

Applying the Divergent Component of Motion Method for Quadrupedal Locomotion to a Robot with Series Elastics Actuators

Scientific work to obtain the degree of

Master of Science

in the Department of Mechanical Engineering at the Technical
University of Munich.

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

Advisors: Dr.-Ing. Johannes Engelsberger
George-Adrian Mesesan, M.Sc.
Cosimo Della Santina, PhD

Author: Markus Wagner

Submission date: January 15, 2021

Abstract

Many approaches for the trajectory generation for dynamical locomotion were developed in order to handle the hybrid and non-linear nature of locomotion of compliant four-legged robots. However, applied methods such as Central Pattern Generator, Finite State Machine, model-predictive control or non-linear optimization approaches do not allow a direct and analytical generation of smoothed CoM trajectories. The *Divergent Component of Motion* method is a trajectory generation method which is already successfully applied to bipedal robots. With this method the direct generation of continuous CoM trajectories of quadrupedals could provide advanced capabilities for the control of dynamical locomotion in challenging terrain. The aim of this thesis is to apply the DCM method to a compliant quadrupedal robot in order to generate CoM and limb trajectories which are executable for this robot. For this purpose a sequence of a compact stance description is used to describe the gait pattern of trotting, pacing and walk without any flight phases. Based on the stance description the geometric center point of the support polygon is calculated for a heuristic VRP placement rule. This placement determines the VRP waypoints for the robot Bert equipped with planar legs. These waypoints are used with the efficient DCM-method to generate the CoM references. Further, these references are used in combination with a cubic interpolation of the limb movement to generate the joint references by a resolved motion rate approach. The implemented DCM-based trajectory generation was tested in a simulation framework for the trotting and walking Bert. The joints were controlled by a PD-controller during the simulation. In conclusion, the multi-body simulation results show that DCM can be used to generate executable references for four-legged compliant robots. However, these results still need to be validated by experiments on the robot Bert.

Keywords: quadrupedal locomotion, divergent component of motion, trajectory generation, serial elastic actuators, dynamical locomotion

Viele Ansätze zur Trajektorien-Generierung für dynamische Lokomotion wurden entwickelt, um die hybride und nichtlineare Natur der Lokomotion von elastischen vierbeinigen Robotern zu berücksichtigen. Angewandte Methoden wie Central Pattern Generator, Finite State Machine, modellprädiktive Steuerung oder nichtlineare Optimierungsansätze erlauben jedoch keine direkte und analytische Generierung von glatten CoM-Trajektorien. Die *Divergent Component of Motion*-Methode stellt eine Methode zur Trajektorien-Generierung dar, die bereits erfolgreich bei bipedalen Robotern eingesetzt wird. Die mit dieser Methode einhergehende direkte Generierung von kontinuierlichen CoM-Trajektorien von vierbeinigen Robotern könnte erweiterte Möglichkeiten für die Steuerung dynamischer Lokomotion in anspruchsvollem Gelände bieten. Ziel dieser Arbeit ist es, die DCM-Methode auf einen elastischen vierbeinigen Roboter anzuwenden, um CoM- und End-Effektor-Trajektorien zu generieren, welche für diesen Roboter ausführbar sind. Zu diesem Zweck wurde eine kompakte Standbeschreibung des Roboters mittels der Kontaktpunkte in Serie verwendet, welche das Gangbild für *trab*, *pacing* oder *walk* ohne jegliche Flugphasen beschreibt. Basierend auf der Standbeschreibung wird der geometrische Mittelpunkt des Stützpolygons für eine heuristische VRP-Platzierungsregel berechnet. Diese Platzierung bestimmt die VRP-Wegpunkte für den mit planaren Beinen ausgestatteten Roboter Bert. Diese Wegpunkte werden in der effizienten DCM-Methode verwendet, um die CoM-Referenzen zu erzeugen. Weiterhin werden diese Referenzen in Kombination mit einer kubischen Interpolation der End-Effektoren im Raum verwendet, um die Gelenkreferenzen mit einem Resolved Motion Rate-Ansatz zu generieren. Das implementierte DCM-Verfahren wurde in einer Simulationsumgebung für den trabenden und gehenden Roboter Bert getestet. Die Gelenke wurden während der Simulation durch einen PD-Controller gesteuert. Zusammenfassend zeigen die Ergebnisse der Mehrkörper-Simulation, dass die DCM-Methode verwendet werden kann, um ausführbare Referenzen für vierbeinige, elastische Roboter zu erzeugen. Diese Ergebnisse gilt es jedoch noch durch Experimente auf Roboter Bert zu validieren.

Acknowledgement

I would like to thank Prof. Dr.-Ing. Alin Albu-Schäffer for the opportunity to undertake my master's thesis at the Institute of Robotics and Mechatronics at DLR.

I would like to express my gratitude to my advisor, Dr.-Ing. Johannes Engelsberger, who guided me throughout this project with invaluable supervision. I wish to show my appreciation to my advisors George Mesesan and Cosimo Della Santina, PhD for their mentorship and their helpful support.

I wish to acknowledge the help provided by the doctoral candidates as well as by technical and support staff in the Institute of Robotics and Mechatronics at DLR.

Contents

Abstract	I
Acknowledgement	III
Contents	V
List of Tables	VI
List of Figures	VIII
List of Symbols	XII
List of Abbreviations	XIII
1 Introduction	1
1.1 Motivation	1
1.2 State of the Art	1
2 Fundamentals	4
2.1 Notation	4
2.2 Basics of Legged Robot Locomotion	5
2.3 Models of Bipedal Locomotion	6
2.4 Models for Quadrupedal Locomotion	8
2.5 Divergent Component of Motion Method	9
2.6 The compliant quadrupedal Bert	16
3 From Bipedal to Quadrupedal Locomotion	18
3.1 Problem Formulation	18
3.2 Gait Definitions	21
3.3 Trajectory Generation	26
3.4 Joint Reference Generation	28
4 Implementation of the Method	30
4.1 Robot model description and Framework	30
4.2 Motion Planning	31
4.3 Motion Control	32
5 Simulation	34
5.1 Simulation base	34
5.2 Simulation results of the Trotting Gait	35
5.3 Simulation results of the Walk Gait	48

6	Discussion and Conclusion	55
6.1	Summary and Discussion	55
6.2	Conclusion	58
	References	62
A	Appendix	63
A.1	Robot Leg	63
A.2	VRP placement	64
A.3	Simulation	66
	A.3.1 Further Information of the simulation Bert model	66
	A.3.2 Joint Reference	66
	A.3.3 Further results	68
A.4	Data folder	72
A.5	Applied Software	73
	Statement of Originality	75

List of Tables

2.1	Kinematic and dynamic parameters of the robot Bert	17
3.1	Parameters for example trajectory generation	27
3.2	Parameters for example trajectory generation in figure 3.7	28
4.1	Values of \mathbf{K}_P and \mathbf{K}_D for the robot Bert	33
5.1	Parameters for selected simulation runs	39

List of Figures

2.1	Support polygon of quadrupedal robot	8
2.2	Sketch of the DCM one point mass	10
2.3	Calculation flow of the DCM method.	15
2.4	Kinemattic parameters of robot bert	16
2.5	applied robot system Bert	17
3.1	schematic definition of static and dynamical locomotion	19
3.2	Definitions and Assumption on quadrupedal robot for DCM	20
3.3	gait definition static walk	22
3.4	gait definition trotting walk	24
3.5	gait definition pacing	25
3.6	Reference example with VRP degree 1	26
3.7	Reference example with VRP degree 5	27
4.1	Overview of the extended Simulink model	30
5.1	Plot of the record of the walk-VRP of Run 40	36
5.2	Plot of the record of the trot-DCM of Run 40	37
5.3	Plot of the record of trot-CoM of Run 40	38
5.4	Records of the VRP, DCM and whole-body CoM in the first and third direc- tion of Run 40	39
5.5	Records of the robot limbs in Run 40	40
5.6	Records of the robot's leg joints in Run 40	41
5.7	Sequence of stills of the simulation run 65 for the first three steps	42
5.8	Plot of the record of walk-VRP of Run 65	43
5.9	Plot of the record of walk-DCM of Run 65	43
5.10	Plot of the record of walk-CoM of Run 65	44
5.11	Records of the VRP, DCM, CoM in the first and third direction of the Run 65	45
5.12	Detail of plot 5.11	45
5.13	Records of the robot limbs in Run 65	46
5.14	Records of the robot leg joints in Run 65	47
5.15	Plot of the record of walk-VRP of Run 46	48
5.16	Plot of the record of walk-DCM of Run 46	49
5.17	Plot of the record of walk-CoM of Run 46	50
5.18	Records of the VRP, DCM and CoM in the first and third direction of Run 40	51
5.19	Records of the robot limbs in Run 46	52
5.20	Records of the robot leg joints in Run 46	53
5.21	Sequence of stills of the simulation run 46 for the first walk cycle of the four legs.	54

A.1	Planar leg of the robot Bert	63
A.2	Desired joint velocity resulting by the implemented method	67
A.3	Desired joint angles resulting by the implemented method	67
A.4	Plot of the record of walk-VRP of Run 46 with ideal VRP placement	68
A.5	Plot of the record of walk-DCM of Run 46 with ideal VRP placement	69
A.6	Plot of the record of walk-CoM of Run 46	70
A.7	Records of the robot limbs in Run 46 with ideal VRP placement	71
A.8	Records of the robot leg joints in Run 46 with ideal VRP placement	72

List of Symbols

$[\mathbf{r} \times]$	skew operator for a three-dimensional vector	
$\ddot{\mathbf{x}}$	acceleration of the CoM	m/s^2
Δ_x	matrix of the CoM Δ coefficients at $t = T_\varphi$	
$\dot{\mathbf{q}}_c$	joint velocity in the configuration space of a robot	
$\dot{\mathbf{x}}_d$	velocity in the world frame	m/s
Γ_x	matrix of the CoM Γ coefficients at $t = T_\varphi$	
Γ_ξ	matrix of the DCM Γ coefficients at $t = 0$	
ψ_b	position vector of the cartesian rotational base DOF	rad
τ	vector of the joint torques	Nm
τ_d	desired motor torques	Nm
τ_{motor}	vector of motor torques	Nm
θ	vector of motor positions	rad
θ_d	desired motor position	rad
Σ	sequence of stances (for motion planning)	
Ξ	matrix of the waypoints for the DCM reference	
ξ	DCM point as position vector	m
A_γ	matrix of the DCM γ coefficients	
A_ξ	matrix of the DCM α coefficients at $t = 0$	
A_x	matrix of the CoM α coefficients at $t = T_\varphi$	
$A_{\alpha\beta}$	matrix of the DCM α and β coefficients	
$A_{\xi,0}$	matrix of the DCM start points	
A_{back}	matrix for the backward DCM iteration	
A_{tc}	matrix for DCM terminal constraints	
B	matrix of constant and diagonal motor inertia	
B_x	matrix of the CoM β coefficients at $t = T_\varphi$	
B_ξ	matrix of the DCM β coefficients at $t = 0$	
C	vector of the coriolis and centrifugal effects	N
D	diagonal, square regularization matrix	
e_{trunk}	unit vector of the trunk frame	

$\mathbf{f}_{C,i}$	contact force vector of the i-th limb	
\mathbf{F}_{CoM}	total force acting on the CoM	N
$\mathbf{F}_{ext,i}$	the i-th external force vector acting on the CoM	N
\mathbf{F}_{ext}	sum of the external forces on the CoM	N
\mathbf{g}	gravity vector in the world frame	m/s^2
\mathbf{g}_σ	geometric center of a support polygon of a stance	m
\mathbf{J}^+	pseudo inverse	
$\mathbf{J}_{C,i}$	i-th contact Jacobian matrix of the i-th limb in contact	
\mathbf{K}_1	diagonal, square gain matrix	
\mathbf{K}_s	symmetric (diagonal and with entries of k) spring stiffness matrix	
\mathbf{M}	inertia matrix of the robot	
\mathbf{p}_c	position vector of contact point	m
\mathbf{q}_a	vector of actuated joints of a robot	
\mathbf{R}_c	contact orientation matrix	
\mathbf{r}_f	position vector of the desired foot point	m
\mathbf{r}_i	the i-th foot point position vector $i = 1..4$	m
\mathbf{r}_b	position vector of the cartesian translation base DOF	m
\mathbf{r}_{eCMP}	position of eCMP	m
$\mathbf{r}_{f,i}$	the i-th foot point position vector	m
\mathbf{r}_{fid}	desired foot point position in the world	m
\mathbf{r}_{ZMP}	position vector of the ZMP	m
\mathbf{S}_0	matrix to select the start points of the segments	
\mathbf{s}	vector in a frame	
\mathbf{S}_T	matrix to select the end points of the segments	
\mathbf{V}	matrix of the VRP waypoint sequence	
\mathbf{v}	position of the VRP	m
$\mathbf{v}_{\varphi,0}$	position vector of the start VRP	m
\mathbf{v}_{wp}	sequence of VRP waypoint	
\mathbf{W}	diagonal, square weighting matrix	
\mathbf{X}	matrix of the waypoints for the CoM reference	
\mathbf{x}_d	desired CoM position	m
$\Delta \mathbf{h}_{CoM}$	offset vector of the trunk frame origin to the CoM	m
$\Delta \mathbf{r}_f$	feedback of the error signal	m
Δh_{step}	desired step height	m

Δx_{step}	foot point displacement in trunk forward direction	m
$\dot{\mathbf{q}}_d$	desired joint velocity	rad/s
$\dot{\mathbf{r}}_{fi_d}$	desired velocity of the foot point	m/s
$\{P\}$	arbitrary frame with the coordinates p_1, p_2, p_3	
$\{W\}$	world frame with the coordinates t_1, t_2, t_3	
σ	stance of teh robot	
Δz_{vrp}	desired CoM heigth	m
${}^{\Xi}\mathbf{C}_V$	VRP calcualtion matrix for the DCM waypoint computation	
${}^{\Xi}\mathbf{c}_{\xi}$	DCM constraint vector for the DCM waypoint computation	
${}^X\mathbf{C}_V$	VRP calcualtion matrix for the CoM waypoint computation	
${}^X\mathbf{c}_x$	CoM constraint vector for the CoM waypoint computation	
${}^X\mathbf{c}_{\xi}$	DCM constraint vector for the CoM waypoint computation	
a	leg segment length	m
b	time constant of the DCM dynamics	s
c_l	a contact tuple	
g	gravity connstant	m/s^2
$h_i \quad i = 1...3$	absolute offset in t_1 - t_3	m
k	spring stiffness of joint	Nm/rad
L	number of the limbs in contact	
m	number of task DOF	
m	total mass	kg
m_l	mass of one robot leg	kg
m_t	mass of robot trunk	kg
N	number of actuated robot joints	
n	number of stances or VRP waypoints	
N_c	number of joints in configutation space	
n_{φ}	number of trajectory segements	
n_{steps}	number of desired steps	
$q_{1,lb}$	lower limit of hip joint (front legs)	rad
$q_{1,ub}$	upper limit of hip joint (front legs)	rad
$q_{3,lb}$	lower limit of knee joint (front legs)	rad
$q_{3,ub}$	upper limit of knee joint (front legs)	rad
s_{len}	step length	m
T	index for trunk frame with the coordinates t_1, t_2, t_3	

T_φ	duration time of a trajectory segment	
T_{DS}	duration time of double support	s
T_{SS}	duration time of single support	s
$f_\varphi^{(k)}(t)$	k-th time differentiation of the interpolation function	
φ	index of a trajectory segment	
${}^P_h \boldsymbol{\nu}_{i,k}$	hybrid velocity of a link j relative to another link i is represented in the coordinate frame that is attached to link i in the frame P	
${}^P_h \mathbf{J}_{i,j}$	hybrid Jacobian of a link j relative to another link i	
${}^P \mathbf{H}_J$	homogeneous transformation matrix from the frame J to the frame P	
${}^P \mathbf{R}_J$	rotation matrix from the frame J to the frame P	
f_φ	polynomial spline interpolation function	
p	degree of the spline interpolation function f_φ	

List of Abbreviations

<i>CoM</i>	center of mass
<i>CoP</i>	center of pressure
<i>DOF</i>	degrees of freedom
<i>DCM</i>	Divergent Component of Motion (point)
<i>eCMP</i>	Enhanced Centroidal Moment Pivot point
<i>LF</i>	left front leg
<i>LH</i>	left hind leg
<i>LIP</i>	Linear Inverted Pendulum
<i>RF</i>	right front leg
<i>RH</i>	right hind leg
<i>SEA</i>	Series Elastic Actuator
<i>SLIP</i>	Spring-Loaded Inverted Pendulum
<i>URDF</i>	Universal Robotic Description Format
<i>VRP</i>	Virtual Repellent Point
<i>ZMP</i>	Zero Moment Point

1 Introduction

1.1 Motivation

The main challenges of locomotion of legged robots are the hybrid and non-linear dynamics of the system and the constraints of the needed contact forces of the robot with its environment. An additional challenging subproblem of the generation of dynamic locomotion is the design of a real-time control architecture which controls the desired robot body motion and limb movement in uneven terrain or under unknown forces. Particularly for dynamic locomotion, the robot has to be continuously in motion during walking or running in order to avoid falling or tilting. For a suitable interaction with the world, the motion planning for dynamic locomotion has to provide dynamically and kinematically feasible motion trajectories for the robot's body and its limbs.

Therefore, the main topic of this thesis is to investigate the generation of continuous closed-form trajectories for a quadrupedal robot with *Series Elastic Actuators* (SEA) in order to perform dynamic locomotion. The selected method contains the concept of three-dimensional *Divergent Component of Motion* point (DCM) and the concept of *Virtual Repellent Point* (VRP). These concepts can be applied in order to generate a smoothed *center of mass* (CoM) and robot limbs' trajectories. Moreover, this method has already been successfully applied to bipedal robots. With this concept it is possible to describe gait pattern as multi-contact scenarios by an alternating sequence of contact stances on the ground. Therefore, this intuitive description of gait pattern with a couple of design parameters should be investigated and extended for compliant four-legged locomotion. The developed extension will be tested for dynamic locomotion on the compliant quadrupedal robot Bert. A successful implementation of the DCM and the VRP concept could result in a reduction of the complexity of the quadrupedal locomotion model, and could provide a compact and efficient method for the generation of feasible trajectories for compliant quadrupedal robots on demanding ground. This has to be validated by experiments on the robot Bert after the implementation.

1.2 State of the Art

This section provides a compact overview of current state of the art on trajectory generation methods for compliant quadrupedal robots. The research indicates the advantage of compliant elements in the limb mechanism for the generation of dynamic locomotion pattern (Alexander, 1990; Geyer, Seyfarth, & Blickhan, 2006; Holmes, Full, Koditschek, & Guckenheimer, 2006). These elements have the capability to store and to release energy temporally during the stances Holmes et al. (2006). Furthermore, this enables the quadrupedal robots to perform faster gaits, like trotting, pacing and dynamic walking compared to static walking gaits (Lakatos et al., 2018; Spröwitz et al., 2013). The application of passive compliant actuators in legged locomotion on robots have shown a higher compatibility with different

grounds and a higher robustness against shocks due to unexpected forces (Hutter et al., 2012; J. E. Pratt, Krupp, & Morse, 2002).

In the last two decades several research groups developed such quadrupedal robot systems and trajectory planners for dynamic locomotion (Bledt et al., 2018; Hutter, Gehring, Hopfinger, Bloesch, & Siegwart, 2014; Hutter et al., 2016; Lakatos et al., 2018; Raibert, Blankespoor, Nelson, & Playter, 2008; Spröwitz et al., 2013). In the following section the focus will be on approaches of the trajectory generation of these robots. Bellicoso, Jenelten, Gehring, and Hutter (2018) generate CoM trajectories as 5th order polynomial curves on the robot system ANYmal with series elastic actuators. The splines are calculated by solving a sequence of non-linear optimization problems to minimize the acceleration of the CoM motion subject to non-linear inequality constraints for the *Zero-Moment Point* (ZMP) (Bellicoso et al., 2018). Furthermore, in this approach the foot placement is realized and formulated as a quadratic programming problem based on an inverted pendulum model for legs (Gehring et al., 2013, 2016; J. E. Pratt & Tedrake, 2006). For the foot placement the next valid capture point is calculated. This point is further used as a stabilizing term in the above problem with a regularization term for the default feet placement for a foot of the gait pattern (Bellicoso et al., 2018). The desired reference trajectories are tracked by a whole-body controller (Bellicoso et al., 2017, 2018).

For a robot with compliant legs as in the approach by Spröwitz et al. (2013), the authors apply a Central Pattern Generator (a neuronal network or finite state machine which generates joint trajectories for a desired gait based on input signals) for trotting gait for a robot with compliant legs. They achieve to produce joint trajectories for an open loop control of the robot, which performs robust walking on slightly leveled terrain (Spröwitz et al., 2013). The passively compliant actuated quadrupedal robot Bert performs dynamic gaits from trotting till dynamic walking with a compliant leg mechanism design. This design enables both, to feature the structure of the dynamics of the *Spring-Loaded Inverted Pendulum* (SLIP) model (Blickhan, 1989) and allows a robust control of the robot by a Finite State Machine and a parameterization of the desired gaits (Lakatos et al., 2018).

Di Carlo, Wensing, Katz, Bledt, and Kim (2018) present a dynamic quadrupedal robot controlled by a convex model-predictive control with an approximation reduced robot dynamics. The system performs dynamic gaits from trotting with flight phases till gallop (Di Carlo et al., 2018). The reference trajectory for the body is created to fulfill a desired body orientation and velocity in space, which is commanded by user input (Di Carlo et al., 2018). Furthermore, a contact sequence for the desired gait is sent by operator input (Di Carlo et al., 2018).

The authors Bledt et al. (2018) present an approach for the CoM reference position calculation for the same robot based on an anticipatory weighting-strategy to predict a virtual support polygon (Bledt et al., 2018).

The application of bipedal motion planning concepts for quadrupedal robots can be used to reduce the motion planning problem. This can be realized through a formulation identical to bipedal walking for dynamic gaits like trotting, bounding and pacing, where leg pairs of the robot act like one virtual leg (Liu et al., 2019; Raibert, Chepponis, & Brown, 1986). Therefore, an obvious consideration is to extend or implement already successfully tested bipedal motion planning methods to quadrupedals based on the *Linear Inverted Pendulum* (LIP) or the SLIP model for these gaits.

A method for bipedal walking planning is the DCM concept and the VRP concept (s. chap-

ter 2.5). This concept provides an analytical solution for continuous 3D trajectories of the robot's CoM for a desired contact pattern of the feet (Englsberger, Mesesan, & Ott, 2017; Englsberger, Ott, & Albu-Schaffer, 2015; Mesesan, Englsberger, Garofalo, Ott, & Albu-Schaffer, 2019; Mesesan, Englsberger, Henze, & Ott, 2017; Mesesan, Englsberger, Ott, & Albu-Schaffer, 2018; J. E. Pratt et al., 2012; Takenaka, Matsumoto, & Yoshiike, 2009). The DCM method decomposes the CoM dynamics of the floating base robot, which follow Newton's second law, into two linear first-order dynamics. One of these two dynamics is the unstable dynamic for the DCM point in space. The other one is the stable CoM dynamic. The CoM will converge to the DCM point and the VRP will push the DCM away (forward). In this way, only the unstable dynamics have to be controlled whereas the stable dynamic can be neglected (Englsberger et al., 2015; Mesesan et al., 2019). The trajectory generation in this concept is based on defining a series of contact points on the ground (the foot placement) and is followed by a mapping onto VRP waypoints, which are applied for a piecewise interpolation (1st, 3rd or 5th order polynomial) (Mesesan et al., 2018). The DCM trajectory and CoM trajectory waypoints can be computed by an efficient matrix multiplication, resulting from the solution of the linear dynamics. Then the trajectories can be obtained by an interpolation between the waypoints (Mesesan et al., 2018). As a result, the DCM and VRP concept handles the hybrid dynamics of legged robot motion in a direct and intuitive way. Consequently the search for a feasible CoM trajectory can be replaced by a manual or automatic method selection of suitable VRP waypoints (Mesesan et al., 2018). Additionally, a DCM-controller could be applied to a whole-body-controller of the robot (Mesesan et al., 2017). Based on the previous method, walking algorithms for bipedal walking and dynamic multi-contact motion were presented with robust results on leveled or compliant grounds for a torque-controlled bipedal robot (Englsberger et al., 2015; Mesesan et al., 2019, 2018).

The results of Lakatos et al. (2018) and Spröwitz et al. (2013) show the benefits of robots designed with intrinsic passive compliance in order to enable dynamic quadrupedal locomotion. The authors present robust control approaches without a trajectory generation for the CoM (Lakatos et al., 2018; Spröwitz et al., 2013).

However, the generation of feasible continuous CoM reference trajectories provide advanced capabilities to control dynamic quadrupedal locomotion, for example to ensure stability in demanding terrain or to increase the versatility. Therefore, an investigation of a directly feasible trajectory generation method (Mesesan et al., 2019, 2018) on compliant quadrupedal robots could provide a suitable motion planning approach for dynamic quadrupedal locomotion. In contrast to other methods, as in Bellicoso et al. (2018), a DCM-based motion planning would provide directly smoothed CoM and limb trajectories. To achieve this, heuristic rules based on automatic VRP-selection could be applied.

2 Fundamentals

This chapter covers the used notation in subchapter 2.1 and fundamentals of this thesis. This includes the basics of legged robot locomotion (section 2.2) and common models of bipedal (section 2.3) and quadrupedal locomotion (section 2.4). Further this chapter introduces the DCM method in subchapter 2.5 and gives an overview of the robot Bert (section 2.6).

2.1 Notation

In this thesis the following notation is applied to vector variables. Bold symbols represent vector variables or position vectors in a frame, where \mathbf{s} in (2.1)

$${}^P\mathbf{s}_{P,k} \quad (2.1)$$

is a vector in the frame $\{P\}$ from the frame origin of $\{P\}$ to the point k . If not noted otherwise, a vector without upper left index is described in the world frame $\{W\}$. And if not otherwise noted, a vector with no first relative right bottom index, have a origin in the noted frame. Further, bold capital symbols indicate matrices. The homogeneous transformation and rotation matrices are notated according to Engelsberger (2016) in (2.2).

$$\begin{bmatrix} {}^P\mathbf{s}_{P,k} \\ 1 \end{bmatrix} = {}^P\mathbf{H}_J \begin{bmatrix} {}^J\mathbf{s}_{J,k} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^P\mathbf{R}_J & {}^J\mathbf{s}_J - {}^J\mathbf{s}_P \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^J\mathbf{s}_{J,k} \\ 1 \end{bmatrix} \quad (2.2)$$

The equation (2.2) describes the transformation of a point k in the frame J to the frame P , where ${}^P\mathbf{H}_J \in \mathbb{R}^{4 \times 4}$ is the homogeneous transformation matrix and ${}^P\mathbf{R}_J \in \mathbb{R}^{3 \times 3}$ is a rotation matrix, which is orthogonal and has a determinant of 1 (Engelsberger, 2016). According to Engelsberger (2016) the inverse of ${}^P\mathbf{H}_J$ can be determined to

$${}^P\mathbf{H}_J^{-1} = \begin{bmatrix} {}^P\mathbf{R}_J^T & -{}^P\mathbf{R}_J^T({}^J\mathbf{s}_J - {}^J\mathbf{s}_P) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (2.3)$$

In this work the classification by Murray, Li, and Sastry (1994) and the definition by Engelsberger (2016) is applied to describe velocity vectors.

For this thesis the *hybrid velocity* is now explained (s. Engelsberger (2016) for further information). The hybrid velocity ${}^P_h\boldsymbol{\nu}_{i,k} \in \mathbb{R}^6$, as a combination of "linear and angular velocity of a link j relative to another link i is represented in the coordinate frame that is attached to link i " (frame $\{P\}$) (Engelsberger, 2016), has the correlation to the configuration space of the robot $\dot{\mathbf{q}}_c \in \mathbb{R}^{N_c}$ with N_c joints,

$${}^P_h\boldsymbol{\nu}_{i,k} = {}^P_h\mathbf{J}_{i,j}\dot{\mathbf{q}}_c \quad (2.4)$$

where ${}^P_h \mathbf{J}_{i,j} \in \mathbb{R}^{6 \times N_c}$ is the hybrid Jacobian matrix (Engelsberger, 2016; Murray et al., 1994). Thereby a bottom left index is used to indicate the type of velocity. The notation $\mathbf{I}_{3 \times 3}$ indicates e.g. an identity matrix $\in \mathbb{R}^{3 \times 3}$ and $\mathbf{0}_{3 \times 3}$ is a zero matrix $\in \mathbb{R}^{3 \times 3}$.

2.2 Basics of Legged Robot Locomotion

Legged robots are under-actuated free-floating base systems. *Free-floating* describes the fact that the robot's base is not permanently fixed in the environment. Instead it is floating over the ground. Therefore, the six *degrees of freedom* (DOF) of the base (three cartesian coordinates and three rotational coordinates) have to be considered in the dynamic description of such a system. The term *under-actuated* means that the number of available DOF of the lower limb (legs) joints, and therefore the available control inputs, are not sufficient to control all DOF of the system directly, especially the DOF of the base (Ratliff, 2014; Tedrake, 2020). In case of a free-floating base robot the DOF of the system, the DOF for the N actuated joints $\mathbf{q}_a \in \mathbb{R}^N$ and the six DOF of the base result in equation (2.5),

$$\mathbf{y} = \begin{bmatrix} \mathbf{r}_b \\ \boldsymbol{\psi}_b \\ \mathbf{q}_a \end{bmatrix} \in \mathbb{R}^{6+N} \quad (2.5)$$

where $\mathbf{r}_b \in \mathbb{R}^3$ are the three cartesian translation DOFs and $\boldsymbol{\psi}_b \in \mathbb{R}^3$ are the three rotational DOFs (roll, pitch and yaw). As a result of these characteristics, legged robots have to generate the necessary forces and torques on the free-floating base by an interaction with their environment in order to move the base in space as desired. The interaction of the limbs' end effector generates reaction forces from the contact to the robot (Ratliff, 2014; Wieber, Tedrake, & Kuindersma, 2016).

As mentioned in the introduction, one of the main challenges of the legged robot locomotion is to place the lower limbs' end-effectors in contact with the environment and to control the actuated joints in a suitable way to avoid tilting or falling. This can be achieved by applying contact forces on the robot trunk by the limbs. The dynamic equations for a multi-body legged robot can be described according to Wieber et al. (2016) by the *Lagrangian Dynamics* in equation (2.6),

$$\mathbf{M}(\mathbf{y}) \left[\begin{pmatrix} \ddot{\mathbf{r}}_b \\ \ddot{\boldsymbol{\psi}}_b \\ \ddot{\mathbf{q}}_a \end{pmatrix} + \begin{pmatrix} -\mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{N \times 1} \end{pmatrix} \right] + \mathbf{C}(\mathbf{y}, \dot{\mathbf{y}}) = \begin{pmatrix} \mathbf{0}_{6 \times 1} \\ \boldsymbol{\tau} \end{pmatrix} + \sum_{i=1}^L \mathbf{J}_{C,i}^T \cdot \mathbf{f}_{C,i} \quad (2.6)$$

where $\mathbf{M} \in \mathbb{R}^{(N+6) \times (N+6)}$ is the inertia matrix of the robot, $\mathbf{C} \in \mathbb{R}^{N+6}$ are the coriolis and centrifugal effects, $\boldsymbol{\tau} \in \mathbb{R}^N$ are the joint torques and $\mathbf{J}_{C,i} \in \mathbb{R}^{(N+6) \times 3}$ is the i -th contact Jacobian matrix of the i -th limb in contact, $\mathbf{f}_{C,i} \in \mathbb{R}^3$ is the contact force vector, where L is the number of limbs in contact and $\mathbf{g} = [0 \ 0 \ -g] \in \mathbb{R}^3$ is the gravity acceleration vector with gravity constant g . The contact Jacobians describe the relation between the contact forces and the DOF of the system.

Switching between stance (limbs have contact) and swing phases (limbs have no contact) of the limbs implies for the dynamics in (2.6) a hybrid nature due to the changing contact events.

2.3 Models of Bipedal Locomotion

A common approach to describe the dynamic of a bipedal (or multiple legged) robot is to apply template models for the robot and the walking. These template models reduce the complexity of the real system by using the CoM for the dynamic description. (Kajita & Ott, 2016; Wieber et al., 2016)

Following this, the dynamics of the CoM can be described according to Newton's second law,

$$m\ddot{\mathbf{x}} = m\mathbf{g} + \mathbf{F}_{ext} \quad (2.7)$$

where m is the total mass of the robot, $\ddot{\mathbf{x}} \in \mathbb{R}^3$ is the acceleration of the CoM and $\mathbf{F}_{ext} \in \mathbb{R}^3$ the sum of all forces acting on the CoM without the gravity force. The legged system can be described as a LIP model, if the legs of a bipedal system are assumed to be mass-less and linear variable, and if the CoM is fixed on top of the legs and concentrated in a single point (Kajita, 2019). Therefore, \mathbf{F}_{ext} can be considered as the sum of the contact forces in equation (2.6). Thereby is a further assumption a zero change of angular momentum around \mathbf{x} , during the swing phases (single support; only one leg in ground contact)(Kajita & Ott, 2016).

For example, in this model the limbs are alternately fixed on the ground (foot in stance) and \mathbf{F}_{ext} is acting along the intersection line between the CoM and the *Zero Moment Point* (Vukobratović & Stepanenko, 1972) (ZMP) (Sugihara, Nakamura, & Inoue, 2002). The ZMP represents a point on the ground at which the horizontal moments of the CoM are zero (Vukobratović, M. and Borovac, B., 2004) and can be considered as a torque free base joint (Sugihara et al., 2002) (Englsberger, Ott, & Albu-Schaffer, 2013). If the ZMP is within the contact surface of the stance foot (base of support), this contact is stable (Vukobratović, M. and Borovac, B., 2004). According to the explanation by Engelsberger (2016), the sum of the external forces on the CoM have to be on a line of action that intersects with the base of support or has to be designed to fulfill this criterion. Consequently the *center of pressure* (CoP) has to be in the base of support. The CoP is the point on a contact surface where the resultant moment of the contact forces is either zero (Sardain & Bessonnet, 2004) or only has a component parallel to the surface normal vector (Goswami, 1999). For planar (flat) contact surfaces (all contact points in one single plane) the CoP is identically to the ZMP (Hardarson, 2002; Sardain & Bessonnet, 2004).

A restriction of the CoM movement to a horizontal plane gives the decoupled dynamic CoM equations of the LIP in horizontal direction based on the equation (2.8) by Sugihara et al. (2002),

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \frac{g}{x_3 - r_{ZMP,3}} \cdot \begin{bmatrix} x_1 - r_{ZMP,1} \\ x_2 - r_{ZMP,2} \end{bmatrix} \quad (2.8)$$

where and $\mathbf{r}_{ZMP} \in \mathbb{R}^3$ is the position of the ZMP. The intersection of (2.8) in (2.7) gives the desired limb (leg) force acting on CoM for a CoM and ZMP position.

$$\mathbf{F}_{ext} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = m \frac{g}{x_3 - r_{ZMP,3}} \cdot \begin{bmatrix} x_1 - r_{ZMP,1} \\ x_2 - r_{ZMP,2} \\ 0 \end{bmatrix} \quad (2.9)$$

An overview of motion planning based on 3D LIP can be found in Kajita (2019). An obvious restriction of (2.8) is that the CoM can only move on a constant height, due to the dynamics decoupling based on limited CoM movement (Englsberger et al., 2015; Kajita & Ott, 2016). To overcome this problem for example Zhao and Sentis (2012) designed a further development of the LIP model. It is based on a decoupling of the 3D dynamics in sagittal and lateral motion by a predefined piece-wise linear surface for the CoM movement. The authors planned the lateral foot placement for given sagittal feet positions and desired sagittal velocities by a numerical search.

A further extension of the linear inverted pendulum template model is called the SLIP, which introduces springs in the mass-less legs (Blickhan, 1989; Geyer et al., 2006). Compared to the LIP, this model allows the modelling of flight phases and therefore running modes additionally to walking modes with double support phases (Geyer et al., 2006).

2.4 Models for Quadrupedal Locomotion

The models of bipedal walking and running (inverted pendulum models) can be applied to the template model-based motion planning of quadrupedal locomotion. Raibert et al. (1986) report such an approach by combining pairs of legs as one single virtual leg and applying a 3D one-legged algorithm to the virtual leg (Kajita & Ott, 2016; Raibert, 1990; Raibert et al., 2008, 1986). Based on this approach gaits like trotting, pacing and bounding are realizable on quadrupedal robots (Raibert et al., 1986). Furthermore, the ZMP is also applied in the quadrupedal locomotion models. Identical to the definition of bipedal robots by Vukobratović, M. and Borovac, B. (2004), the ZMP is also a stability margin for more-legged dynamical walking and can be used for the motion planning and control, e.g. in Bellicoso et al. (2018). Where this stability margin is the minimum distance between the ZMP and the edges of the *support polygon* (Kajita & Ott, 2016). By that, the *support polygon* is constructed by the convex hull of the contact points projections of feet on the *support plane*, where the support plane is a plane perpendicular to gravity vector (McGhee & Frank, 1968). Figure 2.1 shows the construction of the support polygon for a quadrupedal robot. In general, if the projection of the CoM along the gravity vector to the ground is within the support polygon, the robot will not tilt and is statically stable (McGhee & Frank, 1968).

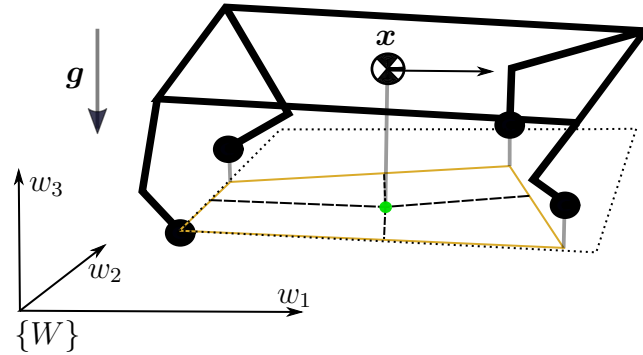


Figure 2.1: Schematic sketch of the support polygon construction for quadrupedal via the projection of feet points along the gravity vector (gray arrow) on a support plane. The support plane (black dotted lines) is orthogonal to the gravity vector and align with the lowest feet in the world frame $\{W\}$. The orange colored lines depict the edges of the convex hull of the feet points and the green point represents the projection of the CoM on the support plane. The black dashed lines in the support plane show the minimum distances of the CoM projection (green dot) to the edges of the support polygon.

2.5 Divergent Component of Motion Method

The Divergent Component of Motion (DCM) Method is based on the approach that divides the second order differential equations of the CoM in a stable and an unstable linear first order dynamics (Englsberger et al., 2013, 2015; Takenaka et al., 2009).

For this the 3D DCM point is defined by Englsberger et al. (2013) in (2.10),

$$\boldsymbol{\xi} = \mathbf{x} + b\dot{\mathbf{x}} \quad (2.10)$$

where $b = \sqrt{\frac{\Delta z_{vrp}}{g}}$ is the time constant of these dynamics and $\Delta z_{vrp} > 0$ is a desired height of the CoM (Englsberger et al., 2013). Therefore, Δz_{vrp} can be considered as a kinematic design parameter (Englsberger et al., 2015) for the walking gait. The DCM point, as a linear combination of the CoM position and the velocity, is a point in front of the CoM in its current movement direction. The reordering of (2.10) gives directly for $b > 0$ the stable linear first order dynamics of the CoM in (2.11)

$$\dot{\mathbf{x}} = -\frac{1}{b}(\mathbf{x} - \boldsymbol{\xi}) \quad (2.11)$$

Equation (2.11) shows that the CoM will follow to the DCM point. The separation in a stable and unstable part of CoM dynamics starts by the differentiation of (2.10) and inserting (2.7) and (2.11) in this result. This results in the linear first order dynamics of the DCM coupled with the CoM coupled as in (2.12).

$$\dot{\boldsymbol{\xi}} = -\frac{1}{b}\mathbf{x} + \frac{1}{b}\boldsymbol{\xi} + \frac{b}{m}(m\mathbf{g} + \mathbf{F}_{ext}) = -\frac{1}{b}\mathbf{x} + \frac{1}{b}\boldsymbol{\xi} + \frac{b}{m}\mathbf{F}_{CoM} \quad (2.12)$$

The force $\mathbf{F}_{CoM} \in \mathbb{R}^3$ is the total force acting on the CoM. Figure 2.2 shows the definition of the DCM point and related forces on the CoM of a legged robot. Thereafter, Englsberger et al. (2013) introduce a linear encoding of the \mathbf{F}_{ext} based on the difference of the CoM position and the *Enhanced Centroidal Moment Pivot point* (eCMP) (s. figure 2.2),

$$\mathbf{F}_{ext} = \frac{m}{b^2}(\mathbf{x} - \mathbf{r}_{eCMP}) \quad (2.13)$$

where $\mathbf{r}_{eCMP} \in \mathbb{R}^3$ is the position of eCMP. Applying (2.13) to (2.12) results in the from CoM decoupled dynamics of the DCM.

$$\dot{\boldsymbol{\xi}} = \frac{1}{b}\boldsymbol{\xi} - \frac{1}{b}\mathbf{r}_{eCMP} + b\mathbf{g} \quad (2.14)$$

For a further simplification of (2.14) for the planning, Englsberger et al. (2013) introduced the concept of the *Virtual Repellent Point*. After Englsberger et al. (2015) the VRP encodes the total force (\mathbf{F}_{CoM}) acting on the CoM (Englsberger et al., 2013). With this further force encoding variable, the VRP in (2.15),

$$\mathbf{v} = \mathbf{r}_{eCMP} + [0 \ 0 \ b^2 \cdot g]^T = \mathbf{r}_{eCMP} + [0 \ 0 \ \Delta z_{vrp}]^T = \mathbf{x} - b^2 \cdot \ddot{\mathbf{x}} \quad (2.15)$$

where $\mathbf{v} \in \mathbb{R}^3$ is the position of the VRP, Engelsberger et al. (2013) shorten (2.14) to

$$\dot{\xi} = \frac{1}{b}(\xi - \mathbf{v}) \quad (2.16)$$

(Engelsberger et al., 2013; Mesesan et al., 2018)

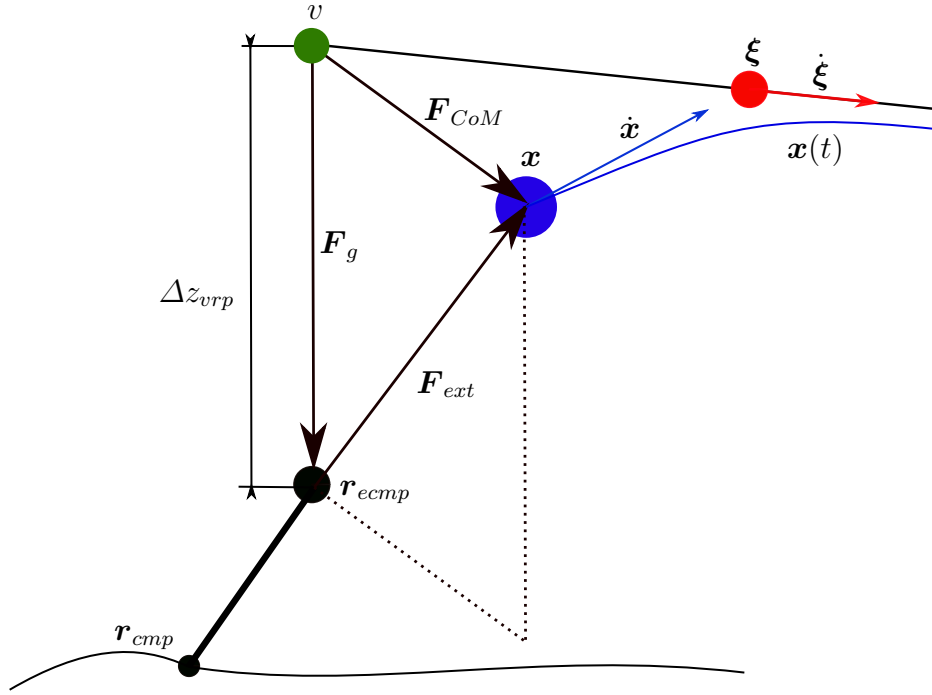


Figure 2.2: Sketch of the definition of DCM and VRP. The VRP is placed with an offset of Δz_{vrp} over the eCMP. The total force on the CoM ($\mathbf{F}_{ext} + m \cdot \mathbf{g}$) is encoded by the VRP. The blue items correspond to the CoM, the green to the VRP and the red to the DCM. Engelsberger et al. (2015)

This equation is the unstable linear first order dynamics of the DCM, which is pushed forward by the VRP, the DCM “diverges” (Engelsberger et al., 2017). The decoupled DCM dynamics in (2.16) allows to design a control law which only has to stabilize the unstable DCM dynamics and without the need to control the naturally stable CoM dynamics (Engelsberger et al., 2015) in real-time (Engelsberger, 2016). Equation (2.15) shows that the first two components of the VRP and the eCMP are identical and vertical components of both differ by Δz_{vrp} (s. figure 2.2) (Engelsberger, 2016). Both, the VRP and eCMP could be interpreted as a generalization of the ZMP and CMP concept to the three-dimensional space (Engelsberger, 2016).

With the assumptions for the planning process reported by (Engelsberger et al., 2015) the \mathbf{r}_{eCMP} is assumed to be identically to the desired foot point position and the CoP on the ground and can be replaced by the foot point position \mathbf{r}_f in (2.15). This allows a selection of a sequence of VRP waypoints in which the first two coordinates are identically to the desired

feet placement (not limited on one level height) in order to describe the walk sequence (Englsberger et al., 2015). A design aspect for the VRP waypoints depending on the desired feet placement and contact properties of the surface, is to select positions which can act as a CoM rest position (Mesesan et al., 2017, 2018). Figure 2.3 shows the general planning and calculation process of the DCM method. A general problem formulation to describe the multi-contact of a robot can be found in Mesesan et al. (2017). Following this formulation, a set of L contacts of the robot limbs with the environment can be described as a *stance* (Bouyarmane & Kheddar, 2012).

$$\boldsymbol{\sigma} = \{c_l \mid l = 1..L\} \quad (2.17)$$

Where a contact

$$c_l = (\mathbf{p}_c, \mathbf{R}_c) \quad (2.18)$$

is defined by a contact point $\mathbf{p}_c \in \mathbb{R}^3$ and the orientation of the contact frame $\mathbf{R}_c \in SO(3)$, where the third coordinate is identical to the surface normal at the contact Mesesan et al. (2017). Mesesan et al. (2017) define a VRP way point for each stance which can be chosen to describe a walking sequence by a sequence of stances. Solving equations (2.11) and (2.16) with a selected polynomial spline interpolation function f_φ from Englsberger et al. (2017) for the VRP in (2.19) gives the solutions for the DCM and CoM trajectory according to Englsberger et al. (2017) in (2.20) and (2.21). Thereby, a selected VRP waypoint sequence $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_n]^T \in \mathbb{R}^{n \times 3}$ is applied with n waypoints.

Thereby the desired references for VRP, DCM and CoM are interpolated segmentally between the start waypoints $(\mathbf{v}_{\varphi,0}, \boldsymbol{\xi}_{\varphi,0}, \mathbf{x}_{\varphi,0})$ and the end waypoints $(\mathbf{v}_{\varphi,T_\varphi}, \boldsymbol{\xi}_{\varphi,T_\varphi}, \mathbf{x}_{\varphi,T_\varphi})$ of a segment. A segment φ of n_φ segments is defined as fixed-VRP or a transition-VRP phase. During a fixed phase, the VRP waypoint is not changing and during a transition phase the VRP is interpolated to the next waypoint. The duration of a segment is determined by T_φ . To ensure consistent trajectories the end points of the current segment correspond to the start points of the next segment. The following equations (2.19) till (2.21) can be found in Mesesan et al. (2018).

$$\mathbf{v}_\varphi(t) = (1 - f_\varphi(t))\mathbf{v}_{\varphi,0} + f_\varphi(t)\mathbf{v}_{\varphi,T_\varphi} \quad (2.19)$$

$$\boldsymbol{\xi}(t) = \alpha_{\varphi,\xi}(t)\mathbf{v}_{\varphi,0} + \beta_{\varphi,\xi}(t)\mathbf{v}_{\varphi,T_\varphi} + \gamma_{\varphi,\xi}(t)\boldsymbol{\xi}_{\varphi,T_\varphi} \quad (2.20)$$

$$\mathbf{x}(t) = \alpha_{\varphi,x}(t)\mathbf{v}_{\varphi,0} + \beta_{\varphi,x}(t)\mathbf{v}_{\varphi,T_\varphi} + \gamma_{\varphi,x}(t)\boldsymbol{\xi}_{\varphi,T_\varphi} + \delta_{\varphi,x}(t)\mathbf{x}_{\varphi,0} \quad (2.21)$$

where

$$\alpha_{\varphi,\xi}(t) = 1 - \sigma_\varphi(t) - e^{\frac{t-T_\varphi}{b}}(1 - \sigma_\varphi(T_\varphi)) \quad (2.22)$$

$$\beta_{\varphi,\xi}(t) = \sigma_\varphi(t) - e^{\frac{t-T_\varphi}{b}} \cdot \sigma_\varphi(T_\varphi) \quad (2.23)$$

$$\gamma_{\varphi,\xi}(t) = e^{\frac{t-T_\varphi}{b}} \quad (2.24)$$

$$\sigma_\varphi(t) = \sum_{k=0}^p \left(b^k f_\varphi^{(k)}(t) \right) \quad (2.25)$$

$$\alpha_{\varphi,x}(t) = \left(1 - \rho_{\varphi}(t) - \frac{1 - \rho_{\varphi}(0)}{e^{\frac{t}{b}}} - \frac{e^{\frac{t}{b}} - e^{-\frac{t}{b}}}{2e^{\frac{T_{\varphi}}{b}}} (1 - \sigma_{\varphi}(T_{\varphi})) \right) \quad (2.26)$$

$$\beta_{\varphi,x}(t) = \rho_{\varphi}(t) - \frac{\rho_{\varphi}(0)}{e^{\frac{t}{b}}} - \frac{e^{\frac{t}{b}} - e^{-\frac{t}{b}}}{2e^{\frac{T_{\varphi}}{b}}} \sigma_{\varphi}(T_{\varphi}) \quad (2.27)$$

$$\gamma_{\varphi,x}(t) = \frac{e^{\frac{t}{b}} - e^{-\frac{t}{b}}}{2e^{\frac{T_{\varphi}}{b}}} \quad (2.28)$$

$$\delta_{\varphi,x}(t) = e^{-\frac{t}{b}} \quad (2.29)$$

$$\rho_{\varphi}(t) = \sum_{k=0}^{\lfloor p/2 \rfloor} \left(b^{2k} f_{\varphi}^{(2k)}(t) \right) \quad (2.30)$$

In this notation $f_{\varphi}^{(k)}(t)$ means the k -th time derivation of the interpolation function. This function has to be selected according to the reported properties in Engelsberger et al. (2017). Furthermore, with the design choices for an ensuring continuity for standing-to-walking transition, as reported by Engelsberger et al. (2017), and the terminal constraint $\boldsymbol{\xi}_{n,T_{\varphi}} = \mathbf{v}_n$ (i.e. the robot will stop at this waypoint) the sequence of VRP waypoint $\mathbf{v}_{wp} \in \mathbb{R}^3$ can be computed in equation (2.31)

$$\mathbf{v}_{wp} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} \\ & \mathbf{A}_{\xi,0}(1:3,:) & & \\ \mathbf{0}_{n-3 \times 3} & \mathbf{0}_{n-3 \times 3} & \mathbf{I}_{n-3 \times 3} & \mathbf{0}_{n-3 \times 3} \\ \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_1 \\ \mathbf{v}_{free} \\ \mathbf{v}_n \end{bmatrix} \quad (2.31)$$

where

$$\mathbf{A}_{\xi,0} = (\mathbf{I}_{3n \times 3n} - \mathbf{A}_{\gamma} \cdot \mathbf{A}_{back})^{-1} \cdot (\mathbf{A}_{\alpha\beta} + \mathbf{A}_{\gamma} \cdot \mathbf{A}_{tc}) \quad (2.32)$$

$$\mathbf{A}_{\gamma} = \begin{bmatrix} \gamma_{1,\xi}(0) \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} & \gamma_{n,\xi}(0) \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (2.33)$$

$$\mathbf{A}_{back} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0}_{3 \times 3} \\ \vdots & & & \ddots & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \cdots & \cdots & \cdots & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (2.34)$$

$$\mathbf{A}_{\alpha\beta} = \begin{bmatrix} \alpha_{1,\xi}(0)\mathbf{I}_{3\times 3} & \beta_{,\xi}(0)\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \alpha_{n_\varphi,\xi}(0)\mathbf{I}_{3\times 3} & \beta_{n_\varphi,\xi}(0)\mathbf{I}_{3\times 3} \end{bmatrix} \quad (2.35)$$

$$\mathbf{A}_{tc} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \cdots & \cdots & \cdots & \mathbf{0}_{3\times 3} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \end{bmatrix} \quad (2.36)$$

and $\mathbf{A}_{\xi,0}(1:3,:)$ denotes the first three rows of the matrix $\mathbf{A}_{\xi,0}$. Further, the vector $\mathbf{v}_{free} \in \mathbb{R}^{3 \cdot (n-3)}$ are the waypoints which can be freely selected for the walk planning. The terminal constraint for DCM results in a stop of the DCM at the end of the planned trajectory (Englsberger et al., 2015). Mesesan et al. (2018) present a compact computation way to determine the needed waypoints for the DCM reference $\Xi = [\xi_1 \dots \xi_n]^T \in \mathbb{R}^{3n \times 3}$ and the waypoints for the CoM reference $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n]^T \in \mathbb{R}^{3n \times 3}$ in equations (2.20) and (2.21). First, equation (2.20) is evaluated for $t = 0$ and equation (2.21) is evaluated for $t = T_\varphi$. Second, the equations are formulated in matrix form for all segments by using the matrix $\mathbf{S}_0 = [\mathbf{I}_{n-1 \times n-1}, \mathbf{0}_{n-1 \times 1}]$ to select the start points and $\mathbf{S}_T = [\mathbf{0}_{n-1 \times 1}, \mathbf{I}_{n-1 \times n-1}]$ to select the end points of the segments. Third, the linear systems are solved for the DCM and CoM waypoint matrices shown in equations (2.37) and (2.38)

$$\Xi = [\Xi C_V \quad \Xi c_\xi] \begin{bmatrix} \mathbf{V} \\ \xi_{n,T_\varphi}^T \end{bmatrix} \quad (2.37)$$

$$\mathbf{X} = [{}^X C_V \quad {}^X c_\xi \quad {}^X c_x] \begin{bmatrix} \mathbf{V} \\ \xi_{n,T_\varphi}^T \\ \mathbf{x}_{1,0}^T \end{bmatrix} \quad (2.38)$$

where

$$\Xi C_V = (\mathbf{I}_{3 \times 3} - \mathbf{S}_o^T \Gamma_\xi \mathbf{S}_T)^{-1} \mathbf{S}_0^T (\mathbf{A}_\xi \mathbf{S}_0 + \mathbf{B}_\xi \mathbf{S}_T) \quad (2.39)$$

$$\Xi c_\xi = (\mathbf{I}_{3 \times 3} - \mathbf{S}_0^T \Gamma_\xi \mathbf{S}_T)^{-1} \begin{bmatrix} \mathbf{0}_{n-1 \times 1} \\ 1 \end{bmatrix} \quad (2.40)$$

$${}^X C_V = (\mathbf{I}_{3 \times 3} - \mathbf{S}_T^T \Delta_x \mathbf{S}_0)^{-1} \cdot \mathbf{S}_T^T (\mathbf{A}_x \mathbf{S}_0 + \mathbf{B}_x \mathbf{S}_T + \Gamma_x \mathbf{S}_T^T \Xi C_V) \quad (2.41)$$

$${}^X c_\xi = (\mathbf{I}_{3 \times 3} - \mathbf{S}_T^T \Delta_x \mathbf{S}_0)^{-1} \cdot \mathbf{S}_T^T (\Gamma_x \mathbf{S}_T^T \Xi c_\xi) \quad (2.42)$$

$${}^X\mathbf{c}_x = (\mathbf{I}_{3 \times 3} - \mathbf{S}_T^T \mathbf{\Delta}_x \mathbf{S}_0)^{-1} \cdot \begin{bmatrix} 1 \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} \quad (2.43)$$

and the matrices \mathbf{A}_ξ , \mathbf{B}_ξ , $\mathbf{\Gamma}_\xi$, \mathbf{A}_x , \mathbf{B}_x , $\mathbf{\Gamma}_x$ and $\mathbf{\Delta}_x \in \mathbb{R}^{n_\varphi \times n_\varphi}$ in the above mentioned equations are square and diagonal. These matrices include for the DCM waypoints the values of functions (2.22) to (2.24) evaluated at $t = 0$ and for the CoM waypoints the function values (2.26) to (2.29) evaluated at $t = T_\varphi$. The initial constraint for the start CoM waypoint is selected to $\mathbf{x}_{1,0} = \mathbf{v}_1$. (Englsberger et al., 2015) Mesesan et al. (2018)

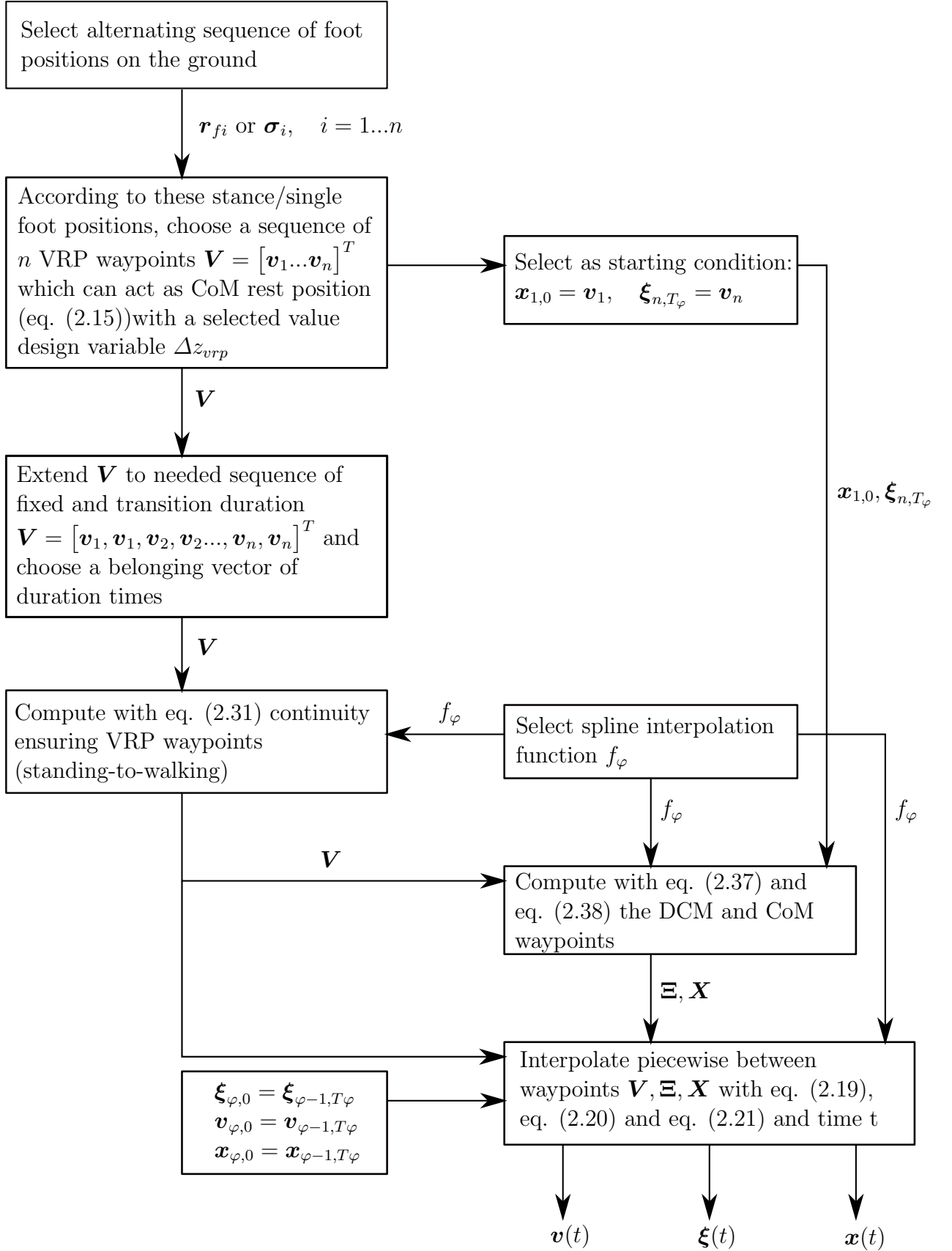


Figure 2.3: Calculation flow of the DCM method. The boxes are calculation steps and the arrows indicate the calculation flow and the needed signals or variables.

2.6 The compliant quadrupedal Bert

The legged robot system Bert is a quadrupedal with *serial elastic actuators* (G. A. Pratt & Williamson, 1995), which is designed to feature the structure of the dynamics of the SLIP (Lakatos et al., 2018). Figure 2.5 shows a image of the applied robot system Bert. Figure 2.4 and table 2.1 show the kinematic and mechanical properties of Bert. The robot has eight actuated joint DOF, two DOF for each planar leg, and it can perform dynamical gaits like pronk, trot and dynamic walk (Lakatos et al., 2018).

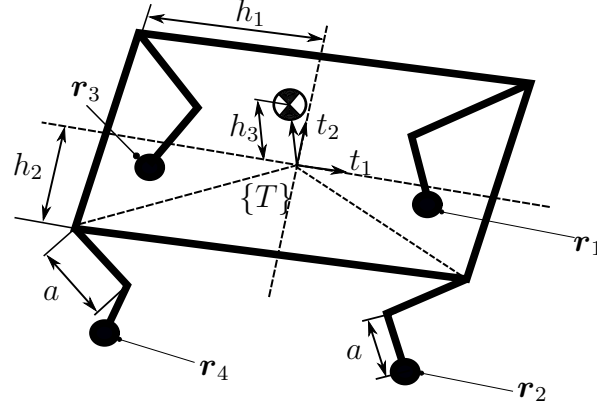


Figure 2.4: Schematic representation of kinematic parameters of the robot Bert. The frame $\{T\}$ is the trunk frame, where t_1 is the positive forward direction. The hips align in the same plane of the trunk ($t_1 t_2$) and the 2-DOF planar front legs are placed symmetrically to the hind legs. The positive direction of the rotation vector of the joints points is in the positive direction of t_2 . The legs are encoded with RF: right front, LF: left front, RH: right hind and LH: left hind, where $\mathbf{r}_1 \in \mathbb{R}^3$ is the LF-, $\mathbf{r}_2 \in \mathbb{R}^3$ is the RF-, $\mathbf{r}_3 \in \mathbb{R}^3$ is the LH- and $\mathbf{r}_4 \in \mathbb{R}^3$ is the RH-foot point position.

The mechanical properties (leg position, robot design and inertias) are designed to be symmetric in order to match the eigenmode for a vertical bouncing motion of the robot (Lakatos et al., 2018). The hind legs are placed symmetric to the front legs. Furthermore, the robot legs are encompassing two equal length leg segments and two servo drives, their positions are controlled by a PD-law via pulse width modulation (Lakatos et al., 2018). The system is working at a rate of 1 kHz and trunk orientation is measured by inertial measurement unit (Lakatos et al., 2018). The trunk position is measured by an optical tracking system and the leg mass $m_l \approx 0.1$ kg is much smaller than the trunk mass $m_t \approx 2.4$ kg (Lakatos et al., 2018). The motor positions are $\boldsymbol{\theta} \in \mathbb{R}^8$ and ordered according to [LF RF LH RH].

Furthermore, the final joint vector $\mathbf{q} \in \mathbb{R}^8$ contains two DOF for each leg in the same order as $\boldsymbol{\theta}$. Figure A.1 in the appendix shows a sketch of an articulated robot leg and the placed joints. The CoM of the robot can only be placed in the sagittal plane, caused by the planar leg design (Lakatos et al., 2019; Seidel, Hermann, Gumpert, Loeffl, & Albu-Schaffer, 2020). Therefore the base features two translational and two rotational DOF (Lakatos et al., 2019, 2018; Seidel et al., 2020). With this restriction, the earlier described kinematic properties as well as the assumption that angular velocity of the motor rotors depends only on own spinning (de Luca & Book, 2016), the general dynamic equations (2.6) for the quadrupedal result in (2.44),

Table 2.1: Kinematic and dynamic parameters of the robot Bert

Symbol	Name	Value
h_1	absolute offset in t_1	0.165 m
h_2	absolute offset in t_2	0.103 m
h_3	absolute offset in t_3	0.032 m
a	leg segment length	0.08 m
m	total mass	2.875 kg
k	spring stiffness of joint	2.7 Nm/rad
$q_{1,lb}$	lower limit of hip joint (front legs)	-75 deg
$q_{1,ub}$	upper limit of hip joint (front legs)	105 deg
$q_{3,lb}$	lower limit of knee joint (front legs)	-120 deg
$q_{3,ub}$	upper limit of knee joint (front legs)	120 deg

$$\begin{bmatrix} M(\mathbf{y}) & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{y}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} 0 \\ g \\ \mathbf{0}_{10 \times 1} \\ \mathbf{0}_{8 \times 1} \end{bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{y}, \dot{\mathbf{y}}) \\ k(\boldsymbol{\theta} - \mathbf{q}) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{4 \times 1} \\ k(\boldsymbol{\theta} - \mathbf{q}) \\ \boldsymbol{\tau}_{motor} \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^L \mathbf{J}_{C,i}^T \cdot \mathbf{f}_{C,i} \\ \mathbf{0}_{8 \times 1} \end{bmatrix} \quad (2.44)$$

where $B \in \mathbb{R}^{8 \times 8}$ is the constant and diagonal motor inertia and $\boldsymbol{\tau}_{motor} \in \mathbb{R}^8$ are the motor torques.

**Figure 2.5:** The applied robot Bert

3 From Bipedal to Quadrupedal Locomotion

This chapter describes the application of the DCM method to quadrupedal locomotion. It is divided in four subsections. The first section 3.1 contains the problem formulation of quadrupedal locomotion, followed by the section 3.2 which covers the gait definition and a possible placement strategy for the VRP waypoints used for the motion planning. The third section 3.3 covers the model-based trajectory generation planning and the last section 3.4 treats the topic of joint reference generation.

3.1 Problem Formulation

For the dynamical quadrupedal locomotion, or in general every dynamical legged locomotion, the robot system has to solve generally two tasks.

First, to place the robot's feet in a suitable sequence on the ground which supports a *continuous forward* locomotion of the robot. Therefore, the robot has to hold its feet in these suitable positions during the stance phases of the limbs, with regard to ground unevenness or other interactions with the environment (e.g. additional forces) in order to avoid *tilting or falling*. Second, the robot has to control the desired walk sequence in *real-time*, in order to move to the desired position in space.

With the basics of legged locomotion mentioned in chapter 2 in mind, we can identify two necessary main components to solve these tasks:

- Generation of dynamically and kinematically feasible CoM and limb trajectories.
- Control of limb movements to ensure that the desired trajectories are maintained.

For the trajectory generation, the desired limb and CoM movements have to be designed to give the possibility to fulfil the constraint of the force \mathbf{F}_{ext} . Therefore, the line of action of \mathbf{F}_{ext} has to pass through the base of support. A possible generation approach, the DCM method, was summarized in chapters 1.2 and 2.5. This method is employed to generate the CoM references for quadrupedal locomotion in this thesis.

First, the dynamical locomotion of quadrupedals has to be considered for the application of the DCM method. Second, a suitable gait definition has to be formulated. Third, a VRP placement has to be designed. In the following, we make these simplifying assumptions in order to provide a problem description which can be used to solve the above mentioned tasks:

Assumption 3.1.1 (foot point) *All contacts of the robot's stance feet $\mathbf{r}_{f,i}$ are foot points with one single point of contact on the ground (or environment).*

This is a good approximation of small robot feet. Furthermore, the robot can only apply contact forces whose projections along the contact normal vector are positive, i.e. only push on the ground, and not apply a contact moment. Figure 3.2 shows the assumptions and definitions made for a quadrupedal robot. Furthermore, it is assumed:

Assumption 3.1.2 (point mass with mass-less legs) *The mass of the robot legs is neglected and the robot can be considered as point-mass, which is concentrated in the CoM of the trunk.*

The multi-body system of the quadrupedal can be described by a LIP model. In this model, the force \mathbf{F}_{ext} on the CoM is the resulting force of the external forces. The intersection point of the line of action for this force with the base of support is the joint base of the LIP model, i.e CoP or ZMP.

Quadrupedal robots (with a minimum number of 12 DOF for a six DOF floating base) can perform a wide range of gaits and transitions between these (s. chapter 1.2). In general, one can distinguish between *static* and *dynamical* quadrupedal locomotion, which are shown in figure 3.1. In the static locomotion the robot is always balanced, i.e with the definition in chapter 2.4 the CoM projection is within the support polygon at each time point. In the static gait *walk*, e.g. the robot shifts only one leg per step.

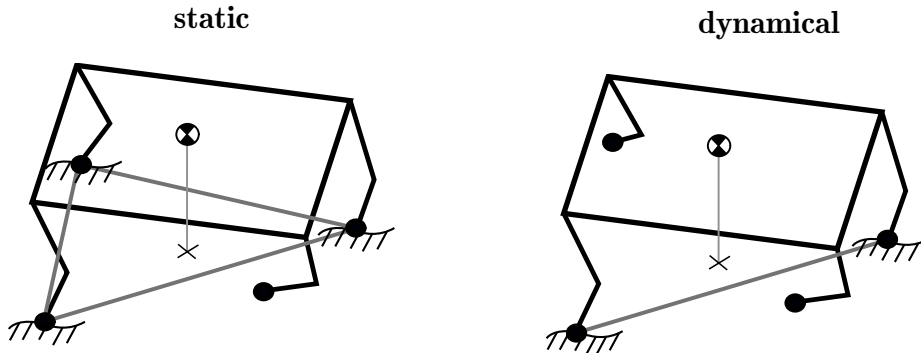


Figure 3.1: Schematic definition of static and dynamical locomotion for quadrupedal in forward movement. The left part of the figure shows the static state, which is true for gaits like *walk*. The right part represents dynamical locomotion, which is hold for gaits like *trotting* or *pacing*. The gray lines between the foot points with ground contact determine the convex hull of the contact feet (base of support) and the black cross indicates the vertical projection of the CoM to a support plane (s. 2.4).

However, in the dynamical locomotion more than one leg are in the swing phases and therefore the support polygon is reduced to a connection line between the contact feet. This holds for gaits like trotting, pacing, dynamical walk or bounding (Lakatos et al., 2018). In this case the intersection point of the resulting force with the ground is to be placed and to be held on this line or point. The robot has to be in continuous forward motion, in order to avoid falling. Alternatively, the robot has to move in a CoM position (configuration) which allows the robot to stop safely, i.e. in a static configuration (static equilibrium). These static

equilibrium configurations are considered for the later motion planning. The motivation for this selection is based on the consideration of possible robot stances for a rest position. This is similar to the definition of *multi-contact-locomotion* for bipedal robots according to Mesesan et al. (2017, 2018). Therefore, following Mesesan et al. (2018) (see section 2.5), a possible rest position will be selected to plan a desired VRP \mathbf{v}_i . In the CoM rest position the external force is

$$\mathbf{F}_{ext,i} = -m\mathbf{g} \quad (3.1)$$

as $\ddot{\mathbf{x}} = \mathbf{0}$. This represents the desired external force. Together with the gravity effect it has to be encoded by the VRP ($\mathbf{F}_{CoM} = \frac{m}{b^2}(\mathbf{x} - \mathbf{v})$).

The CoM rest position of the robot is defined in the following:

Assumption 3.1.3 (Projection of CoM on horizontal plane) *If a configuration of the robot exists without slipping in the contact points for a static equilibrium, $m\ddot{\mathbf{x}} = \mathbf{0}$, with no angular momentum around \mathbf{x} and \mathbf{x} is over the base of support, the projection of the CoM on the support plane will be within the support polygon in (2.4).*

Assumption 3.1.3 holds true in case of a flat or slightly uneven ground, i.e in the best case all contact points are co-planar and all feet apply a contact force which is in the specific friction cone and is (or have components) parallel to the gravity vector (Bretl & Lall, 2008).

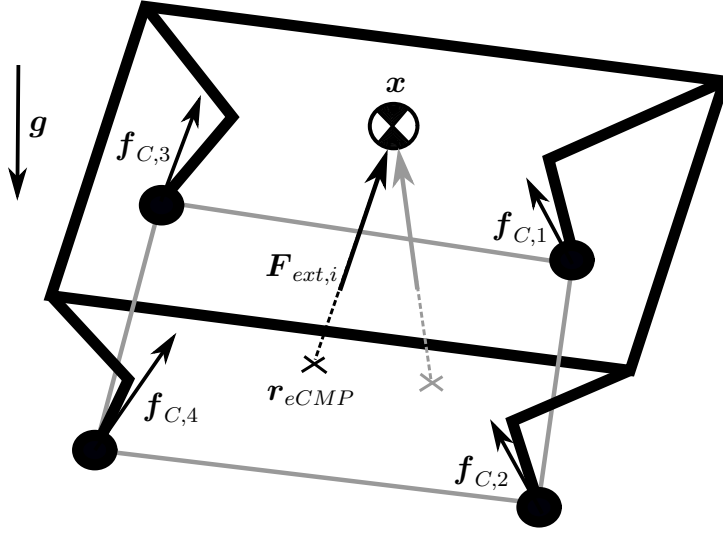


Figure 3.2: Definitions on a quadrupedal for DCM. The forces $\mathbf{f}_{C,i}$ are the contact forces. These forces cause the desired $\mathbf{F}_{ext,i}$ for the i -th planned eCMP \mathbf{r}_{eCMP} . For the i -th stance $\mathbf{F}_{ext,i}$ has to pass through the desired eCMP. The gray vector represent the next desired eCMP after the transition.

In this case the convex hull of the contact points in space builds the base of support and the foot points' projection on a horizontal plane builds a sufficient approximation of this area. Furthermore, the CoP is contained in the convex hull of the contact points (Wieber et

al., 2016) and can be considered as the average of their contact points position, where the positions are weighted by the magnitude of their contact forces (Hardarson, 2002). Therefore, according to Wieber et al. (2016), a necessary condition for the static equilibrium on flat ground is that the projection of the CoM along \mathbf{g} lies within the boundaries of the planar convex hull of the contact points in space (also for support planes with a surface normal vector which is not parallel to \mathbf{g}).

One has to note that the CoM projection is not equal to the eCMP, which is assumed to be identical to the CoP for VRP planning by Engelsberger et al. (2015), in particular, the planned eCMP has to be within the base of support Engelsberger (2016). But in a possible rest position the CoP is identical to the CoM projection on the real plane of the convex hull of contact points (Hardarson, 2002). In this thesis the assumption 3.1.3 is considered for the foot point projection (s. chapter 2.4). Following the above mentioned points, the following assumption 3.1.4 is considered:

Assumption 3.1.4 (Planar ground) *The ground is entirely flat or slightly uneven that all contact points are nearly in one contact plane.*

The transition from one VRP waypoint to the next is placed between two swing phases for the motion planning. Hence, the quadrupedal has at least three legs in contact with the ground, i.e the robot is potentially in static equilibrium or can stop. Consequently, the eCMP (CoP) could be designed to be in the convex hull of contact points and the robot would be alternating between these stable rest positions. Of course, the robot can tilt in the dynamical phases of a gait (just two feet in contact) and therefore the eCMP has to be placed within in the support polygon, i.e on the line between these two contact points.

In summary, points within the support polygon or on the edge of a support polygon are applied for the planned VRP sequence for walking on a slightly uneven terrain. The base of support is determined by projection of the foot points to a support plane perpendicular to the gravity vector. The VRP waypoints are constructed with these points as possible eCMP and by equation (2.15).

The definitions and assumptions introduced here form the basis for the following chapters.

3.2 Gait Definitions

This section is dedicated to an approach that describes gait planning (also called step planning) for the DCM method, based on the quadrupedal gaits: walk, trotting, pacing and dynamical walk (Lakatos et al., 2019, 2018; Seidel et al., 2020). Firstly, the basic gait for static locomotion is described, namely the static *walk*. Depending on the kinematics of the quadrupedal robot, the robot lifts one leg per step in order to achieve a forward movement of the CoM over a sequence of steps. Figure 3.3 shows the fall pattern of an example walk, where two single steps are shown without the return in a final stance. A final stance is defined as the stance in which the front leg parallel and the hind legs are parallel. After this, based on definitions for the walk are the dynamic gaits are described.

In the walk only one leg is swinging and a step can be determined by the phases of time in which the leg swings and the time in of full contact on the ground in which the robot shift in the next rest position (s.chapter 3.1).

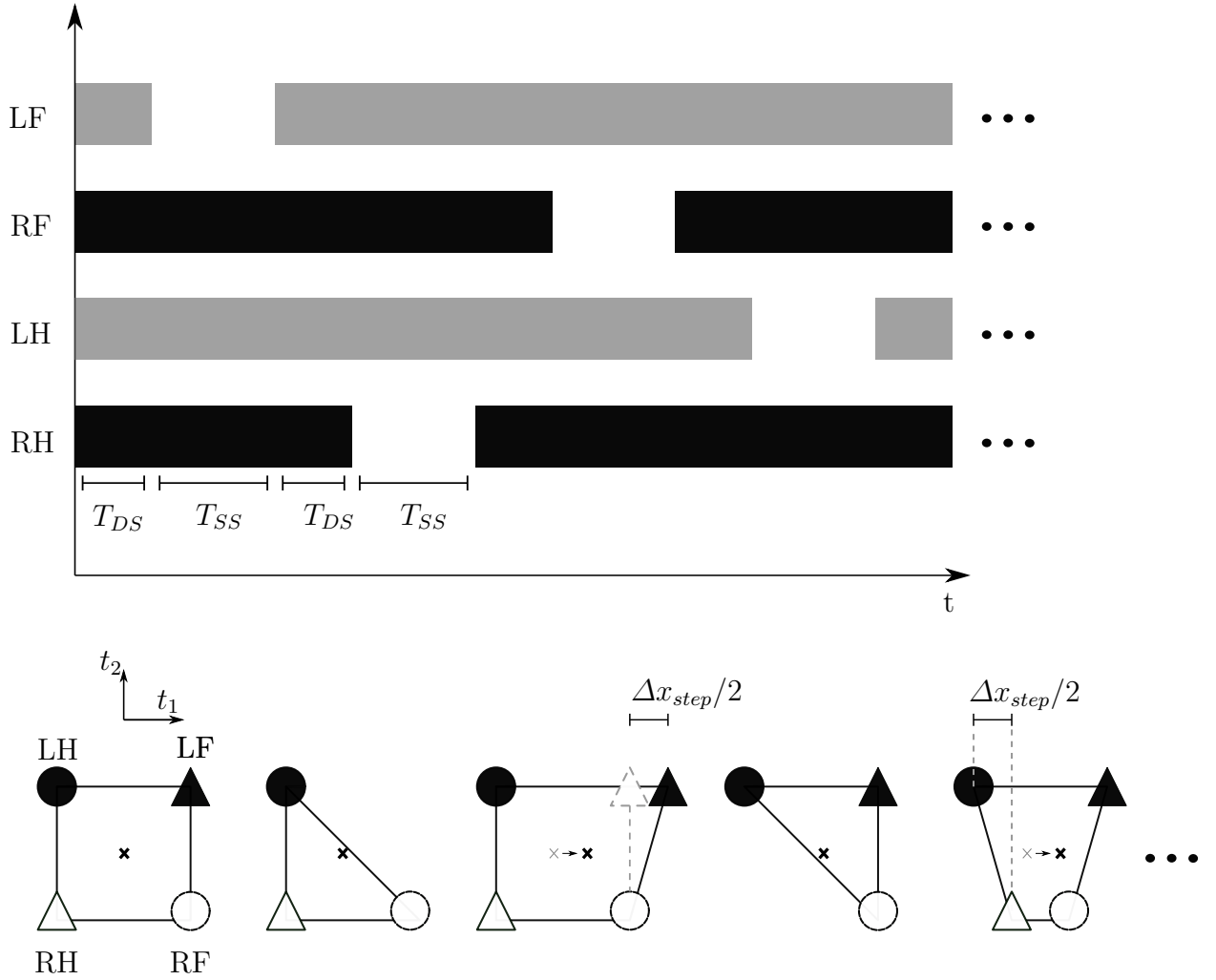


Figure 3.3: Gait definition scheme of the static walk. The top figure is the fall pattern of the walk, where the gray and black bars indicate the contact time with the ground. T_{DS} is the double support time duration and T_{SS} is single support time duration. The bottom figure shows a stance sequence of two single steps of the walk. The unit vectors t_1 and t_2 indicate the horizontal trunk plane and t_1 points in the robot forward direction. The black cross in the bottom figure shows a possible eCMP position in the support polygon and the transition during the double support phases. Δx_{step} is value used for the foot point shifting in t_1 .

These times are now defined similar to definition for the bipedal walking (compare Englsberger et al. (2015)) as:

Definition 3.2.1 (double support) *The time duration in which transition from a CoP reference point or VRP waypoint to the next is placed is the **double support** and have the duration time T_{DS} . This time duration corresponds to the transition phase of the VRP trajectory in chapter 2.5.*

Definition 3.2.2 (single support) *The time duration in which a CoP reference point or VRP waypoint is constant (fixed) is the **single support** and have the duration time T_{SS} . During*

this duration time the limbs are swinging from one contact point \mathbf{p}_{c,l_i} to the next $\mathbf{p}_{c,l_{i+1}}$, $l = 1...4$ and $i = 1...n$ (s. chapter 2.5). This time duration correspond to fixed phases of the VRP trajectory in chapter 2.5.

The duration time of a step (step time) is the sum of the duration time of double support and single support phase. In the determined static walk the foot points are shifted with a fourth of the desired step length. For this the forward step variable is introduced:

Definition 3.2.3 (step length) *The step length s_{len} is the displacement of a contact point to its next contact point in trunk forward direction. Thereby is the variable Δx_{step} the half step length and is the displacement between the both front foot points or hind foot points in forward direction (s. figure 3.4).*

A further needed variable is the desired step height which is defined as:

Definition 3.2.4 (step height) *For the foot point movement during the single support, the variable Δh_{step} is the desired step height in the vertical direction (w_3) at the half of the duration time.*

Figure 3.3 shows in the bottom a stance sequence for two single steps of the proposed static walk with a possible selection of the eCMP (CoP) positions in the support polygon. For the static walk the half of Δx_{step} is used as displacement between the foot points respectively. Based on the sequence of stances in Figure 3.3 the sequence of single support, start and end stances is defined in equation (3.2).

$$\Sigma = \{\sigma_1 \quad \dots \quad \sigma_n\} \quad (3.2)$$

This sequence contains the stances according to equations (2.17) and (2.18), and is used in the further motion planning for the VRP placement. If a contact for a limb is not in the current stance $c_l \notin \sigma_i \wedge c_l \in \sigma_{i-1}$, then the limb is detached from the environment. If a limb contact appears new in $c_l \in \sigma_i \wedge c_l \notin \sigma_{i-1}$, then it is attached to environment, and the contact is held in contact, if $c_l \in \sigma_i \wedge c_l \in \sigma_{i-1}$, (Mesesan et al., 2017). Otherwise the contact is detached. A quadrupedal with four legs, and thus four limbs, can maximum have four limb contacts with the environment (s. definition 3.2.2). This applied stance description allows directly to determine between which stances and contact points in space a limb has later to be interpolated.

The dynamic gait trotting and pacing can be described according to Raibert et al. (1986), where diagonal pairs (trotting) or lateral pairs (pacing) of the legs are considered as a virtual single leg. These pairs are swinging synchronously and showed in the fall pattern in figures 3.4 and 3.5. The above present description based on stance also allows to determine these *dynamic walking gaits*, i.e. without any *flight phases* (= no single limb contact). In the gaits discussed so far, the fall pattern is designed with duration times and the specific touch down sequence for the gait. These duration times can be selected arbitrarily in order to modify the gait properties for changing requirements, e.g. for new terrain requirements, gait transition

or motion modification, and the specific kinematic requirements of the legged robot.

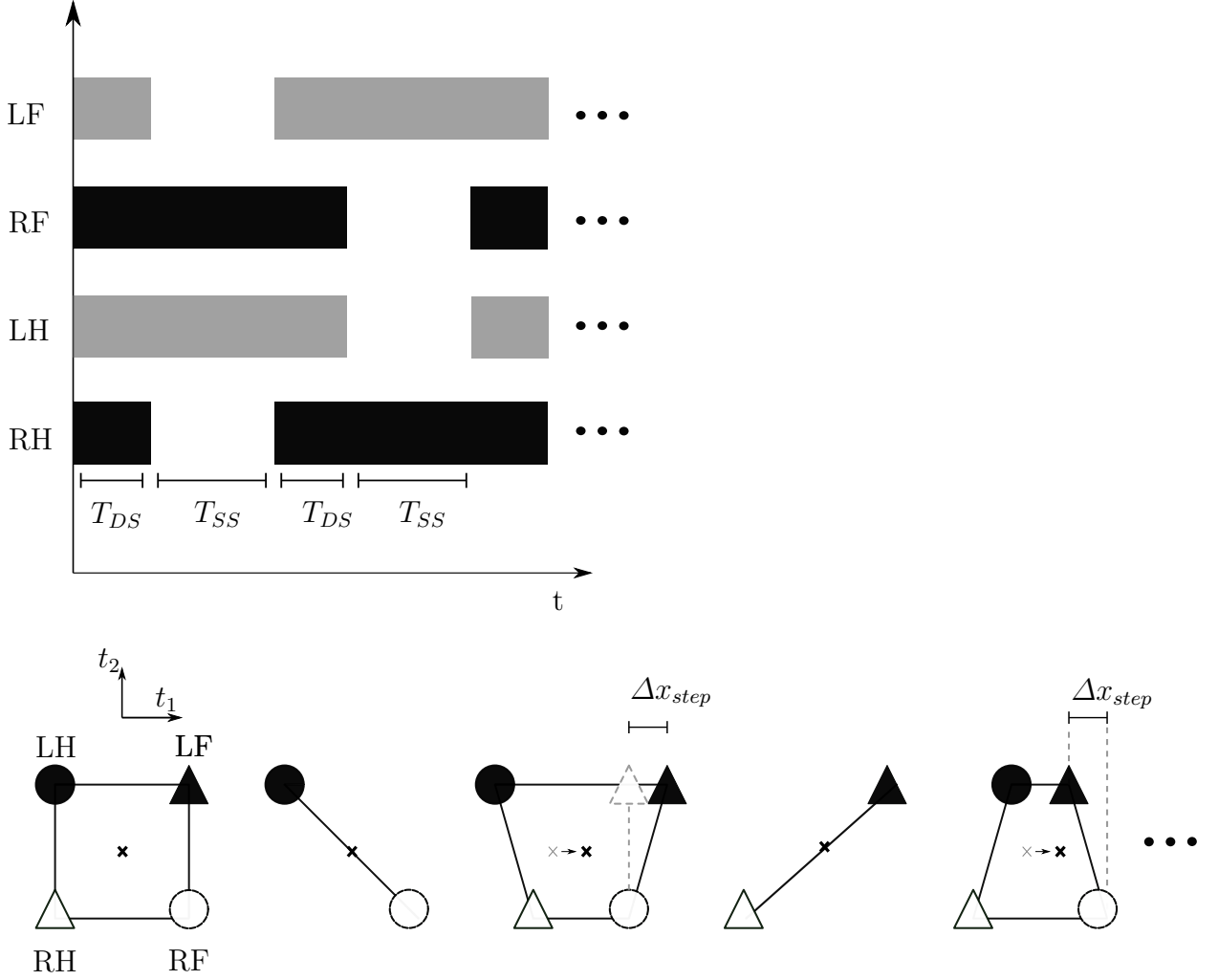


Figure 3.4: Gait definition scheme of the trotting walk. The top figure is the fall pattern of trotting, where the gray and black bars indicate the contact time with the ground. T_{DS} is the double support time duration and T_{SS} is single support time duration. The bottom figure shows a stance sequence of two single steps. The unit vectors t_1 and t_2 indicates the horizontal trunk plane and t_1 points in the robot forward direction. The black cross in the bottom figure shows a possible eCMP position in the support polygon and the transition during the double support phases. Δx_{step} is value used for the foot point shifting in t_1 .

The dynamical walk can be defined by a touch down sequence: left hind, left front, right hind, right front, similar to the above static walk, whereby the next foot point is lifted before the next foot touches down (Lakatos et al., 2018). Therefore, the dynamical walk has phases with only two feet in contact and is not statically stable at all times (Lakatos et al., 2018) compared to the walk in figure 3.3.

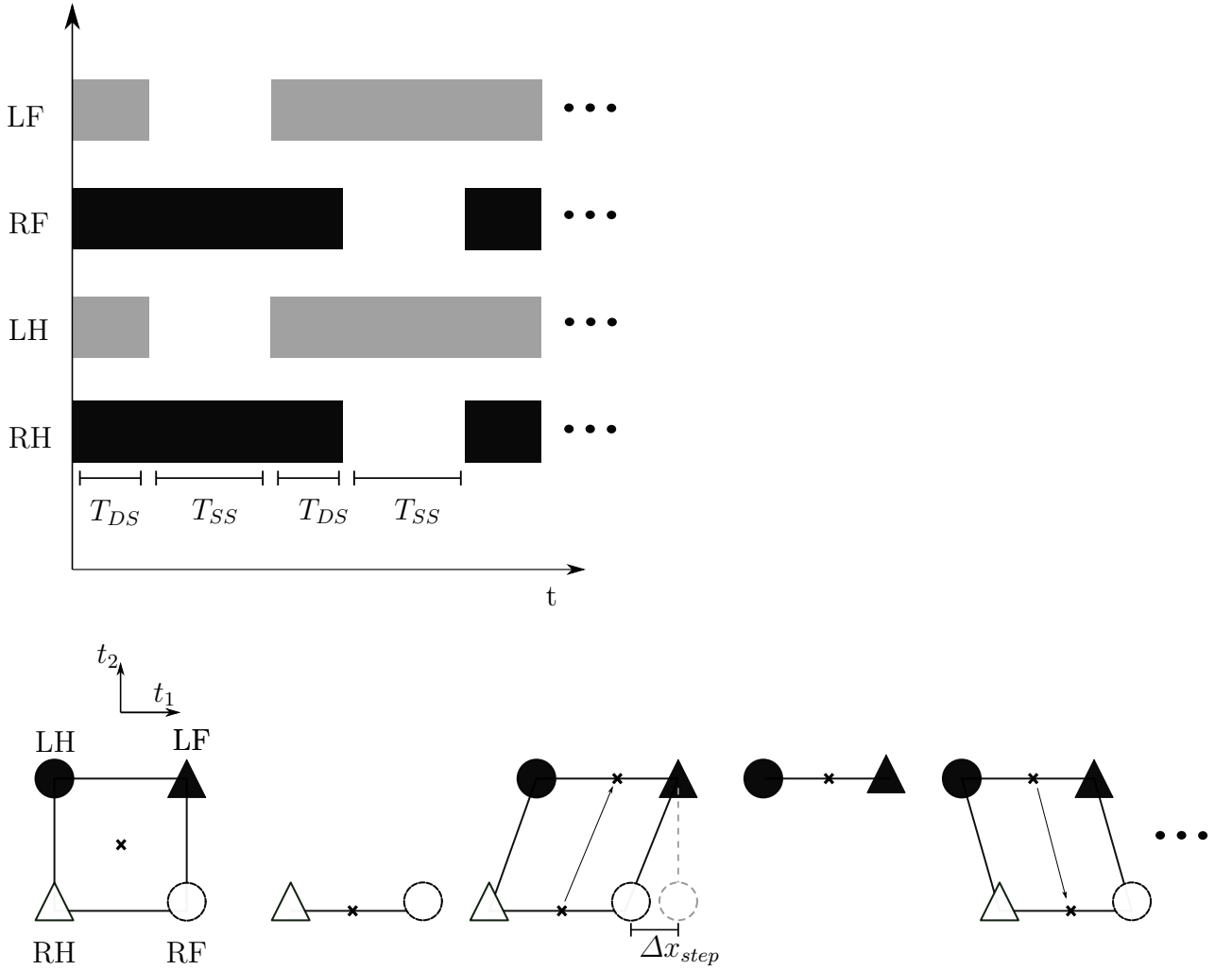


Figure 3.5: Gait definition scheme of the pacing walk. The top figure is the fall pattern of pacing gait, where the gray and black bars indicate the contact time with the ground. T_{DS} is the double support time duration and T_{SS} is single support time duration. The bottom figure shows a stance sequence of two single steps. The unit vectors t_1 and t_2 indicates the horizontal trunk plane and t_1 point in the robot forward direction. The black cross in the bottom figure shows a possible eCMP position in the support polygon and the transition during the double support phases. Δx_{step} is value used for the foot point shifting in t_1 .

The definitions in 3.1 requires that a VRP waypoint has to be chosen, which have a eCMP (*remark (2.15)*: $\mathbf{r}_{eCMP} = \mathbf{v} - [0 \ 0 \ \Delta z_{vrp}]^T$), which is in the support polygon. With the above stance sequence definition (eq. (3.2)), a VRP waypoint is defined for each stance by the geometric center $\mathbf{g}_\sigma \in \mathbb{R}^2$ of its support polygon in (3.3)

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{g}_{\sigma,i} \\ 0 \end{bmatrix} + [0 \ 0 \ \Delta z_{vrp}]^T \quad (3.3)$$

3.3 Trajectory Generation

Based on the stance definition and the determination of the VRP, the DCM method from section 2.5 can be applied to generate a continuous and smoothed CoM reference for a desired robot motion determined by steps in 3D space. For this purpose, the algorithm of figure 2.3 is used. The spline interpolation function f_φ in Engelsberger et al. (2017) has to be selected, according to the desired interpolation degree p . Figure 3.6 shows the resulting trajectories for a robot in trotting gait, $p = 1$ and the example value in table 3.1. The limbs (s. figure 3.7) are interpolated during the single support phases between the contact points with a cubic function (s. Engelsberger et al. (2017)).

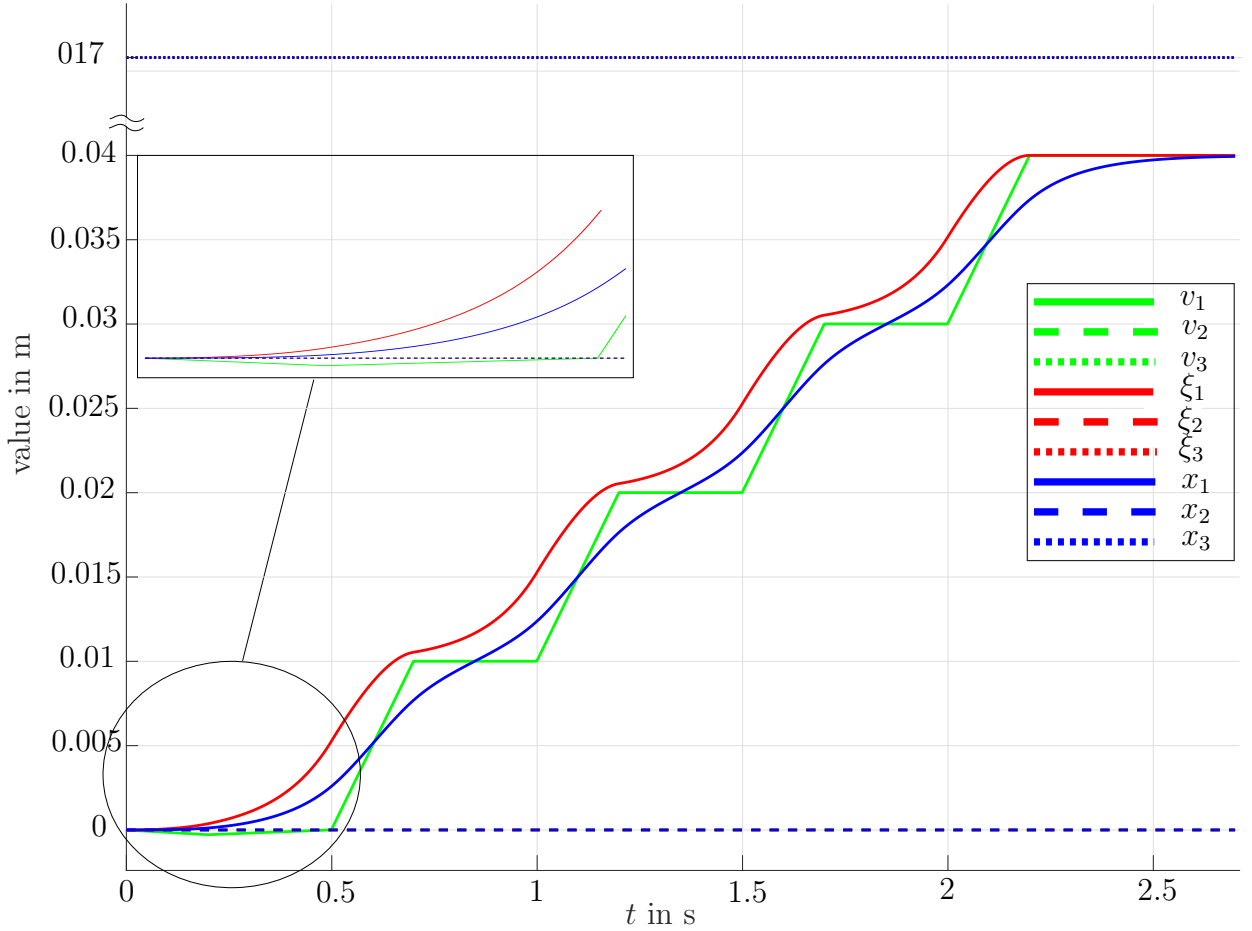


Figure 3.6: Interpolation example of the trajectory method for $p = 1$. The reference created for the values in table 3.1 and a forward trotting robot. The green lines indicate the VRP, the red lines the DCM and the blue lines the CoM, the solid lines indicate the interpolation in the forward direction (w_1), the dashed lines the horizontal direction (w_2) and the dotted lines the vertical direction (w_3). The detail shows the standing-to-walking condition realized by (2.31). The lines in the second and third direction are constant due to the limitation of the applied robot (s. chapter 2.6)

Thereby the interpolation in the third direction (vertical direction) is split in two segments with an additional waypoint in the desired step height Δh_{step} at the half time of the duration (s. definition 3.2).

Table 3.1: Parameters for example trajectory generation

Variable	Value
Δz_{vrp}	0.17 m
T_{DS}	0.2 s
T_{SS}	0.3 s
gait type	trotting

Figure 3.7 shows a reference generation example of the interpolation degree of 5 for f_φ in combination with the desired limb interpolation (3rd degree). The quantities of the VRP, DCM and CoM are constant in the second and the third direction, because the robot Bert (s. chapter 2.6) can only place the CoM in one plane and therefore the second and third coordinate are constant in an ideal trotting walk.

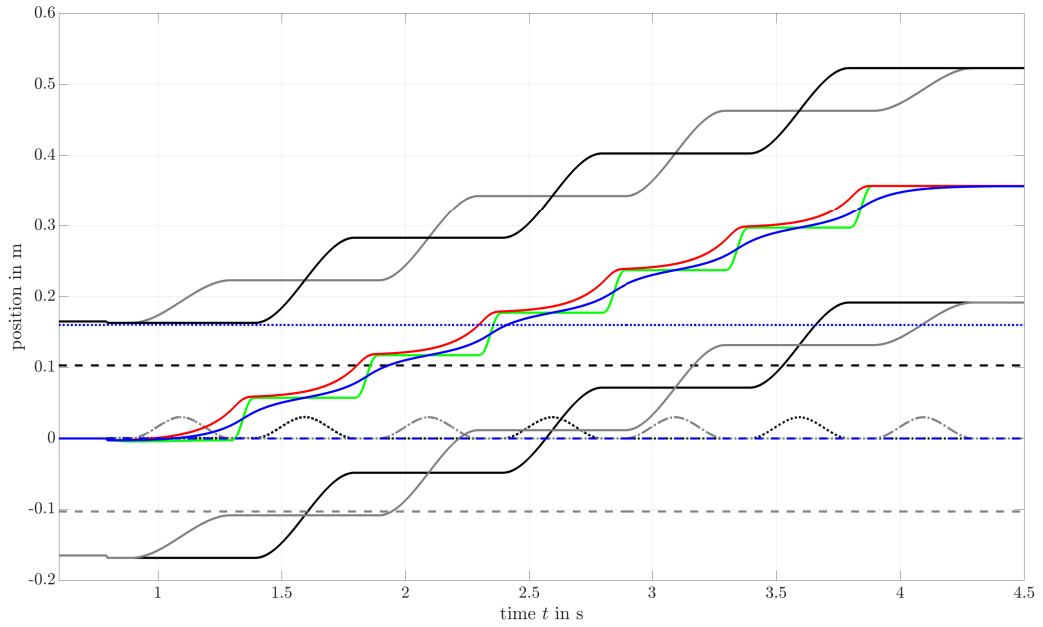


Figure 3.7: Example of the trajectory method for $p = 5$. The reference created for the values in table 3.2 and a forward trotting robot. The green solid line indicates the VRP, the red line the DCM and the blue line the CoM in the first coordinate. The dashed lines are the second direction and the dotted lines are the third direction. The gray lines indicate the limb reference of legs 1 and 4. The black lines indicate the limb reference of legs 2 and 3. The solid black and gray lines are the interpolation in the forward direction (w_1), the dashed lines the horizontal direction (w_2) and the dotted lines are the vertical direction (w_3).

Table 3.2: Parameters for example trajectory generation in figure 3.7

Variable	Value
Δz_{vrp}	0.16 m
T_{DS}	0.1 s
T_{SS}	0.4 s
Δx_{step}	0.06 m
Δh_{step}	0.03 m
n_{steps}	6
gait type	trotting

3.4 Joint Reference Generation

The previously shown trajectory generation method gives the reference for the CoM and the limbs.

These references can be applied to the joint reference generation by the resolved motion rate approach. In this solution method for the inverse kinematic problem, the Jacobian matrices of the robot are used in a pseudo inverse $\mathbf{J}^+ \in \mathbb{R}^{N \times m}$ in combination with a feedback of the error signal $\Delta \mathbf{r}_f$ of the task coordinated as reported by Siciliano (1990) and Siciliano et al. (2003) in a transformation $\dot{\mathbf{q}} = \mathbf{J}^+(\dot{\mathbf{r}}_f + \Delta \mathbf{r}_f)$.

Further ${}^W_h \mathbf{J}_{W,T} \in \mathbb{R}^{m \times N}$ is the hybrid Jacobian of the trunk link of the robot, where m is the number of task DOF, and ${}^W_h \mathbf{J}_{W,k_i}$ is the hybrid Jacobian of the shank link of a robot's leg. And $\mathbf{r}_{f_{id}} \in \mathbb{R}^3$ is the desired foot point position in the world and $\dot{\mathbf{r}}_{f_{id}} \in \mathbb{R}^3$ is the desired velocity of this foot point. Furthermore, $\mathbf{x}_d \in \mathbb{R}^3$ is the desired CoM position and $\dot{\mathbf{x}}_d \in \mathbb{R}^m$ is velocity in the world frame. With the propagation formula (Englsberger, 2016) the hybrid Jacobian of $\mathbf{r}_{f,i}$ can be determined to

$${}^W_h \mathbf{J}_{W,f_i} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -{}^W \mathbf{R}_{k_i} [{}^{k_i} \mathbf{r}_{k,f_i} \times] {}^W \mathbf{R}_{k_i}^T \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} {}^W_h \mathbf{J}_{W,k_i} \quad (3.4)$$

where ${}^{k_i} \mathbf{r}_{k,f_i} = [0, 0, -a]^T$, $[\mathbf{r} \times]$ is the skew operator for a three-dimensional vector and ${}^W \mathbf{R}_{k_i}$ is the rotation matrix between a shank link and the world. Now the relative Jacobian between the trunk link and the foot point result in

$${}^W_h \mathbf{J}_{T,f} = \begin{bmatrix} {}^W_h \mathbf{J}_{W,f_1} \\ {}^W_h \mathbf{J}_{W,f_2} \\ {}^W_h \mathbf{J}_{W,f_3} \\ {}^W_h \mathbf{J}_{W,f_4} \end{bmatrix} - \begin{bmatrix} {}^W_h \mathbf{J}_{W,T} \\ {}^W_h \mathbf{J}_{W,T} \\ {}^W_h \mathbf{J}_{W,T} \\ {}^W_h \mathbf{J}_{W,T} \end{bmatrix} \quad (3.5)$$

Equation (3.5) can be divided in two parts. The first part comprises the columns which describe the transformation of the base DOF. The second part are the N columns, \mathbf{J}_A , which describe the transformation of the actuated DOF. The pitch angle velocity feedback is used for a compensation of the trunk pitch rotation.

$${}^W_h \mathbf{J}_{T,f} = [\dots, \dot{j}_{pitch}, \mathbf{J}_A] \quad (3.6)$$

The pseudo inverse can be calculated with equation (3.7), where \mathbf{D} and \mathbf{W} are the diagonal, square regularization and weighting matrices.

$$\mathbf{J}^+ = (\mathbf{J}_A^T \mathbf{W} \mathbf{J}_A + \mathbf{D})^{-1} \cdot \mathbf{J}_A^T \mathbf{W} \quad (3.7)$$

These above mentioned equations can be applied to calculate with the current foot point and CoM position the desired joint velocity $\dot{\mathbf{q}}_d$ to

$$\dot{\mathbf{q}}_d = \mathbf{J}^+ \{ \dot{\mathbf{r}}_{f_d} - \dot{\mathbf{x}}_{A,d} + \mathbf{K}_1 [(\mathbf{r}_{f_d} - \mathbf{x}_{A,d}) - (\mathbf{r}_f - \mathbf{x}_A)] + j_{pitch} \dot{q}_{pitch} \} \quad (3.8)$$

where \dot{q}_{pitch} is the current pitch angular velocity, $\mathbf{K}_1 \in \mathbb{R}^{m \times m}$ is a diagonal, square gain matrix and

$$\mathbf{r}_{f(d)} = \begin{bmatrix} \mathbf{r}_{f1(d)} \\ \dots \\ \mathbf{r}_{f4(d)} \end{bmatrix} \in \mathbb{R}^m \quad (3.9)$$

$$\dot{\mathbf{r}}_{f(d)} = \begin{bmatrix} \dot{\mathbf{r}}_{f1(d)} \\ \dots \\ \dot{\mathbf{r}}_{f4(d)} \end{bmatrix} \in \mathbb{R}^m \quad (3.10)$$

$$\mathbf{x}_{A,(d)} = \begin{bmatrix} \mathbf{x}_{(d)} \\ \dots \\ \mathbf{x}_{(d)} \end{bmatrix} \in \mathbb{R}^m \quad (3.11)$$

$$\dot{\mathbf{x}}_{A,(d)} = \begin{bmatrix} \dot{\mathbf{x}}_{(d)} \\ \dots \\ \dot{\mathbf{x}}_{(d)} \end{bmatrix} \in \mathbb{R}^m \quad (3.12)$$

are the stacked position and velocity vectors of the CoM and foot points. The damping matrix entries are selected to small values (e.g. of 1e-12) in order to prevent singularity issues. The presented formula is later applied to provide the joint reference for a point-to-point control. For that, the given joint velocity reference is numerically integrated in the framework. Figures A.2 and A.3 in the appendix show a reference example for the values in table 3.2 based on the applied method.

4 Implementation of the Method

This chapter contains a short description about the implementation of the DCM method for quadrupedal locomotion on the robot Bert. It is divided in three parts. First, the implementation framework and the model is described in subchapter 4.1, followed by a summary of the motion planning in section 4.2. Finally the applied motion control is presented in subchapter 4.3.

4.1 Robot model description and Framework

In order to test and simulate the formulated DCM-based method a model of the robot Bert (s. chapter 2.6) is implemented into an existing Simulink model. The robot's kinematic model is thereby described and defined by a XML-based *Universal Robotic Description Format* (URDF) file (Sucan & Kay, 2019) (s. in appendix A.4). In the Simulink model the *LucaDynamics*-library (DLR's custom dynamics library) is used to provide forward kinematic functions and values of the state variables of the robot (\mathbf{y} , $\dot{\mathbf{y}}$ etc.).

The Simulink model (s. appendix A.4) has five subsystems and figure 4.1 shows a schematic overview of the model. The numerical calculation of the robot model is placed in the block *robot*. The step size must be selected to $50 \mu s$ or smaller for a stable calculation of the multi body model.

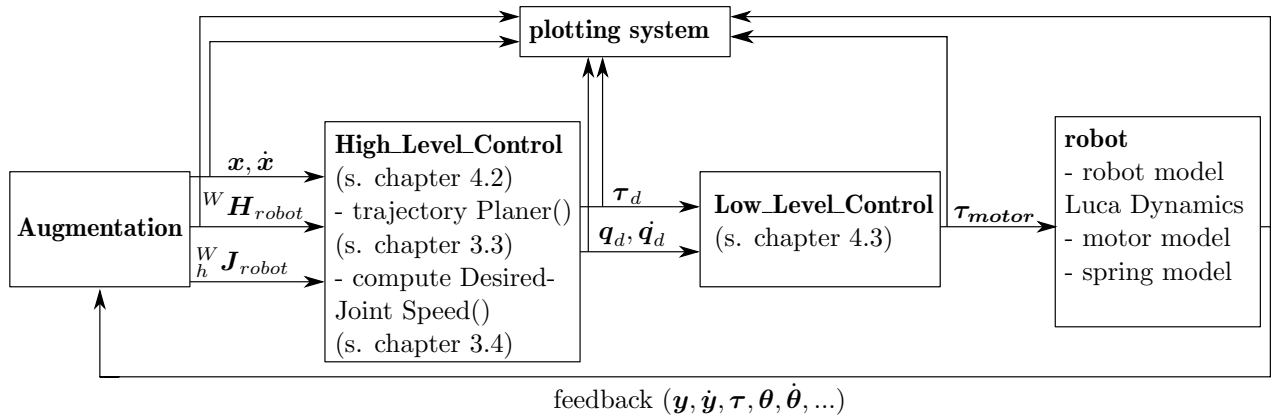


Figure 4.1: Overview of the extended Simulink model. The bold titles are the names of the subsystem blocks and the arrows show the applied signals. ${}^W\mathbf{H}_{robot}$ represent the global matrix of the homegenues transformation matris of the robot links and ${}^W_h\mathbf{J}_{robot}$ represent the global hybrid Jacobian matrizes of the robot links. The robot model of Bert has 9 links.

This block comprises the LucaDynamics model function (robot multi-body dynamics, contact model (s. (2.44))), the motor model and spring model of the joints (s. appendix

A.3.1). The block *Augmentation (AUG)* contains the augmentation of the feedback bus of the robot's model block and provides kinematic and dynamic information. These signals contain the homogeneous transformation matrices from the link frames to the world frame, the hybrid Jacobian matrices of the links and the CoM values in the world frame. The next block *High_Level_Control (HLC)* includes an extension of the reference generation presented in this thesis. The extension encompasses two versions. The first version is an open loop approach, which applies a simple analytic inverse for each planar robot leg (s. A.1) for the joint reference. The second version applies the method presented in chapter 3.4. The motor control for the simulation model is placed in the block *Low_Level_Control (LLC)s* and apply the desired joint reference. The last block *Plotting System* represents a subsystem for the signal plotting and output of these for later documentation.

The kinematic model of the robot is formulated in the file *URDFrealBert2.urdf* and allows a simulation of the robot Bert version *greyBert* with eight DOF (s. chapter 2.6) with a floating base of three DOF ($r_{b_1}, r_{b_2}, \psi_{b_2}$). The DOF of the roll angle is neglected for simulation, because only a small angle movement was observed. The floating base is placed in the origin of trunk frame with an equal orientation (s. figure 2.4). The foot point of each leg is modeled by one small capsule.

The extension of the framework allows to command a number of steps for the robot in trunk forward direction and to perform these steps. The robot moves from an initial stance to a final stance with parallel front and hind feet.

The Simulink model *controller.slx* (s. in the appendix A.4) contains an existing control framework for *greyBert* (control environment of Lakatos et al. (2019, 2018); Seidel et al. (2020)) and is extended in this thesis (sub-model *trajgen_walking.slx*). This framework could be applied for experiments. Furthermore, MATLAB functions are implemented as a workaround for the used *LucaDynamics* forward functions, because no library for these functions exists for the processor of the Bert robot yet.

4.2 Motion Planning

The extension of the framework consists of two main MATLAB function blocks. The first one is called the *trajectoryPlanner*. In this block the reported DCM functions of chapter 2.5 are implemented (s. folder *DCMfunctions* and *MultiContactPlanner*). It calculates for given parameters of the user control bus a desired stance sequence in space in forward body direction, i.e. the quantities $\mathbf{v}(t), \boldsymbol{\xi}(t), \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{r}_f(t)$ and $\dot{\mathbf{r}}_f(t)$. The time step Δt is chosen equal to 1 ms and the interpolation between the way points (s. figure 2.3) is calculated at each time step. Thereby the iteration from one to the next waypoints of $\mathbf{X}, \boldsymbol{\Xi}$ and \mathbf{V} is based on the time duration phases T_φ . In the subsequent simulation, the duration times for the double and single support phases are selected globally. In the implementation the set duration times and contact points are mapped to the needed number of fixed and transition phases in order to realize steps represented by the selected stances. After the designated number of steps has been completed, the system can be triggered again and applies the current stance as a new initial stance $\boldsymbol{\sigma}_1$.

The robot Bert can only place the CoM in the sagittal plane and thus can only control the CoM in this plane (s. chapter 2.6). Therefore the VRP planning is implemented accord-

ing to the procedure *greyBert VRP planning* in A.2. In that, the geometric center of the support polygon is projected along the forward trunk direction and ensures no violation of the stability margin. The stability margin is selected as minimum distance to the next support polygon edge in the case of stance with more than two contact feet. Furthermore, two functions are implemented for the VRP placement based on the above described procedure (s. chapter A.2) in which one function always considers the stability margin and the other continuously places the next VRP forward, i.e. places the eCMP on a support polygon edge in the cases that the last VRP is on the edge or the next VRP waypoint can not be placed behind the last one. This leads to no planned backward movement of the CoM, but involves the risk of tilting, since this point is not a suitable rest position for a static equilibrium (compare chapter 2.4 and 3.1). Therefore this should and is only set for stances which are not the final stance or the start stance, because in the other stances one point of the base of support chosen as planned eCMP (or CoP) (s. chapter 3.1).

The contact forces are considered to be feasible with the definitions and assumptions (s. chapter 3.1) made for the real Bert robot, i.e. be within the friction cone for the assumed ground surface and be always $\mathbf{f}_{C,i3} \geq 0$.

The second main MATLAB-function block is called the *computeDesiredJointSpeed*, which contains the calculation according to chapter 3.4. The current CoM and foot point position is thereby calculated by the forward kinematic in equations (4.1) and (4.2), where $\Delta \mathbf{h}_{CoM}$ is the offset vector of the trunk frame origin to the CoM.

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = {}^W \mathbf{H}_T \begin{bmatrix} {}^T \Delta \mathbf{h}_{CoM} \\ 1 \end{bmatrix} \quad (4.1)$$

$$\begin{bmatrix} \mathbf{r}_{f,i} \\ 1 \end{bmatrix} = {}^W \mathbf{H}_{f_i} \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ 1 \end{bmatrix} \quad (4.2)$$

4.3 Motion Control

The selected control law for the point-to-point control of the robot's joint reference (s. chapter 3.4) in simulation is a *proportional-derivative* (PD)-law according to de Luca and Book (2016) for robots with flexible joints in absence of gravity. Latter can be assumed for the leg swing-phases, because the mass of the legs can be neglected (s. chapter 3.1). Further, with this control law it is desired to stabilize the desired equilibrium state $\mathbf{q} = \boldsymbol{\theta} = \mathbf{q}_d = \boldsymbol{\theta}_d, \dot{\mathbf{q}} = \dot{\boldsymbol{\theta}}$ (de Luca & Book, 2016). This control approach is chosen for a first investigation of the generated joint trajectories in the simulation. The control law is determined to

$$\boldsymbol{\tau}_{motor} = \mathbf{K}_P(\boldsymbol{\theta}_d - \boldsymbol{\theta}) - \mathbf{K}_D\dot{\boldsymbol{\theta}} \quad (4.3)$$

where $\mathbf{K}_D \in \mathbb{R}^{N \times N}$ and $\mathbf{K}_P \in \mathbb{R}^{N \times N}$ are positive-definite and symmetric gain matrices de Luca and Book (2016).

Further, the desired motor position $\boldsymbol{\theta}_d$ is determined with the right side of equation (4.4) for the contact stance needed joint torque $\boldsymbol{\tau}_d = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}})$, where $\mathbf{K}_s \in \mathbb{R}^{N \times N}$ is the symmetric (diagonal and with entries of k) spring stiffness matrix.

$$\boldsymbol{\tau} = \mathbf{K}_s(\boldsymbol{\theta} - \mathbf{q}) \quad \rightarrow \quad \boldsymbol{\theta}_d = \mathbf{K}_s^{-1}\boldsymbol{\tau}_d + \mathbf{q}_d \quad (4.4)$$

The above equation are implemented, because the already implemented motor model in the simulation require a desired motor torque which is the reference for τ (s. appendix A.3.1). The joint springs of the SEA are assumed to be linear in robot system and the gain value identified for the *greyBert* (s. chapter 2.6) are reported in table 4.1.

Table 4.1: Values of K_P and K_D for the robot Bert

Symbol	Description	Value
k_{p1}	Gain for hip joints	16 Nm/rad
k_{p3}	Gain for knee joints	1.6 Nm/rad
k_{d1}	Damping gain for hip joints	3.375e-2 (Nm · s)/rad
k_{d3}	Damping gain for knee joints	4e-3 (Nm · s)/rad
k_{d13}	Damping gain between joints	4.8e-3 (Nm · s)/rad

The present motor controller runs in the simulation at 1 kHz. The values in table 4.1 are the results of a manually tuning in order to achieve an equilibrium state for constant commanded joint values. Note that equations (4.3) and (4.4) are only planned to apply in the simulation, because in the real robot system a position based controller is already implemented.

The real Bert controller applies a motor position control-law for θ_d , i.e. only require a desired motor position. For the later experiments will be $\theta_d = \mathbf{q}_d$ applied with respect to de Luca and Book (2016).

5 Simulation

In this chapter selected simulation results are reported for the trotting gait and walk gait. These simulation runs were chosen to show the observed behavior of the robot model in combination with the present motion planning and control ¹. The chapter is divided in three parts. In the first section 5.1 the simulation base is described, followed by results of the trotting gait 5.2. In section 5.3 results of the static walk are reported.

5.1 Simulation base

Due to the reported restriction of the applied robot system (s. chapter 2.6) only the trotting and walk gait were considered and tested. In the pacing gait, the robot has to move the CoM also in its transverse plane, because the planned eCMP are placed alternating on the line of support of the lateral foot points, which are in stance (s. chapter 3.2). A CoM movement in the transverse plane can not be controlled by the robot Bert (s. chapter 2.6). The simulation of these trotting and walk gaits had the purpose to test the kinematic and dynamical realization of the present method for the robot Bert. Therefore, the model's behavior was investigated using the present stance-based gait definition and the DCM-generated CoM references. Afterwards, experiments should be conducted on the real robot system.

The robot model was placed in an initial position short of the ground and its t_1t_3 -plane was set identically to the w_1w_3 -plane. At the start of the simulation the robot was dropped on the model ground. After the time required to reach a steady standing, i.e. no strong movement of the feet, the system was manually triggered to start walking. This caused the robot to perform the desired number of steps.

The described design parameters (s. chapter 3.2) of the DCM-based motion planning were tuned during the simulation process. The aim of the tuning was to find a parameter set which allows the robot to achieve the planned references of the CoM and the limbs. Summarized, these design parameters of the gaits are:

- $T_{SS} \leftarrow$ duration of the single support phase, i.e. duration in which the limbs are swinging and the VRP is constant. In this thesis T_{SS} is constant and equal for all single support segments during walking.
- $T_{DS} \leftarrow$ duration of the double support phase, i.e. duration in which the the VRP transition is performed. In this thesis T_{DS} is constant and equal for all double support segments during walking.
- $\Delta z_{vrp} \leftarrow$ kinematic design parameter of the desired average height of the CoM over the ground. This parameter also determines b , the time-constant of the DCM dynamics

¹The simulation data of all performed runs can be found in the project folder (s. appendix A.4)

(s. (2.16)), due to $b = \sqrt{\frac{\Delta z_{vrp}}{g}}$.

- $p \leftarrow$ the degree of the interpolation function f_φ , which is applied for the interpolation between the VRP waypoints. A degree of 5 is recommended to increase the smoothness of the VRP, DCM and CoM trajectories and also for smoothed motor/joint torques, since the differentiability of the VRP trajectories increases with higher degrees (Englsberger et al., 2017).
- $\Delta x_{step} \leftarrow$ design parameter for the stance planning. The parameter determines the offset between two contact points of the front or hind leg pairs in the forward direction (s. chapter 3.2). For pacing and trotting this parameter can be selected to half of the desired step length. For the walk this parameter can be selected to a fourth of the step length. In this thesis the parameter is set globally and equally for all single steps.
- $\Delta h_{step} \leftarrow$ design parameter for the stance planning. This parameter determines the desired step height (s. chapter 3.2), which is used for interpolation in the vertical direction. In this thesis the parameter is set globally and equally for all single steps.

Note that the parameters T_{DS} , T_{SS} , Δx_{step} and Δh_{step} have not been set to constants for all steps in the motion planning.

In this work, however, constant parameters are used for each simulation run to investigate the behavior of the robot model when walking on a planar ground for different parameter sets.

5.2 Simulation results of the Trotting Gait

In this section the results of a selection of simulation runs are reported for the trot gait. Figure 5.1 shows the plotted results of the VRP signal in the first and second direction in the world frame of run 40 in table 5.1. Thereby, the real quantities of $\mathbf{v}(t)$ was calculated by equation (2.15) and for this the CoM acceleration was approximated with the `MATLAB-gradient()`. This was applied in all subsequent VRP plots, too. The VRP oscillates around the desired VRP reference (constant waypoint) in the single support phases and follows the reference in the transition phases (double support). Moreover, the plot shows an overshoot with each VRP waypoint transition phase. Figure 5.4 shows the quantities of the VRP, DCM and CoM in the first and third direction. In this plot (and in the other plots of run 40, too) the first 0.5 s show the dropping of the robot model on the ground and the transient oscillation. After the robot model ends with the step sequence, the VRP converges to the desired reference.

Furthermore, figure 5.2 shows the recorded DCM point of this run and figure 5.3 shows the records of the CoM calculated from the multi-body model and the CoM of the trunk link. The CoM of the trunk and the DCM calculated with this CoM are in the DCM and CoM plots reported (e.g. s. figures 5.3 and 5.2). The trunk CoM record is shown in the CoM plots, because its position is applied as reference in the joint reference generation (s. chapter 3.4). This is based on the model simplification of mass-less legs (s. chapter 3.1) and the assumption that only the CoM of the trunk is the relevant mass. Quantities caused by the

trunk CoM are plotted as brighter lines in the figures. The CoM of the trunk shown in figure 5.3 is close to the reference in third direction. But with each step the trunk CoM sinks a few millimeters in and then moves towards the reference. The CoM of the whole system (dark, solid line) is ≈ 1.5 mm lower than the trunk CoM and shows the same behavior in the third direction. Both CoMs show the same behavior in the first position and velocity direction. In figure 5.2 the signal which is caused by the trunk CoM oscillates around the reference in the third direction. In the first direction one can see that the DCM overshoots with the double support phase (compare also to figure 5.4). This corresponds to the overshoot of the CoM after the double support phase. In the double support phase the robot moves its CoM forward, but too far and the system steers it back after the overshoot. This slowing down during the single support phases leads to the desired minimum velocity being exceeded.

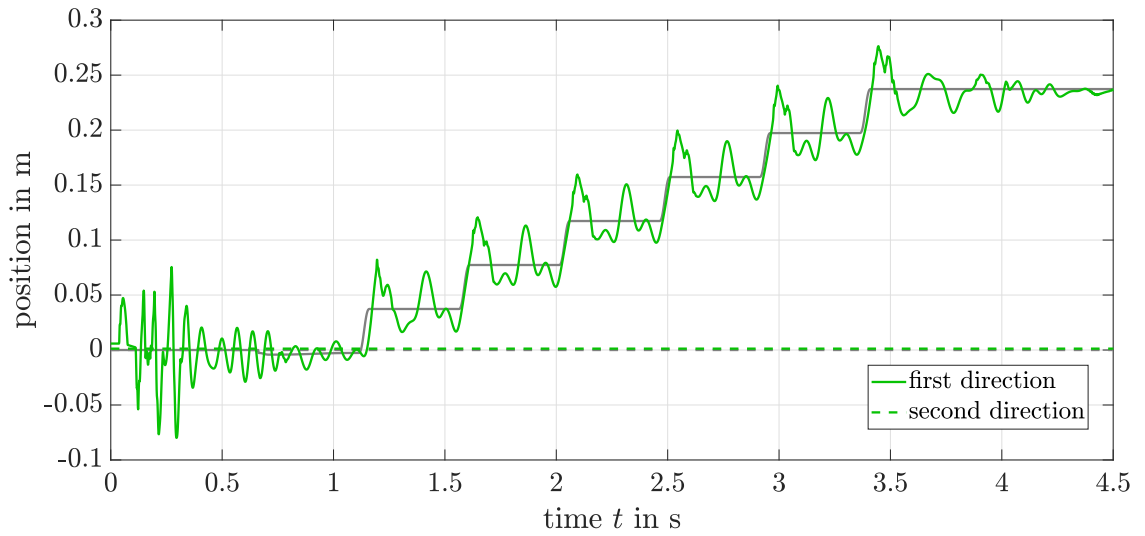


Figure 5.1: Simulation record of the VRP of the trotting walk in the first (w_1) and second direction (w_2). The green solid line represents the calculated VRP in the first direction (forward) and the dashed line represents the VRP in the second direction (horizontal). The gray lines indicate the VRP reference. This plot corresponds to run 40 in table 5.1.

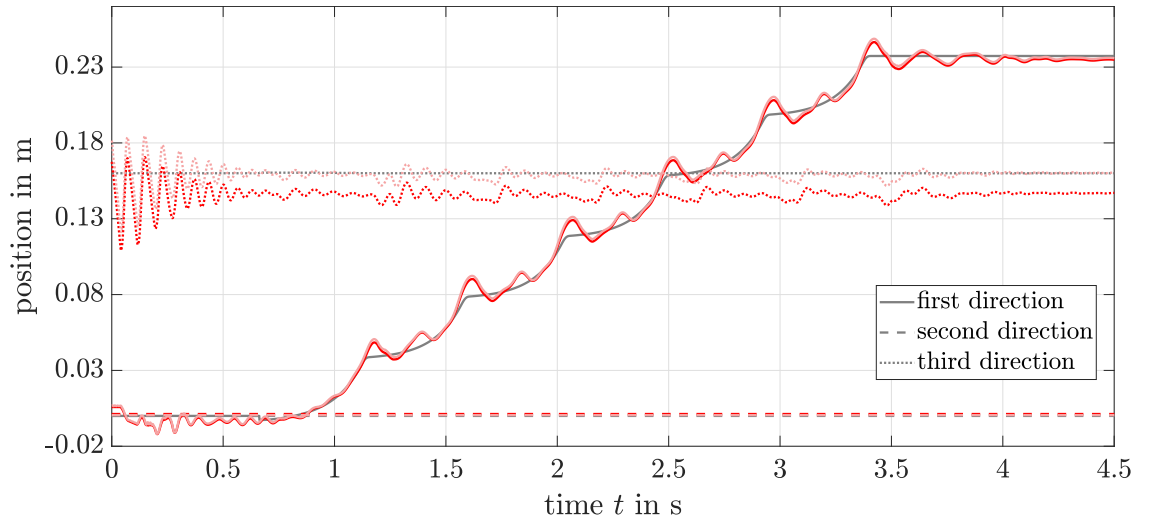


Figure 5.2: Simulation record of DCM of the trotting gait. The red solid line represents the DCM in the first direction (w_1 , forward). The dashed line indicates the DCM in the second direction (w_2 , horizontal) and the dotted lines indicates the DCM in the third direction (w_3 , vertical). The dark red lines correspond to a DCM calculation based on the whole-body CoM. The brighter red lines represent the DCM calculated for the trunk CoM. The gray lines represent the applied DCM reference. This plot corresponds to run 40 in table 5.1.

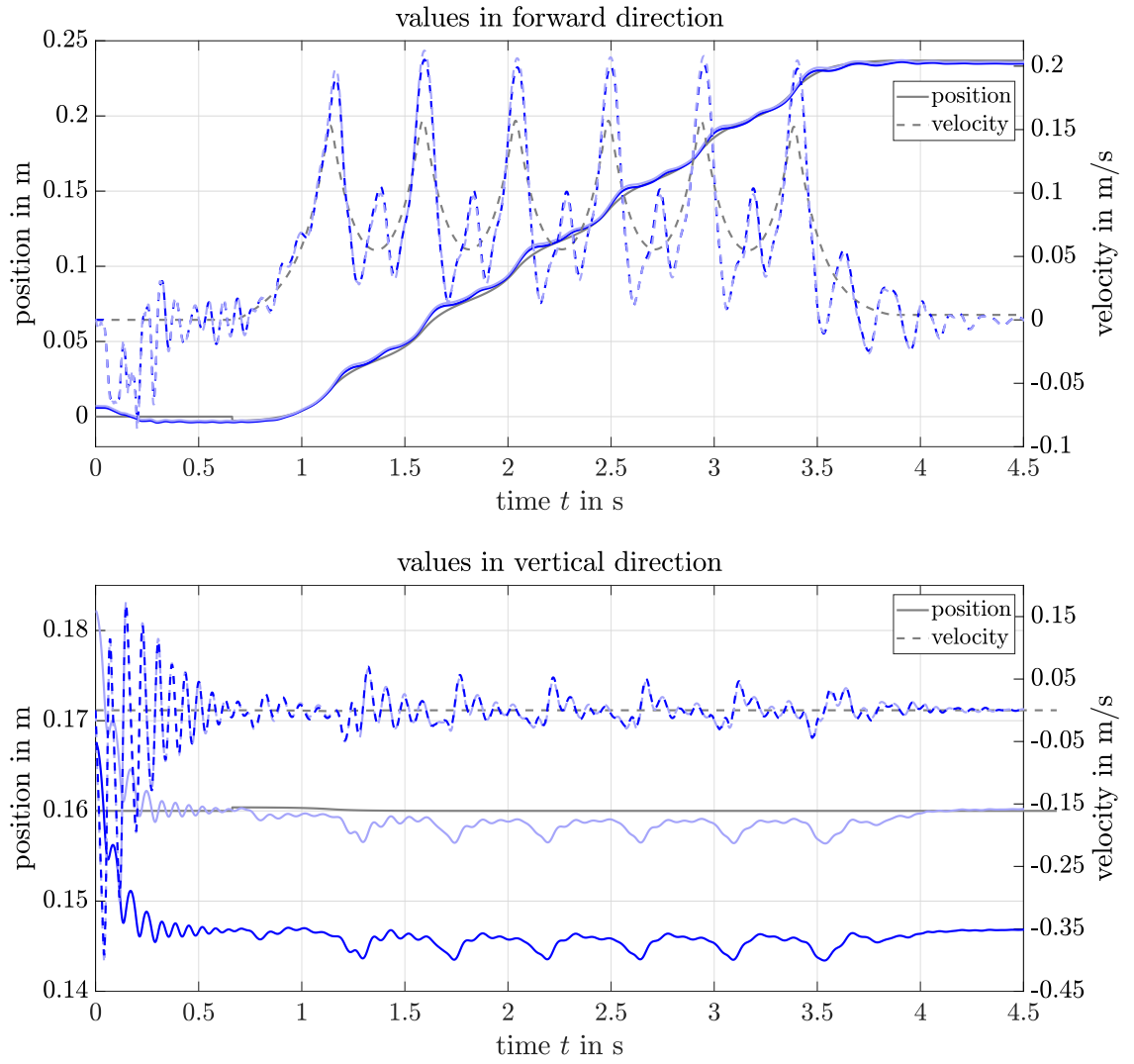


Figure 5.3: Simulation record of the CoM of the trotting gait. The top plot shows the record of the CoM in the first direction (w_1 , forward). The solid lines represent the positions and the dashed lines represent the CoM velocity in the first direction. The bottom plot shows the record of the CoM in the third direction (vertical). The solid lines are the positions and the dashed lines are the CoM velocity in the third direction. In both plots the blue dark lines represent the quantities of the CoM for the whole multi-body model (trunk mass + leg masses). The brighter blue lines represent the quantities of the CoM for the trunk mass and the gray lines indicate the applied CoM reference. On the right is the scale of the velocity. This figure corresponds to run 40 in table 5.1.

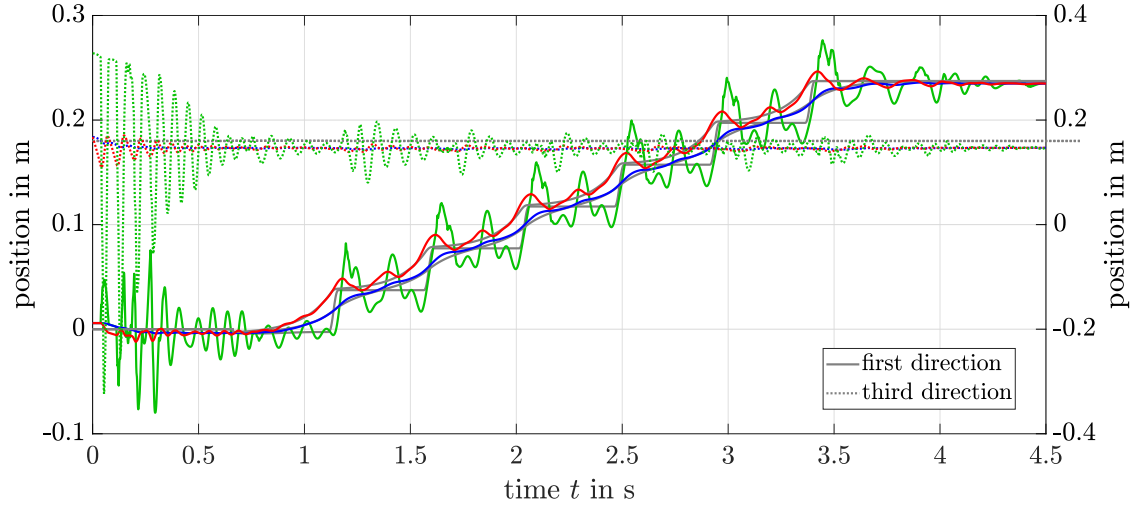


Figure 5.4: Plot of the quantities VRP (green lines), DCM (red lines) and CoM (blue lines) in the first (w_1) and third (w_3) direction. The solid lines represent the first direction and the dotted lines represent the third direction. On the right is the scale of the quantities in the third direction. The gray lines indicate the references. This figure corresponds to run 40 in table 5.1.

Moreover, figure 5.5 reports the recorded limb movement of run 40 in the world frame. The control was able to follow the foot point references with a delay of a few milliseconds. Furthermore, the robot model holds the foot points at the desired position during the stance phases of the legs. However, the desired reference in the vertical direction were not achieved. Figure 5.5 shows the joint reference and the recorded values of the signals. The values for the hip joints were sufficiently tracked. The same applies to the knee joints in the swing phases. In the stance phases (foot has contact with ground) the values provided by the joint generation method were not achieved.

Table 5.1: Parameters for selected simulation runs

Run	Δz_{vrp}	T_{DS}	T_{SS}	Δx_{step}	Δh_{step}	n_{step}	p	gait type
40	0.16 m	0.05 s	0.4 s	0.04 m	0.03 m	6	5	trotting
46	0.16 m	0.1 s	0.2 s	0.06 m	0.03 m	6	5	walk
65	0.17 m	0.4 s	0.5 s	0.05 m	0.02 m	6	5	trotting

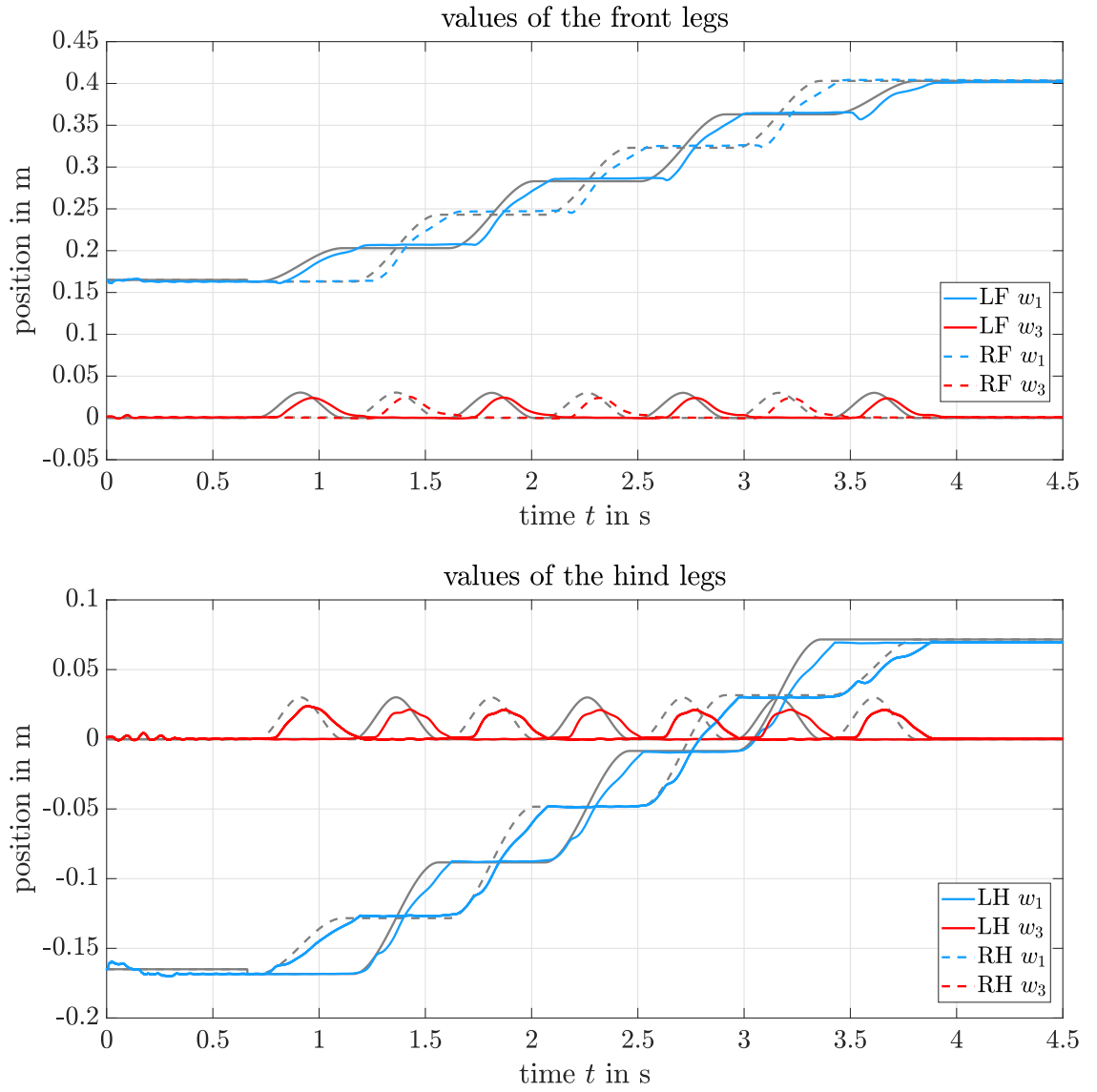


Figure 5.5: Simulation records of the robot limbs in run 40 in the world frame. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the first (w_1) direction and the red lines the third (w_3) direction. The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references. This figure corresponds to run 40 in table 5.1.

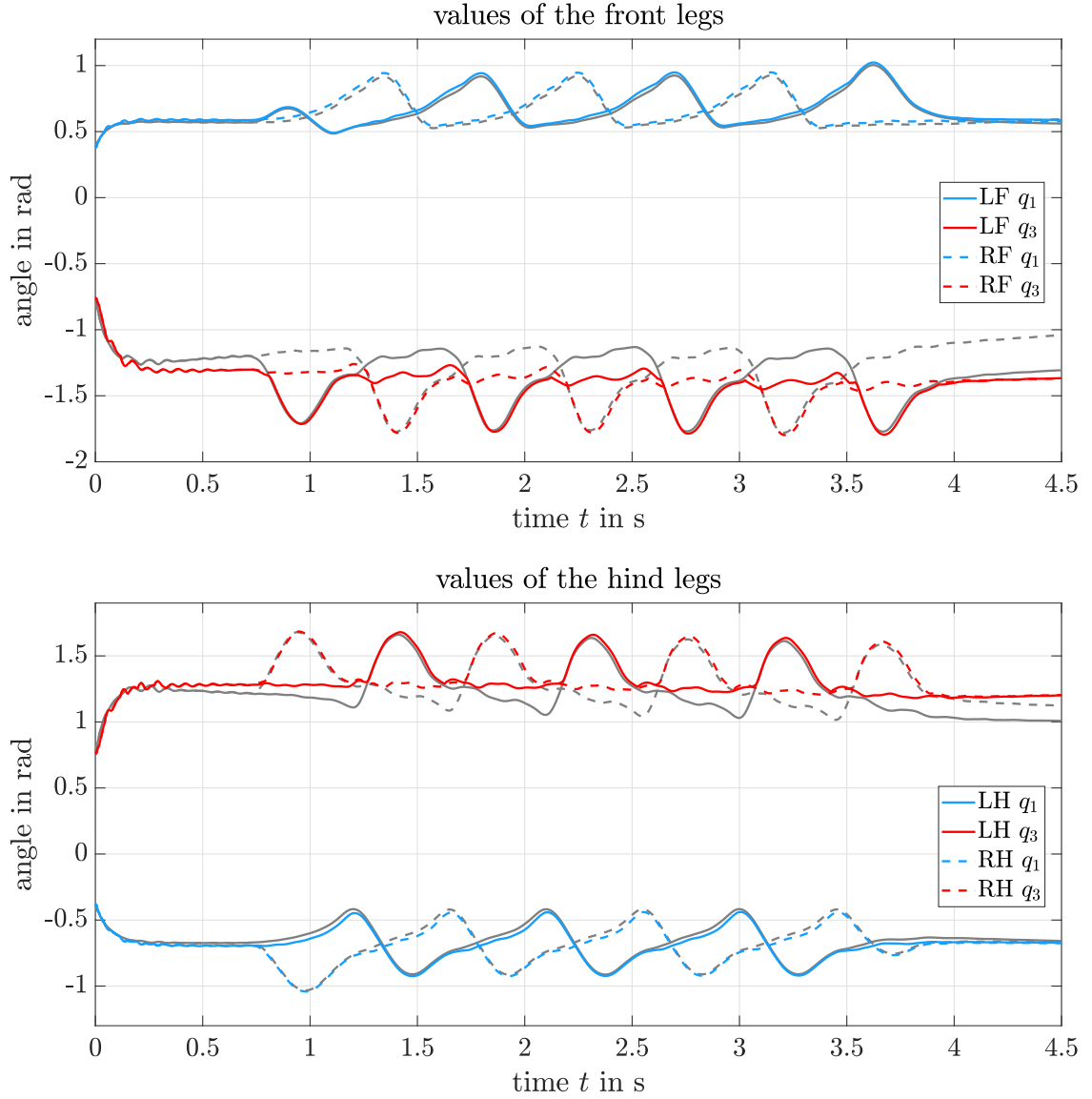


Figure 5.6: Simulation records of the robot's leg joints in run 40. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the hip joints (q_1) and the red lines the knee joints (q_3). The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references. This figure corresponds to run 40 in table 5.1.

On average, the DCM follows the desired reference and converges in the final stance to the final VRP. The CoM (of the trunk and the whole system) follows the reference in the first and third direction (w_1, w_3), but the CoM velocity oscillates strongly during the single support phase (s. figure 5.3) and overshoots at the end of a double support phase. The desired average maximum CoM forward velocity was 0.15 m/s and the record average maximum forward velocity was around 0.20 m/s at a step length of 0.08 m and a step time of 0.45 s. For run 65 larger T_{DS} and T_{SS} were selected (s. table 5.1), a lower step height and a higher half step length compared to run 40. Figure 5.8 shows the VRP during the simulation. For a better visualization of the model's behavior during the performed trotting gait the model's dropping to the ground was cut out. Figure 5.7 shows an example of the three first steps

of the trotting robot model of the simulation of run 65. The robot started in an initial stance and with a double support phase (s. figure 5.11). The stills show the sequence of the alternating single and double support phases.

The VRP still oscillates around the reference in the single support phases, but also in the double support phases. However, the strong overshoot at end of the double support phases does not occur in the first direction compared to the above presented simulation. In figure 5.9 the DCM has smaller oscillations in the first direction compared to 5.2 and is closer to the references. During the single support phases (time duration in which the VRP is constant, s. figures 5.11 and its detail plot 5.12, the DCM still oscillates. The CoM figure 5.3 shows the recorded CoM in the first direction close to the reference. During the single support phases one can see the same behavior in run 40. However, the slowing down of the CoM is stronger here and leads to a short backward movement of the CoM (s. figures 5.10 and 5.12).

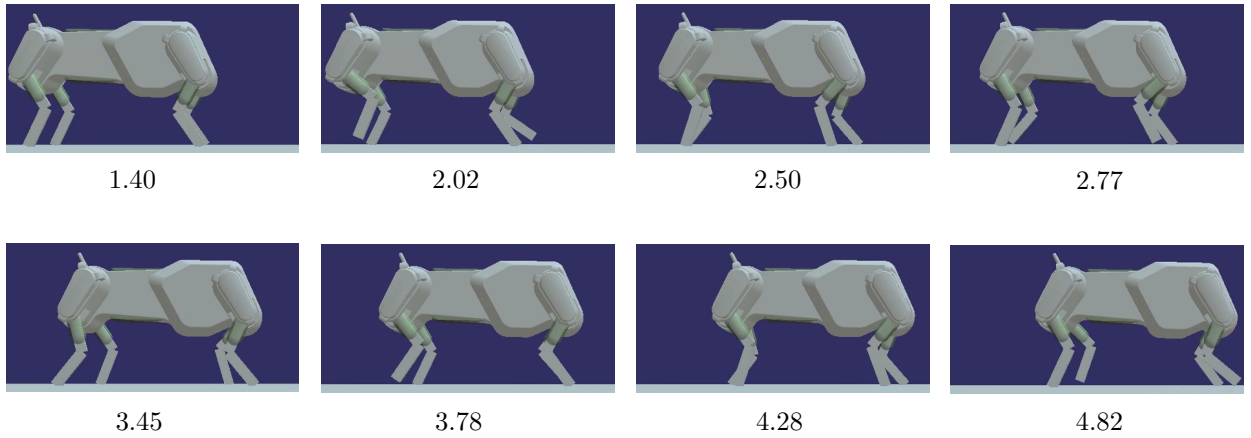


Figure 5.7: Sequence of stills of the simulation run 65 for the first three steps. The sequence begins in an initial stance and with a double support. After this, the alternating sequence of the double support phases (all four legs in stance) and single support phases (two legs in stance) are shown till the fourth single support. The bottom numbers indicate the time point of the stills corresponding to the plots of run 65 in seconds.

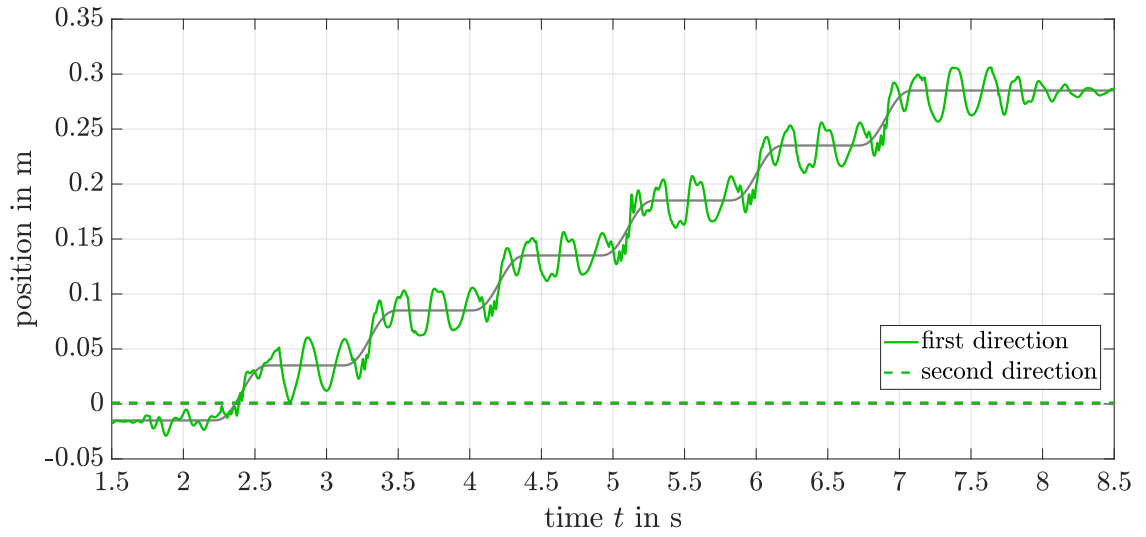


Figure 5.8: Simulation record of the VRP of the trotting walk in the first (w_1) and second direction (w_2). The green solid line represents the calculated VRP in the first direction (forward) and the dashed line represents the VRP in the second direction (horizontal). The gray lines indicate the VRP reference. This plot corresponds to run 65 in table 5.1.

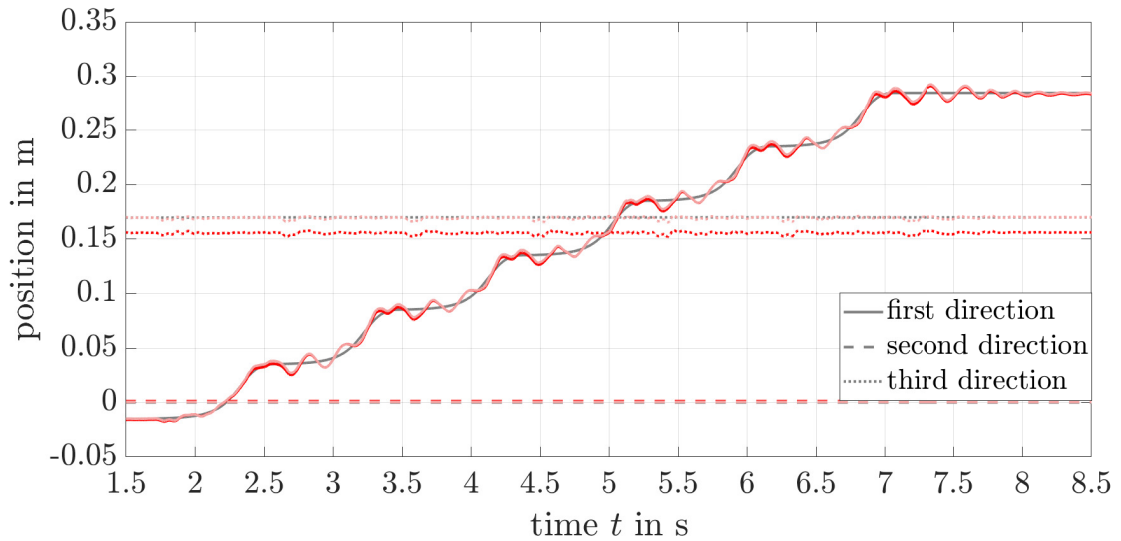


Figure 5.9: Record of DCM of the gait walk. The red solid line represents the DCM in the first direction (w_1 , forward). The dashed line indicates the DCM in the second direction (w_2 , horizontal) and the dotted lines in the third direction (w_3 , vertical). The dark red lines correspond to a DCM calculation based on the whole-body CoM. The brighter red lines represent the DCM calculated for the trunk CoM. The gray lines represent the applied DCM reference. This plot corresponds to run 65 in table 5.1.

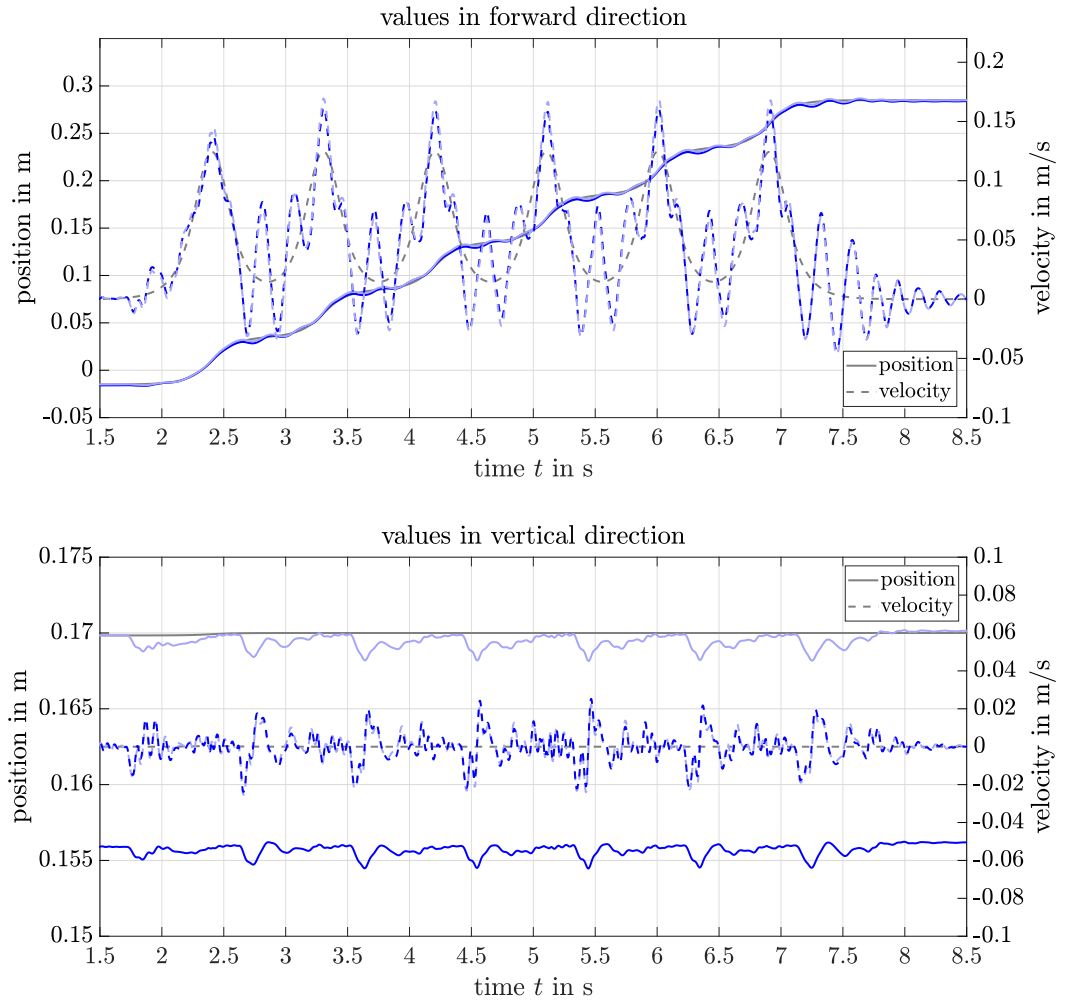


Figure 5.10: Record of CoM of the gait walk. The top plot shows the record of the CoM in the first direction (w_1 , forward). The solid lines represent the positions and the dashed lines represent the CoM velocity in the first direction. The bottom plot shows the record of the CoM in the third direction (w_3 , vertical). The solid lines represent the positions and the dashed lines the CoM velocity in the third direction. In both plots the blue dark lines represent the quantities of the CoM for the whole multi-body model (trunk mass + leg masses). The brighter blue lines represent the quantities of the CoM for the trunk mass and the gray lines indicate the applied CoM reference. On the right is the scale for the velocity. This figure corresponds to run 65 in table 5.1.

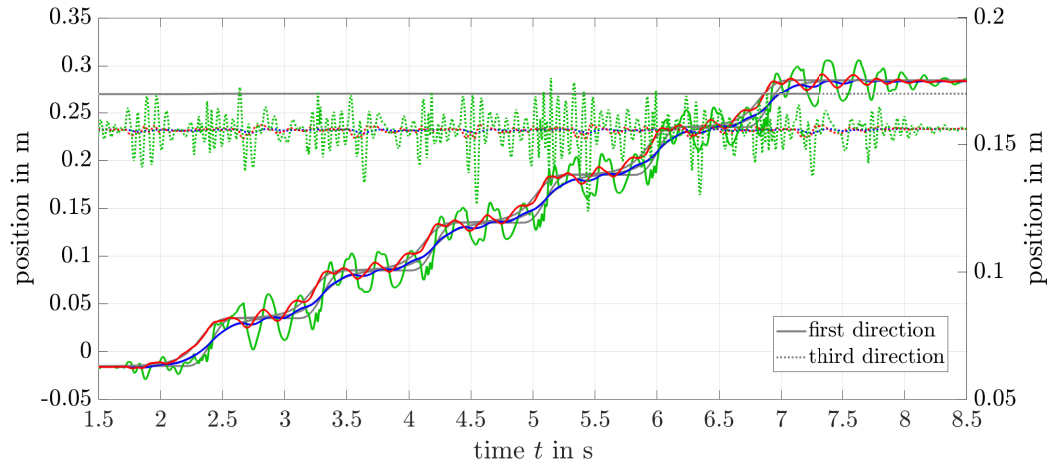


Figure 5.11: Plot of the quantities VRP (green lines), DCM (red lines) and whole-body CoM (blue lines) in the first and third direction. The solid lines represent the first direction and the dotted lines represent the third direction. On the right is the scale of the quantities in the third direction. The gray lines indicate the references. This figure corresponds to run 65 in table 5.1.

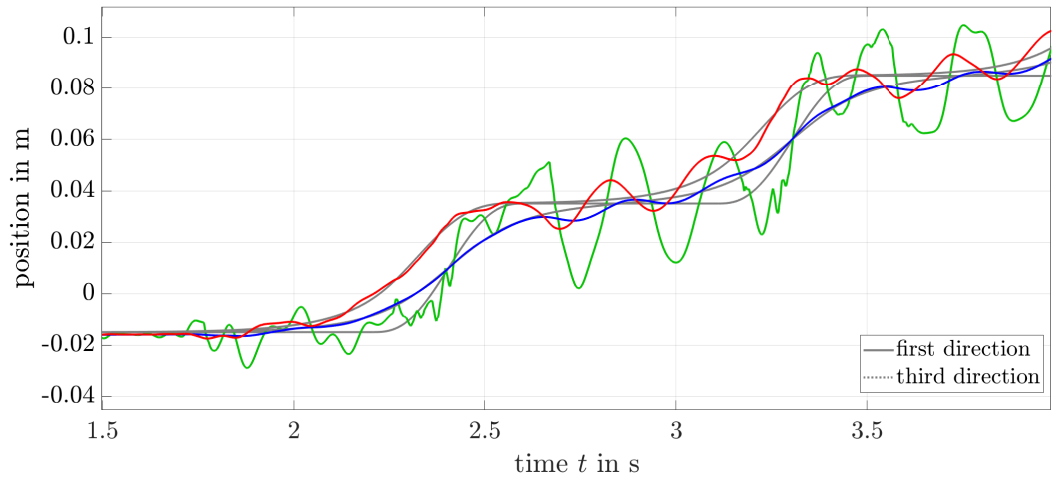


Figure 5.12: Detail plot of plot 5.11. This plot shows quantities VRP (green lines), DCM (red lines) and whole-body CoM (blue lines) in the first direction (w_1) for the first two seconds. The solid lines represent the first direction. On the right is the scale of the quantities in the third direction. The gray lines indicate the references. This figure corresponds to run 65 in table 5.1.

Figure 5.13 shows the limb movement during simulation 65. The plots show the same delay behavior as for the simulation of run 40. However, the model is here closer to the reference in the third direction and the feet movement in the first direction is smoother than in the previous run. After trying to take off a foot, one can see that the front feet detach a few ms to late from the ground. This is also true for the hind legs, but with a smaller effect for the movement in the first direction.

This behavior can be seen in the simulation run 40, too (s.figure5.5).

The joint record in figure 5.14 shows the same characteristics as in the previous simulation run. However, an improvement in the stance phases of the legs (compare 2.5 s till 3.5 s in bottom plot of figure 5.14, i.e. in the stance of RH) and in the swing phases of the legs is noticeable.

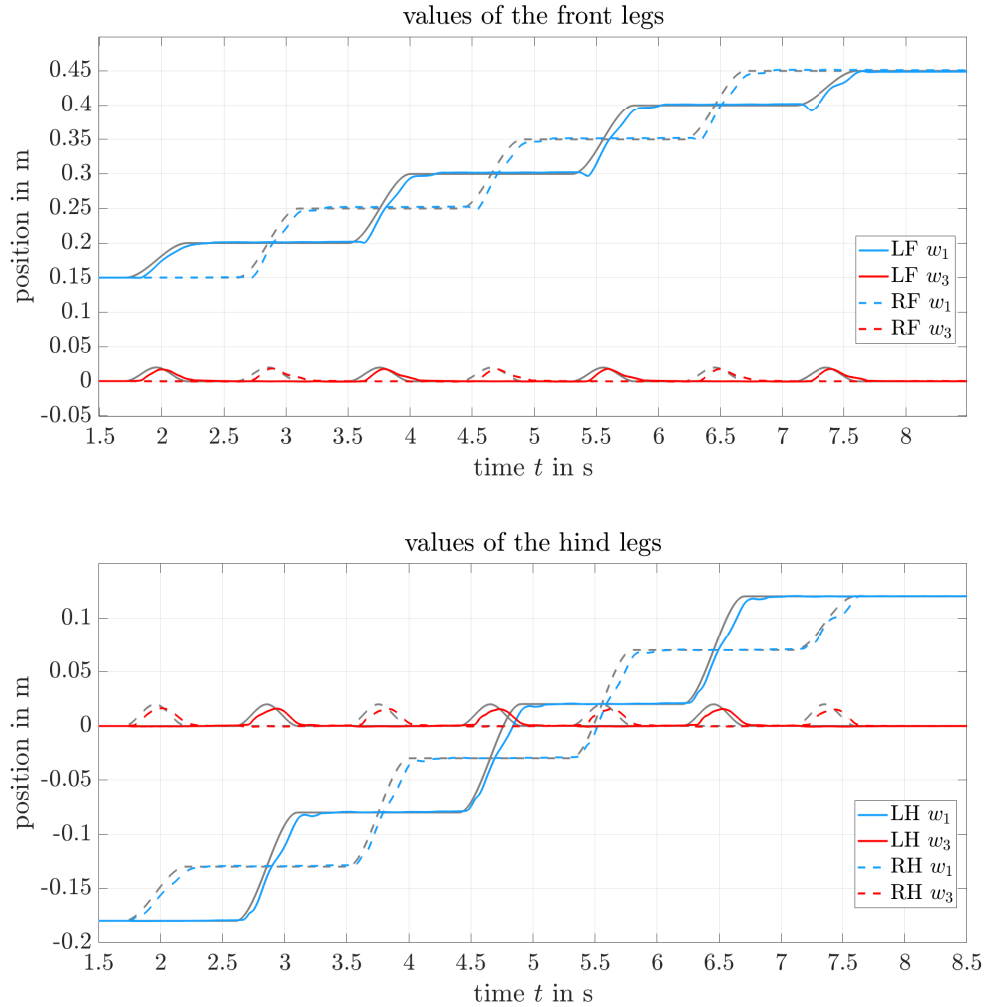


Figure 5.13: Records of the robot limbs in run 65 in the world frame. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the first direction and the red lines the third direction. The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references. This figure corresponds to run 65 in table 5.1.

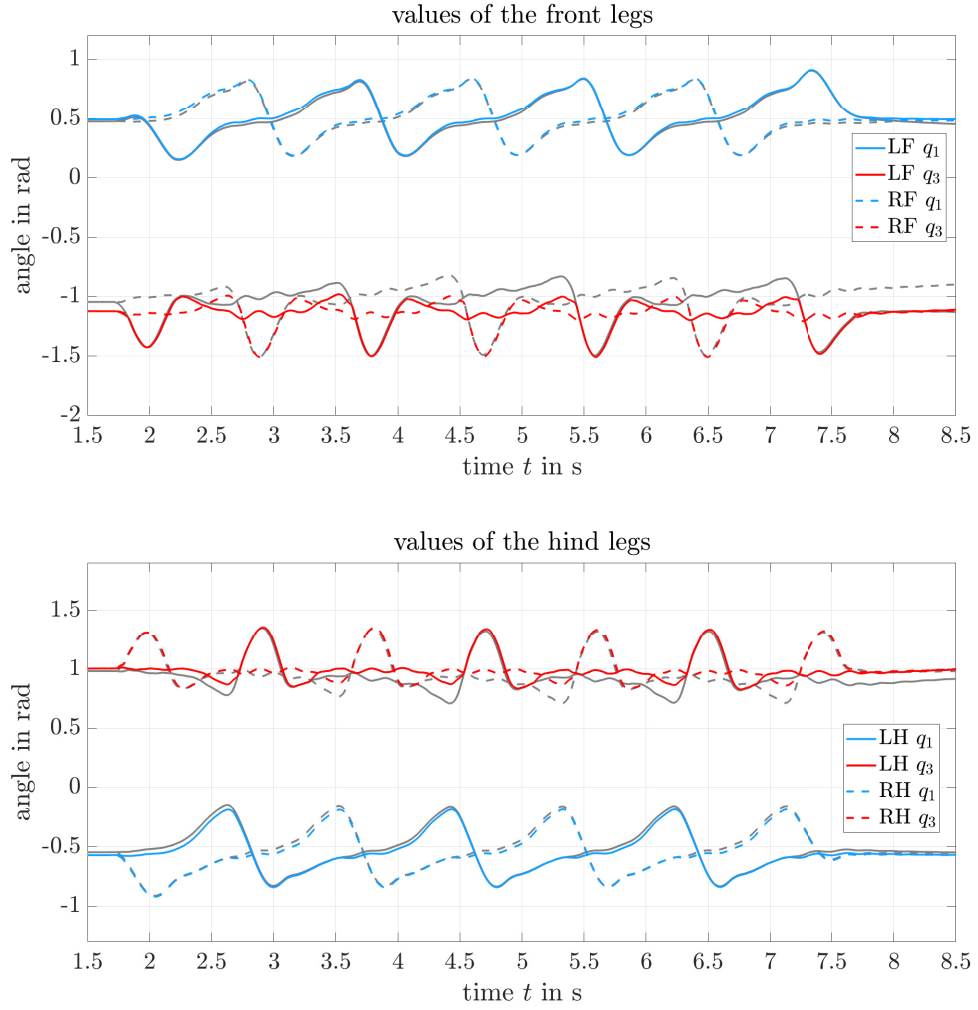


Figure 5.14: Records of the robot leg joints in run 65. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the hip joints (q_1) and the red lines the knee joints (q_3). The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references. This figure corresponds to run 65 in table 5.1.

On average, the DCM in simulation run 65 follows the desired reference and converges in the final stance to the final VRP. The CoM (of the trunk and the whole system) closely follows the reference in the first and third direction (w_1 , w_3), but the CoM velocity oscillates strongly during the single support phase, which leads to a back movement of the CoM (s. figure 5.10). Furthermore, the CoM overshoots at the end of a double support phase. The desired average maximum CoM forward velocity was ≈ 0.125 m/s and the record average maximum forward velocity was ≈ 0.16 m/s at a step length s_{len} of 0.10 m and a step time of 0.9 s.

The model was able to perform the desired number of steps in the above presented simulations of the trotting gait. However, the desired joint reference in the stance of a leg was not achieved for the knee joints and a oscillating of the quantities (VRP, DCM and CoM) occurred, which could be reduced by a larger step time, especially with a larger T_{DS} .

5.3 Simulation results of the Walk Gait

In this section the results of a simulation run for the walk gait are reported. Figure 5.15 shows the plotted results of the VRP signal of run 46 in the world frame in table 5.1. The signal in forward direction (first direction) shows a strongly oscillating VRP (or eCMP) along the references.

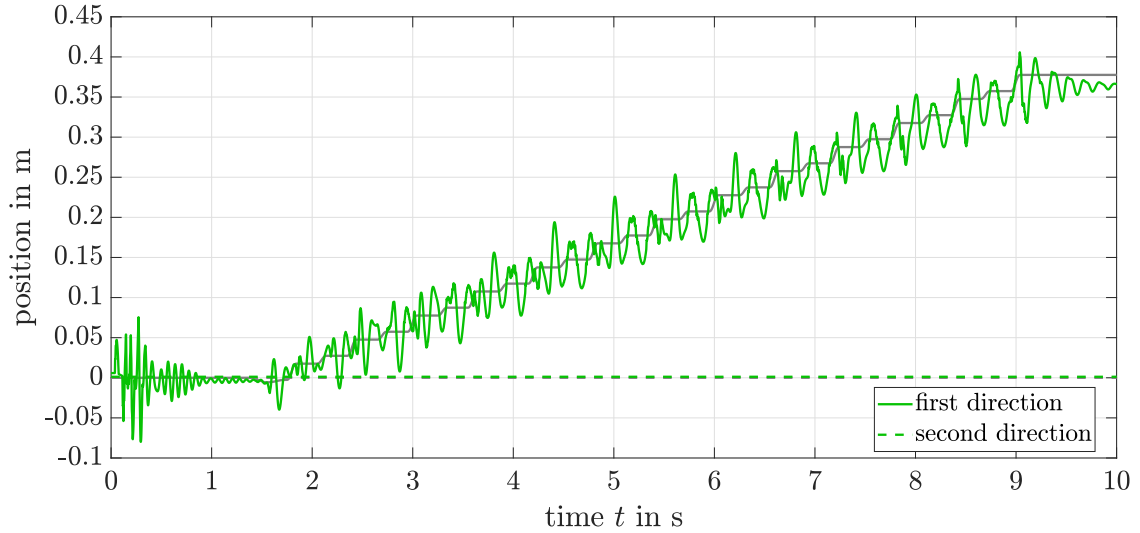


Figure 5.15: Record of the VRP of the gait walk on the ground. The green solid line represents the calculated VRP in the first direction (w_1 , forward) and the dashed line shows the VRP in the second direction (w_2 , horizontal). This plot corresponds to run 46 in table 5.1. The gray lines indicate the VRP reference. The applied waypoints were determined with the continuous forward version of the method introduced in chapter 4.2.

Furthermore, figure 5.16 shows the recorded DCM of this run and figure 5.17 shows the records of the CoM of the whole system and the CoM of the trunk link. The VRP plot in the forward direction indicates a shift after the end of the six global steps (six global steps are equal to 24 single walk steps). This becomes clear in the DCM plot (s. figure 5.16). The DCM is also oscillating strongly around the reference. The same is valid for the CoM. However, the CoM is closely to the reference in the first single step of a leg, but it also shifts away toward the end.

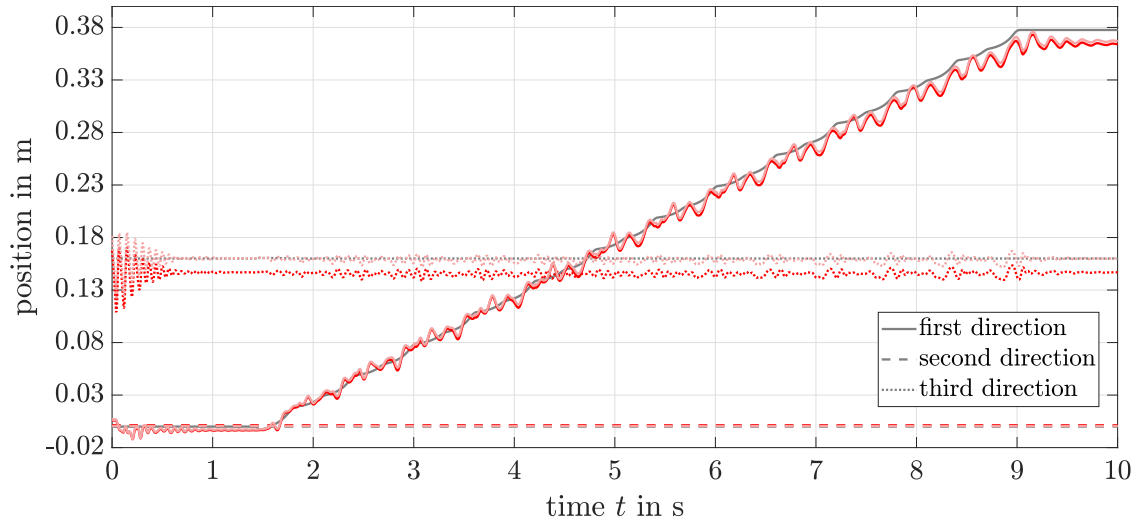


Figure 5.16: Record of DCM of the gait walk. The red solid line represents the DCM in the first direction (w_1 , forward). The dashed line shows the DCM in the second direction (w_2 , horizontal) and the third direction (w_3 , vertical). The dark red lines correspond to a DCM calculation based on the real CoM. The brighter red lines indicate the DCM calculated for the trunk CoM. The gray lines represent the applied DCM reference. This plot corresponds to run 46 in table 5.1.

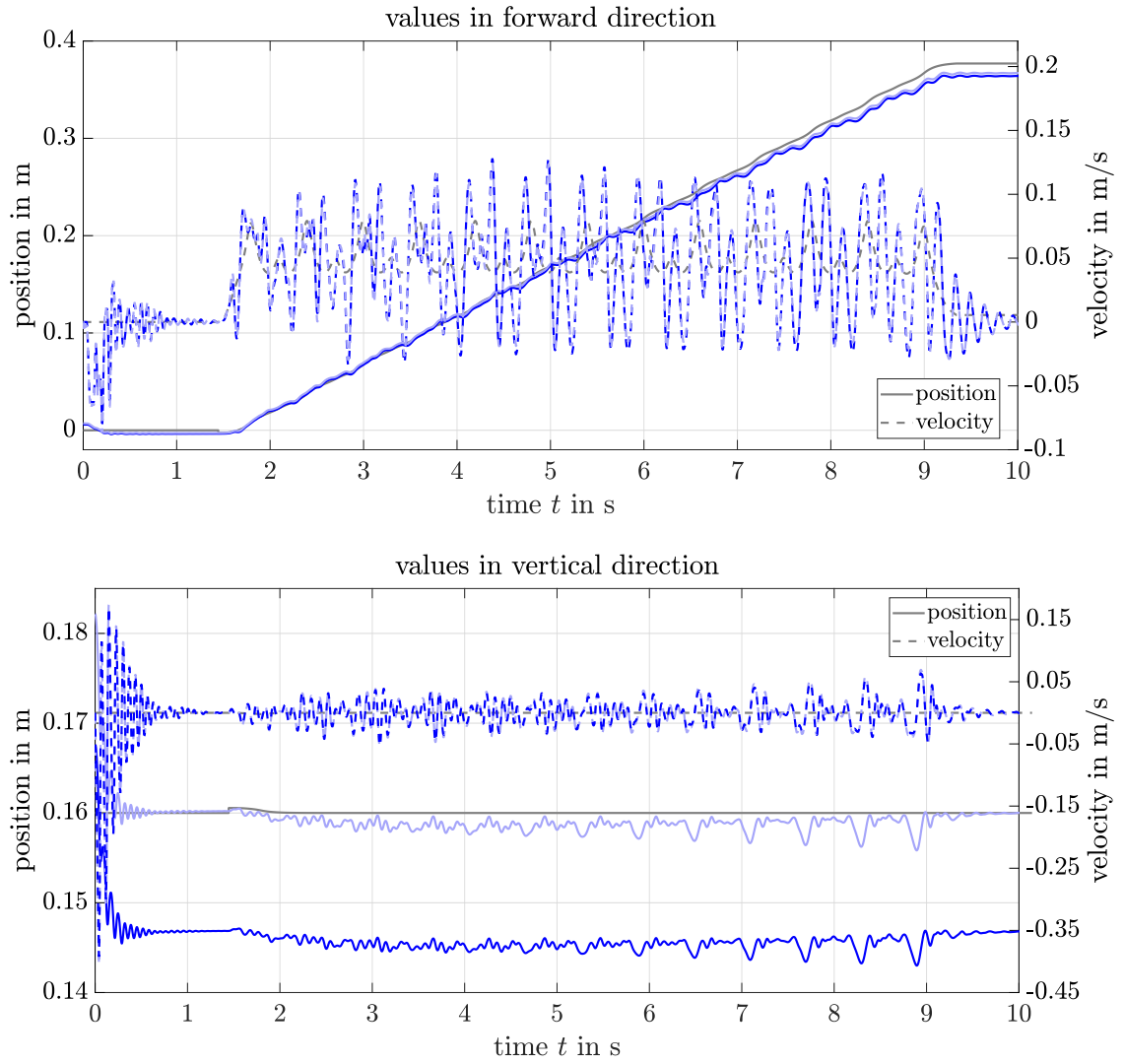


Figure 5.17: Record of CoM of the gait walk. The top plot shows the record of the CoM in the first direction (w_1 , forward). The solid lines represent the positions and the dashed lines represent the CoM velocity in the first direction. The bottom plot shows the record of the CoM in the third direction (w_3 , vertical). The solid lines represent the positions and the dashed lines represent the CoM velocity in the third direction. In both plots the blue dark lines represent the quantities of the CoM for the whole multi-body model (trunk mass + leg masses). The brighter blue lines represent the quantities of the CoM for the trunk mass and the gray lines indicate the applied CoM reference. On the right is the scale for the velocity. This figure corresponds to run 46 in table 5.1.

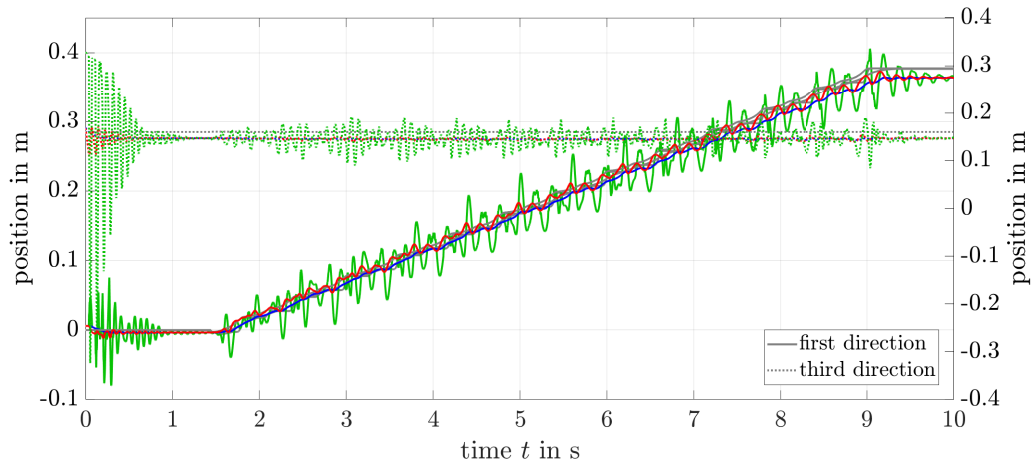


Figure 5.18: Plot of the quantities VRP (green lines), DCM (red lines) and CoM (blue lines) in the first and third direction. The solid lines represent the first direction and the dotted lines represent the third direction. On the right is the scale of the quantities in the third direction. The gray lines indicate the references. This figure corresponds to run 46 in table 5.1.

The limb and joint movement are shown in figures 5.19 and 5.20. The limb movement follows approximately the references. The plot of the limbs (s. figure 5.19) shows the insufficiently tracked feet movement in the third direction. Additionally, it shows a delay compared to the references and back slipping at the beginning of a swing phase. It is striking that there is an increasing shift between the recorded foot points and the desired references. This is more pronounced in the movement of the hind feet. As in the trotting gait, it can also be shown for the walk (s. figure 5.20) that the hip joint reference could be tracked accurately compared to the knee joints. In case of the walk, a large offset between the references and the record for the knee joints occurs.

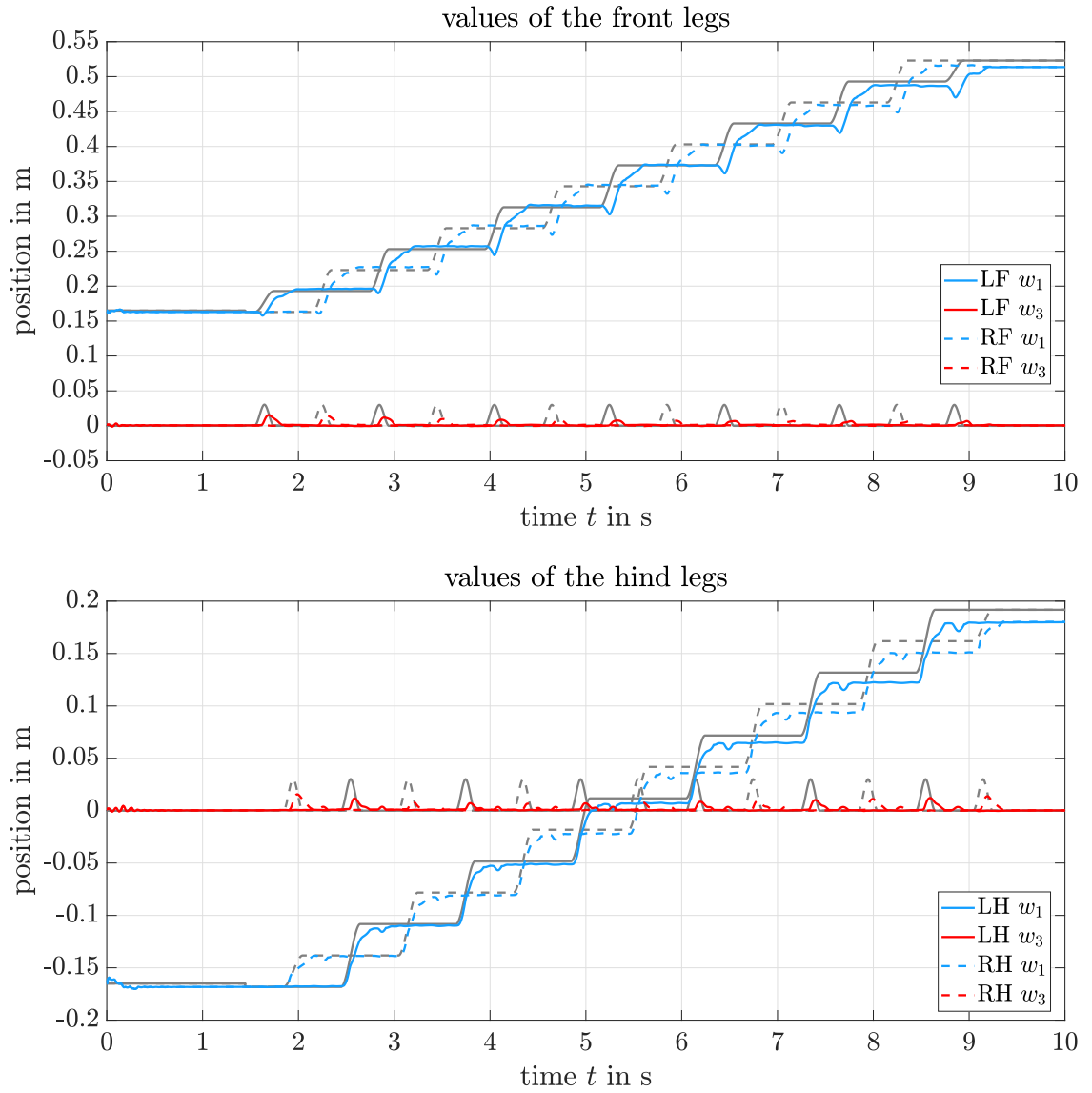


Figure 5.19: Records of the robot limbs in run 46 in the world frame. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the first direction and the red lines the third direction. The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references.

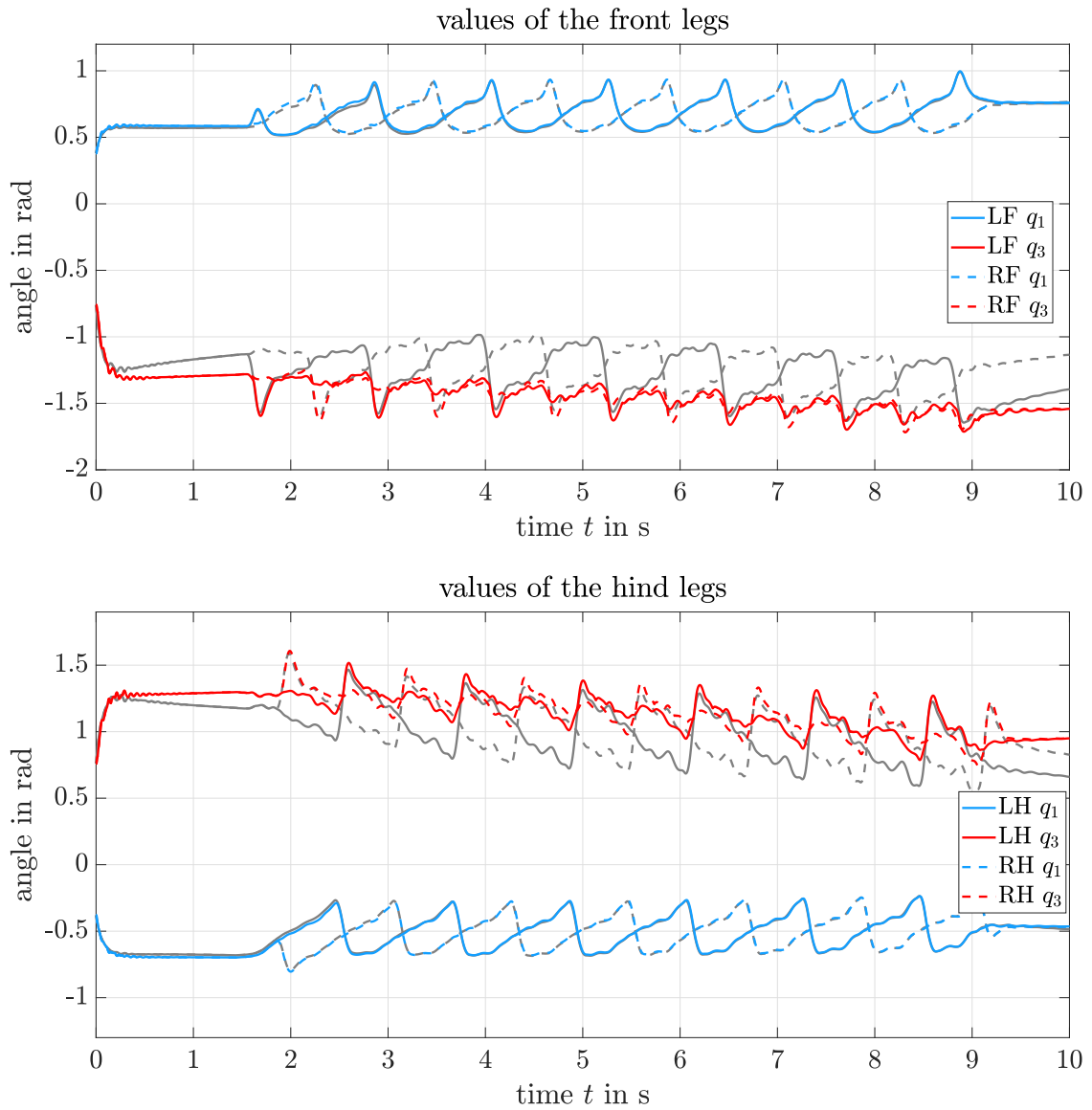


Figure 5.20: Records of the robot leg joints in run 46. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the hip joints (q_1) and the red lines the knee joints (q_3). The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references.

Figure 5.21 shows an example of a walk cycle for the walk gait (take off of $LF - RH - RF - LH$) of the simulation run 46. The robot started in an initial stance and with a double support phase (s. figure 5.11). The stills show the subsequent sequence of the alternating single and double support phases of the individual steps.

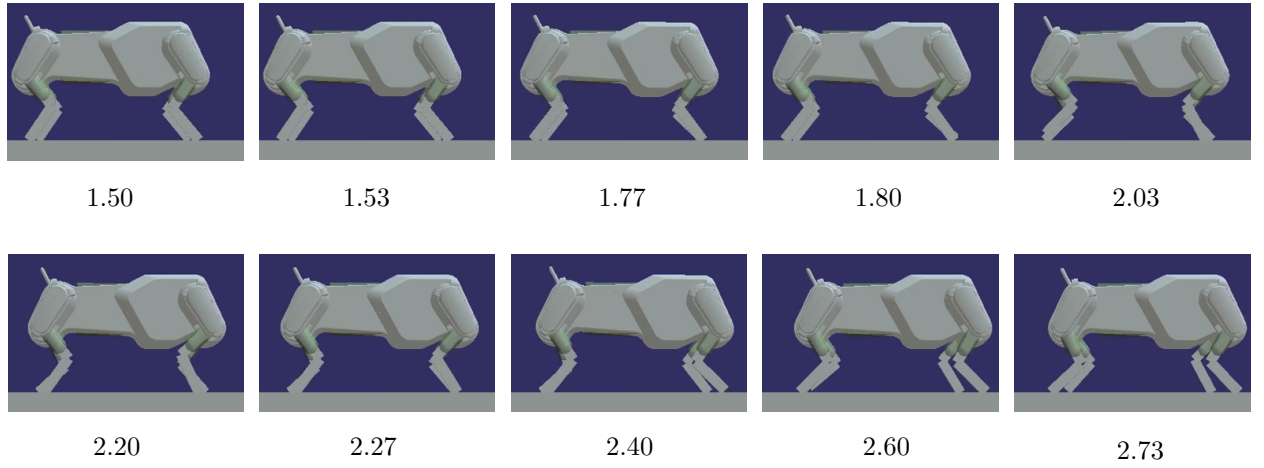


Figure 5.21: Sequence of stills of the simulation run 46 for the first walk cycle of the four legs (take of: $LF - RH - RF - LH$). The sequence begins in an initial stance and with a double support. After this the alternating sequence of the double support phases (all four legs in stance) and single support phases (three legs in stance) are showed till the touch down of LH and movement in the double support (last still). The bottom numbers indicate the time point of the still corresponding to the plots of run 46 in seconds.

In the appendix results of a walk simulation with a VRP placement are reported, which use the \mathbf{g}_σ for the VRP waypoints planning.

The simulation of the walk gait showed strong oscillations in the quantities (VRP, DCM and CoM) and shifting away from the references. Furthermore, the tracking of the desired joint references was not sufficient for the knee joints.

6 Discussion and Conclusion

6.1 Summary and Discussion

The goal of this thesis is to adapt the already successfully applied DCM method from bipedal locomotion to quadrupedal locomotion. The DCM method (s. fundamental chapter 2.5) allows to handle the bipedal locomotion problem of the generation of CoM trajectories in an analytical and efficient way. The solution for this locomotion problem are kinematically and dynamically feasible CoM and limb trajectories. In chapter 3.1 the basic approach of the DCM method is applied to the problem formulation of quadrupedal locomotion. For quadrupedal walking, just as for bipedal walking, a feasible resulting force on CoM for a given possible stance (support area) has to be found, in order to ensure a stable contact and a forward motion of the CoM. Or, equivalently, a feasible step area for a foot has to be found to support a desired CoM motion. Thereby the applied simplification of the multi-body floating base system is the LIP template model. In this model the line of action of the resulting force on the CoM has to point through the base of support.

For quadrupedal walking on mainly planar ground an appropriate point in the support polygon has to be chosen as a desired eCMP for the VRP waypoint planning. The support polygon is calculated as the convex hull of the projection of the contact feet on a plane perpendicular to the gravity vector, i.e. the support plane, in this thesis. With the assumption that each point in this support polygon is a projection of a suitable CoM rest position (gravity compensation and no additional perturbation and forces acting) on the support plane, the center of the support polygon can be chosen as a desired eCMP.

Thereby, a suitable eCMP (or CoP) for the motion planning would be a point in the base of support. However, in this thesis a point in the above mentioned support polygon is selected for VRP waypoint planning. The first two coordinates of the desired VRP waypoint are set equally to the first two coordinates of this selected point. A desired CoM trunk height Δz_{vrp} is added to the third coordinate of the chosen point. Such a point can directly be determined in the case of three legs in stance and is limited for the robot Bert to be in the sagittal trunk plane. The VRP placement has to be limited, because the robot can only control the CoM in this plane and the chosen VRP position determines the interpolation of the CoM trajectory in space. In the present planning, the motion planning is always able to select an appropriate point of the support polygon for the VRP way point planning.

However, for an arbitrarily uneven ground (i.e. not all contact points are co-planar) the assumptions made in this thesis (s. chapter 3.1) are not valid and the true support area of the quadrupedal has to be calculated (Bretl & Lall, 2008) for each planned stance in order to find a projection point for a CoM rest position. Implementing a solution to this problem goes beyond the scope of the work. Nevertheless an overview of possible solutions is provided below.

The authors Bretl and Lall (2008), for example, present an approach to test the membership

of desired CoM positions in the projection of the nonlinear convex support area. Another approach could be to approximate the support area (or polygon) for the planning, according to Lakatos et al. (2018) and select a point of this area as the planned eCMP. Furthermore, a pseudo-CoP on a hyperplane could be computed for the convex hull of the contact points in space and be selected for this support area, e.g. as in Sardain and Bessonnet (2004) reported. A further solution approach for the VRP/eCMP planning could be to constrain the desired eCMP to be in a potentially stable support polygon, as in the method applied by Winkler et al. (2015), or in the base of support (Winkler, Farshidian, Pardo, Neunert, & Buchli, 2017).

The gait definition (s. chapter 3.2) in this thesis is based on the stance description in Bouyarmane and Kheddar (2012) (Mesesan et al., 2017). According to the gait-specific touch down sequences, a stance sequence of the robot can be determined. For this purpose, the number of required stances for one step is derived from the fall pattern of the trotting, pacing and walk gait. The step sequence is designed to start in an initial stance and ends in a final stance with parallel front legs and parallel hind legs. The spatial shift distances in the forward direction are determined by the design variable Δx_{step} . The stance according to double support or single support is mapped to desired duration times of the trajectory segments. In this thesis, the time duration for the fixed and transition segments are selected globally with fixed values. Further, a globally desired trunk CoM height Δz_{vrp} is applied for the motion planning. The definitions of the DCM method (s. chapter 2.5) allow a placement of desired eCMP at different ground levels (Englsberger et al., 2015). However, in this thesis walking on a flat ground is assumed. The developed gait description and application of the DCM-method in this thesis result in a very compact gait description that requires only a few parameters (T_{DS} , T_{SS} , Δz_{vrp} , p , Δx_{step} , Δh_{step}) for the motion planning. The motion planning can provide the desired CoM trajectories in a compact and analytical way for further planning and control. In conclusion, this gait description can be used to specify any trot, pace or walk pattern, with at least one limb in contact with the ground in each stance. It is also conceivable to describe a gait pattern that includes a transition between these gait types using the present planning method. The determined static walk gait in this thesis is no dynamical gait (s. chapter 3.1), but can be applied as a base for a dynamical walk description. For this purpose, an additional time parameter could be used to determine the time point before the next foot point touches down at which the next foot point has to lift off. The realization of the dynamical walk would provide a more stable dynamical gait compared to the trotting gait, because in dynamic walking there is a period of time when three feet are in contact with the ground (Lakatos et al., 2018).

In order to test the present application of the DCM method to quadrupedal locomotion, an exiting simulation framework was extended for a multi-body model of the robot Bert. This DCM-based framework was applied for a simulation to investigate the kinematic and dynamic possibilities on the robot Bert for the first time, as far as the author is aware. The framework allows the robot to perform a desired number of steps forward (or backward). Due to the kinematic restrictions of the robot (s. chapter 2.6), the simulation used a floating-base with two DOF in the robot's sagittal plane and a possible pitch rotation of the trunk link. The selected simulation runs (s. chapter 5) show a possible application of the DCM method to quadrupedal robots and the generated CoM trajectories. Thereby, the results in chapter 5 show that the CoM is able to follow the DCM and oscillates around the references. This leads to a CoM oscillating close to the reference. The generated limb and CoM trajectories

(for trotting and static walk) are tracked by the position-controlled joints, where desired joint positions are provided by a resolved motion approach (s. chapters 4.2 and 4.3). The applied control method for the simulation was not able to follow exactly the reference (s. figures 5.6 and 5.20). This could be due to an insufficient consideration of the elasticities of the SEA in the control approach. Additionally, only the transformation of the rigid torque to a desired motor position (s. chapter 4.3) used in the simulation, could be insufficient to close the discrepancy between motor position and link position in stance for flexible joint. This issue is noted in Seidel et al. (2020) for the robot Bert and has to be considered in following experiments.

Furthermore, the un-constraint contact force, i.e. the forces are considered to be suitable in this thesis (s. chapter 3.1), could be the reason for the oscillations (s. VRP and DCM plots in chapter 5.2). For this reason, the implicit assumption of sufficient and well-designed contact forces during the leg stand could be an oversimplification. In addition, the applied manual control parameter tuning for gain matrices showed a trend to instability for larger values than those in table 4.1 in combination with the joint reference generation method. Additionally, too large values for the gain matrix \mathbf{K}_1 result in instabilities.

The performed simulations show that for $T_{DS} \geq 0.5s$ the magnitude of the oscillation are reduced.

In the performed simulation runs, the motion planning and motion control always provided references, which were within the kinematic ranges (s. chapter 2.6) and lower than the maximum absolute joint velocity (10 rad/s (Lakatos et al., 2018)) of the robot (s. appendix A.3.2).

The achieved average forward velocities of the trotting gait were lower in the simulation than the reported results of the robot Bert according to Lakatos et al. (2018). The foot point tracking in the simulation plots shows that the robot was able to perform the desired movement. During the swing phases of the legs the movement has a delay compared to the references. The reason for this is a too late release of the feet from the ground and a failure to adjust to the absolute reference.

A further restriction is the limitation of CoM movement in the sagittal robot plane. This reduces the theoretical possibilities of the CoM planning compared to a quadrupedal with 12 actuated DOF. Therefore, the VRP (or eCMP) placement is limited to be on the direct forward line of the trunk. A 12 DOF quadrupedal with a possible point selection or constraint on the full base of support would allow further investigations of a DCM-based gait design. For this purpose, a step planning parameter could be applied for the foot point shift distance in the horizontal robot trunk plane. In addition, the globally selected duration times of the trajectory limit the investigation of impacts of these design parameters on motion behavior. For this reason, the motion planning with varying time duration of the single and double support phases has to be further investigated. Mesesan et al. (2017) present an approach for an automatic multi-contact transition planer for the DCM method to humanoid (bipedal) robots. This approach could be adapt for DCM-based planning method for quadrupedal robots. An investigation of such an approach for DCM-based quadrupedal locomotion could provide insight into the use of varying duration times on challenging ground (uneven or compliant ground).

The main advantage of the DCM method, namely to control the unstable DCM instead of the naturally stable CoM dynamic, was not applied in this thesis. The further design and application of the reported DCM-tracking control for bipedal robots according to

Englsberger et al. (2017) and Englsberger et al. (2015) to quadrupedal robots could improve the reference tracking of the present DCM-based quadrupedal motion planning in the future. This is associated with the further application of a step adjustment method for uneven terrain, which is already reported for bipedal locomotion (Englsberger et al., 2017, 2015). The existing control framework of the robot Bert (s. chapter 4.1) was extended with a DCM-based trajectory generation for subsequent experiments. However, due to the Corona pandemic, final tests and the performance of the planned experiments were not possible. The application of a whole-body torque control approach could offer an alternative to the control approach used in this thesis. Such a control approach, e.g. the applied control framework in Mesesan et al. (2019) for a bipedal, would allow to optimize the base accelerations, joint accelerations and the contact forces with subject to the kinematic and dynamic constraints of the robot. For a future application of such a control approach the realization on the robot Bert has to be examined. A further control alternative for a trajectory control could be a feedback linearization approach for robots with flexible joints according to de Luca and Book (2016). Here, too, the possibility of implementation on the robot Bert has to be considered and tested.

6.2 Conclusion

This thesis presents the first investigation - as far as the author knows - of an application of the DCM method from bipedal to quadrupedal locomotion. For this purpose, the common gaits trotting, pacing and static walk are described as an example for the step planning in the proposed stance description. As a result, the simulation performed in this thesis shows the possibility of an application of the DCM-based trajectory generation to quadrupedal robots. The implemented DCM-based motion planning can set a foundation for further investigations and extensions of the DCM application to quadrupedal robots.

A possible further extension could be a DCM-tracking control and whole-body-torque-control in combination with a fast and efficient support area-based VRP planning. Another possible extension could be a step adaption to generate suitable CoM trajectories also for challenging terrain under perturbations. An investigation of impacts of the SEAs on the DCM-based gait parameters could provide more insights to this kind of motion planning. Moreover, the fundamentally implementation of the dynamical walk gait could be tested and extended. Finally, an extension of the robot Bert to 12 actuated DOF (3 DOF for each leg) would improve the possibilities of the DCM-based motion planning and future extensions of the present approach.

However, these results still need to be validated by experiments on the robot Bert.

References

- Alexander, R. M. (1990). Three uses for springs in legged locomotion. *The International Journal of Robotics Research*, 9(2), 53–61. doi: 10.1177/027836499000900205
- Bellicoso, C. D., Jenelten, F., Fankhauser, P., Gehring, C., Hwangbo, J., & Hutter, M. (2017). Dynamic locomotion and whole-body control for quadrupedal robots. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3359–3365). IEEE. doi: 10.1109/IROS.2017.8206174
- Bellicoso, C. D., Jenelten, F., Gehring, C., & Hutter, M. (2018). Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. *IEEE Robotics and Automation Letters*, 3(3), 2261–2268. doi: 10.1109/LRA.2018.2794620
- Bledt, G., Powell, M. J., Katz, B., Di Carlo, J., Wensing, P. M., & Kim, S. (2018). Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2245–2252). IEEE. doi: 10.1109/IROS.2018.8593885
- Blickhan, R. (1989). The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11-12), 1217–1227. doi: 10.1016/0021-9290(89)90224-8
- Bouyarmane, K., & Kheddar, A. (2012). Humanoid robot locomotion and manipulation step planning. *Advanced Robotics*, 26(10), 1099–1126. doi: 10.1080/01691864.2012.686345
- Bretl, T., & Lall, S. (2008). Testing static equilibrium for legged robots. *IEEE Transactions on Robotics*, 24(4), 794–807. doi: 10.1109/TRO.2008.2001360
- de Luca, A., & Book, W. J. (2016). Robots with flexible elements. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 243–282). Berlin and Heidelberg: Springer. doi: 10.1007/978-3-319-32552-1{\textunderscore}11
- Di Carlo, J., Wensing, P. M., Katz, B., Bledt, G., & Kim, S. (2018). Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1–9). IEEE. doi: 10.1109/IROS.2018.8594448
- Englsberger, J. (2016). *Combining reduced dynamics models and whole-body control for agile humanoid locomotion* (Dissertation, Technische Universität München, München). Retrieved from <http://nbn-resolving.de/urn:nbn:de:bvb:91-diss-20161129-1311519-1-2>
- Englsberger, J., Mesesan, G., & Ott, C. (2017). Smooth trajectory generation and push-recovery based on divergent component of motion. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4560–4567). IEEE. doi: 10.1109/IROS.2017.8206324
- Englsberger, J., Ott, C., & Albu-Schaffer, A. (2013). Three-dimensional bipedal walking control using divergent component of motion. In *2013 IEEE international conference on robotics and automation* (pp. 2600–2607). IEEE. doi: 10.1109/IROS.2013.6696723
- Englsberger, J., Ott, C., & Albu-Schaffer, A. (2015). Three-dimensional bipedal walking control based on divergent component of motion. *IEEE Transactions on Robotics*,

- 31(2), 355–368. doi: 10.1109/TRO.2015.2405592
- Gehring, C., Coros, S., Hutter, M., Bloesch, M., Hoepflinger, M. A., & Siegwart, R. (2013). Control of dynamic gaits for a quadrupedal robot. In *2013 IEEE International Conference on Robotics and Automation* (pp. 3287–3292). IEEE. doi: 10.1109/ICRA.2013.6631035
- Gehring, C., Hoepflinger, M., Siegwart, R., Coros, S., Hutter, M., Bellicoso, C. D., ... Hwangbo, J. (2016). Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics & Automation Magazine*, 23(1), 34–43. doi: 10.1109/MRA.2015.2505910
- Geyer, H., Seyfarth, A., & Blickhan, R. (2006). Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings. Biological sciences*, 273(1603), 2861–2867. doi: 10.1098/rspb.2006.3637
- Goswami, A. (1999). Postural stability of biped robots and the foot-rotation indicator (fri) point. *The International Journal of Robotics Research*, 18(6), 523–533. doi: 10.1177/02783649922066376
- Hardarson, F. (2002). *Stability analysis and synthesis of statically balanced walking for quadruped robots* (PhD thesis). KTH.
- Holmes, P., Full, R. J., Koditschek, D., & Guckenheimer, J. (2006). The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review*, 48(2), 207–304. doi: 10.1137/S0036144504445133
- Hutter, M., Gehring, C., Bloesch, M., HOEPFLINGER, M. A., REMY, C. D., & Siegwart, R. (2012). Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion. In A. K. M. Azad, N. J. Cowan, M. O. Tokhi, G. S. Virk, & R. D. Eastman (Eds.), *Adaptive mobile robotics* (pp. 483–490). WORLD SCIENTIFIC. doi: 10.1142/9789814415958{\textunderscore}0062
- Hutter, M., Gehring, C., Hopflinger, M. A., Bloesch, M., & Siegwart, R. (2014). Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped. *IEEE Transactions on Robotics*, 30(6), 1427–1440. doi: 10.1109/TRO.2014.2360493
- Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., ... Hoepflinger, M. (2016). Anymal - a highly mobile and dynamic quadrupedal robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 38–44). IEEE. doi: 10.1109/IROS.2016.7758092
- Kajita, S. (2019). Linear inverted pendulum-based gait. In A. Goswami & P. Vadakkepat (Eds.), *Humanoid robotics: A reference* (pp. 905–922). Dordrecht: Springer Netherlands. doi: 10.1007/978-94-007-6046-2{\textunderscore}42
- Kajita, S., & Ott, C. (2016). Limbed systems. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 419–442). Berlin and Heidelberg: Springer. doi: 10.1007/978-3-319-32552-1{\textunderscore}17
- Lakatos, D., Federigi, Y., Gumpert, T., Henze, B., Hermann, M., Loeffl, F., ... Albu-Schaffer, A. (2019). A coordinate-based approach for static balancing and walking control of compliantly actuated legged robots. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 9509–9515). IEEE. doi: 10.1109/ICRA.2019.8793920
- Lakatos, D., Ploeger, K., Loeffl, F., Seidel, D., Schmidt, F., Gumpert, T., ... Albu-Schaffer, A. (2018). Dynamic locomotion gaits of a compliantly actuated quadruped with slip-like articulated legs embodied in the mechanical design. *IEEE Robotics and Automation Letters*, 3(4), 3908–3915. doi: 10.1109/LRA.2018.2857511
- Liu, M., Qu, D., Xu, F., Zou, F., Di, P., & Tang, C. (2019). Quadrupedal robots whole-

- body motion control based on centroidal momentum dynamics. *Applied Sciences*, 9(7), 1335. doi: 10.3390/app9071335
- McGhee, R. B., & Frank, A. A. (1968). On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3, 331–351. doi: 10.1016/0025-5564(68)90090-4
- Mesanan, G., Engelsberger, J., Garofalo, G., Ott, C., & Albu-Schaffer, A. (2019). Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)* (pp. 25–32). IEEE. doi: 10.1109/Humanoids43949.2019.9035053
- Mesanan, G., Engelsberger, J., Henze, B., & Ott, C. (2017). Dynamic multi-contact transitions for humanoid robots using divergent component of motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4108–4115). IEEE. doi: 10.1109/icra.2017.7989473
- Mesanan, G., Engelsberger, J., Ott, C., & Albu-Schaffer, A. (2018). Convex properties of center-of-mass trajectories for locomotion based on divergent component of motion. *IEEE Robotics and Automation Letters*, 3(4), 3449–3456. doi: 10.1109/LRA.2018.2853557
- Murray, R. M., Li, Z., & Sastry, S. (1994). *A mathematical introduction to robotic manipulation*. Boca Raton, Fla.: CRC Press. Retrieved from <http://www.loc.gov/catdir/enhancements/fy0730/93033167-d.html>
- Pratt, G. A., & Williamson, M. M. (1995). Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference 1995* (pp. 399–406). doi: 10.1109/IROS.1995.525827
- Pratt, J. E., Koolen, T., de Boer, T., Rebula, J., Cotton, S., Carff, J., ... Neuhaus, P. (2012). Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *The International Journal of Robotics Research*, 31(10), 1117–1133. doi: 10.1177/0278364912452762
- Pratt, J. E., Krupp, B., & Morse, C. (2002). Series elastic actuators for high fidelity force control. *Industrial Robot: An International Journal*, 29(3), 234–241. doi: 10.1108/01439910210425522
- Pratt, J. E., & Tedrake, R. (2006). Velocity-based stability margins for fast bipedal walking. In M. Diehl & K. Mombaur (Eds.), *Fast motions in biomechanics and robotics* (Vol. 340, pp. 299–324). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-36119-0{\textunderscore}14
- Raibert, M. (1990). Trotting, pacing and bounding by a quadruped robot. *Journal of Biomechanics*, 23, 79–98. doi: 10.1016/0021-9290(90)90043-3
- Raibert, M., Blankespoor, K., Nelson, G., & Playter, R. (2008). Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2), 10822–10825. doi: 10.3182/20080706-5-KR-1001.01833
- Raibert, M., Chepponis, M., & Brown, H. (1986). Running on four legs as though they were one. *IEEE Journal on Robotics and Automation*, 2(2), 70–82. doi: 10.1109/JRA.1986.1087044
- Ratliff, N. (2014). *Controlling floating-based robots*. Retrieved 02.11.2020, from <http://www.nathanratliff.com/pedagogy/advanced-robotics>
- Sardain, P., & Bessonnet, G. (2004). Forces acting on a biped robot. center of pressure—zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(5), 630–637. doi: 10.1109/TSMCA.2004.832811
- Seidel, D., Hermann, M., Gumpert, T., Loeffl, F. C., & Albu-Schaffer, A. (2020). Using

- elastically actuated legged robots in rough terrain: Experiments with dlr quadruped bert. In *2020 ieee aerospace conference* (pp. 1–8). IEEE. doi: 10.1109/AERO47225.2020.9172301
- Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3), 201–212. doi: 10.1007/BF00126069
- Siciliano, B., Sciavicco, L., Chiaverini, S., Chiacchio, P., Villani, L., & Caccavale, F. (2003). Jacobian-based algorithms: A bridge between kinematics and control. In *Proceedings of the special celebratory symposium in the honor of professor bernie roth's 70th birthday* (pp. 4–35). Citeseer.
- Spröwitz, A., Tuleu, A., Vespignani, M., Ajallooeian, M., Badri, E., & Ijspeert, A. J. (2013). Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot. *The International Journal of Robotics Research*, 32(8), 932–950. doi: 10.1177/0278364913489205
- Sucan, I., & Kay, J. (2019). *urdf - ros wiki*. Retrieved 07.01.2021, from <http://wiki.ros.org/urdf>
- Sugihara, T., Nakamura, Y., & Inoue, H. (2002). 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)* (pp. 1404–1409). IEEE. doi: 10.1109/ROBOT.2002.1014740
- Takenaka, T., Matsumoto, T., & Yoshiike, T. (2009). Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-. In *2009 ieee/rsj international conference on intelligent robots and systems* (pp. 1084–1091). IEEE. doi: 10.1109/IROS.2009.5354662
- Tedrake, R. (2020). *Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832)*. Retrieved 02.11.2020, from <http://underactuated.mit.edu/>
- Vukobratović, M., & Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15(1-2), 1–37. doi: 10.1016/0025-5564(72)90061-2
- Vukobratović, M. and Borovac, B. (2004). Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01), 157–173.
- Wieber, P.-B., Tedrake, R., & Kuindersma, S. (2016). Modeling and control of legged robots. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 1203–1234). Berlin and Heidelberg: Springer. doi: 10.1007/978-3-319-32552-1{\textunderscore}48
- Winkler, A. W., Farshidian, F., Pardo, D., Neunert, M., & Buchli, J. (2017). Fast trajectory optimization for legged robots using vertex-based zmp constraints. *IEEE Robotics and Automation Letters*, 2(4), 2201–2208. doi: 10.1109/LRA.2017.2723931
- Winkler, A. W., Mastalli, C., Havoutis, I., Focchi, M., Caldwell, D. G., & Semini, C. (2015). Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In *2015 ieee international conference on robotics and automation (icra)* (pp. 5148–5154). IEEE. doi: 10.1109/ICRA.2015.7139916
- Zhao, Y., & Sentis, L. (2012). A three dimensional foot placement planner for locomotion in very rough terrains. In *2012 12th ieee-ras international conference 2012* (pp. 726–733). doi: 10.1109/HUMANOIDS.2012.6651600

A Appendix

A.1 Robot Leg

Figure A.1 shows a sketch of the planar robot leg with the DOF of the motors for the SEAs.

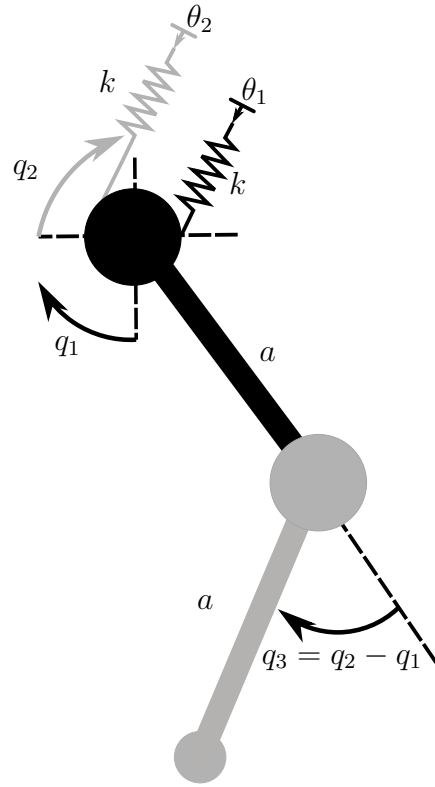


Figure A.1: Schematic sketch of one planar leg of the robot Bert. The joints q_1 and q_2 act as pulley and result in the knee joint q_3 . The hip joint is affected by the joint q_1 and the motor position θ_1 (black parts). The gray parts are affected by both SEAs. The linear torsion springs have the stiffness k . (Lakatos et al., 2018)

The two joint DOF of a leg are coupled by equation (A.1), where q_3 is the resulting knee joint DOF and only q_1 and q_2 are actuated by the SEAs (Lakatos et al., 2018), where q_1 is the hip joint DOF.

$$q_3 = q_2 - q_1 \quad (\text{A.1})$$

From equation (A.1) follows directly the joint torque τ_3 for one leg in equation (A.2).

$$\tau_3 = \tau_2 - \tau_1 = k(\theta_2 - q_2) - k(\theta_1 - q_1) = k((\theta_2 - \theta_1) - (q_2 - q_1)) \quad (\text{A.2})$$

A.2 VRP placement

In this section the VRP placement procedure is reported.

Procedure greyBert VRP planning correspond to the function *vrpPlacementHyperplaneGeometric_XX()* in the folder *MultiContactPlaner* (s. appendix A.4). The function *vrpPlacementHyperplaneGeometric_XX()_contiFor* in the same folder calculates the VRP waypoints continues forward, i.e. the next VRP waypoints is always equal or fore the last VRP in the forward direction. The difference between these functions are that the latter violates the static stability margin, which only results in a different behavior in case of the gait static walk. The function *vrpPlacementHyperplaneGeometric_ideal()* applies only \mathbf{g}_σ for the VRP planning. The function *vrpPlacementHyperplaneGeometric_ideal()* applies only \mathbf{g}_σ for the VRP planning.

Procedure greyBert VRP planning

input: Σ , \mathbf{e}_{trunk} (unit vector of the robot trunk)**output:** $V = [\mathbf{v}_1 \dots \mathbf{v}_n]^T$ **foreach** σ of Σ **do**| $\mathbf{v}_i \leftarrow [\mathbf{g}_\sigma, \Delta z_{vrp}]$;**end** $\mathbf{v}_1 \leftarrow \mathbf{v}_1$;**for** $i = 2$ **to** $n - 1$ **do**| *determine the number of contact n_c in σ_i ;*| *determine the direction vector of $\mathbf{d} \leftarrow \mathbf{v}_i - \mathbf{v}_{i-1}$;*| $d_{sign} \leftarrow$ *determine the sign of the projection of \mathbf{d} in \mathbf{e}_{trunk} ;*| **if** $n_c == 2$ **then**| | **if** \mathbf{v}_i on the forward line: $\mathbf{v}_{i-1} + d_{sign}5\mathbf{e}_{trunk} \wedge d_{sign} > 0$ **then**| | | $\mathbf{v}_i \leftarrow \mathbf{v}_i$ | | **end**| | **else**| | | *project \mathbf{v}_{i-1} on the line pointing from \mathbf{v}_{i-1} forward until an intersection with support line*| | **end**| **end**| **if** $n_c > 2$ **then**| | **if** \mathbf{v}_i on the forward line: $\mathbf{v}_{i-1} + d_{sign}5\mathbf{e}_{trunk} \wedge d_{sign} > 0$ **then**| | | $\mathbf{v}_i \leftarrow \mathbf{v}_i$;| | **end**| | **else**| | | **if** \mathbf{v}_i not on the forward line: $\mathbf{v}_{i-1} + d_{sign}5\mathbf{e}_{trunk} \wedge d_{sign} > 0$ **then**| | | | *project \mathbf{v}_i on the forward line ;*| | | **end**| | | **if** \mathbf{v}_i not on the forward line: $\mathbf{v}_{i-1} + d_{sign}5\mathbf{e}_{trunk} \wedge d_{sign} < 0$ **then**| | | | *project \mathbf{v}_i on the forward line ;*| | | | *shift \mathbf{v}_i until minium distance to a support polygon edge is violated;*| | | **end**| | **end**| | **if** \mathbf{v}_i on the forward line: $\mathbf{v}_{i-1} + d_{sign}5\mathbf{e}_{trunk} \wedge d_{sign} < 0$ **then**| | | *shift \mathbf{v}_i forward until the minium distance to a support polygon edge is violated ;*| | **end**| **end**| **else**| | $\mathbf{v}_i \leftarrow \mathbf{v}_i$ | **end****end** $\mathbf{v}_n \leftarrow \mathbf{v}_{i+1}$;

A.3 Simulation

This section contains extended information about the used simulation framework and simulations results.

The simulation runs presented in chapter 5 were selected from the runs performed. All other runs can be found in the remote project folder (s. appendix A.4).

A.3.1 Further Information of the simulation Bert model

The already implemented motor model in the simulation requires a desired motor torque which is the reference for τ and integrates twice equation (A.3) in order to provide θ .

$$\ddot{\theta} = B^{-1}(\tau_{motor} - \tau) \quad (A.3)$$

The already implemented spring model computes the torque joint by

$$\tau = K_s(\theta - q) + D_s(\dot{\theta} - \dot{q}) \quad (A.4)$$

where $D_s \in \mathbb{R}^{N \times N}$ is the square and diagonal damping gain matrix with value of 1.1499 Nm·s/rad for the hip SEAs and a value of 0.6123 Nm·s/rad for the knee SEAs. These values were selected for a critical damping to solve a simulation issue. This problem was caused by the contact model of the framework and could have occurred due to the small total mass of the robot. This caused the robot to be pushed in the positive vertical direction (bouncing), resulting in forward and backward slippage in case of no damping in the spring model.

Note that in the simulation the model applies q_3 (s. chapter A.1) for the calculation. The real robot applies a position control for the joint angles q_1 and q_2 . Therefore q_2 can be computed with (A.1) in the experiments and with respect to (A.2).

The entries of B are selected to 0.0001 kg·m², because the applied Bert motors are assumed to have a small moment of inertia.

Furthermore, an already implemented friction model at joint torque level (Coulomb friction by an offset and viscous friction by a linear gain) can be activated in the Simulink model. The values are selected to 0.0 for the coulomb friction and to -0.02 for the viscous friction.

For the entries of the weighting matrix W in chapter 3.4 a value of 100 is selected and the entries for the gain values of the first and second task coordinate of K_1 are selected to 8 and for third coordinate to 12.

A.3.2 Joint Reference

Figure A.2 shows a plot of the joint velocity in a simulation run for the method introduced in chapter 3.4. The plot in figure A.2 shows the corresponding joint reference to figure A.2.

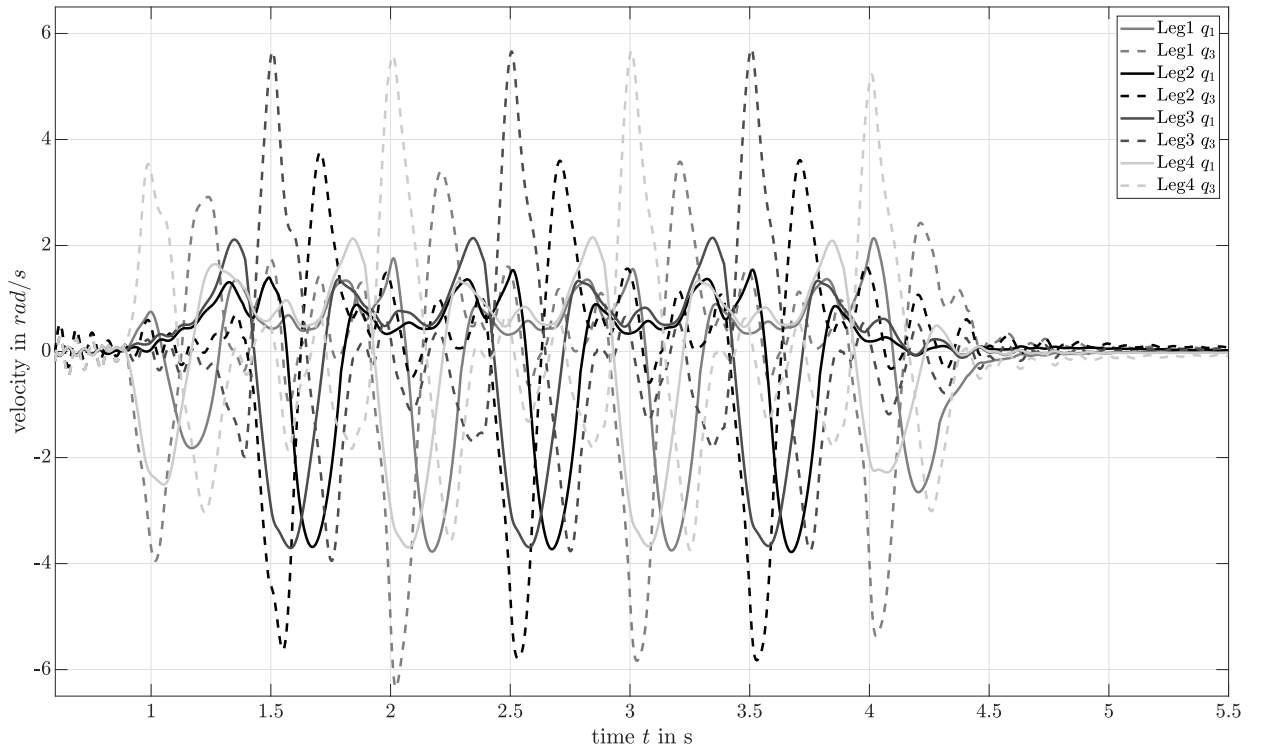


Figure A.2: Desired joint velocity resulting by the implemented method for an example parameter set in chapter 3.4. The joint velocities are not violated the absolute maximum of 10 rad/s.

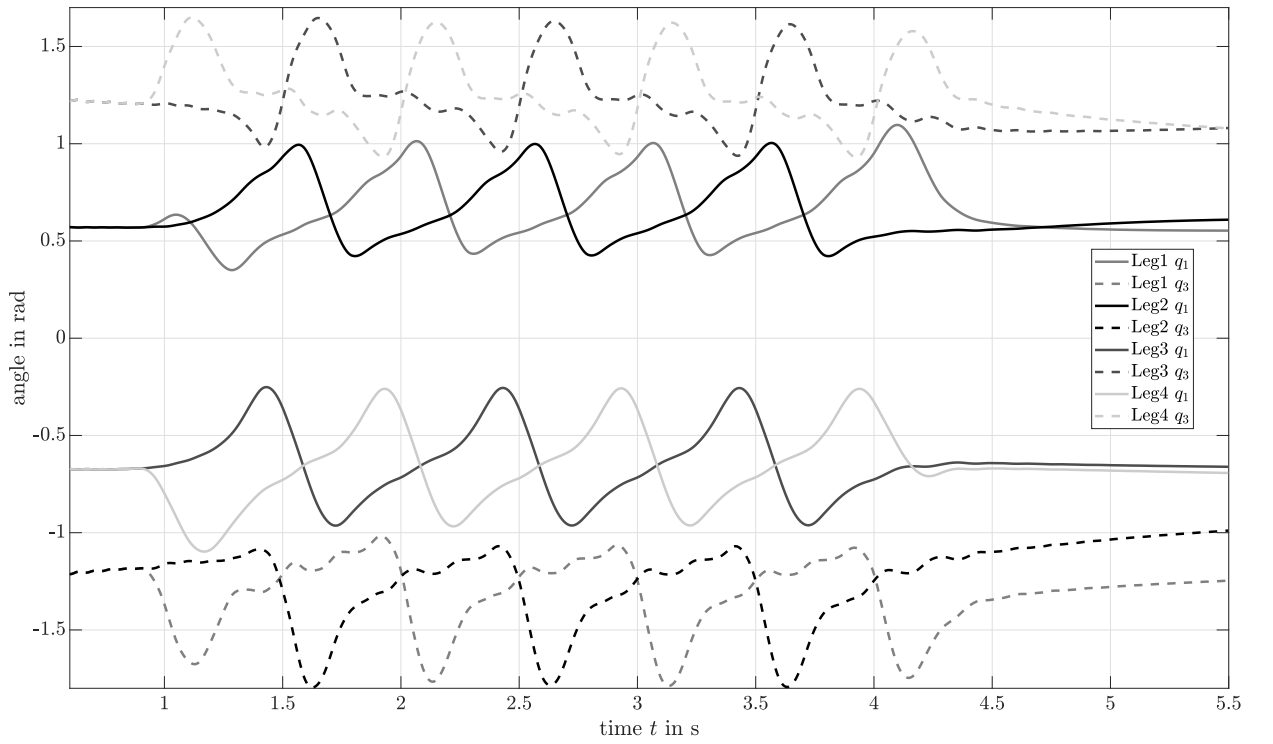


Figure A.3: Desired joint angles resulting by the implemented method for an example parameter set in chapter 3.4. The joint limits in chapter 2.6 are not violated.

A.3.3 Further results

Figure A.4 shows the plotted results of the VRP signal of run 46 in table 5.1 for a ideal VRP placement, i.e. with the \mathbf{g}_σ as planned eCMP.

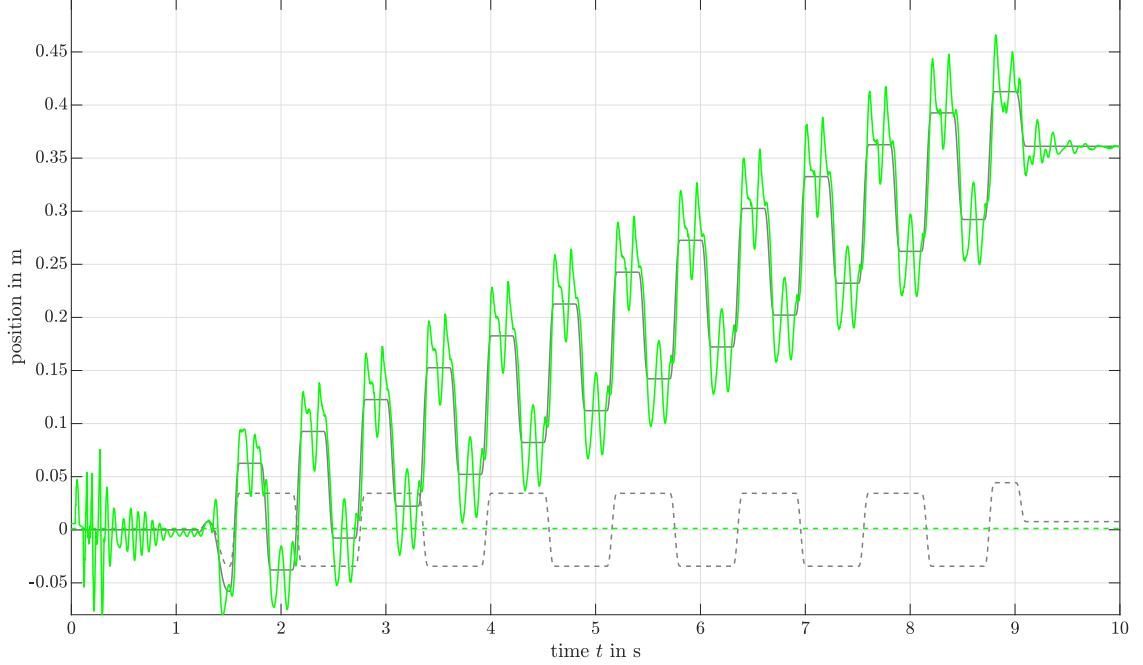


Figure A.4: Record of the VRP of the gait walk in the first and second direction of the world frame with an ideal VRP placement. The green solid line represents the calculated real VRP in the first direction (forward) and the dashed line shows the VRP in the second direction (horizontal). This plot corresponds to run 46 in table 5.1. The gray lines indicate the VRP reference and the applied waypoints were determined by \mathbf{g}_σ .

Furthermore, figure A.5 shows the recorded DCM of this run and figure A.6 shows the records of the true CoM and the CoM of the trunk link.

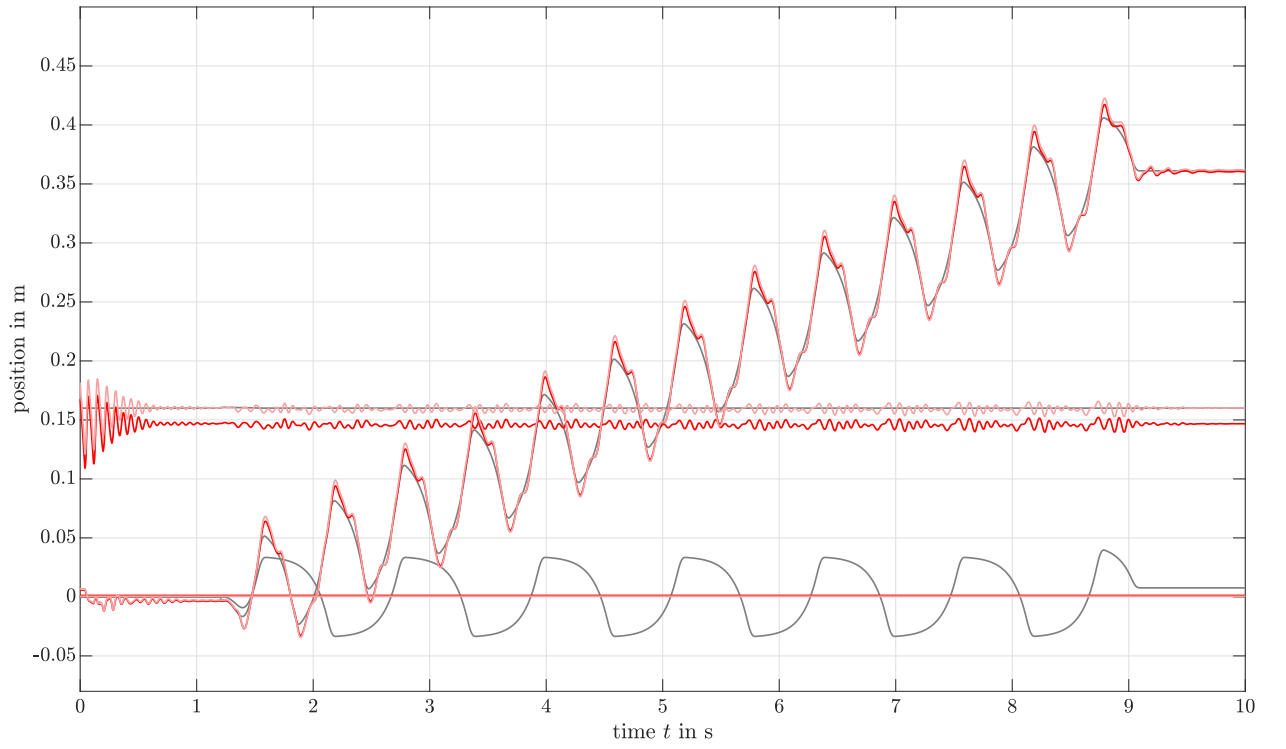


Figure A.5: Record of DCM of the gait walk with ideal VRP placement in the world frame. The red solid line represents the DCM in the first direction (forward). The dashed line shows the DCM in the second direction (horizontal) and the third direction (vertical). The dark red lines correspond to a DCM calculation based on the whole-body CoM. The brighter red lines indicate the DCM calculated with the trunk CoM. The gray lines represent the applied DCM reference. This plot corresponds to run 46 in table 5.1, where the VRP waypoints are \mathbf{g}_σ (s. figure A.4).

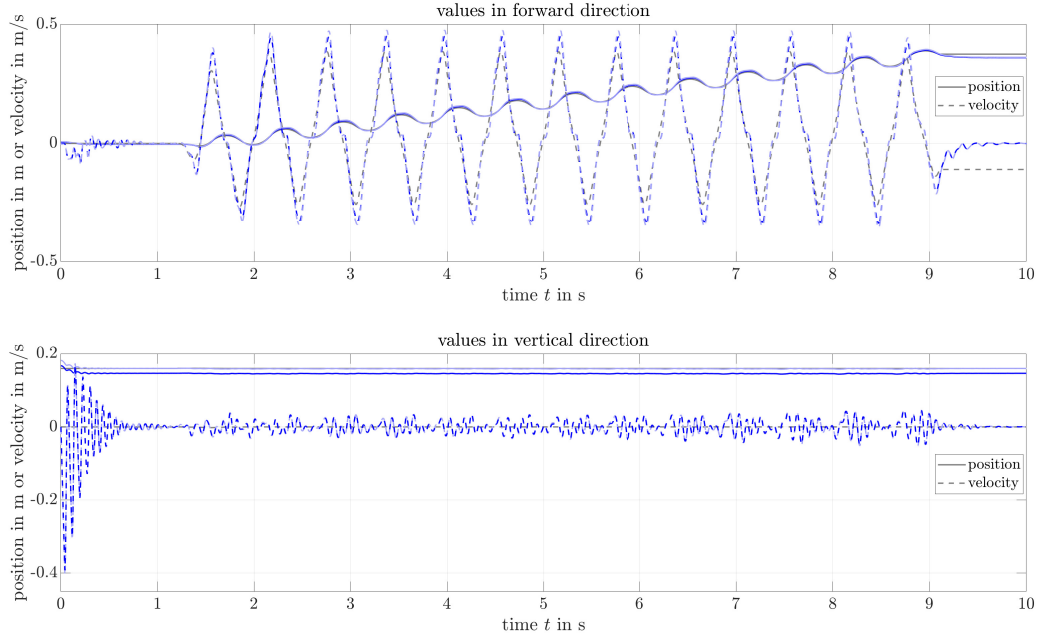


Figure A.6: Record of CoM of the gait walk in the world frame. The top plot shows the record of the CoM in the first direction (forward). The solid lines represent the positions and the dashed lines represent the CoM velocity in the first direction. The bottom plot shows the record of the CoM in the third direction (vertical). The solid lines represent the positions and the dashed lines represent the CoM velocity in the third direction. In both plots the blue dark lines represent the quantities of the CoM for the whole multi-body model (trunk mass + leg masses). The brighter blue lines represent the quantities of the CoM for the trunk mass and the gray lines indicate the applied CoM reference. The right scale is for the scale for the velocity. This plot corresponds to run 46 in table 5.1, where the VRP waypoints are \mathbf{g}_σ (s. figure A.4).

The limb and joint movement are shown in the figures A.7 and A.8.

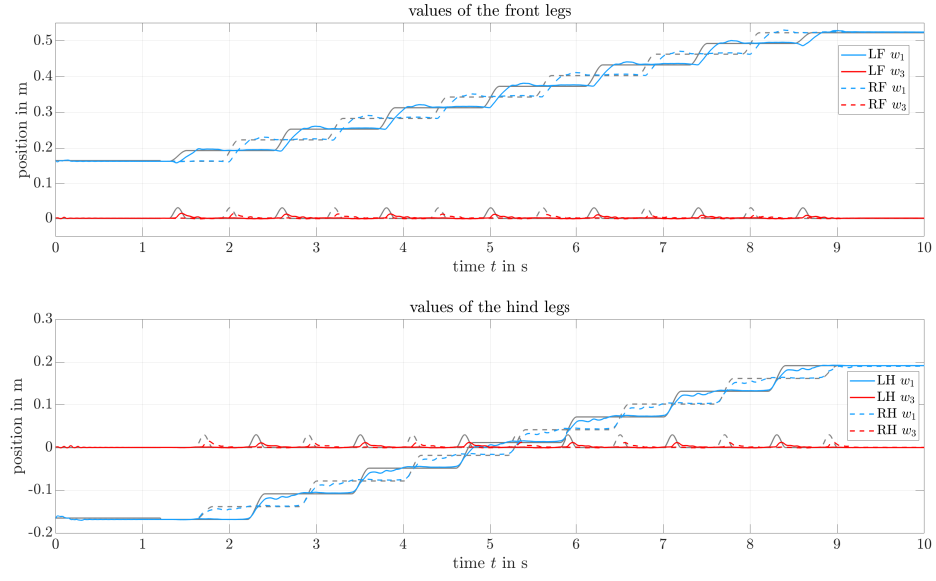


Figure A.7: Records of the robot limbs in run 46 with ideal VRP placement in the world frame. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the first direction and the red lines the third direction. The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references, where the VRP waypoints are \mathbf{g}_σ (s. figure A.4).

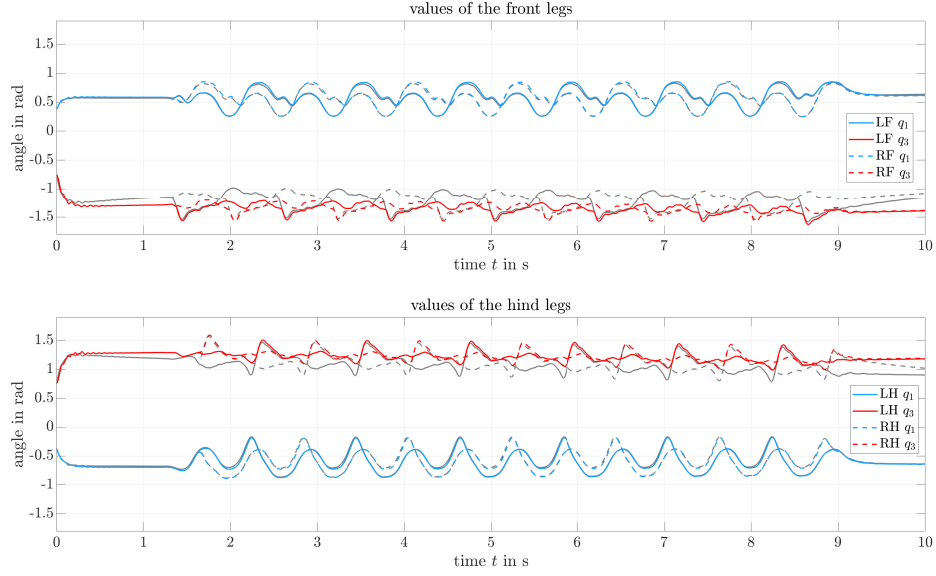


Figure A.8: Records of the robot leg joints in run 46 with ideal VRP placement. The upper plot shows the values of the front legs (LF, RF). The bottom plot shows the values of the hind legs (LH, RH). The blue lines represent the hip joints (q_1) and the red lines the knee joints (q_3). The solid lines correspond to the left legs and the dashed lines to the right legs. The gray lines indicate the references, where the VRP waypoints are \mathbf{g}_σ (s. figure A.4).

A.4 Data folder

This section shows the main part of the structure of the data folder, which contains the required files and the simulation results. The remote project folder can be found on the Github-project web page: <https://rmc-github.robotics.dlr.de/wagn-mk/BertSave>.

BertSave:

- *DCMfunctions* \leftarrow folder with the implemented DCM functions
- *MultiContactPlaner* \leftarrow folder with the implemented framework functions
- *Bert_Johannes_Exper_GreyBert3dof.slx* \leftarrow is the extended Simulink model with the current implementation:
 - MATLAB function block *trajectoryPlaner()*
 - MATLAB function block *computeDesiredJointSpeed()*
- *init.m* \leftarrow MATLAB-program which initializes the simulation environment
- *InitialParBert_77.m* \leftarrow MATLAB-program which initializes the robot parameters and is called by the Simulink InitFcn-Callback
- *FilesForExperBert* \leftarrow files for the prepared experiments
 - *controller.slx* \leftarrow Simulink model of the Bert control framework extended with MATLAB function blocks-based workaround of the needed LucaDynamics functions

- *trajgen_walking.slx* ← Simulink sub-model with implemented DCM-based trajectory generator
- *buses.mat* ← file of the Bert Simulink buses with the *LucaDynamics* bus
- *main.m* ← MATLAB-program which initialize the controller and calls the file *init_Bert_DCM.m*. This file initializes the needed parameters for the DCM method in the experiment.
- *CallLDfunctions.m* ← MATLAB-program to create the workaround functions of the *LucaDynamics* functions
- *Bert_LD* ← folder with the *LucaDynamics* workaround MATLAB-functions for 3 DOF Bert created by the MATLAB-program *CallLDfunctions.m*
- *models* ← folder with the sub folder *stl* and *urdf*
 - *urdf* ← folder with the created URDF files for the robot in the simulation and further model version of the robot Bert:
 - * *URDFrealBert2.urdf*
 - * .
 - * .
 - * .
 - *stl* ← folder with the CAD-files to visualize the robot model
- *Run10.mat* ← MATAB - workspace save of the simulation run
- *Run11.mat* ← MATAB - workspace save of the simulation run
- .
- .
- .

A.5 Applied Software

This section lists the applied software for this thesis and other resources.

Software used for the creation of this thesis:

- MiKTeX Version 20.10 ← applied LaTeX enviroment
- TeXstudio 3.0.1 (git 3.0.1) ← applied LaTeX Editor enviroment
- Inkscape 1.0.1 (3bc2e813f5, 2020-09-07) ← applied to generate the figures
- MATLAB® Version 9.8.01.1359463 (R2020a) Update 1 ← applied for programming and evaluation of simulation results

Software used for the simulation:

- MATLAB® 9.1.0.960167 (R2016b) Update 6
- Simulink® Version 8.8
- *LucaDynamics* Library Version 0.3.50 for osl42-x86_64

Hardware used for the simulation:

- Computer system:

-
- openSUSE Leap 15.1 Kernel Version: 4.12.15-lp151.28.52-default 64-bit
 - 8xIntel® Xeon® CPU E5-1620 v3 @ 3.50 GHz
 - 7.7 GiB of RAM

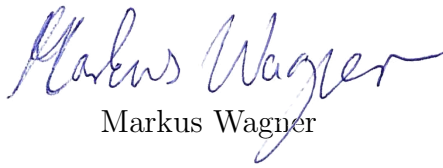
Statement of Originality

I hereby confirm that I have written the accompanying thesis by myself, without contributions from any sources other than those cited in the text and acknowledgements.

Erklärung zur Selbstständigkeit

Ich versichere hiermit, dass ich die von mir eingereichte Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

München, den 15. Januar 2021



Markus Wagner