_{\chi,i}^T \left(\boldsymbol{B}_{\chi,i}\boldsymbol{B}_{\chi,i}^T\right)^{-1} \boldsymbol{H}_{\chi,i} + \boldsymbol{r}_{\chi,i}\tilde{\boldsymbol{p}}_{\chi,i}. \tag{3.11}$$

where $\tilde{\boldsymbol{p}}_{\chi,i} = \begin{bmatrix}\tilde{p}_{x,i} & \tilde{p}_{y,i}\end{bmatrix} \in \mathbb{R}^2$ is a row vector of scalar multiples of the nullspace spanned by the column vector

$$\boldsymbol{r}_{\chi,i} = \begin{bmatrix} -\boldsymbol{B}_{\chi,i,\text{square}}^{-1}\boldsymbol{b}_{\chi,i,final} \\ 1 \end{bmatrix}. \tag{3.12}$$

Here, $\boldsymbol{b}_{\chi,i,final}$ is the last row of $\boldsymbol{B}_{\chi,i}$. $\boldsymbol{B}_{\chi,i,\text{square}}$ contains the remaining rows of $\boldsymbol{B}_{\chi,i}$. The goal is to find $\tilde{\boldsymbol{p}}_{\chi,i}$ which provides the best possible focus of leg forces at an intersection point with the ground ($z = z_{floor}$). Figure 3.4 illustrates the focus of leg forces projected from the CoM trajectory during stance. We want to minimize the mean square deviation of the intersection point with the ground, corresponding to the foot contact position. The horizontal components of the intersection point for the i-th stance phase ($t \in [0, T_{s,i}]$) are

$$\begin{aligned}\boldsymbol{\chi}_{\text{int},i}(t) &= \boldsymbol{\chi}(t) - \frac{f_{\text{leg},\chi,i}(t)}{f_{\text{leg},z,i}(t)}(z(t) - z_{\text{floor},i}) \\ &= \underbrace{\left[\boldsymbol{t}_\chi^T(t) - \frac{\boldsymbol{t}_{\ddot{\chi}}^T(t)}{(\boldsymbol{t}_{\ddot{z}}^T(t)\boldsymbol{p}_{z,i} + g)}(\boldsymbol{t}_z^T(t)\boldsymbol{p}_{z,i} - z_{\text{floor},i})\right]}_{\boldsymbol{d}_{\chi,i}^T(t)} \boldsymbol{p}_{\chi,i}.\end{aligned} \tag{3.13}$$

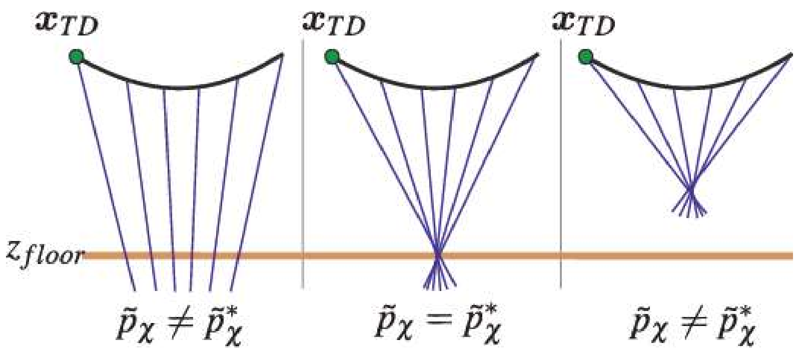

$\boldsymbol{x}_{TD}$
$\boldsymbol{x}_{TD}$
$\boldsymbol{x}_{TD}$
z_{floor}
$\tilde{p}_\chi \neq \tilde{p}_\chi^*$
$\tilde{p}_\chi = \tilde{p}_\chi^*$
$\tilde{p}_\chi \neq \tilde{p}_\chi^*$

Figure 3.4: Three different CoM trajectories with the force focus point below (left figure), on (center figure), and above (right figure) the floor height. The force focus point with the optimal parameter $\tilde{\boldsymbol{p}}_\chi$ is located at the floor height. [3]

The variation of the force focus point over one phase is defined as

$$\Delta\boldsymbol{\chi}_{\text{int},i}(t) = \boldsymbol{\chi}_{\text{int},i}(t) - \overline{\boldsymbol{\chi}}_{\text{int},i} = \underbrace{(\boldsymbol{d}_{\chi,i}(t) - \boldsymbol{l}_{\chi,i})}_{\boldsymbol{k}_{\chi,i}^T(t)} \boldsymbol{p}_{\chi,i}, \tag{3.14}$$

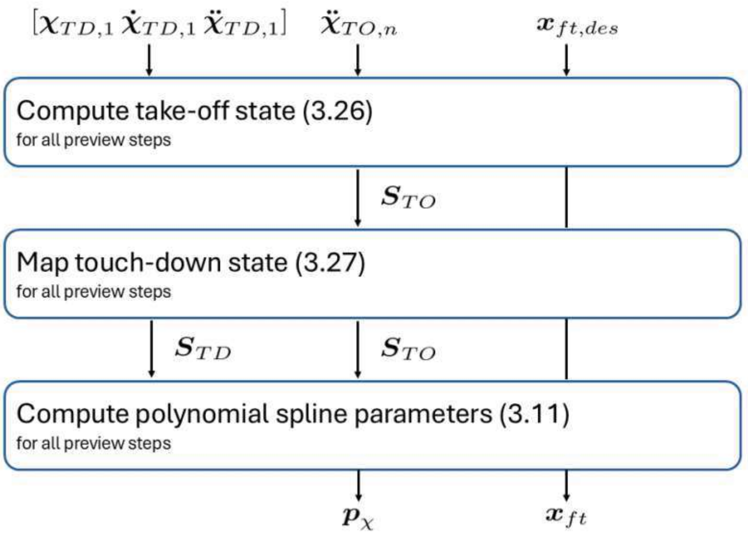

$[\chi_{TD,1}\ \dot{\chi}_{TD,1}\ \ddot{\chi}_{TD,1}]$
$\ddot{\chi}_{TO,n}$
$x_{ft,des}$
Compute take-off state (3.26)
for all preview steps
S_{TO}
Map touch-down state (3.27)
for all preview steps
S_{TD}
S_{TO}
Compute polynomial spline parameters (3.11)
for all preview steps
p_χ
x_{ft}

Figure 3.5: Computation flow of the horizontal CoM trajectory planning.

where $\boldsymbol{l}_{\chi,i} = \int_0^{T_{s,i}} \boldsymbol{d}_{\chi,i}(t)dt$ maps the mean force focus point $\overline{\chi}_{\text{int},i}$ using the parameter vector $\boldsymbol{p}_{\chi,i}$.

The mean square deviation of the force focus point results in

$$\begin{aligned}\Delta\chi_{\text{int},i,ms} &= \int_0^{T_{s,i}} \Delta\chi^2_{\text{int},i}(t)dt \\ &= \boldsymbol{p}^T_{\chi,i} \underbrace{\int_0^{T_{s,i}} \boldsymbol{k}_{\chi,i}(t)\boldsymbol{k}^T_{\chi,i}(t)dt}_{\boldsymbol{M}_{\chi,i}}\ \boldsymbol{p}_{\chi,i} \\ &= \boldsymbol{r}^T_{\chi,i}\boldsymbol{M}_{\chi,i}\boldsymbol{r}^T_{\chi,i}\tilde{p}^2_{\chi,i} + 2\boldsymbol{r}_{\chi,i}\boldsymbol{M}_{\chi,i}\boldsymbol{p}_{\chi,i,0}\tilde{p}^2_{\chi,i} + \boldsymbol{p}^T_{\chi,i}\boldsymbol{M}_{\chi,i}\boldsymbol{p}_{\chi,i,0}.\end{aligned} \tag{3.15}$$

Due to nonlinearity of (3.13), we approximate $\boldsymbol{M}_{\chi,i}$ with n_f approximation points. We minimize (3.15) with respect to $\tilde{p}_{\chi,i}$ and find the optimal parameter $\tilde{p}^*_{\chi,i}$. Equation (3.11) with optimal parameters yields

$$\boldsymbol{p}_{\chi,i} = \underbrace{\left(\boldsymbol{I} - \frac{\boldsymbol{r}_{\chi,i}\boldsymbol{r}^T_{\chi,i}\boldsymbol{M}_{\chi,i}}{\boldsymbol{r}^T_{\chi,i}\boldsymbol{M}_{\chi,i}\boldsymbol{r}}\right)}_{\boldsymbol{\Omega}_{\chi,i}} \underbrace{\boldsymbol{B}^T_{\chi,i}\left(\boldsymbol{B}_{\chi,i}\boldsymbol{B}^T_{\chi,i}\right)^{-1}}_{\boldsymbol{B}^+_{\chi,i}} \boldsymbol{H}_{\chi,i}. \tag{3.16}$$

Therefore, the boundary conditions are fulfilled and the best possible force focus point is achieved. As proposed by Egle et al. [5], the way-points are computed analytically using a matrix formulation for all n preview phases. Figure 3.5 illustrates the computation flow of the proposed horizontal planning approach. The TO state $\boldsymbol{S}_{TO}$ at the end of each preview stance phase i is defined as

$$\underbrace{\begin{bmatrix}\boldsymbol{\chi}_{TO,i}\\ \dot{\boldsymbol{\chi}}_{TO,i}\end{bmatrix}}_{\boldsymbol{S}_{TO,i}} = \underbrace{\begin{bmatrix}\boldsymbol{t}_{\chi}^{T}(T_{s,i})\\ \boldsymbol{t}_{\dot{\chi}}^{T}(T_{s,i})\end{bmatrix}}_{\boldsymbol{T}_{\chi,i}} \boldsymbol{p}_{\chi,i}. \tag{3.17}$$

Here, we insert (3.16) as

$$\boldsymbol{S}_{TO,i} = \underbrace{\boldsymbol{T}_{\chi,i}\boldsymbol{\Omega}_{\chi,i}\boldsymbol{B}_{\chi,i}^{+}}_{\boldsymbol{D}_{\chi,i}\in\mathbb{R}^{2x5}}\boldsymbol{H}_{\chi,i}. \tag{3.18}$$

The matrix $\boldsymbol{D}_{\chi,i}$ is separated into the first two columns as the matrix $\boldsymbol{D}_{\alpha,i}$ and the remaining columns as the vectors $\boldsymbol{d}_{\beta,i}$, $\boldsymbol{d}_{\gamma,i}$ and $\boldsymbol{d}_{\delta,i}$. Encoding $\boldsymbol{H}_{\chi,i}$ accordingly, the TO state for each preview phase i results in

$$\boldsymbol{S}_{TO,i} = \boldsymbol{D}_{\alpha,i}\underbrace{\begin{bmatrix}\boldsymbol{\chi}_{TD,i}\\ \dot{\boldsymbol{\chi}}_{TD,i}\end{bmatrix}}_{\boldsymbol{S}_{TD,i}} + \boldsymbol{d}_{\beta,i}\ddot{\boldsymbol{\chi}}_{TD,i} + \boldsymbol{d}_{\gamma,i}\ddot{\boldsymbol{\chi}}_{TO,i} + \boldsymbol{d}_{\delta,i}\boldsymbol{\chi}_{ft,i}. \tag{3.19}$$

Combining the matrices for all preview phases yields

$$\begin{aligned}\boldsymbol{S}_{TO} &= \begin{bmatrix}\boldsymbol{S}_{TO,1}^{T} & \cdots & \boldsymbol{S}_{TO,n}^{T}\end{bmatrix}^{T},\\ \boldsymbol{S}_{TD} &= \begin{bmatrix}\boldsymbol{S}_{TD,1}^{T} & \cdots & \boldsymbol{S}_{TD,n}^{T}\end{bmatrix}^{T},\\ \boldsymbol{X}_{ft} &= \begin{bmatrix}\boldsymbol{\chi}_{ft,1}^{T} & \cdots & \boldsymbol{\chi}_{ft,n}^{T}\end{bmatrix}^{T}.\end{aligned} \tag{3.20}$$

The starting acceleration $\boldsymbol{\chi}_{TD,1}$ is defined by the current robot state, the acceleration at the end of the last preview step $\boldsymbol{\chi}_{TO,n}$ can be chosen as a design parameter. All other way-point accelerations are zero due to the CoM flight dynamics. The TO state in matrix form is expressed as

$$\boldsymbol{S}_{TO} = \boldsymbol{D}_{\alpha}\boldsymbol{S}_{TD} + \boldsymbol{D}_{\beta}\ddot{\chi}_{TD,1} + \boldsymbol{D}_{\gamma}\ddot{\chi}_{TO,n} + \boldsymbol{D}_{\delta}\boldsymbol{X}_{ft}, \tag{3.21}$$

where $\boldsymbol{D}_{\beta} = \boldsymbol{Z}_{1}^{T}\boldsymbol{d}_{\beta,1}$, $\boldsymbol{D}_{\gamma} = \boldsymbol{Z}_{N}^{T}\boldsymbol{d}_{\gamma,n}$ and $\boldsymbol{D}_{\alpha}$ and $\boldsymbol{D}_{\delta}$ are block diagonal matrices whose diagonal contains the single-phase matrices $\boldsymbol{D}_{\alpha,i}$ and vectors $\boldsymbol{d}_{\gamma,i}$, respectively. The selection matrices are defined as

$$\begin{aligned}\boldsymbol{Z}_{1} &= \begin{bmatrix}\boldsymbol{e}_{2n,1} & \boldsymbol{e}_{2n,2}\end{bmatrix}^{T}, \boldsymbol{Z}_{N} = \begin{bmatrix}\boldsymbol{e}_{2n,2n-1} & \boldsymbol{e}_{2n,2n}\end{bmatrix}^{T},\\ \boldsymbol{Z}_{T} &= \begin{bmatrix}\boldsymbol{e}_{2n,3} & \cdots & \boldsymbol{e}_{2n,2n}\end{bmatrix}^{T}, \boldsymbol{Z}_{0} = \begin{bmatrix}\boldsymbol{e}_{2n,1} & \cdots & \boldsymbol{e}_{2n,2n-3}\end{bmatrix}^{T},\end{aligned} \tag{3.22}$$

where $\boldsymbol{Z}_{1}$ and $\boldsymbol{Z}_{N}$ select the first and last state from $\boldsymbol{S}_{TO}$ and $\boldsymbol{S}_{TD}$. The selection matrices $\boldsymbol{Z}_{T}$ and $\boldsymbol{Z}_{0}$ select the states $[2\cdots n]$ and $[1\cdots n-1]$, respectively.

We can separate the mapping of the TD states using $\boldsymbol{Z}_{1}^{T}\boldsymbol{D}_{\alpha,1}$, which corresponds to the first two columns of $\boldsymbol{D}_{\alpha}$, and $\boldsymbol{D}_{\alpha}\boldsymbol{Z}_{T}^{T}$ containing all remaining columns of $\boldsymbol{D}_{\alpha}$. Thus, the TO state matrix results in

$$\boldsymbol{S}_{TO} = \boldsymbol{Z}_{1}^{T}\boldsymbol{D}_{\alpha,1}\boldsymbol{S}_{TD,1} + \boldsymbol{D}_{\alpha}\boldsymbol{Z}_{T}^{T}\boldsymbol{S}_{TD,T} + \boldsymbol{D}_{\beta}\ddot{\chi}_{TD,1} + \boldsymbol{D}_{\gamma}\ddot{\chi}_{TO,n} + \boldsymbol{D}_{\delta}\boldsymbol{X}_{ft}, \tag{3.23}$$

where $\boldsymbol{S}_{TD,T} = \boldsymbol{Z}_T\boldsymbol{S}_{TD}$ encodes all TD states except the first. The TD state $\boldsymbol{S}_{TD,i}$ is expressed using the flight dynamics from (3.3) as

$$\underbrace{\begin{bmatrix}\boldsymbol{\chi}_{TD,i+1}\\ \dot{\boldsymbol{\chi}}_{TD,i+1}\end{bmatrix}}_{\boldsymbol{S}_{TD,i+1}} = \underbrace{\begin{bmatrix}1 & T_{f,i+1}\\ 0 & 1\end{bmatrix}}_{\boldsymbol{A}_{F,i}}\underbrace{\begin{bmatrix}\boldsymbol{\chi}_{TO,i}\\ \dot{\boldsymbol{\chi}}_{TO,i}\end{bmatrix}}_{\boldsymbol{S}_{TO,i}}. \tag{3.24}$$

Therefore, the matrix formulation for all TD states, except the initial state, is defined as

$$\boldsymbol{S}_{TD,T} = \boldsymbol{A}_F\boldsymbol{Z}_0\boldsymbol{S}_{TO}, \tag{3.25}$$

where $\boldsymbol{A}_F$ is a block diagonal matrix whose diagonal contains the single flight phase mapping matrices $\boldsymbol{A}_{F,i}$. We insert (3.23) into (3.25) and solve for $\boldsymbol{S}_{TO}$ as

$$\boldsymbol{S}_{TO} = \boldsymbol{D}_R(\boldsymbol{Z}_1^T\boldsymbol{D}_{\alpha,1}\boldsymbol{S}_{TD,1} + \boldsymbol{D}_\beta\ddot{\chi}_{TD,1} + \boldsymbol{D}_\gamma\ddot{\chi}_{TO,n} + \boldsymbol{D}_\delta\boldsymbol{X}_{ft}), \tag{3.26}$$

with $\boldsymbol{D}_R = (I - \boldsymbol{D}_\alpha\boldsymbol{Z}_T^T\boldsymbol{A}_F\boldsymbol{Z}_0)^{-1}$. This solution is valid if the matrix $\boldsymbol{D}_R^{-1}$ is a lower triangular matrix with non-zero elements on the diagonal, thus invertible. After computing the TO way-points $\boldsymbol{S}_{TO}$, we insert (3.26) into (3.21) which leads to the TD way-points for all preview phases as

$$\boldsymbol{S}_{TD} = \boldsymbol{D}_\alpha^{-1}(\boldsymbol{S}_{TO} - \boldsymbol{D}_\beta\ddot{\chi}_{TD,1} - \boldsymbol{D}_\gamma\ddot{\chi}_{TO,n} - \boldsymbol{D}_\delta\boldsymbol{X}_{ft}). \tag{3.27}$$

Here, the solution is valid if $\boldsymbol{D}_\alpha$ is a lower triangular matrix with non-zero elements on the diagonal and is thus invertible. With the results from (3.26) and (3.27) the parameter vector for the horizontal polynomial spline (3.16) is solved for each preview step. The design parameters for horizontal planning are the final desired CoM acceleration $\ddot{\boldsymbol{\chi}}_{TO,n}$ and the target footstep locations $\boldsymbol{X}_{ft}$. The initial TD positions, velocities, and accelerations are given.

3.3 Planner structure

The combination of vertical and horizontal CoM trajectory planning offers a comprehensible method to compute the CoM trajectory for several preview footsteps. Here, we want to show an overview of the planning procedure. Given a general preview phase i, the planned CoM trajectory $\boldsymbol{x}_{p,i}$, at the current phase time $t_{p,i} \in [0, T_{p,i}]$, is defined within the phase period $T_{p,i} = T_{f,i} + T_{s,i}$ as

$$\boldsymbol{x}_{p,i}(t_{p,i}) = \begin{cases}\boldsymbol{x}(0) + \dot{\boldsymbol{x}}(0)t_{p,i} + \boldsymbol{g}\frac{t_{p,i}^2}{2}, & \text{if } 0 \le t_{p,i} < T_{f,i}\\ \boldsymbol{t}_\chi^T(t_{p,i} - T_{f,i})\boldsymbol{p}_i, & \text{if } T_{f,i} \le t_{p,i} < T_{p,i}\end{cases} \tag{3.28}$$

with the general parameter vector $\boldsymbol{p}_i = [\boldsymbol{p}_{\chi,i}^T, \boldsymbol{p}_{z,i}^T, 0]^T$. During the flight phase, the planned trajectory is described using the CoM flight dynamics. The TO position $\boldsymbol{x}(0)$ and velocity $\dot{\boldsymbol{x}}(0)$ are given. In the stance phase, the planned CoM trajectory is described using the polynomial splines evaluated at the corresponding phase time.

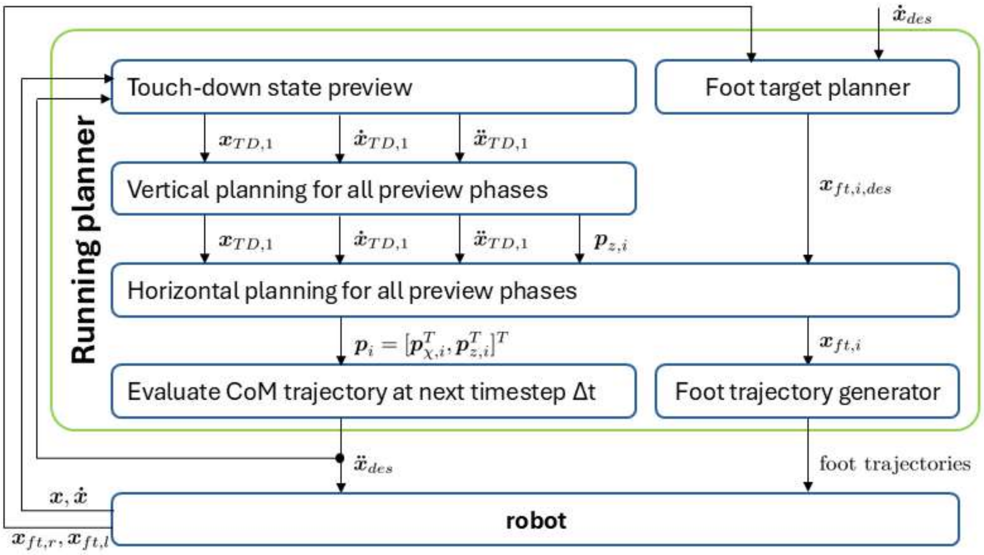

$\dot{\boldsymbol{x}}_{des}$
Touch-down state preview
Foot target planner
Running planner
$\boldsymbol{x}_{TD,1}$
$\dot{\boldsymbol{x}}_{TD,1}$
$\ddot{\boldsymbol{x}}_{TD,1}$
Vertical planning for all preview phases
$\boldsymbol{x}_{ft,i,des}$
$\boldsymbol{x}_{TD,1}$
$\dot{\boldsymbol{x}}_{TD,1}$
$\ddot{\boldsymbol{x}}_{TD,1}$
$\boldsymbol{p}_{z,i}$
Horizontal planning for all preview phases
$\boldsymbol{p}_i = [\boldsymbol{p}_{\chi,i}^T, \boldsymbol{p}_{z,i}^T]^T$
$\boldsymbol{x}_{ft,i}$
Evaluate CoM trajectory at next timestep Δt
Foot trajectory generator
$\ddot{\boldsymbol{x}}_{des}$
foot trajectories
$\boldsymbol{x}, \dot{\boldsymbol{x}}$
robot
$\boldsymbol{x}_{ft,r}, \boldsymbol{x}_{ft,l}$

Figure 3.6: Computation flow of the CoM trajectory running planner.

The planned trajectory for all n preview phases, at a given time t within the preview horizon, is defined as

$$\boldsymbol{x}_p(t) = \begin{cases} \boldsymbol{x}_{p,1}(t), & \text{if } 0 \leq t < T_{p,1} \\ \boldsymbol{x}_{p,2}(t - T_{p,1}), & \text{if } T_{p,1} \leq t < T_{p,1} + T_{p,2} \\ \vdots & \\ \boldsymbol{x}_{p,n}(t - \tau_{n-1}), & \text{if } \tau_{n-1} \leq t < \tau_n \end{cases} \tag{3.29}$$

where $\tau_k = \sum_{j=1}^{k} T_{p,j}$ is the time passed after k phase periods. The planned CoM velocity and acceleration are defined analogous, considering the derived flight dynamics and time mapping vector in Section 3.28.

Figure 3.6 shows the computation flow of the planner structure outline. The first step is the touch-down state preview, where the upcoming CoM touch-down state is computed using the CoM flight dynamics. Further explanation for this step can be found in Section 3.3.1. In parallel, the foot target planner computes the desired foot targets for all preview phases using the commanded desired speed $\dot{\boldsymbol{x}}_{des}$ and the current foot positions.

The second step computes the vertical polynomial spline parameters $\boldsymbol{p}_{z,i}$ (see Section 3.1) for all preview phases. After that, we compute the horizontal polynomial spline parameters $\boldsymbol{p}_{\chi,i}$ (see Section 3.2) for all preview phases. In the last step, the planned trajectory is evaluated at the next timestep $t' = t + dt$, where dt is the sampling time. The resulting desired CoM acceleration $\ddot{\boldsymbol{x}}_{\text{des}}$ is forwarded to the robot model, where the desired leg force $\boldsymbol{F}_{\text{leg,des}}$ is computed using (3.1).

The foot trajectory generator computes spline-based trajectories for both feet, according to the foot target positions $\boldsymbol{x}_{ft,i}$ for the first and second preview phases. The foot target positions $\boldsymbol{x}_{ft,i}$ are considered equal to the desired foot target positions $\boldsymbol{x}_{ft,i,\text{des}}$. For more details about the foot trajectory generator, see Section 3.3.2.

3.3.1 Online planning

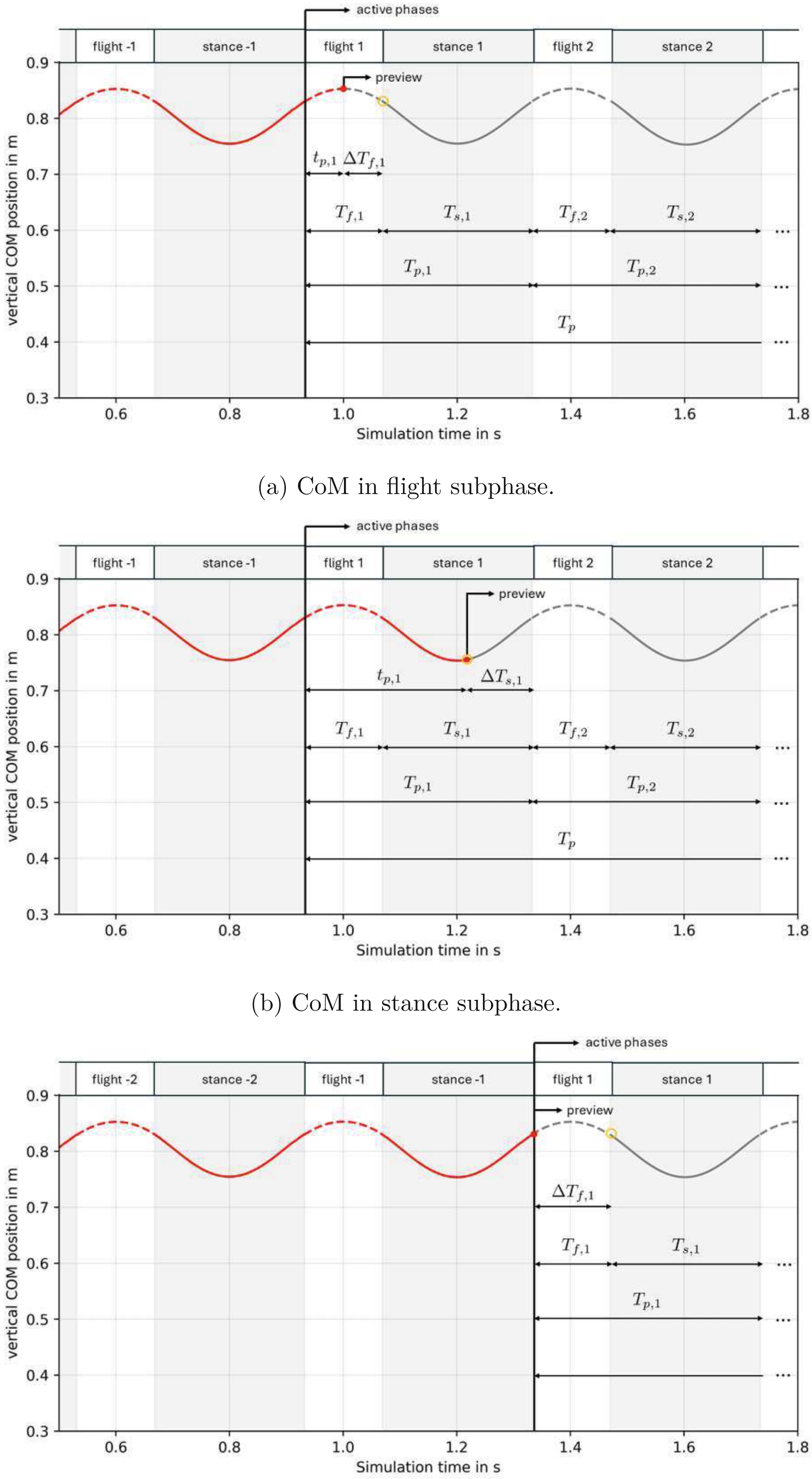
To enable robust and dynamic running, we propose an online planning approach using a finite horizon, spanned by n preview phases. The planned trajectory is re-computed at each timestep t, according to the current CoM state. Therefore, the planned CoM trajectory and foot target positions are adapted online. Figure 3.7 illustrates the online planning approach with three different scenarios in vertical planning. The current CoM state is used as an input parameter to determine the touch-down state of the first preview phase ($i = 1$), thus the initial touch-down state (marked with an orange circle). The computation of the first touch-down state depends on the current subphase (flight or stance).

The first scenario (see Figure 3.7a) shows the current CoM position in flight. Here, $\Delta T_{f,1} = T_{f,1} - t_{p,1}$ denotes the remaining flight time using the passed phase time $t_{p,1}$, starting from the previous take-off waypoint. In the flight phase, the upcoming TD state is estimated using the flight dynamics as

$$\boldsymbol{x}(t + \Delta T_{f,1}) = \boldsymbol{x}(t) + \dot{\boldsymbol{x}}(t)\Delta T_{f,1} + \boldsymbol{g}\frac{\Delta T_{f,1}^2}{2}, \tag{3.30}$$

with the remaining flight time $\Delta T_{f,1} = T_{f,1} - t_{p,1}$ and the current CoM position $\boldsymbol{x}(t)$ and velocity $\dot{\boldsymbol{x}}(t)$.
The second scenario (see Figure 3.7b) shows the current CoM position in stance. Here, the TD state is set to the current CoM state. Therefore, the polynomial spline for the first preview phase is evaluated from the current CoM state to the take-off with a reduced stance time $\Delta T_{s,1} = T_{p,1} - t_{p,1}$. Both subphases, allow for continuous trajectory adaptation.
The third scenario (see Figure 3.7c) shows a take-off, where the phase time exceeded the first preview phase $t_{p,1} > T_{p,1}$. Thus, the phase indices are shifted, so that the second phase is considered as the new first preview phase. An additional preview phase is added at the last index $i = n$. This ensures a continuous definition of the initial CoM touch-down state.

The planner output is a trajectory evaluated at the upcoming timestep $t' = t + \Delta t$, where Δt is the sampling time, for each timestep. The output includes the full reference CoM state (position, velocity, and acceleration), and the reference foot state (position, velocity, and acceleration) for each foot. Figure 3.8 illustrates the planner output at the upcoming timestep t' (marked with the green circle).



(a) CoM in flight subphase.

(b) CoM in stance subphase.

(c) CoM in next flight subphase. Subphase-indices are shifted to the right.

Figure 3.7: Online CoM trajectory planning with three scenarios. The orange dot illustrates the first TD position. The red dot shows the current CoM position.

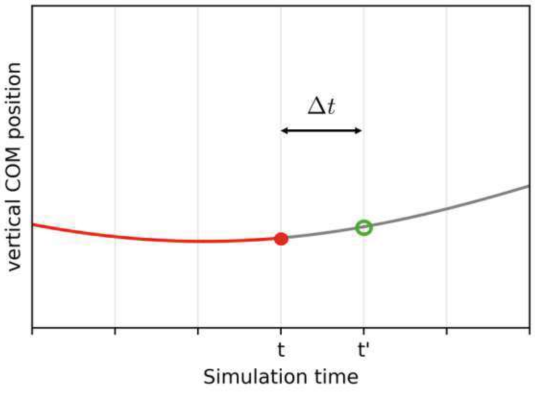


Figure 3.8: Current CoM trajectory (red dot) at timestep t with the reference CoM position (green circle) at timestep t'. The sampling rate is denoted as Δt.

The online planner increases the robustness of the planning method significantly, due to continuous CoM trajectory re-planning.

3.3.2 Foot trajectory planning

The planning method offers foot target positions, which could be used on a real robot with set-point control for the swing foot. However, this work uses foot trajectories with 5-th order polynomial splines. These enable smooth transitions adapted for the corresponding timing setting.

For the current stance foot, we assume no slippage and command a constant reference foot position with zero velocity and acceleration. On the other hand, during flight, the footstep trajectory is computed using 5-th order polynomials for each axis separately. Similar to the CoM spline polynomials from Section 3.1 and Section 3.2, the footstep trajectory for the swing foot is defined as

$$\begin{bmatrix} \sigma_{ft}^T(t) \\ \dot{\sigma}_{ft}^T(t) \\ \ddot{\sigma}_{ft}^T(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_{\sigma_{ft}}^T(t) \\ \boldsymbol{t}_{\dot{\sigma}_{ft}}^T(t) \\ \boldsymbol{t}_{\ddot{\sigma}_{ft}}^T(t) \end{bmatrix} \boldsymbol{p}_{\sigma_{ft}} , \quad \sigma_{ft} \in [x_{ft}, y_{ft}, z_{ft}]. \tag{3.31}$$

The time-mapping vectors are equal to Section 3.2. The trajectory should enable a smooth transition between the start foot position $\sigma_{ft}(t=0)$ and the target foot position $\sigma_{ft}(t=T_{ft})$ with zero velocity and acceleration at the start and end of the foot flight

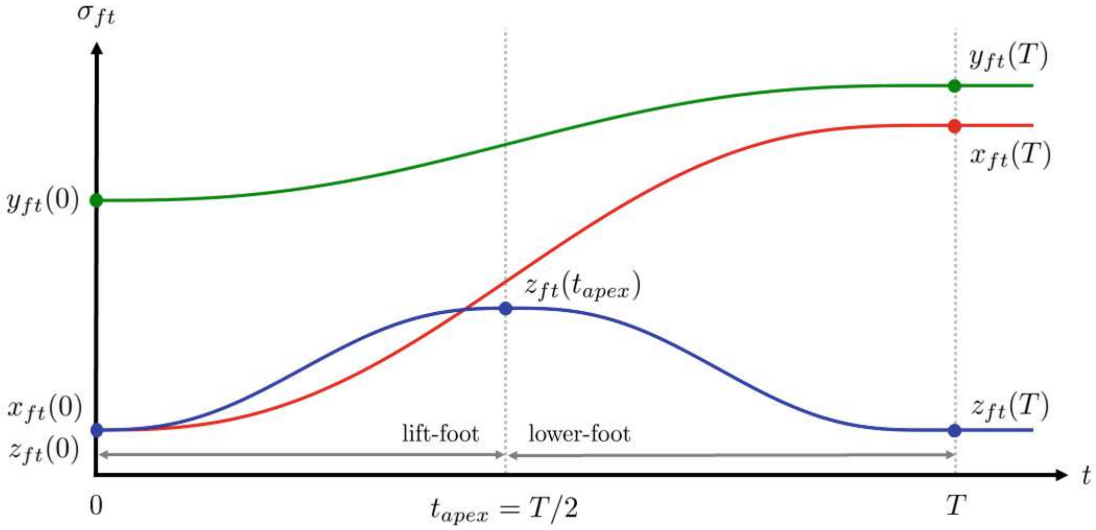


Figure 3.9: Exemplary foot trajectories.

phase T_{ft}. Six boundary conditions fulfill these requirements as

$$\underbrace{\begin{bmatrix} \sigma_{ft}(0) \\ \dot{\sigma}_{ft}(0) \\ \ddot{\sigma}_{ft}(0) \\ \sigma_{ft}(T) \\ \dot{\sigma}_{ft}(T) \\ \ddot{\sigma}_{ft}(T) \end{bmatrix}}_{\boldsymbol{H}_\sigma} = \underbrace{\begin{bmatrix} \boldsymbol{t}_{\sigma_{ft}}(0) \\ \boldsymbol{t}_{\dot{\sigma}_{ft}}(0) \\ \boldsymbol{t}_{\ddot{\sigma}_{ft}}(0) \\ \boldsymbol{t}_{\sigma_{ft}}(T) \\ \boldsymbol{t}_{\dot{\sigma}_{ft}}(T) \\ \boldsymbol{t}_{\ddot{\sigma}_{ft}}(T) \end{bmatrix}}_{\boldsymbol{B}_\sigma} \boldsymbol{p}_\sigma \,, \quad \sigma_{ft} \in [x_{ft}, y_{ft}, z_{ft}]. \tag{3.32}$$

The solution of the parameter vector is $\boldsymbol{p}_\sigma = \boldsymbol{B}_\sigma^{-1}\boldsymbol{H}_\sigma$. The horizontal components (x,y) are computed for the whole foot flight phase $t = [0, T_{ft}]$ with the starting footstep $\sigma_{ft}(0) = \sigma_{ft,\text{start}}$ and target footstep $\sigma_{ft}(T) = \sigma_{ft,\text{target}}$. The vertical axis (z) is separated into two subphases, lifting and lowering the foot. The lifting phase reaches from $t = [0, T_{ft}/2]$ with the starting footstep $z_{ft}(0) = z_{ft,\text{start}}$ and target foot position $z_{ft}(T) = z_{ft,\text{apex}}$. The lowering phase reaches from $t = [T_{ft}/2, T_{ft}]$ with the starting footstep $z_{ft}(0) = z_{ft,\text{apex}}$ and target foot position $z_{ft}(T) = z_{ft,\text{apex}}$. Figure 3.9 shows the foot trajectories for a single foot flight phase.

3.4 Conclusion

The planning approach presented in this chapter offers a comprehensible method to compute CoM trajectories that approximate human-like running GRF profiles. For vertical and horizontal CoM trajectory planning, polynomial splines of 4-th and 5-th order are used, both fulfilling a continuous transition between the stance and the flight phase for several preview phases. The vertical CoM trajectory planning is solved locally, for each preview phase separately, and aims to reach a desired TD height. The horizontal CoM trajectory planning aims to find a CoM trajectory according to a sequence of n desired footsteps, the final CoM TO velocity, and a minimized vertical distance of the leg force intersection point with the ground. The combination of both planning approaches enables an online planning structure with a finite horizon (n preview footsteps). The planned CoM trajectory is updated after each timestep using the current CoM state. Thus, perturbations on the CoM are considered in planning, increasing the planner´s robustness.

The limitations of this method are within its design parameters. Sub-optimally placed desired foot positions and fixed phase timing can cause the running locomotion to diverge, thus causing the robot to fall. Especially during challenging robot states, such as running from a standstill or with external perturbations, the method does not converge to a stable running motion.

Therefore, we extend the planning method with optimized design parameters to enable robust running even during challenging robot states or with high perturbations.

4 Optimal Step Location Adapation

This chapter extends the proposed biologically-inspired CoM trajectory planning method with optimal foot target placement and consequently optimal CoM waypoints. This enables a robust running locomotion method that can withstand strong perturbations and allows stable running even during challenging robot configurations using step adaptation. Additionally, with this method, the force focus point is shifted optimally within the stance foot contact area. Thus, two adaptation strategies are used to compensate for perturbations.
The optimization affects only the horizontal trajectory planning part. Therefore, the vertical CoM trajectory planning is computed locally for each preview phase separately without optimization as proposed in Section 3.1. Phase timing is not considered for optimization. This would result in a more general non-linear optimization problem which might be challenging to solve in real-time.

The adapted computation flow scheme is illustrated in Figure 4.1. Here, the analytical solution of the CoM take-off states (for all preview steps) from (3.26) is integrated in an optimization problem. This enables optimal target foot positions for all preview steps and optimal CoM waypoints. More details on the optimization problem formulation are elaborated in this chapter.

4.1 Optimization Problem

The cost function G is minimized in the quadratic program (QP). Here, the foot adjustment distance $\boldsymbol{\Delta\chi}_{ft,i}$ between the target foot positions $\boldsymbol{\chi}_{ft,i}$ and desired foot positions $\boldsymbol{\chi}_{ft,i,d}$ is minimized, for each phase and each horizontal axis separately. Additionally, the error between the target CoM velocity $\dot{\boldsymbol{\chi}}_{TO,n}$ and desired CoM velocity $\dot{\boldsymbol{\chi}}_{TO,n,d}$ at the last preview phase $(i = n)$ is minimized, for each axis separately.

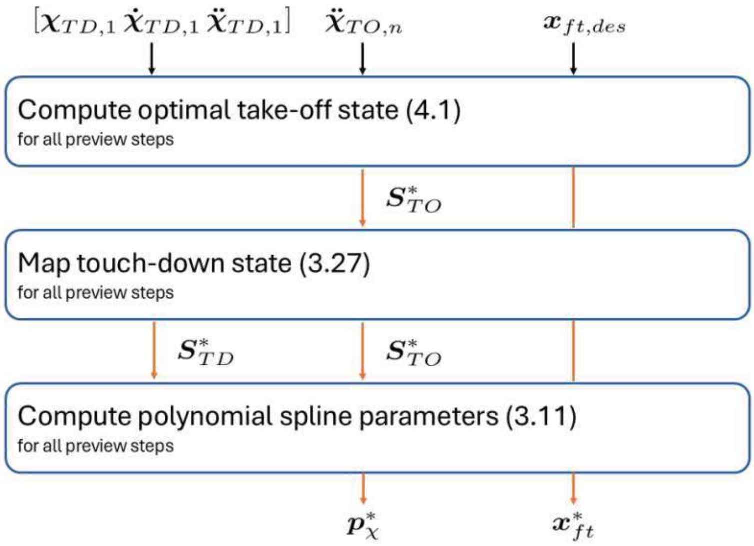

$[\boldsymbol{\chi}_{TD,1}\ \dot{\boldsymbol{\chi}}_{TD,1}\ \ddot{\boldsymbol{\chi}}_{TD,1}]$
$\ddot{\boldsymbol{\chi}}_{TO,n}$
$\boldsymbol{x}_{ft,des}$
Compute optimal take-off state (4.1)
for all preview steps
$\boldsymbol{S}^*_{TO}$
Map touch-down state (3.27)
for all preview steps
$\boldsymbol{S}^*_{TD}$
$\boldsymbol{S}^*_{TO}$
Compute polynomial spline parameters (3.11)
for all preview steps
$\boldsymbol{p}^*_{\chi}$
$\boldsymbol{x}^*_{ft}$

Figure 4.1: Computation flow diagram for horizontal CoM planning, extended with optimal target foot placement and optimal CoM waypoints for all preview phases. Changes to Figure 3.5 are marked in orange.

The optimization problem is formulated as

$$\begin{aligned}\min_{\boldsymbol{\epsilon}} \quad & G = \sum_{i=1}^{n} \alpha_{ft,i}(||x_{ft,i} - x_{ft,i,d}|| + ||y_{ft,i} - y_{ft,i,d}||) + \\ & \alpha_{\chi_p,n}(||\dot{x}_{TO,n} - \dot{x}_{TO,n,d}|| + ||\dot{y}_{TO,n} - \dot{y}_{TO,n,d}||), \\ \text{s.t.} \quad & \boldsymbol{S}_{TO} = \boldsymbol{D}_R(\boldsymbol{D}_\delta \boldsymbol{\chi}_{ft} + \boldsymbol{Z}_1^T \boldsymbol{D}_{\alpha,1} \boldsymbol{S}_{TD,1} + \boldsymbol{D}_\beta \ddot{\boldsymbol{\chi}}_{TD,1} + \boldsymbol{D}_\gamma \ddot{\boldsymbol{\chi}}_{TO,N}), \\ & \boldsymbol{\chi}_{ft,i} \in \begin{cases} \boldsymbol{\Omega}_s, & \text{if } i = 1 \text{ and stance} \\ \boldsymbol{\Omega}_i, & \text{otherwise} \end{cases} \qquad \forall i \in n,\end{aligned} \tag{4.1}$$

with the full optimization vector

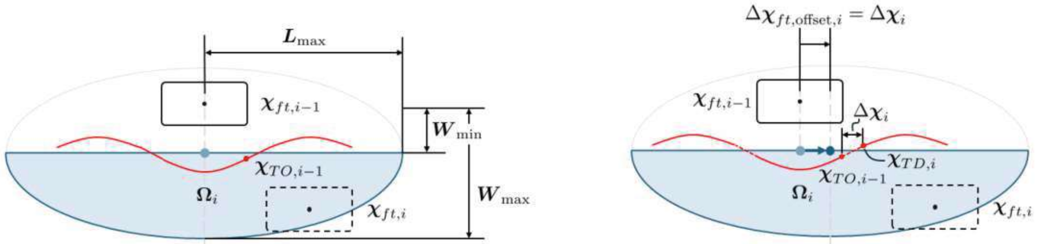
$$\begin{aligned}\boldsymbol{\epsilon} = [& x_{ft,1}, \cdots, x_{ft,n}, \\ & y_{ft,1}, \cdots, y_{ft,n}, \\ & x_{TO,1}, \dot{x}_{TO,1}, \cdots, x_{TO,n}, \dot{x}_{TO,n}, \\ & y_{TO,1}, \dot{y}_{TO,1}, \cdots, y_{TO,n}, \dot{y}_{TO,n}].\end{aligned} \tag{4.2}$$

The first $2n$ elements of $\boldsymbol{\epsilon}$ are the target foot positions in both horizontal axes and the last $4n$ elements are the TO states $\boldsymbol{S}_{TO}$ (position and velocity) for all preview phases and both horizontal axes.

Equation (3.26) is used as an equality constraint in (4.1), thus the biologically-inspired horizontal trajectory boundaries are fulfilled. The inequality constraints ensure that the target foot positions are within a specified footstep region $\mathbf{\Omega}$, thus leg collisions are prevented and the robot´s maximum step length and width are not exceeded. In stance, the weight for the first ($i = 1$) foot adjustment distance is set to zero $\alpha_{ft,1} = 0$, as well as the footstep region for the first target footstep position is defined within the foot contact area to enable an adaptation of the force focus point. Further details regarding the footstep regions are shown in the following section.

4.2 Convex Footstep Regions

The footstep region $\mathbf{\Omega}$ is defined as an area where the target foot position is considered feasible to reach. In this method, we want to prevent the robot´s foot distance from exceeding its kinematic limits and prevent leg collisions. To fulfill these conditions and meet the convexity requirements of the quadratic program we use the approximation of half an ellipse as a feasible footstep region.



(a) Walking

(b) Running

Figure 4.2: Feasible footstep region Ω_i (blue area) for the right foot target $\chi_{ft,i}$ considering the left foot $\chi_{ft,i-1}$ in stance and as a starting point ($k = -1$) with the locomotion modes (a) walking and (b) running. The CoM trajectory (red) is shown with flight phases (dashed) while running.

Figure 4.2a shows the footstep region $\mathbf{\Omega}_i$ for the target foot position $\chi_{ft,i}$ with one preview footstep. The center of the ellipse is located at the previous target foot position $\chi_{ft,i-1}$, shifted by the minimum step width W_{min} in the lateral axis. The ellipse´s length, thus the feasible area in the longitudinal direction, is set to the maximum step length L_{max}. The feasible step width is defined by the difference of maximum step width and minimum step with as $W_{\text{max}} - W_{\text{min}}$. The area of the ellipse limits the robot´s footstep distance from exceeding its kinematic limits. Leg collisions are prevented by limiting the lateral footstep area of the ellipse to a step width greater than W_{min}, see the blue area in Figure 4.2a. The footstep region $\mathbf{\Omega}_i$ is defined as

$$\frac{(x_{ft,i} - x_{ft,i-1})^2}{L_{\text{max}}^2} + \frac{(y_{ft,i} - y_{ft,i-1} - kW_{\text{min}})^2}{(W_{\text{max}} - W_{\text{min}})^2} <= 1, \text{ with } k\Delta y_{ft,i} >= W_{\text{min}}. \qquad (4.3)$$

The target footstep $\chi_{ft,i}$ can be placed anywhere in the footstep region (blue area). In Figures 4.2a and 4.2b the visualized target foot positions are selected arbitrarily and can be placed anywhere within the footstep region. The footstep region of the ellipse alternates in the lateral direction (lower half or upper half) depending on the previous footstep $\boldsymbol{\chi}_{ft,i-1}$, set with the variable k. The variable k is selected $k = 1$ for the right foot as the previous footstep, and $k = -1$ for the left foot as the previous footstep.

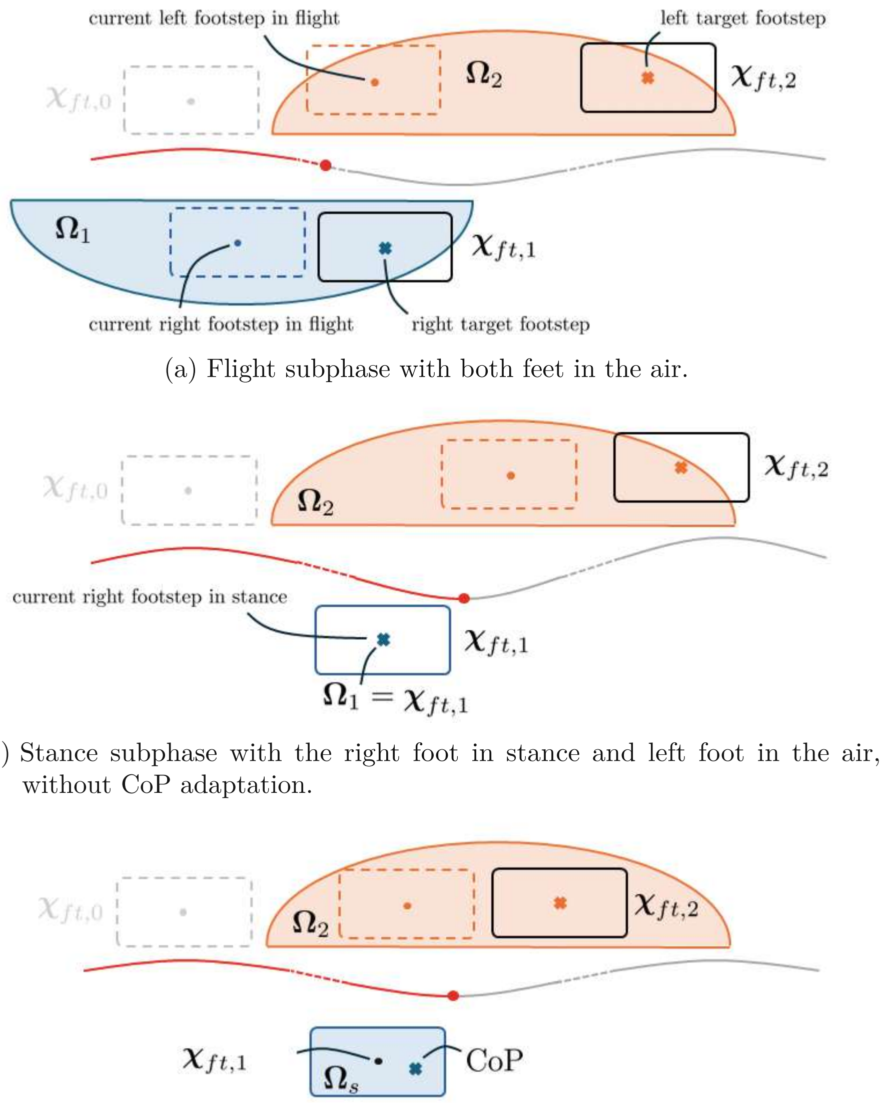
During walking (see Figure 4.2a), i.e. without flight phases, the CoM traveled distance in flight $\Delta\boldsymbol{\chi}_i = \boldsymbol{\chi}_{TD,i} - \boldsymbol{\chi}_{TO,i-1}$ equals to zero. Thus, at the moment of impact of the i-th footstep (i-th TD), the distance to the previous footstep is exactly $\boldsymbol{\chi}_{ft,i} - \boldsymbol{\chi}_{ft,i-1}$. While running (see Figure 4.2b), the robot´s body movement in flight shifts significantly, especially with long flight phases. Therefore, we propose to shift the position of the footstep region with the CoM traveled distance in flight $\Delta\boldsymbol{\chi}_i$. This extension enables a faster running speed, due to a shift of the footstep region ellipse in the running direction.

Figure 4.3a shows the footstep regions ($\boldsymbol{\Omega_1}$ and $\boldsymbol{\Omega_2}$) for two preview steps in the flight subphase. Both footstep regions are computed using the corresponding previous target foot position ($\boldsymbol{\chi}_{ft,0}$ and $\boldsymbol{\chi}_{ft,1}$) as a starting point with the foot distance offset according to the CoM traveled distance in flight.

Figure 4.3b illustrates the footstep regions for the stance subphase. The stance foot (here right foot) is constrained to the touch-down foot position. Here, the robot´s center of pressure (CoP)[1] (force focus point) is located at the stance foot position and fixed. However, robots with planar feet can shift the CoP within the foot support polygon, i.e. foot contact points with the ground, while ensuring stability. Therefore, we propose a stance CoP region $\boldsymbol{\Omega}_s$ spanned by the foot support polygon in which the CoP can be shifted. Figure 4.3c shows the footstep regions with the right foot in stance and a shifted CoP, thus with active CoP adaptation.

Due to less controllability without CoP adaptation, the second target footstep (in Figure 4.3b) needs to be adjusted more to ensure stable running.

[1] The center of pressure (CoP) is the location where the resultant force vector would act if it could be considered to have a single point of application [20].


current left footstep in flight
left target footstep
$\mathcal{X}_{ft,0}$
$\boldsymbol{\Omega}_2$
$\boldsymbol{\mathcal{X}}_{ft,2}$
$\boldsymbol{\Omega}_1$
$\boldsymbol{\mathcal{X}}_{ft,1}$
current right footstep in flight
right target footstep

(a) Flight subphase with both feet in the air.

$\mathcal{X}_{ft,0}$
$\boldsymbol{\Omega}_2$
$\boldsymbol{\mathcal{X}}_{ft,2}$
current right footstep in stance
$\boldsymbol{\mathcal{X}}_{ft,1}$
$\boldsymbol{\Omega}_1 = \boldsymbol{\mathcal{X}}_{ft,1}$

(b) Stance subphase with the right foot in stance and left foot in the air, without CoP adaptation.

$\mathcal{X}_{ft,0}$
$\boldsymbol{\Omega}_2$
$\boldsymbol{\mathcal{X}}_{ft,2}$
$\boldsymbol{\mathcal{X}}_{ft,1}$
$\boldsymbol{\Omega}_s$
CoP

(c) Stance subphase with the right foot in stance and left foot in the air, with active CoP adaptation.

Figure 4.3: Feasible footstep regions for two preview steps and both subphases: (a) flight, (b) stance without CoP adaptation, and (c) stance with CoP adaptation.

4.3 Hopping locomotion

The proposed CoM trajectory planning approach for running locomotion, with single-stance footsteps, can be easily adapted to hopping using double-stance footsteps. Figure 4.4 shows the analogy between the two locomotion gaits. Here, each single-stance footstep from the running locomotion corresponds to one double-stance footstep. The foot trajectories for both feet, during hopping, share the same target foot locations in longitudinal (x-axis) and vertical (z-axis) directions. The lateral offset corresponds to the user-defined step width.

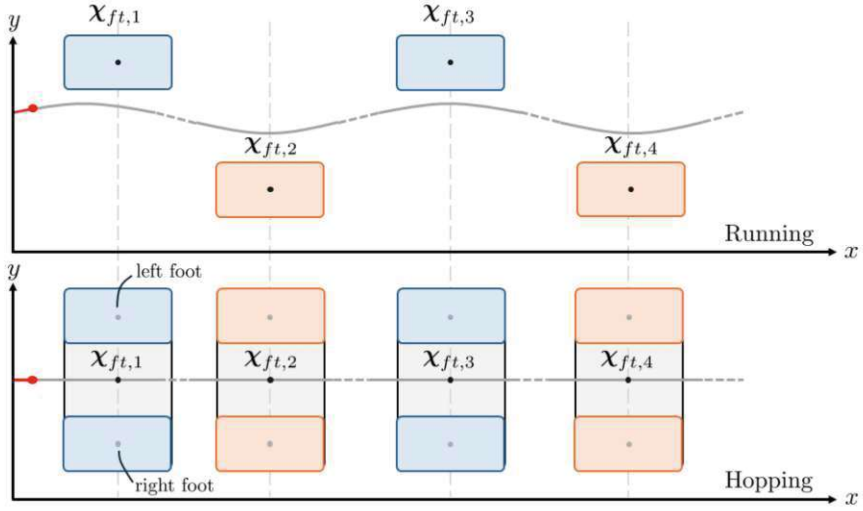
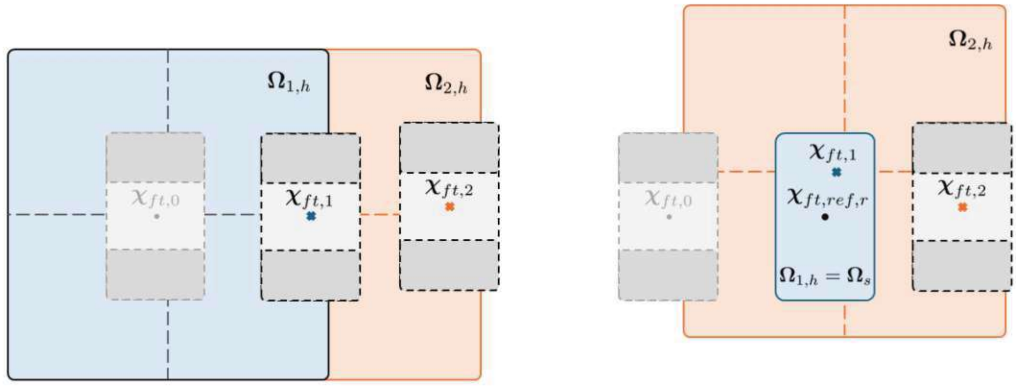


Figure 4.4: Comparison of the planned target foot positions for running and hopping.

Additionally to the adapted footsteps, the footstep regions $\boldsymbol{\Omega}_i$ are adapted. In general, the footstep regions $\boldsymbol{\Omega}_{i,h}$ for hopping are defined using a rectangular-shaped area centered at the previous hopping footstep (center between both previous footsteps). The total length of the area is twice the maximum step length $L_{\max}$. The total width of the area is twice the maximum step width $W_{\max}$. The center of the area is shifted with an offset according to the CoM traveled distance during flight $\boldsymbol{\Delta\chi}_i$. The first footstep region (only in stance) $\boldsymbol{\Omega}_{s,h}$ for hopping is defined within the foot support polygon, spanned by the outer contact points of both feet, i.e. area between the two feet. Therefore, the position of the CoP is restricted within a large area, which offers higher flexibility in case of perturbations or challenging robot configurations.


$\boldsymbol{\Omega}_{1,h}$
$\boldsymbol{\Omega}_{2,h}$
$\chi_{ft,0}$
$\boldsymbol{\chi}_{ft,1}$
$\boldsymbol{\chi}_{ft,2}$
$\boldsymbol{\Omega}_{2,h}$
$\boldsymbol{\chi}_{ft,1}$
$\boldsymbol{\chi}_{ft,ref,r}$
$\boldsymbol{\Omega}_{1,h} = \boldsymbol{\Omega}_s$
$\chi_{ft,0}$
$\boldsymbol{\chi}_{ft,2}$

(a) (b)

Figure 4.5: Feasible footstep regions in hopping for two preview steps and both subphases: (a) flight and (b) stance with CoP adaptation.

4.4 Conclusion

In this chapter, we propose an optimal CoM trajectory planning approach, extending the analytical planning method shown in Chapter 3. The CoM trajectory in the vertical axis is evaluated analytically as proposed in Section 3.1. The horizontal CoM trajectory planning, from Section 3.2, is extended with optimal foot target placement and consequently optimal CoM trajectory waypoints. Additionally, with this method, the CoP is shifted optimally within the stance foot contact area. Thus, two adaptation strategies are used to compensate for perturbations.
In the optimization formulation (Section 4.1), the foot adjustment distance $\Delta\boldsymbol{\chi}_{ft,i}$ is minimized for all preview footsteps, i.e. the distance from the target footstep to the corresponding desired footstep. Additionally, the CoM take-off velocity $\dot{\boldsymbol{\chi}}_{TO,n}$ at the last preview step $i = n$ is minimized to a desired final CoM velocity, e.g. $\dot{\boldsymbol{\chi}}_{TO,n} \overset{!}{=} \mathbf{0}$. The solution for the take-off states from (3.26) is used as an equality constraint, to ensure a stable running CoM trajectory considering the optimized target foot positions.
The placement of the target foot positions is restricted within feasible footstep regions $\boldsymbol{\Omega}$ (see Section 4.2). The footstep regions guarantee kinematic feasibility and prevent leg collisions while running. The CoP in the stance foot is adapted within the foot support polygon (contact points with the ground), enabling higher robustness of the planner.
The last section shows how the CoM trajectory planning can be used to enable a stable hopping locomotion gait. For that, the target footsteps are converted to target hopping footsteps (combining both feet) and the footstep regions $\boldsymbol{\Omega}$ are adapted for hopping to $\boldsymbol{\Omega}_{i,h}$.

The next chapter evaluates the presented method using a point-mass in simulation, assuming ideal CoM and foot position tracking. Subsequently, the method is evaluated using the robot Kangaroo [7] in simulation.

5 Evaluation of the presented planning method

This chapter evaluates the proposed biologically-inspired CoM trajectory planning method using a simulated point-mass (PM) robot model. In particular, we assume ideal CoM and foot position tracking, i.e. the reference output of the planner at timestep t is considered the input at the next timestep $t' = t + \Delta t$.

Figure 5.1 shows a simplified version of the planner structure (from Figure 3.6) with the PM model. The output of the running planner is the desired CoM acceleration $\ddot{\boldsymbol{x}}_{\text{des}}$ and the desired foot trajectories. The PM model integrates the desired CoM acceleration, once to obtain the desired CoM velocity $\dot{\boldsymbol{x}}$, and twice for the desired CoM position $\boldsymbol{x}$. The desired CoM state is fed back to the running planner input. The foot trajectories are fed back to the input of the running planner, which enables ideal foot tracking.

In this chapter, we evaluate the functionality of the proposed method. Furthermore, the optimal CoM trajectory planning approach is compared to the analytical planning approach. The last experiment investigates the robustness of the method with different initial robot configurations while comparing the impact of the CoP adaptation.

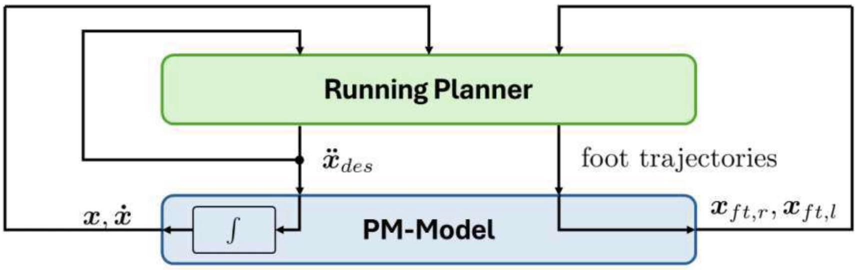

Running Planner
$\ddot{\boldsymbol{x}}_{des}$
foot trajectories
$\boldsymbol{x}, \dot{\boldsymbol{x}}$
$\int$
PM-Model
$\boldsymbol{x}_{ft,r}, \boldsymbol{x}_{ft,l}$

Figure 5.1: Planner structure with the Point-Mass model. Ideal feedback for the CoM state and foot positions.

The planning algorithm was implemented in Python[1] 3.10 using the QP solver qpSWIFT [21] for the foot target optimization from Chapter 4. The simulation frequency of 1ms could not be reached in real-time with the implementation in Python. However, the planning method was also implemented in Matlab/Simulink which was feasible for real-time performance.

The following experiments use a commanded reference CoM velocity of $v_x = 0.5\text{ms}^{-1}$ (in

[1] https://www.python.org/

x-direction). The flight time T_f is set to 0.17s and the stance time T_s is set to 0.33s for each preview phase. Further parameters are found in the appendix Section A.1.

Figure 5.2 shows the spatial view of a running gait using the proposed CoM trajectory planning method. The red line corresponds to the past CoM trajectory and the dashed lines show the flight phases with the corresponding way-points (touch-down and take-off). The preview CoM trajectory is marked in grey. The filled rectangles show the footstep history of each foot separately. The cross- and circle-shaped markers show the target and desired footsteps, respectively.
This experiment shows a stable running pattern, where the adjustment of the first right footstep is sufficient to converge to the desired motion. The optimal trajectory planner uses two strategies to compensate for perturbations, such as a difference in commanded and measured CoM velocity ($v_{x,ref} = 0.5\text{ms}^{-1} > v_{x,ini} = 0.0\text{ms}^{-1}$). The first strategy adapts the CoP in the stance foot within the foot support area to counteract small perturbations. However, larger perturbations need to be adapted using footstep adaptation. In this experiment, the first right footstep is adapted to accelerate the robot to the desired velocity, after that, all target footstep positions correspond to the desired footstep positions. Therefore, step adaptation only affects the first footstep while the CoP adaptation compensates for smaller deviations throughout the process.

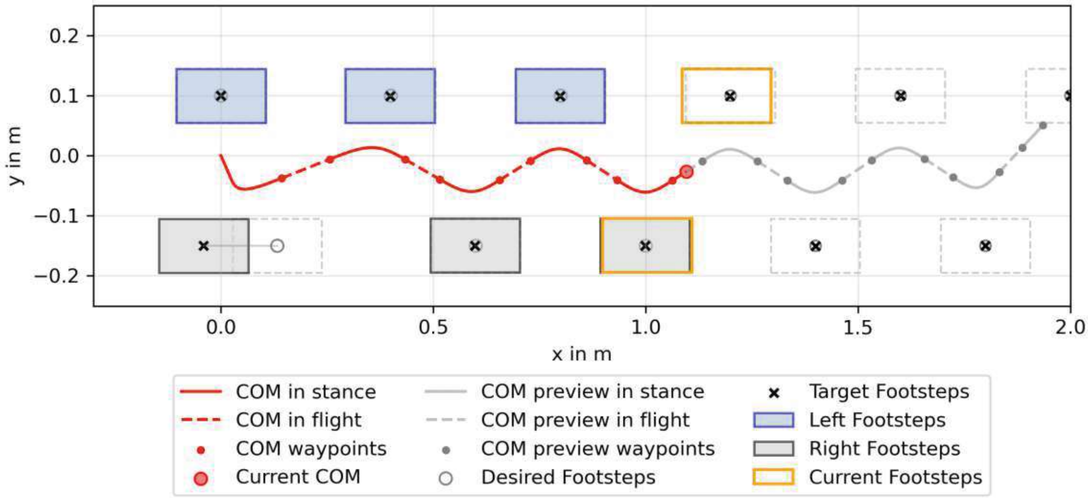


Figure 5.2: Spatial view of a stable running gait using the proposed planning method with optimal target footstep placement and optimal CoM trajectory waypoints..

Figure 5.3 shows the CoM trajectory in the vertical axis (side view). Here, the height at touch-down corresponds to the commanded desired TD height, as proposed in (3.5). The TO height results from the evaluated polynomial splines at the end of each stance phase (3.6).

The CoM behavior in time is shown in Figure 5.4. Here, the horizontal CoM trajectories (x

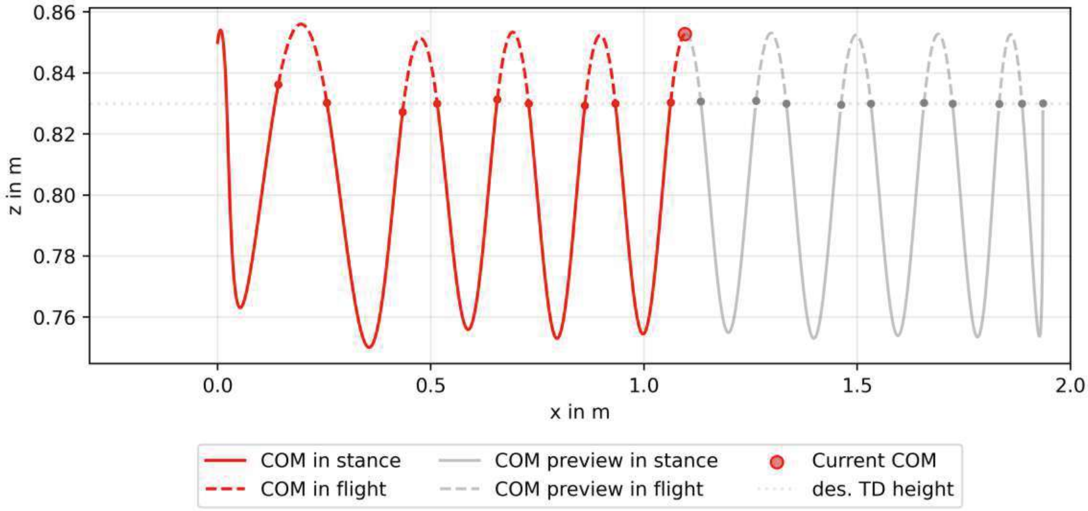


Figure 5.3: Vertical CoM trajectory with optimal target footstep placement.

and y) show linear flight phases, due to zero acceleration in flight (3.10). The vertical CoM trajectory (z-axis) shows parabolic flight curves, corresponding to a constant acceleration.

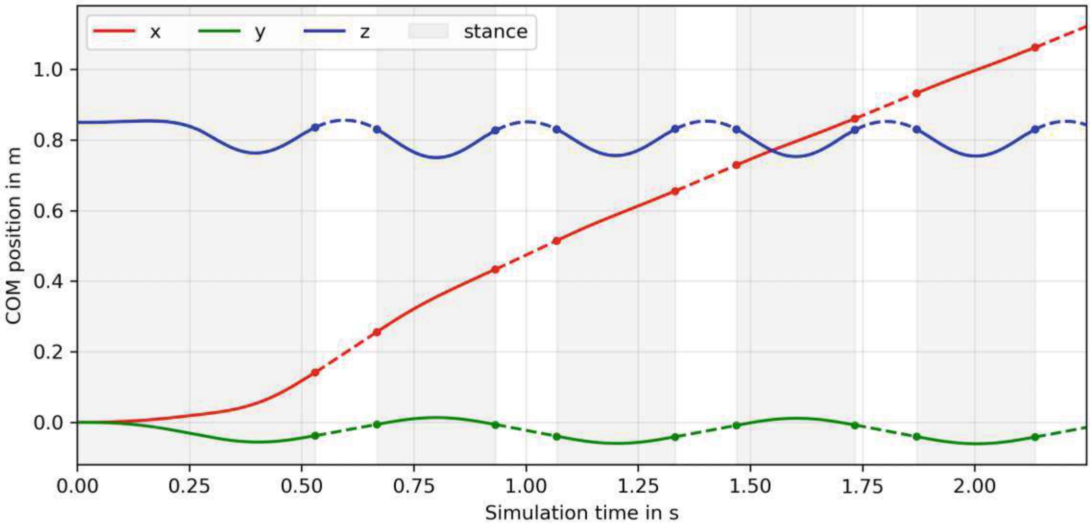


Figure 5.4: CoM trajectories in time with ideal robot behavior (point-mass model).

Figure 5.5 displays the desired leg forces $\boldsymbol{F}_{leg}$, computed using the desired CoM acceleration, robot mass $m = 40kg$ and gravitational vector $\boldsymbol{g}$ as

$$\boldsymbol{F}_{leg} = m(\ddot{\boldsymbol{x}}_{des} + \boldsymbol{g}). \tag{5.1}$$

The vertical component $F_{leg,z}$ shows strictly positive and rather high forces compared to the horizontal components, $F_{leg,x}$ and $F_{leg,y}$. These show repetitive longitudinal forces (x-axis), due to a constant forward velocity. The lateral forces (y-axis) show alternating signs, depending on the current foot in stance. For example in the third stance phase $(1.05 < t < 1.38)$ the right foot is in stance, therefore, the resulting lateral forces are positive (pushing towards the left leg).
The commanded leg forces resemble measured human leg forces (compared with Figure 3.1). Therefore, the biologically inspired CoM trajectory planning successfully imitates the GRF of a human running gait.

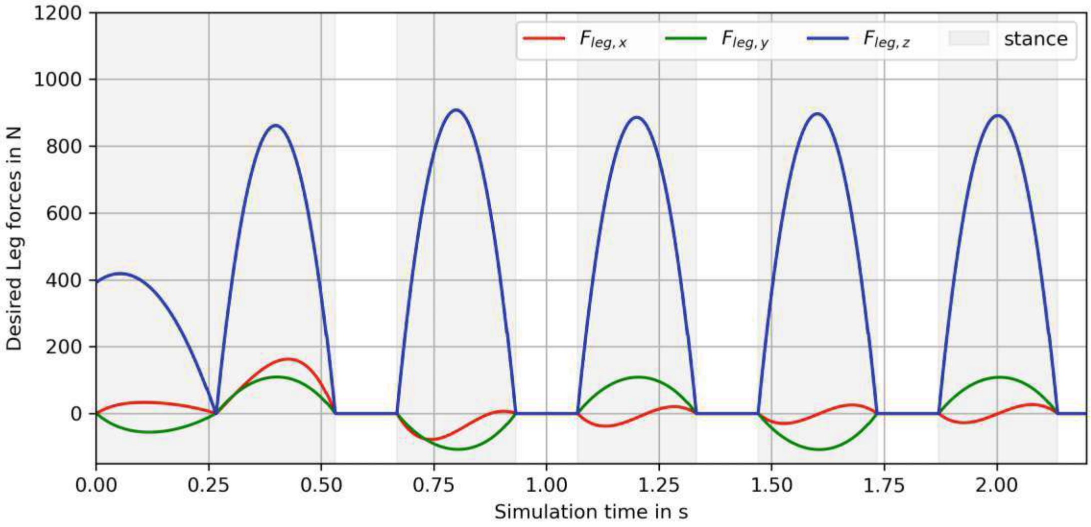


Figure 5.5: Commanded desired robot leg forces (5.1) in Newton.

5.1 Optimal footstep placement

In this experiment, we show that the purely analytical planning method will diverge even for finely tuned initial conditions. On the other hand, the extended optimized planner approach will converge from a large set of initial conditions.
The analytical planner computes the CoM trajectory take-off waypoints (3.26) using fixed target footsteps (desired footsteps). Therefore, a sub-optimal set of desired footsteps and initial robot configurations leads to unstable running, i.e. the robot falls. The optimized planning method minimizes the footstep adjustment distance while ensuring stable running locomotion (4.1). Therefore, the selection of the desired footsteps must not be ideal.
The selection of the initial CoM velocity ($\boldsymbol{v}_{ini} = [0.46, -0.1, -0.2]^T$), for this experiment, was determined empirically. This starting configuration enables stable running, for the analytical planning approach, for a few steps until the CoM trajectory diverges.

The experiment is executed using a reference CoM forward velocity of $v_x = 0.5\text{ms}^{-1}$, flight

time of 0.17s and stance time of 0.33s. Further parameters are found in the appendix Section A.1.
Figures 5.6 and 5.7 show the CoM trajectory and footstep pattern for both planning approaches. The analytical planner approach shows that every target footstep is equal to the corresponding desired footstep, due to fixed target footsteps, defined by the user. At first, the running gait is stable, but due to sub-optimal footstep placement the CoM trajectory diverges after five steps. The continuous trajectory re-planning (online) enables the planner to adapt the reference leg forces (derived from the CoM acceleration), however, this is not sufficient to avoid divergence.

On the other hand, the optimal planner approach (Figure 5.7) shows a stable running gait using optimal target footstep placement. As discussed in the previous experiment, the CoM velocity difference is compensated by step adaptation in the first right footstep. The consecutive footsteps show perfect footstep placement (corresponding to the desired footsteps).

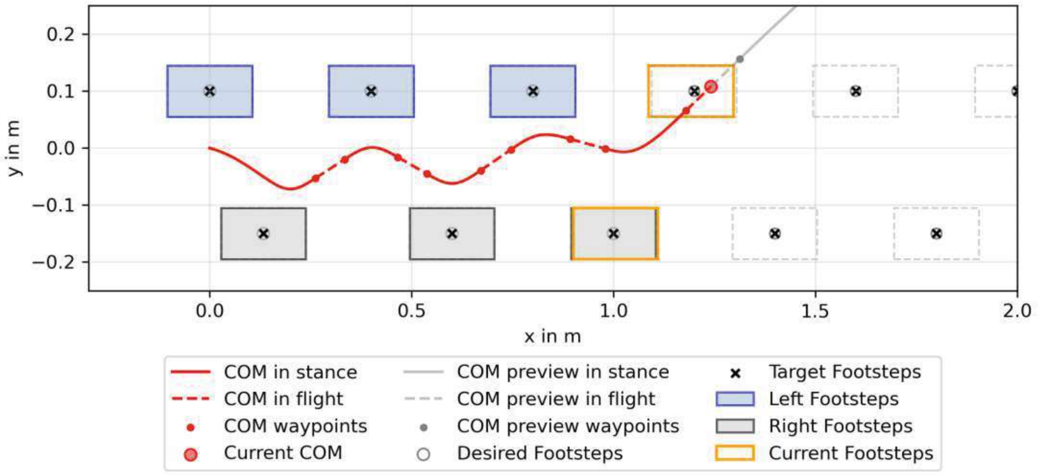


Figure 5.6: Purely analytical planner approach

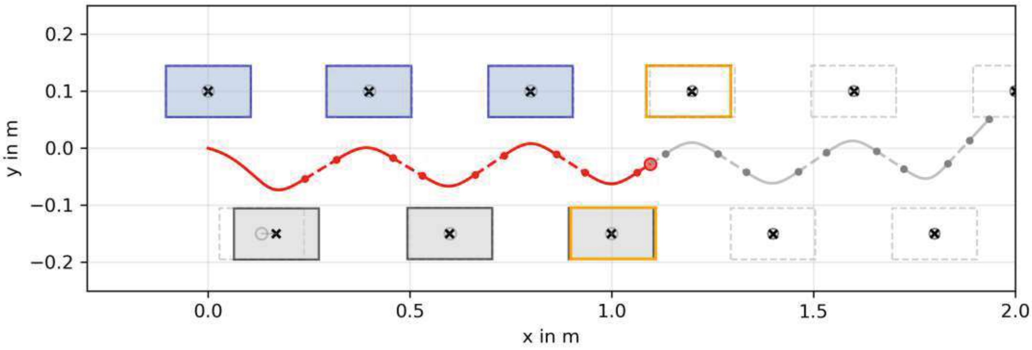


Figure 5.7: Optimized planner approach

5.2 Challenging initial robot configurations

In this experiment, we investigate the robustness of the planning method using three challenging initial configurations, i.e. initial CoM velocities. Additionally, we discuss the impact of the CoP adaptation (see Section 4.2), offering higher robustness for the planning method. The following experiments use the same parameter settings as the previous experiments (see appendix Section A.1), with different initial CoM velocities v_{ini}.

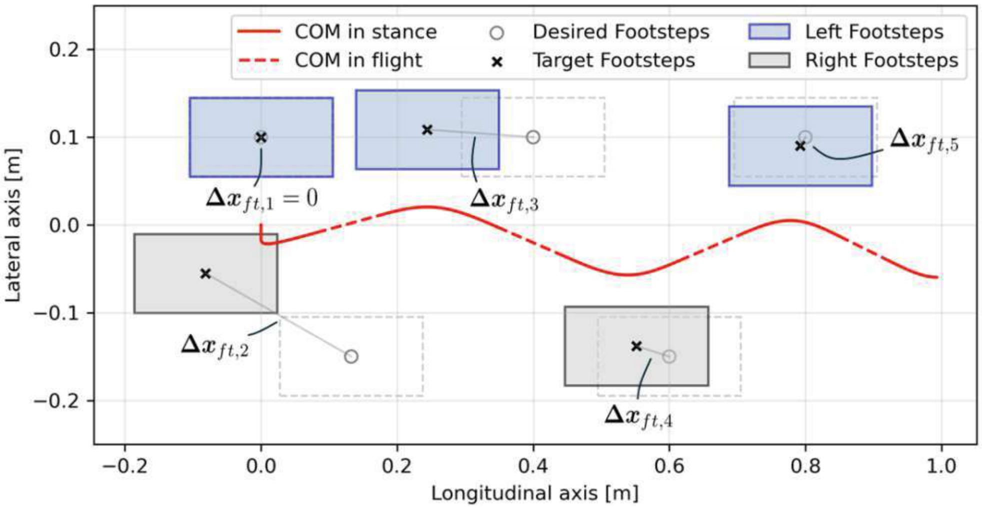
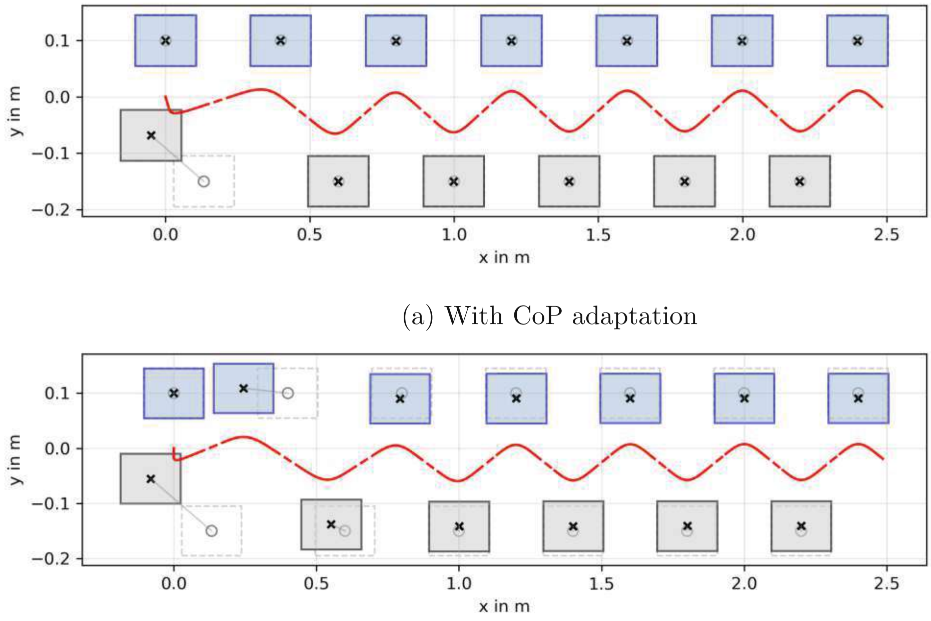


Figure 5.8: Horizontal planning for the CoM in a standstill without CoP adaptation - initial velocity $v_{ini} = [0, 0, 0]^T$.

First, we show the running gait for the CoM in a standstill without CoP adaptation, i.e. initial CoM velocity $v_{ini} = [0, 0, 0]^T$. Figure 5.8 displays the CoM trajectory and footstep placement (filled rectangles) with the corresponding footstep adjustment distance $\Delta\boldsymbol{x}_{ft,i} = \boldsymbol{x}_{ft,i} - \boldsymbol{x}_{ft,i,des}$, i.e. difference between optimized and desired footstep locations. The first footstep is considered as the initial footstep in stance (fixed), thus the first footstep distance $\Delta\boldsymbol{x}_{ft,1}$ is always zero. The consecutive footsteps $i > 1$ show a significant footstep distance, which converges after five footsteps. This footstep pattern achieves a stable running motion using only step adaptation (CoP adaptation inactive).

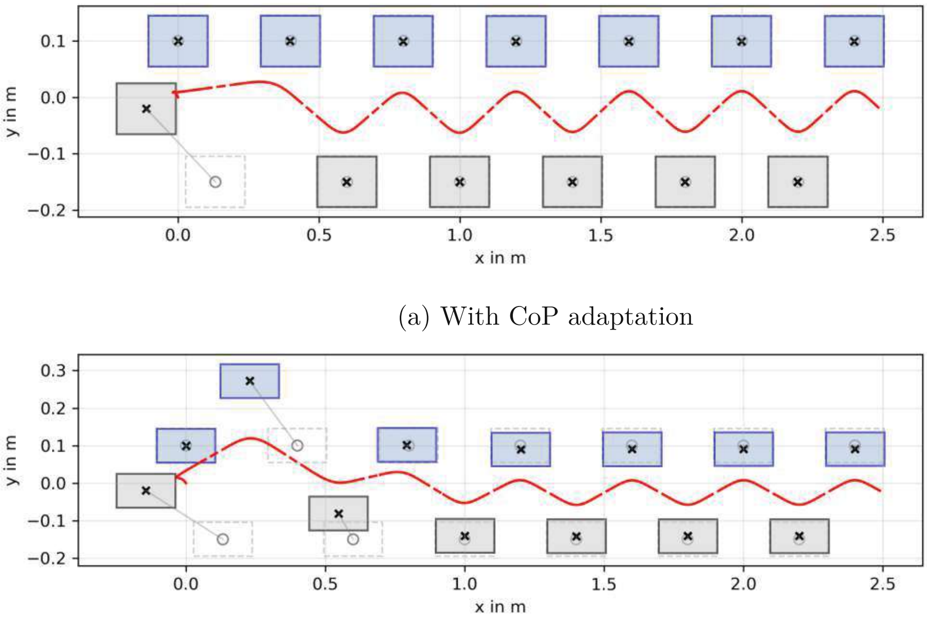
Next, we compare running gaits with three challenging initial robot configurations, with and without CoP adaptation.
The first experiment (Figure 5.9) shows a running pattern with the robot in a standstill. In the second and third experiment (Figure 5.10 and Figure 5.11) the robot starts in the opposite direction of the desired running direction and different lateral velocities. All experiments show stable running gaits. The use of CoP adaptation acts as an additional strategy to compensate for disturbances (e.g. challenging initial conditions), in addition to footstep adjustment.



(a) With CoP adaptation

(b) Without CoP adaptation

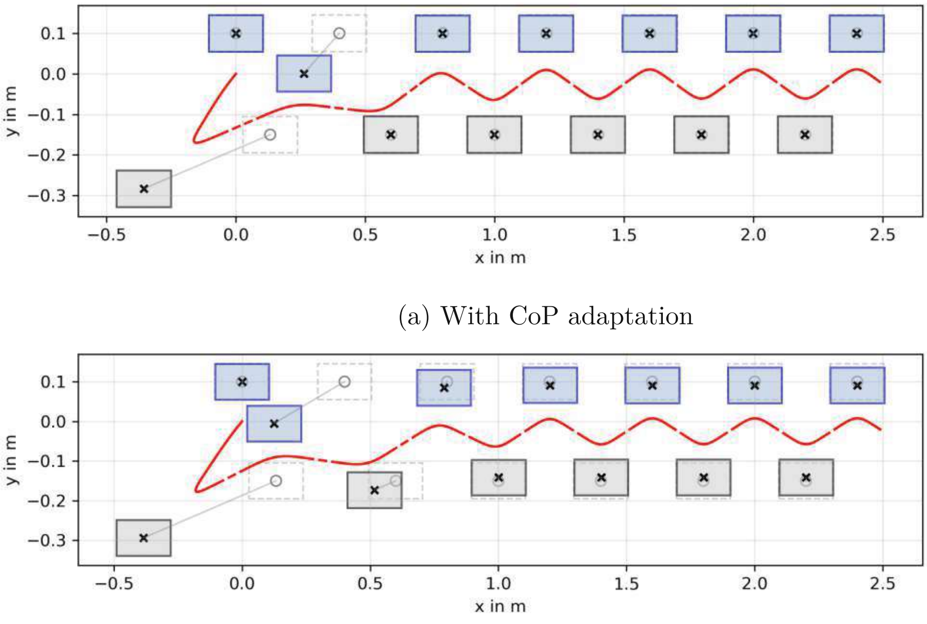
Figure 5.9: Comparision of the horizontal planning with and without CoP adaptation using the initial CoM velocity $v_{ini} = [0, 0, 0]^T$ (standstill).



(a) With CoP adaptation

(b) Without CoP adaptation

Figure 5.10: Comparision of the horizontal planning with and without CoP adaptation using the initial CoM velocity $v_{ini} = [-0.1, 0.1, 0.0]^T$.



(a) With CoP adaptation

(b) Without CoP adaptation

Figure 5.11: Comparision of the horizontal planning with and without CoP adaptation using the initial CoM velocity $v_{ini} = [-0.5, -0.4, 0.0]^T$.

The convergence of the footstep adjustment distance is used as a metric for comparison. Figure 5.12 shows the L2-norm of the footstep adjustment distance $||\boldsymbol{x}_{ft,i}||^2$ for various initial configurations. Colors separate the different initial configurations. Continuous lines and filled circle-shaped markers indicate active CoP adaptation. For each initial configuration, active CoP adaptation causes a faster footstep distance convergence. Additionally, the footstep distance after convergence is much lower for the experiments with active CoP adaptation. Without adaptation, the footstep distance converges to approximately 9mm, for all starting configurations. With adaptation, the footstep distance converges to approximately 1.8mm, for all starting configurations.

Due to a not optimally selected step width, the optimal footsteps without CoP adaptation are moved closer to the body, achieving a higher footstep distance after convergence. This is likely due to the optimization selecting a smaller optimal step width for faster running. The CoP adaptation compensates for this effect and enables a small footstep distance after convergence. At even higher velocities, the converged footstep distance will decrease even for active CoP adaptation.

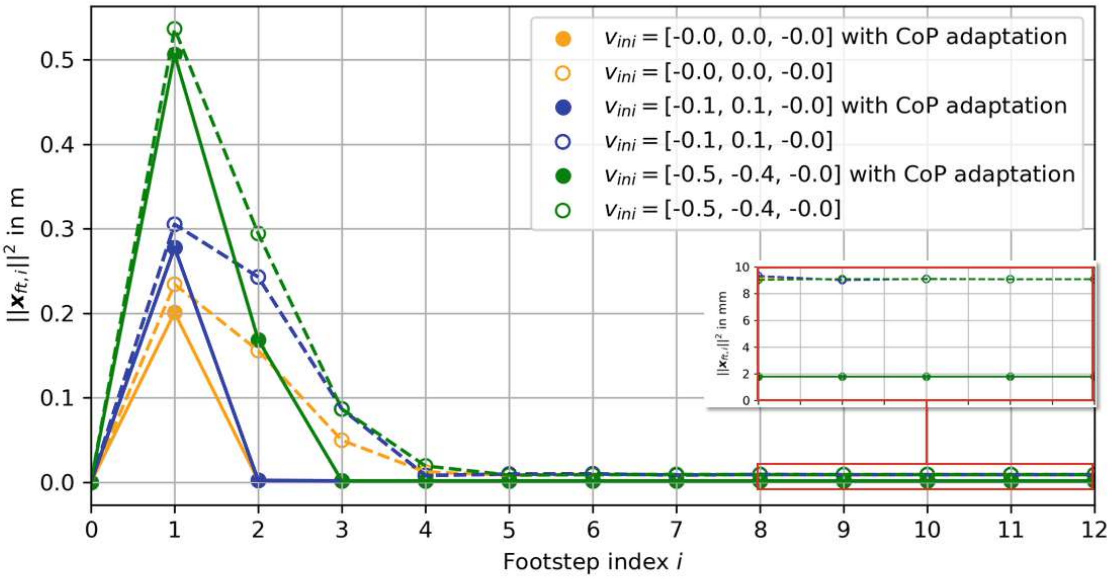


Figure 5.12: Norm of the footstep adjustment distance $||\boldsymbol{\Delta x}_{ft,i}||^2 = ||\boldsymbol{x}_{ft,i,target} - \boldsymbol{x}_{ft,i,des}||^2$ for different initial CoM velocities v_{ini}.

5.3 Conclusion

In this chapter, the proposed biologically-inspired CoM planning method is evaluated using a point-mass model, assuming ideal CoM and foot position tracking. The online planner is tested for a forward velocity of $v_x = 0.5\text{ms}^{-1}$ converging to a stable running motion. The commanded leg forces, retrieved by the reference CoM acceleration, replicate human leg forces from running experiments on a force plate treadmill [3].
Next, the purely analytical planner approach is evaluated showing a diverging running gait, even for finely tuned parameters and starting conditions. On the other hand, the optimized planning approach compensates the initial difference in the CoM measured and commanded velocity, resulting in a stable running gait.

In the last experiment, various challenging initial robot configurations are tested, such as running from a standstill. Additionally, CoP adaptation is compared to a fixed CoP in the stance foot. Generally, active CoP adaptation shows a faster convergence for the footstep distance (difference between the target and desired footstep position), as well as a lower overall footstep distance after convergence.

The following chapters introduce the whole-body control framework and integrate the optimized planning method in a simulated robot using a whole-body controller.

6 Whole-Body Control

This chapter presents an overview of the whole-body control (WBC) framework, which enables humanoid robots to execute a variety of tasks, such as balancing, running, or other complex movements. Humanoids typically have a large number of degrees of freedom (DoF) that are exploited to various degrees for different tasks. The WBC framework is used to find the optimal control outputs while executing various tasks. For instance, a controlled body orientation plays an important role in balancing during running and can be implemented as a WBC task. This work uses an inverse-dynamics approach introduced by Englsberger [22] and is embedded in an QP based optimization framework. In this work, the WBC is used to enable stable running using commanded reference foot trajectories and a commanded reference CoM acceleration.

6.1 Dynamic model

Humanoid robots are described with a dynamic free-floating model, gaining an additional six degrees of freedom compared to stationary robots. These DoF make the system underactuated because they are not controllable directly. The floating base state vector $\boldsymbol{q}_{base} \in \mathbb{R}^6$ is composed by the base position $\boldsymbol{x}_{base} \in \mathbb{R}^3$ and the base local orientation $\boldsymbol{\alpha}_{base} \in \mathbb{R}^3$, e.g. roll-pitch-yaw angles. The robot joints are included in the joint state vector $\boldsymbol{q}_{jnt} \in \mathbb{R}^n$. The extended state vector results in

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{x}_{base} \\ \boldsymbol{\alpha}_{base} \\ \boldsymbol{q}_{jnt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{base} \\ \boldsymbol{q}_{jnt} \end{bmatrix}. \tag{6.1}$$

The equation of motion for a free-floating dynamic model is defined as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_g(\boldsymbol{q}) = \boldsymbol{S}^T\boldsymbol{\tau} + \boldsymbol{\tau}_{ext}. \tag{6.2}$$

Here, the system matrices are denoted with the inertia matrix $\boldsymbol{M}$ and the coriolis and centrifugal matrix $\boldsymbol{C}$. The vector $\boldsymbol{\tau}_g$ includes the gravitational forces and moments. The matrix $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{0}_{nx6} & \boldsymbol{I}_{nxn} \end{bmatrix}$ is a selection matrix for the torque vector $\boldsymbol{\tau}$, that selects only the n joint torques that are actuated directly. The generalized forces and torques acting on the robot are combined in the vector $\boldsymbol{\tau}_{ext}$.

6.2 Contact constraints

Due to the gravitational force, the free-floating robot body moves towards the ground surface. Thus, the robot needs to make contact with its environment. The contact is modeled by approximating each robot foot with a rectangular foot contact area, each spanned by four contact points (see figure 6.1). For each contact point j, three

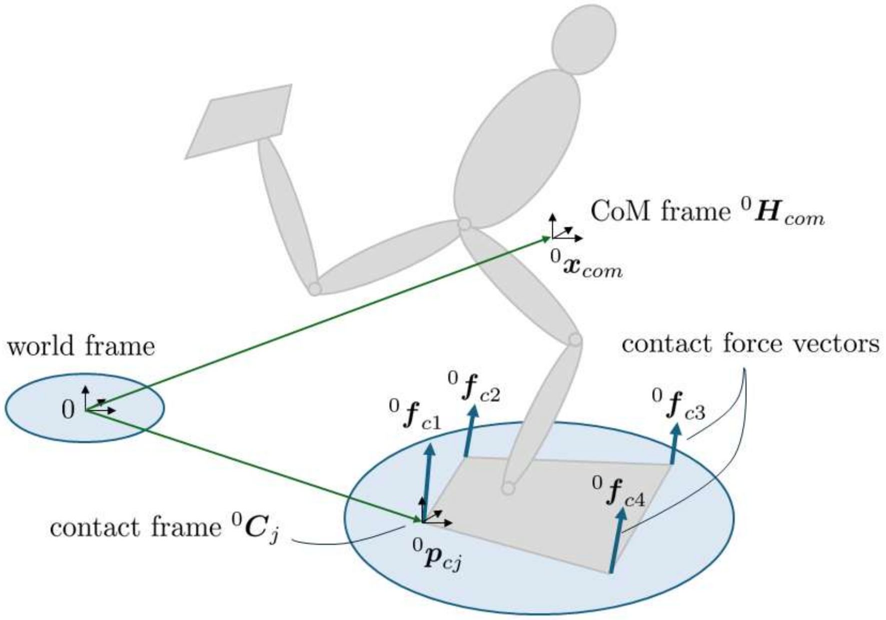

CoM frame ${}^0\boldsymbol{H}_{com}$
${}^0\boldsymbol{x}_{com}$
world frame
0
contact force vectors
${}^0\boldsymbol{f}_{c2}$
${}^0\boldsymbol{f}_{c1}$
${}^0\boldsymbol{f}_{c3}$
${}^0\boldsymbol{f}_{c4}$
contact frame ${}^0\boldsymbol{C}_j$
${}^0\boldsymbol{p}_{cj}$

Figure 6.1: Outline of CoM frame, contact frame, and contact force vectors. Figure adapted from [22].

perpendicular contact force vectors are considered ${}^0\boldsymbol{f}_{cj} = \begin{bmatrix} f_{cj,x} & f_{cj,y} & f_{cj,z} \end{bmatrix}^T$. The prefix ${}^0\square$ indicates, that $\square$ is represented in the world frame. The generalised contact force vector ${}^0\boldsymbol{F}_{cf,k} = \begin{bmatrix} {}^0\boldsymbol{f}_{c1}^T & \cdots & {}^0\boldsymbol{f}_{c4}^T \end{bmatrix}^T \in \mathbb{R}^{12}$ combines the contact force vectors ${}^0\boldsymbol{f}_{cj}$ for four contact points for each foot $k = 2$, resulting in a total of $n_c = 8$ contact points. The generalised spacial wrench ${}^0\boldsymbol{w}_{cf,k}$ for each foot k is defined as

$$ {}^0\boldsymbol{w}_{cf,k} = \begin{bmatrix} {}^0\boldsymbol{f}_{cf,k} \\ {}^0\boldsymbol{\omega}_{cf,k} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{I}_{3\times3} & \cdots & \boldsymbol{I}_{3\times3} \\ {}^0\tilde{\boldsymbol{p}}_{c1} & \cdots & {}^0\tilde{\boldsymbol{p}}_{c4} \end{bmatrix}}_{\boldsymbol{J}^T_{cf,k}} \underbrace{\begin{bmatrix} {}^0\boldsymbol{f}_{c1} \\ \vdots \\ {}^0\boldsymbol{f}_{c4} \end{bmatrix}}_{{}^0\boldsymbol{F}_{cf,k}} \tag{6.3} $$

where ${}^0\tilde{\boldsymbol{p}}_{cj}$ is the skew-symmetric cross-product matrix

$$ {}^0\tilde{\boldsymbol{p}}_{cj} = \begin{bmatrix} 0 & -p_{cj,z} & p_{cj,y} \\ p_{cj,z} & 0 & -p_{cj,x} \\ -p_{cj,y} & p_{cj,x} & 0 \end{bmatrix} \tag{6.4} $$

for the vector ${}^0\boldsymbol{p}_{cj} = \begin{bmatrix} p_{cj,x} & p_{cj,y} & p_{cj,z} \end{bmatrix}^T$, denoting the position of the contact frame ${}^0\boldsymbol{C}_j$. Further, ${}^0\boldsymbol{f}_{cf,k} = \begin{bmatrix} f_{x,k} & f_{y,k} & f_{z,k} \end{bmatrix}^T$ and ${}^0\boldsymbol{\omega}_{cf,k} = \begin{bmatrix} \omega_{cf,k,x} & \omega_{cf,k,y} & \omega_{cf,k,z} \end{bmatrix}^T$ denote the generalized force vector and generalized angular momentum vector for one foot, respectively.

The equation of motion, with inserted contact forces, results to

$$
\begin{aligned}
\boldsymbol{M}\ddot{\boldsymbol{q}} + \underbrace{\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_g}_{\boldsymbol{h}} &= \boldsymbol{S}^T\boldsymbol{\tau} + \sum_{k=1}^{2}(\underbrace{{}^0\boldsymbol{J}_{f,k}^T\,{}^0\boldsymbol{J}_{cf,k}^T}_{\boldsymbol{J}_k^T}\boldsymbol{F}_{cf,k}) \\
&= \underbrace{\begin{bmatrix}\boldsymbol{S}^T & \boldsymbol{J}_1^T & \boldsymbol{J}_2^T\end{bmatrix}}_{\boldsymbol{A}_u} \underbrace{\begin{bmatrix}\boldsymbol{\tau} \\ \boldsymbol{F}_{cf,1} \\ \boldsymbol{F}_{cf,2}\end{bmatrix}}_{\boldsymbol{u}}.
\end{aligned}
\tag{6.5}
$$

Here, ${}^0\boldsymbol{J}_{f,k}^T$ denotes the spacial end effector Jacobian corresponding to the k-th end effector wrench ${}^0\boldsymbol{w}_k$. The control input vector $\boldsymbol{u}$ combines the robot´s actuated joint torques and contact forces for both feet. The matrix $\boldsymbol{A}_u$ combines the selection matrix $\boldsymbol{S}$ and the contact force mapping matrix $\boldsymbol{J}_{cf}$. These map the control input vector $\boldsymbol{u}$ to the generalized joint torques. The generalized accelerations $\ddot{\boldsymbol{q}}$ are found as

$$
\ddot{\boldsymbol{q}} = \underbrace{-\boldsymbol{M}^{-1}\boldsymbol{h}}_{\ddot{\boldsymbol{q}}_{MB}} + \underbrace{\boldsymbol{M}^{-1}\boldsymbol{A}_u}_{\boldsymbol{Q}}\boldsymbol{u}. \tag{6.6}
$$

Here, $\boldsymbol{Q}$ maps the control inputs $\boldsymbol{u}$ to the generalized accelerations $\ddot{\boldsymbol{q}}$. In $\ddot{\boldsymbol{q}}_{MB}$ the acceleration due to multi-body effects is included.

Since the robot typically relies on pushing off the ground and moves forward primarily due to friction, it is essential to account for unilateral contact and friction constraints. The friction constraints [23] are defined, for each contact point j, as

$$
\begin{aligned}
|f_{cj,x}| &\leq \mu f_{cj,z} \\
|f_{cj,y}| &\leq \mu f_{cj,z} \\
f_{cj,z} &\geq 0
\end{aligned}
\tag{6.7}
$$

6.3 Whole-body controller tasks

The whole-body controller enables robots to perform basic activities like walking and running. Due to the high number of joints, the controller structure is separated into multiple whole-body control tasks to find a feasible control output for all joints. The whole-body control tasks are defined using the control input vector $\boldsymbol{u}$ as:

$$
\boldsymbol{d}_{i,des} \overset{!}{=} \boldsymbol{D}_i\boldsymbol{u}, \tag{6.8}
$$

with the desired task space vector $\boldsymbol{d}_{i,des}$ and the task space mapping matrix $\boldsymbol{D}_i$ for the corresponding i-th task.

6.3.1 CoM acceleration task

One of the outputs of the running planner (3) is the desired CoM acceleration $\ddot{\boldsymbol{x}}_{des}$. To map the acceleration to the control input $\boldsymbol{u}$, the following transformation of the CoM acceleration to the leg force is considered:

$$\boldsymbol{F}_{leg,des} = m(\ddot{\boldsymbol{x}}_{des} + \boldsymbol{g}), \tag{6.9}$$

where m is the total robot weight and $\boldsymbol{g}$ is the gravitational vector. The resulting leg force is then mapped to the control input $\boldsymbol{u}$ as

$$\underbrace{-\boldsymbol{F}_{leg,des}}_{\boldsymbol{d}_{com,des}} = \underbrace{\boldsymbol{J}_{com}}_{\boldsymbol{D}_{com}} \boldsymbol{u}. \tag{6.10}$$

Here, $\boldsymbol{J}_{com}$ maps the foot contact forces to the desired leg force.

6.3.2 Foot acceleration task

The foot acceleration task aims to follow a commanded reference foot trajectory (see 3.3.2). A task space PD controller is used:

$$\ddot{\boldsymbol{x}}_{f,des} = \ddot{\boldsymbol{x}}_{f,ref} + k_d(\dot{\boldsymbol{x}}_f - \dot{\boldsymbol{x}}_{f,ref}) + k_p(\boldsymbol{x}_f - \boldsymbol{x}_{f,ref}). \tag{6.11}$$

Here, the desired foot acceleration $\ddot{\boldsymbol{x}}_{f,des}$ is computed using the reference foot acceleration $\ddot{\boldsymbol{x}}_{f,ref}$, the foot velocity error weighted with $k_d > 0$, and the foot position error weighted with $k_p > 0$. The hybrid foot velocity ${}^0\boldsymbol{v}_f$ introduced by Murray et al. [24],

$${}^0\boldsymbol{v}_f = \begin{bmatrix} {}^0\dot{\boldsymbol{x}}_f \\ {}^0\boldsymbol{\omega}_f \end{bmatrix} = {}^0\boldsymbol{J}_f\dot{\boldsymbol{q}} \tag{6.12}$$

combines the translational velocity ${}^0\dot{\boldsymbol{x}}_{\boldsymbol{f}}$ with the angular velocity ${}^0\boldsymbol{\omega}_{\boldsymbol{f}}$. The hybrid Jacobian ${}^0\boldsymbol{J}_f$ maps the generalized joint velocities to the hybrid velocity vector. The derivative of ${}^0\boldsymbol{v}_f$ results in

$${}^0\dot{\boldsymbol{v}}_f = {}^0\dot{\boldsymbol{J}}_f\dot{\boldsymbol{q}} + {}^0\boldsymbol{J}_f\ddot{\boldsymbol{q}}. \tag{6.13}$$

The derivative of the desired hybrid velocity vector is defined as

$${}^0\dot{\boldsymbol{v}}_{f,des} = \begin{bmatrix} \ddot{\boldsymbol{x}}_{f,des} \\ \dot{\boldsymbol{\omega}}_{f,des} \end{bmatrix}. \tag{6.14}$$

The desired angular velocity is set to zero ${}^0\boldsymbol{\omega}_{\boldsymbol{f,des}} = \boldsymbol{0}_{3x1}$, thus not allow foot rotation during the swing phase. We insert the generalized joint accelerations 6.6 into the derived hybrid velocity vector 6.13 and set ${}^0\dot{\boldsymbol{v}}_f = {}^0\dot{\boldsymbol{v}}_{f,des}$, resulting to

$$\underbrace{{}^0\dot{\boldsymbol{v}}_{f,des} - {}^0\dot{\boldsymbol{J}}_f\dot{\boldsymbol{q}} - {}^0\boldsymbol{J}_f\ddot{\boldsymbol{q}}_{MB}}_{\boldsymbol{d}_{f,des}} = \underbrace{{}^0\boldsymbol{J}_f\boldsymbol{Q}}_{\boldsymbol{D}_f} \boldsymbol{u}. \tag{6.15}$$

Thus, the foot acceleration task can be written as a function of the control input vector $\boldsymbol{u}$ using the desired foot acceleration task vector $\boldsymbol{d}_{f,des}$ and the task mapping matrix $\boldsymbol{D}_f$.

6.3.3 Further tasks

A stable running locomotion gait requires more WBC tasks, e.g. to stabilize the torso´s orientation. Table 6.1 decodes the desired task vector $\boldsymbol{d}_{i,des}$ and the task mapping matrix $\boldsymbol{D}_i$ for each task i separately. The torso orientation task is defined similarly to the foot acceleration task, only considering the angular terms, i.e. lower half of the hybrid velocity vector. The Jacobian $\boldsymbol{J}_{torso}$ maps the generalized joint accelerations to the hybrid torso velocity vector.

Another important task forces the foot to stay in contact during the stance phase. Here, the foot acceleration notation is used with zero desired foot acceleration $\ddot{\boldsymbol{x}}_{f,des} = \boldsymbol{0}$, thus the stance foot should not move.

Task	Desired task vector $\boldsymbol{d}_{i,des}$	Task mapping matrix $\boldsymbol{D}_i$
CoM acceleration	$-m(\ddot{\boldsymbol{x}}_{des} + \boldsymbol{g})$	${}^0\boldsymbol{J}_{com}$
Foot acceleration	${}^0\dot{\boldsymbol{v}}_{f,des} - {}^0\dot{\boldsymbol{J}}_f\dot{\boldsymbol{q}} - {}^0\boldsymbol{J}_f\ddot{\boldsymbol{q}}_{MB}$	${}^0\boldsymbol{J}_f\boldsymbol{Q}$
Foot contact tracking	${}^0\dot{\boldsymbol{J}}_f\dot{\boldsymbol{q}} - {}^0\boldsymbol{J}_f\ddot{\boldsymbol{q}}_{MB}$	${}^0\boldsymbol{J}_f\boldsymbol{Q}$
Torso orientation	${}^0\dot{\boldsymbol{\omega}}_{torso,des} - {}^0\dot{\boldsymbol{J}}_{torso}\dot{\boldsymbol{q}} - {}^0\boldsymbol{J}_{torso}\ddot{\boldsymbol{q}}_{MB}$	${}^0\boldsymbol{J}_{torso}\boldsymbol{Q}$

Table 6.1: Whole-body controller tasks with their corresponding desired task vector $\boldsymbol{d}_{i,des}$ and task mapping matrix $\boldsymbol{D}_i$

6.4 Optimization via a quadratic program

The cost function G is minimized in the QP using the task errors $\boldsymbol{D}_i\boldsymbol{u} - \boldsymbol{d}_{i,des}$:

$$\begin{aligned}
\min_{\boldsymbol{u}} \quad & G = \frac{1}{2}\sum_i ((\boldsymbol{D}_i\boldsymbol{u} - \boldsymbol{d}_{i,des})^T \boldsymbol{W}_i(\boldsymbol{D}_i\boldsymbol{u} - \boldsymbol{d}_{i,des})), \\
\text{s.t.} \quad & -\boldsymbol{\tau}_{max} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{max}, \\
& |f_{cj,x}| \leq \mu f_{cj,z}, \\
& |f_{cj,y}| \leq \mu f_{cj,z}, \qquad \forall j \in n_c, \\
& 0 \leq f_{cj,z} \leq f_{z,max},
\end{aligned} \tag{6.16}$$

where $\boldsymbol{W}_i$ are diagonal weighting matrices for each task i and $\boldsymbol{\tau}_{max}$ denotes the joint torque limits. The residual inequality constraints decode the friction constraints for each contact point $j \in n_c = 8$. The solution yields the optimal input vector $\boldsymbol{u}$, which balances the individual tasks based on their respective weights while ensuring compliance with torque limits and contact friction constraints.

6.5 Conclusion

This Chapter outlines the structure of the inverse-dynamics whole-body control framework, adapted from [22]. First, the free-floating dynamic model for a humanoid robot is introduced with the corresponding equation of motion. The contact constraints are modelled with force vectors, one for each foot corner point with corresponding friction constraints.
Next, various whole-body controller tasks are presented, e.g. CoM acceleration tracking, torso orientation tracking, and foot tracking. The combination of all tasks enables the robot to perform basic activities, such as running, while also balancing the robot. In the last part, the whole-body tasks are integrated into a QP optimization. Here, constraints are applied to the maximum robot torques and friction.

In the next chapter, the proposed planning method is integrated on the robot Kangaroo in simulation using a whole-body controller and evaluated.

7 Whole-Body Simulation

This chapter evaluates the proposed biologically-inspired CoM trajectory planning method using a simulated robot model of the robot Kangaroo [7] from PAL Robotics. The robot is a bipedal humanoid robot without arms with an approximate height of 1.45m and a weight of 40kg, specially designed for dynamic running motions.
The robot is simulated in the physics simulator MuJoCo 3.2.0 [6], short for Multi-Joint dynamics with Contact, using Python 3.10. For the foot target optimization, we use the QP solver qpSWIFT [21]. The optimization of the control input vector $\boldsymbol{u}$ in the whole-body control framework (6.16) is solved with the QP solver OSQP [25]. The controller and simulator frequency is set to 1kHz.

Figure 7.1 shows a simplified version of the planner structure (from Figure 3.6) with the whole-body control framework and the robot model. The output of the running planner is the desired CoM acceleration and desired foot trajectories. These act as an input reference for the whole-body control framework. The WBC is used to find the optimal control input while executing various tasks, e.g. CoM acceleration tracking and foot acceleration tracking. The output of the WBC represents the torque commands for each of the robot´s joints. Further details on the whole-body control framework can be found in Chapter 6. The current robot´s CoM position and velocity, as well as the current foot positions, are fed back to the running planner input from the WBC. The CoM reference acceleration is directly fed back to the planner input, due to high noise levels in accelerometer sensors.

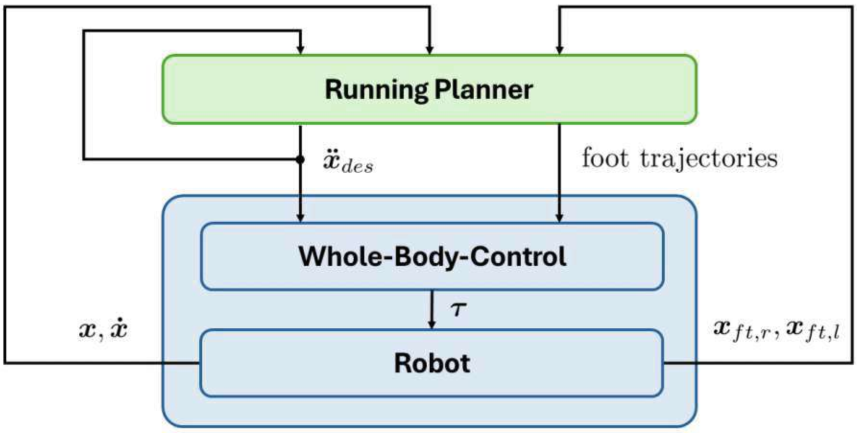
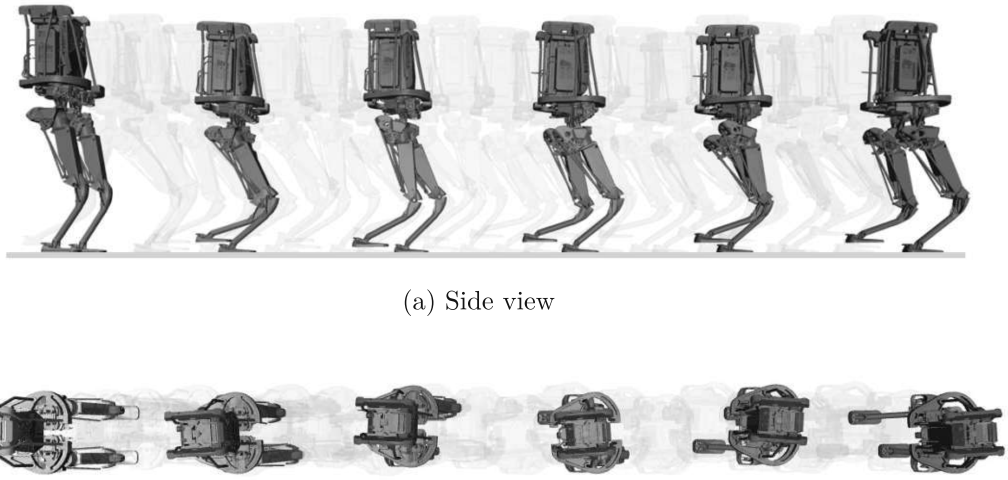

Running Planner
$\ddot{\boldsymbol{x}}_{des}$
foot trajectories
Whole-Body-Control
$\boldsymbol{\tau}$
$\boldsymbol{x}, \dot{\boldsymbol{x}}$
$\boldsymbol{x}_{ft,r}, \boldsymbol{x}_{ft,l}$
Robot

Figure 7.1: Modified planning structure (from Figure 3.6) with the integrated whole-body controller and robot model.

The first experiment on the simulated real-robot platform Kangaroo shows a running gait with a commanded reference CoM velocity of 0.5m/s, starting from a standstill ($v_{ini} = [0, 0, 0]^T$). The step frequency is set to $f_{step} = 2.5$Hz (steps per second) with a stance percentage of 66% ($c_s = 0.66$). Thus, one phase (stance subphase and flight subphase) lasts $T_{phase} = f_{step}^{-1} = 0.4$s, with a stance time of $T_s = c_s T_{phase} = 0.264$s and a flight time of $T_f = (1 - c_s)T_{phase} = 0.136$s. Further parameters used for this experiment are shown in Appendix A.2.
Figure 7.2 shows a timeseries of the running gait in simulation.

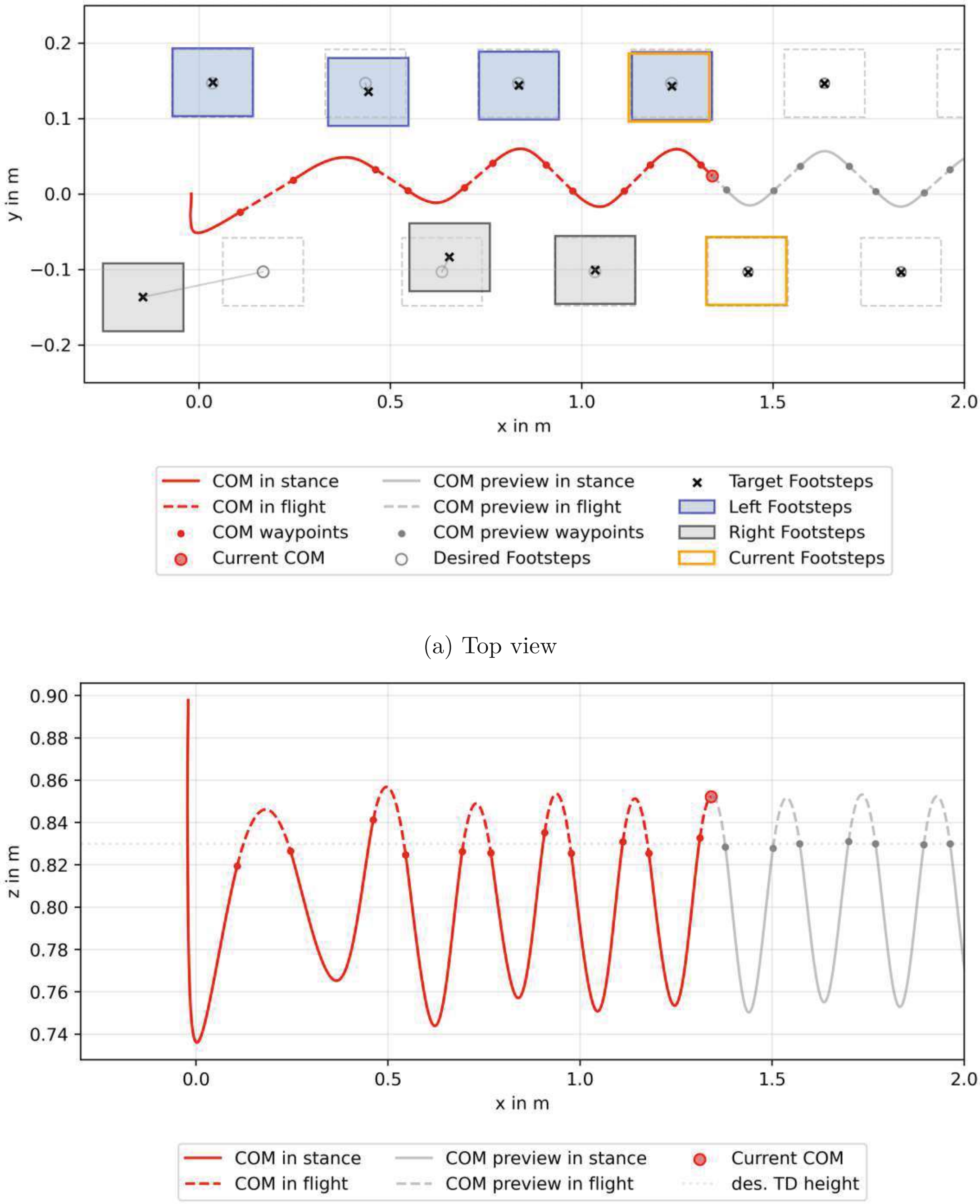


(a) Side view

(b) Top view

Figure 7.2: Timeseries of the humanoid robot Kangaroo [7] running in the simulation environment MuJoCo with a reference CoM forward velocity of $v_{ref,x} = 0.5$.

The spatial views (top view and side view) of the planned CoM trajectory and planned footsteps are shown in Figure 7.3.
Due to a standstill at the start, the robot needs to adapt the first right footstep to the back to achieve the desired running speed (see the grey footstep in the left lower corner of Figure 7.3a). After three footsteps, the footstep error converges to zero and the desired CoM velocity is reached. The point-mass model results (Figure 5.2) from Chapter 5 with the same settings, show similar results. The footstep error converges faster, after one footstep. Therefore, the WBC framework manages to track the reference trajectories successfully.
The vertical CoM trajectory, see Figure 7.3b, shows that the running motion is stable and corresponds to the planned trajectory. The initial CoM height is 0.9m and the desired touch-down height (grey dashed horizontal line) is 0.83m. The measured touch-down height shows an average height difference of $\Delta z = 5$mm to the desired touch-down height. Due to fixed timing, the upcoming vertical touch-down position is estimated using 3.30, thus the actual CoM touch-down height does not necessarily need to correspond with the desired TD height.



(a) Top view

(b) Side view

Figure 7.3: Spatial view of a stable running gait using the proposed planning method with optimal target footstep placement and optimal CoM trajectory waypoints. Simulated with the robot Kangaroo in a standstill ($v_{ini} = [0, 0, 0]^T$).

Figure 7.4 displays the trajectories for the measured foot positions (continuous lines) and the corresponding target foot positions (dashed lines). The foot trajectories are computed, as shown in Section 3.3.2, using a polynomial spline for each axis separately. The reference foot position is computed by evaluating the spline curve at the current phase timestep, and is then used as an input for the whole-body controller for foot tracking. The goal of this WBC task is to reach the referenced foot target position after the foot flight time. The foot target task is prioritized only at the end of the foot flight phase. Therefore, the time-variant weight of the foot acceleration WBC task is selected lower $\alpha_{ft,min}$ at the beginning of the foot flight phase and higher $\alpha_{ft,max}$ at the end of the foot flight phase. For instance, the foot position in the second foot flight phase ($t \in [1.0, 1.4]s$) does not correspond to the referenced foot trajectory at the beginning of the foot flight phase. At the end of the foot flight phase, the foot position corresponds to the commanded position. Thus, the foot position tracking after stance is not prioritized to enable higher tracking performance for other WBC tasks. The movement of the left foot in negative x- and y-direction after stance, shows that the foot creates a moment counter-acting the body´s moment (illustrated in Figure 7.4 on the right), thus the torso orientation task is prioritized over the foot tracking task.

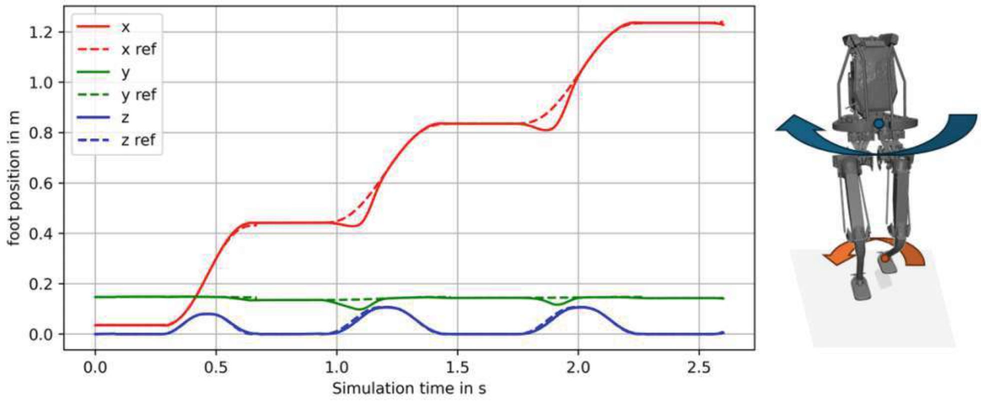


Figure 7.4: Foot position trajectories (continuous lines) with the reference foot position trajectories (dashed lines) for the left foot.

Another essential WBC task is the CoM acceleration task. The reference CoM acceleration is mapped to the desired leg forces $\boldsymbol{F}_{leg}$ (5.1) using the robot mass $m = 40kg$ and gravitational vector $\boldsymbol{g}$. The commanded desired leg forces are shown in Figure 7.5. The vertical component $F_{leg,z}$ shows strictly positive and rather high forces in comparison to the horizontal components, $F_{leg,x}$ and $F_{leg,y}$. These show repeating longitudinal forces (x-axis), due to a constant forward velocity. The lateral forces in y-axis show an alternating sign, depending on the current foot in stance, e.g. in the third stance phase ($1.05 < t < 1.38$) the right foot is in stance, therefore, the resulting lateral forces are positive. The commanded leg forces resemble measured human leg forces [3], thus the biologically inspired CoM trajectory planning successfully imitates a human running gait.

The desired CoM acceleration and reference foot trajectories are used as a WBC task for the whole-body controller.

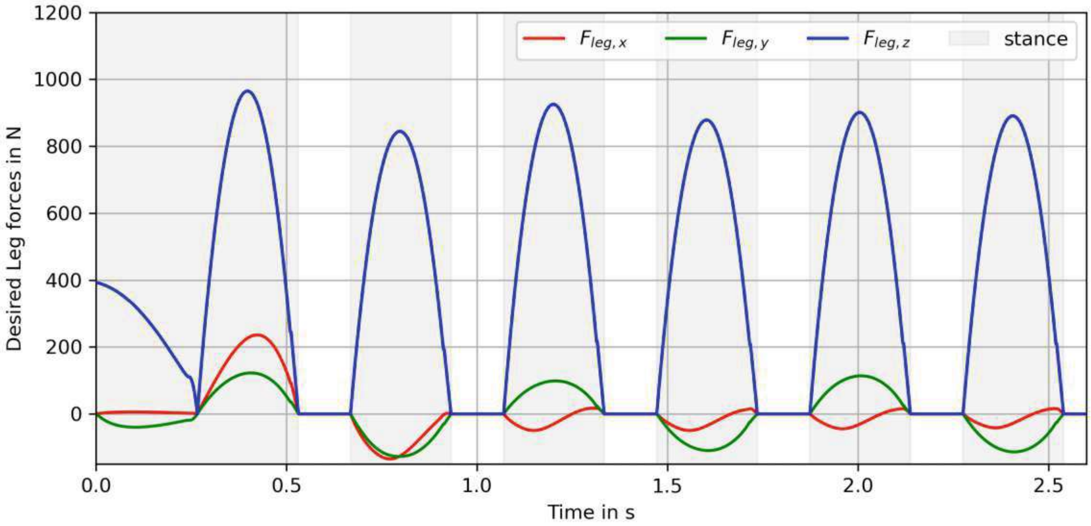


Figure 7.5: Commanded desired robot leg forces $\boldsymbol{F}_{leg}$ in Newton computed using (5.1) and the commanded desired CoM acceleration $\ddot{\boldsymbol{x}}_{des}$.

Figure 7.6 shows the robot joint torques, commanded from the whole-body controller. The torque limits of $\tau_{max} = 1000$Nm are included in the WBC optimization formulation (6.16) and are never exceeded. The linear joints, controlling the leg length for each foot, show the highest torque values in stance. As shown in Figure 7.5, the commanded vertical leg forces are the highest, thus the leg length joints controlling mostly the vertical position of the robot show much higher torque demands than other joints.

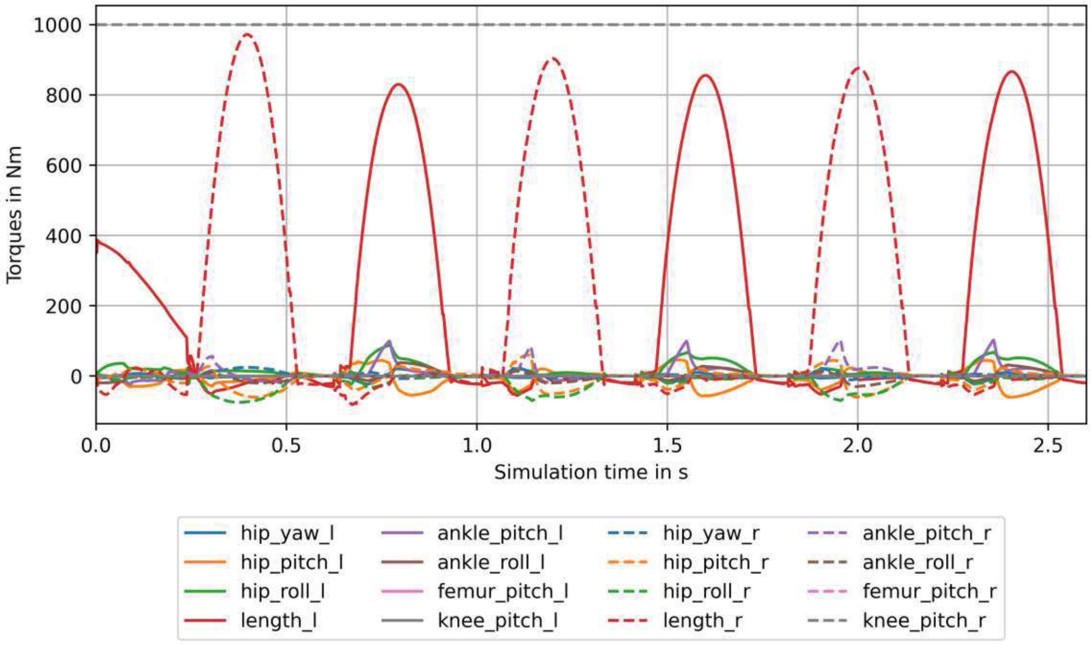
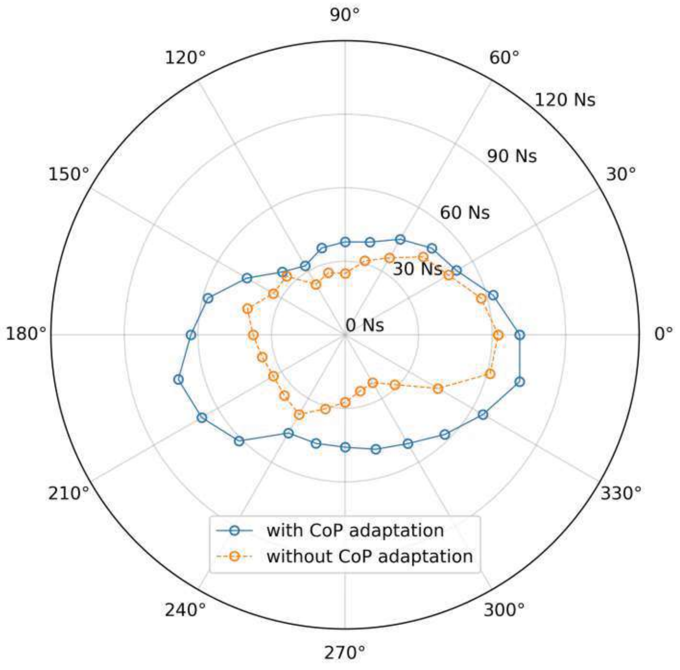


Figure 7.6: Commanded robot joint torques with the upper limit of $\tau_{max} = 1000$Nm. Left leg torques are continuous lines and right leg torques are dashed lines.

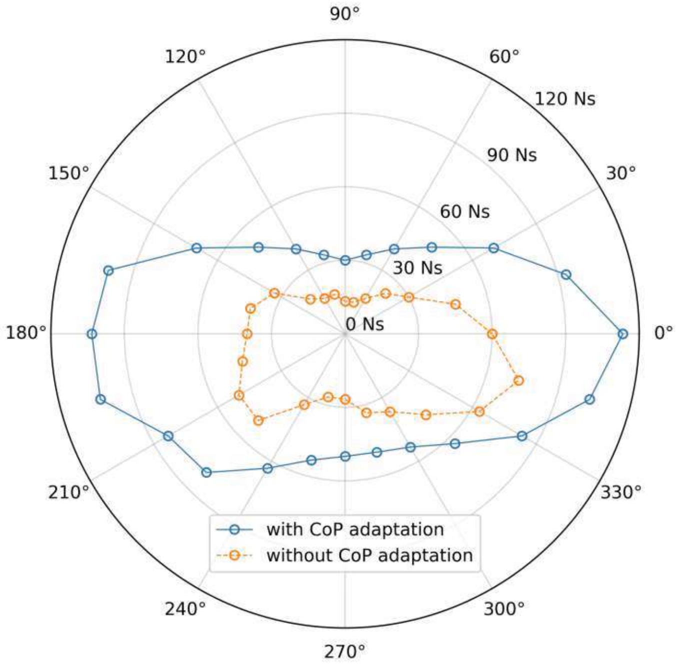
7.1 Push recovery analysis

In this experiment, the robustness of the controller is evaluated. The robot is pushed on the torso with a disturbance force F_{dist} for a duration of Δt from 24 different angles while running forward at $v_{ref} = 0.5\text{ms}^{-1}$. The push occurs in the middle of the stance phase of the left foot. The maximal recoverable disturbance impulse $J = F_{dist}\Delta t$ for each angle is recorded and visualized. The experiment settings are found in the appendix Section A.2.

The experiments were executed for two different push durations, a short push $\Delta t = 0.2$s and a longer push $\Delta t = 0.8$s, as well as with and without CoP adaptation. The experiment for a short push duration, in Figure 7.7a, illustrates a good push recovery for all angles especially in the longitudinal directions. The maximum recoverable disturbance impulse of $J = 74$Ns corresponds to a disturbance force of 370N for $\Delta t = 0.2$s, in forward running direction (345°) with CoP adaptation. The maximum disturbances that can be compensated for lateral pushes are slightly larger on the right side of the robot (lower half of the plot) because the stance foot at the moment of the push is the left foot.
The results without CoP adaptation show lower recoverable disturbance impulses due to less controllability of the robot, especially for pushes to the back (180°). Due to the short push duration, both strategies, step adaptation and CoP adaptation, are used to compensate the disturbance. On the other hand, a longer push duration forces the robot to compensate the disturbance also during stance using only CoP adaptation.



(a) $\Delta t = 0.2\text{s}$

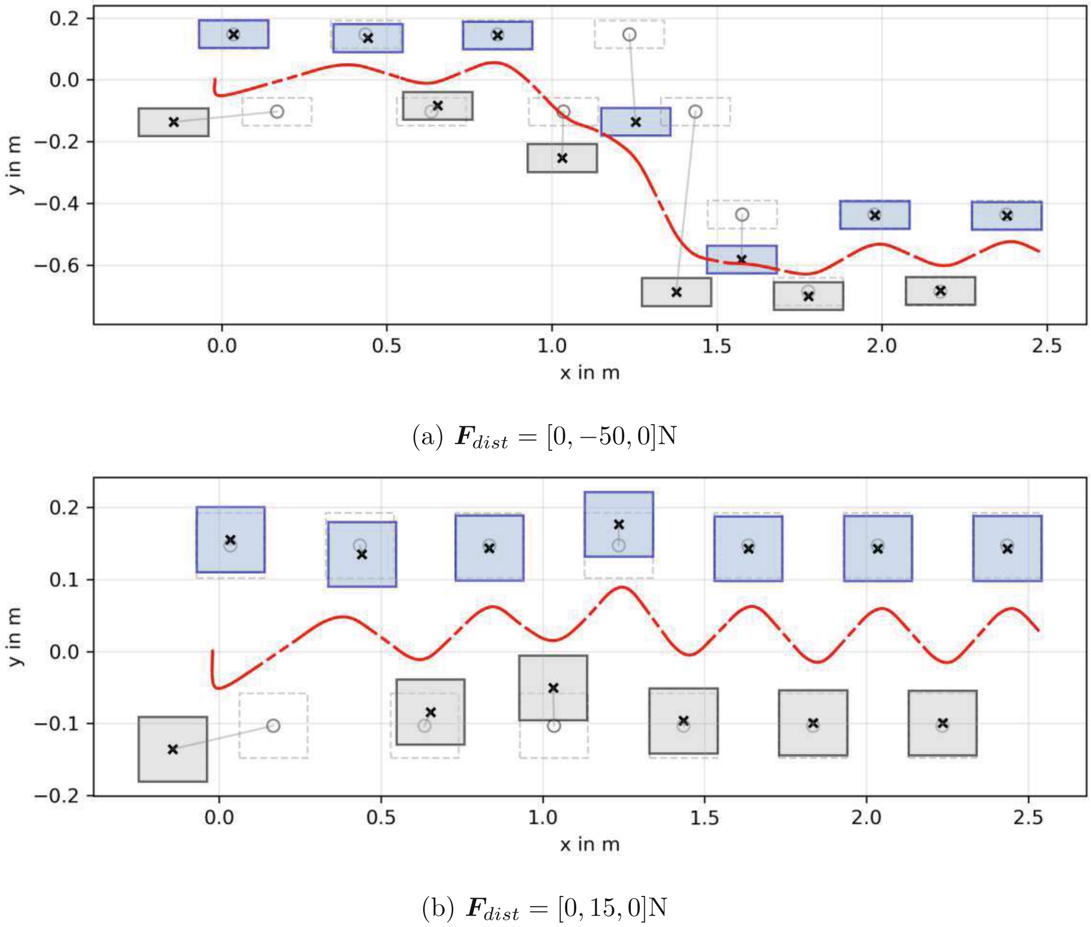


(b) $\Delta t = 0.8\text{s}$

Figure 7.7: Comparision of the maximal recoverable disturbance impulse for a push while running forward at $v_{ref} = 0.5ms^{-1}$. The push occurs in the middle of the left foot stance phase for push durations of a) $\Delta t = 0.2\text{s}$ and b) $\Delta t = 0.8\text{s}$.

Figure 7.7b shows the experiment results with a longer push duration of $\Delta t = 0.8$s. Here, the maximal disturbances without CoP adaptation differs slightly from the short push. However, the maximal recoverable disturbances with CoP adaptation increase significantly, especially in longitudinal pushing directions. The maximum recoverable disturbance impulse of $J = 113$Ns corresponds to a disturbance force of 141N for $\Delta t = 0.8$s, in forward running direction (0°) with CoP adaptation.

Two recovered pushes are visualized in Figures 7.8a and 7.8b for a push in lateral direction (y-axis) with a duration of $\Delta t = 0.8$s. First, a push with a force of $F_{dist,y} = -50$N (Figure 7.8a) is examined. Here, the running gait is stabilized using both strategies. For higher performance, the stepping stones are re-planned when a large footstep error is reached, enabling a faster convergence of stable running. The second experiment showcases a push with $F_{dist,y} = 15$N (Figure 7.8b). During the push, the difference between the target and desired footsteps is minimal, due to the CoP adaptation during stance, compressing the small disturbance force almost without step adaptation.

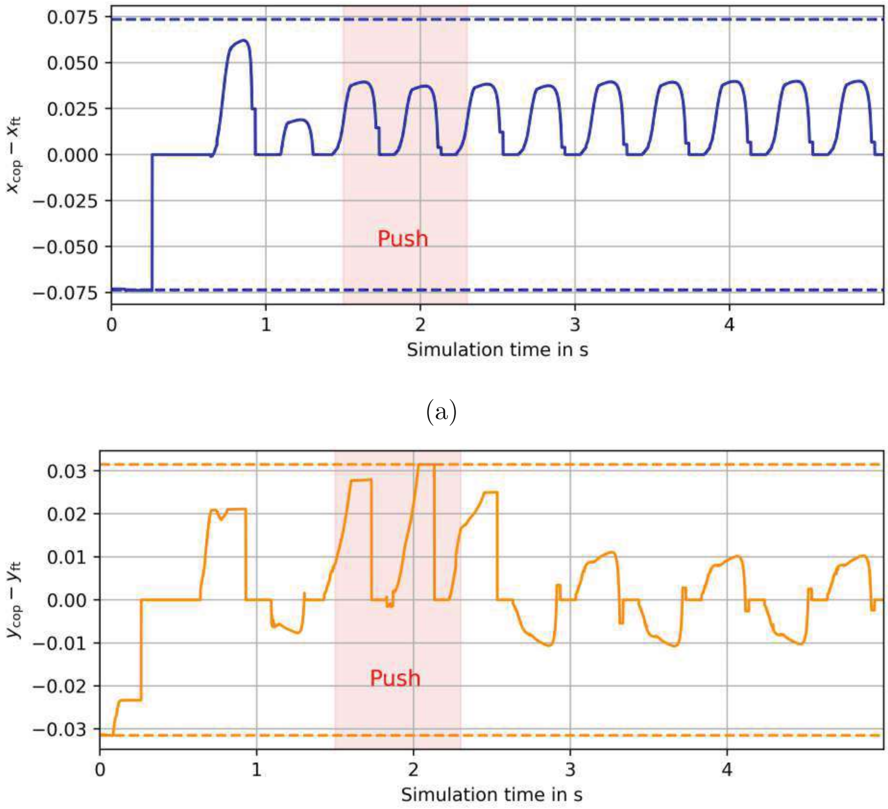


(a) $\boldsymbol{F}_{dist} = [0, -50, 0]$N

(b) $\boldsymbol{F}_{dist} = [0, 15, 0]$N

Figure 7.8: Two running gaits with an external push at $t = 1.5$s with different forces $\boldsymbol{F}_{dist}$ for $\Delta t = 0.8$s. The robot starts in a standstill with a commanded reference forward velocity of $v_{ref} = 0.5\text{ms}^{-1}$

Figures 7.9a and 7.9b decode the horizontal positions of the CoP in the stance foot over time, for the previously shown experiment in Figure 7.8b. The CoP can be shifted in the stance foot within the foot contact area, displayed with the upper and lower boundaries in the figures. At the start of the experiment, the robot is in a standstill, thus the CoP is shifted to the back (negative x-axis) to push the robot in front and gain velocity. The robot starts with the left foot in stance and falls to the right-hand side, therefore the CoP is also shifted in negative y-direction.
During stable running ($t > 2.5$s), the CoP is shifted in the same direction as the CoM approaches the foot to decelerate the robot. Due to forward running the x-component of the CoP is always shifted to the front, the y-component is shifted to the right for the right foot in stance and to the left for the left foot in stance.
The push occurs at $t = 1.5$s, here the CoP is significantly shifted in the pushing direction to compress the disturbance. The x-component does not show any changed behavior because the push is orthogonal to this direction. Therefore, the CoP adaptation compresses the small disturbance with minimal footstep adaptation.



(a)

(b)

Figure 7.9: Relative position of the CoP with origin in the footstep for the a) longitudinal and b) lateral components for a push at $t = 1.5$s for $\Delta t = 0.8$s with a magnitude of $F_{dist} = [0, 15, 0]$N.

7.2 Locomotion transitions

The evaluations of the optimized planner approach showed high robustness to disturbances and challenging initial conditions. This enables locomotion transitions using a different controller for other gaits. For instance, in this experiment, the walking locomotion controller from Egle et al. [26] is used for CoM velocities up to 0.4ms^{-1} and is then switched to the proposed running planner. The methods are switched at the beginning of the left foot stance phase. Figure 7.10 shows a time series of the humanoid robot Kangaroo during the transition from a standstill to a final running velocity of $v_x = 1.0\text{ms}^{-1}$.

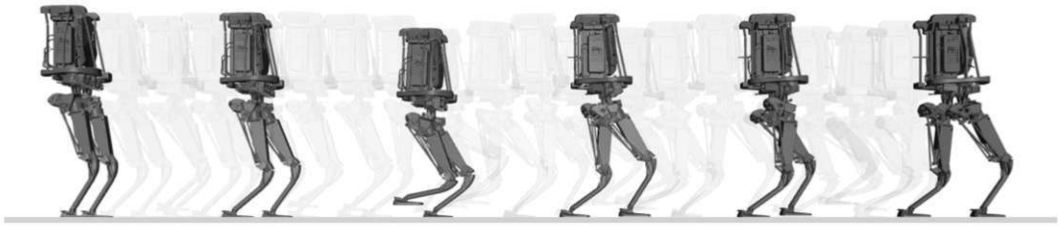


Figure 7.10: Time series of the humanoid robot Kangaroo [7] during a walking to running transition.

7.3 Conclusion

This chapter integrates and evaluates the proposed planning method on the robot Kangaroo in simulation with an inverse-dynamics whole-body controller from Chapter 6. First, a stable running gait is evaluated. The footstep pattern shows that the footstep adjustment distance converges to zero after three footsteps. Foot trajectories show, that the foot tracking is not following the reference after the foot leaves stance (lift-off), this is caused by a small time-variant foot task weight at the beginning of the foot flight phase, which enables the foot to create a counter-momentum to the torso momentum. At the end of the foot flight phase the foot tracking task is prioritized to guarantee that the swing foot reaches its target position. The measured robot torques show that the whole-body controller is successfully limiting the maximum torques and generating a feasible control input for the robot.

Next, the maximum recoverable disturbance impulses for pushes from various pushing angles are compared. The planner successfully compensates disturbances of up to $F_{dist} = 370\text{N}$ for $\Delta t = 0.2\text{s}$ in the forward running direction. Also pushes in lateral direction are recovered with slightly lower magnitudes. The influence of CoP adaptation during stance shows significant improvement in robustness, especially for longer pushes $\Delta t = 0.8\text{s}$. The planning method shows robustness against fast dynamic disturbances using step adaptation and static disturbances using CoP adaptation. Additionally, high robustness to challenging initial conditions (e.g. standstill) enables locomotion transitions using different controllers. For instance the DCM-walking controller [26] is used for a walking to running transition, showing a transition $v_x = 0.0\text{ms}^{-1}$ to $v_x = 1.0\text{ms}^{-1}$.

8 Discussion and conclusion

8.1 Discussion

The main goal of this thesis is to provide a planning method for humanoid running, based on the CoM dynamics. The work is based on [3, 5] which provide a biologically inspired CoM trajectory planning method that approximates human-like running ground-reaction force profiles. In [3] the planned trajectory is stabilized using the BID controller. Simulation results show fast running gaits on the humanoid robot Toro with robust footstep adaptation on the first preview footstep. The BID controller has performance limitations when handling challenging initial robot states, such as initiating a run from a standstill. Consequently, the robot's starting configuration must be adjusted to ensure a stable running gait.

We propose a different approach, where the CoM trajectory is re-planned at every timestep (online planning), considering the current CoM position. This approach is extended with optimal n-preview footstep adjustment and CoP adaptation in the stance foot, allowing for more robustness to disturbances and initial conditions. Additionally, kinematic limitations of the robot are considered and leg collisions are prevented (also in [3]) using convex footstep regions. To meet the convexity requirements of the quadratic program, the footstep regions cannot be defined for leg cross-over which would be favorable. A solution to this could be provided by solving multiple QPs with footstep regions that allow leg cross-over or by using a learned parameter to rotate the convex region around the current stance foot [27].
The CoP adaptation in the current stance foot, allows the robot to shift the CoP within the foot contact region, thus enhancing the robustness of the planner.

The proposed method is integrated into a WBC framework and evaluated in simulation with the humanoid robot Kangaroo [7]. The results show robust and stable running gaits with various initial conditions and external disturbances. The CoP adaptation enables high accuracy on stepping stones and can adjust for small static disturbances with minimal footstep adjustment. Even for large disturbances, CoP adaptation enables faster convergence of stable running. The maximum velocity reached for this controller reaches up to $v_x = 2.1\mathrm{ms}^{-1}$. Here, the robot diverges from a stable running motion, due to a significant torso momentum in the vertical z-axis. The foot tracking WBC task manages to counter-act this moment up to certain speeds. This problem could be solved using a different robot platform with arms to generate an additional counter-momentum, as used in [3].
Overall, the method features a robust humanoid running algorithm showing stable running even when exposed to unknown disturbances or challenging initial robot states.

8.2 Conclusion

In Chapter 2 and 3 the biologically inspired CoM trajectory planning algorithm is presented and adapted. The planning algorithm is separated into vertical and horizontal trajectory planning, using 4-th and 5-th order polynomial splines, respectively. Both trajectories fulfill a continuous transition between the stance and the flight phase for several preview phases. The vertical CoM trajectory is solved locally, for each preview phase separately, and aims to reach a desired touch-down height after each flight phase. The horizontal planning part aims to find a CoM trajectory according to a sequence of n desired footsteps, the final CoM take-off velocity, and a minimized vertical distance of the leg force intersection point with the ground. The combination of both planning approaches enables an online planning structure with a finite horizon (n preview footsteps). The CoM trajectory is re-planned after each timestep using the current CoM state, thus, perturbations on the CoM are considered in planning, increasing the planner's robustness.
The limitations of the purely analytical planning method are within its design parameters. Sub-optimally placed desired foot positions and fixed phase timing can cause the running locomotion to diverge, thus the robot falls. Especially during challenging robot states, such as running from a standstill or with external perturbations, the method does not converge to a stable running motion.

Therefore, we extend the purely analytical planning method with optimized design parameters via a quadratic program (QP), in Chapter 4, to enable robust running even during challenging robot states or with high external perturbations acting on the robot. The horizontal planning part is extended with optimal foot target placement and optimal CoP adaptation in the stance foot. The solution for the take-off states from (3.26) is used as an equality constraint, to ensure a stable running CoM trajectory considering the optimized foot target positions. The placement of the foot target positions is restricted within feasible footstep regions $\boldsymbol{\Omega}$, which guarantees kinematic feasibility and prevents leg collisions while running. The CoP in the stance foot is adapted within the foot support polygon, enabling higher robustness of the planner.

The proposed method is evaluated in simulation using a point-mass model, assuming ideal CoM and foot position tracking. The results, in Chapter 5, show stable running gaits with commanded leg forces replicating human-like ground-reaction force profiles from experiments [3]. The optimized planning approach can also manage challenging initial robot configurations, i.e. initial CoM velocities. Here, the CoP adaptation in the stance foot shows faster convergence of the foot adjustment distance, compared to a fixed CoP in the stance foot. This highlights the advantages of this method, where two strategies are used to compensate for disturbances, i.e. step location adaptation and CoP adaptation.

In Chapter 7, the proposed method is integrated on the humanoid robot Kangaroo in the simulation environment MuJoCo. The robot is controlled using the inverse-dynamics whole-body controller from Chapter 6, which finds the optimal control input using a set of WBC tasks via QP optimization.
The simulation results show stable running gaits with feasible robot torques. Additionally, a set of robot pushes is examined, showing promising results for different push angles and

durations, especially in the forward running direction with active CoP adaptation.
The proposed method enables robust running for humanoid robots with n-preview footstep adjustment and CoP adaptation. Additionally, high robustness to challenging initial conditions (e.g. standstill) enables locomotion transitions using different controllers, for instance walking to running transitions using a DCM-based walking controller.

Given the progress made and the challenges identified, this thesis provides a substantial foundation for future enhancements and improvements:

- Firstly, as mentioned in the discussion, leg cross-over is prevented by convex-shaped footstep regions. Multiple quadratic programs (QP) could be solved for different footstep regions allowing for leg cross-over. A different approach could use a learned parameter to rotate the convex region around the current stance foot [27]. This improvement could enable human-like dynamic running especially when exposed to perturbations.

- Phase timing in horizontal planning is not considered for optimization. This would result in a more general non-linear optimization problem which might be challenging to solve in real-time. This limits the adaptability of the method due to empirically tuned timing parameters, that are not adapted to specific situations, e.g. for a disturbance.

- The planning method is integrated on the robot Kangaroo in Simulation which offers no arms. Thus, a significant torso moment is observed at higher running speeds limiting the maximum running speed. A robot with arms could be used to create a counter-momentum and stabilize the robot.

- So far, the method was only validated in simulation. Humanoid running on a real-life robot platform is a challenging experiment, due to dynamic movement and high force demands. However, the high robustness of the planning method could allow for an integration in real-life.

These advancements could considerably enhance the running planner's functionality and allow for its integration into practical, real-world scenarios.

A System Parameters

A.1 Point-Mass Simulations

The parameters shown in Table A.1 and A.2 are used for the point-mass model experiments from Chapter 5.

Parameter	Value	Description
f_s	1000 Hz	Simulation frequency
f_p	1000 Hz	Planner frequency
f_{step}	2.5 Hz	Step frequency
c_{stance}	0.66	Stance percentage
n	5	Number of preview phases
$z_{TD,des}$	0.83m	Desired touch-down height
$\alpha_{ft,i}$	$1e4$	QP weight for the footstep error ($\forall i$)
$\alpha_{\chi_p,n}$	$1e3$	QP weight for the final CoM take-off velocity
$\boldsymbol{v}_{ref}$	$\begin{bmatrix}0.5 & 0 & 0\end{bmatrix}^T$ ms^{-1}	Reference CoM velocity
$\dot{\boldsymbol{\chi}}_{TO,n,des}$	$\begin{bmatrix}0 & 0\end{bmatrix}^T$ ms^{-1}	Desired final CoM take-off velocity
$\ddot{\boldsymbol{\chi}}_{TO,n,des}$	$\begin{bmatrix}0 & 0\end{bmatrix}^T$ ms^{-2}	Desired final CoM take-off acceleration

Table A.1: Default planner and simulation environment parameters.

Parameter	Value	Description
m	40kg	Robot mass
Δx_{ft}	0.21m	Foot length
Δy_{ft}	0.09m	Foot width
W_{nom}	0.25m	Nominal step width
W_{min}	0.12m	Minimal step width
W_{max}	0.55m	Maximal step width
L_{max}	0.8m	Maximal step length
$\boldsymbol{x}_{ini}$	$\begin{bmatrix}0 & 0 & 0.85\end{bmatrix}^T$ ms^{-1}	Initial CoM position
$\boldsymbol{v}_{ini}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ ms^{-1}	Initial CoM velocity
$\boldsymbol{a}_{ini}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ ms^{-1}	Initial CoM acceleration
$\boldsymbol{x}_{ft,right}$	$\begin{bmatrix}0 & -0.1 & 0\end{bmatrix}^T$ ms^{-1}	Initial right foot position
$\boldsymbol{x}_{ft,left}$	$\begin{bmatrix}0 & 0.1 & 0\end{bmatrix}^T$ ms^{-1}	Initial left foot position

Table A.2: Default point-mass model parameters.

A.2 WBC Simulations

The parameters shown in Table A.3 and A.4 are used for the experiments from Chapter 7.

Parameter	Value	Description
f_s	1000 Hz	Simulation frequency
f_p	1000 Hz	Planner frequency
f_{step}	2 Hz	Step frequency
c_{stance}	0.62	Stance percentage
n	5	Number of preview phases
$z_{TD,des}$	0.83m	Desired touch-down height
$\alpha_{ft,i}$	1e4	QP weight for the footstep error ($i > 1$)
$\alpha_{ft,1}$	1e4/0	QP weight for the first footstep error (flight/stance)
$\alpha_{\chi_p,n}$	1e3	QP weight for the final CoM take-off velocity
$\boldsymbol{v}_{ref}$	$\begin{bmatrix}0.5 & 0 & 0\end{bmatrix}^T$ ms^{-1}	Reference CoM velocity
$\dot{\boldsymbol{\chi}}_{TO,n,des}$	$\begin{bmatrix}0.2 & 0\end{bmatrix}^T$ ms^{-1}	Desired final CoM take-off velocity
$\ddot{\boldsymbol{\chi}}_{TO,n,des}$	$\begin{bmatrix}0 & 0\end{bmatrix}^T$ ms^{-2}	Desired final CoM take-off acceleration

Table A.3: Default planner and simulation environment parameters.

Parameter	Value	Description
m	40kg	Robot mass
Δx_{ft}	0.21m	Foot length
Δy_{ft}	0.09m	Foot width
W_{nom}	0.25m	Nominal step width
W_{min}	0.12m	Minimal step width
W_{max}	0.55m	Maximal step width
L_{max}	0.8m	Maximal step length
$\boldsymbol{v}_{ini}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ ms^{-1}	Initial CoM velocity
$\boldsymbol{a}_{ini}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ ms^{-1}	Initial CoM acceleration

Table A.4: Default parameters used for the humanoid robot Kangaroo in the simulation environment MuJoCo. The position of the robot and its feet is set by MuJoCo.

Task	Parameter	Value
CoM acceleration	$\boldsymbol{w}$	$\begin{bmatrix}100 & 100 & 100\end{bmatrix}$
Torso orientation	w	220
	$\boldsymbol{k}_p$	$\begin{bmatrix}500 & 800 & 200\end{bmatrix}$
	$\boldsymbol{k}_d$	$\begin{bmatrix}25 & 25 & 25\end{bmatrix}$
Foot stance	w	1000
Foot acceleration	w	400
	$\boldsymbol{k}_p$	$\begin{bmatrix}2000 & 2000 & 4000\end{bmatrix}$
	$\boldsymbol{k}_d$	$\begin{bmatrix}70 & 70 & 150\end{bmatrix}$
Foot orientation	w	150
	$\boldsymbol{k}_p$	$\begin{bmatrix}500 & 500 & 1500\end{bmatrix}$
	$\boldsymbol{k}_d$	$\begin{bmatrix}80 & 80 & 200\end{bmatrix}$
Closed linkage	w	1000

Table A.5: Default parameters of the whole-body controller.

Bibliography

[1] A. Grau, M. Indri, L. L. Bello, and T. Sauter, "Industrial robotics in factory automation: From the early stage to the Internet of Things," in *IECON*, Oct. 2017, pp. 6159–6164.

[2] Y. Rivero-Moreno *et al.*, "Robotic Surgery: A Comprehensive Review of the Literature and Current Trends," *Cureus*, vol. 15, no. 7, Jul. 2023.

[3] J. Englsberger, P. Kozłowski, C. Ott, and A. Albu-Schäffer, "Biologically Inspired Deadbeat Control for Running: From Human Analysis to Humanoid Control and Back," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 854–867, Aug. 2016.

[4] T. Egle, "Development of a Planning Approach for the Transition between Robotic Walking and Running," M.S. thesis.

[5] T. Egle, J. Englsberger, and C. Ott, "Analytical Center of Mass Trajectory Generation for Humanoid Walking and Running with Continuous Gait Transitions," in *IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, Nov. 2022, pp. 630–637.

[6] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[7] A. Roig, S. Kishor Kothakota, N. Miguel, P. Fernbach, E. Mingo Hoffman, and L. Marchionni, "On the Hardware Design and Control Architecture of the Humanoid Robot Kangaroo," in *6th Workshop on Legged Robots during the International Conference on Robotics and Automation*, Philadelphie, PA, May 2022.

[8] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-Real Learning of All Common Bipedal Gaits via Periodic Reward Composition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021.

[9] D. Crowley, J. Dao, H. Duan, K. Green, J. Hurst, and A. Fern, "Optimizing Bipedal Locomotion for The 100m Dash With Comparison to Human Running," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2023.

[10] T. Sugihara, Y. Nakamura, and H. Inoue, "Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control," in *Proceedings IEEE International Conference on Robotics and Automation*, May 2002.

[11] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time zmp gait generation and feedback stabilization," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015.

[12] T. Takenaka, T. Matsumoto, and T. Yoshiike, “Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1084–1091.

[13] J. Englsberger, G. Mesesan, and C. Ott, “Smooth trajectory generation and push-recovery based on Divergent Component of Motion,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 4560–4567.

[14] G. Schultz and K. Mombaur, “Modeling and Optimal Control of Human-Like Running,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 783–792, Oct. 2010.

[15] H. Geyer, A. Seyfarth, and R. Blickhan, “Compliant leg behaviour explains basic dynamics of walking and running,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[16] B. Dadashzadeh, H. R. Vejdani, and J. Hurst, “From template to anchor: A novel control strategy for spring-mass running of bipedal robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2566–2571.

[17] A. Wu and H. Geyer, “The 3-d spring–mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1114–1124, 2013.

[18] S. Sovukluk, J. Englsberger, and C. Ott, “Highly Maneuverable Humanoid Running via 3D SLIP+Foot Dynamics,” *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1131–1138, Feb. 2024.

[19] G. Mesesan, R. Schuller, J. Englsberger, C. Ott, and A. Albu-Schäffer, “Unified Motion Planner for Walking, Running, and Jumping Using the Three-Dimensional Divergent Component of Motion,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4443–4463, Dec. 2023, Conference Name: IEEE Transactions on Robotics.

[20] B. Benda, P. Riley, and D. Krebs, “Biomechanical relationship between center of gravity and center of pressure during standing,” *IEEE Transactions on Rehabilitation Engineering*, vol. 2, no. 1, pp. 3–10, Mar. 1994.

[21] A. G. Pandala, Y. Ding, and H.-W. Park, “Qpswift: A real-time sparse quadratic program solver for robotic applications,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3355–3362, 2019.

[22] J. Englsberger, “Combining reduced dynamics models and whole-body control for agile humanoid locomotion,” Ph.D. dissertation, Dec. 2016.

[23] S. Sovukluk, J. Englsberger, and C. Ott, “Whole Body Control Formulation for Humanoid Robots with Closed/Parallel Kinematic Chains: Kangaroo Case Study,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 10 390–10 396.

[24] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC Press, Sep. 2017, ISBN: 978-1-315-13637-0.

[25] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: An operator splitting solver for quadratic programs,” *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[26] T. Egle, J. Englsberger, and C. Ott, “Step and Timing Adaptation during Online DCM Trajectory Generation for Robust Humanoid Walking with Double Support Phases,” in *IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, Austin, TX, Dec. 2023, pp. 1–8.

[27] T. Egle, Y. Yan, D. Lee, and C. Ott, “Enhancing Model-Based Step Adaptation for Push Recovery through Reinforcement Learning of Step Timing and Region,” *Conference Paper accepted at HUMANOIDS 2024*, Oct. 2024.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct - Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im September 2024

David Demattio