

MASTER'S THESIS

DEVELOPMENT OF A PLANNING APPROACH FOR THE TRANSITION BETWEEN ROBOTIC WALKING AND RUNNING

Freigabe:

Der Bearbeiter:

Unterschriften

Tobias Egle



Betreuer:

Johannes Engelsberger

Der Institutsdirektor

Prof. Alin Albu-Schäffer

Dieser Bericht enthält 81 Seiten, 23 Abbildungen und 8 Tabellen

Tobias Egle (03665705)

Development of a Planning Approach for the Transition between Robotic Walking and Running

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science (M.Sc.)
at the Technical University of Munich Department of Mechanical Engineering

Supervisor: Prof. Dr.-Ing. Alin Albu-Schäffer
Department of Informatics
Chair of Sensor Based Robotic Systems and Intelligent Assistance Systems

Advisor: Dr.-Ing. Johannes Engelsberger
German Aerospace Center
Institute of Robotics and Mechatronics

Submission date: 31 August 2021

Declaration of Authorship

I hereby confirm that this Master's thesis is my own work and I have documented all sources and material used.

31 August 2021 in Munich

A handwritten signature in black ink, appearing to be 'T. Egle', written over a horizontal line.

(Tobias Egle)

Abstract

Bipedal legged locomotion promises improved accessibility and navigation of complex and non-barrier-free terrain compared to wheel-based mobility. The tradeoff for increased versatility, especially for humanoid robots, is slow locomotion. Based on existing methods for walking and running, this work presents a new approach that allows for smooth transitions between the two gaits, and thus faster locomotion. Due to the distinct biomechanics of the two gaits, different mathematical models and control approaches are used. The existing trajectory generation algorithms for walking and running developed at the German Aerospace Center were modified and integrated into a combined trajectory generation framework that generates continuous trajectories for multiple walking and running sequences and their transitions. Due to the coupling of different state variables between walking and running, the complete trajectory in the horizontal direction was generated in a single matrix calculation. By resolving the coupling in the vertical direction, the vertical trajectory was computed in a forward recursion from the first to the last gait sequence. The control strategies are unified by integrating the proposed trajectory generation into an inverse dynamics based whole-body controller. The presented approaches are validated in simulations with the humanoid robot Toro.

Contents

Abstract	v
1 Introduction	1
1.1 Contribution	1
1.2 Content outline	2
2 Related Work	3
3 Walking Trajectory Generation	5
3.1 Divergent Component of Motion and encoding of forces on the CoM . . .	5
3.2 Generation of consistent multi-step VRP, DCM and CoM trajectories . .	7
3.2.1 Consistent VRP, DCM and CoM interpolation in single transition phase	7
3.2.2 Computation of multi-step preview matrices	9
4 Running Trajectory Generation	13
4.1 Center of mass dynamics in the flight phase	14
4.2 Vertical planning and boundary conditions	15
4.3 Horizontal planning and boundary conditions	16
5 Continuous Gait Transitions between Walking and Running	21
5.1 Vertical planning	22
5.1.1 Running boundary conditions	23
5.1.2 Walking boundary conditions	24
5.1.3 Vertical planning for multiple walking and running sequences . . .	24
5.2 Horizontal planning	26
5.2.1 Running matrix assembly	28
5.2.2 Walking matrix assembly	33
5.2.3 Global matrix equation assembly	36
5.2.4 Structure of the global constraint and target mapping matrix . . .	39

5.2.5	Ensuring continuity for the transition between standing and walking	40
5.2.6	Postprocessing	42
5.3	Divergent Component of Motion tracking control	42
6	Evaluation and Visualization of the Presented Methods	45
6.1	Foot trajectories	50
6.2	Walking and running over stepping stones	51
7	Whole-Body Control	53
7.1	Dynamic model	53
7.2	Contact constraints on the robot	54
7.3	Derivation of tasks for the whole-body controller	56
7.3.1	Centroidal momentum task	56
7.3.2	Foot acceleration task	57
7.3.3	Further tasks and task summary	57
7.4	Optimization via a quadratic program	58
8	Whole-Body Simulation	59
9	Conclusion and Future Work	65
	List of Figures	68
	List of Tables	69
	References	73

Chapter 1

Introduction

Legged locomotion in robotics continues to gain attention in both the research community and in the general public. Videos of robots dancing or doing parkour from Boston Dynamics and news of Tesla’s intention to build a humanoid robot reach millions of people within days. However, even with great progress in recent years, today’s robots just barely reach a fraction of human locomotion capabilities to operate safely and reliably in human environments. What seems natural for humans is regarded as a very difficult problem for robots. Although, the improved versatility of legged locomotion in complex terrain, especially bipedal locomotion, attracts with great advantages compared to wheel-based mobility.

Bipedal robots can more easily overcome barriers (such as stairs), narrow passages, tight corners, and stepping stones, to name a few examples. However, this versatility comes with the drawback that today’s humanoid robots can generally only move very slowly. This is the substantial advantage of wheel-based mobility and also the reason why humans use bicycles, buses and trains. In contrast to most of today’s robots, humans can change their gait pattern from walking to running to significantly increase their locomotion speed.

There are specialized robots based on the rimless wheel model that can handle running very well [3]. In contrast, running is still a challenging problem for humanoid robots designed for versatility. This is mainly due to the very short contacts during running, which are potentially underactuated, and the high torque demands on the robot’s joints. Furthermore, the biomechanics of the basic human gaits, walking and running, differ significantly. In walking, the body’s center of mass height reaches a maximum in the middle of the stance phase, while the opposite is true for running [17]. To reproduce this center of mass motion for a humanoid robot, most often different mathematical models and control strategies for the two gaits are adopted.

1.1 Contribution

This thesis aims at developing a planning approach that enables a smooth transition between the two gaits and the different mathematical models behind them. The algorithms for generating walking [9, 11, 19] and running [7, 10] trajectories developed at the Institute of Mechatronics and Robotics at the German Aerospace Center serve as the basis for this work. A subsequent goal is to generate a full-body walking and run-

ning motion and its transition on a humanoid robot in simulation by implementing the developed planning approach in a whole-body controller [5].

1.2 Content outline

The work is organized as follows. Chapter 2 summarizes related work of walking and running trajectory generation in the field of robotics and references approaches for the transition between walking and running. In Chapter 3 an overview of the walking trajectory generation algorithm is presented. Chapter 4 introduces the trajectory generation utilized for running. The two algorithms for walking and running from the previous Chapters are modified and integrated into a combined trajectory generation framework that generates continuous trajectories for multiple walking and running sequences and their transitions. The trajectory generation is divided into horizontal and vertical directions and is derived in Chapter 5. Subsequently, the developed approach is evaluated in Chapter 6 and additional trajectories for the feet are generated for visualization purposes and integration in the whole-body controller. The whole-body control framework is derived in Chapter 7. The integration of the proposed trajectory generation method in the whole-body controller is evaluated in simulation with the humanoid Robot Toro (see Fig. 1.1) in Chapter 8. Chapter 9 concludes the work.

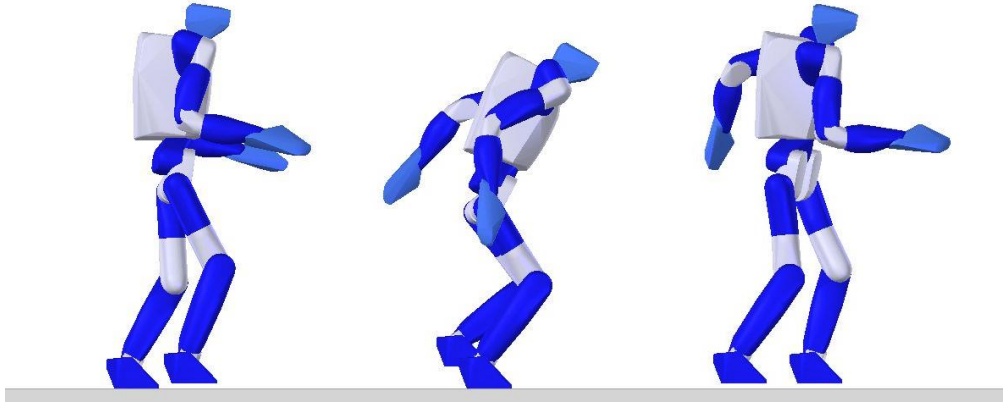


Figure 1.1: Simulation of the humanoid robot Toro [8] during the walk-to-run (W2R) transition (time series with 0.2s interval). The robot is displayed in the walking sequence (left), the transition stance phase (center) and in the flight phase during running (right).

Chapter 2

Related Work

Most scientific work in the field of robotic bipedal locomotion follows the idea to focus on the robot’s center of mass (CoM) dynamics for generating gait trajectories. The center of mass motion captures the most important aspects of a specific gait without having to consider the complexity of the robot’s nonlinear general dynamics. However, due to unilateral contact constraints, not every CoM trajectory is feasible. More specifically, the robot can only push on its environment with its feet and is therefore not able to accelerate towards the ground faster than gravity. Furthermore, without considering rotational inertia, all external forces have to pass through the robot’s support polygon (e.g. the convex hull of the stance feet) and the CoM to be physically feasible. To fulfill these requirements, a variety of models exist for different gaits.

One of the most popular models for walking is the Linear Inverted Pendulum (LIP) model [26, 28]. In this approach, a focus point of all external forces is designed to be equivalent to the torque-free base joint of the LIP model. The external force is then only dependent on the position of the CoM and the LIP base joint on the ground and always passes through these two points.

Other research suggests introducing an additional state variable and split the second-order CoM dynamics into a stable and an unstable part. Takenaka et al. [27] introduce the divergent component of motion (DCM) which is extended to 3D by Engelsberger et al. [11] as a generalization of the capture point (or 2D DCM) [6, 22]. Using the DCM, the general second-order CoM dynamics can be separated into two decoupled first-order dynamics. The DCM is defined as a point a certain distance in front of the CoM and is attracting it. Similar to the force-to-point transformation in the LIP model, a linear repelling force law uses a point, the so-called *enhanced Centroidal Moment Pivot* point (eCMP) to encode the direction and magnitude of the external forces. Starting from a eCMP trajectory that stays in the robot’s support polygon, the CoM trajectory can be calculated to always be physically feasible. The walking trajectory generation algorithm [11] provides a smooth CoM reference trajectory with continuous transitions from standing to walking and back to standing.

Furthermore, the second-order control problem turns into a first-order one, as only the unstable DCM dynamics needs to be stabilized, while the naturally stable CoM dynamics remains unaffected. By embedding the DCM-based walking controller into a whole-body control framework [5], stable full-body walking behavior can be shown in simulation and experiment on the humanoid robot Toro [8, 14].

The generation of running motions is most commonly based on the Spring-Loaded Inverted Pendulum (SLIP) model, which consists of a point mass on top of a massless, compliant leg [24]. Previous works show that the SLIP model is also suitable to describe the CoM dynamics of walking, where compression of the leg in the stance phase can also be observed, although not as pronounced as in running [1, 12]. With the right parameters and initial conditions, the SLIP model can be shown to be open-loop stable [12] or can be controlled to return to a stable limit cycle [4, 29]. However, there are no closed-form solutions available for the SLIP model. This motivates the approach by Engelsberger et al. [10], which aims to use polynomial splines to design CoM trajectories that produce approximately natural ground reaction forces by fulfilling a set of boundary conditions. The CoM trajectories are stabilized by a deadbeat controller, the so-called Biologically Inspired Deadbeat (BID) controller, that can also be implemented in a whole-body control framework and produces stable full-body running motion in simulation [10].

In the field of robotic bipedal locomotion, there is not much research on the transition between walking and running. The existing literature primarily focuses on the SLIP model. Rummel et al. [23] find an overlap between stable limit cycles of walking and running, showing that the same locomotion speed can be achieved with both gaits. Martinez and Carbajal [18] use a new gait called 'hopping' to connect the walking and running limit cycles and Shahbazi et al. [25] utilize a SLIP model with adjustable compliance to realize transitions between walking and running.

Chapter 3

Walking Trajectory Generation

The walking algorithm presented in this chapter was first introduced by Englsberger et al. [9, 11] and is based on the concept of separating the general second-order center of mass (CoM) dynamics into a stable and an unstable part. This is achieved by introducing an additional state variable, called the divergent component of motion (DCM).

3.1 Divergent Component of Motion and encoding of forces on the CoM

The DCM $\boldsymbol{\xi} \in \mathbb{R}^3$ is defined as

$$\boldsymbol{\xi} = \mathbf{x} + b\dot{\mathbf{x}}, \quad (3.1)$$

where $\mathbf{x} \in \mathbb{R}^3$ and $\dot{\mathbf{x}} \in \mathbb{R}^3$ are the CoM position and velocity, respectively, and b is the time constant of the DCM dynamics. Reordering (3.1) yields the first-order CoM dynamics:

$$\dot{\mathbf{x}} = -\frac{1}{b}(\mathbf{x} - \boldsymbol{\xi}). \quad (3.2)$$

This equation shows that the CoM dynamics is stable for $b > 0$ and the CoM naturally follows the DCM. The DCM dynamics is found by differentiating (3.1) and inserting (3.2) and Newton's 2nd law $\ddot{\mathbf{x}} = \mathbf{F}/m$:

$$\dot{\boldsymbol{\xi}} = -\frac{1}{b}\mathbf{x} + \frac{1}{b}\boldsymbol{\xi} + \frac{b}{m}\mathbf{F}_{com}. \quad (3.3)$$

Here, m is the robot's total mass and $\mathbf{F}_{com} = \mathbf{F}_g + \mathbf{F}_{ext}$ is the total force acting on the CoM, which consists of the gravitational force \mathbf{F}_g and the external force \mathbf{F}_{ext} . Due to the focus on locomotion, the external force is hereafter referred to as leg force \mathbf{F}_{leg} . Without rotational inertia, the leg force is only physically feasible if it passes through both the robot's support polygon, e.g. the convex hull of the stance feet, and the CoM. The intersection point of the leg forces line of action with the foot plane is denoted center of pressure (CoP). If rotational inertia is considered, the line of action \mathbf{l}_{act} of the external force can temporarily leave the support polygon, i.e. when it is shifted by a torque around the CoM. To fulfill the leg force feasibility constraint, a linear repelling force law is constructed that uses a point, the so-called *enhanced Centroidal Moment Pivot* point

(eCMP) \mathbf{r}_{ecmp} , to encode the direction and magnitude of the leg forces relative to the CoM:

$$\mathbf{F}_{leg} = \frac{m}{b^2} (\mathbf{x} - \mathbf{r}_{ecmp}). \quad (3.4)$$

Consequently, the eCMP is not constrained to the ground while the centroidal moment pivot (CMP) is located at the intersection of the line eCMP-to-CoM with the ground and acts as an effective CoP in the presence of a torque caused by changes in the robot's angular momentum. Conversely, the CMP and CoP coincide when the angular momentum of the robot does not change. Fig. 3.1 illustrates the relationship between the eCMP, the CMP and the CoP. With the specific definition of the leg force (3.4), the

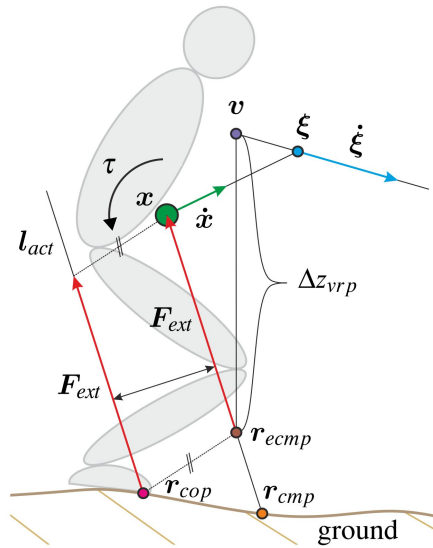


Figure 3.1: Point correlations for general robot dynamics (adapted from [9]).

DCM dynamics (3.3) can be written in a form that is independent of the CoM, which corresponds to a decoupling. To further simplify, the so-called *Virtual Repellent Point* (VRP) \mathbf{v} is introduced as

$$\mathbf{v} = \mathbf{r}_{ecmp} + \begin{bmatrix} 0 & 0 & b^2 g \end{bmatrix}^T = \mathbf{r}_{ecmp} + \begin{bmatrix} 0 & 0 & \Delta z_{vrp} \end{bmatrix}^T, \quad (3.5)$$

which encodes the total force on the CoM in another linear repelling force law, based on the difference of the CoM and the VRP:

$$\mathbf{F}_{com} = \frac{m}{b^2} (\mathbf{x} - \mathbf{v}). \quad (3.6)$$

Inserting (3.6) into (3.3), the DCM dynamics can be simplified to

$$\dot{\xi} = \frac{1}{b} (\xi - v) . \quad (3.7)$$

This equation shows that the DCM dynamics is unstable for $b = \sqrt{\Delta z_{vrp}/g} > 0$ and the DCM, as the name implies, diverges away from the VRP.

3.2 Generation of consistent multi-step VRP, DCM and CoM trajectories

First, a footstep plan is generated that is compatible with the kinematics of the robot. To satisfy the feasibility constraint on the leg forces, N eCMP waypoints are chosen that lie within the planned footstep plane. Consequently, the corresponding leg forces always pass through both the CoM and the convex hull of the robot's stance foot/feet. To find the corresponding VRP waypoints, a desired eCMP-to-VRP height difference Δz_{vrp} is added to the z-component of the eCMP as in (3.5):

$$\mathbf{v}_{wp,\varphi} = \mathbf{r}_{ecmp,wp,\varphi} + [0 \quad 0 \quad \Delta z_{vrp}]^T. \quad (3.8)$$

The walking sequence is then split into $N_\varphi = N-1$ transition phases $\varphi \in \{1, \dots, N_\varphi\}$. During each transition phase, the VRP trajectory is generated by interpolation between the φ -th VRP waypoint $\mathbf{v}_{wp,\varphi}$ and the $(\varphi+1)$ -th VRP waypoint $\mathbf{v}_{wp,\varphi+1}$. Desired DCM and CoM waypoints can be derived from the VRP waypoints by solving the DCM dynamics (3.7) and the CoM dynamics (3.2), respectively. The next section describes the interpolation scheme for a single transition phase.

3.2.1 Consistent VRP, DCM and CoM interpolation in single transition phase

In general, for each transition phase φ , low-order polynomial splines are used as interpolation functions $f_\varphi(t_\varphi)$ between a VRP start point $\mathbf{v}_{0,\varphi}$ and a VRP end point $\mathbf{v}_{T,\varphi}$. The smoothness of the resulting trajectories such as DCM and CoM depends on the polynomial order n_{poly} of the VRP waypoint interpolation function. The polynomial splines have the following general form:

$$\mathbf{v}_\varphi(t_\varphi) = (1 - f_\varphi(t_\varphi)) \mathbf{v}_{0,\varphi} + f_\varphi(t_\varphi) \mathbf{v}_{T,\varphi}, \quad (3.9)$$

where $\mathbf{v}_{0,\varphi} = \mathbf{v}_\varphi(0)$, $\mathbf{v}_{T,\varphi} = \mathbf{v}_\varphi(T_\varphi)$ and $t_\varphi \in [0, T_\varphi]$ is the local time of the transition phase φ . Concerning the transition between walking and running in Chapter 5, it is convenient to use the same continuity requirements between the transition phases φ for walking and running. Since the VRP interpolation directly influences the smoothness of the total force via (3.6) and the gravitational force is always constant, it is only necessary to compare the smoothness of the leg force. In running the leg force is C^0 continuous on the boundary of the transition phases, i.e. between the flight and stance phase. Therefore, it is sufficient to choose a linear interpolation function for the VRP waypoints of the following form:

$$f_\varphi(t_\varphi) = \frac{t_\varphi}{T_\varphi}. \quad (3.10)$$

Higher-order interpolation schemes are presented by Engelsberger et al. [11]. Inserting the time-dependent VRP trajectory (3.9) into the DCM dynamics (3.7) results in an ODE for the DCM dynamics in the transition phase φ :

$$\dot{\boldsymbol{\xi}}_\varphi(t_\varphi) = \frac{1}{b} (\boldsymbol{\xi}_\varphi(t_\varphi) - \mathbf{v}_\varphi(t_\varphi)). \quad (3.11)$$

Multiplying (3.11) by the integrating factor $e^{-t_\varphi/b}$ facilitates partial integration and results in the solution of the ODE as

$$\boldsymbol{\xi}_\varphi(t_\varphi) = \mathbf{v}_{\Sigma,\varphi}(t_\varphi) + e^{\frac{t_\varphi}{b}} (\boldsymbol{\xi}_{0,\varphi} - \mathbf{v}_{\Sigma,\varphi}(0)), \quad (3.12)$$

where

$$\mathbf{v}_{\Sigma,\varphi}(t_\varphi) = (1 - \sigma_\varphi(t_\varphi)) \mathbf{v}_{0,\varphi} + \sigma_\varphi(t_\varphi) \mathbf{v}_{T,\varphi}, \quad (3.13)$$

with

$$\sigma_\varphi(t_\varphi) = \sum_{j=0}^{n_{poly}} \left(b^j f_\varphi^{(j)}(t_\varphi) \right). \quad (3.14)$$

Here, $\boldsymbol{\xi}_{0,\varphi} = \boldsymbol{\xi}_\varphi(0)$ is the DMC start point and the operator $\square^{(j)}$ denotes the j -th derivative of any function \square . With (3.13), the solution (3.12) can be reformulated to result in an equation, which depends on the VRP start and end point $\mathbf{v}_{0,\varphi}$ and $\mathbf{v}_{T,\varphi}$, respectively, and the DCM start point $\boldsymbol{\xi}_{0,\varphi}$ as boundary conditions:

$$\boldsymbol{\xi}_\varphi(t_\varphi) = \left(1 - \sigma_\varphi(t_\varphi) - e^{\frac{t_\varphi}{b}} (1 - \sigma_{0,\varphi}) \right) \mathbf{v}_{0,\varphi} + \left(\sigma_\varphi(t_\varphi) - e^{\frac{t_\varphi}{b}} \sigma_{0,\varphi} \right) \mathbf{v}_{T,\varphi} + e^{\frac{t_\varphi}{b}} \boldsymbol{\xi}_{0,\varphi}, \quad (3.15)$$

where $\sigma_{0,\varphi} = \sigma_\varphi(0)$. The solution to (3.11) can alternatively be expressed in terms of both VRP start and end points and the DCM end point $\boldsymbol{\xi}_{T,\varphi} = \boldsymbol{\xi}_\varphi(T_\varphi)$:

$$\begin{aligned} \boldsymbol{\xi}_\varphi(t_\varphi) = & \underbrace{\left(1 - \sigma_\varphi(t_\varphi) - e^{\frac{t_\varphi - T_\varphi}{b}} (1 - \sigma_{T,\varphi}) \right)}_{\alpha_{\xi,\varphi}(t_\varphi)} \mathbf{v}_{0,\varphi} + \\ & + \underbrace{\left(\sigma_\varphi(t_\varphi) - e^{\frac{t_\varphi - T_\varphi}{b}} \sigma_{T,\varphi} \right)}_{\beta_{\xi,\varphi}(t_\varphi)} \mathbf{v}_{T,\varphi} + \underbrace{e^{\frac{t_\varphi - T_\varphi}{b}}}_{\gamma_{\xi,\varphi}(t_\varphi)} \boldsymbol{\xi}_{T,\varphi}, \end{aligned} \quad (3.16)$$

where $\sigma_{T,\varphi} = \sigma_\varphi(T_\varphi)$.

Further, the DCM trajectory (3.16) can be inserted in the CoM dynamics (3.2) to find the ODE for the CoM dynamics in transition phase φ :

$$\dot{\mathbf{x}}_\varphi(t_\varphi) = -\frac{1}{b} (\mathbf{x}_\varphi(t_\varphi) - \boldsymbol{\xi}_\varphi(t_\varphi)) \quad (3.17)$$

Again, the partial integration of the CoM dynamics is enabled by multiplication with the integrating factor $e^{t_\varphi/b}$ and the solution of (3.17) results in:

$$\mathbf{x}_\varphi(t_\varphi) = \boldsymbol{\xi}_{\Sigma,\varphi}(t_\varphi) + e^{-\frac{t_\varphi}{b}} (\mathbf{x}_{0,\varphi} - \boldsymbol{\xi}_{\Sigma,\varphi}(0)), \quad (3.18)$$

where

$$\begin{aligned} \boldsymbol{\xi}_{\Sigma,\varphi}(t_\varphi) = & \left(1 - e^{-\frac{t_\varphi}{b}} - \rho_\varphi(t_\varphi) - \frac{e^{\frac{2t_\varphi - T_\varphi}{b}} - e^{-\frac{T_\varphi}{b}}}{2e^{\frac{T_\varphi}{b}}} (1 - \sigma_{T,\varphi}) \right) \mathbf{v}_{0,\varphi} + \\ & + \left(\rho_\varphi(t_\varphi) - \frac{e^{\frac{2t_\varphi - T_\varphi}{b}} - e^{-\frac{T_\varphi}{b}}}{2e^{\frac{T_\varphi}{b}}} \sigma_{T,\varphi} \right) \mathbf{v}_{T,\varphi} + \\ & + \underbrace{\frac{e^{\frac{2t_\varphi - T_\varphi}{b}} - e^{-\frac{T_\varphi}{b}}}{2e^{\frac{T_\varphi}{b}}}}_{\gamma_{x,\varphi}(t_\varphi)} \boldsymbol{\xi}_{T,\varphi}, \end{aligned} \quad (3.19)$$

with

$$\rho_\varphi(t_\varphi) = \sum_{j=0}^{n_{poly}} \left((-1)^j b^j \sigma_\varphi^{(j)}(t_\varphi) \right). \quad (3.20)$$

The CoM start point is given by $\mathbf{x}_{0,\varphi}$. With (3.19), the solution (3.18) can be reformulated as

$$\begin{aligned} \mathbf{x}_\varphi(t_\varphi) = & \underbrace{\left(1 - e^{-\frac{t_\varphi}{b}} (1 + \rho_{0,\varphi}) - \rho_\varphi(t_\varphi) - \gamma_{x,\varphi}(t_\varphi) (1 - \sigma_{T,\varphi}) \right)}_{\alpha_{x,\varphi}(t_\varphi)} \mathbf{v}_{0,\varphi} + \\ & + \underbrace{\left(\rho_\varphi(t_\varphi) - e^{-\frac{t_\varphi}{b}} \rho_{0,\varphi} - \gamma_{x,\varphi}(t_\varphi) \sigma_{T,\varphi} \right)}_{\beta_{x,\varphi}(t_\varphi)} \mathbf{v}_{T,\varphi} + \\ & + \gamma_{x,\varphi}(t_\varphi) \boldsymbol{\xi}_{T,\varphi} + \underbrace{e^{-\frac{t_\varphi}{b}}}_{\delta_{x,\varphi}(t_\varphi)} \mathbf{x}_{0,\varphi}, \end{aligned} \quad (3.21)$$

where $\rho_{0,\varphi} = \rho_\varphi(0)$. This equation returns the CoM trajectory in the transition phase φ , using the VRP start and end points, the DCM end point and the CoM start point as input.

3.2.2 Computation of multi-step preview matrices

This section extends the trajectory generation in a single phase to a gait sequence with multiple steps and transition phases. The considered walking sequence consists of N VRP waypoints and $N_\varphi = N-1$ transition phases. As described at the beginning of Section 3.2, during each transition phase φ , the VRP trajectory is generated by interpolation between the φ -th VRP waypoint

$$\mathbf{v}_\varphi(t_\varphi = 0) = \mathbf{v}_{0,\varphi} = \mathbf{v}_{wp,\varphi} \quad (3.22)$$

and the $(\varphi+1)$ -th VRP waypoint

$$\mathbf{v}_\varphi(t_\varphi = T_\varphi) = \mathbf{v}_{T,\varphi} = \mathbf{v}_{wp,\varphi+1}. \quad (3.23)$$

Since all VRP waypoints are known from (3.5), the VRP trajectory for the complete walking sequence can be computed by (3.9) and (3.10). Now the DCM and CoM trajectories that are consistent with the VRP trajectory can be derived. First, a terminal constraint at the end of the walking sequence, i.e. on the last DCM waypoint $\boldsymbol{\xi}_{wp,N}$, is used to find the consistent DCM waypoints by backward iteration. For this, the trajectory in transition phase φ (3.16) is evaluated for $t_\varphi = 0$ to obtain

$$\boldsymbol{\xi}_{0,\varphi} = \underbrace{\left(1 - \sigma_{0,\varphi} - e^{-\frac{T_\varphi}{b}} (1 - \sigma_{T,\varphi}) \right)}_{\alpha_{\xi,0,\varphi}} \mathbf{v}_{0,\varphi} + \underbrace{\left(\sigma_{0,\varphi} - e^{-\frac{T_\varphi}{b}} \sigma_{T,\varphi} \right)}_{\beta_{\xi,0,\varphi}} \mathbf{v}_{T,\varphi} + \underbrace{e^{-\frac{T_\varphi}{b}}}_{\gamma_{\xi,0,\varphi}} \boldsymbol{\xi}_{T,\varphi}. \quad (3.24)$$

This equation gives the DCM start point in each transition phase and therefore all DCM waypoints except for the last one, which is defined by the terminal constraint:

$$\boldsymbol{\xi}_{T,N_\varphi} = \boldsymbol{\xi}_{wp,N} = \mathbf{v}_{wp,N}. \quad (3.25)$$

The terminal constraint ensures that the DCM comes to a stop at the end of the walking sequence, i.e. $\dot{\xi}_\varphi(t_\varphi = T_\varphi) = 0$. Starting from this terminal condition, the DCM end points $\xi_{T,\varphi}$ of each transition phase $\varphi \in \{1, \dots, N_\varphi - 1\}$ are chosen to coincide with the start points of the next transition phase $\xi_{0,\varphi+1}$:

$$\xi_{T,\varphi} = \xi_{0,\varphi+1} \quad \forall \varphi \in \{1, \dots, N_\varphi - 1\}. \quad (3.26)$$

This backward iteration scheme assures continuity between transition phases. With $\xi_{0,\varphi} = \xi_{wp,\varphi}$, $\xi_{T,\varphi} = \xi_{wp,\varphi+1}$, (3.22) and (3.23), the solution to the DCM Dynamics (3.24) can be written in terms of VRP and DCM waypoints:

$$\xi_{wp,\varphi} = \alpha_{\xi,0,\varphi} \mathbf{v}_{wp,\varphi} + \beta_{\xi,0,\varphi} \mathbf{v}_{wp,\varphi+1} + \gamma_{\xi,0,\varphi} \xi_{wp,\varphi+1}. \quad (3.27)$$

Equations (3.25) and (3.27) can be written in matrix form as

$$\underbrace{\begin{bmatrix} \xi_{wp,1} \\ \vdots \\ \vdots \\ \vdots \\ \xi_{wp,N} \end{bmatrix}}_{\xi_{wp}} = \underbrace{\begin{bmatrix} 0 & \gamma_{\xi,0,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \gamma_{\xi,0,N_\varphi} \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix}}_{\mathbf{A}_\xi \in \mathbb{R}^{N \times N}} \underbrace{\begin{bmatrix} \xi_{wp,1} \\ \vdots \\ \vdots \\ \vdots \\ \xi_{wp,N} \end{bmatrix}}_{\xi_{wp}} + \underbrace{\begin{bmatrix} \alpha_{\xi,0,1} & \beta_{\xi,0,1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \alpha_{0,N_\varphi} & \beta_{0,N_\varphi} \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix}}_{\mathbf{A}_v \in \mathbb{R}^{N \times N}} \underbrace{\begin{bmatrix} \mathbf{v}_{wp,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{v}_{wp,N} \end{bmatrix}}_{\mathbf{v}_{wp}}. \quad (3.28)$$

Rearranging (3.28) and solving for the DCM waypoint vector ξ_{wp} gives

$$\xi_{wp} = (\mathbf{I} - \mathbf{A}_\xi)^{-1} \mathbf{A}_v \mathbf{v}_{wp}, \quad (3.29)$$

which relates the VRP waypoints \mathbf{v}_{wp} to the DCM waypoints ξ_{wp} in a single matrix equation.

The consistent CoM waypoints are found by iterating forwards from a CoM start point

$$\mathbf{x}_{0,1} = \mathbf{x}_{wp,1} = \mathbf{v}_{wp,1}. \quad (3.30)$$

Therefore, the trajectory (3.19) is evaluated for $t_\varphi = T_\varphi$ to obtain:

$$\begin{aligned} \mathbf{x}_{T,\varphi} = & \underbrace{\left(1 - e^{-\frac{T_\varphi}{b}}(1 + \rho_{0,\varphi}) - \rho_{T,\varphi} - \gamma_{x,T,\varphi}(1 - \sigma_{T,\varphi})\right)}_{\alpha_{x,T,\varphi}} \mathbf{v}_{0,\varphi} + \\ & + \underbrace{\left(\rho_{T,\varphi} - e^{-\frac{T_\varphi}{b}}\rho_{0,\varphi} - \gamma_{x,T,\varphi}\sigma_{T,\varphi}\right)}_{\beta_{x,T,\varphi}} \mathbf{v}_{T,\varphi} + \\ & + \gamma_{x,T,\varphi} \xi_{T,\varphi} + \underbrace{e^{-\frac{T_\varphi}{b}}\mathbf{x}_{0,\varphi}}_{\delta_{x,T,\varphi}}. \end{aligned} \quad (3.31)$$

This equation gives the CoM end point in each transition phase. Starting from the initial condition (3.30), the CoM start points $\mathbf{x}_{0,\varphi}$ of each transition phase $\varphi \in \{2, \dots, N_\varphi\}$ are defined to coincide with the end points of the previous transition phase $\mathbf{x}_{T,\varphi-1}$:

$$\mathbf{x}_{0,\varphi} = \mathbf{x}_{T,\varphi-1} \quad \forall \varphi \in \{2, \dots, N_\varphi\}. \quad (3.32)$$

This forward iteration scheme guarantees continuity between the transition phases. With $\mathbf{x}_{0,\varphi} = \mathbf{x}_{wp,\varphi}$, $\mathbf{x}_{T,\varphi} = \mathbf{x}_{wp,\varphi+1}$, $\boldsymbol{\xi}_{T,\varphi} = \boldsymbol{\xi}_{wp,\varphi+1}$, (3.22) and (3.23), the solution to the CoM Dynamics (3.31) can be written as:

$$\mathbf{x}_{wp,\varphi+1} = \alpha_{x,T,\varphi} \mathbf{v}_{wp,\varphi} + \beta_{x,T,\varphi} \mathbf{v}_{wp,\varphi+1} + \gamma_{x,T,\varphi} \boldsymbol{\xi}_{wp,\varphi+1} + \delta_{x,T,\varphi} \mathbf{x}_{wp,\varphi}. \quad (3.33)$$

Equations (3.30) and (3.33) can be written in matrix form as

$$\begin{aligned} \underbrace{\begin{bmatrix} \mathbf{x}_{wp,1} \\ \vdots \\ \mathbf{x}_{wp,N} \end{bmatrix}}_{\mathbf{x}_{wp}} &= \underbrace{\begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ \delta_{x,T,1} & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \delta_{x,T,N_\varphi} & 0 \end{bmatrix}}_{\mathbf{B}_x \in \mathbb{R}^{N \times N}} \underbrace{\begin{bmatrix} \mathbf{x}_{wp,1} \\ \vdots \\ \mathbf{x}_{wp,N} \end{bmatrix}}_{\mathbf{x}_{wp}} + \\ &+ \underbrace{\begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ 0 & \gamma_{x,T,1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \gamma_{x,T,N_\varphi} \end{bmatrix}}_{\mathbf{B}_\xi \in \mathbb{R}^{N \times N}} \underbrace{\begin{bmatrix} \boldsymbol{\xi}_{wp,1} \\ \vdots \\ \boldsymbol{\xi}_{wp,N} \end{bmatrix}}_{\boldsymbol{\xi}_{wp}} + \\ &+ \underbrace{\begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ \alpha_{x,T,1} & \beta_{x,T,1} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \alpha_{x,T,N_\varphi} & \beta_{x,T,N_\varphi} \end{bmatrix}}_{\mathbf{B}_v \in \mathbb{R}^{N \times N}} \underbrace{\begin{bmatrix} \mathbf{v}_{wp,1} \\ \vdots \\ \mathbf{v}_{wp,N} \end{bmatrix}}_{\mathbf{v}_{wp}}. \end{aligned} \quad (3.34)$$

Rearranging (3.34) and solving for the CoM waypoint vector \mathbf{x}_{wp} gives

$$\mathbf{x}_{wp} = (\mathbf{I} - \mathbf{B}_x)^{-1} (\mathbf{B}_\xi \boldsymbol{\xi}_{wp} + \mathbf{B}_v \mathbf{v}_{wp}), \quad (3.35)$$

which yields all CoM waypoints \mathbf{x}_{wp} from the DCM and VRP waypoints $\boldsymbol{\xi}_{wp}$ and \mathbf{v}_{wp} in a single matrix equation.

Chapter 4

Running Trajectory Generation

A biologically inspired trajectory generation introduced by Engelsberger et al. [10] is proposed to generate running motions of the center of mass (CoM). The main idea behind this approach is to design CoM trajectories that produce approximately natural ground reaction forces (GRF) compared to a human experiment. It is shown that the human GRF profiles can be well approximated by a second-order polynomial in the vertical direction and a third-order polynomial in the horizontal direction. The total force on the CoM \mathbf{F}_{com} consists of the gravitational force \mathbf{F}_g and the leg force \mathbf{F}_{leg} :

$$\mathbf{F}_{com} = \mathbf{F}_g + \mathbf{F}_{leg} = m\mathbf{g} + \mathbf{F}_{leg}. \quad (4.1)$$

Here, m is the robot's total mass and $\mathbf{g} = [0 \ 0 \ -g]^T$ is the gravitational acceleration vector. The forces \mathbf{F}_{com} and \mathbf{F}_{leg} in (4.1) differ only by the constant offset of the gravitational force \mathbf{F}_g . Consequently, according to Newton's second law ($\mathbf{F}_{com} = m\ddot{\mathbf{x}}$), leg forces are represented by the desired second and third-order polynomials, respectively, when the CoM position is expressed by fourth-order polynomials for the vertical direction and fifth-order polynomials for the horizontal direction. Furthermore, different objectives besides the continuity constraints between the stance and flight phases are fulfilled in the vertical and horizontal direction. The vertical trajectory planning is solved locally in each step with the main goal to reach a certain apex height in the following flight phase. The horizontal planning approach aims at finding the most suitable CoM trajectory for a sequence of footsteps and a terminal condition at the end of the running sequence. It is therefore solved globally for the whole running sequence. For this reason, the horizontal and vertical trajectories are generated separately, while the vertical trajectory is calculated before the horizontal one.

The trajectory generation for running can be illustrated by a current flight phase and two successive stance and flight phases, as shown in Fig. 4.1. The stance phase described by polynomials is shown in red. The flight phase, in which the CoM follows a parabolic path similar to that of a thrown object under the influence of gravity, is displayed in blue. The desired relative apex $\Delta z_{apex,des}$ and touch down (TD) heights $\Delta z_{TD,des}$ are used as design parameters. They specify the maximum height of the CoM in the flight phase (i.e. $\dot{z} = 0$) and the height at which the TD should occur. Another design parameter is the total stance time T_s in each step, whereas the flight time results from the aforementioned parameters $\Delta z_{TD,des}$ and $\Delta z_{apex,des}$ and the gravitational acceleration g .

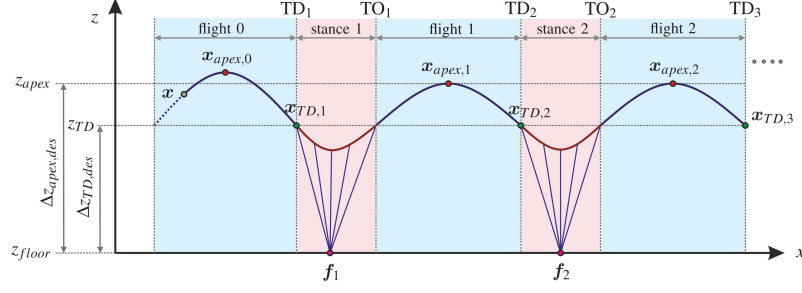


Figure 4.1: Preview of upcoming flight and stance phases (planar sketch) - used for the design of boundary conditions [10].

4.1 Center of mass dynamics in the flight phase

The CoM position $\mathbf{x}(t)$ and velocity $\dot{\mathbf{x}}(t)$ during flight can be described by

$$\mathbf{x}(t) = \mathbf{x}_0 + \dot{\mathbf{x}}_0 t + \mathbf{g} \frac{t^2}{2}, \quad (4.2)$$

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}_0 + \mathbf{g} t, \quad (4.3)$$

where \mathbf{x}_0 and $\dot{\mathbf{x}}_0$ are the initial position and velocity, respectively. As mentioned at the beginning of the chapter, an important task of running trajectory planning is to achieve the desired apex height in the flight phase. At the apex, the highest point of a ballistic curve, the vertical CoM velocity is zero: $\dot{z}_{\text{apex}} = 0$. Starting from a current velocity \dot{z} , this condition can be used to calculate the current time to apex from the third line of (4.3) as

$$\Delta t_{\text{apex}} = \frac{\dot{z}}{g}. \quad (4.4)$$

The remaining time to TD is dependent on the TD height $z_{\text{TD}} = z_{\text{floor}} + \Delta z_{\text{TD}}$ and is computed as

$$\Delta t_{\text{TD}} = \Delta t_{\text{apex}} + \sqrt{\Delta t_{\text{apex}}^2 + \frac{2}{g} (z - z_{\text{TD}})} \quad (4.5)$$

Inserting (4.5) into (4.2) and (4.3) yields the upcoming TD state for a given CoM position and velocity as

$$\begin{bmatrix} \mathbf{x}_{\text{TD}} \\ \dot{\mathbf{x}}_{\text{TD}} \end{bmatrix} = \begin{bmatrix} \mathbf{x} + \Delta t_{\text{TD}} \dot{\mathbf{x}} + \frac{\Delta t_{\text{TD}}^2}{2} \mathbf{g} \\ \dot{\mathbf{x}} + \Delta t_{\text{TD}} \mathbf{g} \end{bmatrix} \quad (4.6)$$

The relative TD height Δz_{TD} is either defined as $\Delta z_{\text{TD,des}}$ or a minimum height difference between apex and TD $\Delta z_{\text{apex,TD,min}}$ is guaranteed, such that the square root term in (4.5) is positive and a real solution is obtained:

$$\Delta z_{\text{TD}} = \min \left(\Delta z_{\text{TD,des}}, z - z_{\text{floor}} + \frac{\dot{z}^2}{2g} - \Delta z_{\text{apex,TD,min}} \right). \quad (4.7)$$

This describes the dynamics of the CoM in the flight phase. In the following, the CoM dynamics in the stance phase, divided into the vertical and horizontal direction, will be examined.

4.2 Vertical planning and boundary conditions

The fourth-order polynomial encoding is given by

$$\begin{bmatrix} z(t) \\ \dot{z}(t) \\ \ddot{z}(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 \\ 0 & 1 & 2t & 3t^2 & 4t^3 \\ 0 & 0 & 2 & 6t & 12t^2 \end{bmatrix} \mathbf{p}_z = \begin{bmatrix} \mathbf{t}_z^T(t) \\ \mathbf{t}_{\dot{z}}^T(t) \\ \mathbf{t}_{\ddot{z}}^T(t) \end{bmatrix} \mathbf{p}_z, \quad (4.8)$$

where \mathbf{t}_z^T , $\mathbf{t}_{\dot{z}}$ and $\mathbf{t}_{\ddot{z}}$ are time-mapping row vectors that map the polynomial parameter vector $\mathbf{p}_z \in \mathbb{R}^5$ to the CoM position z , velocity \dot{z} and acceleration \ddot{z} for a given time t . The five polynomial parameters can be derived using five boundary conditions. For each stance phase i , four linear vertical boundary conditions are determined as:

$$\underbrace{\begin{bmatrix} z_{\text{TD},i} \\ \dot{z}_{\text{TD},i} \\ -g \\ -g \end{bmatrix}}_{\mathbf{b}_{z,i}} = \underbrace{\begin{bmatrix} \mathbf{t}_z^T(0) \\ \mathbf{t}_{\dot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(T_{s,i}) \end{bmatrix}}_{\mathbf{B}_{z,i}} \mathbf{p}_{z,i}. \quad (4.9)$$

Here, $\mathbf{b}_{z,i}$ is the boundary condition vector and $\mathbf{B}_{z,i}$ denotes the boundary condition mapping matrix. The first two elements in $\mathbf{b}_{z,i}$ determine that the initial CoM position and velocity are equal to the TD state of the previous flight phase. The other two elements in $\mathbf{b}_{z,i}$ ensure that the CoM acceleration at the beginning and end of the stance phase is the negative gravitational acceleration so that the vertical leg force is zero. The last boundary condition aims to achieve the desired apex height of the next flight phase:

$$z_{\text{apex},i,\text{des}} = z_{\text{floor},i+1} + \Delta z_{\text{apex},\text{des}}. \quad (4.10)$$

The apex height can be computed with (4.2) and (4.4) as:

$$z_{\text{apex},i} = z_{\text{TO},i} + \frac{\dot{z}_{\text{TO},i}^2}{2g}. \quad (4.11)$$

Since the boundary condition is non-linear, it needs to be fulfilled in the nullspace of $\mathbf{B}_{z,i}$. The general solution of (4.9) is given by

$$\mathbf{p}_{z,i} = \underbrace{\mathbf{B}_{z,i}^T (\mathbf{B}_{z,i} \mathbf{B}_{z,i}^T)^{-1} \mathbf{b}_{z,i}}_{\mathbf{p}_{z,i,0}} + \mathbf{r}_{z,i} \tilde{p}_{z,i}, \quad (4.12)$$

where $\tilde{p}_{z,i}$ is a scalar multiple of the one-dimensional nullspace spanned by the vector $\mathbf{r}_{z,i}$, which is computed as

$$\mathbf{r}_{z,i} = \begin{bmatrix} -\mathbf{B}_{z,i,\text{square}}^{-1} \mathbf{b}_{z,i,\text{final}} \\ 1 \end{bmatrix}. \quad (4.13)$$

The vector $\mathbf{b}_{z,i,\text{final}}$ is the last column of $\mathbf{B}_{z,i}$ and $\mathbf{B}_{z,i,\text{square}}$ consists of all other columns. To solve for the unknown scalar variable $\tilde{p}_{z,i}$, the i -th takeoff (TO) state can be expressed via (4.8) as

$$\begin{aligned} z_{\text{TO},i} &= \mathbf{t}_z^T(T_{s,i}) \mathbf{p}_{z,i} \\ \dot{z}_{\text{TO},i} &= \dot{\mathbf{t}}_z^T(T_{s,i}) \mathbf{p}_{z,i} \end{aligned} \quad (4.14)$$

and inserted into (4.11). The only valid solution to the resulting quadratic equation is

$$\begin{aligned} \tilde{p}_{z,i} &= \frac{2\dot{z}_{\text{TD},i} - gT_{s,i} - \Gamma}{4T_{s,i}^3} \quad \text{with} \\ \Gamma &= \sqrt{g \left(gT_{s,i}^2 - 4\dot{z}_{\text{TD},i}T_{s,i} + 8(z_{\text{apex},i,\text{des}} - z_{\text{TD},i}) \right)}. \end{aligned} \quad (4.15)$$

Equations (4.15) and (4.12) form the solution of the vertical polynomial parameter vector and thus the solution of the vertical CoM dynamics of each stance phase.

4.3 Horizontal planning and boundary conditions

The planning for the x - and y -component is equivalent. Therefore, the vector $\boldsymbol{\chi} = [x \ y]$ is chosen to summarize horizontal quantities. The fifth-order polynomial encoding in the horizontal direction is given by

$$\begin{bmatrix} \boldsymbol{\chi}(t) \\ \dot{\boldsymbol{\chi}}(t) \\ \ddot{\boldsymbol{\chi}}(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix} \mathbf{P}_\chi = \begin{bmatrix} \mathbf{t}_\chi^T(t) \\ \dot{\mathbf{t}}_\chi^T(t) \\ \ddot{\mathbf{t}}_\chi^T(t) \end{bmatrix} \mathbf{P}_\chi. \quad (4.16)$$

Here, \mathbf{t}_χ^T , $\dot{\mathbf{t}}_\chi$ and $\ddot{\mathbf{t}}_\chi$ are the time-mapping row vectors for the horizontal direction, which map, for a given time t , the polynomial parameter matrix $\mathbf{P}_\chi = [\mathbf{p}_x \ \mathbf{p}_y] \in \mathbb{R}^{6 \times 2}$ to the CoM position $\boldsymbol{\chi}$, velocity $\dot{\boldsymbol{\chi}}$ and acceleration $\ddot{\boldsymbol{\chi}}$. Due to the higher polynomial order compared to the vertical direction, there is an additional polynomial parameter in the horizontal direction and thus a total of six boundary conditions must be found for the six degrees of freedom. For each stance phase i five linear horizontal boundary conditions are defined as

$$\underbrace{\begin{bmatrix} \boldsymbol{\chi}_{\text{TD},i} \\ \dot{\boldsymbol{\chi}}_{\text{TO},i} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\chi}_{\text{TD},i+1,\text{des}} \end{bmatrix}}_{\mathbf{H}_{\chi,i}} = \underbrace{\begin{bmatrix} \mathbf{t}_\chi^T(0) \\ \dot{\mathbf{t}}_\chi^T(T_{s,i}) \\ \ddot{\mathbf{t}}_\chi^T(0) \\ \ddot{\mathbf{t}}_\chi^T(T_{s,i}) \\ \mathbf{t}_\chi^T(T_{s,i}) + T_{f,i} \mathbf{t}_\chi^T(T_{s,i}) \end{bmatrix}}_{\mathbf{B}_{\chi,i}} \mathbf{P}_{\chi,i}. \quad (4.17)$$

Here, $\mathbf{H}_{\chi,i}$ is the horizontal boundary condition matrix and $\mathbf{B}_{\chi,i}$ denotes the boundary condition mapping matrix. The first row in $\mathbf{H}_{\chi,i}$ sets the initial CoM positions equal to the TD positions. Here, the second row specifies the TO velocities $\dot{\boldsymbol{\chi}}_{\text{TO},i}$ and not the TD velocities $\dot{\boldsymbol{\chi}}_{\text{TD},i}$ as in [10]. The next two rows ensure that the horizontal CoM

accelerations at the beginning and end of the stance phase are zero, i.e. horizontal leg forces are zero. The fifth boundary condition determines the desired CoM TD positions $\chi_{\text{TD},i+1,\text{des}}$ of the next stance phase. Assuming no perturbations in the flight phase and thus constant horizontal velocities, the next TD positions can be calculated as

$$\chi_{\text{TD},i+1,\text{des}} = \chi_{\text{TO},i} + T_{f,i} \dot{\chi}_{\text{TO},i} = \left(\mathbf{t}_{\chi,i}^T(T_{s,i}) + T_{f,i} \mathbf{t}_{\dot{\chi}}^T(T_{s,i}) \right) \mathbf{P}_{\chi,i}. \quad (4.18)$$

Again the last boundary condition is non-linear and is fulfilled in the nullspace of $\mathbf{B}_{\chi,i}$. The general solution of (4.17) is

$$\mathbf{P}_{\chi,i} = \underbrace{\mathbf{B}_{\chi,i}^T \left(\mathbf{B}_{\chi,i} \mathbf{B}_{\chi,i}^T \right)^{-1} \mathbf{H}_{\chi,i} + \mathbf{r}_{\chi,i} \tilde{\mathbf{p}}_{\chi,i}}_{\mathbf{P}_{\chi,i,0}}, \quad (4.19)$$

with $\tilde{\mathbf{p}}_{\chi,i} = [\tilde{p}_{x,i} \ \tilde{p}_{y,i}] \in \mathbb{R}^2$ being the row vector of scalar multiples of the nullspace column vector:

$$\mathbf{r}_{\chi,i} = \begin{bmatrix} -\mathbf{B}_{\chi,i,\text{square}}^{-1} \mathbf{b}_{\chi,i,\text{final}} \\ 1 \end{bmatrix}. \quad (4.20)$$

The goal is to find $\tilde{\mathbf{p}}_{\chi,i}$, i.e. a scalar value for each direction x and y , which provides the best possible focusing of leg forces at an intersection point with the ground. For this purpose, the time-dependent intersection point $\mathbf{x}_{\text{int},i} = [x_{\text{int},i} \ y_{\text{int},i} \ z_{\text{floor},i}]$ of the leg forces with the ground is calculated and the integral of the mean squared deviation $\mathbf{x}_{\text{int},i,\text{ms}}$ from its mean value $\bar{\mathbf{x}}_{\text{int},i}$ (i.e. the variation) is minimized. The horizontal components of $\mathbf{x}_{\text{int},i,\text{ms}}$ are given by

$$\begin{aligned} \chi_{\text{int},i,\text{ms}} &= \mathbf{P}_{\chi,i}^T \underbrace{\frac{1}{T_{s,i}} \int_{t_s=0}^{T_{s,i}} \mathbf{L}_{\chi,i}(t_s) dt_s}_{\mathbf{M}_{\chi,i}} \mathbf{P}_{\chi,i} \\ &\stackrel{(4.19)}{=} \tilde{\mathbf{p}}_{\chi,i}^T \mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{r}_{\chi,i} \tilde{\mathbf{p}}_{\chi,i} + 2 \mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{P}_{\chi,i,0} \tilde{\mathbf{p}}_{\chi,i} + \mathbf{P}_{\chi,i,0}^T \mathbf{M}_{\chi,i} \mathbf{P}_{\chi,i,0}. \end{aligned} \quad (4.21)$$

For details on the analytical computation of matrices $\mathbf{L}_{\chi,i}$ and $\mathbf{M}_{\chi,i}$ see Englsberger et al. [10]. Equating the derivative of (4.21) with respect to $\tilde{\mathbf{p}}_{\chi,i}$ to zero yields, with $\mathbf{M}_{\chi,i}$ symmetric, the optimal vector of scalar nullspace multipliers

$$\tilde{\mathbf{p}}_{\chi,i}^* = \frac{\mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{P}_{\chi,i,0}}{\mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{r}_{\chi,i}}, \quad (4.22)$$

which minimizes the mean squared deviation as described above. Inserting (4.22) into (4.19) gives

$$\mathbf{P}_{\chi,i} = \underbrace{\left(\mathbf{I} - \frac{\mathbf{r}_{\chi,i} \mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i}}{\mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{r}_{\chi,i}} \right)}_{\boldsymbol{\Omega}_{\chi,i}} \underbrace{\mathbf{B}_{\chi,i}^T \left(\mathbf{B}_{\chi,i} \mathbf{B}_{\chi,i}^T \right)^{-1} \mathbf{H}_{\chi,i}}_{\mathbf{B}_{\chi,i}^+}. \quad (4.23)$$

This equation returns the polynomial parameters $\mathbf{P}_{\chi,i}$ for the given horizontal boundary conditions $\mathbf{H}_{\chi,i}$ including the best focusing of the leg forces at the foot contact point (see Fig. 4.2). The disadvantage is that the horizontal position $\bar{\mathbf{x}}_{\text{int},i}$ of the focused leg forces cannot be chosen arbitrarily, leading to problems such as leg crossover or unprecise foot

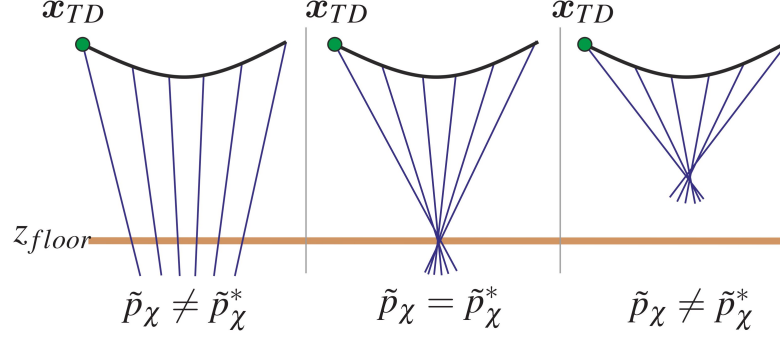


Figure 4.2: Effect of $\tilde{p}_{\chi,i}$ on force ray focusing (lines of action) at the contact point [10].

placement in challenging terrain. So far, the foot position results from the choice of the upcoming desired CoM TD position $\chi_{TD,i+1,des}$. To allow arbitrary foot placement, this dependency is reversed by choosing $\bar{\chi}_{int,i} = \chi_{foot,i}$ to find the upcoming desired CoM TD position, which corresponds to the desired foot position $\chi_{foot,i}$ (see [10]). Inserting this particular $\chi_{TD,i+1,des}$ into (4.23) results in

$$\mathbf{P}_{\chi,i} = \left[\underbrace{\left(\mathbf{I} - \mathbf{e}_{\chi,i}^{\oplus} \mathbf{e}_{\chi,i}^T \right) \Omega_{\chi,i} \boldsymbol{\mu}_{\chi,i}}_{\mathbf{a}_{p,\chi,i}} \underbrace{\left(\mathbf{I} - \mathbf{e}_{\chi,i}^{\oplus} \mathbf{e}_{\chi,i}^T \right) \Omega_{\chi,i} \boldsymbol{\nu}_{\chi,i}}_{\mathbf{a}_{v,\chi,i}} \underbrace{\frac{\Omega_{\chi,i} \boldsymbol{\pi}_{\chi,i}}{\mathbf{e}_{\chi,i}^T \Omega_{\chi,i} \boldsymbol{\pi}_{\chi,i}}}_{\mathbf{e}_{\chi,i}^{\oplus}} \right] \begin{bmatrix} \chi_{TD,i} \\ \dot{\chi}_{TO,i} \\ \chi_{foot,i} \end{bmatrix}, \quad (4.24)$$

where $\mathbf{a}_{p,\chi,i}$, $\mathbf{a}_{v,\chi,i}$ and $\mathbf{e}_{\chi,i}^{\oplus}$ map the i -th TD position, TO velocity and foot position to the matrix of polynomial coefficients $\mathbf{P}_{\chi,i}$. The column vectors $\boldsymbol{\mu}_{\chi,i}$, $\boldsymbol{\nu}_{\chi,i}$ and $\boldsymbol{\pi}_{\chi,i}$ denote the first, second and fifth (final) column of $\mathbf{B}_{\chi,i}^+$. As the third and fourth entry of $\mathbf{b}_{\chi,i}$ is zero (see (4.17)), the corresponding boundary conditions are implicitly satisfied. Equation (4.24) can be written in matrix form as

$$\underbrace{\begin{bmatrix} \mathbf{P}_{\chi,1} \\ \vdots \\ \mathbf{P}_{\chi,N} \end{bmatrix}}_{\mathbf{P}_{\chi}} = \underbrace{\begin{bmatrix} \mathbf{a}_{p,\chi,1} & \mathbf{a}_{v,\chi,1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{a}_{p,\chi,N} & \mathbf{a}_{v,\chi,N} \end{bmatrix}}_{\mathbf{A}_{\chi} \in \mathbb{R}^{6N \times 2N}} \underbrace{\begin{bmatrix} \chi_{TD,1} \\ \dot{\chi}_{TO,1} \\ \vdots \\ \chi_{TD,N} \\ \dot{\chi}_{TO,N} \end{bmatrix}}_{\boldsymbol{\omega}_{wp}} + \underbrace{\begin{bmatrix} \mathbf{e}_{\chi,1}^{\oplus} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{e}_{\chi,N}^{\oplus} \end{bmatrix}}_{\mathbf{E}_{\chi} \in \mathbb{R}^{6N \times N_f}} \underbrace{\begin{bmatrix} \chi_{foot,1} \\ \vdots \\ \chi_{foot,N_f} \end{bmatrix}}_{\mathbf{X}_{foot}}, \quad (4.25)$$

where $\mathbf{P}_\chi \in \mathbb{R}^{6N \times 2}$ summarizes the individual polynomial coefficients matrices $\mathbf{P}_{\chi,i}$ and $\boldsymbol{\omega}_{wp} \in \mathbb{R}^{2N \times 2}$ contains the TO position $\boldsymbol{\chi}_{TD,i}$ and TD velocity $\dot{\boldsymbol{\chi}}_{TO,i}$ for each stance phase. Matrix \mathbf{A}_χ is the constraint mapping matrix, \mathbf{E}_χ is the footstep mapping matrix and N_f denotes the number of footsteps. With a desired set of footsteps $\mathbf{x}_{\text{foot},i} = \mathbf{x}_{\text{foot},i,\text{des}}$, an initial TD position

$$\boldsymbol{\chi}_{TD,1} = \boldsymbol{\chi}_{TD,1,\text{des}} \quad (4.26)$$

and a terminal TO velocity

$$\dot{\boldsymbol{\chi}}_{TO,N} = \dot{\boldsymbol{\chi}}_{TO,N,\text{des}}, \quad (4.27)$$

a complete multistep trajectory can be calculated. The choice of the terminal velocity boundary condition guarantees the stability of the running trajectory and allows for the integration into a series of several alternating walking and running sequences in Chapter 5. Each upcoming TD position can be recursively defined as

$$\boldsymbol{\chi}_{TD,i+1} = \underbrace{\left(\mathbf{t}_{\chi,i}^T(T_{s,i}) + T_{f,i} \mathbf{t}_{\chi}^T(T_{s,i}) \right)}_{\mathbf{f}_i} \mathbf{P}_{\chi,i} \quad (4.28)$$

Similarly each previous TO velocity can be determined in a backward recursion as

$$\dot{\boldsymbol{\chi}}_{TO,i-1} = \underbrace{\mathbf{t}_{\chi}^T(0)}_{\mathbf{b}_i} \mathbf{P}_{\chi,i} \quad (4.29)$$

Equations (4.26) to (4.29) can be summarized in matrix form as

$$\underbrace{\begin{bmatrix} \boldsymbol{\chi}_{TD,1} \\ \dot{\boldsymbol{\chi}}_{TO,1} \\ \boldsymbol{\chi}_{TD,2} \\ \dot{\boldsymbol{\chi}}_{TO,2} \\ \vdots \\ \boldsymbol{\chi}_{TD,N-1} \\ \dot{\boldsymbol{\chi}}_{TO,N-1} \\ \boldsymbol{\chi}_{TD,N} \\ \dot{\boldsymbol{\chi}}_{TO,N} \end{bmatrix}}_{\boldsymbol{\omega}_{wp}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_2 & \ddots & & & & \vdots \\ \mathbf{f}_1 & \mathbf{0} & \mathbf{0} & \ddots & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{b}_3 & & \ddots & & \vdots \\ \vdots & \ddots & & \ddots & & \ddots & \vdots \\ \vdots & & \ddots & & \mathbf{f}_{N-2} & \mathbf{0} & \mathbf{0} \\ \vdots & & & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{b}_N \\ \vdots & & & & \ddots & \mathbf{f}_{N-1} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{S}_\chi \in \mathbb{R}^{2N \times 6N}} \underbrace{\begin{bmatrix} \mathbf{P}_{\chi,1} \\ \mathbf{P}_{\chi,2} \\ \mathbf{P}_{\chi,3} \\ \vdots \\ \mathbf{P}_{\chi,N-2} \\ \mathbf{P}_{\chi,N-1} \\ \mathbf{P}_{\chi,N} \end{bmatrix}}_{\mathbf{P}_\chi} + \underbrace{\begin{bmatrix} \boldsymbol{\chi}_{TD,1,\text{des}} \\ \mathbf{0} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\boldsymbol{\chi}_I} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\dot{\boldsymbol{\chi}}_T}. \quad (4.30)$$

Here, \mathbf{S}_χ represents the recursive mapping of the polynomial parameter matrix to the waypoint matrix and $\boldsymbol{\chi}_I$ and $\dot{\boldsymbol{\chi}}_T$ contain the initial condition and terminal condition for the first TD position and last TO velocity, respectively. Inserting (4.25) into (4.30) and solving for the running waypoint matrix $\boldsymbol{\omega}_{wp}$ yields

$$\boldsymbol{\omega}_{wp} = (\mathbf{I} - \mathbf{S}_\chi \mathbf{A}_\chi)^{-1} (\mathbf{S}_\chi \mathbf{E}_\chi \mathbf{X}_{\text{foot}} + \boldsymbol{\chi}_I + \dot{\boldsymbol{\chi}}_T). \quad (4.31)$$

The solution for the horizontal parameter matrix \mathbf{P}_χ can be obtained by inserting (4.31) into (4.25), which results in

$$\boldsymbol{\omega}_{wp} = \mathbf{A}_\chi (\mathbf{I} - \mathbf{S}_\chi \mathbf{A}_\chi)^{-1} (\mathbf{S}_\chi \mathbf{E}_\chi \mathbf{X}_{\text{foot}} + \boldsymbol{\chi}_I + \dot{\boldsymbol{\chi}}_T) + \mathbf{E}_\chi \mathbf{X}_{\text{foot}}. \quad (4.32)$$

This equation returns the polynomial parameter matrix \mathbf{P}_χ and thus the solution of the horizontal CoM dynamics of each stance phase.

Chapter 5

Continuous Gait Transitions between Walking and Running

In the following chapter, the transition between walking and running is presented. Due to the separate calculation of vertical and horizontal components of the center of mass (CoM) trajectory in the running framework, this distinction is also maintained for the computation of the gait transition trajectories. Moreover, the continuity requirements for the gait transition are determined by the running algorithm, since the CoM position in running is C^2 continuous by design. Thus, the position, velocity and acceleration of the CoM need to be continuous between different gaits, resulting in three transition boundary conditions for each gait change. However, due to the use of the DCM as a state

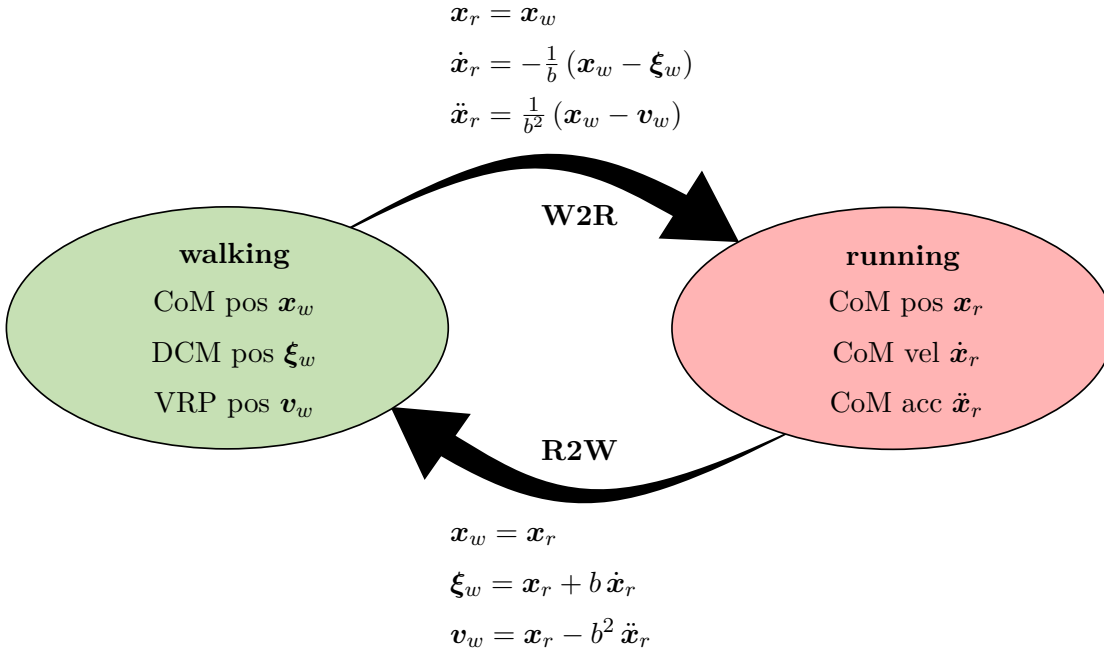


Figure 5.1: Correlations between walking and running quantities.

variable and the VRP encoding the accelerations in the walking algorithm, it is necessary to convert between the walking and running quantities in each gait transition in order to fulfill the boundary conditions between walking and running. Fig. 5.1 illustrates the

relation between walking and running quantities. Here, the transition from walking to running is denoted walk-to-run (W2R) transition and the transition from running to walking is called run-to-walk (R2W) transition.

In the vertical direction, the trajectory can be solved in a forward recursion from the first to the last gait sequence. The coupling of the states by conversion between the walking and running quantities and additional initial and final conditions in the walking sequence leads to the fact that the entire trajectory in the horizontal direction must be solved in one computation. In the following, the trajectory generation for the vertical and horizontal direction is presented.

5.1 Vertical planning

Trajectory generation for walking essentially involves solving a second-order differential equation, where it is sufficient to specify two boundary conditions. For running, the vertical component of the trajectory is obtained by specifying five boundary conditions for a polynomial spline. Therefore, most of the transition boundary conditions are fulfilled by the running algorithm. In the W2R transition, all three vertical boundary conditions are satisfied by the running algorithm, while in the R2W transition, two boundary conditions are fulfilled by the running algorithm and one by the walking algorithm.

Table 5.1 shows the first and final CoM position, velocity and acceleration waypoints and the corresponding transition boundary conditions for three consecutive gait sequences. Additionally, the first and last DCM position waypoint is displayed in each sequence. The color denotes which algorithm fulfills the boundary condition. According to the definition of the DCM (see (3.1)), the unhighlighted boundary conditions are implicitly satisfied by fulfilling either CoM position and velocity or CoM and DCM position boundary conditions in the W2R transition or R2W transition, respectively.



gait sequence	...	walking j		running j		walking $j+1$...
transition		W2R		R2W				
waypoint		start	end	start	end	start	end	
DCM position	...	$\xi_{z,1,w_j}$	ξ_{z,N,w_j} ①	$\xi_{z,TD,1,r_j}$	ξ_{z,TO,N,r_j} ②	$\xi_{z,1,w_{j+1}}$	$\xi_{z,N,w_{j+1}}$...
CoM position	...	z_{1,w_j}	z_{N,w_j} ③	$z_{TD,1,r_j}$	z_{TO,N,r_j} ④	$z_{1,w_{j+1}}$	$z_{N,w_{j+1}}$...
CoM velocity	...	\dot{z}_{1,w_j}	\dot{z}_{N,w_j} ⑤	$\dot{z}_{TD,1,r_j}$	\dot{z}_{TO,N,r_j} ⑥	$\dot{z}_{1,w_{j+1}}$	$\dot{z}_{N,w_{j+1}}$...
CoM acceleration	...	\ddot{z}_{1,w_j}	\ddot{z}_{N,w_j} ⑦	$\ddot{z}_{TD,1,r_j}$	\ddot{z}_{TO,N,r_j} ⑧	$\ddot{z}_{1,w_{j+1}}$	$\ddot{z}_{N,w_{j+1}}$...
		boundary condition fulfilled by walking: 				boundary condition fulfilled by running: 		

Table 5.1: Vertical transition boundary conditions for three consecutive gait phases.

As described in Chapter 4, the vertical planning for running is solved locally in each stance phase with the main goal of reaching the desired apex height in the subsequent flight phase. By choosing the goal of matching the initial DCM position of the subsequent walking sequence in the last stance phase, all vertical transition boundary conditions can be fulfilled by the running algorithm, except for the CoM position boundary condition in the R2W transition. The walking algorithm satisfies the remaining transition boundary

condition by selecting the last CoM takeoff (TO) position from running as the initial position for the following walking sequence. Thus, the position boundary conditions can be directly assigned as

$$\textcircled{3} : z_{\text{TD},1,r_j} = z_{N,w_j} \text{ and} \quad (5.1)$$

$$\textcircled{4} : z_{1,w_{j+1}} = z_{\text{TO},N,r_j}. \quad (5.2)$$

Due to the DCM as a state variable during walking, there is a DCM boundary condition in the R2W transition, which can be written in terms of the polynomial encoding form (4.8) as

$$\begin{aligned} \textcircled{2} : \quad \xi_{z,\text{TO},N,r_j} &= z_{\text{TO},N,r_j} + b \dot{z}_{\text{TO},N,r_j} \\ &\stackrel{(4.8)}{=} \left(\mathbf{t}_z^T(T_{s,N}) + b \mathbf{t}_{\dot{z}}^T(T_{s,N}) \right) \mathbf{p}_z \stackrel{!}{=} \xi_{z,1,w_{j+1}}. \end{aligned} \quad (5.3)$$

The velocity boundary condition is specified by using the DCM definition (3.1) as

$$\textcircled{5} : \dot{z}_{\text{TD},1,r_j} = \dot{z}_{N,w_j} = \frac{1}{b} \left(\xi_{z,N,w_j} - z_{N,w_j} \right). \quad (5.4)$$

With $\ddot{z} = F_z/m$ and the third row of (3.6) the vertical acceleration boundary condition in the W2R transition can be formulated in terms of the CoM and VRP position as

$$\textcircled{7} : \ddot{z}_{\text{TD},1,r_j} = \ddot{z}_{N,w_j} = \frac{1}{b^2} \left(z_{N,w_j} - v_{z,N,w_j} \right). \quad (5.5)$$

Since only the VRP position waypoints in the next walking sequence w_{j+1} are known, the acceleration boundary condition $\textcircled{8}$ in the R2W transition is in fact a VRP boundary condition:

$$\begin{aligned} v_{z,\text{TO},N,r_j} &= z_{\text{TO},N,r_j} - b^2 \ddot{z}_{\text{TO},N,r_j} \\ &\stackrel{(4.8)}{=} \left(\mathbf{t}_z^T(T_{s,N}) - b^2 \mathbf{t}_{\ddot{z}}^T(T_{s,N}) \right) \mathbf{p}_z \stackrel{!}{=} v_{z,1,w_{j+1}}. \end{aligned} \quad (5.6)$$

By satisfying the CoM (5.2) and VRP (5.6) position boundary condition, the acceleration boundary condition $\textcircled{8}$ in the R2W transition is implicitly fulfilled. With the highlighted transition boundary conditions in Table 5.1, all other boundary conditions are also satisfied as the not highlighted DCM $\textcircled{1}$ and CoM velocity $\textcircled{6}$ boundary conditions are implicitly satisfied by fulfilling boundary conditions $\textcircled{3}$ and $\textcircled{5}$ and boundary conditions $\textcircled{2}$ and $\textcircled{4}$, respectively.

5.1.1 Running boundary conditions

Table 5.2 shows the boundary condition vector $\mathbf{b}_{z,i}$, boundary condition mapping matrix $\mathbf{B}_{z,i}$ and the nullspace target for different stance phases in the running sequence. For stance phase 1 to $N-1$ there are four linear boundary conditions and one nonlinear boundary condition (apex height), that is fulfilled in the nullspace of $\mathbf{B}_{z,i}$. In stance phase 1, the boundary conditions (5.1), (5.4) and (5.5) are used to specify the initial conditions of this stance phase, whereas the acceleration terminal condition is set to $\ddot{z}_{\text{TO},1} = -g$. For stance phases $i = 2..N-1$, the matrices are unchanged to the ones presented in Section 4.2. In stance phase N , there are five boundary conditions, which are all linear. The first three are the same as for stance phases $i = 2..N-1$ and the last two are defined as boundary conditions (5.3) and (5.6). Thereby, in each stance phase, all five boundary conditions for the running trajectory generation are given.

stance phase	phase 1	phase $i = 2..N-1$	phase N
boundary condition vector	$\mathbf{b}_{z,1} = \begin{bmatrix} z_{TD,1,r_j} \\ \dot{z}_{TD,1,r_j} \\ \ddot{z}_{TD,1,r_j} \\ -g \end{bmatrix}$	$\mathbf{b}_{z,i} = \begin{bmatrix} z_{TD,i} \\ \dot{z}_{TD,i} \\ -g \\ -g \end{bmatrix}$	$\mathbf{b}_{z,N} = \begin{bmatrix} z_{TD,N} \\ \dot{z}_{TD,N} \\ -g \\ v_{z,TO,N,r_j} \\ \xi_{z,TO,N,r_j} \end{bmatrix}$
boundary condition mapping matrix	$\mathbf{B}_{z,1} = \begin{bmatrix} \mathbf{t}_z^T(0) \\ \mathbf{t}_{\dot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(T_{s,1}) \end{bmatrix}$	$\mathbf{B}_{z,i} = \begin{bmatrix} \mathbf{t}_z^T(0) \\ \mathbf{t}_{\dot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(T_{s,i}) \end{bmatrix}$	$\mathbf{B}_{z,N} = \begin{bmatrix} \mathbf{t}_z^T(0) \\ \mathbf{t}_{\dot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(0) \\ \mathbf{t}_z^T(T_{s,N}) - b^2 \mathbf{t}_{\ddot{z}}^T(T_{s,N}) \\ \mathbf{t}_z^T(T_{s,N}) + b \mathbf{t}_{\ddot{z}}^T(T_{s,N}) \end{bmatrix}$
nullspace boundary condition	$z_{\text{apex},1} = z_{TO,1} + \frac{\dot{z}_{TO,1}^2}{2g}$	$z_{\text{apex},i} = z_{TO,i} + \frac{\dot{z}_{TO,i}^2}{2g}$	-

Table 5.2: Vertical boundary conditions for different running stance phases displayed as boundary condition vector $\mathbf{b}_{z,i}$, boundary condition mapping matrix $\mathbf{B}_{z,i}$ and the nullspace target.

5.1.2 Walking boundary conditions

With (5.2), only one of two boundary conditions in the walking sequence is defined. Similar to (3.25), the vertical component of the last DCM position is set equal to the last VRP position in the walking sequence, i.e.

$$\xi_{z,N,w_j} = v_{z,N,w_j}, \quad (5.7)$$

to specify the remaining walking boundary condition. With this terminal constraint, the DCM dynamics (of walking sequence w_j) is decoupled from the next running sequence r_j and can be precomputed separately for each walking sequence, while the computation of the CoM dynamics is still dependent on an initial position provided by the previous running or standing sequence.

5.1.3 Vertical planning for multiple walking and running sequences

This section gives an overview of the procedure for calculating the vertical trajectory for multiple walking and running sequences in succession. An outline of the computation flow is given in Fig. 5.2. First, the vertical DCM dynamics is precomputed for all walking sequences since it depends only on the vertical components of the known VRP waypoints and the terminal condition (5.7), which is already included in the third row of (3.29), i.e.

$$\xi_{z,wp,w_j} = (\mathbf{I} - \mathbf{A}_\xi)^{-1} \mathbf{A}_v \mathbf{v}_{z,wp,w_j}. \quad (5.8)$$

This equation returns all DCM waypoints for each walking sequence w_j .

In the following, the CoM dynamics of walking and running are calculated in a forward recursion from the first gait sequence to the last. Since the initial CoM position is

dependent on the previous running or standing sequence, the mapping of the vertical components of the first CoM waypoint to the CoM waypoint vector z_{wp} needs to be separated by modifying (3.35) to

$$z_{wp} = (\mathbf{I} - \mathbf{B}_x)^{-1} (\mathbf{B}_\xi \xi_{z,wp} + \mathbf{B}_{\alpha\beta} \mathbf{v}_{z,wp}) + (\mathbf{I} - \mathbf{B}_x)^{-1} \mathbf{b}_{IC} z_{IC}, \quad (5.9)$$

with

$$\mathbf{B}_{\alpha\beta} = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ \alpha_{x,T,1} & \beta_{x,T,1} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \alpha_{x,T,n_\varphi} & \beta_{x,T,n_\varphi} \end{bmatrix} \quad \text{and} \quad \mathbf{b}_{IC} = \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}. \quad (5.10)$$

Here, Matrix $\mathbf{B}_{\alpha\beta} \in \mathbb{R}^{N \times N}$ is equivalent to matrix \mathbf{B}_v in (3.34) except for the first entry in the upper left corner, which so far constrained the initial CoM position waypoint to be equal to the first VRP waypoint.

Starting with the first walking sequence, the initial CoM position is defined (only for the first walking sequence) as the first standing VRP position (as in equation (3.30)) as

$$z_{1,w_1} = v_{z,1,w_1}, \quad (5.11)$$

Inserting this initial CoM position, the VRP waypoints and the DCM terminal condition (5.7) in (5.9) yields the vertical components of the CoM position waypoints for the first walking sequence.

The vertical running trajectory planning is solved locally in each step with the goal to find polynomial parameters $\mathbf{p}_{z,i}$ that satisfy the following equation (see (4.9))

$$\mathbf{b}_{z,i} = \mathbf{B}_{z,i} \mathbf{p}_{z,i}, \quad (5.12)$$

where the boundary condition vector $\mathbf{b}_{z,i}$ and the boundary condition mapping matrix $\mathbf{B}_{z,i}$ depend on the index of the respective stance phase (see Table 5.2). For the non-boundary stance phases $i = 2..N-1$, the polynomial parameters are calculated using the same procedure as described in Section 4.2. The calculation of the polynomial parameters in the W2R (first stand phase) and R2W transition (last stand phase) is described below.

In the walk-to-run transition, the boundary conditions ③ (5.1), ⑤ (5.4) and ⑦ (5.5) specify the initial TO position, velocity and acceleration in the first stance phase. The complete boundary condition vector for the first stance phase $\mathbf{b}_{z,1}$ is summarized in Table 5.2. Since the system of equations (5.14) is under-determined in the first stance phase and in stance phases $i = 2..N-1$, the last (non-linear) boundary condition must be satisfied in the nullspace of $\mathbf{B}_{z,i}$. Again, the goal is to reach the desired apex height $z_{\text{apex},i,\text{des}}$ in the following flight phase. The derivation follows the one described in Section 4.2. Due to the modified boundary conditions, the only valid solution to the quadratic equation (4.11) results in

$$\tilde{p}_{z,1} = \frac{6(1 + 4T_s^2) \dot{z}_{\text{TD},1} + (3T_s + 16T_s^3) \ddot{z}_{\text{TD},1} + 4T_s^3 g - 3g(1 + 4T_s^2) \Gamma_1}{12(4T_s^5 + T_s^3)} \quad \text{with} \quad (5.13)$$

$$\Gamma_1 = \sqrt{\frac{24z_{\text{apex},i,\text{des}} - 24z_{\text{TD},1} - 12T_s \dot{z}_{\text{TD},1} - 2T_s^2 \ddot{z}_{\text{TD},1} + T_s^2 g}{3g}}.$$

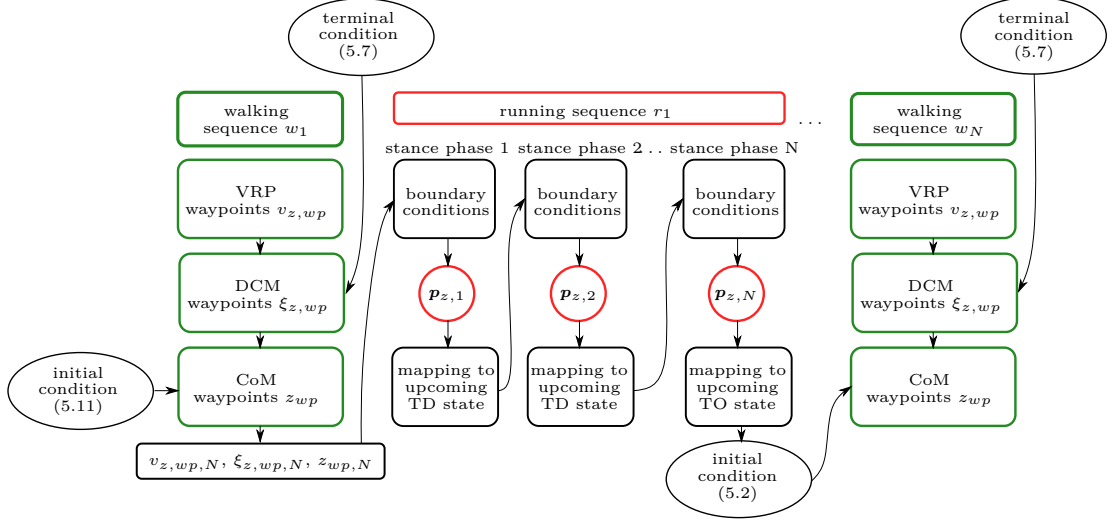


Figure 5.2: Outline of the vertical planning computation flow.

The calculation of the polynomial parameters in the run-to-walk transition is straightforward as $\mathbf{B}_{z,N}$ in (5.14) has full rank. The boundary condition vector $\mathbf{b}_{z,N}$ is assembled with boundary conditions ② (5.3) and the VRP boundary condition (5.6) that implicitly satisfies boundary condition ⑧. The vertical polynomial parameters in the last stance phase are obtained by inversion of $\mathbf{B}_{z,N}$ as

$$\mathbf{p}_{z,N} = (\mathbf{B}_{z,N})^{-1} \mathbf{b}_{z,N}. \quad (5.14)$$

For the next walking sequence, the initial CoM position is specified by boundary condition ④ (5.2). With the DCM trajectory already known, the CoM position of the next walking sequence can be calculated with (5.9). In the described way, the CoM trajectory can be propagated forward to the last walking sequence by successive calculation of walking and running sequences.

5.2 Horizontal planning

In horizontal trajectory generation, both the walking and the running sequence are interdependent from the first to the last step. As described in Chapter 3 and Chapter 4, in each gait sequence there is a CoM position initial condition and a DCM or CoM velocity terminal condition for walking or running, respectively. This implies that the CoM position trajectory has to be solved by a forward recursion and the DCM or CoM velocity trajectory by a backward recursion. For a single walking sequence, the DCM dynamics are decoupled from the CoM dynamics, but by converting between DCM position and CoM velocity in each gait transition the coupling is reestablished. The solution is to set up a global matrix equation for multiple gait sequences that takes into account all boundary conditions and solves for the waypoints of both trajectories simultaneously.

Table 5.3 shows the horizontal components of the first and final DCM position and CoM position, velocity and acceleration waypoints for three consecutive gait sequences with

gait sequence	...	walking j		running j		walking $j+1$...
transition		W2R		R2W				
waypoint		start	end	start	end	start	end	
DCM position	...	$\xi_{\chi,1,w_j}$	$\xi_{\chi,N,w_j} \stackrel{\textcircled{1}}{=}$	$\xi_{\chi,\text{TD},1,r_j}$	$\xi_{\chi,\text{TO},N,r_j} \stackrel{\textcircled{2}}{=}$	$\xi_{\chi,1,w_{j+1}}$	$\xi_{\chi,N,w_{j+1}}$...
CoM position	...	χ_{1,w_j}	$\chi_{N,w_j} \stackrel{\textcircled{3}}{=}$	$\chi_{\text{TD},1,r_j}$	$\chi_{\text{TO},N,r_j} \stackrel{\textcircled{4}}{=}$	$\chi_{1,w_{j+1}}$	$\chi_{N,w_{j+1}}$...
CoM velocity	...	$\dot{\chi}_{1,w_j}$	$\dot{\chi}_{N,w_j} \stackrel{\textcircled{5}}{=}$	$\dot{\chi}_{\text{TD},1,r_j}$	$\dot{\chi}_{\text{TO},N,r_j} \stackrel{\textcircled{6}}{=}$	$\dot{\chi}_{1,w_{j+1}}$	$\dot{\chi}_{N,w_{j+1}}$...
CoM acceleration	...	$\ddot{\chi}_{1,w_j}$	$\ddot{\chi}_{N,w_j} \stackrel{\textcircled{7}}{=}$	$\ddot{\chi}_{\text{TD},1,r_j}$	$\ddot{\chi}_{\text{TO},N,r_j} \stackrel{\textcircled{8}}{=}$	$\ddot{\chi}_{1,w_{j+1}}$	$\ddot{\chi}_{N,w_{j+1}}$...
boundary condition fulfilled by walking:				boundary condition fulfilled by running:				

Table 5.3: Horizontal transition boundary conditions for three consecutive gait phases.

the corresponding transition boundary conditions highlighted. The position boundary condition in the W2R transition can be readily specified as the corresponding quantities exist for both gaits:

$$\textcircled{3}: \quad \chi_{\text{TD},1,r_j} = \chi_{N,w_j}. \quad (5.15)$$

Since the first TD position $\chi_{\text{TD},1,r_j}$ in the running sequence is mapped only up to the last TD position χ_{TD,N,r_j} , the final TO position

$$\chi_{\text{TO},N,r_j} = \underbrace{t_{\chi}^T(T_{s,N})}_{t_N} P_{\chi,N} \quad (5.16)$$

must be introduced by using the polynomial encoding from (4.16) as additional variable in the last stance phase to satisfy boundary condition

$$\textcircled{4}: \quad \chi_{1,w_{j+1}} = \chi_{\text{TO},N,r_j}. \quad (5.17)$$

Similarly the additional variable

$$\dot{\chi}_{\text{TD},1,r_j} = \underbrace{t_{\dot{\chi}}^T(0)}_{b_1} P_{\chi,1} \quad (5.18)$$

is required by the DCM boundary condition in the W2R transition which can be computed with the definition of the DCM (see (3.1)) as

$$\textcircled{1}: \quad \xi_{\chi,N,w_j} = \xi_{\chi,\text{TD},1,r_j} = \chi_{\text{TD},1,r_j} + b \dot{\chi}_{\text{TD},1,r_j}. \quad (5.19)$$

Here, with $\textcircled{1}$ and $\textcircled{3}$, the DCM and CoM position boundary condition are fulfilled, thus in this case the CoM velocity boundary condition $\textcircled{5}$ is implicitly satisfied. The R2W velocity boundary condition is also formulated via DCM definition as

$$\textcircled{6}: \quad \dot{\chi}_{1,w_{j+1}} = \dot{\chi}_{\text{TO},N,r_j} = \frac{1}{b} (\xi_{\chi,1,w_{j+1}} - \chi_{1,w_{j+1}}). \quad (5.20)$$

In the R2W transition, the DCM boundary condition $\textcircled{2}$ is implicitly satisfied by fulfilling boundary conditions $\textcircled{4}$ and $\textcircled{6}$. With $\ddot{\chi} = \mathbf{F}_{\chi}/m$ and the first and second row of (3.6) the acceleration boundary conditions can be written in terms of the CoM position and VRP waypoint as

$$\textcircled{7}: \quad \ddot{\chi}_{\text{TD},1,r_j} = \ddot{\chi}_{N,w_j} = \frac{1}{b^2} (\chi_{N,w_j} - \mathbf{V}_{\chi,N,w_j}) \quad \text{and} \quad (5.21)$$

$$\textcircled{8}: \quad \ddot{\chi}_{\text{TO},N,r_j} = \ddot{\chi}_{1,w_{j+1}} = \frac{1}{b^2} (\chi_{1,w_{j+1}} - \mathbf{V}_{\chi,1,w_{j+1}}). \quad (5.22)$$

5.2.1 Running matrix assembly

stance phase	stance phase 1	stance phase $i = 1..N-1$	stance phase N
boundary condition matrix	$\tau_{\chi,1} = \begin{bmatrix} \chi_{TD,1} \\ \dot{\chi}_{TD,1} \\ \dot{\chi}_{TO,1} \\ \ddot{\chi}_{TD,1} \end{bmatrix}$	$\tau_{\chi,i} = \begin{bmatrix} \chi_{TD,i} \\ \dot{\chi}_{TO,i} \end{bmatrix}$	$\tau_{\chi,N} = \begin{bmatrix} \chi_{TD,N} \\ \chi_{TO,N} \\ \dot{\chi}_{TD,N} \\ \ddot{\chi}_{TO,N} \end{bmatrix}$

Table 5.4: Horizontal boundary condition matrix $\tau_{\chi,i}$ for different running stance phases.

Table 5.4 shows the boundary condition matrix for different stance phases with additional variables introduced in the first and last stance phases to satisfy the transition boundary conditions towards walking. The first TD velocity $\dot{\chi}_{TD,1}$ and last TO position $\chi_{TO,N}$ are defined in (5.18) and (5.16), respectively. The first TD acceleration $\ddot{\chi}_{TD,1}$ and last TO acceleration $\ddot{\chi}_{TO,N}$ are already defined in Chapter 4. However, unlike in (4.17), these are not zero on the boundary between running and walking but must satisfy the corresponding transition boundary conditions (5.21) and (5.22). With the additional variables in the transition stance phases, the mapping from the polynomial parameters to the boundary conditions (see (4.30)) has to be modified:

$$\underbrace{\begin{bmatrix} \tau_{\chi,1} \\ \vdots \\ \tau_{\chi,i} \\ \vdots \\ \tau_{\chi,N} \end{bmatrix}}_{\Sigma_{wp,r}} = \underbrace{\begin{bmatrix} S_1 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & & \vdots \\ \vdots & \ddots & S_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & S_N \end{bmatrix}}_{S \in \mathbb{R}^{N_c \times 6N}} \underbrace{\begin{bmatrix} P_{\chi,1} \\ \vdots \\ P_{\chi,i} \\ \vdots \\ P_{\chi,N} \end{bmatrix}}_{P_\chi} + \underbrace{\begin{bmatrix} I_1 \\ \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{I_{IC}} \tau_{\chi,IC} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{0} \\ I_N \end{bmatrix}}_{I_{TC}} \tau_{\chi,TC},$$

$$\text{with } I_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad I_N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

(5.23)

Here, $\Sigma_{wp,r} \in \mathbb{R}^{N_c \times 2}$ is the running waypoint matrix consisting of the boundary condition matrices $\tau_{\chi,i}$ and $N_c = 2N + 4$ is the number of combined waypoints (two for each stance phase and two additional waypoints each for the first and last stance phase, see Table 5.4). The definition of S_i in matrix S depends on the particular stand phase and is defined with Eqs. (4.28), (4.29), (5.16) and (5.18) in Table 5.5. $\tau_{\chi,IC}$ and $\tau_{\chi,TC}$ are arbitrary initial and terminal constraint matrices, respectively, which are mapped to the waypoint matrix using I_{IC} and I_{TC} . The selections matrices I_1 and I_2 select only the first and last row and the third and last row of $\tau_{\chi,IC}$ and $\tau_{\chi,TC}$, respectively, i.e. the boundary conditions that are specified for running in Table 5.3.

stance phase	stance phase 1	stance phase $i = 1..N-1$	stance phase N
	$S_1 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_1 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$	$S_i = \begin{bmatrix} \mathbf{f}_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}_{i+1} \end{bmatrix}$	$S_N = \begin{bmatrix} \mathbf{f}_{N-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{l}_N \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$

Table 5.5: Polynomial coefficient mapping matrix S_i for different running stance phases.

Since the waypoint matrix $\Sigma_{wp,r}$ has changed, matrix equation (4.25) must also be adjusted. The mapping of the additional acceleration variables to the polynomial parameter matrix is already contained in (4.24), which is written out in full to explicitly account for these quantities:

$$P_{\chi,i} = \left[\underbrace{\left(I - e_{\chi,i}^{\oplus} e_{\chi,i}^T \right) \Omega_{\chi,i} \Pi_{\chi,i}}_{Q_{\chi,i}} \underbrace{\frac{\Omega_{\chi,i} \pi_{\chi,i}}{e_{\chi,i}^T \Omega_{\chi,i} \pi_{\chi,i}}}_{e_{\chi,i}^{\oplus}} \right] \begin{bmatrix} \chi_{TD,i} \\ \dot{\chi}_{TO,i} \\ \ddot{\chi}_{TD,i} \\ \ddot{\chi}_{TO,i} \\ \chi_{foot,i} \end{bmatrix}. \quad (5.24)$$

Here, the matrix $\Pi_{\chi,i}$ combines the first to the fourth column vector of $B_{\chi,i}^+$ in (4.23) and

$$Q_{\chi,i} = \begin{bmatrix} \mathbf{q}_{i,c_1} & \mathbf{q}_{i,c_2} & \mathbf{q}_{i,c_3} & \mathbf{q}_{i,c_4} \end{bmatrix} \quad (5.25)$$

is the boundary condition mapping matrix consisting of four columns \mathbf{q}_{i,c_n} . The mappings of the first TD acceleration $\ddot{\chi}_{TD,1}$ and last TO acceleration $\ddot{\chi}_{TO,N}$ are stated in the third and fourth row of (5.24), respectively. These are the only two acceleration waypoints that are non-zero during the running sequence, so it is not necessary to explicitly calculate the acceleration waypoints in each stance phase. Therefore, the TO and TD acceleration boundary condition in the first and last stance phase, respectively, and both acceleration boundary conditions for the stance phases $2..N-1$ are implicitly fulfilled as in (4.24). The additional variables $\dot{\chi}_{TD,1}$ and $\chi_{TO,N}$ have no influence on the polynomial parameter matrix and are therefore mapped with the zero vector. With Eqs. (5.24) and (5.25), the relation between the boundary condition matrix $\tau_{\chi,i}$ and the polynomial parameter matrix $P_{\chi,i}$ for different stand phases is stated in Table 5.6.

Accordingly, equation (5.24), considering the additional variables specified in Table 5.4, can be written in matrix form as

$$\underbrace{\begin{bmatrix} \mathbf{P}_{\chi,1} \\ \vdots \\ \mathbf{P}_{\chi,i} \\ \vdots \\ \mathbf{P}_{\chi,N} \end{bmatrix}}_{\mathbf{P}_{\chi}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \mathbf{A}_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{A}_N \end{bmatrix}}_{\mathbf{A} \in \mathbb{R}^{6N \times N_c}} \underbrace{\begin{bmatrix} \tau_{\chi,1} \\ \vdots \\ \tau_{\chi,i} \\ \vdots \\ \tau_{\chi,N} \end{bmatrix}}_{\boldsymbol{\Sigma}_{wp,r}} + \underbrace{\begin{bmatrix} \mathbf{e}_{\chi,1}^{\oplus} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \mathbf{e}_{\chi,i}^{\oplus} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{e}_{\chi,N}^{\oplus} \end{bmatrix}}_{\mathbf{E}_{\chi} \in \mathbb{R}^{6N \times N_f}} \underbrace{\begin{bmatrix} \chi_{\text{foot},1} \\ \vdots \\ \chi_{\text{foot},i} \\ \vdots \\ \chi_{\text{foot},N_f} \end{bmatrix}}_{\mathbf{X}_{\text{foot}}}, \quad (5.26)$$

where \mathbf{A} is the constraint mapping matrix, \mathbf{E}_{χ} is the footstep mapping matrix and N_f

stance phase	stance phase 1	stance phase $i = 1..N-1$	stance phase N
$\mathbf{A}_1 = \begin{bmatrix} \mathbf{q}_{1,c_1} & \mathbf{q}_{1,c_2} & \mathbf{0} & \mathbf{q}_{1,c_3} \end{bmatrix}$	$\mathbf{A}_i = \begin{bmatrix} \mathbf{q}_{i,c_1} & \mathbf{q}_{i,c_2} \end{bmatrix}$	$\mathbf{A}_N = \begin{bmatrix} \mathbf{q}_{N,c_1} & \mathbf{0} & \mathbf{q}_{N,c_2} & \mathbf{q}_{N,c_4} \end{bmatrix}$	

Table 5.6: Constraint mapping matrix \mathbf{A}_i for different running stance phases.

is the number of footsteps. The definition of \mathbf{A}_i in \mathbf{A} again depends on the stance phase and is specified in Table 5.6. Inserting (5.26) into (5.23) and solving for the waypoint vector $\boldsymbol{\Sigma}_{wp,r}$ yields

$$\boldsymbol{\Sigma}_{wp,r} = \underbrace{(\mathbf{I} - \mathbf{S} \mathbf{A})^{-1} \mathbf{I}_{IC}}_{\mathbf{C}} \tau_{\chi,IC} + \underbrace{(\mathbf{I} - \mathbf{S} \mathbf{A})^{-1} \mathbf{I}_{TC}}_{\mathbf{D}} \tau_{\chi,TC} + \underbrace{(\mathbf{I} - \mathbf{S} \mathbf{A})^{-1} \mathbf{S} \mathbf{E}_{\chi}}_{\mathbf{F}} \mathbf{X}_{\text{foot}}, \quad (5.27)$$

in which

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{N_c} \end{bmatrix} \in \mathbb{R}^{N_c \times 4}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N_c} \end{bmatrix} \in \mathbb{R}^{N_c \times 4} \quad \text{and} \quad (5.28)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{N_c} \end{bmatrix} \in \mathbb{R}^{N_c \times N_f}.$$

Here, the vectors \mathbf{c}_j , \mathbf{d}_j and \mathbf{f}_j are the rows of the matrices \mathbf{C} , \mathbf{D} and \mathbf{F} , respectively. As shown in equation (5.23), $\chi_{\text{TD},1}$ and $\ddot{\chi}_{\text{TD},1}$ are given as initial conditions in the

first stance phase and $\dot{\chi}_{\text{TO},N}$ and $\ddot{\chi}_{\text{TO},N}$ are specified as terminal conditions in the last stance phase. Since these values are not influenced by any other waypoints in the running sequence, the corresponding row vectors of the matrices \mathbf{C} , \mathbf{D} and \mathbf{F} result as follows

$$\begin{aligned} \mathbf{c}_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{c}_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{d}_{N_c-1} &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{d}_{N_c} &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{f}_1 &= \mathbf{f}_4 = \mathbf{f}_{N_c-1} = \mathbf{f}_{N_c} = \mathbf{0}_{1 \times N_f}. \end{aligned} \quad (5.29)$$

By replacing the corresponding rows 1, 4, N_c-1 and N_c in (5.27) with the boundary conditions ③ (5.15) and ⑥ to ⑧ (Eqs. (5.20) to (5.22)), a recursive matrix equation can be assembled which maps the initial and terminal constraints to the remaining running waypoints:

$$\underbrace{\begin{bmatrix} \chi_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TO},1,r_j} \\ \ddot{\chi}_{\text{TD},1,r_j} \\ \vdots \\ \chi_{\text{TD},N,r_j} \\ \chi_{\text{TO},N,r_j} \\ \dot{\chi}_{\text{TO},N,r_j} \\ \ddot{\chi}_{\text{TO},N,r_j} \end{bmatrix}}_{\Sigma_{wp,r,j}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{c}_2 & \mathbf{d}_2 \\ \mathbf{c}_3 & \mathbf{d}_3 \\ \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \vdots \\ \mathbf{c}_{N_c-3} & \mathbf{d}_{N_c-3} \\ \mathbf{c}_{N_c-2} & \mathbf{d}_{N_c-2} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{R}_j \in \mathbb{R}^{N_c \times N_c}} \underbrace{\begin{bmatrix} \chi_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TO},1,r_j} \\ \ddot{\chi}_{\text{TD},1,r_j} \\ \vdots \\ \chi_{\text{TD},N,r_j} \\ \chi_{\text{TO},N,r_j} \\ \dot{\chi}_{\text{TO},N,r_j} \\ \ddot{\chi}_{\text{TO},N,r_j} \end{bmatrix}}_{\Sigma_{wp,r,j}} + \underbrace{\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \vdots \\ \mathbf{f}_{N_c-3} \\ \mathbf{f}_{N_c-2} \\ \mathbf{f}_{N_c-1} \\ \mathbf{f}_{N_c} \end{bmatrix}}_{\mathbf{F}_j \in \mathbb{R}^{N_c \times N_f}} \underbrace{\begin{bmatrix} \chi_{\text{foot},1,r_j} \\ \chi_{\text{foot},2,r_j} \\ \vdots \\ \chi_{\text{foot},N-1,r_j} \\ \chi_{\text{foot},N,r_j} \end{bmatrix}}_{\mathbf{X}_{\text{foot},j}} + \underbrace{\begin{bmatrix} \chi_{N,w_j} \\ \mathbf{0} \\ \mathbf{0} \\ \frac{1}{b^2} (\chi_{N,w_j} - \mathbf{V}_{\chi,N,w_j}) \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \frac{1}{b} (\xi_{\chi,1,w_{j+1}} - \chi_{1,w_{j+1}}) \\ \frac{1}{b^2} (\chi_{1,w_{j+1}} - \mathbf{V}_{\chi,1,w_{j+1}}) \end{bmatrix}}_{\mathbf{X}_{\text{BC},r,j}}. \quad (5.30)$$

Here, j denotes the index of the considered running sequence and $\Sigma_{wp,r,j}$, \mathbf{R}_j , \mathbf{F}_j , $\mathbf{X}_{\text{foot},j}$ and $\mathbf{X}_{\text{BC},r,j}$ denote the corresponding running waypoint matrix, running constraint mapping matrix, footstep mapping matrix, footstep matrix and running boundary condition

matrix, respectively. Extending the running waypoint matrix $\Sigma_{wp,r,j}$ to the right side of R_j in (5.30) from above by the last waypoint of the previous walking sequence w_j and from below by the first two waypoints of the next walking sequence w_{j+1} , part of the boundary conditions in $\chi_{BC,r,j}$ can be integrated into an augmented running constraint matrix R_j^\otimes . The remaining part, concerning VRP waypoints, can be integrated into an augmented footstep mapping matrix F_j^\otimes by extending the footstep matrix $X_{\text{foot},j}$ from above by the last VRP waypoint of the previous walking sequence w_j and from below by the first VRP waypoint of the next walking sequence w_{j+1} . Thus (5.30) turns into

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \chi_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TO},1,r_j} \\ \ddot{\chi}_{\text{TD},1,r_j} \\ \vdots \\ \chi_{\text{TD},N,r_j} \\ \chi_{\text{TO},N,r_j} \\ \dot{\chi}_{\text{TO},N,r_j} \\ \ddot{\chi}_{\text{TO},N,r_j} \end{bmatrix}}_{\Sigma_{wp,r,j} \in \mathbb{R}^{N_c \times 2}} &= \underbrace{\begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{c}_2 & \mathbf{d}_2 & 0 & 0 \\ 0 & \mathbf{c}_3 & \mathbf{d}_3 & 0 & 0 \\ 1/b^2 & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \vdots & \vdots & \mathbf{0} & \vdots & \vdots \\ 0 & \mathbf{c}_{N_c-3} & \mathbf{d}_{N_c-3} & 0 & 0 \\ 0 & \mathbf{c}_{N_c-2} & \mathbf{d}_{N_c-2} & 0 & 0 \\ 0 & \mathbf{0} & \mathbf{0} & 1/b & -1/b \\ 0 & \mathbf{0} & \mathbf{0} & 0 & 1/b^2 \end{bmatrix}}_{R_j^\otimes \in \mathbb{R}^{N_c \times (N_c+3)}} \underbrace{\begin{bmatrix} \chi_{N,w_j} \\ \chi_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TD},1,r_j} \\ \dot{\chi}_{\text{TO},1,r_j} \\ \ddot{\chi}_{\text{TD},1,r_j} \\ \vdots \\ \chi_{\text{TD},N,r_j} \\ \chi_{\text{TO},N,r_j} \\ \dot{\chi}_{\text{TO},N,r_j} \\ \ddot{\chi}_{\text{TO},N,r_j} \\ \xi_{\chi,1,w_{j+1}} \\ \chi_{1,w_{j+1}} \end{bmatrix}}_{\Sigma_{wp,r,j}^\otimes \in \mathbb{R}^{(N_c+3) \times 2}} + \\
 &+ \underbrace{\begin{bmatrix} 0 & \mathbf{0} & 0 \\ -1/b^2 & \mathbf{f}_2 & 0 \\ 0 & \mathbf{f}_3 & 0 \\ 0 & \mathbf{0} & 0 \\ \vdots & \vdots & \vdots \\ 0 & \mathbf{f}_{N_c-3} & 0 \\ 0 & \mathbf{f}_{N_c-2} & 0 \\ 0 & \mathbf{0} & 0 \\ 0 & \mathbf{0} & -1/b^2 \end{bmatrix}}_{F_j^\otimes \in \mathbb{R}^{N_c \times (N_f+2)}} \underbrace{\begin{bmatrix} v_{\chi,N,w_j} \\ \chi_{\text{foot},1,r_j} \\ \chi_{\text{foot},2,r_j} \\ \vdots \\ \chi_{\text{foot},N-1,r_j} \\ \chi_{\text{foot},N,r_j} \\ v_{\chi,1,w_{j+1}} \end{bmatrix}}_{X_{\text{foot},j}^\otimes \in \mathbb{R}^{(N_f+2) \times 2}}, \tag{5.31}
 \end{aligned}$$

where \square^\otimes denotes the corresponding augmented quantity of any matrix \square and all nonzero entries of the matrices R_j^\otimes and F_j^\otimes are highlighted. At this point, equation (5.31) is prepared to be embedded in a global matrix equation for multiple walking and running sequences. Thus by stacking up the waypoints of alternating walking and running sequences, a global waypoint matrix can be assembled and solved for, resulting in the

waypoints of the entire trajectory for multiple walking and running sequences from a single matrix calculation. This requires computing the constraint and target mapping matrix for walking as part of the assembly, which is discussed in the next section.

5.2.2 Walking matrix assembly

In Chapter 3, the DCM terminal constraint $\xi_{wp,N} = \mathbf{v}_{wp,N}$ is used such that the DCM comes to a stop at the end of the walking sequence (see (3.25)). Here, however, due to the transition boundary condition (5.19), the horizontal components of the DCM are specified by the subsequent running sequence. To separate the mapping of the horizontal components of the terminal DCM waypoint $\xi_{\chi,TC}$ to the horizontal DCM waypoint matrix $\xi_{\chi,wp}$, equation (3.29) can be reformulated as

$$\xi_{\chi,wp} = \underbrace{(I - A_\xi)^{-1} A_{\alpha\beta}}_{\tilde{A}_v} V_{\chi,wp} + \underbrace{(I - A_\xi)^{-1} \mathbf{a}_{TC}}_{\mathbf{a}_\xi} \xi_{\chi,TC}, \quad (5.32)$$

with

$$A_{\alpha\beta} = \begin{bmatrix} \alpha_{\xi,0,1} & \beta_{\xi,0,1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \alpha_{0,n_\varphi} & \beta_{0,n_\varphi} \\ 0 & \dots & \dots & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{a}_{TC} = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}. \quad (5.33)$$

Matrix $A_{\alpha\beta} \in \mathbb{R}^{N \times N}$ is equivalent to matrix A_v in (3.28) except for the last entry in the lower right corner, which previously represented the terminal constraint. Any DCM terminal waypoint $\xi_{\chi,TC}$ is mapped by the vector

$$\mathbf{a}_\xi = \begin{bmatrix} a_{\xi,1} \\ \vdots \\ a_{\xi,N-1} \\ 1 \end{bmatrix} \in \mathbb{R}^N \quad (5.34)$$

in (5.32) to the DCM waypoint matrix. The matrix

$$\tilde{A}_v = \begin{bmatrix} \mathbf{a}_{v,1} \\ \vdots \\ \mathbf{a}_{v,N-1} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N \times N} \quad (5.35)$$

in (5.32), where $\mathbf{a}_{v,i}$ denote single rows, relates VRP waypoints to DCM waypoints.

Similarly, the first horizontal CoM position is specified by the boundary condition (5.17). Therefore, the mapping of the horizontal components of the first CoM waypoint to the CoM waypoint vector χ_{wp} needs to be separated by modifying (3.35) to

$$\chi_{wp} = (I - B_x)^{-1} (B_\xi \xi_{wp} + B_{\alpha\beta} \mathbf{v}_{wp}) + \underbrace{(I - B_x)^{-1} \mathbf{b}_{IC}}_{\mathbf{b}_x} \chi_{IC}. \quad (5.36)$$

This equation is the equivalent of (5.9) for the horizontal direction. Matrix $\mathbf{B}_{\alpha\beta} \in \mathbb{R}^{N \times N}$ and vector \mathbf{b}_{IC} are defined in (5.10). Inserting (5.32) into (5.36) isolates the mapping from the DCM terminal point to the CoM start point:

$$\begin{aligned} \chi_{wp} = & \underbrace{(\mathbf{I} - \mathbf{B}_x)^{-1} (\mathbf{B}_\xi \tilde{\mathbf{A}}_v + \mathbf{B}_{\alpha\beta})}_{\tilde{\mathbf{B}}_v} \mathbf{V}_{\chi,wp} + \underbrace{(\mathbf{I} - \mathbf{B}_x)^{-1} \mathbf{B}_\xi \mathbf{a}_\xi}_{\mathbf{b}_\xi} \boldsymbol{\xi}_{\chi,TC} + \\ & + \underbrace{(\mathbf{I} - \mathbf{B}_x)^{-1} \mathbf{b}_{IC}}_{\mathbf{b}_x} \chi_{IC}, \end{aligned} \quad (5.37)$$

The mapping of the VRP waypoints $\mathbf{V}_{\chi,wp}$, an arbitrary DCM terminal waypoint $\boldsymbol{\xi}_{\chi,TC}$ and CoM initial waypoint χ_{IC} to the CoM waypoint matrix is represented in (5.37) by

$$\tilde{\mathbf{B}}_v = \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_{v,2} \\ \vdots \\ \mathbf{b}_{v,N} \end{bmatrix} \in \mathbb{R}^{N \times N}, \quad \mathbf{b}_\xi = \begin{bmatrix} 0 \\ b_{\xi,2} \\ \vdots \\ b_{\xi,N} \end{bmatrix} \in \mathbb{R}^N \text{ and } \mathbf{b}_x = \begin{bmatrix} 1 \\ b_{x,2} \\ \vdots \\ b_{x,N} \end{bmatrix} \in \mathbb{R}^N, \quad (5.38)$$

respectively. Here, $\mathbf{b}_{v,i}$ denote single rows of $\tilde{\mathbf{B}}_v$. With these preparations, equations (5.32) and (5.37) can be combined row by row into a single matrix equation:

$$\underbrace{\begin{bmatrix} \boldsymbol{\xi}_{\chi,1} \\ \chi_1 \\ \boldsymbol{\xi}_{\chi,2} \\ \chi_2 \\ \vdots \\ \boldsymbol{\xi}_{\chi,N-1} \\ \chi_{N-1} \\ \boldsymbol{\xi}_{\chi,N} \\ \chi_N \end{bmatrix}}_{\boldsymbol{\Sigma}_{wp,w}} = \underbrace{\begin{bmatrix} \mathbf{a}_{v,1} \\ \mathbf{0} \\ \mathbf{a}_{v,2} \\ \mathbf{b}_{v,2} \\ \vdots \\ \mathbf{a}_{v,N-1} \\ \mathbf{b}_{v,N-1} \\ \mathbf{0} \\ \mathbf{b}_{v,N} \end{bmatrix}}_{\mathbf{Y}} \underbrace{\begin{bmatrix} \mathbf{v}_{\chi,1} \\ \mathbf{v}_{\chi,2} \\ \vdots \\ \mathbf{v}_{\chi,N-1} \\ \mathbf{v}_{\chi,N} \end{bmatrix}}_{\mathbf{V}_\chi} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ b_{x,2} \\ \vdots \\ 0 \\ b_{x,N-1} \\ 0 \\ b_{x,N} \end{bmatrix}}_{\mathbf{c}_{x,b}} \chi_{IC} + \underbrace{\begin{bmatrix} a_{\xi,1} \\ 0 \\ a_{\xi,2} \\ b_{\xi,2} \\ \vdots \\ a_{\xi,N-1} \\ b_{\xi,N-1} \\ 1 \\ b_{\xi,N} \end{bmatrix}}_{\mathbf{c}_{\xi,a,b}} \boldsymbol{\xi}_{\chi,TC} \quad (5.39)$$

Here, matrix $\boldsymbol{\Sigma}_{wp,w}$ summarizes both DCM and CoM position waypoints in a single matrix, \mathbf{Y} is the combined target mapping matrix and $\mathbf{c}_{x,b}$ and $\mathbf{c}_{\xi,a,b}$ denote the combined initial condition and terminal condition mapping vector, respectively. Similar to the previous Section 5.2.1 for running, replacing row 2 and N_c-1 with the CoM position

boundary condition ④ (5.17) and the DCM position boundary condition ① (5.19) and writing (5.39) as a recursive matrix equation results in:

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \xi_{\chi,1,w_j} \\ \chi_{1,w_j} \\ \xi_{\chi,2,w_j} \\ \chi_{2,w_j} \\ \vdots \\ \xi_{\chi,N-1,w_j} \\ \chi_{N-1,w_j} \\ \xi_{\chi,N,w_j} \\ \chi_{N,w_j} \end{bmatrix}}_{\Sigma_{wp,w,j}} &= \underbrace{\begin{bmatrix} 0 & 0 & a_{\xi,1} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a_{\xi,2} & 0 \\ 0 & b_{x,2} & b_{\xi,2} & 0 \\ \vdots & \vdots & \mathbf{0} & \vdots \\ 0 & 0 & a_{\xi,N-1} & 0 \\ 0 & b_{x,N-1} & b_{\xi,N-1} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & b_{x,N} & b_{\xi,N} & 0 \end{bmatrix}}_{\mathbf{W}_j \in \mathbb{R}^{N_c \times N_c}} \underbrace{\begin{bmatrix} \xi_{\chi,1,w_j} \\ \chi_{1,w_j} \\ \xi_{\chi,2,w_j} \\ \chi_{2,w_j} \\ \vdots \\ \xi_{\chi,N-1,w_j} \\ \chi_{N-1,w_j} \\ \xi_{\chi,N,w_j} \\ \chi_{N,w_j} \end{bmatrix}}_{\Sigma_{wp,w,j}} + \\
 &+ \underbrace{\begin{bmatrix} a_{v,1} \\ \mathbf{0} \\ a_{v,2} \\ b_{v,2} \\ \vdots \\ a_{v,N-1} \\ b_{v,N-1} \\ \mathbf{0} \\ b_{v,N} \end{bmatrix}}_{\mathbf{Y}_j \in \mathbb{R}^{N_c \times N_v}} \underbrace{\begin{bmatrix} v_{\chi,1,w_j} \\ v_{\chi,2,w_j} \\ \vdots \\ v_{\chi,N-1,w_j} \\ v_{\chi,N,w_j} \end{bmatrix}}_{\mathbf{V}_{\chi,j}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \chi_{\text{TO},N,r_{j-1}} \\ \mathbf{0} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{0} \\ \chi_{\text{TD},1,r_j} + b \dot{\chi}_{\text{TD},1,r_j} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{X}_{\text{BC},w,j}}, \quad (5.40)
 \end{aligned}$$

where j denotes the index of the considered walking sequence and $\Sigma_{wp,w,j}$, \mathbf{W}_j , \mathbf{Y}_j , $\mathbf{V}_{\chi,j}$ and $\mathbf{X}_{\text{BC},w,j}$ denote the corresponding walking waypoint matrix, walking constraint mapping matrix, VRP mapping matrix, VRP matrix and walking boundary condition matrix, respectively. By extending the walking waypoint matrix $\Sigma_{wp,w,j}$ to the right side of \mathbf{W}_j in (5.40) with the last three waypoints of the previous running sequence r_{j-1} from the top and with the first two waypoints of the next running sequence r_j from the

bottom, the calculation of the walking waypoints can also be expressed by a mapping of only two matrices:

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \xi_{\chi,1,w_j} \\ \chi_{1,w_j} \\ \xi_{\chi,2,w_j} \\ \chi_{2,w_j} \\ \vdots \\ \xi_{\chi,N-1,w_j} \\ \chi_{N-1,w_j} \\ \xi_{\chi,N,w_j} \\ \chi_{N,w_j} \end{bmatrix}}_{\Sigma_{wp,w,j} \in \mathbb{R}^{N_c \times 2}} &= \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & a_{\xi,1} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{\xi,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{x,2} & b_{\xi,2} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & a_{\xi,N-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{x,N-1} & b_{\xi,N-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & b \\ 0 & 0 & 0 & 0 & b_{x,N} & b_{\xi,N} & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{W}_j^{\otimes} \in \mathbb{R}^{N_c \times (N_c+5)}} \underbrace{\begin{bmatrix} \chi_{TO,N,r_{j-1}} \\ \dot{\chi}_{TO,N,r_{j-1}} \\ \ddot{\chi}_{TO,N,r_{j-1}} \\ \xi_{\chi,1,w_j} \\ \chi_{1,w_j} \\ \xi_{\chi,2,w_j} \\ \chi_{2,w_j} \\ \vdots \\ \xi_{\chi,N-1,w_j} \\ \chi_{N-1,w_j} \\ \xi_{\chi,N,w_j} \\ \chi_{N,w_j} \\ \chi_{TD,1,r_j} \\ \dot{\chi}_{TD,1,r_j} \end{bmatrix}}_{\Sigma_{wp,w,j}^{\otimes} \in \mathbb{R}^{(N_c+5) \times 2}} + \\
 &+ \underbrace{\begin{bmatrix} a_{v,1} \\ 0 \\ a_{v,2} \\ b_{v,2} \\ \vdots \\ a_{v,N-1} \\ b_{v,N-1} \\ 0 \\ b_{v,N} \end{bmatrix}}_{\mathbf{Y}_j \in \mathbb{R}^{N_c \times N_v}} \underbrace{\begin{bmatrix} v_{\chi,1,w_j} \\ v_{\chi,2,w_j} \\ \vdots \\ v_{\chi,N-1,w_j} \\ v_{\chi,N,w_j} \end{bmatrix}}_{\mathbf{V}_{\chi,j} \in \mathbb{R}^{N_v \times 2}}. \tag{5.41}
 \end{aligned}$$

Again \square^{\otimes} denotes the respective augmented quantity of any matrix \square and all nonzero entries of the matrices \mathbf{W}_j^{\otimes} and \mathbf{Y}_j are highlighted. This completes the preparations for the assembly of the global matrix calculation in the next section.

5.2.3 Global matrix equation assembly

In this section, a global equation for multiple walking and running sequences is constructed by stacking the waypoints of the alternating gait sequences on top of each

other. Similarly, the constraint mapping matrices \mathbf{R}_j^\otimes and \mathbf{W}_j^\otimes and the target mapping matrices \mathbf{F}_j^\otimes and \mathbf{Y}_j are arranged in a diagonal-like structure to obtain a global constraint mapping and target mapping matrix for multiple gait sequences. This global equation can be solved to obtain all waypoints of the whole trajectory in a single calculation. Before this, however, the first and last walking sequences must be adapted, since special boundary conditions occur in the transition between standing and walking. For the first and last walking sequence, the initial constraint for the CoM position

$$\boldsymbol{\chi}_{1,w_1} = \mathbf{v}_{\chi,1,w_1} \quad (5.42)$$

and the terminal constraint for the DCM position

$$\boldsymbol{\xi}_{\chi,N,w_N} = \mathbf{v}_{\chi,N,w_N} \quad (5.43)$$

are used, respectively. The initial constraint (5.42) can be incorporated by omitting the first three rows in (5.41) and defining \mathbf{Y}_1 as:

$$\mathbf{Y}_1 = \begin{bmatrix} \mathbf{a}_{v,1} \\ \mathbf{b}_{v,1} \\ \mathbf{a}_{v,2} \\ \mathbf{b}_{v,2} \\ \vdots \\ \mathbf{a}_{v,N-1} \\ \mathbf{b}_{v,N-1} \\ \mathbf{0} \\ \mathbf{b}_{v,N} \end{bmatrix} \quad \text{with } \mathbf{b}_{v,1} = \begin{bmatrix} 1 & \mathbf{0}_{1 \times N-1} \end{bmatrix}. \quad (5.44)$$

Similarly, the terminal constraint (5.43) can be considered by removing the last two rows in (5.41) and choosing \mathbf{Y}_N as:

$$\mathbf{Y}_N = \begin{bmatrix} \mathbf{a}_{v,1} \\ \mathbf{0} \\ \mathbf{a}_{v,2} \\ \mathbf{b}_{v,2} \\ \vdots \\ \mathbf{a}_{v,N-1} \\ \mathbf{b}_{v,N-1} \\ \mathbf{a}_{v,N} \\ \mathbf{b}_{v,N} \end{bmatrix} \quad \text{with } \mathbf{a}_{v,N} = \begin{bmatrix} \mathbf{0}_{1 \times N-1} & 1 \end{bmatrix}. \quad (5.45)$$

The global matrix equation for multiple walking and running sequences can then be assembled considering the above-mentioned variations in the first and last walking sequence by (5.31) and (5.41):

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \Sigma_{wp,w,1} \\ \Sigma_{wp,r,1} \\ \Sigma_{wp,w,2} \\ \vdots \\ \Sigma_{wp,r,N} \\ \Sigma_{wp,w,N} \end{bmatrix}}_{\Sigma_{wp}} &= \underbrace{\begin{bmatrix} \boxed{W_1^\otimes} & & & & \\ & \boxed{R_1^\otimes} & & & \\ & & \boxed{W_2^\otimes} & & \\ & & & \ddots & \\ & & & & \boxed{R_N^\otimes} \\ & & & & & \boxed{W_N^\otimes} \end{bmatrix}}_{G \in \mathbb{R}^{N_{wp} \times N_{wp}}} \underbrace{\begin{bmatrix} \Sigma_{wp,w,1} \\ \Sigma_{wp,r,1} \\ \Sigma_{wp,w,2} \\ \vdots \\ \Sigma_{wp,r,N} \\ \Sigma_{wp,w,N} \end{bmatrix}}_{\Sigma_{wp}} + \\
 &+ \underbrace{\begin{bmatrix} \boxed{Y_1} & & & & \\ & \boxed{F_1^\otimes} & & & \\ & & \boxed{Y_2} & & \\ & & & \ddots & \\ & & & & \boxed{F_N^\otimes} \\ & & & & & \boxed{Y_N} \end{bmatrix}}_{T \in \mathbb{R}^{N_{wp} \times N_t}} \underbrace{\begin{bmatrix} V_{\chi,1} \\ X_{\text{foot},1} \\ V_{\chi,2} \\ \vdots \\ X_{\text{foot},N} \\ V_{\chi,N} \end{bmatrix}}_{X_{\text{target}}}
 \end{aligned} \tag{5.46}$$

Here, $\Sigma_{wp} \in \mathbb{R}^{N_{wp} \times 2}$ is the global matrix of all horizontal waypoints for multiple gait sequences, $X_{\text{target}} \in \mathbb{R}^{N_t \times 2}$ is the global horizontal target waypoint matrix and N_{wp} and N_t denote the number of all horizontal waypoints and target waypoints, respectively. The global constraint mapping matrix G satisfies, by mapping the waypoint matrix onto itself, all constraints of the waypoints between each other and the global target mapping matrix T maps the target waypoints onto the waypoint matrix Σ_{wp} . Solving (5.46) for Σ_{wp} yields

$$\Sigma_{wp} = \underbrace{(I - G)^{-1} T}_{U} X_{\text{target}}. \tag{5.47}$$

Here, $I \in \mathbb{R}^{N_{wp} \times N_{wp}}$ is an identity matrix and the global combined target and constraint mapping matrix $U \in \mathbb{R}^{N_{wp} \times N_t}$ maps the global target waypoint matrix to the global

waypoint matrix considering the waypoint constraints among each other. Equation (5.47) solves the complete horizontal trajectory for multiple gait phases in a single matrix operation.

5.2.4 Structure of the global constraint and target mapping matrix

This section is intended to give an insight into the structure of the global mapping matrices derived in the previous section. Fig. 5.3 shows the structure of the global con-

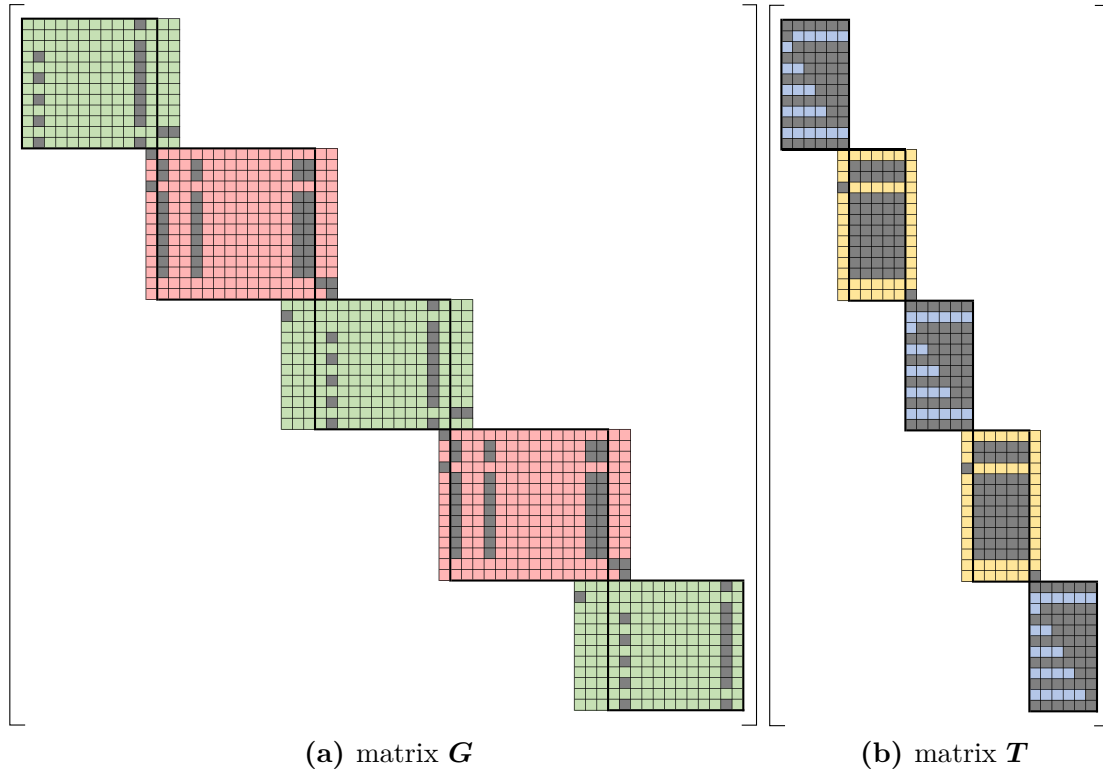


Figure 5.3: Heatmap of global constraint mapping matrix \mathbf{G} and the global target mapping matrix \mathbf{T} in (5.46) for five consecutive gait sequences with three footsteps each consisting of 12 waypoints in each walking phase and 14 waypoints in each running sequence, i.e. 64 waypoints in total. Submatrices according to the coloring scheme in (5.46) with non-zero entries colored in gray.

straint mapping matrix \mathbf{G} and the global target mapping matrix \mathbf{T} for five consecutive gait sequences with three footsteps each. The robot's CoM starts and ends in a steady state, i.e. standing. The individual constraint matrices for a gait sequence are shown in green and red, and the target matrices for walking and running are shown in blue and yellow, respectively. They have the same structure as in equations (5.31) and (5.41). All non-zero entries are colored in gray. The mapping from the waypoints of a sequence to each other is framed in bold in matrix \mathbf{G} and the mapping from the VRP or foot target waypoints to the gait sequence waypoints is framed in bold in matrix \mathbf{T} . Outside the bold framed region, the transition boundary conditions to the neighboring gait sequences are considered. In the running sequence, each waypoint depends on the four boundary conditions, while in the walking sequence the decoupling of the DCM dynamics from the

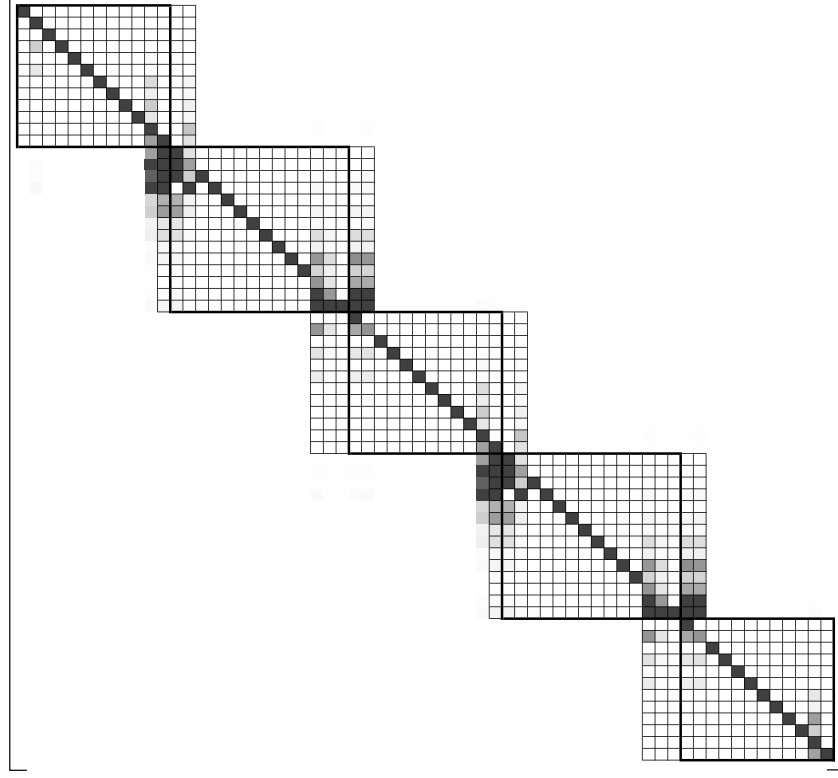


Figure 5.4: Heatmap of the matrix $(\mathbf{I} - \mathbf{G})^{-1}$ in (5.47) with same settings as in Fig. 5.3. The values of each cell \square are displayed on a linear scale from $\square = 0$ (white) to $|\square| \geq 1$ (dark gray).

CoM dynamics is evident, such that only the CoM waypoint depends on both boundary conditions and the DCM waypoint depends only on the DCM terminal condition.

Fig. 5.4 illustrates the structure of the matrix $(\mathbf{I} - \mathbf{G})^{-1}$ of the solution equation (5.47). The gray tone of the individual entries changes linearly from $\square = 0$ (white) to $|\square| \geq 1$ (dark gray), where \square denotes the value of the corresponding cell. Matrix $(\mathbf{I} - \mathbf{G})^{-1}$ is multiplied by the target mapping matrix \mathbf{T} to yield matrix \mathbf{U} in Fig. 5.5, whose structure reveals how the individual waypoints are influenced by the target waypoints of their own and neighboring gait sequences. Overall, it can be seen that both matrices have a diagonal-like structure and that the waypoints are influenced at most by the surrounding ten target waypoints. The influence is most pronounced in the boundary regions to the neighboring gait sequences and especially in running since there is a stronger coupling and more boundary conditions compared to walking.

5.2.5 Ensuring continuity for the transition between standing and walking

When the robot is stationary, the desired CoM position, DCM position and VRP position coincide, such that DCM velocity and CoM velocity are zero (see (3.2) and (3.7)). It follows that

$$\chi_{1,w_1} = \xi_{\chi,1,w_1} = v_{\chi,1,w_1} \quad (5.48)$$

must be valid in the first walking sequence and

$$\chi_{N,w_N} = \xi_{\chi,N,w_N} = v_{\chi,N,w_N} \quad (5.49)$$

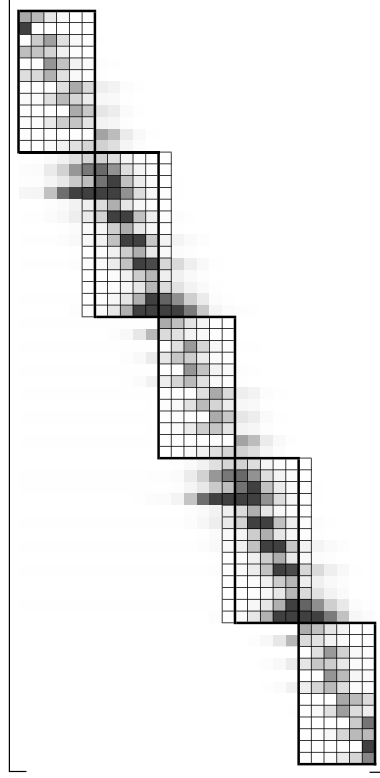


Figure 5.5: Heatmap of the global target mapping matrix \mathbf{U} in (5.47) with same settings as in Fig. 5.3. The values of each cell \square are displayed on a linear scale from $\square = 0$ (white) to $|\square| \geq 1$ (dark gray).

in the last walking sequence. In the stand-to-walk (S2W) transition, the initial CoM waypoint in the first walking sequence χ_{1,w_1} is already constrained to coincide with the first VRP waypoint $v_{\chi,1,w_1}$ via (5.42). Similarly, in the walk-to-stand (W2S) transition, the terminal DCM waypoint ξ_{χ,N,w_N} in the last walking sequence is constrained to be equal to the terminal VRP waypoint v_{χ,N,w_N} via (5.43). Thus, two additional constraints are necessary for a continuous transition between standing and walking. In the S2W transition, the first DCM waypoint must coincide with the first VRP waypoint. Due to the stable CoM dynamics (3.2) the CoM position converges to the steady state (5.48) anyway, but to reach it in finite time the last CoM position in the W2S transition has to be set equal to the last VRP waypoint. The two additional constraints can be expressed in matrix form:

$$\underbrace{\begin{bmatrix} v_{\chi,1,w_1} \\ v_{\chi,1,w_1} \\ \mathbf{X}_{\text{target,rem}} \\ v_{\chi,N,w_N} \\ v_{\chi,N,w_N} \end{bmatrix}}_{\mathbf{K}_{wpc} \in \mathbb{R}^{N_t \times 2}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ & & \mathbf{u}_1 & & \\ \mathbf{0}_{N_{rem}} & \mathbf{0}_{N_{rem}} & \mathbf{I}_{N_{rem}} & \mathbf{0}_{N_{rem}} & \mathbf{0}_{N_{rem}} \\ & & \mathbf{u}_{N_{wp}} & & \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}_{tcm} \in \mathbb{R}^{N_t \times N_t}} \mathbf{X}_{\text{target}}. \quad (5.50)$$

Here, \mathbf{K}_{wpc} denotes the target waypoint constraint matrix and \mathbf{K}_{tcm} denotes the target constraint mapping matrix, with $\mathbf{0}_{N_{rem}} \in \mathbb{R}^{N_{rem}}$ and $\mathbf{I}_{N_{rem}} \in \mathbb{R}^{N_{rem} \times N_{rem}}$. The matrix

$\mathbf{X}_{\text{target,rem}}$ summarizes the $N_{\text{rem}} = N_t - 4$ remaining waypoints, which can be freely specified together with $\mathbf{v}_{\chi,1,w_1}$ and \mathbf{v}_{χ,N,w_N} during the footstep planning. The column vectors \mathbf{u}_1 and $\mathbf{u}_{N_{wp}}$ denote the first and last row of matrix \mathbf{U} in (5.47). The second row in (5.50) encodes the first DCM position in walking sequence w_1 as

$$\xi_{\chi,1,w_1} = \mathbf{u}_1 \mathbf{X}_{\text{target}} = \mathbf{v}_{\chi,1,w_1} \quad (5.51)$$

and the second to last row encodes last CoM position in walking sequence w_N as

$$\chi_{N,w_N} = \mathbf{u}_{N_{wp}} \mathbf{X}_{\text{target}} = \mathbf{v}_{\chi,N,w_N}. \quad (5.52)$$

Thereby, the degree of freedom of the second and second to last VRP position in the respective walking sequence w_1 and w_N is given up in order to fulfill the particular constraint. By inverting matrix \mathbf{K}_{tcm} , equation (5.50) results in

$$\mathbf{X}_{\text{target}} = \mathbf{K}_{tcm}^{-1} \mathbf{K}_{wpc}, \quad (5.53)$$

which maps the target waypoint constraint matrix \mathbf{K}_{wpc} to a waypoint matrix $\mathbf{X}_{\text{target}}$ that is compatible with constraints (5.51) and (5.52).

5.2.6 Postprocessing

For the subsequent calculation of the horizontal trajectory, the waypoints Σ_{wp,w,w_i} and Σ_{wp,r,r_i} of the individual walking and running sequences w_i and r_i , respectively, are extracted from the global horizontal waypoint matrix Σ_{wp} .

During the walking sequence the DCM and CoM trajectory is calculated with (3.16) and (3.21) for any time t_φ in phase φ , i.e.

$$\xi_{\chi,\varphi}(t_\varphi) = \alpha_{\xi,0,\varphi} \mathbf{v}_{wp,\varphi} + \beta_{\xi,0,\varphi} \mathbf{v}_{wp,\varphi+1} + \gamma_{\xi,0,\varphi} \xi_{\chi,wp,\varphi+1} \quad (5.54)$$

and

$$\chi_\varphi(t_\varphi) = \alpha_{x,\varphi}(t_\varphi) \mathbf{v}_{wp,\varphi} + \beta_{x,\varphi}(t_\varphi) \mathbf{v}_{wp,\varphi+1} + \gamma_{x,\varphi}(t_\varphi) \xi_{\chi,wp,\varphi+1} + \delta_{x,\varphi}(t_\varphi) \chi_{wp,\varphi}. \quad (5.55)$$

Here, $\mathbf{v}_{wp,\varphi}$, $\mathbf{v}_{wp,\varphi+1}$, $\xi_{\chi,wp,\varphi+1}$ and $\chi_{wp,\varphi}$ denote the individual waypoints in the VRP waypoint matrix $\mathbf{V}_{\chi,i}$ and walking waypoint matrix Σ_{wp,w,w_i} .

For running, the polynomial parameters \mathbf{P}_χ for all stance phases in running sequence r_i are calculated according to (5.26) as

$$\mathbf{P}_\chi = \mathbf{A} \Sigma_{wp,r,r_i} + \mathbf{E}_\chi \mathbf{X}_{\text{foot},i}. \quad (5.56)$$

The horizontal CoM trajectory for running is calculated for any time $t_s \in [0..T_{s,i}]$ during stance phase i by inserting the polynomial parameters \mathbf{P}_χ in (4.16). During flight, the horizontal trajectory is given by the first and second row of (4.2) and (4.3).

5.3 Divergent Component of Motion tracking control

Due to the naturally stable CoM dynamics, only the unstable first-order DCM dynamics must be controlled. The control framework used for this purpose is adopted from

Englsberger et al. [9]. To track the reference DCM trajectory a closed-loop dynamics of the form

$$\underbrace{\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_{\text{ref}}}_{\dot{\mathbf{e}}_{\boldsymbol{\xi}}} = -k_{\boldsymbol{\xi}} \underbrace{(\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{ref}})}_{\mathbf{e}_{\boldsymbol{\xi}}} \quad (5.57)$$

is desired, which is stable for $k_{\boldsymbol{\xi}} > 0$. The DCM error $\mathbf{e}_{\boldsymbol{\xi}}$ converges asymptotically and thus also the desired VRP \mathbf{v}_{des} and the corresponding eCMP $\mathbf{r}_{\text{eCMP,des}}$ converge to their reference values. Inserting the DCM dynamics (3.7) into the desired dynamics and solving for the input \mathbf{v} yields the tracking control law of the form

$$\mathbf{v}_{\text{des}} = \boldsymbol{\xi} - b \dot{\boldsymbol{\xi}}_{\text{ref}} + k_{\boldsymbol{\xi}} b (\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{ref}}) \quad (5.58)$$

Writing the DCM dynamics (3.7) in terms of reference quantities and inserting into (5.58) yields the DCM tracking controller of the form

$$\mathbf{v}_{\text{des}} = \mathbf{v}_{\text{ref}} + (1 + b k_{\boldsymbol{\xi}}) (\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{ref}}), \quad (5.59)$$

which takes a VRP reference trajectory \mathbf{v}_{ref} and the DCM tracking error $\mathbf{e}_{\boldsymbol{\xi}}$ as input to determine the desired VRP position. Inserting (5.59) into (3.6) returns the desired linear force on the CoM as

$$\mathbf{F}_{\text{com,des}} = \frac{m}{b^2} (\mathbf{x} - \mathbf{v}_{\text{ref}} - (1 + b k_{\boldsymbol{\xi}}) (\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{ref}})). \quad (5.60)$$

By subtracting the gravitational force \mathbf{F}_g , the desired leg force is given by

$$\mathbf{F}_{\text{leg,des}} = \frac{m}{b^2} (\mathbf{x} - \mathbf{v}_{\text{ref}} - (1 + b k_{\boldsymbol{\xi}}) (\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{ref}})) - \mathbf{F}_g, \quad (5.61)$$

which is commanded to the whole-body controller in Chapter 7.

Chapter 6

Evaluation and Visualization of the Presented Methods

In the previous section, a method for generating smooth and consistent trajectories of multiple gait transitions between walking and running was introduced. The results obtained are presented below. The visually meaningful force-to-point transformations

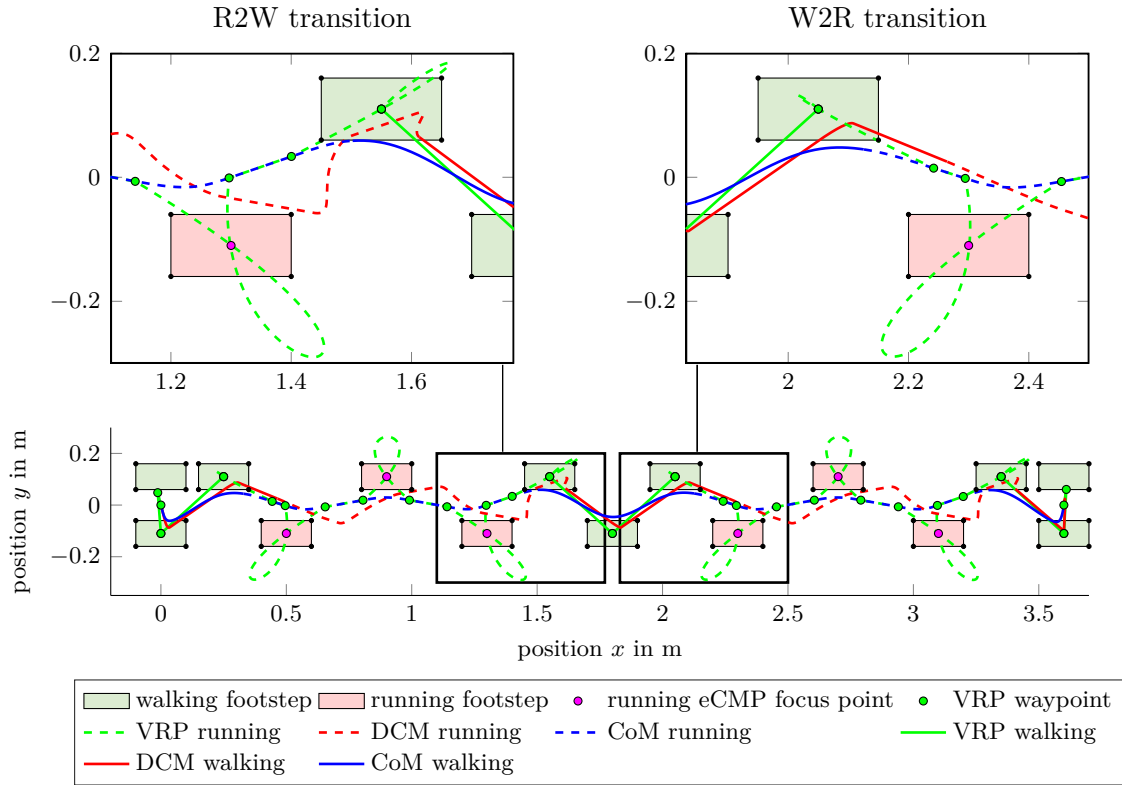


Figure 6.1: Top view of a trajectory consisting of five consecutive gait sequences with 0.25 m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking (eCMP stationary in foot center during single support) and 0.4 m step length and stance time $T_s = 0.12$ s during running (eCMP focus point on the foot plane).

(VRP and eCMP) and the DCM introduced in Chapter 3 are additionally calculated for the running sequences and displayed for the whole trajectory. Provided the CoM

position is continuous, a continuous DCM trajectory corresponds to a continuous velocity trajectory according to (3.1). Furthermore, equations (3.6) and (3.4) show that continuous VRP and eCMP trajectories correspond to continuous forces on the CoM and external forces, respectively. In the horizontal direction, the VRP is equal to the eCMP (see (3.5)) and it is sufficient to plot one of the two quantities. Fig. 6.1 shows a series of five consecutive gait sequences with three footsteps each. The robot's CoM alternates between walking and running motions, while additionally switching from standing to walking at the beginning of the first walking sequence, and switching back to standing at the end of the last walking sequence. As described in Chapter 5, the entire trajec-

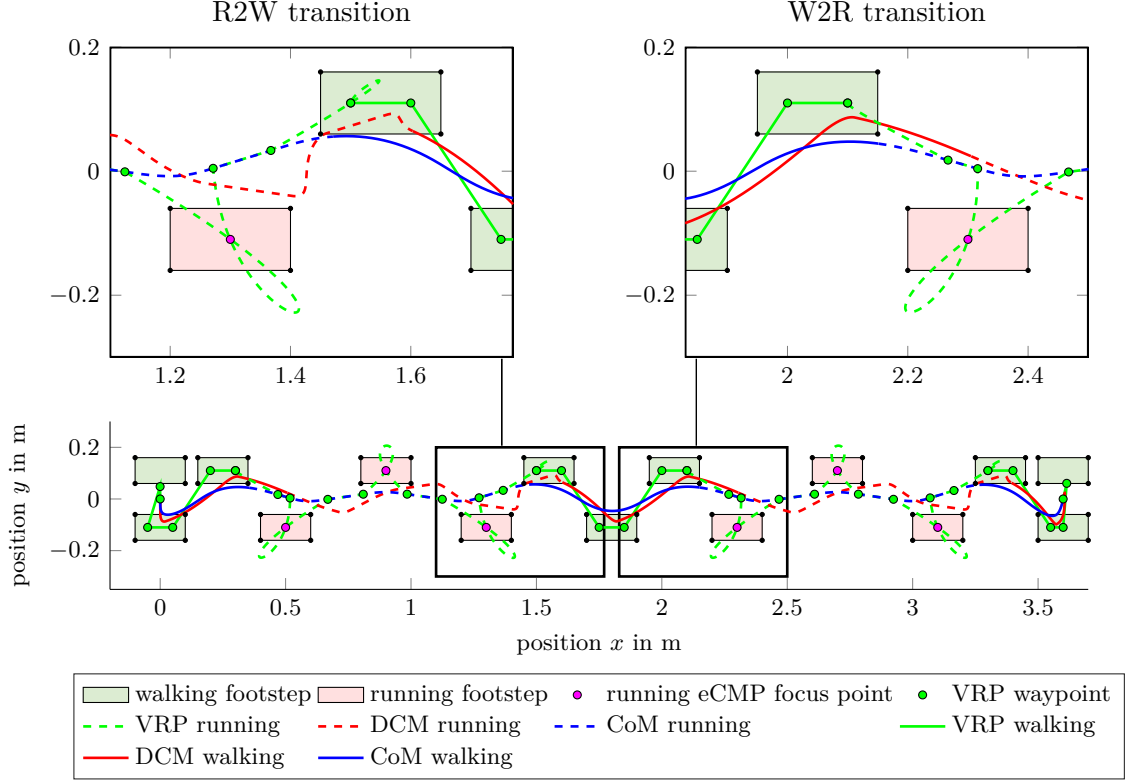


Figure 6.2: Top view of a trajectory consisting of five consecutive gait sequences with 0.25 m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking (eCMP moves from the heel of the foot) and 0.4 m step length and stance time $T_s = 0.12$ s during running (eCMP focus point 0.3 m vertically below the foot plane).

tory depends only on the VRP waypoints for walking and the eCMP focus points for running, which can be freely chosen with the exception of the second and second to last horizontal VRP (see Section 5.2.5). The VRP waypoints can be readily computed from the eCMP waypoints via (3.5). Therefore, it is sufficient to plan the eCMP waypoints. For running, in addition to footstep center points, the last and first eCMP waypoint of the previous and subsequent walking sequences is used as eCMP focus point. This creates the push-off or landing in the single support phase during the transition between walking and running.

The placement of eCMP waypoints for walking depends on the walking sequence, which are distinguished between standing-to-running (S2R), running-to-running (R2R) and running-to-standing (R2S) walking sequences. In the S2R and R2S walking sequence, the first and last eCMP waypoint, respectively, is chosen to correspond to the standing

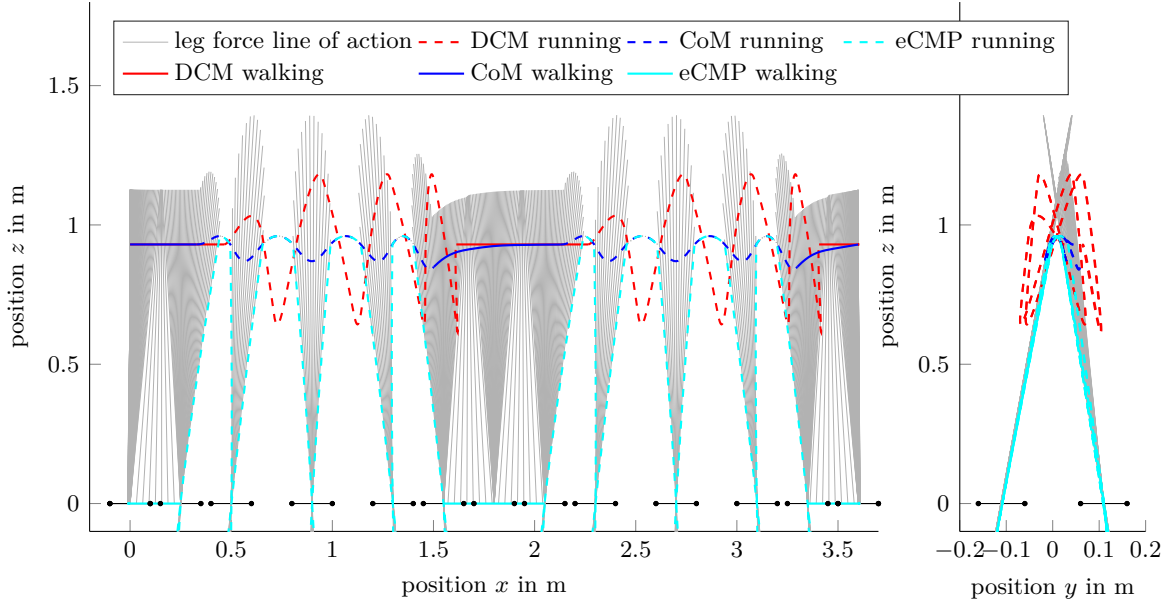


Figure 6.3: Side view of the trajectory in x- and y-direction with the eCMP stationary in the foot center during walking single support and the eCMP focus on the foot plane during running. In the diagram on the right, only the running sequences are displayed since the eCMP waypoints are constrained to the footstep surface during walking anyway.

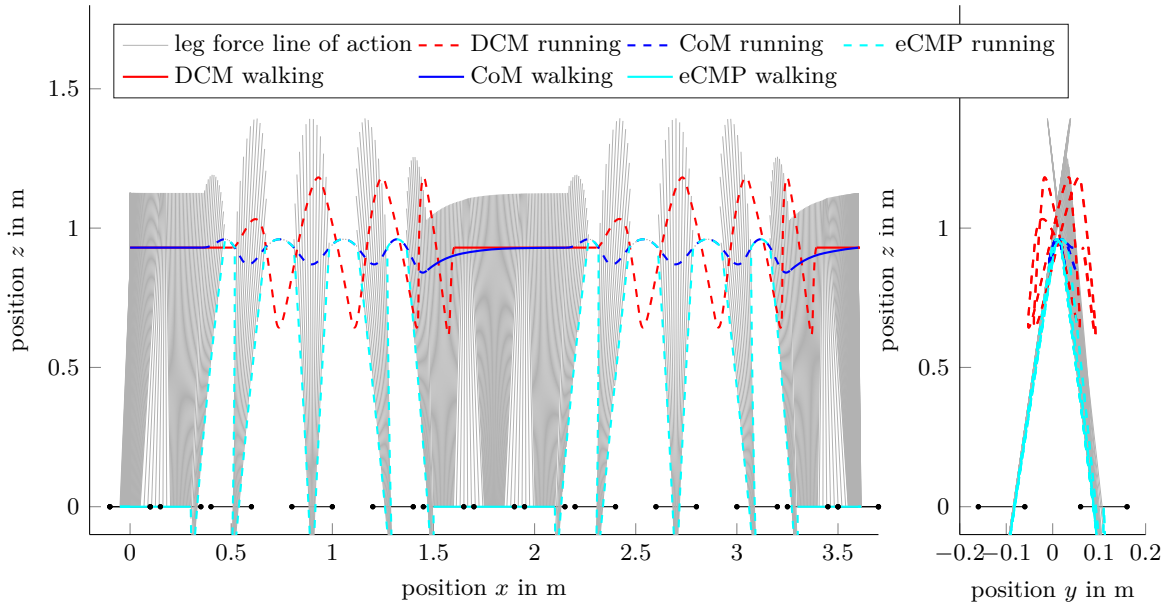


Figure 6.4: Side view of the trajectory in x- and y-direction with eCMP heel-to-toe movement during walking single support and the eCMP focus point 0.3m vertically below the foot plane during running. In the diagram on the right, only the running sequences are displayed since the eCMP waypoints are constrained to the footstep surface during walking anyway.

eCMP waypoint that lies exactly between the two footsteps in case of no perturbations. An additional eCMP waypoint is inserted, which is determined in Section 5.2.5 in such a way that there is a continuous transition between standing and walking.

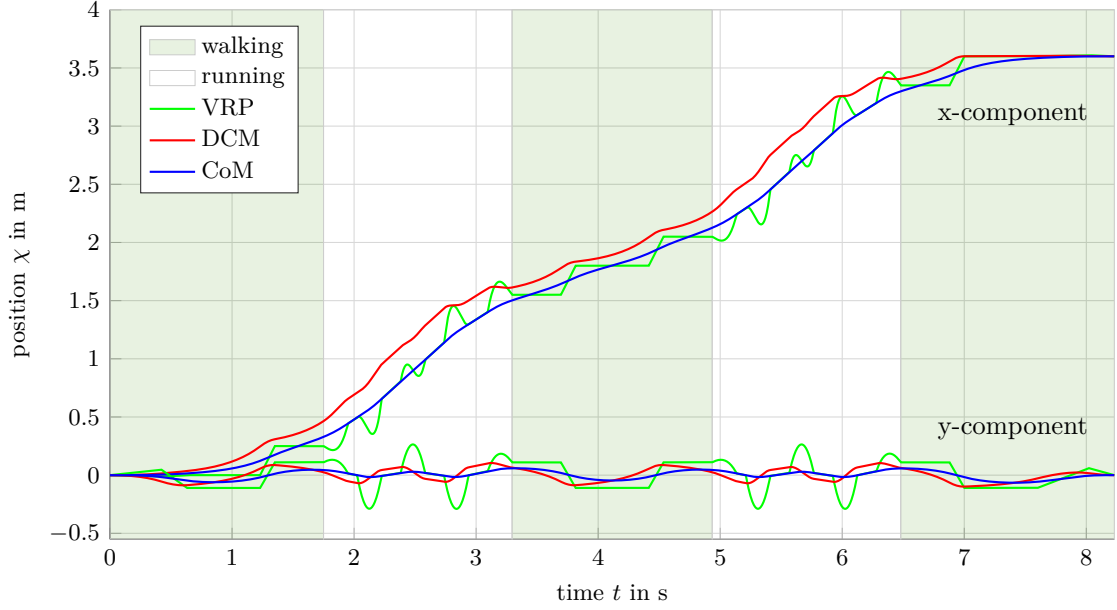


Figure 6.5: Time diagram of the horizontal trajectory consisting of five consecutive gait sequences with 0.25m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking and 0.4m step length and stance time $T_s = 0.12$ s during running.

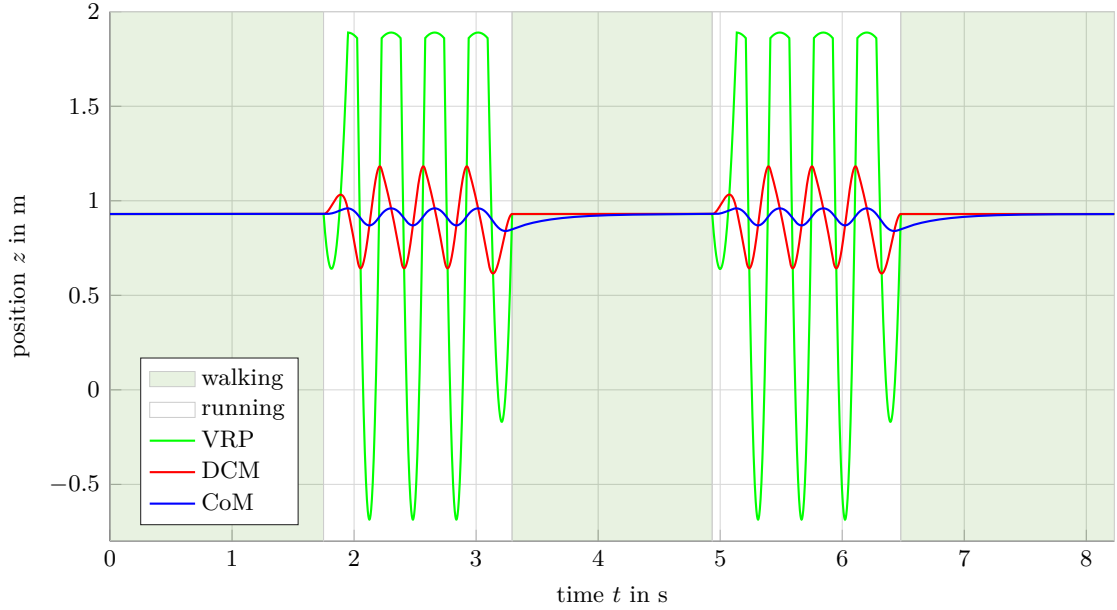


Figure 6.6: Time diagram of the vertical trajectory consisting of five consecutive gait sequences with 0.25m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking and 0.4m step length and stance time $T_s = 0.12$ s during running.

For the remaining footsteps as well as all footsteps in the R2R walking sequence, two eCMP waypoints are planned per footstep. There are several ways to place the waypoints. In Fig. 6.1, similar to running, the eCMP waypoints are chosen to be both at the foot center, i.e. the eCMP trajectory remains at the foot center throughout the single support phase. Another option is to place the eCMP waypoints on the longitudinal axis of the foot, allowing the eCMP trajectory to move from the heel to the ball of the foot

during the single support phase, as shown in Fig. 6.2. Additionally, the eCMP focus point for running is shifted 0.3m in the vertical direction, such that it lies under the foot. In combination, this creates a slightly more natural gait and also makes the DCM trajectory a little smoother especially in the R2W transition.

Fig. 6.3 shows the side view of the trajectory and illustrates that the external force acting on the CoM always passes through the center of the foot in the single support phase. This corresponds to the same eCMP start and endpoints in the single support phase of walking as in Fig. 6.1. According to (3.4), the eCMP trajectory in the flight phase, i.e. the leg force is zero, is equal to the CoM trajectory. During the stance phase of running, the eCMP is below the foot position in a way that maintains the chosen eCMP focus point.

The same plot is shown in Fig. 6.4 with the heel-to-toe transition of the eCMP during walking and the shift of the eCMP focus point under the foot during running. It can be observed that the line of action of the external force is more distributed across the foot, but at all times remains within the footstep boundary.

Already in the spatial plots, it became clear that the DCM is always in front of the CoM and attracts it. Figs. 6.5 and 6.6 show the time diagram of the horizontal and vertical trajectories and illustrate even more clearly that the DCM always precedes the CoM, which is particularly well visible in the x-direction in Fig. 6.5.

Since the VRP trajectory is continuous and thus with (3.6) and $\ddot{\mathbf{x}} = \mathbf{F}_{com}/m$ the acceleration is continuous, it can be concluded that the chosen continuity requirements are satisfied at the transition between walking and running. The CoM trajectory is C^2 continuous and the DCM trajectory is C^1 continuous, which corresponds to a C^1 continuity of the velocities. Fig. 6.7 shows the leg forces of the humanoid robot Toro [8] with a weight of 79.2 kg, which fulfill C^0 continuity within the gait sequences and at the gait transitions.

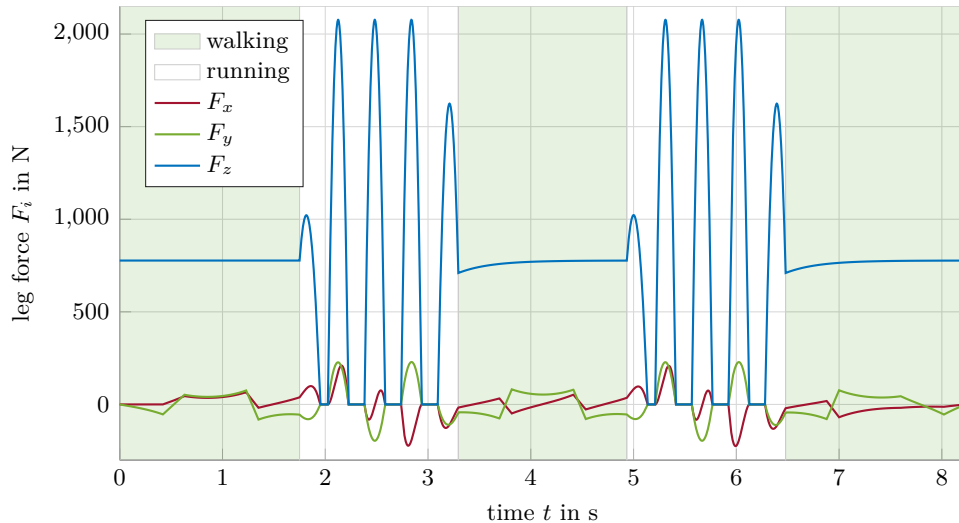


Figure 6.7: Leg forces over time of the humanoid robot Toro (79.2 kg). Step lengths and timings according to Fig. 6.6.

6.1 Foot trajectories

To obtain the foot trajectories, two consecutive foot positions for a single leg are interpolated with a fifth-order polynomial and a sixth-order polynomial in the horizontal and vertical direction, respectively. The chosen polynomials have enough degrees of freedom to allow the velocity and acceleration at the beginning and end of the swing phase to be zero and, in addition, in the vertical direction to reach the desired apex height in the middle of the swing phase. A linear system of equations can be set up that yields the vectors of the polynomial coefficients in the horizontal and vertical direction

$$\mathbf{p}_{\chi,f} = \underbrace{\begin{bmatrix} \mathbf{t}_{\chi}^T(0) \\ \mathbf{t}_{\dot{\chi}}^T(0) \\ \mathbf{t}_{\ddot{\chi}}^T(0) \\ \mathbf{t}_{\chi}^T(T_{sw}) \\ \mathbf{t}_{\dot{\chi}}^T(T_{sw}) \\ \mathbf{t}_{\ddot{\chi}}^T(T_{sw}) \end{bmatrix}}_{\mathbf{T}_{c,\chi}}^{-1} \underbrace{\begin{bmatrix} \boldsymbol{\chi}_{f,sw}(0) \\ \dot{\boldsymbol{\chi}}_{f,sw}(0) \\ \ddot{\boldsymbol{\chi}}_{f,sw}(0) \\ \boldsymbol{\chi}_{f,sw}(T_{sw}) \\ \dot{\boldsymbol{\chi}}_{f,sw}(T_{sw}) \\ \ddot{\boldsymbol{\chi}}_{f,sw}(T_{sw}) \end{bmatrix}}_{\mathbf{F}_{c,\chi}} \quad \text{and} \quad \mathbf{p}_{z,f} = \underbrace{\begin{bmatrix} \mathbf{t}_z^T(0) \\ \mathbf{t}_{\dot{z}}^T(0) \\ \mathbf{t}_{\ddot{z}}^T(0) \\ \mathbf{t}_z^T\left(\frac{T_{sw}}{2}\right) \\ \mathbf{t}_z^T(T_{sw}) \\ \mathbf{t}_{\dot{z}}^T(T_{sw}) \\ \mathbf{t}_{\ddot{z}}^T(T_{sw}) \end{bmatrix}}_{\mathbf{T}_{c,z}}^{-1} \underbrace{\begin{bmatrix} z_{f,sw}(0) \\ \dot{z}_{f,sw}(0) \\ \ddot{z}_{f,sw}(0) \\ z_{f,sw}\left(\frac{T_{sw}}{2}\right) \\ z_{f,sw}(T_{sw}) \\ \dot{z}_{f,sw}(T_{sw}) \\ \ddot{z}_{f,sw}(T_{sw}) \end{bmatrix}}_{\mathbf{f}_{c,z}}, \quad (6.1)$$

respectively. Here, the matrices $\mathbf{T}_{c,\chi}$ and $\mathbf{T}_{c,z}$ are assembled with the following time mapping row vectors

$$\begin{aligned} \mathbf{t}_{\chi}^T(t_{sw}) &= [1, t_{sw}, t_{sw}^2, t_{sw}^3, t_{sw}^4, t_{sw}^5] & \mathbf{t}_z^T(t_{sw}) &= [1, t_{sw}, t_{sw}^2, t_{sw}^3, t_{sw}^4, t_{sw}^5, t_{sw}^6] \\ \mathbf{t}_{\dot{\chi}}^T(t_{sw}) &= [0, 1, 2t_{sw}, 3t_{sw}^2, 4t_{sw}^3, 5t_{sw}^4] & \mathbf{t}_{\dot{z}}^T(t_{sw}) &= [0, 1, 2t_{sw}, 3t_{sw}^2, 4t_{sw}^3, 5t_{sw}^4, 6t_{sw}^5] \\ \mathbf{t}_{\ddot{\chi}}^T(t_{sw}) &= [0, 0, 2, 6t_{sw}, 12t_{sw}^2, 20t_{sw}^3] & \mathbf{t}_{\ddot{z}}^T(t_{sw}) &= [0, 0, 2, 6t_{sw}, 12t_{sw}^2, 20t_{sw}^3, 30t_{sw}^4]. \end{aligned} \quad (6.2)$$

The local time is defined as $t_{sw} \in [0, T_{sw}]$ and T_{sw} denotes the total swing time.

Table 6.1 shows the appropriate boundary conditions for the components of $\mathbf{F}_{c,\chi}$ and $\mathbf{f}_{c,z}$ in (6.1), where $\mathbf{x}_{f,0} = [\boldsymbol{\chi}_{f,0} \ z_{f,0}]$ is the starting position of the foot and $\mathbf{x}_{f,T} = [\boldsymbol{\chi}_{f,T} \ z_{f,T}]$ is the foot target position. The foot positions, velocities and accelerations for the current time t_{sw} are calculated with the polynomial coefficients from (6.1) as

$$\begin{bmatrix} \boldsymbol{\chi}_{f,sw}(t_{sw}) \\ \dot{\boldsymbol{\chi}}_{f,sw}(t_{sw}) \\ \ddot{\boldsymbol{\chi}}_{f,sw}(t_{sw}) \end{bmatrix} = \begin{bmatrix} \mathbf{t}_{\chi}^T(t_{sw}) \\ \mathbf{t}_{\dot{\chi}}^T(t_{sw}) \\ \mathbf{t}_{\ddot{\chi}}^T(t_{sw}) \end{bmatrix} \mathbf{p}_{\chi} \quad \text{and} \quad \begin{bmatrix} z_{f,sw}(t_{sw}) \\ \dot{z}_{f,sw}(t_{sw}) \\ \ddot{z}_{f,sw}(t_{sw}) \end{bmatrix} = \begin{bmatrix} \mathbf{t}_z^T(t_{sw}) \\ \mathbf{t}_{\dot{z}}^T(t_{sw}) \\ \mathbf{t}_{\ddot{z}}^T(t_{sw}) \end{bmatrix} \mathbf{p}_z. \quad (6.3)$$

In the stance phase, it is assumed that the foot is not slipping on the ground. Accordingly, the stance trajectory can readily be defined as

$$\begin{aligned} \boldsymbol{\chi}_{f,st}(t) &= \boldsymbol{\chi}_{f,0}, \\ z_{f,st}(t) &= z_{f,0}, \end{aligned} \quad (6.4)$$

order	position interpolation	boundary conditions
5	$\chi_{f,sw}(t_{sw}) = \chi_{f,0}$	$\chi_{f,sw}(0) = \chi_{f,0}$
	$-(10t^3(\chi_{f,0} - \chi_{f,T}))/T_{sw}^3$	$\dot{\chi}_{f,sw}(0) = 0$
	$+ (15t^4(\chi_{f,0} - \chi_{f,T}))/T_{sw}^4$	$\ddot{\chi}_{f,sw}(0) = 0$
	$-(6t^5(\chi_{f,0} - \chi_{f,T}))/T_{sw}^5$	$\chi_{f,sw}(T_{sw}) = \chi_{f,T}$
		$\dot{\chi}_{f,sw}(T_{sw}) = 0$
6		$\ddot{\chi}_{f,sw}(T_{sw}) = 0$
	$z_{f,sw}(t_{sw}) = z_{f,0}$	$z_{f,sw}(0) = z_{f,0}$
	$-(2t^3(21z_{f,0} + 11z_{f,T} - 32z_{f,apex}))/T_{sw}^3$	$\dot{z}_{f,sw}(0) = 0$
	$+ (3t^4(37z_{f,0} + 27z_{f,T} - 64z_{f,apex}))/T_{sw}^4$	$\ddot{z}_{f,sw}(0) = 0$
	$-(6t^5(17z_{f,0} + 15z_{f,T} - 32z_{f,apex}))/T_{sw}^5$	$z_{f,sw}(T_{sw}/2) = z_{f,apex}$
	$+ (32t^6(z_{f,0} + z_{f,T} - 2z_{f,apex}))/T_{sw}^6$	$z_{f,sw}(T_{sw}) = z_{f,T}$
		$\dot{z}_{f,sw}(T_{sw}) = 0$
		$\ddot{z}_{f,sw}(T_{sw}) = 0$

Table 6.1: Analytical solution of the foot position interpolation and corresponding boundary conditions in the horizontal and vertical direction.

where $\chi_{f,0}$ and $z_{f,0}$ is the current foot position in the horizontal and vertical direction, respectively, and the corresponding velocities and accelerations are zero.

This allows the foot reference trajectories to be assembled for the entire sequence of multiple stance and swing phases, resulting in

$$\mathbf{x}_{f,ref} = \begin{bmatrix} [\chi_{f,st,1} \ z_{f,st,1}] \\ [\chi_{f,sw,1} \ z_{f,sw,1}] \\ \vdots \\ [\chi_{f,st,N} \ z_{f,st,N}] \end{bmatrix}, \quad \dot{\mathbf{x}}_{f,ref} = \begin{bmatrix} [\dot{\chi}_{f,st,1} \ \dot{z}_{f,st,1}] \\ [\dot{\chi}_{f,sw,1} \ \dot{z}_{f,sw,1}] \\ \vdots \\ [\dot{\chi}_{f,st,N} \ \dot{z}_{f,st,N}] \end{bmatrix}, \quad \ddot{\mathbf{x}}_{f,ref} = \begin{bmatrix} [\ddot{\chi}_{f,st,1} \ \ddot{z}_{f,st,1}] \\ [\ddot{\chi}_{f,sw,1} \ \ddot{z}_{f,sw,1}] \\ \vdots \\ [\ddot{\chi}_{f,st,N} \ \ddot{z}_{f,st,N}] \end{bmatrix}. \quad (6.5)$$

The foot reference trajectories are visualized in the next section over a series of 3D stepping stones and also provide a reference for the foot tracking task in the whole-body controller in Chapter 7.

6.2 Walking and running over stepping stones

Since the foot stepping points for walking and running are specified exactly, the developed method is also suitable for generating trajectories over varying floor positions and heights, e.g. stepping stones. The algorithm facilitates trajectories around curves, but this requires planning foot orientation trajectories, which will be part of future work. Fig. 6.8 shows the CoM trajectory and the foot trajectories derived in Section 6.1 for walking (continuous line) and running (dashed line) over several stepping stones, whose color specifies the gait. To illustrate the relationship between the trajectories, the legs

of the robot are indicated by two lines from the CoM trajectory to the foot trajectories, which are shown at different times during the gait series.

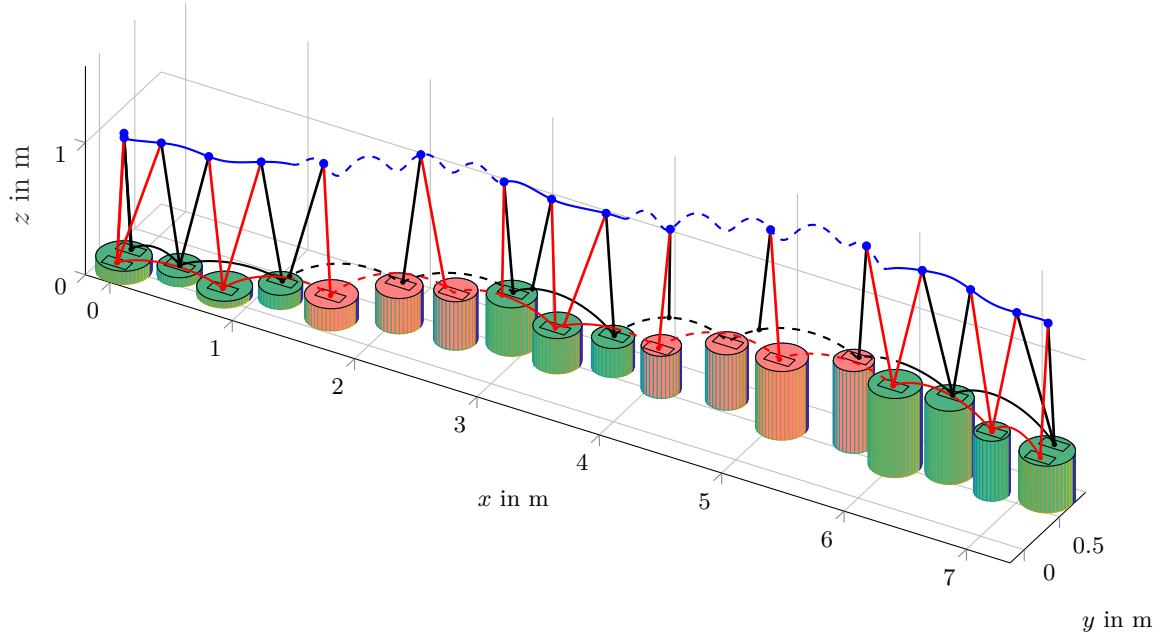


Figure 6.8: Bipedal point mass model walking (continuous line) and running (dashed line) over 3D stepping stones with CoM (blue), left foot (black) and right foot (red) trajectories.

Chapter 7

Whole-Body Control

This chapter provides an overview of the whole-body control (WBC) framework, which allows the robot to perform various tasks such as walking or running. Humanoid robots usually have a large number of degrees of freedom that are exploited to varying degrees for different tasks. The task of the whole-body controller is to find the optimal control outputs for all degrees of freedom, even if they are not primarily involved in the execution of a task. This applies, for example, to the upper body during locomotion, which nevertheless plays an important role in maintaining balance. In this thesis, an inverse dynamics approach introduced in Engelsberger [5] is adopted, embedded in a quadratic program (QP) based optimization framework, which has been shown to provide satisfactory results for both gaits individually. The main objective of the whole-body controller in this work is to be general enough to enable the execution of whole-body walking and running motions and their transitions based on the reference trajectories derived in Chapters 5 and 6.

7.1 Dynamic model

This section gives a brief overview of the free-floating dynamic model for humanoid robots. The advantage of legged robots is that they can move freely in their environment and thus gain an additional six degrees of freedom compared to stationary robots. To account for these, the state of the free floating base is added to the robot's configuration. It is described by the position $\mathbf{x}_{base} \in \mathbb{R}^3$ and orientation $\mathbf{R}_{base} \in SO(3)$ of the base frame relative to the world frame. In the base state vector $\mathbf{q}_{base} \in \mathbb{R}^6$ the orientation is represented in local coordinates $\boldsymbol{\alpha}_{base} \in \mathbb{R}^3$, e.g. roll-pitch-yaw angles. Combined with the joint position $\mathbf{q}_{jnt} \in \mathbb{R}^n$ the variables are concentrated into a single state vector:

$$\mathbf{q} = \begin{bmatrix} \mathbf{x}_{base} \\ \boldsymbol{\alpha}_{base} \\ \mathbf{q}_{jnt} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{base} \\ \mathbf{q}_{jnt} \end{bmatrix} \in \mathbb{R}^{6+n}. \quad (7.1)$$

The additional six degrees of freedom of the base cannot be controlled directly. Thus, robots with a free-floating base belong to a class called underactuated systems, for which the equation of motion is given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_g(\mathbf{q}) = \mathbf{S}^T \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}. \quad (7.2)$$

Here, \mathbf{M} is the inertia matrix, \mathbf{C} denotes the Coriolis and centrifugal matrix and the vector $\boldsymbol{\tau}_g$ captures the gravitational forces and moments. The robot's joint torques are concentrated in the vector $\boldsymbol{\tau}$. The matrix $\mathbf{S} = [\mathbf{0}_{n \times 6} \quad \mathbf{I}_{n \times n}]$ is a selection matrix, that selects only the bottom n components of $\boldsymbol{\tau}$, which can be actuated directly. The vector $\boldsymbol{\tau}_{ext}$ contains the generalized external forces and torques acting on the robot. For simplicity, it is assumed that all external forces act at the end-effector frames $k \in [1..\eta]$. Due to the focus on locomotion, only the two feet are considered to be end effectors, which yields $\eta = 2$. With this assumption $\boldsymbol{\tau}_{ext}$ simplifies to

$$\boldsymbol{\tau}_{ext} = \mathbf{J}(\mathbf{q})^T \mathbf{w}, \quad (7.3)$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$ is the Jacobi matrix¹ and the vector $\mathbf{w} \in \mathbb{R}^m$ contains all Cartesian end effector wrenches \mathbf{w}_k , which are the six-dimensional vectors of external Cartesian forces and torques.

7.2 Contact constraints on the robot

Despite the slightly misleading name of the dynamic model, the robot does not float in space and therefore needs to make contact with its environment. One approach to modeling the contact is to approximate the continuous rectangular foot contact area via single contact points in the four corners [10, 13]. Since the robot can generally only push off the ground and only move forward due to friction, unilateral contact and friction cone constraints must be considered. Here, the friction cone in each contact

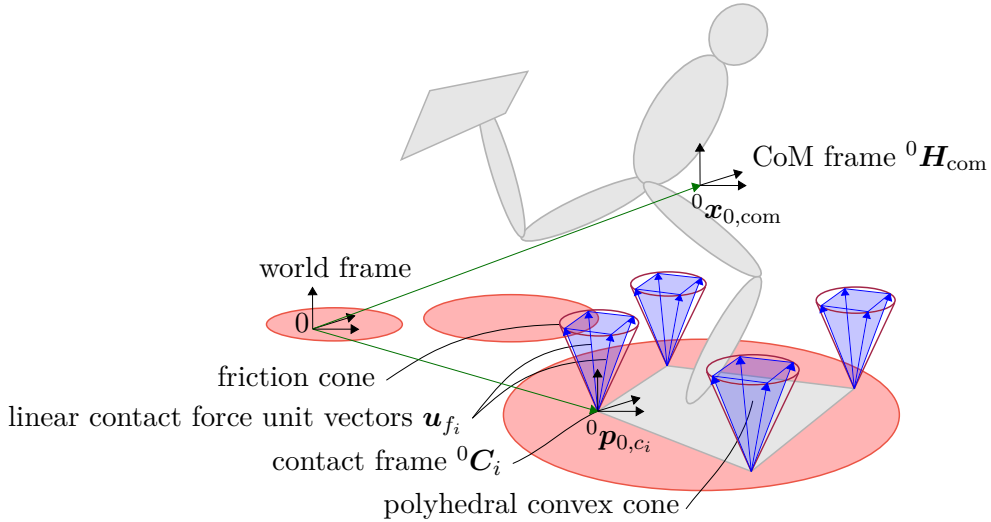


Figure 7.1: Outline of CoM frame, contact frame and polyhedral convex cone approximation (adapted from [5]).

point is approximated by a polyhedral convex cone consisting of four contact forces (see Fig. 7.1). With four contact points in each foot, there are a total of $n_{cf} = 16$ contact

¹Supposing the forward kinematics $\mathbf{x} = \varphi(\mathbf{q})$ is a local mapping between the configuration space $\mathbf{q} \in \mathbb{R}^n$ and an m -dimensional task space $\mathbf{x} \in \mathbb{R}^m$. Then the velocity transformation between these spaces is given by: $\dot{\mathbf{x}} = \frac{d\varphi(\mathbf{q})}{dt} = \frac{\partial \varphi(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}$, where $\mathbf{J}(\mathbf{q})$ is the Jacobi matrix.

forces $\mathbf{f}_i \in \mathbb{R}^3$ per foot, which are defined by a unit vector ${}^0\mathbf{u}_{f_i}$ and a scalar parameter $\rho_i \geq 0$ indicating the direction and magnitude of the contact force, respectively, i.e.

$$\mathbf{f}_i = {}^0\mathbf{u}_{f_i} \rho_i. \quad (7.4)$$

Here, ${}^0\Box$ denotes the representation in the world frame for any quantity \Box . The contact force magnitudes ρ_i for each foot are summarized in the vector $\boldsymbol{\rho}$, which is mapped to the feasible spacial wrench \mathbf{w}_k via

$${}^0\mathbf{w}_k = \underbrace{\begin{bmatrix} {}^0\mathbf{g}_{\rho,1} & \cdots & {}^0\mathbf{g}_{\rho,n_{cf}} \end{bmatrix}}_{{}^0\mathbf{G}_{\rho,k}} \underbrace{\begin{bmatrix} \rho_1 \\ \vdots \\ \rho_{n_{cf}} \end{bmatrix}}_{\boldsymbol{\rho}} \quad (7.5)$$

where ${}^0\mathbf{G}_{\rho,k}$ is the so-called grasp matrix. The column vectors ${}^0\mathbf{g}_{\rho,i}$ are given by

$${}^0\mathbf{g}_{\rho,i} = \begin{bmatrix} {}^0\mathbf{u}_{f_i} \\ {}^0\tilde{\mathbf{p}}_{0,c_i} {}^0\mathbf{u}_{f_i} \end{bmatrix}, \quad (7.6)$$

where \mathbf{u}_{f_i} is the contact force direction vector from (7.4) and $\tilde{\mathbf{p}}_{0,c_i}$ is the skew-symmetric cross-product matrix

$${}^0\tilde{\mathbf{p}}_{0,c_i} = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \quad (7.7)$$

for the vector ${}^0\mathbf{p}_{0,c_i} = [p_x \ p_y \ p_z]$ denoting the spacial position of contact frame ${}^0\mathbf{C}_i$ corresponding to contact force \mathbf{f}_i . The definition of the contact wrench in (7.5) can be inserted in (7.3) and thus the general robot equations of motion (7.2) result in

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} + \underbrace{\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g}_h &= \mathbf{S}\boldsymbol{\tau} + \sum_{k=1}^{\eta} ({}_s\mathbf{J}_{0,k}^T \underbrace{{}^0\mathbf{G}_{\rho,k}\boldsymbol{\rho}}_{{}^0\mathbf{w}_k}) \\ &= \underbrace{\begin{bmatrix} \mathbf{S} & \sum_{k=1}^{\eta} ({}_s\mathbf{J}_{0,k}^T {}^0\mathbf{G}_{\rho,k}) \end{bmatrix}}_{\mathbf{A}_u} \underbrace{\begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\rho} \end{bmatrix}}_{\mathbf{u}}. \end{aligned} \quad (7.8)$$

Here, ${}_s\mathbf{J}_{0,k}^T$ is the spacial end effector Jacobian corresponding to the k -th end effector wrench ${}^0\mathbf{w}_k$. The robot's actuated joint torques and contact force magnitudes are selected as control variables and summarized in the control input vector \mathbf{u} . Matrix \mathbf{A}_u combines the selection matrix \mathbf{S} and the contact force mapping matrices in a single operator that maps the control input vector \mathbf{u} to the generalized joint torques. Since many whole-body control tasks can be written as a relation between the generalized accelerations $\ddot{\mathbf{q}}$ and the control inputs \mathbf{u} , equation (7.8) is solved for $\ddot{\mathbf{q}}$ as

$$\ddot{\mathbf{q}} = \underbrace{-\mathbf{M}^{-1}\mathbf{n}}_{\ddot{\mathbf{q}}_{\text{MB}}} + \underbrace{\mathbf{M}^{-1}\mathbf{A}_u}_{\mathbf{Q}} \mathbf{u}, \quad (7.9)$$

where $\ddot{\mathbf{q}}_{\text{MB}}$ is the acceleration due to multi-body effects and \mathbf{Q} denotes the mapping of control inputs \mathbf{u} to generalized accelerations $\ddot{\mathbf{q}}$. This allows inserting the acceleration directly into the respective tasks and the enforcement of (7.8) via an equality constraint in the QP is not necessary.

7.3 Derivation of tasks for the whole-body controller

Performing basic activities such as walking or running requires multiple whole-body control tasks to find control outputs for all joints. All tasks are formulated with the control input vector \mathbf{u} in the following form:

$$\mathbf{d}_{i,\text{des}} \stackrel{!}{=} \mathbf{D}_i \mathbf{u}. \quad (7.10)$$

Here, $\mathbf{d}_{i,\text{des}}$ and \mathbf{D}_i denote the desired task space vector and the corresponding task space mapping matrix of the i -th task, respectively.

7.3.1 Centroidal momentum task

According to Kajita et al. [15], the angular momentum \mathbf{l} of a robot can be computed as

$$\mathbf{l} = \mathbf{A}_{G,l} \dot{\mathbf{q}}, \quad (7.11)$$

where $\mathbf{A}_{G,l}$ denotes the part of the centroidal momentum matrix in Orin et al. [21], that maps generalized joint velocities to the angular momentum \mathbf{l} . Equation (7.11) is valid, if the angular momentum is expressed in the CoM frame ${}^0\mathbf{H}_{\text{com}}$ with the axis aligned with the world frame, i.e.

$${}^0\mathbf{H}_{\text{com}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & {}^0\mathbf{x}_{0,\text{com}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (7.12)$$

where ${}^0\mathbf{x}_{0,\text{com}}$ denotes the CoM position in the world frame. An angular momentum reference trajectory (\mathbf{l}_{ref} $\dot{\mathbf{l}}_{\text{ref}}$) can be tracked by the following controller

$$\underbrace{\dot{\mathbf{l}}_{\text{des}}}_{\boldsymbol{\tau}_{\text{com,des}}} = \dot{\mathbf{l}}_{\text{ref}} - k_l (\mathbf{l} - \mathbf{l}_{\text{ref}}), \quad (7.13)$$

which is stable for $k_l \geq 0$. However, for walking and running at moderate speeds, it is an acceptable assumption to choose the reference angular momentum \mathbf{l}_{ref} and its rate of change $\dot{\mathbf{l}}_{\text{ref}}$ to be zero. This allows the desired torques $\boldsymbol{\tau}_{\text{com,des}}$ from (7.13) around the CoM and the leg force $\mathbf{F}_{\text{leg,des}}$ on the CoM from (5.61) to be formulated in relation to the control input vector \mathbf{u} as follows

$$\underbrace{\begin{bmatrix} \mathbf{F}_{\text{leg,des}} \\ \boldsymbol{\tau}_{\text{com,des}} \end{bmatrix}}_{\mathbf{d}_{\text{com,des}}} = \underbrace{\begin{bmatrix} \mathbf{0}_{6 \times n_\tau} & {}^{\text{com}}\mathbf{A}_\rho \end{bmatrix}}_{\mathbf{D}_{\text{com}}} \mathbf{u}, \quad (7.14)$$

where n_τ is the number of actuated joints in the robot and the matrix

$${}^{\text{com}}\mathbf{A}_\rho = \mathbf{A} \mathbf{d}_{0\mathbf{H}_{\text{com}}}^T \sum_{k=1}^{\eta} ({}^0\mathbf{A}_{\rho,k}) \quad (7.15)$$

maps the vector of contact force magnitudes $\boldsymbol{\rho}$ to an external wrench in the CoM frame ${}^0\mathbf{H}_{\text{com}}$. It consists of two successive mappings, first from the contact forces $\boldsymbol{\rho}$ to spacial wrenches via (7.5) and then by the adjoint transpose $\mathbf{A} \mathbf{d}_{0\mathbf{H}_{\text{com}}}^T$ from the world frame to the CoM frame (for details see [5]).

7.3.2 Foot acceleration task

The foot reference trajectories derived in Section 6.1 can be tracked by the following task space PD controller

$$\ddot{\mathbf{x}}_{f,\text{des}} = \ddot{\mathbf{x}}_{f,\text{ref}} - \lambda_{f,1}\lambda_{f,2}(\mathbf{x}_f - \mathbf{x}_{f,\text{ref}}) + (\lambda_{f,1} + \lambda_{f,2})(\dot{\mathbf{x}}_f - \dot{\mathbf{x}}_{f,\text{ref}}), \quad (7.16)$$

where $\lambda_{f,1}$ and $\lambda_{f,2}$ are the desired eigenvalues of the foot tracking dynamics. To map the foot accelerations to the selected optimization variables in \mathbf{u} , the following velocity transformation between task space and configuration space is considered:

$${}^0_h\dot{\mathbf{v}}_{0,f} = \begin{bmatrix} {}^0\dot{\mathbf{x}}_{0,f} \\ {}^0\boldsymbol{\omega}_{0,f} \end{bmatrix} = {}^0_h\mathbf{J}_{0,f}\dot{\mathbf{q}}, \quad (7.17)$$

where ${}^0_h\dot{\mathbf{v}}_{0,f}$ is the hybrid velocity introduced by Murray et al. [20], that combines the translational velocity ${}^0\dot{\mathbf{x}}_{0,f}$ and the spacial angular velocity ${}^0\boldsymbol{\omega}_{0,f}$ of the foot in a six-dimensional vector. The hybrid Jacobian ${}^0_h\mathbf{J}_{0,f}$ of the foot maps the generalized joint velocities $\dot{\mathbf{q}}$ to the hybrid velocity. Here, the spacial angular velocity ${}^0\boldsymbol{\omega}_{0,f}$ is chosen to be zero such that the foot always remains parallel to the ground. This assumption still yields satisfactory walking and running motions, however, in order to obtain a natural and human-like gait pattern, the rotation of the foot plays an important but challenging role, as for example rolling over the front edge of the foot significantly reduces the actuability of the robot. The integration of foot orientation reference trajectories is part of future work in this project.

By differentiating (7.17) the hybrid foot acceleration is obtained:

$${}^0_h\ddot{\mathbf{v}}_{0,f} = {}^0_h\dot{\mathbf{J}}_{0,f}\dot{\mathbf{q}} + {}^0_h\mathbf{J}_{0,f}\ddot{\mathbf{q}}. \quad (7.18)$$

With (7.16) the desired hybrid foot acceleration can be written as

$${}^0_h\ddot{\mathbf{v}}_{0,f,\text{des}} = \begin{bmatrix} \ddot{\mathbf{x}}_{f,\text{des}} \\ \dot{\boldsymbol{\omega}}_{f,\text{des}} \end{bmatrix}, \quad (7.19)$$

where $\dot{\boldsymbol{\omega}}_{f,\text{des}} = \mathbf{0}_{3 \times 1}$. By setting ${}^0_h\dot{\mathbf{v}}_{0,f} = {}^0_h\dot{\mathbf{v}}_{0,f,\text{des}}$ in (7.18), the foot acceleration task can be written as a function of the control input vector \mathbf{u} :

$$\underbrace{{}^0_h\ddot{\mathbf{v}}_{0,f,\text{des}} - {}^0_h\dot{\mathbf{J}}_{0,f}\dot{\mathbf{q}} - {}^0_h\mathbf{J}_{0,f}\ddot{\mathbf{q}}_{\text{MB}}}_{\mathbf{d}_{f,\text{des}}} = \underbrace{{}^0_h\mathbf{J}_{0,f}\mathbf{Q}}_{\mathbf{D}_f}\mathbf{u}, \quad (7.20)$$

which provides the desired foot acceleration task vector $\mathbf{d}_{f,\text{des}}$ and the corresponding task mapping matrix \mathbf{D}_f for the integration in the WBC framework.

7.3.3 Further tasks and task summary

Two more tasks, i.e. torso orientation and joint posture reference, are added to WBC framework, whose desired accelerations ${}^0\ddot{\boldsymbol{\omega}}_{0,\text{torso},\text{des}}$ and $\ddot{\mathbf{q}}_{\text{post},\text{des}}$, respectively, can be calculated similarly to foot acceleration tasks via PD controllers (for details see [5]). Table 7.1 summarizes all task mapping matrices and task vectors used in the optimization. Here, ${}^0_{\omega}\mathbf{J}_{0,\text{torso}}$ is the angular Jacobian of the torso and ${}^0_{\omega}\dot{\mathbf{J}}_{0,\text{torso}}$ its time derivative,

task	task mapping matrix \mathbf{D}_i	desired task vector $\mathbf{d}_{i,\text{des}}$
centr. momentum	$\begin{bmatrix} \mathbf{0}_{6 \times n_\tau} & {}^{\text{com}}\mathbf{A}_\rho \end{bmatrix}$	$\begin{bmatrix} \mathbf{F}_{\text{leg,des}} \\ \boldsymbol{\tau}_{\text{com,des}} \end{bmatrix}$
foot tracking	${}^0_h \mathbf{J}_{0,f} \mathbf{Q}$	${}^0_h \dot{\boldsymbol{\nu}}_{0,f,\text{des}} - {}^0_h \dot{\mathbf{J}}_{0,f} \dot{\mathbf{q}} - {}^0_h \mathbf{J}_{0,f} \ddot{\mathbf{q}}_{\text{MB}}$
torso orientation	$\mathbf{J}_{0,\text{torso}} \mathbf{Q}$	${}^0 \dot{\boldsymbol{\omega}}_{0,\text{torso,des}} - {}^0_\omega \dot{\mathbf{J}}_{0,\text{torso}} \dot{\mathbf{q}} - {}^0_\omega \mathbf{J}_{0,\text{torso}} \ddot{\mathbf{q}}_{\text{MB}}$
posture reference	$\mathbf{S}_{\text{post}} \mathbf{Q}$	$\ddot{\mathbf{q}}_{\text{post,des}} - \mathbf{S}_{\text{post}} \ddot{\mathbf{q}}_{\text{MB}}$
regularization	$\mathbf{I}_{n_u \times n_u}$	$\mathbf{0}$

Table 7.1: Summary of tasks for the whole-body controller displayed as task mapping matrices \mathbf{D}_i and desired task vectors $\mathbf{d}_{i,\text{des}}$ (adapted from [14]).

equivalent to the angular component of ${}^0_h \mathbf{J}_{0,f}$ and ${}^0_h \dot{\mathbf{J}}_{0,f}$ introduced in Section 7.3.2. The matrix \mathbf{S}_{post} selects the joints for the reference posture task. The regularization task acts as a damping term in the optimization and penalizes high control inputs \mathbf{u} . This ensures good behavior in the case of singularities and also dampens the nullspace that corresponds to the mapping of the linear contact forces to the contact wrenches (see (7.5)).

7.4 Optimization via a quadratic program

The task errors $\mathbf{D}_i \mathbf{u} - \mathbf{d}_{i,\text{des}}$, obtained by reformulating (7.10), are used to write the cost function G , which is minimized in the QP:

$$\begin{aligned}
 \min_{\mathbf{u}} \quad & G = \frac{1}{2} \sum_i \left((\mathbf{D}_i \mathbf{u} - \mathbf{d}_{i,\text{des}})^T \mathbf{W}_i (\mathbf{D}_i \mathbf{u} - \mathbf{d}_{i,\text{des}}) \right) \\
 \text{s.t.} \quad & \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}} \\
 & \mathbf{0} \leq \boldsymbol{\rho}
 \end{aligned} \tag{7.21}$$

Here, \mathbf{W}_i are diagonal weighting matrices for each task i and $\underline{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{\tau}}$ denote the lower and upper joint torque limits, respectively. The last inequality constraint ensures that the contact force magnitudes and corresponding contact wrenches are feasible. The solution to (7.21) provides the optimal control input vector \mathbf{u}^* that trades off the individual tasks depending on their weights and satisfies the torque limits and contact constraints.

Chapter 8

Whole-Body Simulation

Extending the existing simulation environment of the humanoid robot Toro, the reference trajectories from Chapters 5 and 6 are generated in Matlab and imported into the Matlab Simulink model, which is linked to a simulation platform called OpenHRP (Open Architecture Human-centered Robotics Platform) for visualization. It is a virtual humanoid robot platform consisting of a multibody simulator with contact and collision computation between arbitrary polyhedral objects [16]. Several simulations are performed to test the performance of the proposed trajectory generation with the whole-body controller introduced in Chapter 7. The results for a trajectory with five gait sequences alternating between walking and running are presented below.

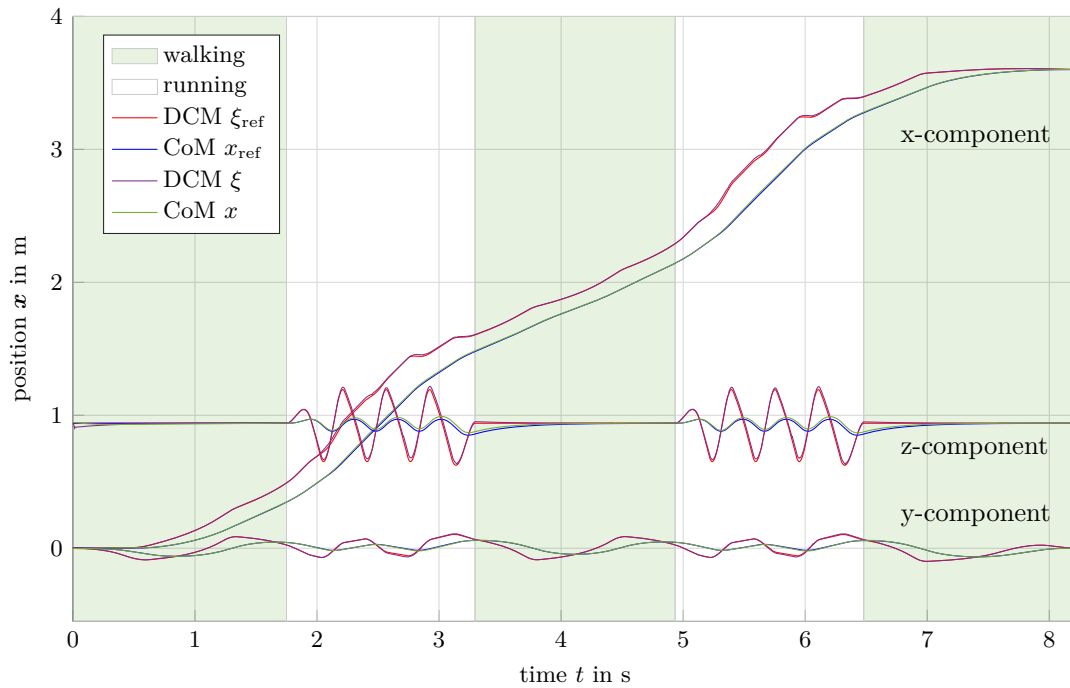


Figure 8.1: Simulation of the humanoid robot Toro [8] following a trajectory consisting of five consecutive gait sequences with 0.25 m step length, double support time $T_{\text{DS}} = 0.12$ s and single support time $T_{\text{SS}} = 0.6$ s during walking and 0.4 m step length and stance time $T_s = 0.12$ s during running.

Fig. 8.1 shows the DCM and CoM reference trajectories compared to the estimated values calculated in the whole-body simulation. Overall, the trajectories are tracked well. The DCM and CoM tracking shows minimal deviations in the walking sequences and slightly larger deviations in the running sequences. The DCM root-mean-square tracking error denotes $\varepsilon_{\xi,x} = 6.3$ mm, $\varepsilon_{\xi,y} = 2.9$ mm and $\varepsilon_{\xi,z} = 12.1$ mm in the respective direction. The CoM tracking error results from the DCM error, but is with $\varepsilon_x = 4.8$ mm, $\varepsilon_y = 2.1$ mm and $\varepsilon_z = 10.0$ mm slightly smaller than the DCM error due to smaller variations of the CoM trajectory. The fast vertical dynamics during running causes with $\varepsilon_{\xi,z,max} = 36.0$ mm and $\varepsilon_{z,max} = 23.2$ mm the largest DCM and CoM tracking error, respectively. It can be observed that the robot does not completely reach the desired apex height during the flight phase.

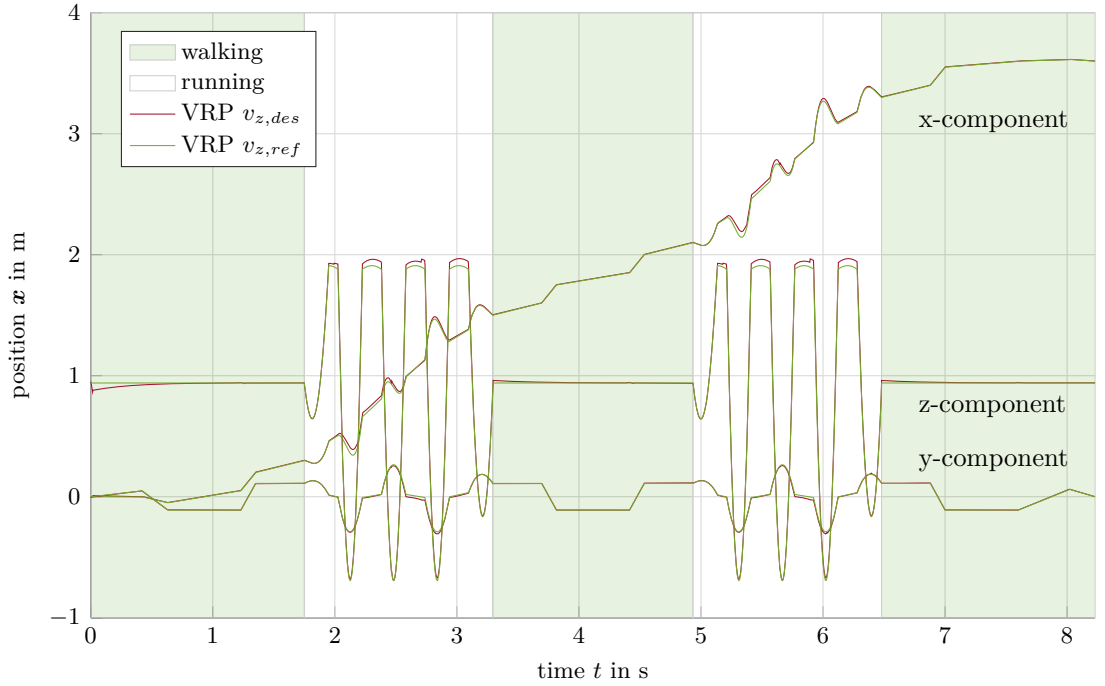


Figure 8.2: Reference and desired VRP trajectories. The VRP root-mean-square tracking error denotes $\varepsilon_{v,x} = 12.6$ mm, $\varepsilon_{v,y} = 5.8$ mm and $\varepsilon_{v,z} = 24.1$ mm in the respective direction.

By looking at the desired VRP trajectory in Fig. 8.3 it can be seen that it deviates from the reference VRP trajectory mostly in the flight phase. Influenced by the contact model and a linear fade-in and fade-out of the contact condition for robustness, a slightly lower force than required to achieve the desired apex height in the following flight phase can be applied in the stance phase. Since no external force can be generated here, the deviation exists for the entire flight phase, which significantly increases the mean tracking error. For the same reason, there are also minor deviations in the horizontal components of the desired VRP trajectory.

The right leg joint torques of the humanoid robot Toro during the same simulation are displayed in Fig. 8.3. Due to the increased load in the running sequence, the torque limits are disabled in the simulation. The joint torques are highest in the stance phases highlighted in yellow. Especially in the short contact phase during running, the peak torques in the hip pitch joint are significantly increased compared to humans. The

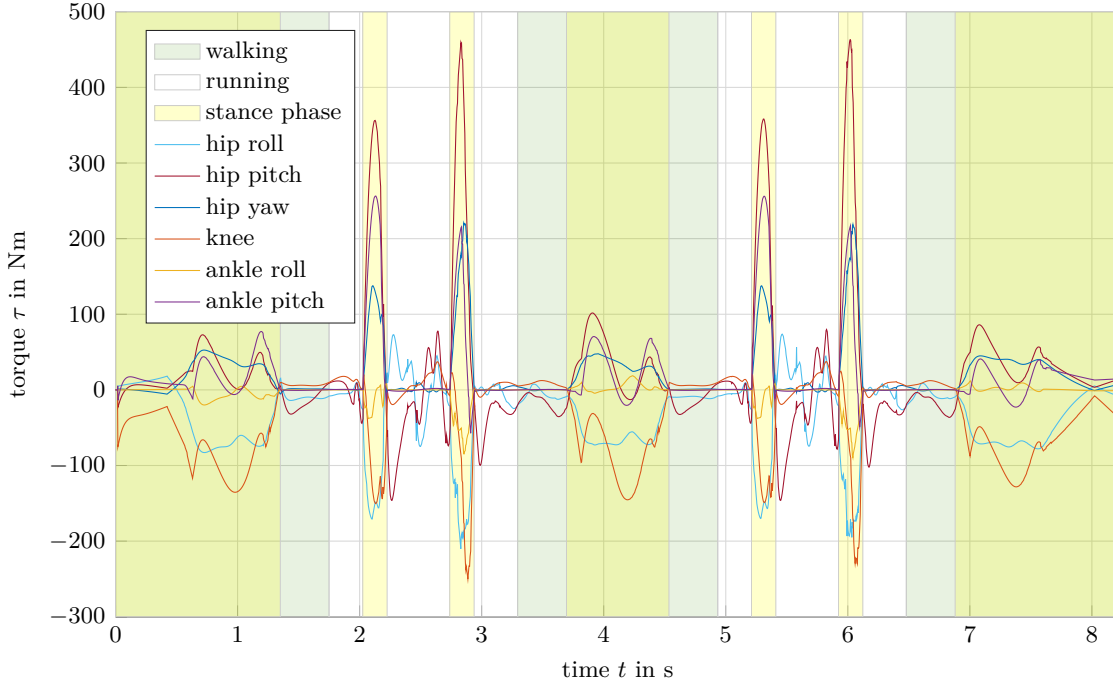


Figure 8.3: Right leg joint torques of the humanoid robot Toro in simulation

torques in the knee and ankle joint are roughly comparable to the load during human walking and running [2].

The increased torque in the hip most likely results from the relatively strong forward flexion of the upper body during the stance phase of running (see Fig. 8.4 at $t = 0.8$ s). Increasing the weighting of the torso orientation task in the QP can improve this slightly. However, it does not have much influence in the stance phase and has the additional disadvantage that the upper body returns to the upright position unnaturally fast after the stance phase.

Fig. 8.4 shows the transition from walking to running of Toro in simulation for time steps of 0.2 s, where the motion of the center of mass is approximated as a solid blue line for walking and a dashed blue line for running. The single support time is $T_{SS} = 0.6$ s for walking and $T_s = 0.2$ s for running. The plot starts at the end of the last double support phase of walking at 0 s. The transition of the gait algorithms occurs in the last single support phase of walking. For this phase to have the same time as the previous single support phase, the last single support time for walking $T_{SS,last}$ is calculated as

$$T_{SS,last} = T_{SS} - T_s, \quad (8.1)$$

where T_{SS} is the walking single support time and T_s is the stance time during running. Here, the gait transition occurs at two-thirds of the last single support phase (0.4 s). The robot then pushes off slightly to reach the desired apex height in the subsequent first short flight phase. The first touchdown occurs at about 0.7 s. In the middle of the first full stance phases of running at 0.8 s, the leg forces are at their maximum (see Fig. 6.7). The following flight phase is longer than the first one because it has a longer ascending part. At 1.0 s the robot is in full flight.

The transition from running to walking is shown in Fig. 8.5 and starts at 0 s in the second to last flight phase. The single support and stance times are chosen as in the

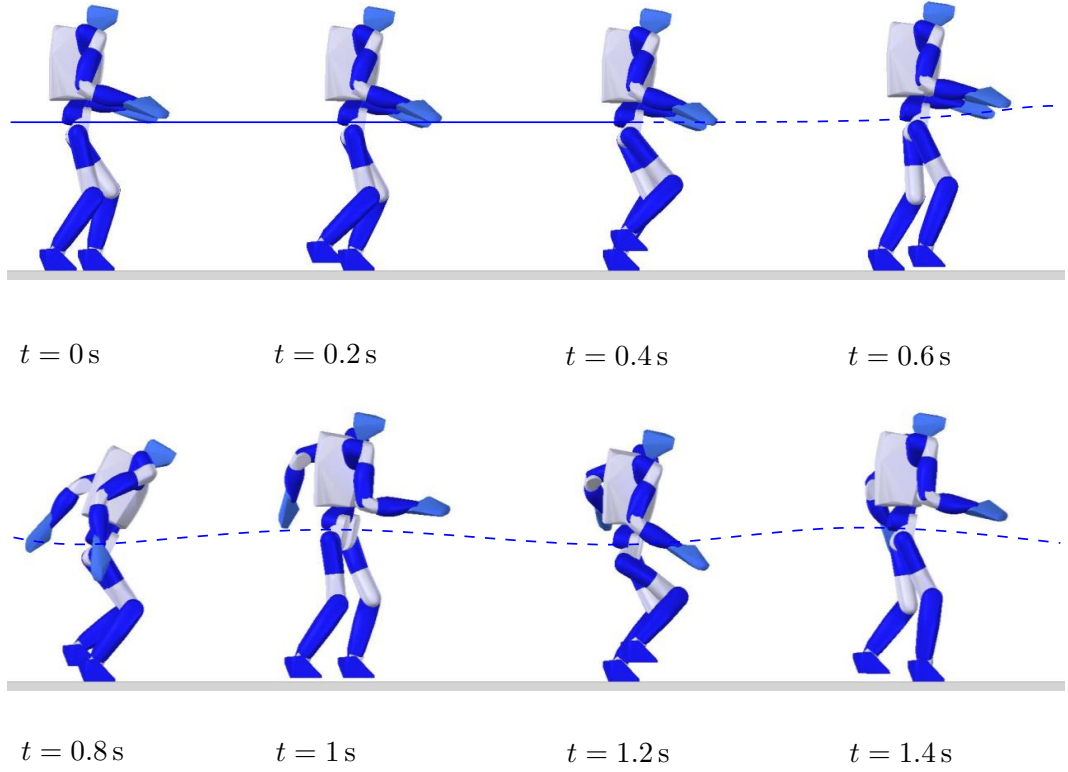


Figure 8.4: Time Series of the humanoid robot Toro [8] in simulation during the walk-to-run (W2R) transition with CoM motion indicated (continuous line: walking, dashed line: running).

W2R transition in Fig. 8.4. The final touchdown of the running sequence takes place at 0.4s and the transition of the gait algorithms occurs one stance time $T_s = 0.2$ s later at 0.6s. The duration of the first single support phase of walking $T_{SS,first}$ is calculated equivalently to (8.1) as

$$T_{SS,first} = T_{SS} - T_s, \quad (8.2)$$

Thus, the robot is back in the first double support phase 0.4s later at 1s. Since the first VRP of the walking sequence was chosen at the original walking height (Δz_{vrp} above the ground), the CoM trajectory converges to this value according to the naturally stable CoM dynamics (3.2).

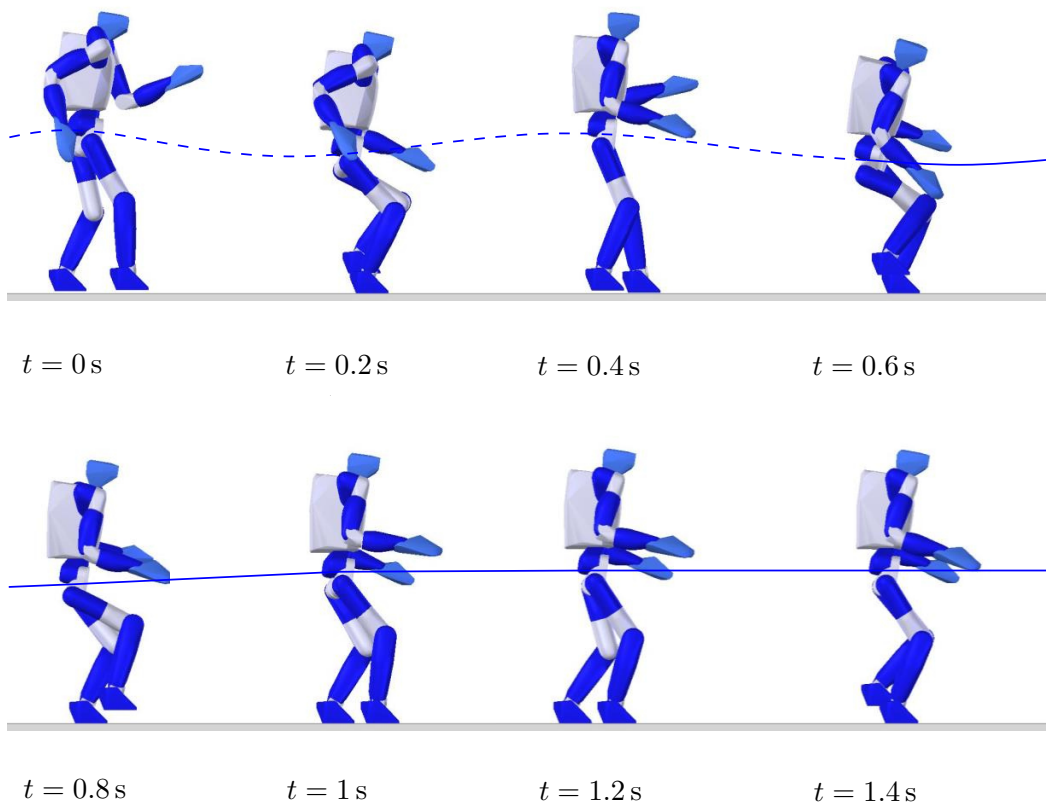


Figure 8.5: Time Series of the humanoid robot Toro [8] in simulation during the run-to-walk (R2W) transition with CoM motion indicated (continuous line: walking, dashed line: running).

Chapter 9

Conclusion and Future Work

The goal of this work was to plan smooth and consistent transitions between walking and running that enable faster locomotion speeds for humanoid robots. At first related work and an overview of the existing trajectory generation algorithms from walking and running were presented. By combining the two algorithms, a new trajectory generation framework was developed that generates continuous trajectories for multiple walking and running sequence and their transitions.

Due to the separate calculation of the horizontal and vertical components of the trajectory in the running sequence, the trajectory generation was split into two parts, which were calculated sequentially. First, the vertical dynamics were solved in a forward recursion from one gait sequence to the next. Then, due to the coupling of different state variables between walking and running, the complete trajectory in the horizontal direction was generated in a single matrix calculation. The assembly of the global matrix equation was shown in detail.

The trajectory generation was evaluated for different design parameters that influence the distribution of leg force over the foot. Furthermore, the applicability for different ground characteristics i.e. walking and running over stepping stones was shown. The different control strategies for walking and running were unified by integrating the proposed trajectory generation into an inverse dynamics based whole-body controller.

In the last chapter, the presented approaches were validated in simulations with the humanoid robot Toro and stable transitions between walking and running were obtained.

Future research will focus on integrating the trajectory generation framework in the existing simulation environment in Matlab Simulink so that the trajectory is re-planned online in each time step based on the current measured values and can thus react to disturbances. Moreover, foot orientation reference trajectories will be planned and integrated into the whole-body control framework to obtain a more natural and human-like gait pattern. Furthermore, the gait parameters like stance times, apex height in the flight phase and step length are adjusted depending on the current locomotion velocity to also create a more natural gait pattern.

These future ideas, combined with the developed framework for generating consistent trajectories, represent an important step toward robotic bipedal locomotion in experiments with smooth transitions between standing, walking and running.

List of Figures

1.1	Simulation of the humanoid robot Toro [8] during the walk-to-run (W2R) transition (time series with 0.2s interval). The robot is displayed in the walking sequence (left), the transition stance phase (center) and in the flight phase during running (right).	2
3.1	Point correlations for general robot dynamics (adapted from [9]).	6
4.1	Preview of upcoming flight and stance phases (planar sketch) - used for the design of boundary conditions [10].	14
4.2	Effect of $\tilde{p}_{\chi,i}$ on force ray focusing (lines of action) at the contact point [10].	18
5.1	Correlations between walking and running quantities.	21
5.2	Outline of the vertical planning computation flow.	26
5.3	Heatmap of global constraint mapping matrix \mathbf{G} and the global target mapping matrix \mathbf{T} for five consecutive gait sequences with three footsteps each consisting of 12 waypoints in each walking phase and 14 waypoints in each running sequence, i.e. 64 waypoints in total. Submatrices according to the coloring scheme in (5.46) with non-zero entries colored in gray. . .	39
5.4	Heatmap of the matrix $(\mathbf{I} - \mathbf{G})^{-1}$ with same settings as in Fig. 5.3. The values of each cell \square are displayed on a linear scale from $\square = 0$ (white) to $ \square \geq 1$ (dark gray).	40
5.5	Heatmap of the global target mapping matrix \mathbf{U} with same settings as in Fig. 5.3. The values of each cell \square are displayed on a linear scale from $\square = 0$ (white) to $ \square \geq 1$ (dark gray).	41
6.1	Top view of a trajectory consisting of five consecutive gait sequences with 0.25m step length, double support time $T_{DS} = 0.12s$ and single support time $T_{SS} = 0.6s$ during walking (eCMP stationary in foot center during single support) and 0.4m step length and stance time $T_s = 0.12s$ during running (eCMP focus point on the foot plane).	45
6.2	Top view of a trajectory consisting of five consecutive gait sequences with 0.25m step length, double support time $T_{DS} = 0.12s$ and single support time $T_{SS} = 0.6s$ during walking (eCMP moves from the heel of the foot) and 0.4m step length and stance time $T_s = 0.12s$ during running (eCMP focus point 0.3m vertically below the foot plane).	46

6.3	Side view of the trajectory in x- and y-direction with the eCMP stationary in the foot center during walking single support and the eCMP focus on the foot plane during running.	47
6.4	Side view of the trajectory in x- and y-direction with eCMP heel-to-toe movement during walking single support and the eCMP focus point 0.3 m vertically below the foot plane during running.	47
6.5	Time diagram of the horizontal trajectory consisting of five consecutive gait sequences with 0.25 m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking and 0.4 m step length and stance time $T_s = 0.12$ s during running.	48
6.6	Time diagram of the vertical trajectory consisting of five consecutive gait sequences with 0.25 m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking and 0.4 m step length and stance time $T_s = 0.12$ s during running.	48
6.7	Leg forces over time of the humanoid robot Toro (79.2 kg). Step lengths and timings according to Fig. 6.6.	49
6.8	Bipedal point mass model walking (continuous line) and running (dashed line) over 3D stepping stones with CoM (blue), left foot (black) and right foot (red) trajectories.	52
7.1	Outline of CoM frame, contact frame and polyhedral convex cone approximation (adapted from [5]).	54
8.1	Simulation of the humanoid robot Toro [8] following a trajectory consisting of five consecutive gait sequences with 0.25 m step length, double support time $T_{DS} = 0.12$ s and single support time $T_{SS} = 0.6$ s during walking and 0.4 m step length and stance time $T_s = 0.12$ s during running.	59
8.2	Reference and desired VRP trajectories. The VRP root-mean-square tracking error denotes $\varepsilon_{v,x} = 12.6$ mm, $\varepsilon_{v,y} = 5.8$ mm and $\varepsilon_{v,z} = 24.1$ mm in the respective direction.	60
8.3	Right leg joint torques of the humanoid robot Toro in simulation	61
8.4	Time Series of the humanoid robot Toro [8] in simulation during the walk-to-run (W2R) transition with CoM motion indicated (continuous line: walking, dashed line: running).	62
8.5	Time Series of the humanoid robot Toro [8] in simulation during the run-to-walk (R2W) transition with CoM motion indicated (continuous line: walking, dashed line: running).	63

List of Tables

5.1	Vertical transition boundary conditions for three consecutive gait phases.	22
5.2	Vertical boundary conditions for different running stance phases displayed as boundary condition vector $\mathbf{b}_{z,i}$, boundary condition mapping matrix $\mathbf{B}_{z,i}$ and the nullspace target.	24
5.3	Horizontal transition boundary conditions for three consecutive gait phases.	27
5.4	Horizontal boundary condition matrix $\tau_{\chi,i}$ for different running stance phases.	28
5.5	Polynomial coefficient mapping matrix \mathbf{S}_i for different running stance phases.	29
5.6	Constraint mapping matrix \mathbf{A}_i for different running stance phases.	30
6.1	Analytical solution of the foot position interpolation and corresponding boundary conditions in the horizontal and vertical direction.	51
7.1	Summary of tasks for the whole-body controller displayed as task mapping matrices \mathbf{D}_i and desired task vectors $\mathbf{d}_{i,\text{des}}$ (adapted from [14]).	58

References

- [1] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11-12):1217–1227, 1989. ISSN 0021-9290. doi: 10.1016/0021-9290(89)90224-8. URL <https://www.sciencedirect.com/science/article/pii/0021929089902248>.
- [2] G. Cappellini, Y. P. Ivanenko, R. E. Poppele, and F. Lacquaniti. Motor patterns in human walking and running. *Journal of Neurophysiology*, 95(6):3426–3437, 2006. ISSN 0022-3077. doi: 10.1152/jn.00081.2006.
- [3] S. Cotton, J. C. Godowski, N. R. Payton, M. Vignati, C. Schmidt-Wetekam, C. Black, et al. Multi-legged running robot, 2016.
- [4] B. Dadashzadeh, H. R. Vejdani, and J. Hurst. From template to anchor: A novel control strategy for spring-mass running of bipedal robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2566–2571. IEEE, 92014. ISBN 978-1-4799-6934-0. doi: 10.1109/IROS.2014.6942912.
- [5] J. Engelsberger. *Combining reduced dynamics models and whole-body control for agile humanoid locomotion*. Dissertation, Technische Universität München, München, 2016.
- [6] J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schaffer, and G. Hirzinger. Bipedal walking control based on capture point dynamics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4420–4427. IEEE, 092011. ISBN 978-1-61284-456-5. doi: 10.1109/IROS.2011.6094435.
- [7] J. Engelsberger, P. Kozlowski, and C. Ott. Biologically inspired deadbeat control for running on 3d stepping stones. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 1067–1074. IEEE, 112015. ISBN 978-1-4799-6885-5. doi: 10.1109/HUMANOIDS.2015.7363501.
- [8] J. Engelsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schaffer. Overview of the torque-controlled humanoid robot toro. In *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2014)*, pages 916–923, Piscataway, NJ, 2014. IEEE. ISBN 978-1-4799-7174-9. doi: 10.1109/HUMANOIDS.2014.7041473.
- [9] J. Engelsberger, C. Ott, and A. Albu-Schaffer. Three-dimensional bipedal walking control based on divergent component of motion. *IEEE Transactions on Robotics*, 31(2):355–368, 2015. ISSN 1552-3098. doi: 10.1109/TRO.2015.2405592.

- [10] J. Engelsberger, P. Kozlowski, C. Ott, and A. Albu-Schaffer. Biologically inspired deadbeat control for running: From human analysis to humanoid control and back. *IEEE Transactions on Robotics*, 32(4):854–867, 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2581199.
- [11] J. Engelsberger, G. Mesesan, and C. Ott. Smooth trajectory generation and push-recovery based on divergent component of motion. In *IROS Vancouver 2017*, pages 4560–4567, Piscataway, NJ, 2017. IEEE. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8206324.
- [12] H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings. Biological sciences*, 273(1603):2861–2867, 2006. ISSN 0962-8452. doi: 10.1098/rspb.2006.3637.
- [13] B. Henze, M. A. Roa, and C. Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *The International Journal of Robotics Research*, 35(12):1522–1543, 2016. ISSN 0278-3649. doi: 10.1177/0278364916653815.
- [14] J. Engelsberger, G. Mesesan, A. Werner, and C. Ott. Torque-based dynamic walking - a long way from simulation to experiment. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 440–447, 2018. ISBN 2577-087X. doi: 10.1109/ICRA.2018.8462862.
- [15] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings / 2003 IEEE International Conference on Robotics and Automation*, pages 1620–1626, Piscataway, NJ, 2003. IEEE Service Center. ISBN 0-7803-7736-2. doi: 10.1109/ROBOT.2003.1241826.
- [16] F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, and K. Yamane. Open architecture humanoid robotics platform. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pages 24–30. IEEE, 2002. ISBN 0-7803-7272-7. doi: 10.1109/ROBOT.2002.1013334.
- [17] C. R. Lee and C. T. Farley. Determinants of the center of mass trajectory in human walking and running. *Journal of Experimental Biology*, 201(21):2935–2944, 1998. ISSN 0022-0949. doi: 10.1242/jeb.201.21.2935.
- [18] H. R. Martinez and J. P. Carbajal. From walking to running a natural transition in the slip model using the hopping gait. In *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO 2011)*, pages 2163–2168, Piscataway, NJ, 2011. IEEE. ISBN 978-1-4577-2138-0. doi: 10.1109/ROBIO.2011.6181612.
- [19] G. Mesesan, J. Engelsberger, C. Ott, and A. Albu-Schaffer. Convex properties of center-of-mass trajectories for locomotion based on divergent component of motion. *IEEE Robotics and Automation Letters*, 3(4):3449–3456, 2018. ISSN 2377-3766. doi: 10.1109/LRA.2018.2853557.
- [20] R. M. Murray, Z. Li, and S. Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, Boca Raton, Fla., 1994. ISBN 0849379814. URL <http://www.loc.gov/catdir/enhancements/fy0730/93033167-d.html>.

- [21] D. E. Orin, A. Goswami, and S.-H. Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2-3):161–176, 2013. ISSN 0929-5593. doi: 10.1007/s10514-013-9341-4.
- [22] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207. IEEE, 122006. ISBN 1-4244-0199-2. doi: 10.1109/ICHR.2006.321385.
- [23] J. Rummel, Y. Blum, and A. Seyfarth. From walking to running. In W. Brauer, R. Dillmann, J. Beyerer, C. Stiller, J. M. Zöllner, and T. Gindele, editors, *Autonome Mobile Systeme 2009*, Informatik aktuell, pages 89–96. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-10283-7. doi: 10.1007/978-3-642-10284-4\textunderscore{12}.
- [24] G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, 2010. ISSN 1941-014X. doi: 10.1109/TMECH.2009.2035112.
- [25] M. Shahbazi, G. Lopes, and R. Babuska. Automated transitions between walking and running in legged robots. *IFAC Proceedings Volumes*, 47(3):2171–2176, 2014. ISSN 14746670. doi: 10.3182/20140824-6-ZA-1003.01717.
- [26] T. Sugihara, Y. Nakamura, and H. Inoue. Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pages 1404–1409. IEEE, 2002. ISBN 0-7803-7272-7. doi: 10.1109/ROBOT.2002.1014740.
- [27] T. Takenaka, T. Matsumoto, and T. Yoshiike. Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009*, pages 1084–1091, Piscataway, NJ, 2009. IEEE. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354662.
- [28] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura. A closed-form solution for real-time zmp gait generation and feedback stabilization. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 936–940. IEEE, 112015. ISBN 978-1-4799-6885-5. doi: 10.1109/HUMANOIDS.2015.7363473.
- [29] 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*, 29(5):1114–1124, 2013. ISSN 1552-3098. doi: 10.1109/TRO.2013.2263718.