DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics: Robotics, Cognition, Intelligence

Contact-Robust Whole-Body Control for Torque-Based Humanoid Robots

Sebastian Lohr

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics: Robotics, Cognition, Intelligence

Contact-Robust Whole-Body Control for Torque-Based Humanoid Robots

Kontakt-Robuste Ganzkörperregelung für Drehmoment-Basierte Humanoide Roboter

Author:	Sebastian Lohr
Supervisor:	Alin Albu-Schäffer
Advisor:	Johannes Englsberger
Submission Date:	17.01.2022

I confirm that this master's thesis in informatics: robotics, cognition, intelligence is my own work and I have documented all sources and material used.

Munich, 17.01.2022 Sebastian Lohr

Abstract

Bipedal humanoid locomotion offers great opportunities for the movement of robots in human environments and interaction with them. It is a challenging control problem for which many different solution approaches have been developed in research in recent years. This work tackles the problem of dynamic modelling errors in the form of constraint violations for inverse dynamics control approaches. A possible constraint violation is contact loss of the foot to the ground, but also other unforeseen disturbances may occur. The introduced methods aim to increase robustness and the probability of a recovery.

This work is based on the idea of controlling relative instead of absolute positions, specifically the relative position of the center of mass to the foot (RCF). This solves the problem of uncontrollable degrees of freedom during contact loss when tracking the absolute position is impossible resulting in potentially dangerous controller behavior. Instead, the developed relative controller keeps the posture of the robot in a desirable configuration during contact loss without compromising on the absolute position tracking under nominal foot-ground contact.

A passivity based control task is derived for the RCF coordinate. This task is evaluated with different free-floating robot models and embedded into the whole-body control framework of the humanoid robot Toro. Two different additional task formulations are derived for correct tracking of the absolute position reference based on the constrained foot acceleration and the center of mass behavior with foot-ground contact. An approach for implicit orientation control via multiple contact point RCF tasks is introduced. Different passivation approaches are evaluated to suppress short instabilities in the system. The derived controller is analysed in detail in simulation for the humanoid robot and then verified in experiments.

Contents

1 Introduction

Technological advancement and research has brought automation into many areas of the human's world. While robots were used for a long time mainly in closed manufacturing settings, their applications have developed into the day to day life in form of vacuum clean robots, driving assistance and collision avoidance or as assistance robots at home or in elderly care homes. With a broader field of applications, tasks become more varying and challenging. Reactions and adaptions to unforeseen changes in the environment become necessary. Beside cognitive improvements also the mechanical limits have to be pushed for new applications. One field of research goes to robots with humanoid locomotion. They provide an entire new set of possible work locations compared to wheeled robots which can easily be stopped by stairs. With similar locomotion to humans they can be expected to move in human environments, perform similar tasks and offer new levels of corporation between robots and humans.

First control approaches for free-floating humanoid robots were using position controlled inverse kinematics approaches [1][2][3]. They were following a preset path of motion. As for position controlled fixed base robots, their motion is very rigid and they cannot be controlled in a compliant mode. Similar to fixed-based robots, the recent development went towards inverse dynamics approaches who offer higher compliance [4][5][6]. The inverse dynamics approaches, however, have robustness issues and are susceptible to constraint violations and external disturbances that are not modelled correctly. Such disturbances as an uneven ground, slippage of the foot, a fall or an external force like a push lead to failure of the controller and potentially dangerous uncontrolled motion. Recent developments have been made that tackle problems of pushes and uneven surfaces [7][8][9].

This work focusses on dynamic modelling errors in form of constraint violations as a contact loss and seeks to increase the corresponding robustness. Preliminary work was already presented in [10]. It is inspired by the control of free-floating robots in space for which the centroidal momentum cannot be controlled. Similarly some degrees of freedom of freedom of a robot become uncontrollable during a fall or slippage. Instead, the motion is expressed in terms of relative movements. A controller, tracking an absolute position in space, may produce large torques when the absolute position diverges from the reference due to its uncontrollability. This large torque is potentially dangerous and makes recovery unlikely in case of a reestablishment of the contact because the posture diverges greatly from the desired posture required for stance or walking. Instead a new control quantity is introduced, the relative coordinate between the center of mass and the foot (RCF). The center of mass has been chosen due to its crucial role in most humanoid locomotion approaches, as the main goal is the movement

of the center of mass. If the foot is fixed in the world due to ground contact, the relative controller tracks the absolute position in space. In case of a contact loss, the foot is kept in an appropriate position relative to the center of mass such that recovery is more likely.

This work will extend the approach from [10] from 1D to 3D. The developed RCF task is embedded into the whole body control framework presented in [11] for the humanoid robot Toro [12]. An approach is introduced for orientation tracking using multiple (contact) points on the robot. For correct tracking of the center of mass in case of foot-ground contact, an additional task is derived that is based on the assumption of zero acceleration of the foot. Simpler 3D models of a free-floating robot than Toro will be derived to evaluate the controller. The approach is tested in the simulation for the humanoid robot Toro. An alternative to the zero foot acceleration task with greater robustness to contact loss will be derived which is based on the center of mass acceleration. The controller exhibited short instabilities during stance and walk on the ground. Different approaches of passivation are presented and embedded into the quadratic programming solver. After satisfying results in the simulation, first experiments on the real robot Toro were made and the results will be presented.

2 Robot Models

This chapter is an introduction to robot dynamics and the control of the robot's behavior. It introduces the different robot models used in this work. Beside the model of the humanoid robot Toro, it contains the one dimensional, two degrees of freedom model presented in [10]. A free-floating 3-joint robot arm is derived that is fully determined by the introduced 3-dimensional relative position controller. Additionally, a six-dimensional model that corresponds to one of Toro's legs is introduced.

2.1 General Robot Dynamics

A robot's state is expressed using generalized coordinates. Those generalized coordinates are chosen to describe the state of the robot with the minimal amount of variables. A vehicle moving on a flat surface is defined by three variables, the 2D position and the rotation about the vertical axes. A vehicle moving in three dimensions like an airplane requires three coordinates for the position and three for the orientation. For fixed-base robotic manipulators with revolute or linear joints, the state is fully defined with one coordinate for each robot joint describing the joint's position. The robots investigated in this work are free-floating robots. Their state is defined by the Cartesian position of the base of the robot in space and the joint position of the actuators.

These generalized coordinates are summarized in the generalized coordinate vector q. The change of the robot's state over time, the robot's dynamic behavior, is described by the time derivatives of the generalized coordinates $\dot{q}$ and $\ddot{q}$, the velocity and the acceleration respectively.

The dynamics can be denoted as a general equation of motion of the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g(q) = \tau \tag{2.1}$$

It contains the inertia of the robot in form of the mass matrix M, the coriolis effects of the motion in the Coriolis matrix C, the influence of gravity τ_g and the generalized forces τ. These generalized forces consist of the joint torques τ_j, internal forces τ_{int} like joint friction and external forces τ_{ext} acting on links connecting the joints. The joint torques and internal forces are mapped to the generalized coordinates via the joint selection matrix S and the external forces are calculated with the stack of link jacobians L that map a six-dimensional wrench acting on that link to the generalized coordinates:

$$\tau = S^T * (\tau_j + \tau_{int}) + L_{all}^T w_{all} \tag{2.2}$$

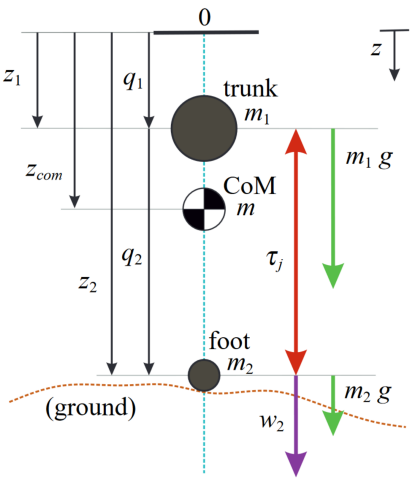


Figure 2.1: Simplest Articulated Free-Floating model [10]

2.2 Simplest Articulated Free-Floating Model

This section reintroduces the simplest articulated free-floating (SAFF) model presented in [10]. The controller introduced in the paper is the base of this work and different controller formulations will be tested on the model. It is the most simple form of a free-floating robot in 1D space. It has two degrees of freedom, the position of the base and the joint position.

2.2.1 Generalized Coordinates and Kinematics

The model shown in 2.1 consists of two point masses that represent a foot and a trunk of a robot. The vertical direction represents the only coordinate used. The two masses are connected via one actuator that can apply a force between the two point masses. Gravity acts on both point masses. In case of contact to the ground, the foot can apply a force in that direction and the position of the foot is fixed. This constrains one degree of freedom of the robot. The generalized coordinates are chosen as the absolute position q_1 of the trunk to a global point of reference and the distance q_2 of the foot to the trunk:

$$q = \left[q_1, q_2\right]^T \tag{2.3}$$

The absolute position coordinates of the trunk z_1 and the foot z_2 can be calculated as

$$z_1 = q_1 \tag{2.4}$$

$$z_2 = q_1 + q_2 \tag{2.5}$$

Their time derivative, the velocity is denoted by $\dot{z}_1$ and $\dot{z}_2$

$$\dot{z}_1 = \dot{q}_1 \tag{2.6}$$

$$\dot{z}_2 = \dot{q}_1 + \dot{q}_2 \tag{2.7}$$

A third coordinate is introduced: the position of the center of mass z_{com}. The center of mass plays a crucial role in this work and other bipedal locomotion control approaches (references). The position of the center of mass depends on the position of the foot and the trunk and their respective masses m_1 and m_2 with a total mass of the robot $m = m_1 + m_2$:

$$z_{com} = \frac{m_1}{m} z_1 + \frac{m_2}{m} z_2 = q_1 + \frac{m_2}{m} q_2 \tag{2.8}$$

Differentiation over time gives the velocity $\dot{z}_{com}$

$$\dot{z}_{com} = \frac{m_1}{m} \dot{z}_1 + \frac{m_2}{m} \dot{z}_2 = \dot{q}_1 + \frac{m_2}{m} \dot{q}_2 \tag{2.9}$$

Due to the linearity of the relation between the generalized coordinates q and the absolute coordinates z the forward kinematics equals the jacobian and will both be denoted by J. The forward kinematics describe the relation of the link coordinates to the generalized coordinates and the jacobians the relation of the link velocities to the generalized velocities.

$$J_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{2.10}$$

$$J_2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{2.11}$$

$$J_{com} = \begin{bmatrix} 1 & \frac{m_2}{m} \end{bmatrix} \tag{2.12}$$

2.2.2 Dynamics

The dynamics of the system is obtained by analyzing each body and the forces on it. Both bodies are subject to gravity g acting in positive z-direction, a torque τ_j acting between the two bodies and a foot wrench w_{foot} acting in the opposite direction of gravity on the foot.

$$m_1 \ddot{z}_1 = m_1 g - \tau_j \tag{2.13}$$

$$m_2 \ddot{z}_2 = m_2 g + \tau_j - w_{foot} \tag{2.14}$$

This is rearranged to obtain the body accelerations $\ddot{z}_1$ and $\ddot{z}_2$

$$\ddot{z}_1 = g - \frac{\tau_j}{m_1} \tag{2.15}$$

$$\ddot{z}_2 = g + \frac{\tau_j - w_{foot}}{m_2} \tag{2.16}$$

The center of mass acceleration is obtained by differentiation of its velocity from (2.9) and using the body accelerations.

$$\ddot{z}_{com} = \frac{m_1}{m} \ddot{z}_1 + \frac{m_2}{m} \ddot{z}_2 \tag{2.17}$$

$$= \frac{m_1}{m} (g - \frac{\tau_j}{m_1}) + \frac{m_2}{m} (g + \frac{\tau_j - w_{foot}}{m_2}) \tag{2.18}$$

$$= g + \frac{w_{foot}}{m} \tag{2.19}$$

Multiplying with the total robot mass m gives the CoM dynamics:

$$m\ddot{z}_{com} = mg + w_{foot} \tag{2.20}$$

2.2.3 Equation of Motion

We have derived three dynamics equations for a system with two degrees of freedom. One of the equations is redundant to express the system in terms of generalized coordinates. The equation for the center of mass and the foot have been chosen because the relative controller will be expressed in a relation between the foot and the center of mass.

$$\begin{bmatrix} m & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{z}_{com} \\ \ddot{z}_2 \end{bmatrix} + \begin{bmatrix} -mg \\ -m_2 g \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau_j + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w_{foot} \tag{2.21}$$

The foot and CoM accelerations can be replaced using the jacobians from (2.10):

$$\begin{bmatrix} \ddot{z}_{com} \\ \ddot{z}_2 \end{bmatrix} = \begin{bmatrix} J_{com} \\ J_2 \end{bmatrix} \ddot{q} = \begin{bmatrix} 1 & \frac{m_2}{m} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \tag{2.22}$$

The foot wrench w_{foot} and the joint torque τ_j can be summarized as a generalized torque vector τ:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau_j + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w_{foot} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \tau_j \\ w_{foot} \end{bmatrix} = \begin{bmatrix} S^T & J_2^T \end{bmatrix} \tau \tag{2.23}$$

with the joint selection matrix $S = \begin{bmatrix} 0 & 1 \end{bmatrix}$. The final equation of motion is the given as

$$\begin{bmatrix} m & m_2 \\ m_2 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -mg \\ -m_2 g \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \tau_j \\ w_{foot} \end{bmatrix} \tag{2.24}$$

This corresponds to a general form with a mass matrix M and gravity vector τ_g and generalized torque τ as in equation 2.1.

$$M\ddot{q} + \tau_g = \tau \tag{2.25}$$

It can be used to calculate the generalized accelerations of the system:

$$\ddot{q} = M^{-1} \left(\tau - \tau_g \right) \tag{2.26}$$

with the inverse mass matrix

$$M^{-1} = \begin{bmatrix} \frac{1}{m_1} & -\frac{1}{m_1} \\ -\frac{1}{m_1} & \frac{m}{m_1 m_2} \end{bmatrix} \tag{2.27}$$

2.2.4 Foot Constraint

The controller is evaluated for its behavior with contact of the foot to the ground and for contact loss. The contact loss is implemented by setting the foot wrench to zero $w_2 = 0$. For the constrained case, the velocity and acceleration is fixed to zero. The foot wrench is calculated from equation (2.14)

$$w_2 = m_2 \left(\ddot{z}_2 - g \right) - \tau_j \tag{2.28}$$

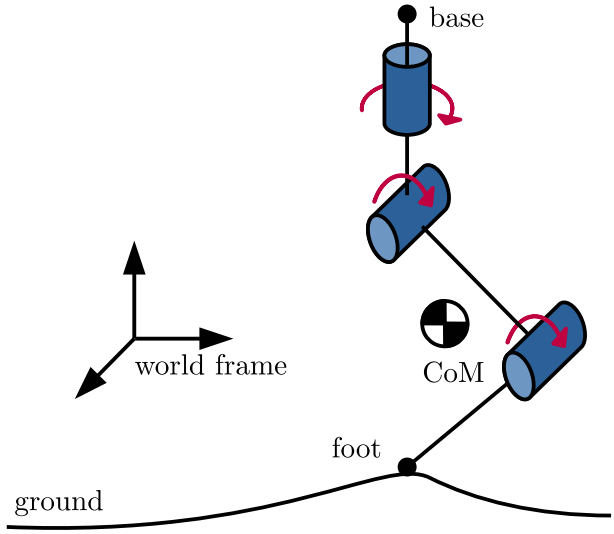


Figure 2.2: Free-floating robot with three revolute joints and six degrees of freedom

2.3 Three Joint Free-Floating Robot

The model for the robot in figure 2.2 was derived to test the introduced three dimensional relative position controller. It has six degrees of freedom with the minimal amount of three joints that are required for the controller, similar to the most simple implementation in 1D with one joint. The linear joint from the SAFF model is replaced by revolute joints as they are used in the humanoid robot. This introduces coupling between the spatial coordinates. The equation of motion will be derived using the Lagrangian method.

2.3.1 Generalized Coordinates and Kinematics

The general coordinates consist of the 3D position of the robot base x in the global reference frame and the three joint positions θ.

$$q = \begin{bmatrix} x & \theta \end{bmatrix}^T \tag{2.29}$$

Based on these coordinates, the homogeneous transformation can be found. This work uses the notation from [13] with a simplified notation of hybrid variables referenced to the world frame ${}^{0}_{h}x_{0,com} = x_{com}$. Variables with a spatial or body reference frame will be indicated. A homogeneous transformation ${}^{B}H_A$ is the transformation from coordinate system B to coordinate system A including the rotation matrix R between the two coordinate systems and the position vector p describing the linear location of the base frame.

$${}^{B}H_A = \begin{bmatrix} {}^{B}R_A & {}^{B}_{h}p_{B,A} \\ 0_{1x3} & 1 \end{bmatrix} \tag{2.30}$$

The homogeneous transformation H_b describes the transformation of the global coordinate system to the base coordinate system. The base frame has the same orientation as

the global one, therefore the rotation matrix is just the identity matrix. The translation between the two frames is given as the vector x_b.

$$H_b = \begin{bmatrix} I_{3x3} & x_b \\ 0_{1x3} & 1 \end{bmatrix} \tag{2.31}$$

The robot's points of interest for which homogeneous transformations have to be found are the three joints, the four links and the endeffector. The center of mass is defined by only a position vector. The links coordinate frames coincide with their center of mass. The transformations are found using the Denavit – Hartenberg parameters. This gives the transformation for each joint, link and the endeffector to the previous joint of the robot, i.e. ${}^{j3}H_{ef}$ defines the transformation of the endeffector to the third joint. The first link and joint are relative to the base of the robot. The transformations of each point of interest to the global coordinate frame is found by multiplication of the individual transformations leading to that link. Exemplary for the endeffector, the homogeneous transformation ${}^{ef}_{0}H$ from the global frame to the endeffector frame calculated as

$${}^{0}H_{ef} = {}^{0}H_b \; {}^{b}H_{j_1} \; {}^{j_1}H_{j_2} \; {}^{j_2}H_{j_3} \; {}^{j_3}H_{ef} \tag{2.32}$$

The center of mass position is calculated from the position vectors, x_1, x_2, x_3 and x_4, and the masses, m_1, m_2, m_3 and m_4, of each link.

$$x_{com} = \frac{1}{4}\sum_{i=1}^{4} m_i x_i \tag{2.33}$$

2.3.2 Dynamics and Lagrangian Function

As mentioned before, the equation of motion is derived using the Lagrangian method. Unlike the derivation of the dynamics, it is not based on the forces on each body but on the energy of each body. The Lagrangian function L is given as the difference between the kinetic energy T and the potential energy V.

$$L = T - V \tag{2.34}$$

The kinetic and potential energy has to be derived for each link of the system. The kinetic energy of link i consists of its linear and rotational part around the center of mass of the link.

$$E_{kin,i} = \frac{1}{2}\dot{x}_i^T m_i \dot{x}_i + \frac{1}{2}\omega^T I_i \omega \tag{2.35}$$

with the moment of inertia I_i. The linear velocity $\dot{x}$ and angular velocity ω need to be calculated for each link. The linear velocity is found by differentiating the position x of each link. The jacobian of each link is found by differentiating the position of each link with respect to the generalized coordinates

$$J_i = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \cdots & \frac{\partial x_1}{\partial q_4} \\ \cdots & & \cdots \\ \frac{\partial x_4}{\partial q_1} & \cdots & \frac{\partial x_4}{\partial q_4} \end{bmatrix} \tag{2.36}$$

The jacobian defines the relation between the generalized coordinates and the velocity of the link

$$\dot{x}_i = J_i \dot{q} \tag{2.37}$$

The angular velocity of a link ω_i is the sum of each link's angular velocities prior to the respective link in the robot.

$$\omega_i = \sum_{n=1}^{i} {}^0R_n \omega_{j,n} \tag{2.38}$$

The potential energy is calculated from the link position x_i and the gravity vector $g = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$

$$E_{pot,i} = m_i g^T x_i \tag{2.39}$$

The energies of all links define the Lagrangian

$$L = \sum_{i=1}^{4} T_i - \sum_{i=1}^{4} K_i \tag{2.40}$$

2.3.3 Derivation of the Equation of Motion

The Euler-Lagrange equation defines the equation of motion without external forces.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0 \tag{2.41}$$

From this equation of motion the mass matrix M, the coriolis matrix C and the gravity vector τ_g are collected. The coriolis matrix has to be factorized such that the skew-symmetry property is fulfilled. $C^T - C$ must be skew symmetric. The matrices are now dependant on the robot's state and the equation of motion is rewritten as.

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + \tau_g(q) = 0 \tag{2.42}$$

The equation is missing the generalized torques τ compared to the general from equation (2.1). The actuators are the three joint torques τ_j and the wrench at the endeffector w_{ef} which is a 3D linear force. The actuators u have to be mapped to the generalized coordinates. The transpose of the joint selection matrix S maps the joint torques and the transpose of the endeffector Jacobian J_{ef} the wrench

$$\tau = S^T \tau_j + J_{ef}^T w_{ef} \tag{2.43}$$

The joint selection matrix selects the actuated joints from the generalized coordinates

$$S = \begin{bmatrix} 0_{3x3} & I_{3x3} \end{bmatrix} \tag{2.44}$$

The joint selection matrix and endeffector jacobian can be collected to one actuation mapping matrix U. This actuation mapping matrix maps the stack of actuators u to the generalized coordinates.

$$\tau = Uu = [S^T J_{ef}^T] \begin{bmatrix} \tau_j \\ w_{ef} \end{bmatrix} \tag{2.45}$$

This yields the final equation of motion of the same form as in (2.1)

$$M\ddot{q} + C\dot{q} + \tau_g = \tau \tag{2.46}$$

2.3.4 Simulation

The simulation is carried out by numerical integration of the system state with the acceleration of the generalized coordinates $\ddot{q}$. The acceleration is obtained from the system's equation of motion

$$\ddot{q} = M^{-1}(\tau - C\dot{q} - \tau_g) \tag{2.47}$$

The forward Euler method has been used for integration

$$q^t = q^{t-1} + \dot{q}^{t-1}\Delta t + \frac{1}{2}\ddot{q}\Delta t^2 \tag{2.48}$$

$$\dot{q}^t = \dot{q}^{t-1} + \ddot{q}\Delta t \tag{2.49}$$

The variables q^{t-1} and $\dot{q}^{t-1}$ correspond the position and velocity of the previous time step. Δt is the time between two discrete time steps.

2.3.5 Foot Constraint

The controller will again be evaluated for the two cases where the foot position is constrained and completely unconstrained. For the case of the unconstrained foot, the wrench is set to zero, resulting in a free fall.

$$w_{ef} = 0_{3x1} \tag{2.50}$$

The resulting wrench in the case of a constrained endeffector has to be derived from the its acceleration.

$$\ddot{x}_{ef} = \frac{d}{dt}\left(J_{ef}\,\dot{q}\right) = \dot{J}_{ef}\,\dot{q} + J_{ef}\,\ddot{q} \tag{2.51}$$

Inserting the generalized acceleration from the system dynamics as in (2.47) yields

$$\ddot{x}_{ef} = \dot{J}_{ef}\,\dot{q} + J_{ef}\,M^{-1}(S^T\tau_j - C\dot{q} - \tau_g) + J_{ef}\,M^{-1}\,J_{ef}^T\,w_{ef} \tag{2.52}$$

The wrench that constrains the foot in the simulation is found by reordering

$$w_{ef} = \left(J_{ef}\,M^{-1}\,J_{ef}^T\right)^{-1}\left(\ddot{x}_{ef} - \dot{J}_{ef}\,\dot{q} - J_{ef}\,M^{-1}(S^T\tau_j - C\dot{q} - \tau_G)\right) \tag{2.53}$$

The constrained foot acceleration $\ddot{x}_{ef}$ is set to zero assuming contact to a non-moving surface. The term can also be used to evaluate a controller's performance in case of an accelerating floor like elevators where this term would be non-zero.

2.4 27 Joint Humanoid Robot Toro

2.4.1 Generalized Coordinates

The humanoid robot Toro has a higher complexity with its 27 joints. Together with the 6D location of the base, the robot has 33 degrees of freedom. The position of the base is defined by a 3D position vector as for the previous model, but additionally also includes the orientation of the base coordinate frame located at the hip. The external forces are

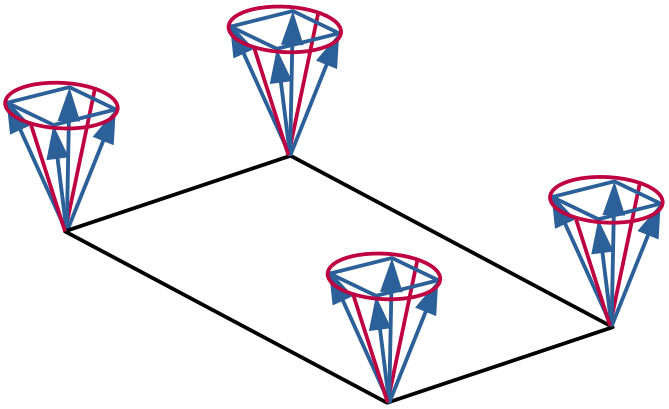


Figure 2.3: Contact point model of a foot with polyhedral convex cone approximation

given by the joint torques and the contact forces at the foot. The contact is modeled as contact force vectors as in 2.3 and friction cones at each of the 4 foot's corners. A friction cone represents all force vectors which are allowable without slipping. The allowable tangential force is defined by the friction coefficient and the normal force.

$$f_T = \mu f_N \tag{2.54}$$

With higher friction, the friction cone becomes wider as higher tangential forces are allowable. As the cone representation is non-linear, it is easier to use a simplified representation as polyhedral convex cones. The convex property ensures that the combination of vectors from within the cone also lie within the friction cone. The polyhedral is chosen to be represented by four linear contact forces out of unit vectors u_i that define the edges of the pyramid and the force magnitude ρ_i.

$$f_i = u_{f_i} \rho_i \tag{2.55}$$

The combination of the four linear contact forces define the force applied at that corner point. With four polyhedral edges per contact point, there are 16 linear contact forces as external forces per foot making a total of 32 contact force magnitudes for the system. The contact force vectors can be mapped to a spatial wrench ${}_s w_{EE}$ at the foot using the mapping ${}_s A_\rho$.

$${}_s w_{EE} = {}_s A_\rho \, \rho \tag{2.56}$$

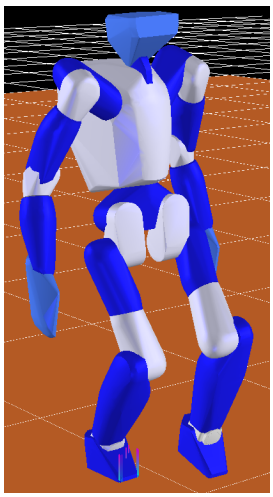


Figure 2.4: Model of the humanoid robot in the simulation using OpenHRP

The spatial wrench is mapped to the generalized coordinates with the spatial jacobian of the endeffector ${}_sJ_{EE}$. The general forces can then be calculated

$$\tau = U\,u = \begin{bmatrix} S_{act}^T & {}_sJ_{EE}^T\ {}_sA_\rho \end{bmatrix} \begin{bmatrix} \tau_j \\ \rho \end{bmatrix} \tag{2.57}$$

2.4.2 Simulation

The simulation for the full model of Toro is implemented with OpenHRP [14] (open architecture humanoid robotics platform) which is built for analysis of humanoid robots including collision and contact computations. The dynamics of the robot are derived as in [15][16]. The OpenHRP simulation uses these system matrices instead of the inbuilt dynamics computation.

2.4.3 Contact Model

The contact model is implemented as in [17]. It uses the forward dynamics of the system and a collision detector to check whether contact is made between two bodies. If collision is detected, a normal force is calculated that stops the intrusion and defines the normal contact force. The collision is checked for the corner points of the foot and the normal force is calculated if contact is made. The tangential force is evaluated as a polyhedral friction cone and the external forces as in (2.1).

2.5 Six Joint Free-Floating Robot Leg

This model is a reduced version of the humanoid robot. It consists of one of the robot's legs until the hip with 6 joints. This is the minimal amount of joints to control the

position and the orientation of an endeffector. With the 6 coordinates of the robot's base position in the global coordinate frame, this model has 12 degrees of freedom. The generalized coordinates are the six joint positions and the position of the base.

The forces u are the robot's joint torques τ_j and a 6-dimensional spatial wrench ${}_s w$ of the foot to the ground. They are mapped to the generalized torque τ by the actuation mapping matrix U. It consists of joint selection matrix S and the spatial jacobian of the foot ${}_s J_{foot}$.

$$\tau = U\,u = \begin{bmatrix} S^T & {}_s J_{foot}^T \end{bmatrix} \begin{bmatrix} \tau_j \\ {}_s w \end{bmatrix} \tag{2.58}$$

The simulation was adduced to analyse instabilities that occurred during stance of the robot Toro. The simulation with this model proved to be numerically instable if the foot was unconstrained, without the influence of a controller. The integration was carried out using Forward-Euler as for the 3-joint model in section 2.3.4. The model was therefore only deducted for analysis purposes with a constrained foot. The foot constraint is implemented similarly to 2.3.5. This model's foot constraint differs in form the extension of the wrench w_{ef} and the endeffector acceleration $\ddot{x}_{ef}$ to 6D.

3 Controller Design and Modular Passive Tracking Control

This chapter gives an overview of methods for controller design on which the controller of this work is based. The modular passive tracking controller (MPTC) introduced in [11] offers a framework for handling multiple tasks, directly taking into account the important issue of passivity and thus achieving a high level of robustness. .

3.1 Task Space

Tasks for a robot are often given in terms of spatial coordinates that a part of the robot should follow. Those spatial coordinates can be six dimensional, but also less in case only position or orientation are of interest. The general equation of motion from (2.1) can be projected into task space and a passivity based control law for this task space derived. The spatial coordinates for a specific task are denoted by x_k. A task specific jacobian J_k maps the time derivative of the generalized coordinates to the time derivative of the task coordinate $\dot{x}_k$:

$$\dot{x}_k = J_k \dot{q} \tag{3.1}$$

The acceleration is obtained by differentiation

$$\ddot{x}_k = J_k \ddot{q} + \dot{J}_k \dot{q} \tag{3.2}$$

The general equation of motion 2.1 for the robot can be rearranged to obtain the generalized acceleration $\ddot{q}$

$$\ddot{q} = M^{-1}\left(\tau - C\dot{q} - \tau_g\right) \tag{3.3}$$

which gives the final task space acceleration

$$\ddot{x}_k = J_k M^{-1}(\tau - \tau_g) - Q_k \dot{q} \tag{3.4}$$

with

$$Q_k = J_k M^{-1} C - \dot{J}_k \tag{3.5}$$

The system matrices are transformed into task space with task jacobian and its pseudo-inverse T_k which maps the generalized torques into the task space

$$M_k = (J_k M^{-1} J_k^T)^{-1} \tag{3.6}$$

$$T_k = M_k J_k M^{-1} \tag{3.7}$$

$$C_k = M_k Q_k T_k^T \tag{3.8}$$

$$\tag{3.9}$$

The task force f_k is found by mapping the generalized torque into the task space

$$f_k = T_k \tau \tag{3.10}$$

3.2 Passivity Based Control Task

A control law was design in [11] to follow a reference trajectory in that task space which is defined by a position, velocity and acceleration reference: $x_{k,ref}$, $\dot{x}_{k,ref}$ and $\ddot{x}_{k,ref}$. The error of the task between the reference and the actual value can be calculated

$$\tilde{x}_k = x_{k,ref} - x_k \tag{3.11}$$

$$\tilde{\dot{x}}_k = \dot{x}_{k,ref} - \dot{x}_k = \dot{x}_{k,ref} - J_k \dot{q} \tag{3.12}$$

$$\tilde{\ddot{x}}_k = \ddot{x}_{k,ref} - \ddot{x}_k = \ddot{x}_{k,ref} - J_k M^{-1}(\tau - \tau_g) + Q_k \dot{q} \tag{3.13}$$

Passivity based control approaches find a Lyapunov function V for the system which correlates to its energy. A control law is stable if it guarantees that the derivative of that Lyapunov function is smaller than zero with that controller. A negative Lyapunov rate correlates to a dissipation of energy from the system and asymptotic stability can be shown. The task Lyapunov function V_k has to be zero if and only if the task space quantities x_k and $\dot{x}_k$ are zero and greater zero if and only if they are unequal zero. The task Lyapunov function is based on the kinetic and potential energy in the task space. The kinetic energy is defined by the square of the task space velocity error $\tilde{\dot{x}}_k$ and the task inertia M_k, the potential energy by the square of the task space position error $\tilde{x}_k$ and the stiffness matrix of the task K_k.

$$V_k = \frac{1}{2}\tilde{\dot{x}}_k^T M_k \tilde{\dot{x}}_k^T + \frac{1}{2}\tilde{x}_k^T K_k \tilde{x}_k \tag{3.14}$$

The stiffness matrix is square symmetric and positive definite. A higher task stiffness leads to a more aggressive movement towards the task reference. Differentiation gives the Lyapunov rate

$$\dot{V}_k = \tilde{\dot{x}}_k^T \left(M_k \tilde{\ddot{x}}_k + \frac{1}{2}\dot{M}_k \tilde{\dot{x}}_k + K_k \tilde{x}_k \right) \tag{3.15}$$

The goal is to define a desired task force $f_{k,des}$ which ensures that the Lyapunov rate simplifies to a purely dissipative function $\dot{V}_k < 0$ if the actual task force is equal to the desired task force.

$$\dot{V}_k = -\tilde{\dot{x}}_k^T D_k \tilde{\dot{x}}_k + \tilde{\dot{x}}_k^T \tilde{f}_k \tag{3.16}$$

with the task force error as the difference between the desired and actual task force

$$\tilde{f}_k = f_{k,des} - f_k \tag{3.17}$$

The dissipation is defined by the damping matrix D_k of the task. The damping matrix is also square and positive definite. The desired task force is given by

$$f_{k,des} = T_k \tau_g + M_k Q_K \dot{q} + M_k \ddot{x}_{k,ref} + (C_k + D_k)\tilde{\dot{x}}_k + K_k \tilde{x}_k \tag{3.18}$$

and the task space error dynamics as

$$M_k\ddot{\tilde{x}}_k + (C_k + D_k)\dot{\tilde{x}}_k + K_k\tilde{x}_k = \tilde{f}_k \tag{3.19}$$

3.3 Stacking of Tasks for Whole Body Control

For a complex robot with many degrees of freedom, the robot's behavior is not fully-defined by one task and multiple tasks become necessary. The overall controller is found by stacking several different tasks that define the desired behavior of the robot. For each task a desired task force $f_{k,des}$ is calculated as in the previous subsection and a task mapping T_k that maps the generalized torque into that task space. The overall controller can be calculated from those quantities for all tasks. Each task is weighted and a trade-off between tasks has to be made in case of conflicting tasks or under-actuation of the system, when more degrees of freedom would be required to fulfill all tasks than degrees of freedom are available in the system. All desired task forces are stacked into one vector f_{des}. The task mapping matrices are multiplied with the actuation mapping matrix U of the robot model to obtain the actuation to task mapping $T_{u,k}$. The mappings are stacked into one matrix T. Each task is weighted by a positive definite square matrix W_k. The size of the weighting is equal to the dimension of the task space. For this work, each task is weighted by one scalar w_i. The task weighting matrix is found by multiplication with the inverse task mass matrix

$$W_k = M_k^{-1} w_i \tag{3.20}$$

A higher task weight increases the importance of that task and reduces the task force error in case of a conflict compared to tasks with a lower weight. The weighting matrices of all tasks form the block-diagonal overall weighting matrix W.

The actual commanded actuation u_{cmd} can be computed analytically or via minimization of a cost function using quadratic programming (QP). The analytical solution is given by

$$u_{cmd} = \left(T^T W T\right)^{-1} T^T W f_{des} \tag{3.21}$$

Quadratic optimization offers the advantage that constraints and limits of actuators can be considered. The optimization problem is given by a quadratic cost function G subject to a linear constraint defined by the boundaries b and the constraint mapping matrix A

$$min\ G = \frac{1}{2}x^T Q x + c^T x \tag{3.22}$$

$$subject\ to\ Ax \leq b \tag{3.23}$$

The cost function for the control problem is given by

$$G = \frac{1}{2}u_{cmd}^T T^T W T u_{cmd} - f_{des} W T u_{cmd} \tag{3.24}$$

The optimization problem is solved with qpOases [18].

3.4 Overall Passivity Analysis

It has been shown in the MPTC paper [11] that the overall Lyapunov rate of the system defined by passive tasks is negative even if the lyapunov rate of some tasks is positive due to conflicting tasks or under-actuation. If the commanded task force is unequal to the desired task force because of the trade-off the task force error may become larger than the dissipation in equation (3.16).

3.5 PD Controller Design - Double Diagonalization

The task error dynamics ()3.19) correspond to a mass-spring-damper system if the task force error is zero. The controller design matrices for the stiffness K and the damping D will be designed for that nominal case. The damping matrix is calculated accordingly to achieve a certain damping ratio for the task. In this work, all tasks have been chosen to be critically damped. Higher damping values are vulnerable to fast oscillations and high frequency measurement noise. Lower damping leads to overshooting and oscillations till convergence. A constant damping matrix would lead to varying damping ratios as the mass matrix of the error dynamics is not constant over time. The damping matrix has to be adapted accordingly to achieve the desired damping ratio with the desired task stiffness at all times. This work uses the double diagonalization method presented in [19]. A non-singular matrix Q is found via decomposition of the mass matrix into eigenvalues and eigenvectors. The relation of Q to the mass matrix is given by $M = QQ^T$ and to the stiffness matrix $K = QK_0Q^T$ with the desired stiffness for each coordinate on the diagonal matrix K_0. The damping matrix is then calculated as

$$D = 2QD_\eta K_d^{1/2} Q^T \tag{3.25}$$

with the desired damping ratio η for each coordinate on the diagonal of the matrix D_η.

4 Relative Controller - Foot Acceleration Constraint

This chapter introduces the relative controller presented in [10]. The paper introduced a controller that tracks the position of the foot relative to the center of mass. This relative center of mass to foot controller (RCF) proved to be bounded-input, bounded-output (BIBO) stable for the case of contact imperfections in form of a contact loss while showing perfect tracking of a reference trajectory for the center of mass if the foot is constraint by the ground by enforcing an acceleration constraint on the foot. This controller has been introduced for one dimension for the SAFF model from section (2.2). The first part of this thesis consists of extending the controller idea of a relative coordinate to 3D and develop a controller based on that. First, the controller will be evaluated using the simple 3-joint robot from section 2.3 which is fully determined by the developed RCF controller. The introduced control tasks will be embedded into a whole body control framework which is evaluated with the simulation for the humanoid robot in terms of the walking performance and the robustness to contact loss and also the six-joint robot leg.

4.1 SAFF Model

A control quantity had to be found that is more robust to position errors due to contact imperfections than absolute coordinates. The controlled quantity diverges from the reference due to a missing reactionary force of the ground and is unstable as it is uncontrollable. In such a falling scenario it is more desirable to keep the robot in a configuration that increases the probability of a recovery after contact has been regained. The configuration can be defined by controlling body parts relative to each other. This lead to the idea of expressing the entire motion as a relative motion between body parts and controlling their relative position. It has been chosen to control body parts relative to the center of mass of the robot. The center of mass is chosen due to its crucial role in humanoid locomotion approaches. The locomotion is implemented by controlling the foot position relative to the center of mass.

4.1.1 Relative Coordinate

The new coordinate is introduced as the difference of the center of mass position z_{com} to the foot position z_2.

$$z_R = z_{com} - z_2 \tag{4.1}$$

The relative velocity is obtained by differentiation.

$$\dot{z}_R = \dot{z}_{com} - \dot{z}_2 = J_{com}q - J_2 q = (J_{com} - J_2)q \tag{4.2}$$

From this equation the relative jacobian is found as it maps the generalized coordinates to the task space velocity

$$J_R = (J_{com} - J_2)q = \begin{bmatrix} 0 & -\frac{m_1}{m} \end{bmatrix} \tag{4.3}$$

The task space system matrices are derived as in the MPTC framework 3.6. The relative task inertia is given by

$$m_R = (J_R M^{-1} J_R^T)^{-1} = \frac{m_2}{m_1} m \tag{4.4}$$

With the relative task mapping matrix T_R

$$T_R = M_R J_R M^{-1} = \begin{bmatrix} \frac{m_2}{m_1} & -\frac{m}{m_1} \end{bmatrix} \tag{4.5}$$

and the task force

$$f_R = T_R \tau = T_R U u = -\frac{m}{m_1}\tau_j - w_2 \tag{4.6}$$

4.1.2 Relative Controller Task

The control task is formulated as an MPTC task, which gives the desired task force

$$f_{R,des} = M_R \ddot{x}_{R,ref} + D_R \dot{\tilde{x}}_R + K_R \tilde{x}_R \tag{4.7}$$

$$= \frac{m_2}{m_1} m \ddot{x}_{R,ref} + d_R \dot{\tilde{x}}_R + k_R \tilde{x}_R \tag{4.8}$$

with the Coriolis term vanished due to the lack of rotary movements in the system. The actuator to relative task force mapping yields

$$T_{U,R} = \begin{bmatrix} -\frac{m}{m_1} & -1 \end{bmatrix} \tag{4.9}$$

In the paper [10], the commanded joint torque is found by setting the estimated task force equal to the desired task force and solve for the commanded joint torque.

$$f_{R,est} = T_{U,R} \begin{bmatrix} \tau_{j,est} \\ w_{2,est} \end{bmatrix} = -\frac{m}{m_1}\tau_{j,cmd} - w_{2,est} \tag{4.10}$$

The estimated task force is equal to the actual task force with a perfect actuator with $\tau_j = \tau_{j,cmd}$ and perfect contact of the foot to the ground that the estimated foot wrench equals the foot wrench from ()2.28)

$$w_{2,est} = m_2(\ddot{z}_{2,est} - g) - \tau_{j,cmd} \tag{4.11}$$

This leads to the commanded torque of the controller.

$$\tau_{j,cmd} = -m_1(\ddot{z}_{2,est} - g) - m\ddot{z}_{R,ref} - \frac{m_1}{m_2} d_R \dot{\tilde{z}}_R - \frac{m_1}{m_2} k_R \tilde{z}_R \tag{4.12}$$

Furthermore, the assumption has been made that the robot is in the regulation case with a constrained foot and the estimated foot acceleration $\ddot{z}_{2,est}$ is zero.

4.1.3 Foot Acceleration Constraint Task

When extending to a robot with more degrees of freedom, the commanded joint torque cannot be solved directly from the RCF task force as it has been done in the paper. A second derivation method of the above control law is found using a second task in addition to the RCF task. A equivalent task will be derived in 3D and can be combined with other tasks for a whole body control framework. This task is based of the assumption of zero foot acceleration as it is in the paper. The foot acceleration is given from equation 2.14

$$\ddot{z}_2 \stackrel{!}{=} 0 = g + \frac{\tau_j + w_2}{m_2} \tag{4.13}$$

This equation shows that the foot wrench and the commanded joint torque need to compensate for the effect of gravity. The actuator to task space mapping is given as

$$T_{U,2} = T_2 U = m_2 J_2 M^{-1} U = \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{4.14}$$

The only multibody effect in the foot acceleration is gravity. The desired force is given by rearranging the foot acceleration equation for the joint and wrench with the task space mapping

$$f_{2,des} = J_2 M^{-1} G = -m_2 g \tag{4.15}$$

This task can be stacked with the desired RCF task to the overall desired task force vector

$$f_{des} = \begin{bmatrix} f_{R,des} \\ f_{2,des} \end{bmatrix} \tag{4.16}$$

and the overall task mapping T_U

$$T = \begin{bmatrix} T_{U,R} \\ T_{U,2} \end{bmatrix} \tag{4.17}$$

Using equation (3.21), the commanded torque and foot wrench is found. As the two tasks are non-conflicting and the system fully determined, the weighting matrix is redundant in the equation and it can be solved without trade-off.

$$\tau = (T^T T)^{-1} T^T f_{des} \tag{4.18}$$

This gives the commanded torque

$$\tau_{j,cmd} = m_1 g - m\ddot{z}_{R,ref} - \frac{m_1}{m_2} d_R \tilde{\dot{z}}_R - \frac{m_1}{m_2} k_R \tilde{z}_R \tag{4.19}$$

which is equal to the commanded torque from the paper in equation (4.12) for the assumption of zero foot acceleration. The commanded wrench is calculated as

$$w_{2,cmd} = -m_2 g - \tau_{j,cmd} = -mg + m\ddot{z}_{R,ref} + \frac{m_1}{m_2} d_R \tilde{\dot{z}}_R + \frac{m_1}{m_2} k_R \tilde{z}_R \tag{4.20}$$

4.1.4 Analysis

For the 1D model, it was possible to analyse the error dynamics analytically. The relative task error dynamics was given as

$$m_R\ddot{\tilde{z}}_R + \frac{m}{m_2}d_R\dot{\tilde{z}}_R + \frac{m}{m_2}k_R\tilde{z}_R = \tilde{f}_R \tag{4.21}$$

The task force error $\tilde{f}_R = f_{R,des} - f_R$ vanishes for the constrained foot case, the task force error becomes zero and equation (4.12) leaves the desired task error dynamics. This holds, if the commanded foot wrench from equation (4.20) equals the actual applied foot wrench from the foot constraint in equation (2.28).

$$m_R\ddot{\tilde{z}}_R + d_R\dot{\tilde{z}}_R + k_R\tilde{z}_R = 0 \tag{4.22}$$

In the free falling case, the applied foot wrench is zero and the error dynamics become

$$m_R\ddot{\tilde{z}}_R + \frac{m}{m_2}d_R\dot{\tilde{z}}_R + \frac{m}{m_2}k_R\tilde{z}_R = w_2 + m(g - \ddot{z}_{R,ref}) \tag{4.23}$$

This term is bounded with a bounded relative acceleration reference as the gravity term is constant. This error leads to an offset from the reference to which it converges. The magnitude of that offset is defined by the stiffness of the RCF task in the regulation case

$$\tilde{z}_R = \frac{m_2(g - \ddot{z}_{R,ref})}{k_R} \tag{4.24}$$

After a contact loss happened, the task Lyapunov rate is positive and converges to zero when the Lyapunov function converges to a constant positive value.

In the paper [10], the controller was compared to an inverse dynamics controller for center of mass tracking and an MPTC controller with absolute position tracking of the foot and the center of mass. Both controllers exhibited uncontrolled large joint torques during free fall, while the introduced controller showed desirable behavior and bounded joint torques due to the bounded task force error.

4.2 Extension to 3D

This section develops the three-dimensional version of the previously introduced RCF task. The tracking task from the MPTC framework will be used for the relative task and additionally introduce the foot acceleration task constraint task formulated.

4.2.1 Relative Coordinate

The relative coordinate x_R is given as the difference by the 3D position vector of the center of mass x_{com} and the foot x_F in the global coordinate frame.

$$x_R = x_{com} - x_F \tag{4.25}$$

To obtain the relative task velocity and the jacobian, we differentiate the relative coordinate over time

$$\dot{x}_R = \dot{x}_{com} - \dot{x}_F = J_{com}\dot{q} - J_F\dot{q} = (J_{com} - J_F)\,\dot{q} = J_R\dot{q} \tag{4.26}$$

This gives the relative jacobian J_R

$$J_R = J_{com} - J_F \tag{4.27}$$

The relative task acceleration is given as the difference between the center of mass acceleration and foot acceleration

$$\ddot{x}_R = \ddot{x}_{com} - \ddot{x}_F \tag{4.28}$$

The relative task references are calculated similarly by the difference of the center of mass and foot reference.

4.2.2 Relative Control Task

The RCF task is defined as in (3.18). The desired task force is given by

$$f_{R,des} = M_R Q_R \dot{q} + M_R \ddot{x}_{R,ref} + (C_R + D_R)\tilde{\dot{x}}_R + K_R \tilde{x}_R \tag{4.29}$$

and the task mapping by

$$T_R = M_R J_R M^{-1} \tag{4.30}$$

Compared to the 1D task, this tasks includes additional multibody effects due to Coriolis effects and the non-constant task Jacobian. The effect of gravity is equal for the center of mass and foot acceleration, so that its effect is cancelled out for the relative coordinate.

$$J_R M^{-1} \tau_g = 0 \tag{4.31}$$

This cancelled out gravity term leads to too small commanded actuation forces in case of foot-ground contact and the additional foot acceleration task becomes necessary.

4.2.3 Foot Acceleration Task

This task uses information about the constraint on the foot enforced by the ground. The foot acceleration can be derived as in section 2.3.5 for the foot constraint in equation (2.52) and is assumed zero

$$\ddot{x}_F = J_F M^{-1}(\tau - \tau_g - C\dot{q}) + \dot{J}_F\dot{q} \stackrel{!}{=} 0 \tag{4.32}$$

Solving for the generalized torques splits the equation into the torques and the multibody effects that have to be compensated by them

$$J_F M^{-1}\tau = \dot{J}_F\dot{q} - J_F M^{-1}(\tau_g + C\dot{q}) \tag{4.33}$$

The desired task force and the task mapping is found by premultiplying with the mass of the foot space M_F. The generalized torques are mapped to the task space with the task mapping T_F

$$T_F = M_F J_F M^{-1} \tag{4.34}$$

and the task force is given as

$$f_{F,des} = M_F \left(\dot{J}_F \dot{q} - J_F M^{-1} (\tau_g + C\dot{q}) \right) \tag{4.35}$$

4.2.4 Analysis

The first analysis of this controller setup is done for the three joint, six degrees of freedom robot from section 2.3 as it offers the possibility to fully determine the system and observe the isolated RCF behavior in 3D. For the models with more degrees of freedom, additional tasks are required to fully determine the system which might lead to trade-offs and an influence on the RCF behavior. The RCF task setup is first evaluated

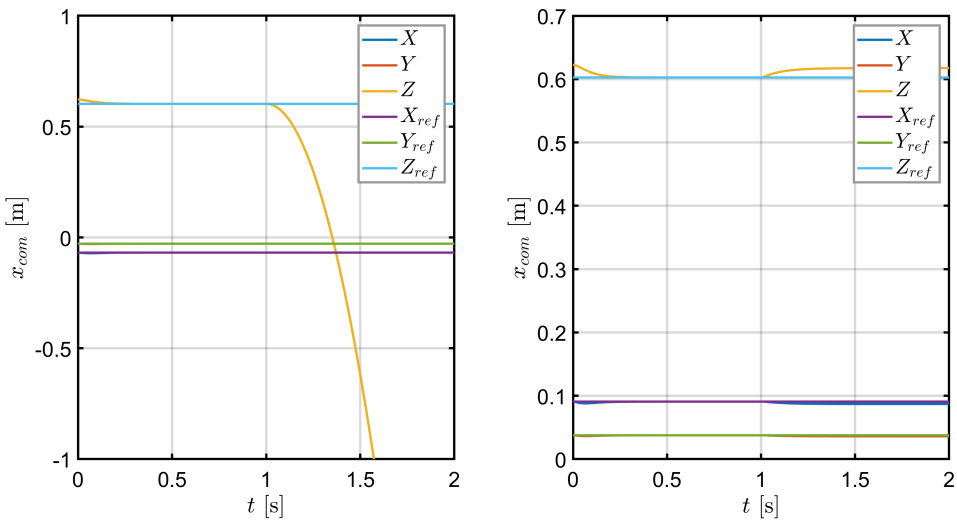
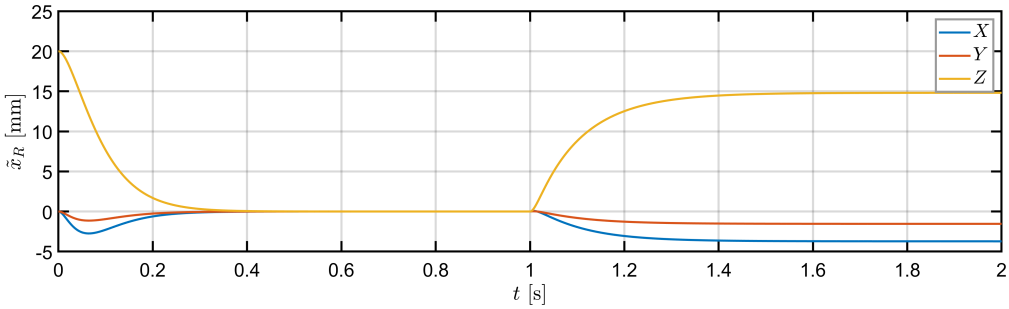


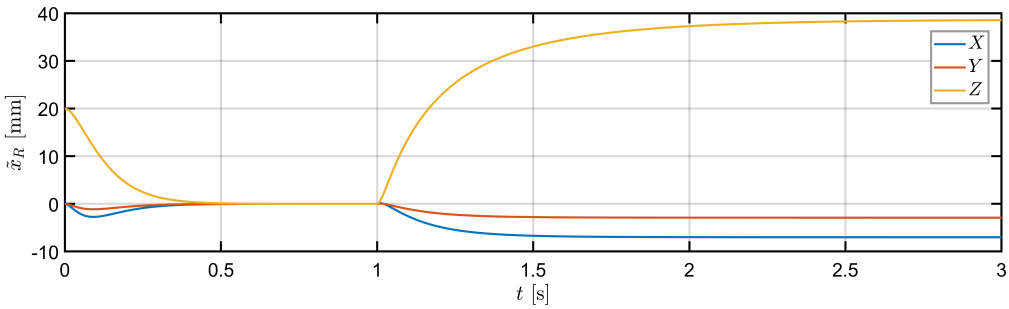
Figure 4.1: Behavior of the center of mass and the relative coordinate during stance and during fall for the 3-joint model

for its behavior during stance with a constant reference with a step response and the behavior during free fall after contact loss at $t = 1\,\mathrm{s}$. Figure 4.1 shows the behavior of the center of mass and the relative coordinate for that scenario. The initial task reference has a two centimeter offset in the vertical z-axis from the start position to see the transient response. Convergence can be observed with the desired error dynamics of a critically damped system while the foot is in contact to the ground. The relative coordinate has the same behavior as the center of mass coordinate, so that the desired center of mass tracking is achieved using the RCF controller. After the contact loss occurs at $t = 1\,\mathrm{s}$ the absolute center of mass coordinate follows the natural falling dynamics. The relative

coordinate however stabilizes and converges to a fix offset. This is similar to the 1D case. However, a coupling between the axes can be observed even though gravity only acts in the vertical z-coordinate in the chosen reference frame and offsets occur in the relative x- and y-coordinate too.



(a) K = 5000

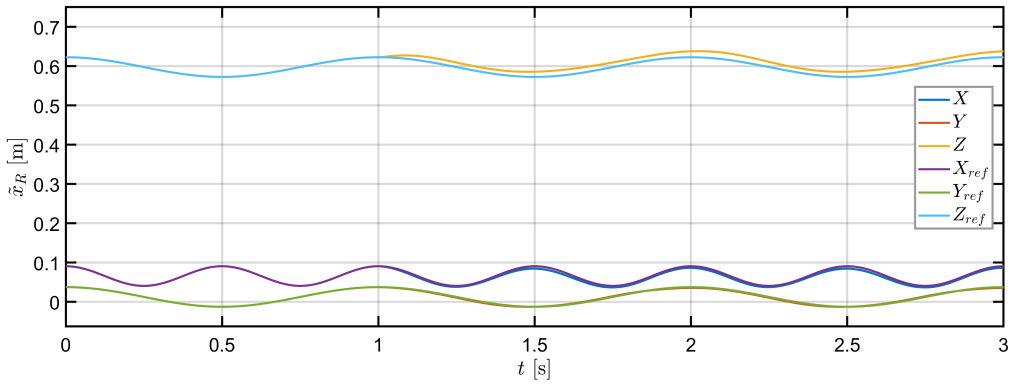


(b) K = 2500

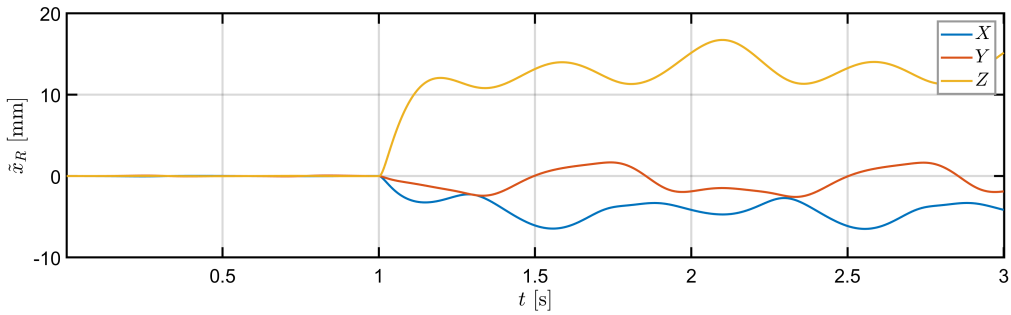
Figure 4.2: Relative task error for different RCF task stiffnesses

Figure 4.2 shows the relative task error $\tilde{x}_R$ for different task stiffnesses in the same scenario with a step response in the beginning and the contact loss and free fall after one second. The task stiffness is halved from the first to the second and again to the third plot. The damping matrices are chosen to critically damp the RCF task. The system converges faster for higher stiffnesses as expected, with critically damped behavior for all three controller gain setups. For the stiffness of $k_i = 5000$, the relative coordinate error converges to almost 15 mm after approximately 0.8 seconds of free fall. For the stiffness of $k_i = 2500$, the convergence time increases to two seconds and the offset to around 38 mm. As expected from the previous 1D chapter, the offset increases for a smaller task stiffness. The anti-proportional relation from the 1D case between the task stiffness and the offset is not given in 3D, but can be used as a good approximation. Close to the singular position, the accuracy of the anti-proportional approximation reduces.

Figure 4.3 shows the tracking of the relative coordinate and its error for the tracking of a trigonometric trajectory. Tracking is almost perfect and no error visible during the



(a) Relative task position



(b) Relative task position error

Figure 4.3: Tracking of the relative coordinate for sine trajectories for each coordinate during stance and fall

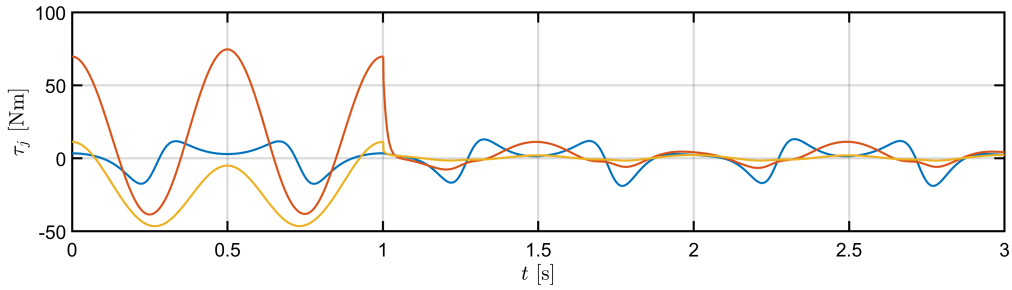


Figure 4.4: Joint torques while following a trigonometric trajectory during stance and free fall

first second of stance. After the contact loss, the relative position diverges from the reference and stabilizes at a non-constant offset which is periodic with the reference signal. The desired motion of the robot is continued. When the stiffness is reduced, that the robot arm gets to a singular position, the controller became instable due to the task inertia becoming infinitely large.

Figure 4.4 shows the joint torques of the tree motors during stance and fall for the same scenario as the previous plot. A reduction of the joint torques is visible after the drop at one second and smooth joint torques are observed.

4.3 Whole Body Control

After the successful extension to 3D, the RCF task will be applied to the more complex humanoid robot with more degrees of freedom which also requires additional tasks and control of the foot orientation during swing phases. A general 3D orientation control task is derived which can be used for the foot, but also to control the upper body orientation. This improved the walking performance when walking on the ground. An alternative to the foot orientation task is introduced. The relative center of mass to foot task is reformulated to a relative center of mass to contact point task which has been primarily used in this work. By using one position control task for each corner point of the foot, the orientation of the foot is controlled implicitly. The contact points coincide with the contact model described in section 2.4.3. To fully define the robot with its 27 joints, many position or orientation tasks would be required. Instead one low weighted fixed joint posture task is derived to fully define the robot's joints in addition to the set of locomotion related tasks. At last, a task that reduces the angular momentum in the system is added which greatly helped to improve the walking and also lifting performance, especially in the experiments with the robot.

4.3.1 Contact Point RCF controller

The position of at least three points on a body need to be controlled such that the orientation of the body is also controlled. The points must not be positioned on a line such that all three rotation degrees of freedom are constrained. It is chosen in this work to control the foot's position and orientation using four RCF position control tasks for each of the corner points of the foot. One foot acceleration task is added to the controller as it also constrains the acceleration of the corner points. This approach has the advantage of its simple extendability to arbitrary (contact) points on the robot such that all tracking tasks in the whole body control might even be carried out by only cartesian relative tasks. Also it is expected that the corner point RCF approach is more robust against tipping over when half of the foot is standing over an edge, e.g. on stair steps. The four relative position tasks give four three-dimensional task forces with corresponding task mappings. The 12-dimensional stack of task forces is redundant as it only constrains six degrees of freedom. The tasks are non-conflicting such that the resulting underactuation does not lead to conflict between the tasks when solving for the commanded torque.

The contact point RCF task is implemented equally to the relative center of mass to foot task. The equations will not be repeated, as only the foot task space quantities are replaced by the contact point quantities which will be derived. The relative positions of the corner points and the contact forces are given with respect to to the foot center ${}^{F}_{b}x_{F,cp}$ and will be denoted as $x_{F,cp}$ for simplicity.

Foot Corner Jacobians

The foot corner Jacobian J_{cp} is given by

$$J_{cp} = A_\alpha \cdot J_{cp} \tag{4.36}$$

with

$$A_\alpha = \left[I_{3x3} - R_F \left[x_{F,cp} \times \right] R_F^T \right] \tag{4.37}$$

I_{3x3} represents a three by three identity matrix. The skew operator $[v\times]$ denotes the matrix that corresponds to the cross product of a vector v with another vector as in [13].

$$[v\times] = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{4.38}$$

The time derivative of the contact point jacobian is given by

$$\dot{J}_{cp} = B\, J_{cp} + A_\alpha\, \dot{J}_F \tag{4.39}$$

with

$$B = \dot{R}_F [x_{F,cp}\times] R_F^T + R_F [x_{F,cp}\times] \dot{R}_F^T \tag{4.40}$$

and time derivative of the contact orientation matrix

$$\dot{R}_F = [\omega_F \times]\, R_F \tag{4.41}$$

Corner Point State

The corner point position is derived from the homogeneous transformation of the foot and the relative corner point position

$$\begin{aligned} x_{cp} &= H_F \, x_{F,cp} \\ &= R_F \, x_{F,cp} + x_F \end{aligned} \tag{4.42}$$

The corner point velocity is calculated with the corner point jacobian

$$\dot{x}_{cp} = J_{cp} \, \dot{q} \tag{4.43}$$

Corner Point References

Also the foot's reference has to be transformed. The position reference is given as the position state.

$$x_{cp,ref} = R_{F,ref} \, x_{F,cp} + x_{F,ref} \tag{4.44}$$

The velocity reference is calculated from the time derivative and not using the jacobian as the velocity state

$$\dot{x}_{cp,ref} = \dot{R}_{F,ref} \, x_{F,cp} + \dot{x}_{F,ref} \tag{4.45}$$

The same is performed for the acceleration reference

$$\ddot{x}_{cp,ref} = \ddot{R}_{F,ref} \, x_{F,cp} + \ddot{x}_{F,ref} \tag{4.46}$$

with the time derivate of the rotation matrix defined by the angular velocity reference of the foot $\omega_{F,ref}$

$$\dot{R}_{cp,ref} = \left[\omega_{F,ref} \times\right] R_{F,ref} \tag{4.47}$$

and the second time derivative by the angular acceleration reference $\dot{\omega}_{F,ref}$

$$\ddot{R}_{cp,ref} = \left[\dot{\omega}_{F,ref} \times\right] R_{F,ref} + \left[\omega_{F,ref} \times\right] \dot{R}_{F,ref} \tag{4.48}$$

4.3.2 Orientation Task

The orientation control task is derived as a MPTC task that tracks a given orientation reference used for the upper body and also for the foot as an alternative to the contact point approach. The reference and the actual orientation are given as quaternions. Other orientation representations like euler angles or rotation matrices can be transformed to quaternions. A quaternion consists of a vector part η that defines a 3D vector r in the reference frame. The target frame is obtained by rotation of the reference frame by an angle θ about the defined vector. Additionally a quaternion consists of a scalar part η which is used to normalize the four dimensional quaternion to unit length. The quaternion is given by

$$\zeta = \begin{bmatrix} \eta \\ \epsilon \end{bmatrix} = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ r \sin(\frac{\theta}{2}) \end{bmatrix} \tag{4.49}$$

The error quaternion $\tilde{\zeta}$ between the state and the reference quaternion is given by the quaternion that rotates the state quaternion ζ onto the reference quaternion ζ_{ref}. This error quaternion is found by quaternion multiplication of the conjugate reference quaternion ζ^*_{ref} with the state quaternion

$$\tilde{\zeta} = \zeta^*_{ref} \circ \zeta \tag{4.50}$$

The conjugate quaternion ζ^* is given by

$$\zeta^* = \begin{bmatrix} \eta \\ -\epsilon \end{bmatrix} \tag{4.51}$$

The quaternion product of two quaternions q and p can be carried out by transforming the first quaternion into a quaternion multiplication matrix. This 4 by 4 matrix is multiplied with the second quaternion

$$q \circ p = \begin{bmatrix} \eta & -\epsilon_x & -\epsilon_y & -\epsilon_z \\ \epsilon_x & \eta & -\epsilon_z & \epsilon_y \\ \epsilon_y & \epsilon_z & \eta & -\epsilon_x \\ \epsilon_z & -\epsilon_y & \epsilon_x & \eta \end{bmatrix} p \tag{4.52}$$

The error quaternion is a four dimensional vector with one redundant entry as it only holds information about three degrees of freedom. The vector part of the error quaternion $\tilde{\epsilon}$ is used as the task error $\tilde{x}_k$. The length of the vector part is zero for no error and increases by $\sin\left(\frac{\theta}{2}\right)$ with the error in θ to a maximum of 1 at a 90deg error.

$$\tilde{x}_k = \tilde{\epsilon} \tag{4.53}$$

The task velocity is the angular velocity ω_k of the link in the link's body frame. The task velocity error $\tilde{\dot{x}}_k$ is given by the difference of the angular velocity reference ω_{ref} and the actual one

$$\tilde{\dot{x}}_k = \omega_{k,ref} - \omega_k \tag{4.54}$$

The task acceleration is the time derivative of task velocity, the angular acceleration $\dot{\omega}$ in the link's body frame. The orientation task space is in the body frame of the respective link. To derive the task space quantities from (3.6), the rotational part of the body jacobian of the respective link ${}_bJ_{k,\omega}$ is used.

$$J_k = {}_bJ_{k,\omega} \tag{4.55}$$

The task velocity, the angular velocity, is not the direct time derivative of the task position, the orientation quaternion. For correct gain design, a transformation of the desired task stiffness K_{des} is required. The new desired task stiffness is given as

$$K_{des,\zeta} = 2(\eta K_{des} + [\eta\times]K_{des} + ([\eta\times]K_{des})^T) \tag{4.56}$$

The damping matrix is the calculated as in section 3.5. For the orientation task it is not necessary to build a relative coordinate as the effect of gravity is cancelled out in the task space.

4.3.3 Joint Posture Task

The joint posture task is used to fully define the control problem for joints that are not primarily used for locomotion and to achieve robustness against singular configurations. It has a fix reference position for each joint that the task tries to track. The task weight is chosen to be low such that the free joints are determined but the impact on the joints required for locomotion is low.

In this work, the joint posture task jacobian J_{jp} is the joint selection matrix S. As the task jacobian is constant, the time derivative $\dot{J}_{jp}$ equals zero. The task is then derived as another MPTC task with the joint positions as task positions which are derived from the generalized coordinates q as $x_k = Sq$. The fix reference is the robot's default standing position. The task velocities reference $\dot{x}_{jp,ref} = \dot{q}_{ref}$ and acceleration reference $\ddot{x}_{jp,ref} = \ddot{q}_{ref}$ are zero. Also for this task the gravitational term reduces to zero.

4.3.4 Angular Momentum Task

The angular momentum task is derived as in [13]. The angular momentum l of the robot around its center of mass is calculated with the centroidal momentum matrix A_G from the generalized velocities

$$l = A_G \dot{q} \tag{4.57}$$

For this controller, the reference of the angular momentum and its time derivative are chosen to be zero. The desired change of the momentum is then given by

$$\dot{l}_{des} = \dot{l}_{ref} - k_l \left(l - l_{ref} \right) = -k_l l \tag{4.58}$$

which corresponds to a damping of the centroidal momentum by a positive gain k_l. The linear movement of the center of mass and the change in angular momentum around it is only dependent on the wrench applied by the foot and not the joint torques both for the linear and the angular coordinate. In order to compute the total torque acting around the CoM, the spatial wrench from each endeffector has to be mapped to the center of mass. The mapping of the spatial wrench is found using the adjoint transpose of the homogeneous transformation of the center of mass coordinate frame ${}^0H_{com}$. The adjoint of a homogeneous transformation H with the rotation matrix R and the position vector p is given as

$$Ad_H = \begin{bmatrix} R & [p\times]R \\ 0_{3x3} & R \end{bmatrix} \tag{4.59}$$

The wrench mapped to the center of mass is given as

$${}^{com}_{s}w = Ad_{H_{com}} \sum_{k=1}^{n_{EE}} {}^{0}_{s}w_k \tag{4.60}$$

The desired task force is given by the desired change of the angular momentum

$$f_{des,l} = \dot{l}_{des} = -k_l l \tag{4.61}$$

The task mapping is the angular part of the adjoint of the homogeneous transformation. For the humanoid robot model, the linear contact forces have to be mapped to the spatial wrench first as in (2.56) to obtain the task mapping.

$$T_{U,l} = Ad_{H_{com},\omega} \sum_{k=1}^{n_E E} {}_s A_{\rho,k} \tag{4.62}$$

4.3.5 Regularization

A regularization is added to the optimization problem from (3.24). The regularization is of the form $\frac{1}{2}u^T \Lambda u$ similar to other optimization problems in machine learning and used to penalize large values of the optimized quantity, in this case the joint torques and linear contact forces. It ensures invertibility of the problem and enables the handling of singular positions. Each joint torque and linear contact force can be penalized with one individual value. It has been chosen to have one value for all joint torques and one value for all contact forces. The task is implemented into the cost function (3.24) as a diagonal matrix with the regularization term on the diagonal for the task mapping $T^T WT$ and a zero vector for the force vector $f_{des} WT$.

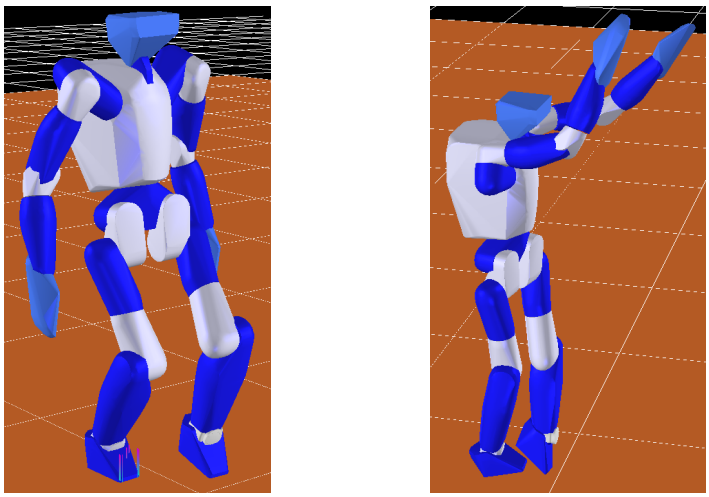
4.4 Simulation Results with the Humanoid Robot Toro

They main focus of this work lied on the relative contact point controller and not the single RCF task with additional orientation control. From time to time, tests with RCF controller in combination with an orientation control task for the foot were carried out but did not lead to noteworthy improvement and will be neglected in the analysis. Figure 4.5 shows the images of the robot in the simulation first for the case where the robot is walking on the ground and second when it is lifted off the ground and floating. The controller setup consists out of four relative CoM-to-contact-point controllers per foot, one foot acceleration task for each foot, an angular momentum task, a joint posture task and a regularization task. The joint posture task is split into three different tasks with individual weights for the upper body, the legs, and the waist. The stiffness can be tuned for each joint individually, but was mainly constant throughout this work. The angular momentum task proved to be a very important factor for success in the experiments and improvements were also observed in the simulation when applying the results from the experiments.

The relative controller analysis is carried out in greater detail for the center of mass approach as described in the next chapter as it showed higher robustness to contact loss of the foot and more time was spent optimizing the controller.

4.4.1 Walking Performance

The simulation started out at the standing position which is equal to the reference of the joint posture task. A trajectory is given for the center of mass and the foot during swing



(a) Toro walking (b) Toro flying

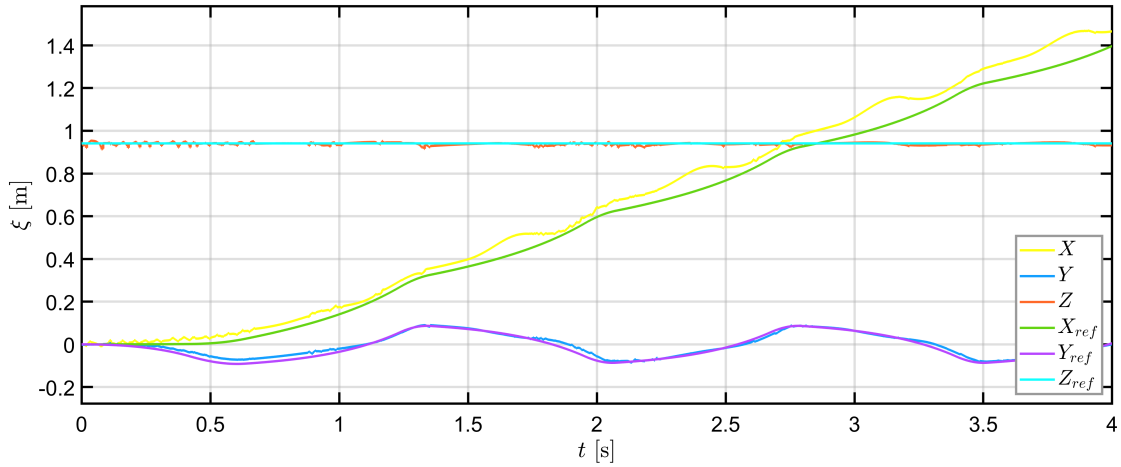
Figure 4.5: Controller behavior in the simulation for the robot Toro

and stance phases. The trajectory contains a contact reliability parameter. The contact reliability is zero for swing phases and one for the stance phases. This contact reliability is multiplied with the task weight for the constraint foot acceleration task such that acceleration of the foot is not blocked by the task during swing phases. The duration of the simulation was most of the time chosen to be four seconds long consisting of four steps, two per foot. Errors and problems mostly added up during the first steps and lead to failure within that time or remained constant after that. The shown controllers stayed stable with similar performance for longer simulation sequences and additional steps.

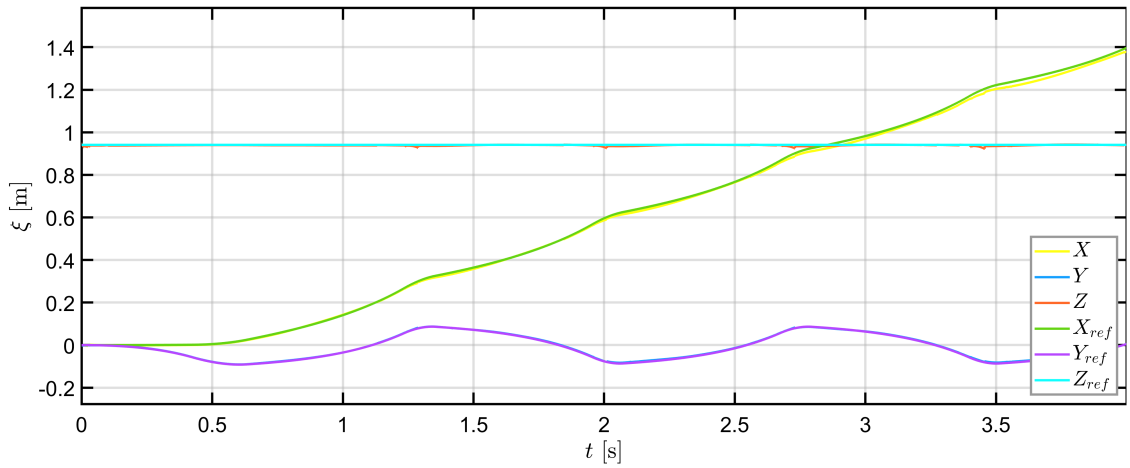
Figure 4.6 shows the tracking of the absolute position of the 3D linear divergent component of motion ξ (DCM) introduced in [13]. The divergent component of motion is defined as

$$\xi = x + b\dot{x} \tag{4.63}$$

with the 3D center of mass position x and velocity $\dot{x}$ and the time constant of the DCM dynamics $b > 0$. The upper plot 4.6a shows the performance with not optimized parameters, while the second plot 4.6b shows the performance after tuning the controller for good walking performance. It can be seen that the tracking in the walking direction diverges from the reference over time. If the RCF task has a constant error due to bad tracking, it adds up over time as there is no tracking of the absolute position, corresponding to a drift in absolute coordinates due to the relative controller formulation. This can be advantageous that it is robust to perturbations that lead to an offset from the target position. The second plot exhibits smaller divergence over time which is smaller due to better tracking. Figure 4.7 shows the tracking of the relative coordinate for one contact point on the foot that performs step two and four. The relative position tracking



(a) Early controller performance



(b) Optimized controller parameters for walking

Figure 4.6: DCM tracking while walking on the ground for differently tuned controller setups

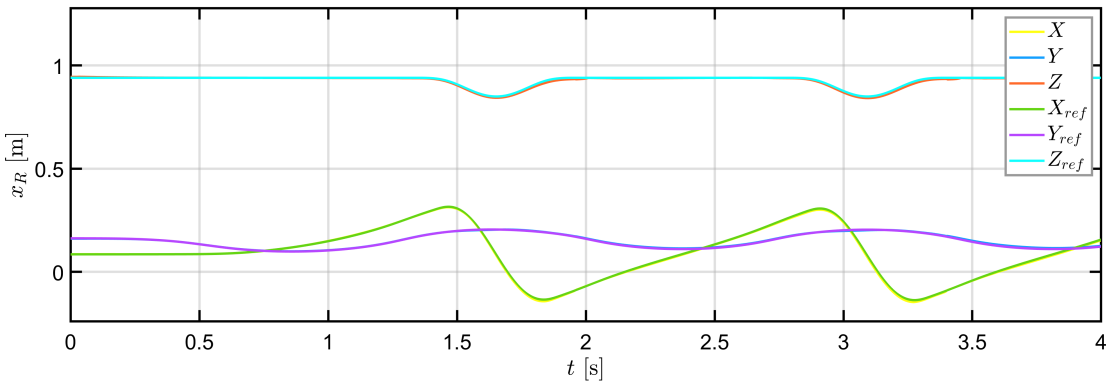


Figure 4.7: Relative coordinate tracking

for the four contact points shows similar behavior with biggest resemblance between the front two contact points and the back two contact points. Very good tracking can be seen and the difference to the DCM plot where small divergence occurred. The impact of the foot to the ground causes the errors occurring at t =1.3, 2, 2.7 and 3.4 s which are especially visible in plot 4.6 in the vertical axes.

Figure 4.8 shows the relative position error from 4.7 and the worse tuned setup. Short instabilities in form of spikes are visible in plot 4.8a for the worse tuned controller. They are especially visible in the first part of the plot with both feet standing, but they reappear after each step when a foot is placed on the ground.

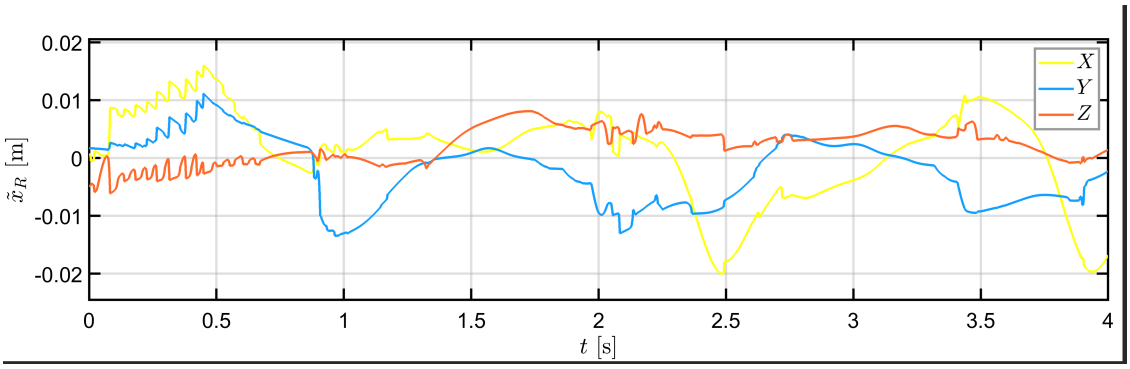
The initial offset in the coordinate originates from an initial 1 mm drop that is necessary for the contact model to ensure the contact points are above the surface. The error converges to zero for the better tuned setup, and stays small while the foot is constrained by the ground, especially compared to the error for the controller in plot 4.8a. The error is larger for the swing phase of the foot.

4.4.2 Examination of Robustness in Case of Contact Loss

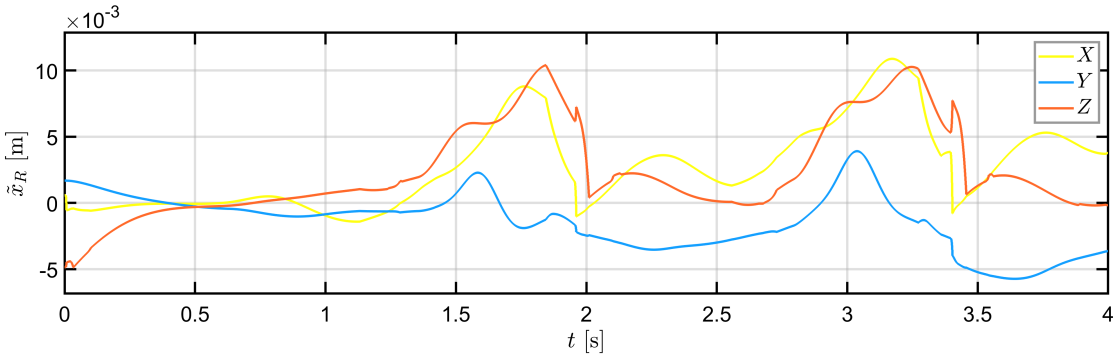
Despite the success of the derived controller in the previous analysis with the SAFF model and the 3D model, it was impossible to find controller parameters that ensured stability during a contact loss. It was tested for free fall and being lifted via the pelvis without contact to the ground. The case of floating is more interesting first, as it is the scenario which will be tested in real experiments where risking a fall is not an option. The lifting is implemented by a wrench w_{lift} that acts at the base of the robot in the hip with the total gravitational force of the robot

$$w_{lift} = \begin{bmatrix} 0 & 0 & mg & 0_{1,3} \end{bmatrix}^T \tag{4.64}$$

The wrench is mapped to the generalized coordinates with the transpose of the hip jacobian J_{hip}^T. (FIX)



(a) Early controller performance



(b) Optimized controller parameters for walking

Figure 4.8: Relative contact point position error for differently tuned controller setups

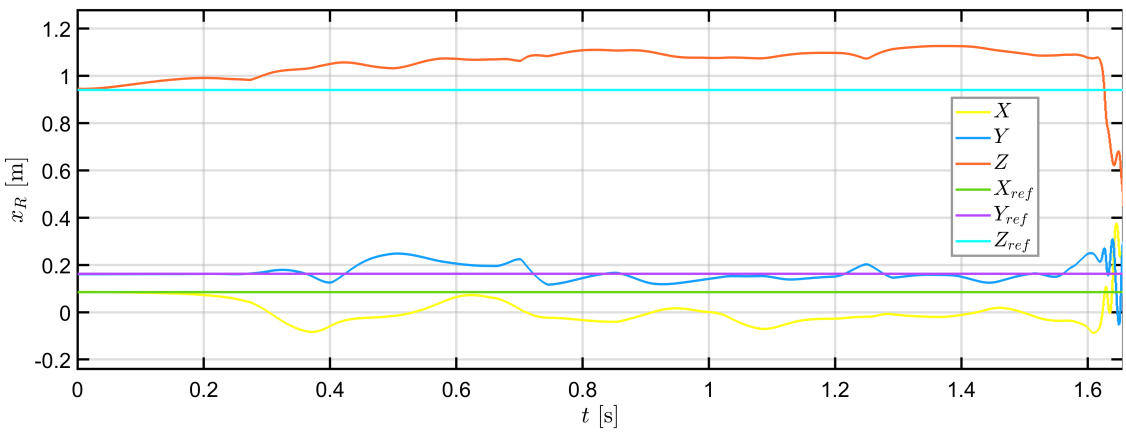


Figure 4.9: Relative coordinate when the robot is lifted resulting in uncontrolled dynamic behavior

The plot in figure 4.9 shows the relative coordinate for a lifted up scenario with a constant reference. The expected behavior would be convergence to a constant symmetric pose of the robot in the air with an offset in the relative coordinate due to the non-existent foot constraint. The stability compared to absolute position inverse dynamics or MPTC approaches has been improved and the system stayed stable for over 1.5 seconds. However the motion of the robot during those 1.5 seconds was not well defined, leading to random motions and greatly depended on the controller parameters. Slightly changed parameters lead to different motions which could result in a quicker instability of the system. The motion can partially be seen in figure 4.5b. There was no convergence to a constant body posture and the symmetry between the left and right limbs was broken quickly. Throughout the 1.5 seconds the arms are lifted higher until they are fully stretched out above the head which is visible in the plot by the increasing error in the vertical direction because the center of mass moves upwards with the arms moving upwards. The stretching out of the entire body leads to a singular position and uncontrolled behavior. This stretching out into a singular configuration seems to be similar to the oscillation and instability occurring in the simple 3D model from section 2.3 when the stiffness is chosen too low. For Toro, it was not possible to resolve the problem with higher stiffnesses and the singular position was not caused by the initial movement after the contact loss rather than uncontrolled slow motions.

5 Relative Controller - CoM Formulation

After unsuccessful tests with the first relative controller setup a different approach had to be found. The foot acceleration task was replaced by a new formulation while the RCF MPTC task unchanged. The new task is designed for perfect tracking of the center of mass in case of a constrained foot.

5.1 Center of Mass Task

The RCF task from 4.2.2 will be used to extract the center of mass behaviour from the RCF task. In the design case of a constrained foot, the center of mass achieves perfect tracking of its reference. If perfect tracking of the RCF task is achieved, the RCF task force error $\tilde{f}_R$ is zero and the task dynamics from 3.19 reduce to the nominal mass-spring-damper system

$$M_R\ddot{\tilde{x}}_R + (C_R + D_R)\tilde{x} + K_R\tilde{x} = 0 \tag{5.1}$$

By expanding the relative acceleration error $\ddot{\tilde{x}}_R$ as in (4.28), the center of mass dynamics for the nominal RCF task dynamics is found

$$M_R\ddot{x}_{com} = M_R(\ddot{x}_{com,ref} - \ddot{x}_{F,ref} + \ddot{x}_{F,est}) + (C_R + D_R)\tilde{x} + K_R\tilde{x} \tag{5.2}$$

By multiplying with mM_R^{-1}, the nominal center of mass dynamics are obtained

$$m\ddot{x}_{com} = m(\ddot{x}_{com,ref} - \ddot{x}_{F,ref} + \ddot{x}_{F,est}) + mM_R^{-1}\left((C_R + D_R)\tilde{x} + K_R\tilde{x}\right) \tag{5.3}$$

The actual center of mass dynamics is given by

$$m\ddot{x}_{com} = {}_sA_{com}\ {}_sw_F - m\bar{g} \tag{5.4}$$

With the two equations the desired wrench for the center of mass can be calculated

$${}_sA_{com}\ {}_sw_F = m(\bar{g} + \ddot{x}_{com,ref} - \ddot{x}_{F,ref} + \ddot{x}_{F,est}) + mM_R^{-1}\left((C_R + D_R)\tilde{x} + K_R\tilde{x}\right) \tag{5.5}$$

As this formulation has been found under the assumption of a constrained foot, the foot acceleration and its reference are assumed to be zero

$$\ddot{x}_F = \ddot{x}_{F,ref} = 0 \tag{5.6}$$

Which gives the final formulation for the task

$${}_sA_{com}\ {}_sw_F = m(\bar{g} + \ddot{x}_{com,ref}) + mM_R^{-1}\left((C_R + D_R)\tilde{x} + K_R\tilde{x}\right) \tag{5.7}$$

This equation shows the desired task force on the right and the mapping from the external force, specifically a spatial wrench, to the center of mass.

$$f_{com,des} = m(\bar{g} + \ddot{x}_{com,ref}) + mM_R^{-1}\left((C_R + D_R)\tilde{x} + K_R\tilde{x}\right) \tag{5.8}$$

$$T_{U,com} = \begin{bmatrix} 0_{3xn_{joints}} & {}_sA_{com,rho} \end{bmatrix} = T_{com}U = M_{com}J_{com}M^{-1}U \tag{5.9}$$

The acceleration of the center of mass only depends on the wrench applied by the foot as in the 1D case. Therefore the mapping of the joint torques is zero and the linear contact forces are mapped by ${}_sA_{com,rho}$ into the task space. The task maps the effect of gravity, the center of mass acceleration reference and relative task forces to the foot wrench. The joint torques are then calculated consistently to the RCF task if nominal foot contact is made.

5.1.1 SAFF Model Analysis

The new center of mass acceleration task was first evaluated with the 1D SAFF model from 2.2 and the task derived for that. The nominal relative task error dynamics for the RCF task are given as

$$m_R\ddot{\tilde{z}}_R + d_R\dot{\tilde{z}}_R + k_R\tilde{z}_R = 0 \tag{5.10}$$

Expanding with the relative coordinate acceleration $\ddot{\tilde{z}}_R = \ddot{\tilde{z}}_{com} - \ddot{\tilde{z}}_{foot}$ and rearranging for the nominal center of mass dynamics $m\ddot{z}_{com}$ gives

$$m\ddot{z}_{com} = m(\ddot{z}_{com,ref} - \ddot{z}_{2,ref} + \ddot{z}_2) + \frac{m_1}{m_2}d_R\dot{\tilde{z}}_R + \frac{m_1}{m_2}k_R\tilde{z}_R \tag{5.11}$$

The actual center of mass dynamics is given from (2.20) as

$$m\ddot{z}_{com} = mg + w_2 \tag{5.12}$$

Using both formulations and the assumption about the foot acceleration and the reference being zero, we find the task formulation

$$m\ddot{z}_{com} = mg + w_2 = m\ddot{z}_{com,ref} + \frac{m_1}{m_2}d_R\dot{\tilde{z}}_R + \frac{m_1}{m_2}k_R\tilde{z}_R \tag{5.13}$$

This equation is rearranged for a desired wrench that is applied to the ground.

$$w_{foot,des} = m(\ddot{z}_{com,ref} - g) + \frac{m_1}{m_2}d_R\dot{\tilde{z}}_R + \frac{m_1}{m_2}k_R\tilde{z}_R \tag{5.14}$$

The control law is found by setting the desired task force from 4.29 FIX equal to the actual task force

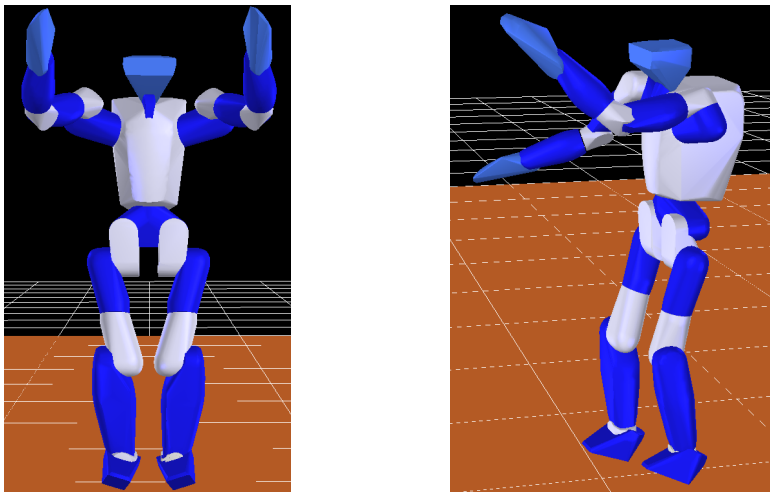
$$f_R = -\frac{m}{m_1}\tau_j - w_2 \overset{!}{=} \frac{m_2}{m_1}m\ddot{x}_{R,ref} + d_R\dot{\tilde{x}}_R + k_R\tilde{x}_R = f_{R,des} \tag{5.15}$$

The desired foot wrench from (5.14) is used for the foot wrench $w_2 = w_{2,des}$ which holds for the nominal constrained foot case. By replacing the joint torque τ_j with the commanded joint torque $\tau_{j,cmd}$ and rearrange for it, the control law is found

$$\tau_{j,cmd} = m_1(g - \ddot{z}_{com,ref}) - m_2\ddot{z}_{R,ref} - \frac{m_1}{m_2}d_R\tilde{z}_R + \frac{m_1}{m_2}k_R\tilde{z}_R \tag{5.16}$$

As in section 4.1.3, the commanded torque could have been found using two task formulations as it is being done for the 3D controller. By summarizing the two body masses with the total mass $m = m_1 + m_2$, it can be seen that the commanded torque from the RCF controller with the constraint foot acceleration task is equal to the RCF controller with the center of mass task from this section. The 1D results are therefore equal to the ones discussed in section 4.1.4 and will not be repeated here.

5.1.2 Three Joint Model Analysis



(a) Stance (b) Walking motion

Figure 5.1: Controller behavior lifted up for the center of mass approach

The 3D RCF controller setup with the center of mass task is analyzed with the three joint, six dof model from section 2.3. It was hoped to see differences to the previous approach in this simple 3D-model to explain the differences for the humanoid robot, but it exhibited no difference as it was the case for 1D. The only differences were visible close to the singular position of the robot due to numerical reasons and no advantage of one approach or the other was apparent.

5.2 Analysis of Robustness against Contact Loss with the Humanoid Robot Toro

The difference between the approaches became apparent when analyzing the center of mass task plus relative controller task in the simulation for the humanoid robot. The controller was tested for different scenarios:

- standing and walking on the ground
- standing and walking while lifted of the ground
- landing after being lifted of the ground in stance and in motion

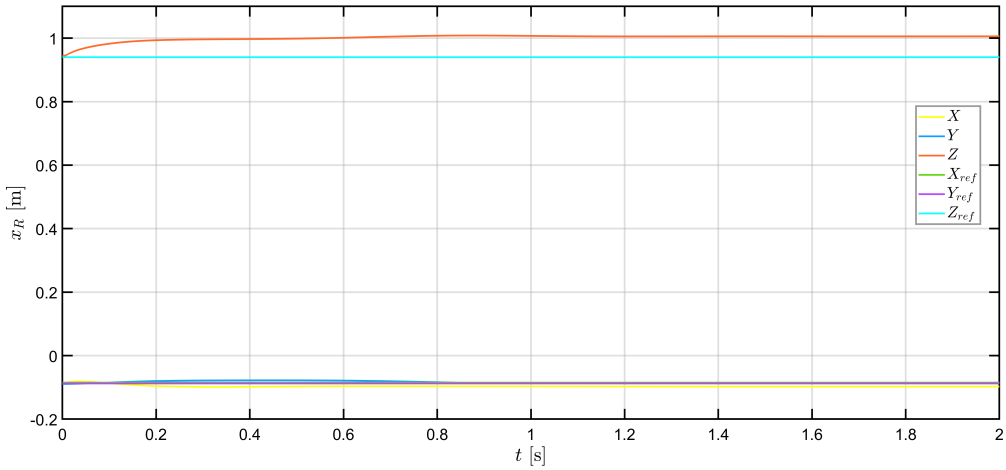
The performance for walking and standing on the ground will be analysed in the next section as it proved to be more difficult to achieve stable performance. Stability when lifted up or falling was possible for a wide range of controller parameters and it showed general stability to different trajectories for the walking motion in the air or repeated lifts and drops in one simulation. To find a controller setting with good performance for motion on the ground as in the air was more challenging. The plots in this section for contact loss and in the next section for walking were created with one controller setup that enables both motion in the air and on the ground without change of parameters. The controller consists of the following tasks:

- Contact point to center of mass relative task
- Center of mass task
- Joint position task
- Angular momentum task
- Regularization

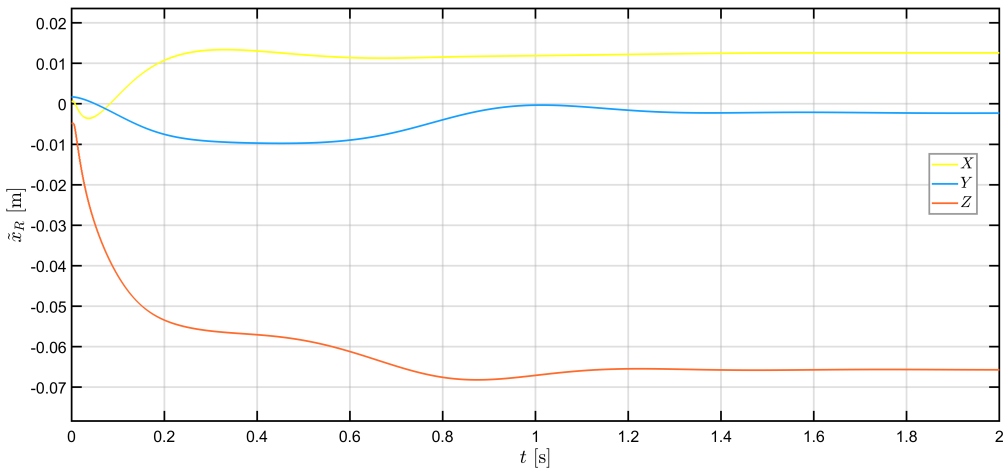
This were the essential tasks for good performance with and without foot contact. Throughout the work, other tasks were considered as well. Especially a task for tracking the upper body orientation was useful for improving the walking performance on the ground but had difficulties with robustness when being lifted up.

Standing without foot-ground contact

Figure 5.2 shows the relative coordinate for the stance scenario in which the controller failed in the previous chapter. Unlike before, the coordinate converges to a constant offset in a stable position as it was expected from the analysis with the simpler models. The converged robot pose can be seen in figure 5.1a. The error is the largest for the vertical axis in which gravity acts but coupling between the axes is given for this complex robot.

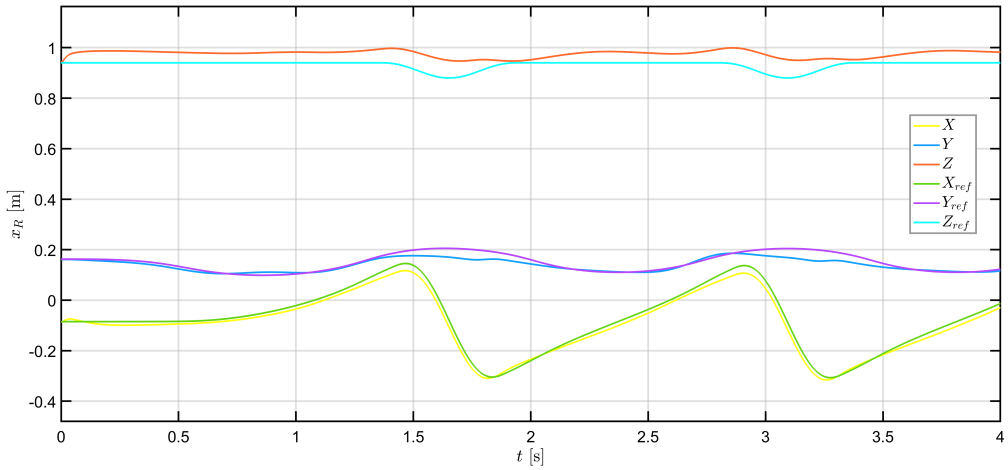


(a) Relative center of mass to contact point position

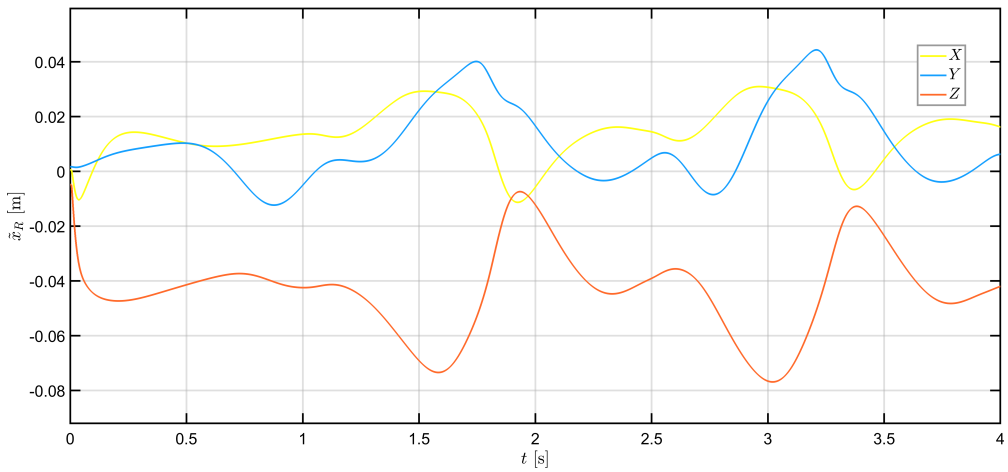


(b) Relative center of mass to contact point position error

Figure 5.2: Controller performance in the relative coordinate for a constant trajectory without foot-ground contact



(a) Relative center of mass to contact point position

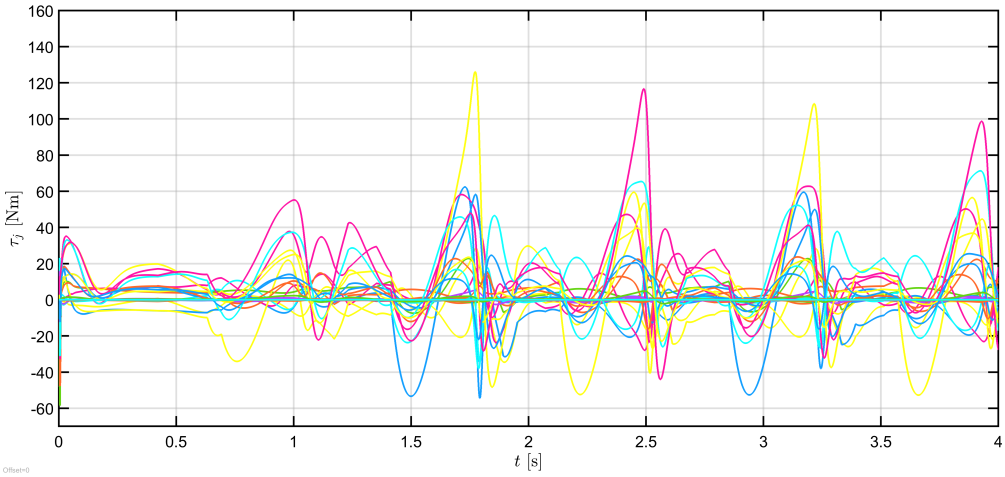


(b) Relative center of mass to contact point position error

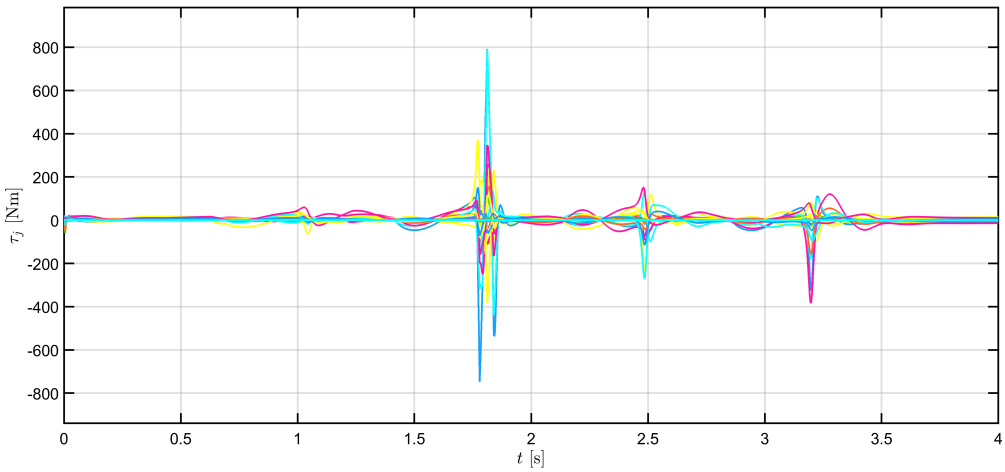
Figure 5.3: Controller performance in the relative coordinate for a walking trajectory without foot-ground contact

Walking without foot-ground contact

The controller was then evaluated for motion without foot-ground contact. The same walking motion from the previous section was used again for that evaluation. Figure 5.3 shows the relative coordinate and its error. Two steps of the respective foot are visible in the plot. The error is very similar for both steps and stayed constant for further steps. The asymmetry of the walking motions introduces rotary movements which have to be compensated by the arms as there no external force can be applied by the robot. The compensation is achieved mostly in form of arm movements. The joint posture task has to constrain the motion to avoid collision between bodies.



(a) Joint torques with tuned controller setup



(b) Large joint torques due to proximity to singular leg position

Figure 5.4: Joint torques of the controller without foot-ground contact during a walking motion

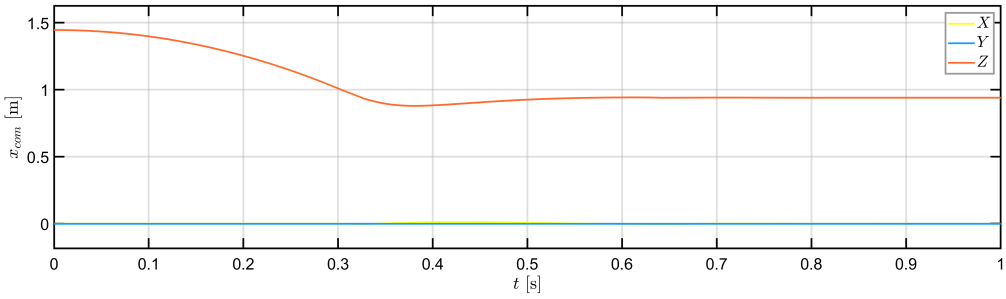
Figure 5.4 shows the joint torques for the walking motion in the air. The first figure is for the well tuned controller from the plots before. It shows smooth torques with one initial spike at the beginning when the initial convergence to the offset occurs which stays within the range of the rest of the walking motion. Afterwards multiple higher torques are visible when a step is performed. Due to the expected counter force from the ground, a torque is applied in the legs similar to the one when foot-ground contact exists. As the counter force is missing, a joint torque opposing the previous peak is visible and corrects the error by the too large joint torque. The second figure shows the joint torque behaviour when the robot comes close to a singular position in the leg when it is too stretched out. This can be achieved by lower stiffness in the RCF task as discussed for the simpler 3D case. The joint torques become excessively large and also potentially dangerous torques even if they don't lead to an overall instability of the system as in this case. The magnitude of the error is not only dependent on the stiffness of the relative task but also on the joint posture task. The joint posture task tries to move the robot away from the singular position back to the initial position. By increasing the stiffness or the task weight of the joint posture task for the legs, singular positions can be avoided. It showed that having a lower stiffness in the relative task was more robust on impacts due to falls and increasing the task weight for the joint posture of the legs led to stable dynamics in the air. Increasing the task weight of the joint posture task is only possible to a certain extend as it also inhibits motion on the ground.

Falling and landing on the ground

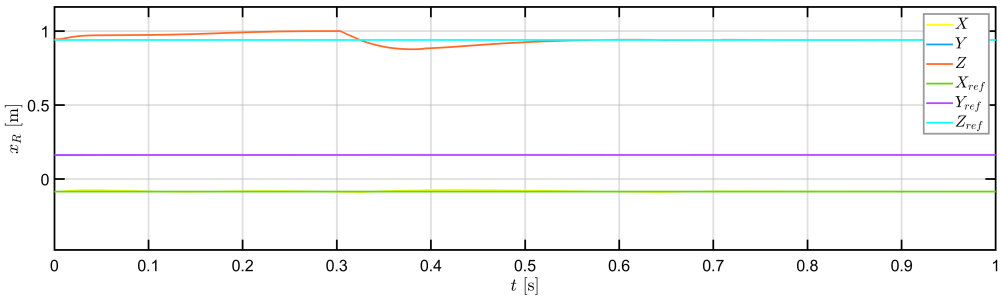
After the successful simulation in the air the robustness to falls with an impact on ground was looked at, first for the behaviour in stance and then for the walking motion. For both cases no changes to the controller and the trajectory have been made and the controller is not aware of existence or timing of the impact. Figure 5.5 shows the absolute center of mass and relative coordinate for a fall from a height of 50 cm to the ground. During the fall, the error is caused by the overstretching of the legs and then recovers and the sign of the error changes as the robot bends the knees to compensate the impact. This recovery can be seen in figure 5.6a.

The controller also proved to be robust to falls in the middle of the walking motion. Figure 5.7 shows the absolute center of mass and the relative coordinate for the walking motion with a fall from 30 cm. The drop occurred during the second step and the robot landed on one foot first. The timing of the landing became important for higher falls and recovery from the impact was only possible if the robot lands on both feet. Landing on one foot induces angular momentum into the system which became too large to compensate with larger heights. In the image in figure 5.6b it can be seen how the angular momentum is compensated by the left arm after the impact.

The plot in figure 5.7a shows no change in the absolute center of mass position (except a small drift in the vertical axis) while the robot is lifted up and then follows the walking motion once it has landed. In the lower plot 5.7b the relative coordinate follows the trajectory in the air and on the ground. The difference in the quality of the tracking is

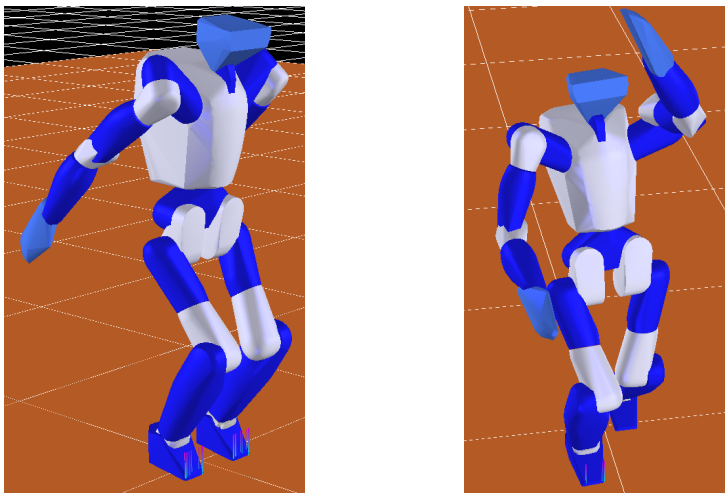


(a) Absolute center of mass position



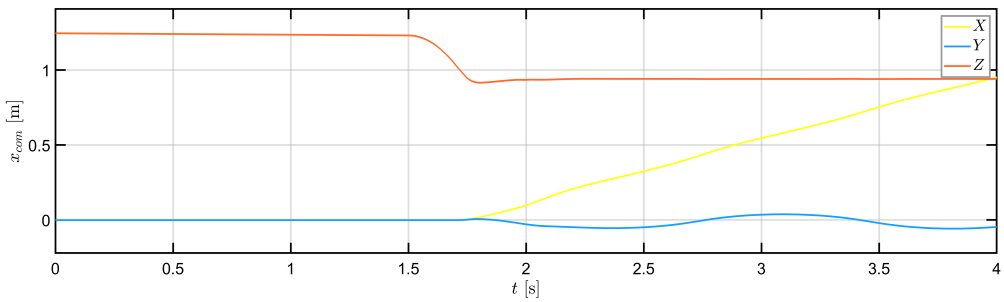
(b) Relative center of mass to contact point position

Figure 5.5: Controller performance when the robot is falling 0.5 m to the ground

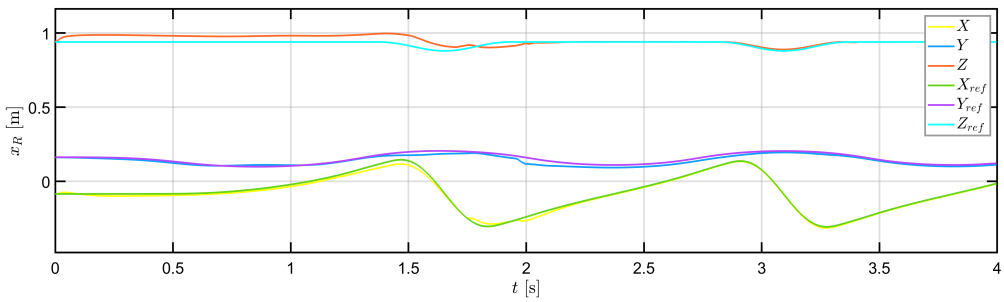


(a) Stance (b) Walking motion

Figure 5.6: Recovery from a fall to the ground from a height of 0.5 and 0.3 m for stance and walking



(a) Absolute center of mass position



(b) Relative center of mass to contact point position

Figure 5.7: Controller performance during a walking motion when the robot is falling 0.3 m to the ground

visible between the first part without contact and after the impact in the second half. Note how the angular momentum by the landing on one foot leads to an error in the y-axis perpendicular to the walking motion.

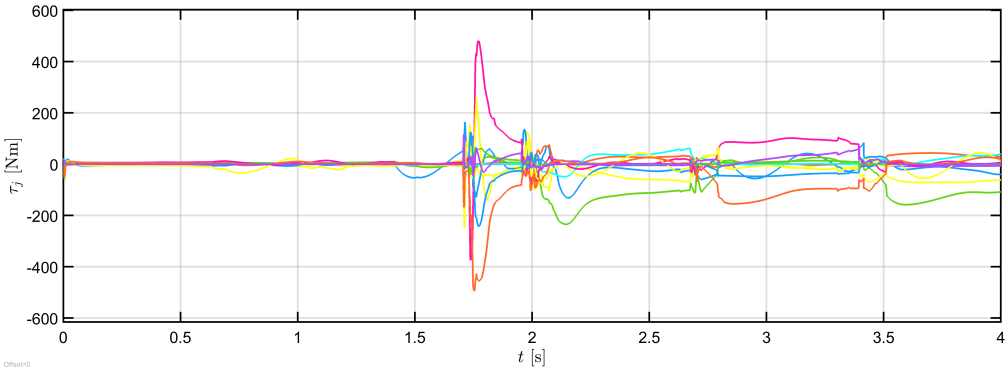
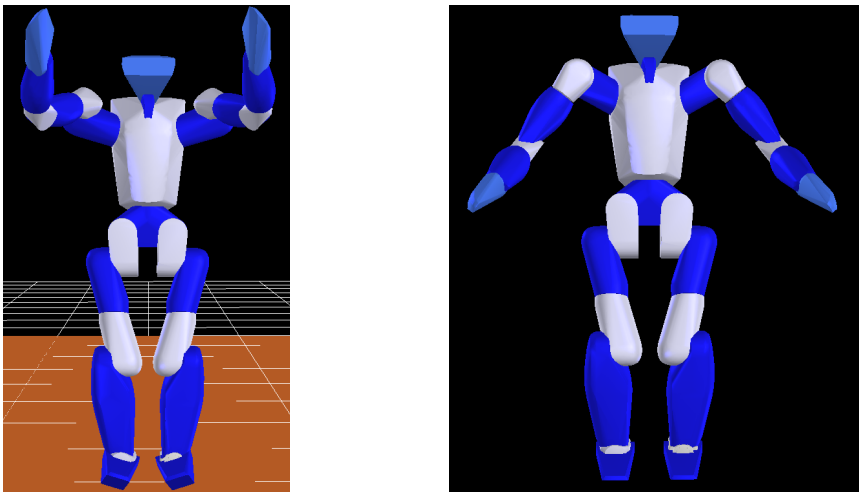


Figure 5.8: Joint torques of the legs during a walking motion when the robot is falling 0.3 m to the ground

The plot in figure 5.8 shows the joint torques of the legs for the walking drop. The large spikes are caused by the impact on the ground. They are larger than the real joint torque limits of the robot Toro but have been disabled for the simulation to inspect the controllers behaviour in these scenarios. For tests with falls from that height changes to the mechanics and actuators would be necessary.

Difference between free fall and lifted floating position



(a) Floating (b) Falling

Figure 5.9: Difference in the stabilized body pose with constant reference

The controller proved to be stable both during free fall and when the robot was lifted up by a force. Figure 5.9 shows the stabilized position of the robot after the initial movement. The pose is very similar to the initial standing pose during free fall. For the floating case with the external force, the pose has diverged a lot. When deriving the RCF task in section 4.2.2, it has been shown that the effect of gravity is cancelled out and which holds for the free fall. If the external force to hold the robot's weight is applied, full gravitational force acts on all links and as an additional disturbance to the controller. Also the impact on the orientation tracking of the foot is large. This impact was smaller for the approach with the single RCF task for the foot with the orientation control task.

5.3 Walking Analysis - Short Instabilities

The developed controller was also overall stable for standing on the ground and following the walking trajectory. Finding a configuration which performed well in most of the above test scenarios for contact loss and was also able to walk on the ground was relatively easy. It was a bit more complicated to find a suitable controller parameters for the fall in motion onto the ground. The biggest issue however, were short instabilities with large torque spikes in the actuators. They did not affect the stability in terms of failure of the controller and trajectory following was possible for a big range of controller parameters.

Figure (5.10) shows the divergent component of motion tracking for the familiar four second walking trajectory. The disturbance is visible in all three axes. They already occur in the beginning when both feet are on the ground and the robot not in motion. The disturbance is reinduced by the impact of the foot to the ground but also appears randomly in the middle of the motion. The lower plot in the figure shows a close up of the vertical axes and it can be seen that the instabilities occur in form of short spikes not in form of periodic oscillations. The spikes result in large joint torques which had to be reduced to be able to test the controller on the real robot.

5.3.1 Passivation

Passivation by Foot Power Limitation

The external forces which increase the system's energy are external forces, the joint torques and the wrench at the foot. The power of the two forces into the system have been investigated.

The power of the foot is given by the product of the velocity and spatial wrench which are given in the spatial frame.

$$P_{foot} = {}_s\dot{x}_{EE}\, {}_s w_{EE} = {}_s\dot{x}_{EE}\, A_{s,\rho}\, \rho \tag{5.17}$$

Figure 5.11 shows the foot power of one foot during the walking motion. Large spikes are visible at the same time as the spikes in the DCM plot. Also they are mostly positive, such that energy is added to the system and may cause the instabilities. The power

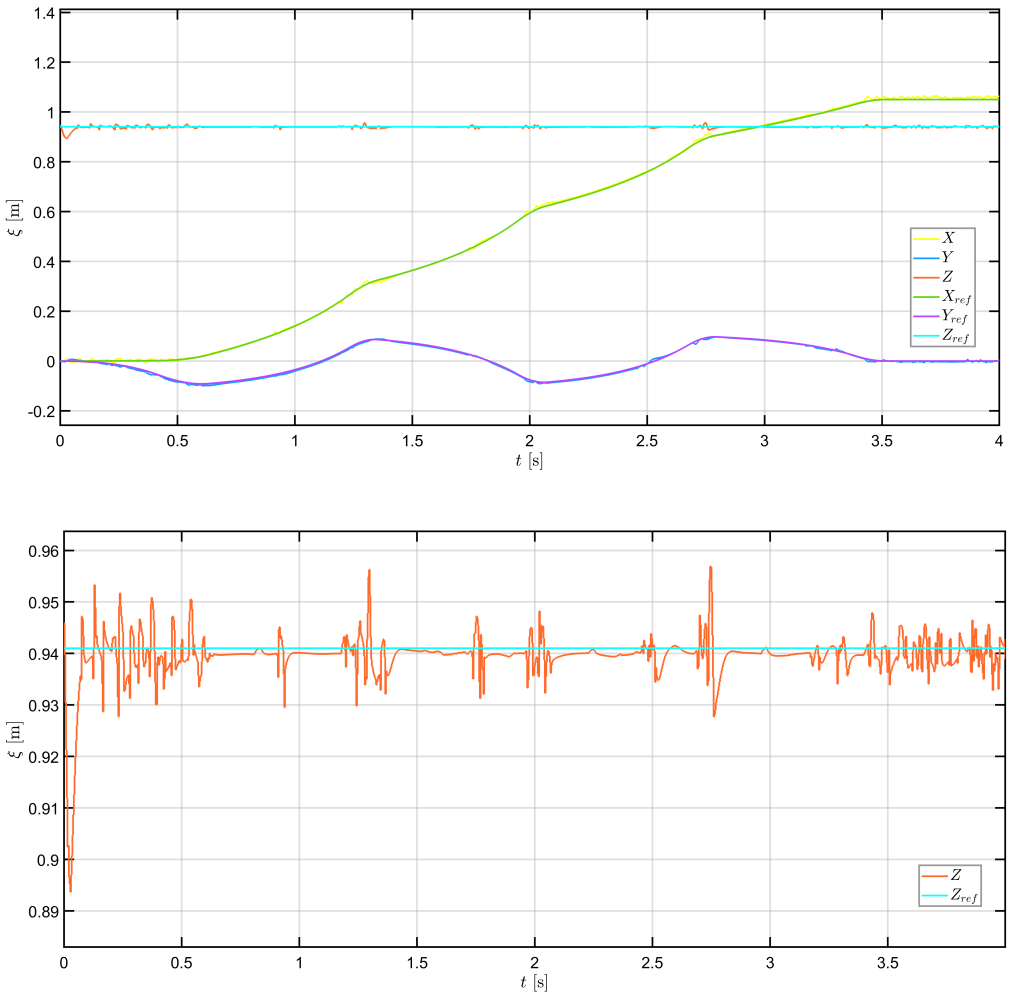


Figure 5.10: Divergent component of motion

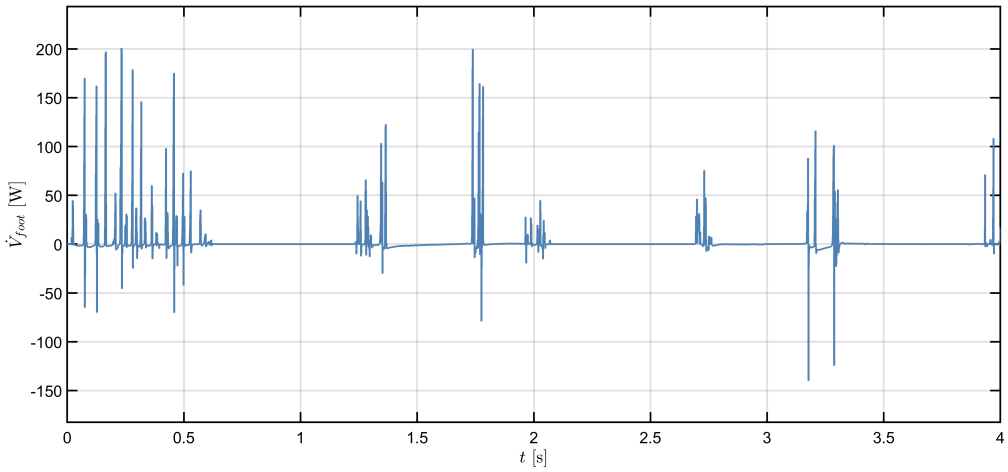


Figure 5.11: Foot power under the influence of short instabilities

exerted by the foot required to perform the walking movement is by a few powers smaller than the spikes such that they are not visible in the plot. The foot power will be limited to a degree where the desired motion is not affected but the large spikes will be removed. The limitation is implemented with a QP-constraint as in (3.22). The linear mapping matrix A maps the actuators τ_j and the linear contact forces to the power. The mapping of the linear contact forces ρ is given in equation (5.17) and the influence of the joint torques is zero.

$$A = {}_s\dot{x}_{EE}^T \left[0_{6xn_{joints}} A_{s,\rho} \right] \tag{5.18}$$

One constraint was added to the optimization for each foot. The bias vector b is zero for this constraint. Figure 5.12 shows the DCM for the vertical axis and a large reduction in spikes. Clipping of the positive spikes at 1 W compared to the 200 W in the plot in 5.11 lead to the largest reductino in spikes. Limiting the foot power in the negative direction positively influenced the behavior too. Reducing the foot power further influenced the motion while walking and especially the ability to recover from falls as in section 5.2 where higher power is required to compensate the impact. Figure 5.13 shows the iterations of the QP-solver during the walking motion that were required to solve the problem while fulfilling the constraints. In the first plot, the power is unlimited and a high number of conflicts is visible during the occurrence of the short instabilities. Adding the additional constraint did not lead to an increase of iterations but rather simplified the problem by suppressing the spikes.

This constraint was implemented using the absolute velocity of the foot and is influenced by larger velocities during fall like the absolute position controllers which is not the goal of this controller. High velocities lead to a high power even with a small foot force. This leads to a smaller allowed forces in the foot during free-fall that the impact of using the absolute velocity is not fatal as for the absolute position controllers

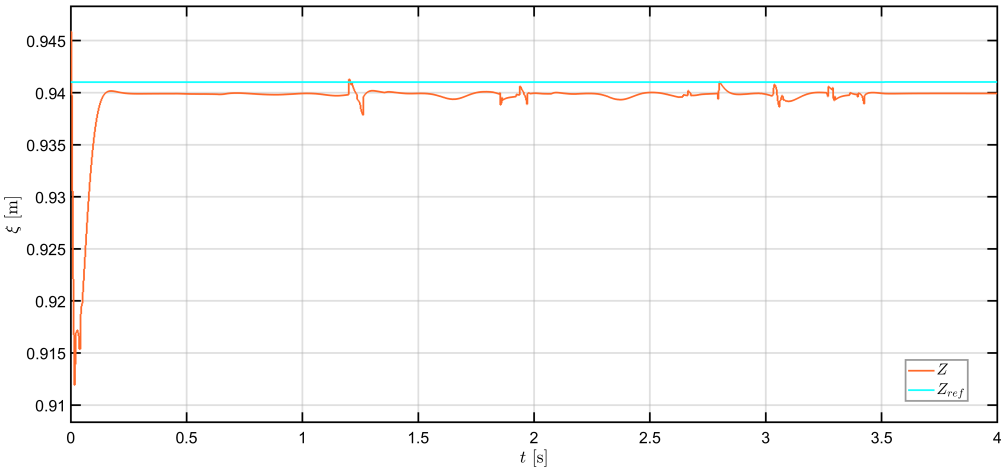


Figure 5.12: Vertical coordinate of the divergent component of motion after limiting the foot power

where high absolute velocities lead to high forces and torques.

The spikes were also visible when mapping the foot force into the relative coordinate task space and calculating the power using the relative velocity. The impact of this constraint was significantly smaller than with the absolute foot velocity. It was also tested to constrain the power in single Cartesian directions of the foot. Especially the vertical axis was expected to be the main cause of the spikes, but constraints had very little effect compared to the total foot power constraint.

Passivation by Joint Power Limitation

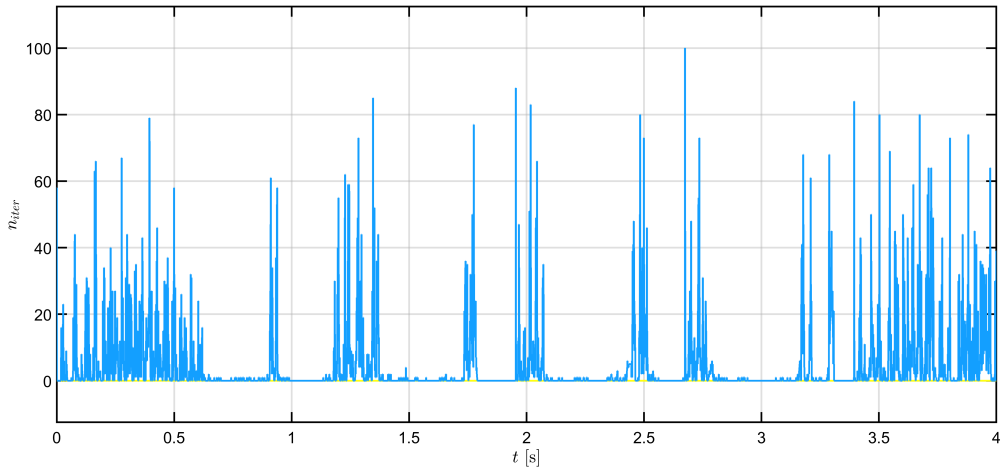
Figure 5.14a shows the power exerted by the joints into the system. The total joint power is given by the sum of the joint torques multiplied with the joint's velocity

$$P_{joint} = (S\dot{q})^T S\tau \tag{5.19}$$

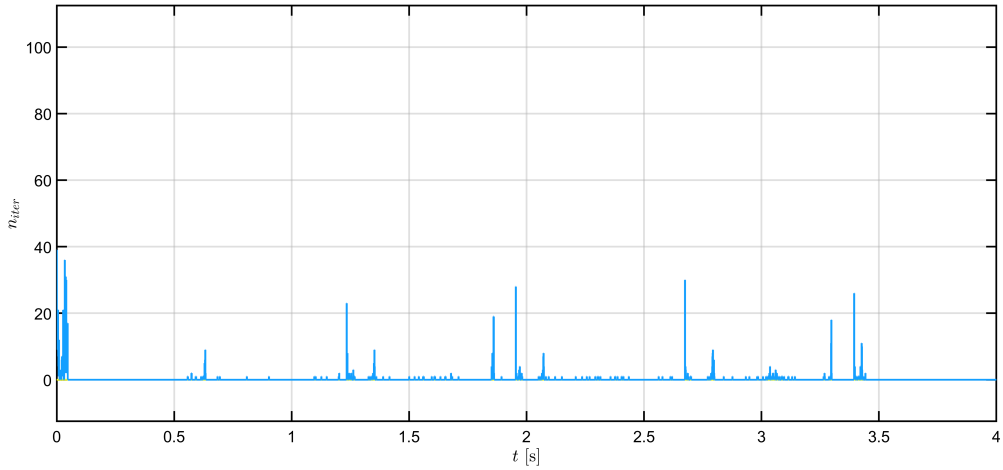
Again spikes are visible but their magnitude is not as large as it was for the foot power and power required for motion is clearly visible. By constraining the joint power outside of the range of the motion the spikes especially in the periods of the movement when standing with both feet on the ground at the beginning and the end of the plot would not be affected. In the plot the difference of the joint power from one control step to the next is shown. It is proportional to the time derivative of the joint power for constant control step intervals it. Here the spikes are visible and a QP constraint was constructed. The mapping is then given with the joint selection matrix S

$$A = (S\dot{q})^T S \tag{5.20}$$

and the bias is the negative joint power from the previous step to form the difference. This limitation did not lead to a clear improvement of the controller and a reduction of

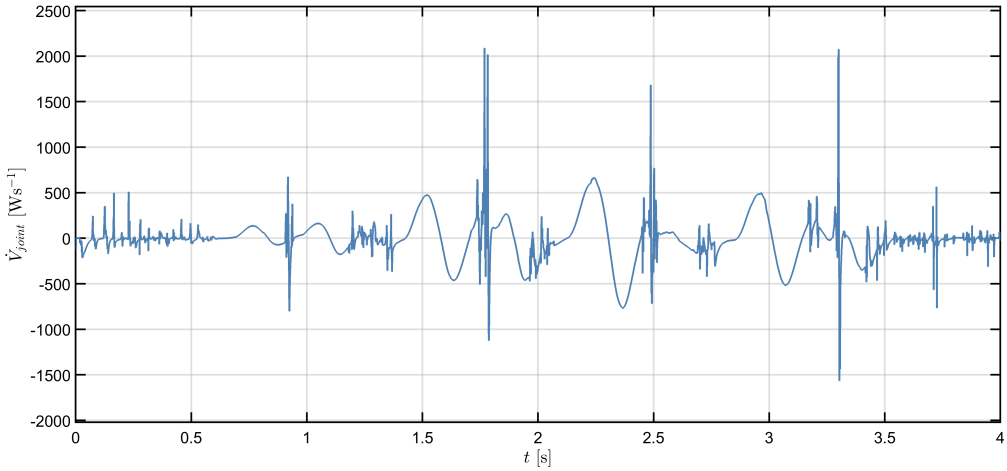


(a) Original controller with short instabilities

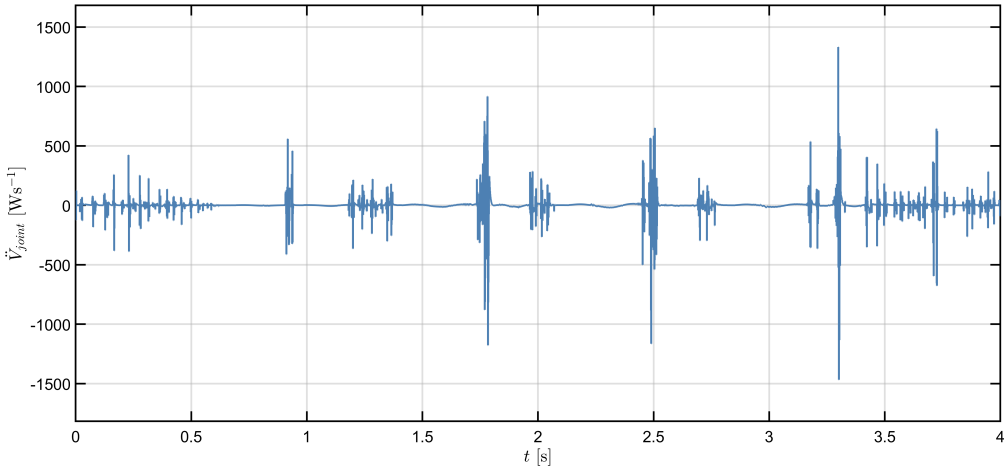


(b) Power limitied

Figure 5.13: QP solver iterations



(a) Joint power



(b) Joint power change

Figure 5.14: Joint power under the influence of short instabilities

spikes without limiting the motion as well.

Passivity Based High Pass Rate Limiter as QP Constraint

The high pass rate limiter (HPRL) introduced in [4] was reformulated as a QP constraint as already the simple linear QP constraint improved the controller. Figure 5.15 shows the original implementation of the high pass rate limiter which consists of a low-pass filter and a rate limiter and aims to reduce the noise and also rapid changes in a signal while minimizing delay. The rate limiting is carried out as a QP constraint.

A signals s from the previous control step is denoted by s^-. The output of the high pass rate limiter s_{HPRL} is given as the sum of the low pass filtered signal s_{lpf} and the rate limited (bounded) high frequency signal $s_{hp,b}$

$$s_{HPRL} = s_{lpf} + s_{hf,b} \tag{5.21}$$

The first order IIR-filter with n filter samples is used as a low pass filter

$$s_{lpf} = \frac{n-1}{n} s_{lpf}^- + \frac{1}{n} s + \tag{5.22}$$

The high frequency part of the signal is then calculated as

$$s_{hf} = s - s_{lpf} \tag{5.23}$$

and the high frequency delta signal as

$$\Delta s_{hf} = s_{hf} - s_{hf,b}^- \tag{5.24}$$

The new bounded high frequency signal is

$$s_{hf,b} = s_{hf,b}^- + \Delta s_{hf,b} \tag{5.25}$$

where simple limits have been used as boundaries instead of a sigmoid limiter in the original implementation to obtain the bounded delta high frequency signal $\Delta s_{hf,b}$

$$s_{min} \leq \Delta s_{hf} \leq s_{max} \tag{5.26}$$

This leads to the filters output signal

$$s_{HPRL} = \frac{n-1}{n} s_{lpf}^- + \frac{1}{n} s + s_{hp,b}^- + \Delta s_{hp,b} \tag{5.27}$$

The limited signal is not available as the limiting is carried out as a QP-constraint. Instead the equation is rearranged for it to obtain an expression for the boundaries.

$$\Delta s_{hp,b} = s_{HPRL} - \frac{n-1}{n} s_{lpf}^- - \frac{1}{n} s - s_{hp,b}^- \tag{5.28}$$

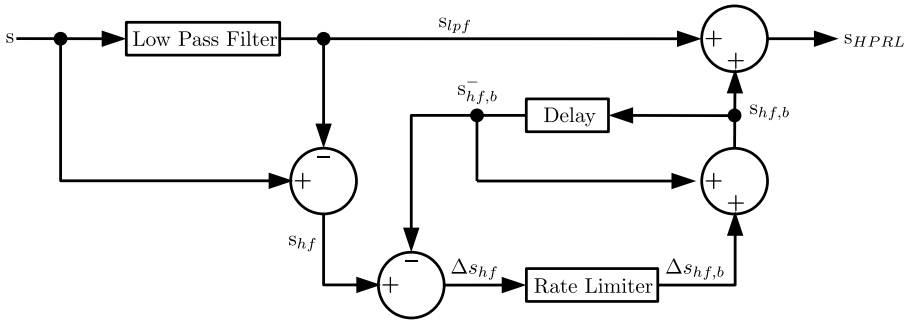


Figure 5.15: High Pass Rate Limiter of a signal s

As the output of the filter s_{HPRL} is not known, an assumption is made that the output of the filter is similar to the actual signal s

$$s_{HPRL} \approx s = s_{lpf} + s_{hf,b} \tag{5.29}$$

using this assumption the equation simplifies to

$$\Delta s_{hp,b} = \frac{n-1}{n} s + \frac{1}{n} s^-_{lpf} - s^- \tag{5.30}$$

Expanding with the limitation from equation 5.26 the boundaries are introduced

$$s_{min} \leq \frac{n-1}{n} s + \frac{1}{n} s^-_{lpf} - s^- \leq s_{max} \tag{5.31}$$

By rearranging for the signal s the lower and upper boundaries are found.

$$\frac{n}{n-1}(s_{min} + s^-) - \frac{1}{n-1} s^-_{lpf} \leq s \leq \frac{n}{n-1}(s_{max} + s^-) - \frac{1}{n-1} s^-_{lpf} \tag{5.32}$$

In contrast to the above limitation where the total joint power was limited, this was used for the signal of individual joints. The rate of the joint torque was limited based on whether the change of the torque leads to an increase or decrease of power. Passivating torque changes are permitted completely while torque changes that lead to higher increase of the energy in the system are limited. Figure 5.16 shows this behavior where for the a positive joint velocity the upper limit is small while the full joint power (normalized to one) is allowed. When the joint velocity becomes negative, negative joint torques are limited. The intermediate phase is determined by a weight $w_{\tau,max}$ given by a sigmoid function based on the motor velocity $\dot{\theta}$.

$$w_{\tau,max} = \frac{1}{1 + \exp(\dot{\theta}/\dot{\theta}_{crit})} \tag{5.33}$$

The weight for the lower bound is found from $w_{\tau,min} = 1 - w_{\tau,max}$. $\dot{\theta}_{crit}$ determines the slope of the function around the zero motor velocity. No significant improvement was made to the controller with this method of passivating the joint power in terms of suppressing the spikes, but it's functionality was demonstrated.

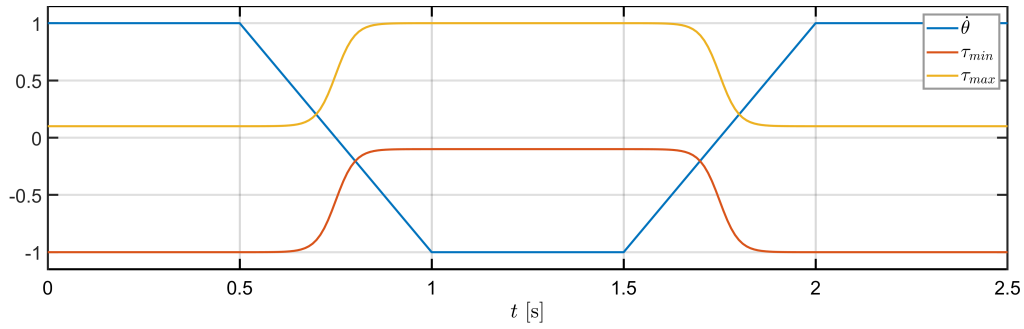


Figure 5.16: Passivity based bounding of the joint torque based on the joint velocity

5.3.2 Alternative Control Tasks

No stable controller behavior was achievable with the measures taken so far. Instead of limiting the power or passivating the system, ideas to modify the existing controller tasks or creating new ones were considered after further analysis of the problem.

Figure 5.17 shows a comparison of the actual and the desired task force for the RCF and the center of mass task during the occurrence of several spikes. The plot uses the task force of one contact point of the left foot in the vertical axis. The plot has high similarity to the other axes and the other contact points of both feet and was chosen as a representative example for the analysis. It can be seen that the actual task force is similar to the desired task force for the relative task during the spikes. For the center of mass task however, the actual task force is diverging from the desired value and opposing the desired change of the task force. This indicates a conflict of the center of mass task with the others during the spikes and that the assumption about the consistency with the relative task is not given. This lead to the idea of modifying the center of mass task.

Reduction of Feedback in the Center of Mass Task

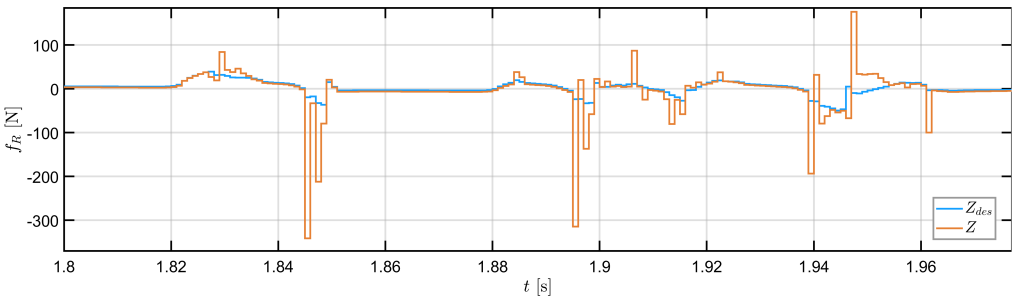
The desired task force of the center of mass task was derived as

$$f_{A,com,des} = m(\bar{g} + \ddot{x}_{com,ref}) + mM_R^{-1}\left((C_R + D_R)\tilde{\dot{x}} + K_R\tilde{x}\right) \tag{5.34}$$

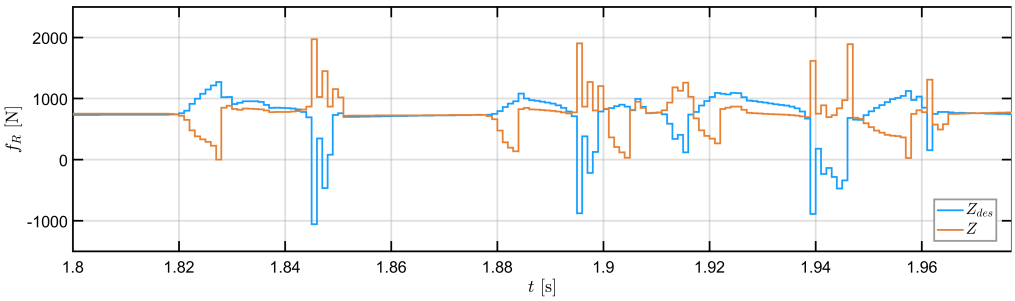
The first part of the task force is the constant gravitational force $m\bar{g}$ on the center of mass and the center of mass acceleration $\ddot{x}_{com,ref}$ which is zero in case of stance reference. The desired task force of the center of mass task was reduced to only those two terms to obtain a plain feed forward task

$$f_{A,com,des} = m(\bar{g} + \ddot{x}_{com,ref}) \tag{5.35}$$

For a constant center of mass acceleration reference (i.e. during stance), this is a constant term and bounded with a bounded center of mass reference for the walking motion. It aims to suppress the effect of the spikes as they have no effect in this task. By having a constant or bounded task, the error without contact is also expected to be constant or



(a) Relative foot to cp task



(b) Center of mass task

Figure 5.17: Desired task force compared to the actual task force in the vertical axis during short instabilities

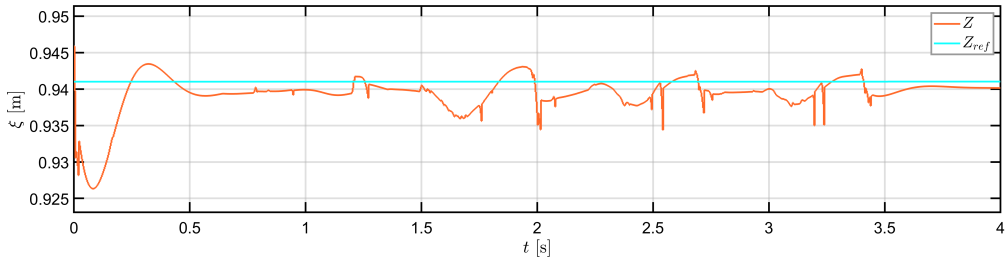


Figure 5.18: Reduction of spikes in the vertical axis due to the removal of feedback in the center of mass task

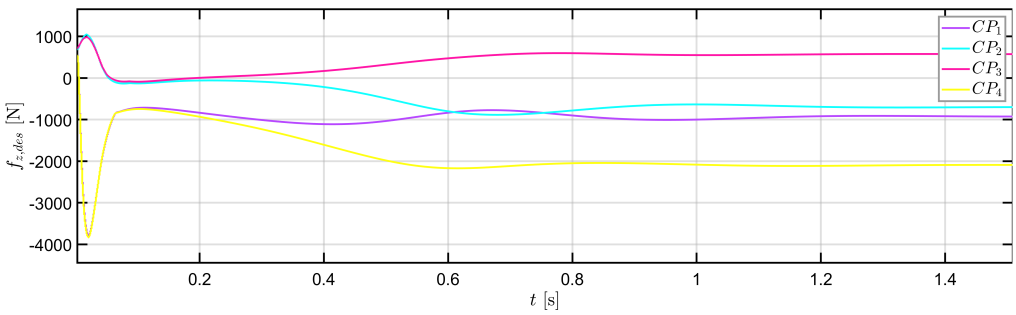


Figure 5.19: Desired vertical task force for the center of mass when lifted off the ground

bounded and not affected by the change of the task in a way that it leads to instability. Figure 5.18 shows the divergent component of motion and a great reduction in spikes in the plot can be seen, although some short instabilities can still be seen. The simulation results for contact loss however, were similar to the constrained foot acceleration task and no stability of the system could be achieved. For the simpler 3-joint robot from section 2.3 the system was stable like for the foot acceleration task. The removal of the feedback removes the mapping of the RCF influence on the center of mass and therefore an incorrect mapping to the linear contact forces. This discrepancy can be clearly seen in the tracking error starting at $t = 1.5\,\mathrm{s}$. For the foot power limitation in 5.12 that tracking error did not occur. It was also tested to keep the Coriolis effects in the desired task force.

$$f_{A,com,des} = m(\bar{g} + \ddot{x}_{com,ref}) + mM_R^{-1}C_R\tilde{x} \tag{5.36}$$

Results were similarly unsuccessful as without the Coriolis effect for contact loss.

Figure 5.19 shows the desired task force of the original center of mass task when lifted up with a constant reference for standing. It can be seen that the desired task force reduces greatly in the beginning and switches the sign when the foot moves downwards due to the lack of the counter force by the ground. For the modified center of mass task without feedback, this term would remain at the starting point of a positive vertical force which would lead to further extension of the leg. Also three of the four desired

vertical task forces are negative after convergence. The difference in task forces is caused by the orientation of the foot that is visible in 5.9a and the therefore different heights of the corner points

Center of Mass MPTC Task

A different idea was to derive the center of mass task not from the error dynamics of the RCF task but as a center of mass MPTC task as in 3.2. The task mapping is equal to the above center of mass task $T_{U,com}$. The desired task force is given as

$$f_{com,des} = J_{com}\tau_g + M_{com}Q_{com}\dot{q} + M_{com}\ddot{x}_{com,ref} + (D_{com} + C_{com})\tilde{\dot{x}}_{com} + K_{com}\tilde{x}_{com} \quad (5.37)$$

It is assumed that for the constrained foot, the center of mass coordinate x_{com} behaves equal to the relative coordinate x_R from section that a new formulation is found

$$f_{com,des} = M_{com}(\bar{g} + \ddot{x}_{com,ref}) + M_{com}Q_{com}\dot{q} + (C_{com} + D_{com})\tilde{\dot{x}}_R + K_{com}\tilde{x}_R \quad (5.38)$$

This equation is similar to the first center of mass with different controller parameters, especially a differently calculated damping matrix D_{com}. The difference of the controller parameters can be seen as a form of transformation of the controller design matrices between the relative and the center of mass space

$$D_{com} = mM_R^{-1}D_R \quad (5.39)$$

It was suspected that the spikes originate in an amplification of controller responses of the relative controller or the foot-ground contact by the term $mM_R^{-1} \geq 1$. This task lead to a reduction in spikes while walking or standing on the ground but again to instability after contact loss. This task also proved to be instable in case of an unconstrained foot for the simple 3D model from section 2.3, showing that this approach leads to conflicts between the relative and the center of mass tasks caused by the missing mass matrix transformation term.

5.3.3 Improvement by Tuning

The previous approaches of improving the controller performance did not lead to a stable result that allowed contact loss with a stable response leaving the original center of mass task formulation as the only one working for both contact loss and walking despite the short instabilities. Figure 5.20 shows the commanded Lyapunov rate of the left foot combined of all RCF tasks. For almost all spikes the Lyapunov rate is negative throughout the spike. This lead to the idea that the commanded joint torques lead not to the expected response of the system as they lead to excitation instead of the expected passivation. A cause for this could be a difference of the foot-ground contact and its model difference between simulation and controller. Figure 5.21 shows the difference of the expected foot acceleration commanded by the controller to the actual acceleration. The actual foot acceleration is calculated as

$$\ddot{x}_{foot,real} = \dot{J}_{foot}\dot{q} + J_{foot}\ddot{q} \quad (5.40)$$

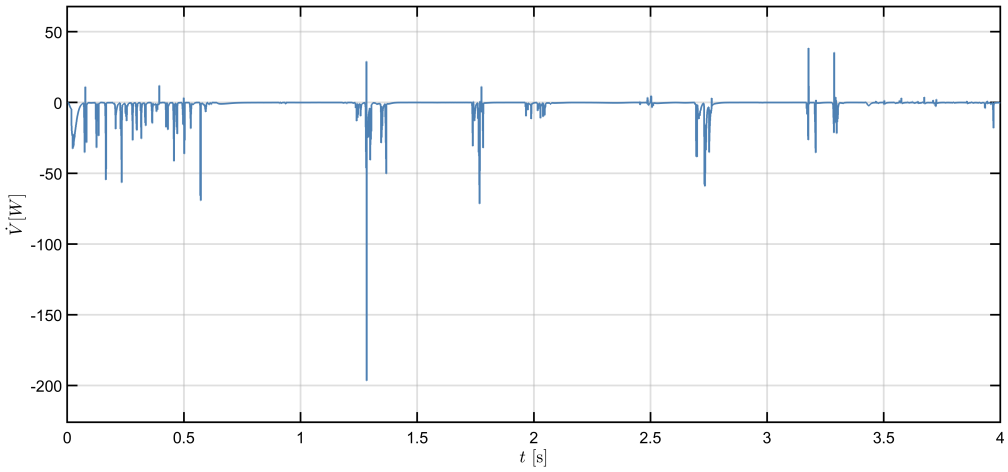


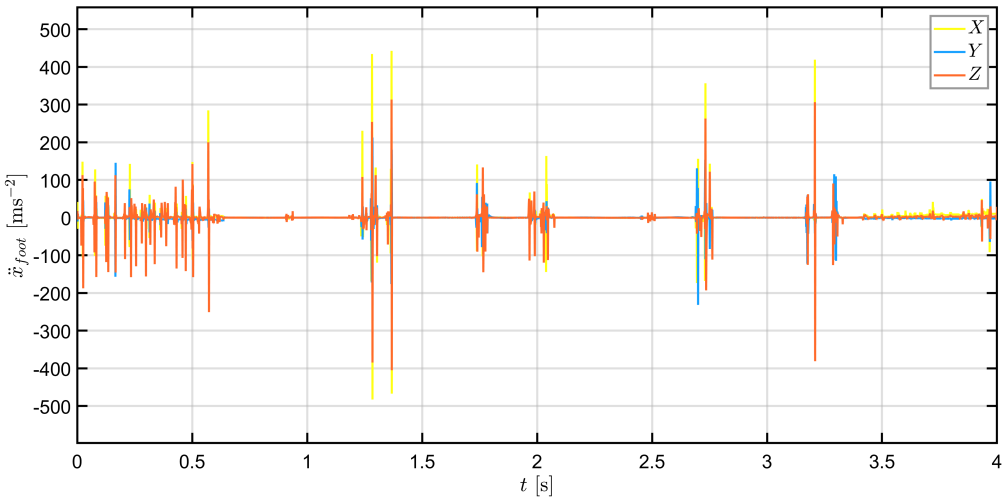
Figure 5.20: Lyapunov rate of the left foot for the walking motion

with the generalized velocity and acceleration given by the OpenHRP simulation where the actual foot wrench is determined. The commanded acceleration is given by

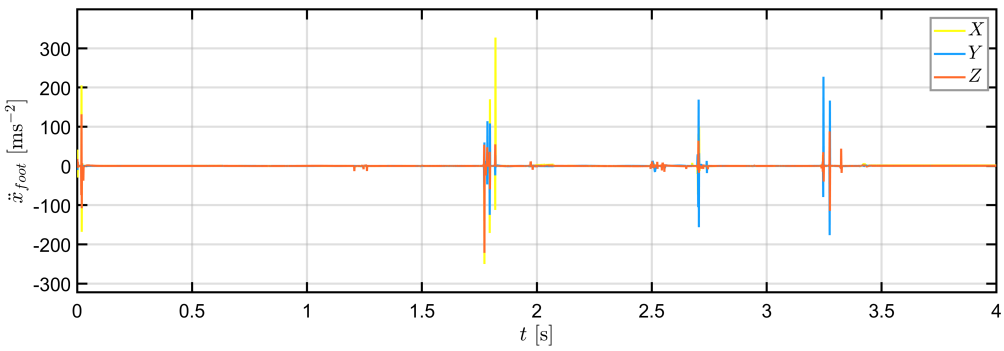
$$\ddot{x}_{foot,cmd} = J_{foot} M^{-1} \tau_{cmd} - \left(J_{foot} M^{-1} (\tau_g + C\dot{q}) - \dot{J}_{foot} \dot{q} \right) \tag{5.41}$$

where τ_{cmd} is given by the actuation from the QP-optimization. For consistent assumptions about the foot constraint, this term should be zero. The plots show large differences occurring during spikes and a reduction of them in the second plot which was created using the above passivation of the foot power. The passivated plot shows that spikes almost only happen during the impact of the feet to the ground and the initial impact after the 1 mm fall. This is explainable as the exact timing of the impact is not modeled perfectly in the controller due to tracking errors and instead a smooth landing is expected. The unexpected response by the foot-ground contact may be the cause of the instabilities and the inconsistency between the tasks. By limiting the foot power, the magnitude of the inconsistency is limited. Therefore also the magnitude of false commanded torques and spikes get suppressed. Why the relative controller would be more sensible to the foot-ground contact than the MPTC controller which also controls the foot position during stance is unclear.

At this point experiments with the six-joint model from section 2.5 was adduced to check for inconsistencies with the different foot constraint and the spikes could not be observed. Further analysis of the contact model was not conducted due to available time for this thesis and improvements by additional tuning. The focus lied on experiments with the real robot for validation. For further research, the contact model of OpenHRP from [17] could be implemented for the simpler models. Additional effort was put into tuning the original version of the controller with the center of mass task and reducing the short instabilities to an extent that the controller can be tested on the real robot. The



(a) Non-passivated



(b) Passivated

Figure 5.21: Difference between the expected and the actual foot acceleration

previously introduced and analysed simpler 3D models were deducted at this point of the thesis. No spikes could be observed. With further analysis of the short instabilities it was not possible to clearly locate the source of the spikes in the derived tasks.

The final setup of control tasks consisted of

- One relative center of mass to contact point task per contact point per foot
- One center of mass task per contact point per foot from 5.1
- Joint posture task with a fix reference
- Angular momentum task
- Regularization task

The influence of each task and the changes which lead to the removal of the short instabilities will be explained in the following.

Regularization Task

The regularization task was primarily used for guaranteeing invertibility in the analytical or QP computation of the actuators. To keep the influence low, the values for both the joint torques and the linear contact forces was chosen to be very low compared to the other tasks. The theory of the cause of the instabilities by the foot-ground contact lead to the idea of using the regularization to penalize large commanded foot wrenches. It was seen above that the positive effect of the power passivation to the short instabilities was most effective when limiting the wrench applied by the foot. The regularization for the linear contact force was increased by factors of 10^3 to 10^5. This change was the major improvement and the other tasks were adapted to achieve robustness after contact loss and good tracking on the ground.

Relative Center of Mass to Contact Point Task

The RCF task was weighted very high compared to the other tasks as it is the main task for locomotion of the robot. Walking was possible for a big range of different stiffness values. A higher stiffness led to better tracking while walking on the ground and also higher robustness against singularities in the legs. The weakness of high stiffness was during the impact of the robot after a fall while in motion. With lower controller gains, recovery was more likely. The goal of the RCF controller is the improvement of robustness for contact loss but also other unforeseen impacts so that the lower stiffness seems desirable and was chosen for the controller.

Center of Mass Task

The center of mass task has the only task weight as a parameter. As it was seen in the previous section, changing the impact of the feedback in the center of mass task improved the stability in terms of the spikes but also reducing the weight had a positive influence. The weight was chosen to be 10% of the weight of the RCF task. Reducing the task weight further led to worse tracking of the walking trajectory.

Joint posture Task

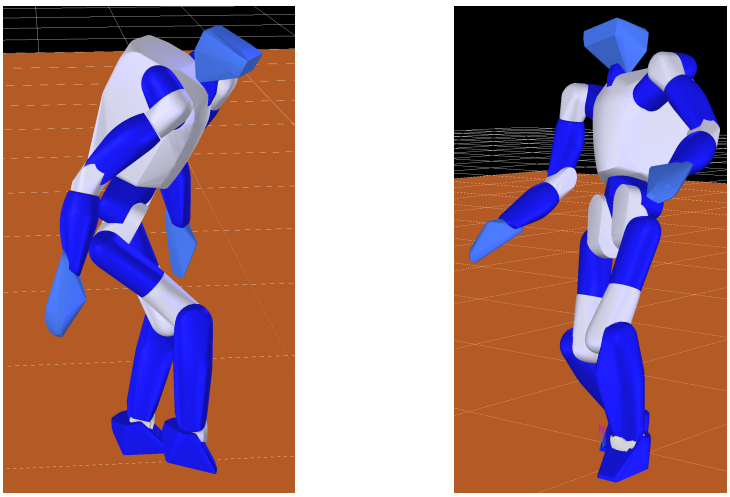
The task weight of the leg posture was increased to compensate for the lower RCF stiffness and avoid singularities during contact loss. Also the hip task weight was increased to avoid strong rotations of the upper body. These rotations occurred during walk but also after the falls in motion to generate required angular momentum. These body rotations led to potentially dangerous configurations of the robot and failure of the controller. By increasing this weight compared to the weight for the arms, angular momentum was compensated stronger by the arms than the entire upper body.

Angular Momentum Task

The angular momentum task was not changed in the tuning effort before the real experiments on the robot. During the experiments, this task proved to be indispensible and an increase of the task gain was necessary. This increase of the damping also had positive effects on the performance in the simulation and the insight of this information was used in the controller that was used for the plot generation.

Upper Body Orientation Task

The controller has no task for control of the orientation of the upper body. For successful walking for longer times an upright posture is essential as mention in the joint posture task. By controlling the upper body orientation with an additional task, the walking performance was increased especially in the beginning of this work when the rest of the controller was not tuned perfectly. This task prevented overstretching of the hip when stepping forward and too strong bending in the hip when moving the center of mass forward. The relative coordinate not only depends on the foot placement but by the center of mass position whose position is influenced by upper body movements. With too low task weights for the angular momentum task and joint posture task this undesirable behavior occurred which can be seen in figure 5.22. The upper body orientation task exhibited its weakness without foot-ground contact. The only way to control the orientation when lifted up or falling is by rotation of limbs specifically the arms as the legs are constrained by the RCF task. Longer time in the air lead to excessive arm movements which resulted in collision and also failure of the controller. Therefore the task was not used in the controller to generate the plots for this analysis. For a real life application this task should be considered as long falls are an unlikely scenario. For



(a) (b)

Figure 5.22: Too strong upper body movements due to bad tuning

short periods of contact loss, controlling the upper body orientation during that time as well might increase the probability of recovery after contact has been regained.

High Pass Rate Limiter

The final version included the original high pass rate limiter from [4]. It was used on the output of the relative and the center of mass task. While not crucial for stability in the simulation, it was necessary in the experiments with the real robot. A slight improvement was also visible in the simulation and it helped removing the short instabilities.

Results

Figure 5.23 shows the number of iterations of the QP-solver for the final controller. It can be seen that the solver conflicts are removed almost entirely except when higher forces occur during the placement of the foot on the ground which is expected and equally existing in the absolute position controllers. Also in the divergent component of motion in figure 5.24, the improvement is clearly visible compared the spikes in 5.10

5.4 Orientation Tracking of the Foot

Figure 5.25 shows the orientation tracking of the presented contact point approach. The foot and its reference tilt during the swing phase of the foot. It can be seen that the desired orientation around the y-axis is not reached perfectly and also errors occur in the other two axes. The error about the vertical stays remains during stance, as the foot

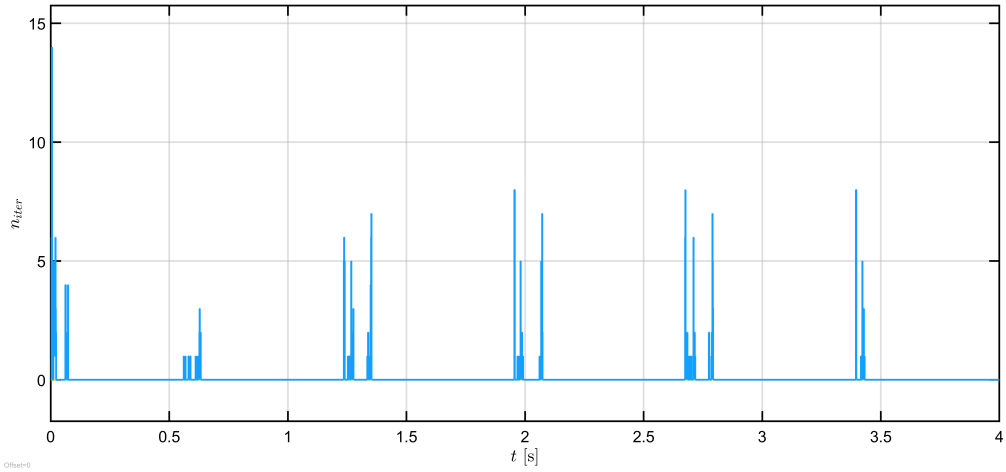


Figure 5.23: QP solver iterations for the presented controller setup

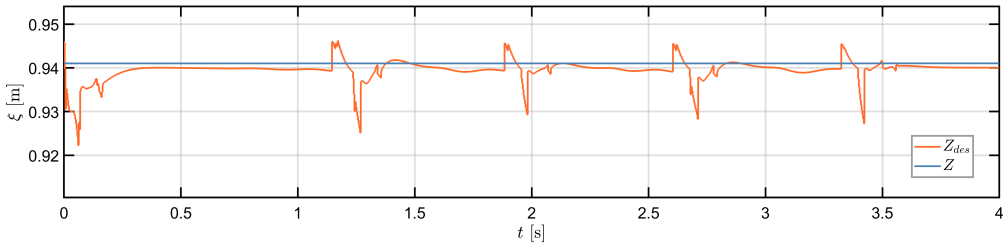


Figure 5.24: Vertical coordinate for the DCM tracking with the final controller setup

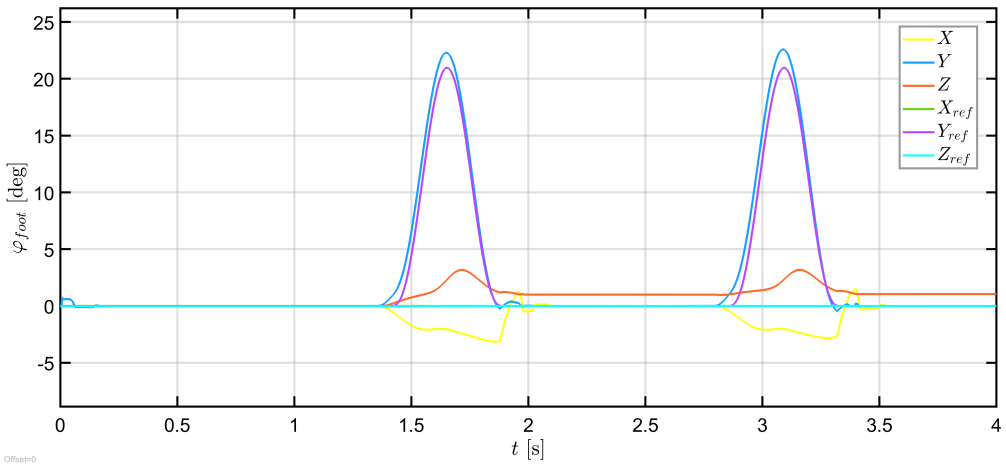


Figure 5.25: Orientation tracking of the foot with the multiple contact point approach

cannot be moved. Little focus of this work lied on optimizing for perfect orientation tracking, but the functionality of the introduced implicit orientation tracking method via multiple RCF tasks is verified.

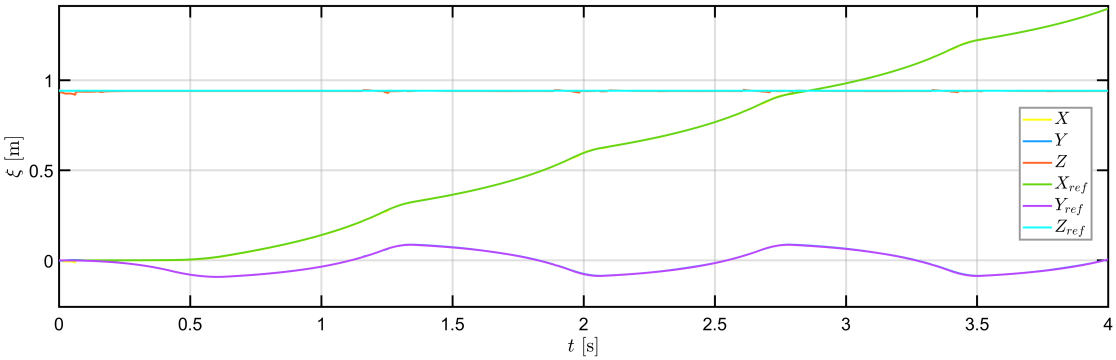
5.5 Comparison of Walking Performance

In this section the walking performance of the controller setup that was used for plots for the ground contact loss will be compared to absolute position controllers from the works in [11] and [4]. It should be noted that the RCF controller was not optimized for best walking performance but for robustness in the different test scenarios of falling and lifting in which the other controllers fail. The inverse dynamics and MPTC controller have been tested for walking with the real robot. The controllers will be compared for the tracking of the divergent component of motion, the tracking of the foot and the center of mass and the emerging joint torques during the walking motion.

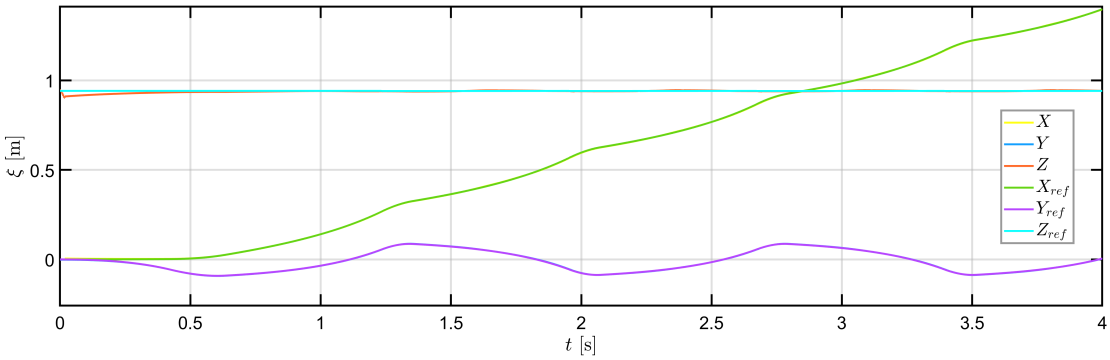
The plots in 5.26 show the tracking of divergent component of motion for the three controllers. Very good tracking can be observed in all plots. Differences occur during the touch-down of the foot onto the ground. The inverse dynamics controller shows the smoothest reaction to the impact of the step. The RCF controller has a visible offset, indicating a harder impact. Another difference can be observed at the beginning of the plot when the robot drops 1 mm onto the ground. The RCF controller recovers faster than the other two controllers which have an addition linear centroidal momentum task similar to the angular momentum task, only applied to the linear motion as well. This faster adaption is advantageous after a drop onto the ground for faster recovery. Implementing the linear momentum task similar to the other controllers is not possible as it uses the absolute velocity of the center of mass. A reformulation to a relative coordinate would be required for implementation in the RCF controller setup.

For the subsequent plots, it has been decided to only show tracking errors between the reference and the actual value due to the bad visibility of differences in the plot with the absolute coordinate. Figure 5.27 shows the tracking error of the absolute coordinate for one foot in the same walking motion. The steps of the foot occur at $t = 0.7$ - 1.3 and seconds and $t = 2.1$ - 2.7 seconds. In the other time, the foot's position is constrained by the ground. It can be seen that movement is not restricted entirely and small slippage occurs, especially for the inverse dynamics controller. The tracking of the RCF controller is worse during the swing phase of the foot, allowing an error of up to 3 mm during the first step. For the other two controllers, the error stays smaller than 1 mm. In all three plots an initial error of 1 mm can be observed in the vertical z-axis after the initial impact on the ground. The error is caused by the trajectory generator which uses the initial height of 1 mm above the ground as reference. For the inverse dynamics and the MPTC controller, the height has been reset for the first step. For the RCF controller this reset was turned off and the 1 mm error stays throughout the motion.

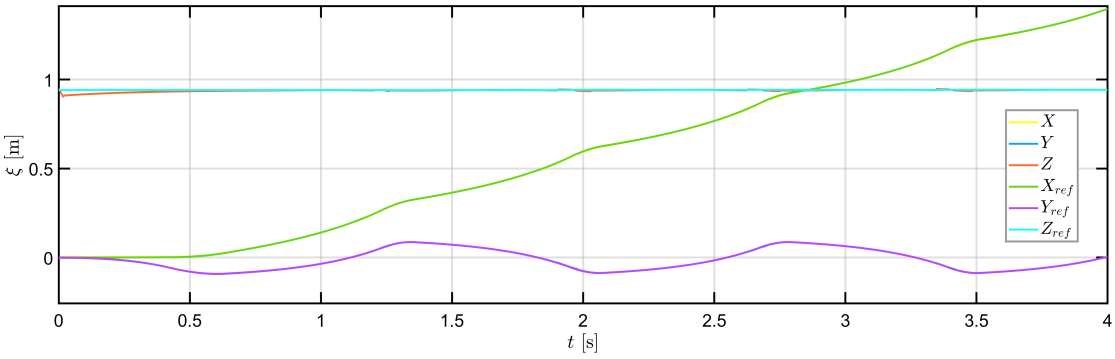
Figure 5.28 shows the absolute position tracking error of the center of mass. A slower convergence to the target in the vertical direction can be seen for the inverse dynamics



(a) RCF

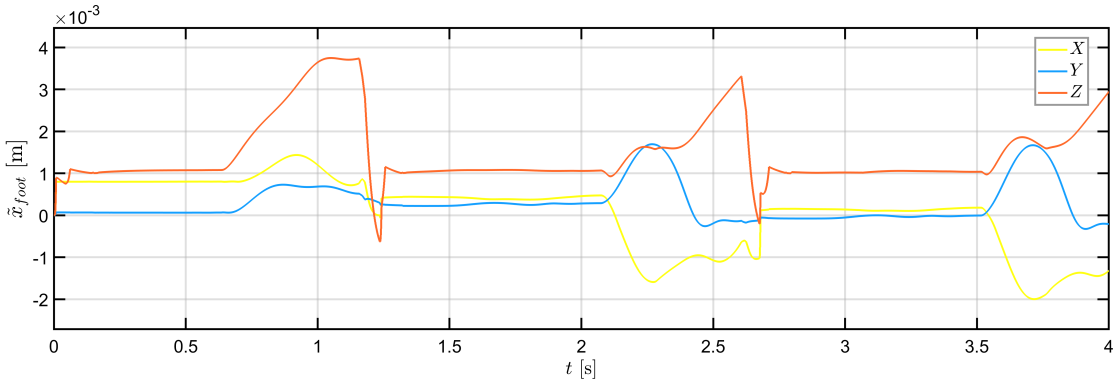


(b) Inverse Dynamics

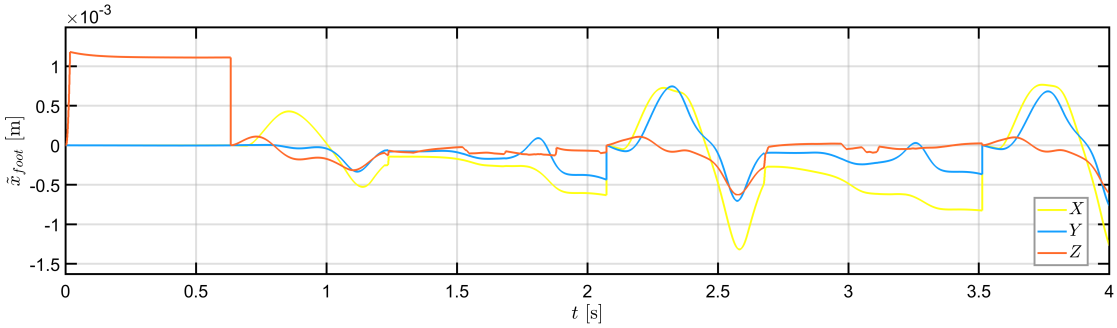


(c) MPTC

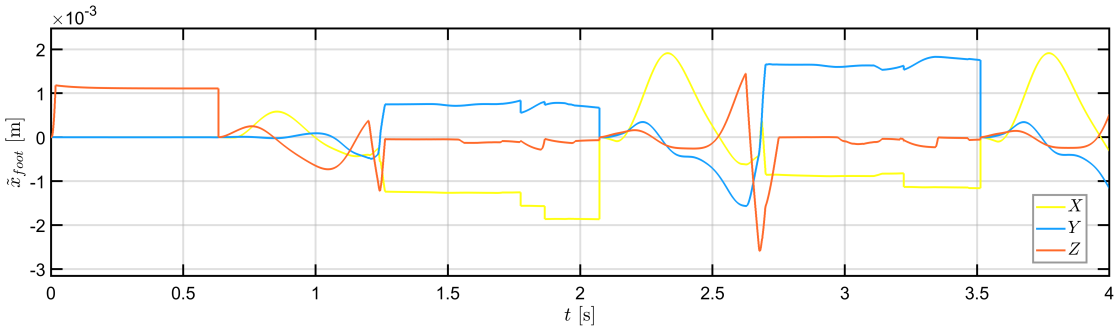
Figure 5.26: Divergent Component of Motion



(a) RCF

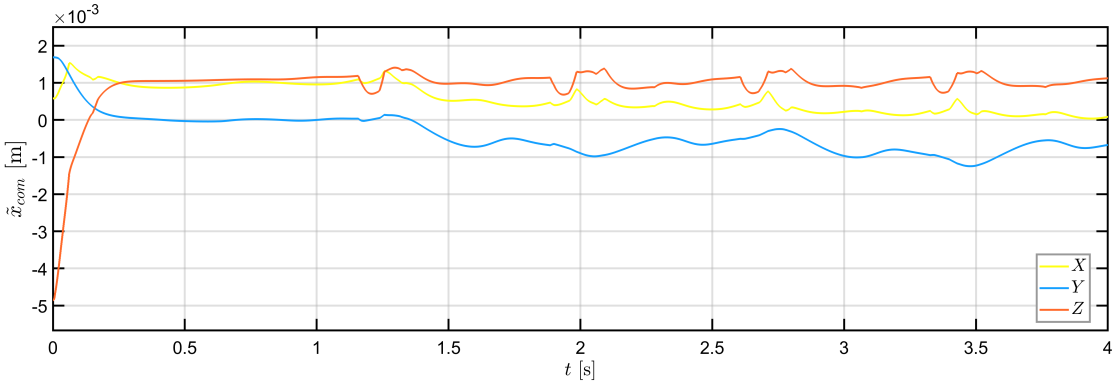


(b) Inverse Dynamics

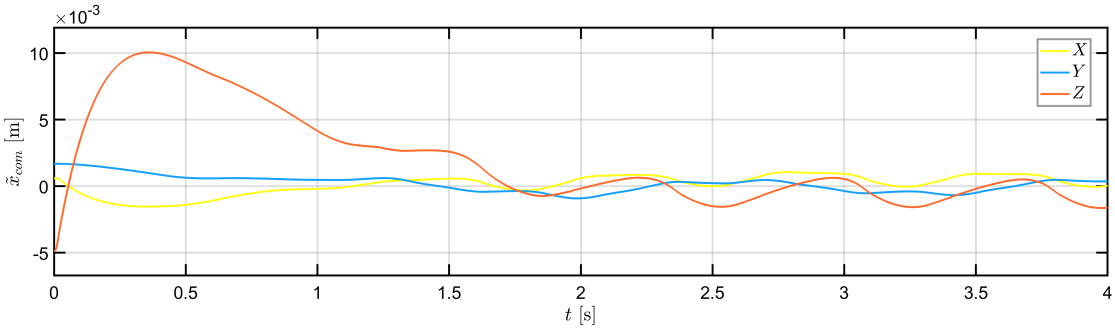


(c) MPTC

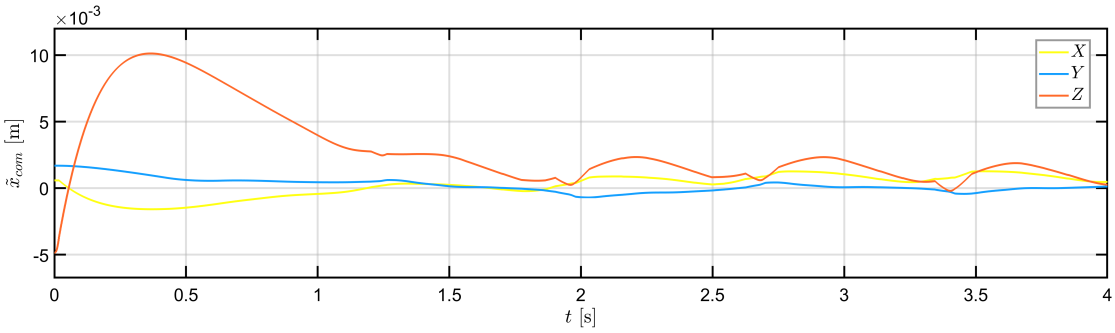
Figure 5.27: Foot tracking error



(a) RCF



(b) Inverse Dynamics

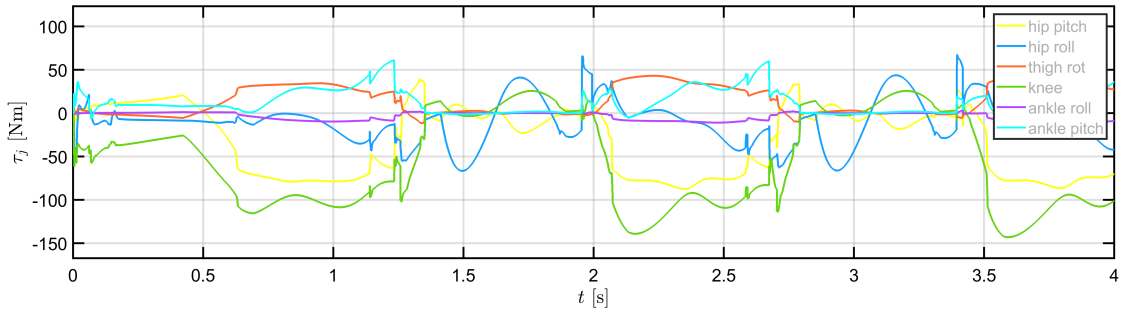


(c) MPTC

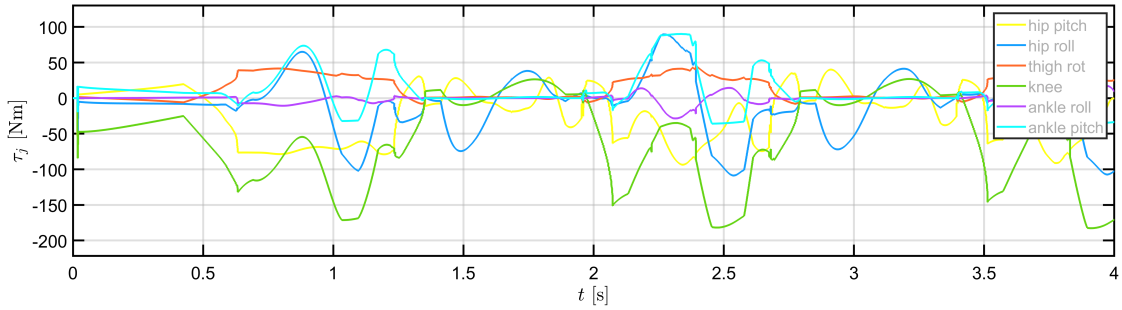
Figure 5.28: Center of mass tracking error

and MPTC controller. This was observed in the DCM comparison as well indicating a lower stiffness of the task. The chose task stiffness was confirmed as practical in the real experiments. An offset of 1 mm is visible in the vertical coordinate for the RCF plot. This is caused by the 1 mm offset of the foot as discussed above. The perfect tracking of the relative coordinate leads to the same error in the center of mass coordinate. The total tracking error apart from that offset is actually smaller than for the other controllers despite the fact, that the controller is not aware of its actual absolute position and good relative coordinate tracking leads to perfect tracking of the absolute coordinate as well. Again the impact of the foot placement is visible stronger in the vertical axis of the RCF plot.

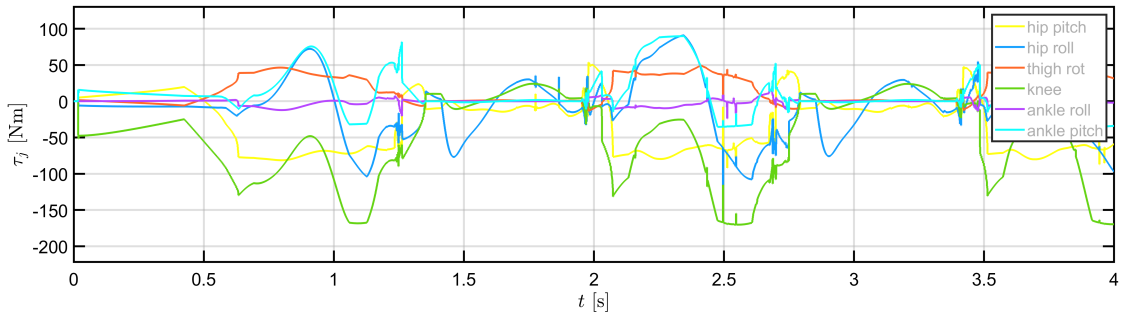
Figure 5.29 shows the joint torques during the walking motion with the three different controllers. The presented RCF controller shows similar performance to the other two controllers in terms of magnitude and smoothness of the torques. The similarity of the singular joint torques is greater between the other two controllers. Spikes occur stronger during impact of the foot, but are not visible throughout the rest of the motion.



(a) RCF



(b) Inverse Dynamics



(c) MPTC

Figure 5.29: Joint torques

6 Experimental Results

The next step was the experimental validation of the derived controller with the humanoid robot Toro. Only three days of testing were possible due to mechanical problems with the robot. On top, the internal measurement unit (IMU) failed at the end of the second day and the experiments on the third day had to be carried out with a worse IMU that exhibits higher measurement noise and delay.

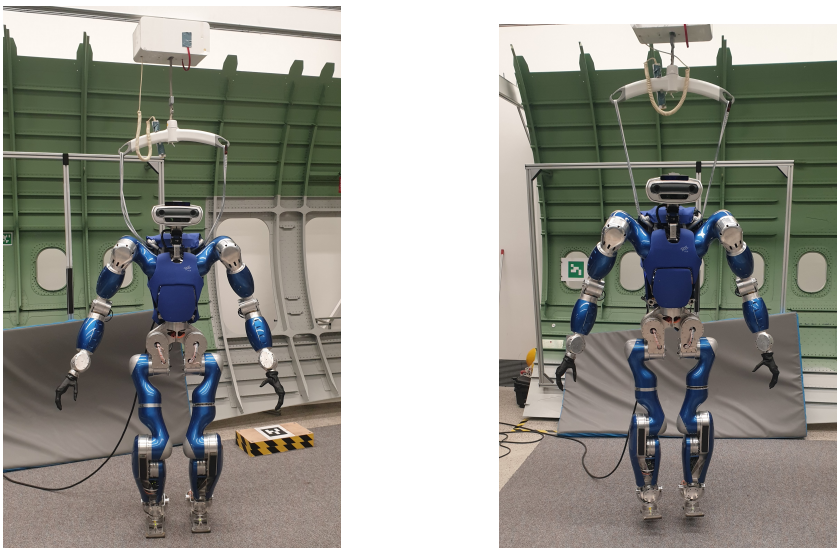


Figure 6.1: The humanoid robot Toro in a stabilized position during stance and in the air without foot-ground contact

On the first day it was possible to find two different configurations of the controller, one that proved to be stable standing on the ground and one lifted off the ground as it can be seen in 6.1. On the second day the two setups were merged. The robot could be lifted off the ground and lowered back on its feet without changes to the controller parameters. The controller was robust against disturbances in the lifted and standing configuration. The experiments on the thirds day were not successful for the walking motion.

6.1 Tuning of the Controller Parameters

Two major changes in the controller parameters were crucial for the successful experiments. The task weight of the center of mass task had to be reduced by a large factor.

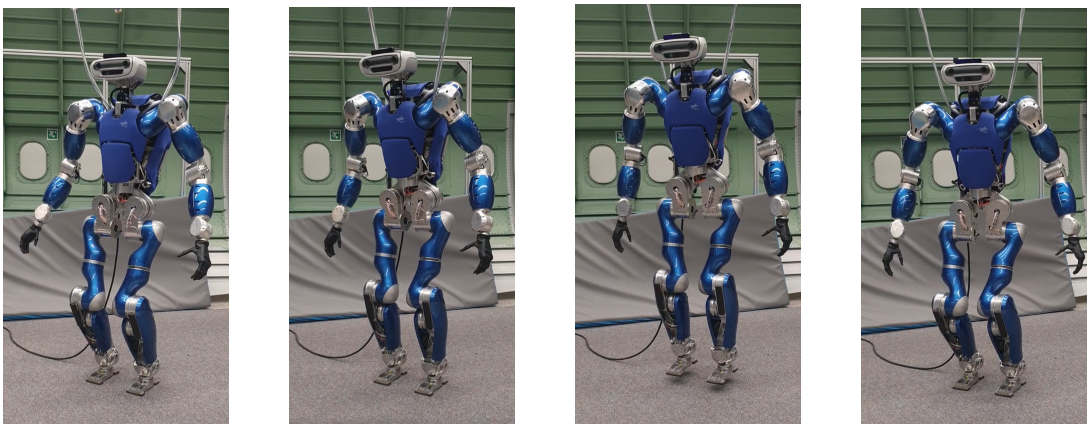


Figure 6.2: The humanoid robot Toro lifted up and lowered back down to the ground. An increase of the leg stretch can be observed as load ist taken off the robot and an increased knee bent due to the additional load when landing on the ground

Higher weights lead to vibrations in the system during stance. Another factor was an increase of the task weight and gain of the angular momentum regulation. It increased the robustness against disturbances on the ground and suppressed vibrations.

The task stiffness and weight of the RCF task were chosen similar to the simulation. A wide range of parameters was possible for this task which demonstrated its compatibility and robustness.

6.2 Verification of Lifting Robustness

Figure 6.2 shows the time series of the robot being lifted off the ground and then placed back to the ground. This experiment will be analysed in this section. The load was taken of the robot in small steps to reduce swinging caused by the abrupt movement of the lifting mechanism. Lowering the robot back to the ground was carried out carefully to protect the hardware. The step by step lifting can be seen in figure 6.3 of the desired and the actual foot wrench. The actual foot wrench reduces all the way to zero and then back to the starting value after the robot is back on its feet. In contrast to that, the commanded wrench expected by the controller does not reduce to zero as it always expects foot-ground contact. Unlike in the simulation, the external force for lifting the robot is applied near the shoulders. This leads to some stabilization of the robot's orientation in the air, but still significant swinging occurred. Figure 6.4a shows the divergent component of motion throughout the experiment and the swinging can be seen in the horizontal axes in the time from 8 to 17 seconds. It can be seen how the vertical coordinate increases gradually when the robot is lifted until the ground contact is lost. The impact of the touch-down of foot is also visible in the plot for the DCM,

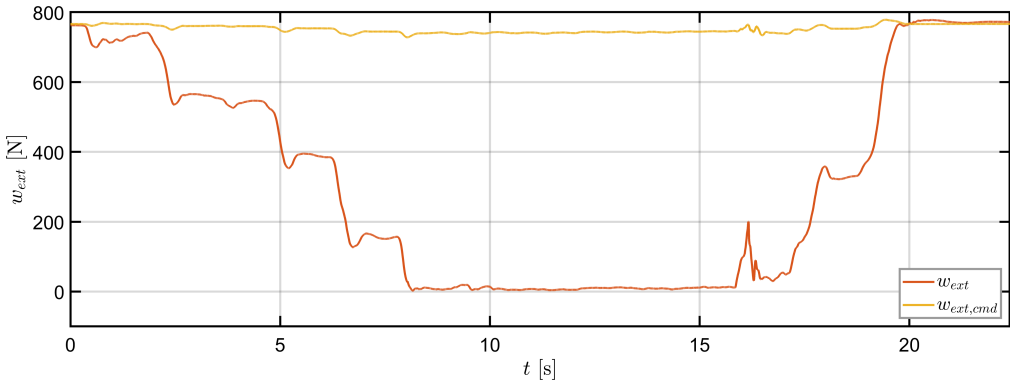
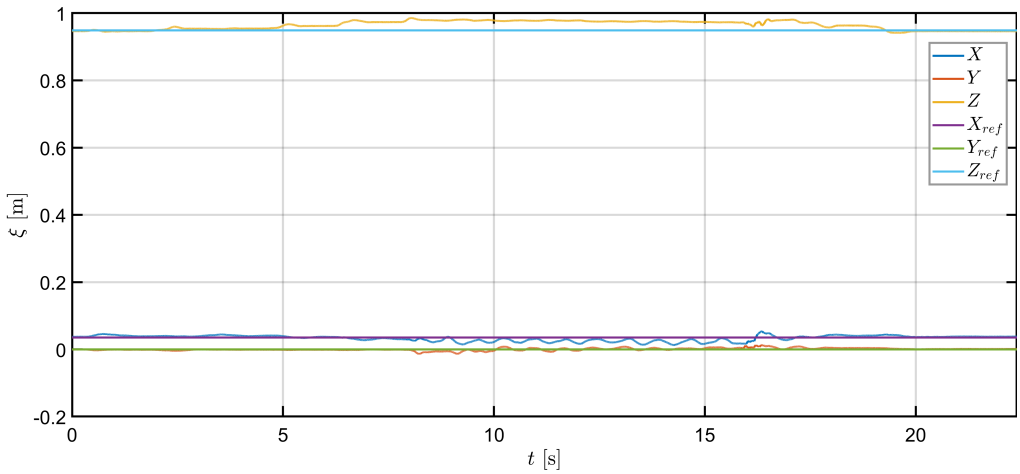


Figure 6.3: The desired and actual foot wrench in the vertical axis during the experiment of lifting the robot

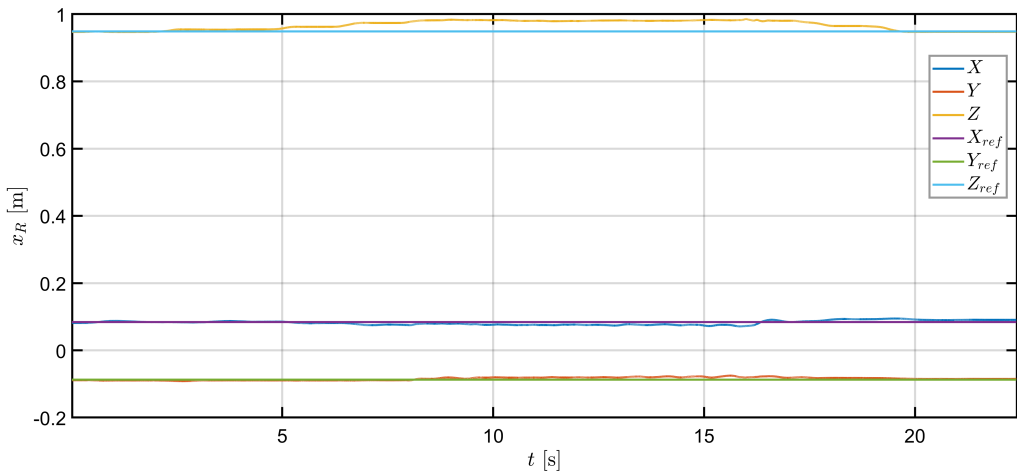
but also for the actual wrench in 6.3 in form of a peak of 200 N. Very good tracking of the absolute position can be observed with established foot-ground contact in the beginning and end of the plot of the DCM and the relative coordinate in figure 6.4b. It can be seen that the controller stays stable throughout the time it is lifted despite the swinging of the robot. The commanded joint torques in figure 6.5a show the reduction of the load on the robot until it is lifted. The reaction of the controller lead to smooth joint torques despite the noise in the estimate of the relative velocity visible in figure 6.5b. The noise is large in the horizontal axes while the foot has ground contact and its movement should be constrained. This noise might be the cause of the vibrations induced by a center of mass acceleration task with higher task weight and further test are required to evaluate that influence. Applying a low pass filter or the HPRL to the estimate might lead to an improvement in the controller performance.

6.3 Cartesian Tracking Without Foot-Ground Contact

Figure 6.6 shows the reaction of the robot when it is disturbed by a force at the foot. The external force induces a momentum about the vertical axis which would lead to rotation of the robot and its foot. It can be seen that the controller holds the desired orientation of the foot and compensates the momentum with a stronger rotation of the upper body. This verifies the introduced contact point approach for orientation tracking also in a scenario without foot-ground contact. The rotation is stopped in the second image with a force applied at the arm.

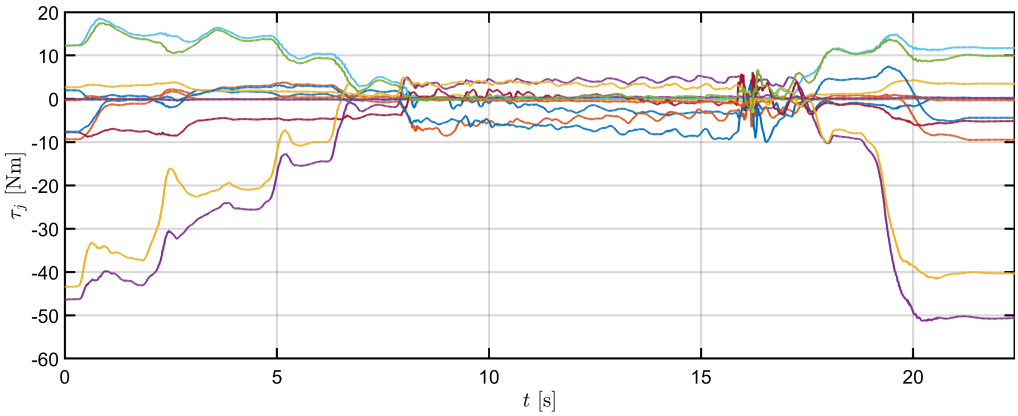


(a) Divergent Component of Motion

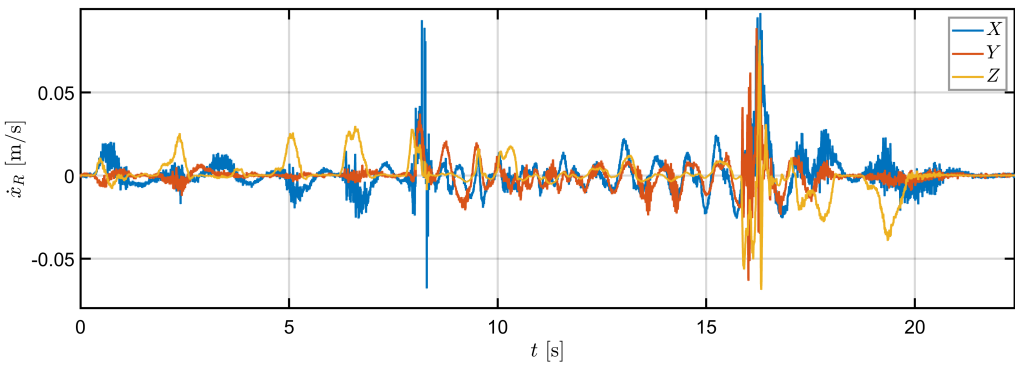


(b) Relative coordinate for one contact point

Figure 6.4: Absolute DCM and relative position tracking during stance and lift of the robot



(a) Joint torques



(b) Relative velocity

Figure 6.5: Joint torques of the legs and the relative velocity of one foot during the experiments. Noise is observed in the horizontal axes of the relative velocity with established foot-ground contact

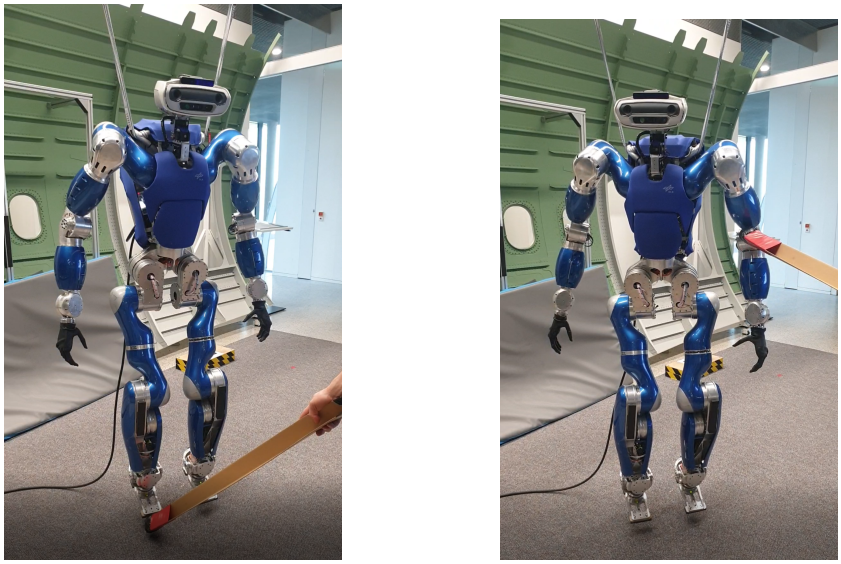
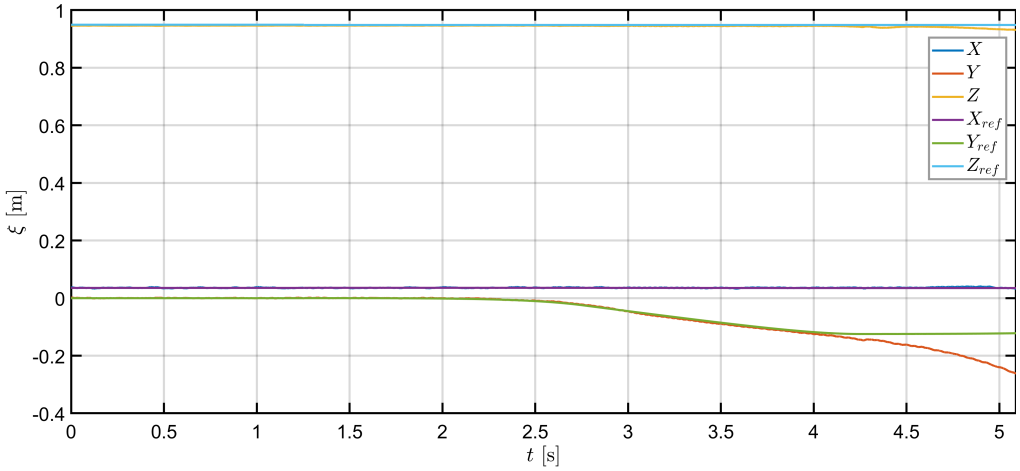


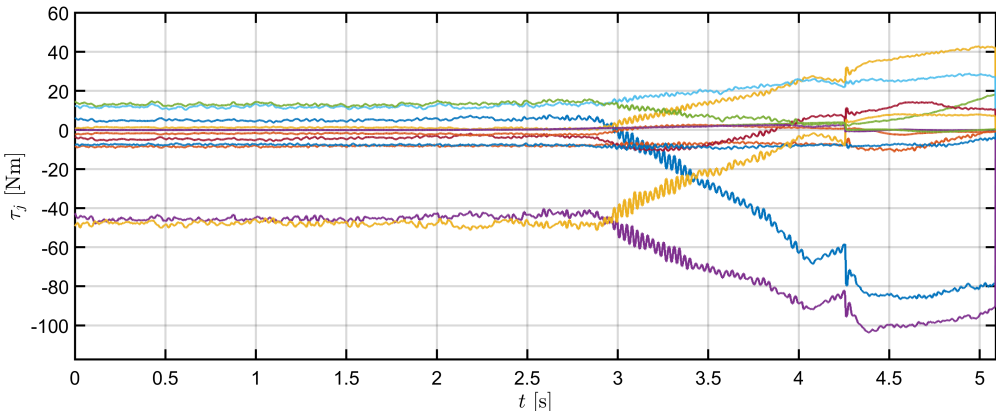
Figure 6.6: The humanoid robot Toro compensating a disturbance at the foot with the upper body to maintain the Cartesian foot position

6.4 Walking Attempts

It was planned to perform walking motions with the robot on the third day, but the necessary switch to the worse IMU caused a setback. It was tried to reach the same performance as on the previous test day. Lifting and standing was again possible, but stable slow oscillations occurred in the lifted position that were impossible to suppress due to the increased delay in the angular rate measurement. This reduced the time available for walking attempts and no successful experiment was carried out. The upper body control task was added to the controller setup for the walking attempts. The task improved the behavior of the controller on the ground and made the robot more robust against pushes. Figure 6.7 shows the divergent component of motion and the corresponding joint torques. It can be seen that the vibrations are stronger in the standing position compared to the plots from the previous test day due to noisier measurements. At approximately 2.3 seconds, the robot starts to shift its weight onto the right foot and takes the left foot off the ground at 4.25 seconds. The walking motion has been slowed down compared to the simulation. During the shifting of the weight, the vibrations increase and become visible in the DCM in figure 6.7a. The sideways motion was not stopped by the controller and the robot fell in that direction after the left foot was lifted of the ground. Figure 6.8 shows the tracking of the relative coordinate for one contact point of each foot. It can be seen that the tracking is very good while standing with both feet on the ground and remains good while the weight is being shifted. When the left foot is lifted of the ground, the tracking becomes worse for the right relative coordinate (6.8b) and the motion in the horizontal direction is not stopped as it was visible in the DCM plot. The tracking of the relative coordinate (6.8a) for the lifted foot

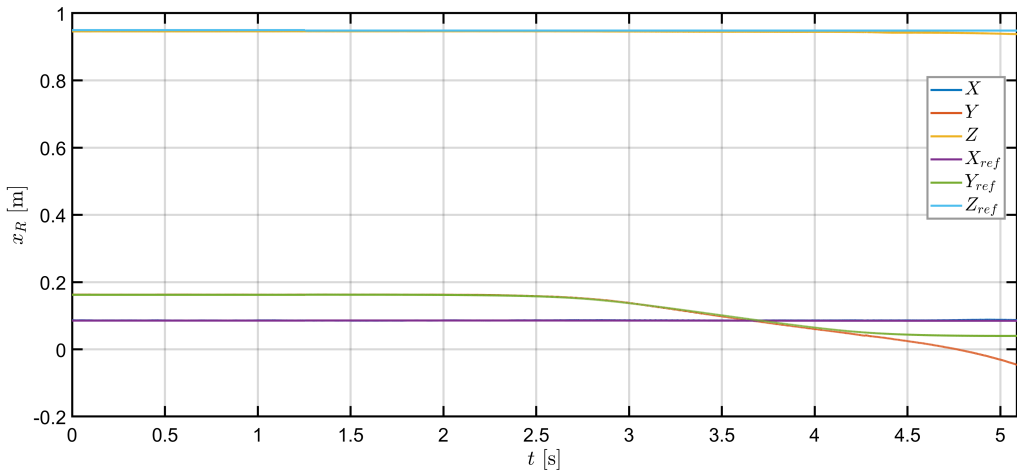


(a) Divergent component of motion

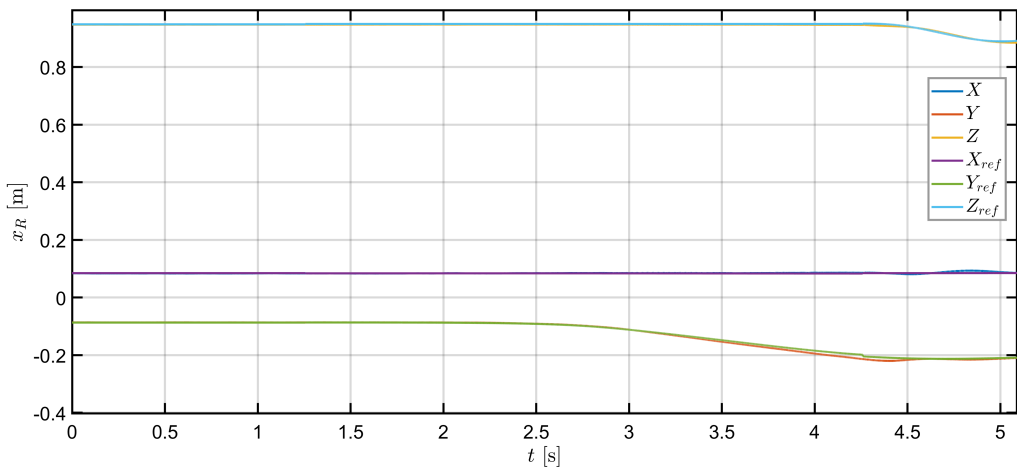


(b) Joint torques of the legs

Figure 6.7: Failed walking attempt and vibrations due to a different IMU



(a) Contact point on right foot



(b) Contact point on left foot

Figure 6.8: Relative coordinate tracking during a walking attempt with failure during the first step

remained good until the controller was shut off by the safety for collision avoidance.

The bad tracking was caused by the too low task weight of the center of mass task. The joint posture task was weighted relatively high and restricts motion with its constant reference. The trade-off between the tasks favored the joint posture task and therefore inhibited the walking motion with incorrect joint torques. No conflict occurs in the standing position of the robot as both the center of mass task and the joint posture task have a constant reference.

7 Conclusion

The goal of this thesis was the derivation of a three-dimensional position controller for humanoid locomotion which is robust against contact loss without compromising on the absolute position tracking under nominal foot-ground contact. A passivity based control task was introduced that is based on the relative position of the center of mass to the foot. If the foot's absolute position is constrained in the world frame when it is placed on the ground, the relative control task tracks the absolute position of the center of mass.

As the developed task does not account for effects of gravity, an additional control task had to be derived to achieve the desired tracking of the absolute coordinate with nominal foot-ground contact. Two different formulations have been found for this task, one based on the foot acceleration constraint applied by the ground to the foot and one based on the nominal center of mass dynamics of the RCF task. Both formulations proved to be stable for the SAFF model and the three joint free-floating robot for contact loss and achieved perfect tracking with foot-ground contact. The tasks were embedded into the whole-body control framework of the humanoid robot Toro. Unexpectedly, the controller setup with the foot acceleration constraint task was instable in the simulation for the humanoid robot. The then introduced center of mass task lead to the desired behavior and robustness in the case of contact loss. The tracking of a walking motion in the relative coordinate without foot-ground contact was possible with a bounded error.

Short instabilities in the form of spikes were observed for the controller when standing or walking on the ground. The instabilities were tackled with different passivation approaches. A quadratic programming constraint was derived which limited the foot power. This lead to a big reduction in spikes but did not resolve the problem entirely. The limitation can lead to instability when the desired motion requires higher foot power than allowed by the constraint as it may be necessary during recovery from a fall or accelerating the motion with faster movements for bipedal running. This constraint used the absolute velocity of the foot and a similar formulation with the relative velocity did not have the same positive effect. In addition it was tested to limit the joint power and the joint power rate as well without observable improvements. A new formulation of the high pass rate limiter with additional passivity based bounding was introduced as a QP-constraint. Again, no significant improvement was observed and concluded that an unstable controller can not be transformed into one with desirable behavior using limitation methods.

Instead, adaptations of the center of mass acceleration task were considered. The adaptations reduced the instabilities in the system with ground contact but did not prove to be stable in the case of contact loss. Further analysis of the instabilities was

carried out using a reduced six joint model of one of Toro's legs where no instabilities were observed during stance. This indicated an interplay of the controller with the foot constraint of the simulation with OpenHRP.

A stable controller setup was found after further analysis of the instabilities by regularization of the linear contact forces at the foot. The controller was compared to absolute position controllers developed at the institute in terms of the absolute position tracking with nominal foot-ground contact in the simulation. Similar performance could be observed to the other controllers and it was demonstrated that absolute position tracking is possible with good tracking of the relative coordinate. The controller was evaluated for different scenarios with loss of foot-ground contact. It was stable when being lifted off the ground entirely and during free fall. Also recovery from falls from significant heights were possible without any adaptations or knowledge about it in the controller or trajectory.

This controller setup was then tested on the real robot. The available time for experiments was limited due to hardware problems and an IMU failure. The robustness of the controller in the case of contact loss could be verified for a standing position. The robot was lifted off the ground from a standing position, remained stable in the air and lowered back to the ground into the standing position. The controller's reaction to disturbances in the form of pushes was robust, both on the ground and when lifted up. The controller compensated for an external force applied at the foot by upper body movements and maintained the Cartesian orientation of the feet in space. The relative MPTC task can also be used as a stable alternative to the joint posture task when the use of Cartesian is advantageous. The task showed great overall compatibility with previously developed control tasks for humanoid locomotion.

The controller could not be verified for a walking motion on the ground. The main cause of this result was the low weighted center of mass acceleration task compared to the joint posture task which inhibits motion due to its constant reference. This low task weight was necessary due to high frequency vibrations caused by a higher weight center of mass task. It could not be determined if the cause of the vibrations was because of the noisy velocity signal in the relative coordinate or induced by the short instabilities that were previously observed in the simulation analysis. Further investigations have to be conducted in this direction. For tests on the ground, the introduced center of mass acceleration formulation with removed feedback would be very useful, as the noisy relative velocity signal has no impact on this task. Another approach for further analysis would be the implementation of the contact model from the OpenHRP simulation in the simplified three- or six-joint robot models and check for occurrence in them.

List of Figures

Bibliography

[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. "Biped walking pattern generation by using preview control of zero-moment point". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Vol. 2. 2003, 1620–1626 vol.2. DOI: 10.1109/ROBOT.2003.1241826.

[2] S. Lohmeier, T. Buschmann, and H. Ulbrich. "Humanoid robot LOLA". In: *2009 IEEE International Conference on Robotics and Automation*. 2009, pp. 775–780. DOI: 10.1109/ROBOT.2009.5152578.

[3] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba. "Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 3411–3416. DOI: 10.1109/IROS.2012.6385840.

[4] J. Englsberger, G. Mesesan, A. Werner, and C. Ott. "Torque-Based Dynamic Walking - A Long Way from Simulation to Experiment". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 440–447. DOI: 10.1109/ICRA.2018.8462862.

[5] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt. "Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas". In: *International Journal of Humanoid Robotics* 13 (Mar. 2016), pp. 1650007–1. DOI: 10.1142/S0219843616500079.

[6] M. A. Hopkins, D. W. Hong, and A. Leonessa. "Compliant locomotion using whole-body control and Divergent Component of Motion tracking". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 5726–5733. DOI: 10.1109/ICRA.2015.7140001.

[7] B. Henze, R. Balachandran, M. A. Roa-Garzón, C. Ott, and A. Albu-Schäffer. "Passivity Analysis and Control of Humanoid Robots on Movable Ground". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3457–3464. DOI: 10.1109/LRA.2018.2853266.

[8] G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer. "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)*. 2019, pp. 25–32. DOI: 10.1109/Humanoids43949.2019.9035053.

[9] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini. "STANCE: Locomotion Adaptation Over Soft Terrain". In: *IEEE Transactions on Robotics* PP (Jan. 2020), pp. 1–15. DOI: 10.1109/TRO.2019.2954670.

[10] J. Englsberger, A. M. Giordano, A. Hiddane, R. Schuller, F. Loeffl, G. Mesesan, and C. Ott. "From Space to Earth - Relative-CoM-to-Foot (RCF) control yields high contact robustness". In: July 2021. DOI: 10.1109/HUMANOIDS47582.2021.9555804.

[11] J. Englsberger, A. Dietrich, G. Mesesan, G. Garofalo, C. Ott, and A. Albu-Schäeffer. "MPTC – Modular Passive Tracking Controller for stack of tasks based control frameworks". In: July 2020. DOI: 10.15607/RSS.2020.XVI.077.

[12] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer. "Overview of the torque-controlled humanoid robot TORO". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 916–923. DOI: 10.1109/HUMANOIDS.2014.7041473.

[13] J. Englsberger. "Combining reduced dynamics models and whole-body control for agile humanoid locomotion". PhD thesis. Dec. 2016.

[14] F. Kanehiro, H. Hirukawa, and S. Kajita. "OpenHRP: Open Architecture Humanoid Robotics Platform". In: *I. J. Robotic Res.* 23 (Feb. 2004), pp. 155–165. DOI: 10.1177/0278364904041324.

[15] G. Garofalo, C. Ott, and A. Albu-Schäeffer. "On the closed form computation of the dynamic matrices and their differentiations". In: Nov. 2013. DOI: 10.1109/IROS.2013.6696688.

[16] G. Garofalo, B. Henze, J. Englsberger, and C. Ott. "On the inertially decoupled structure of the floating base robot dynamics". In: Feb. 2015.

[17] S. Nakaoka, S. Hattori, F. Kanehiro, S. Kajita, and H. Hirukawa. "Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms". In: Dec. 2007, pp. 3641–3647. DOI: 10.1109/IROS.2007.4399415.

[18] coin-or. *qpOASES*. https://github.com/coin-or/qpOASES. 2017.

[19] A. Albu-Schaffer, C. Ott, and G. Hirzinger. "Cartesian Impedance Control of Redundant Robots: Recent Results with the DLR-Light-Weight-Arms". In: vol. 3. Oct. 2003, 3704–3709 vol.3. ISBN: 0-7803-7736-2. DOI: 10.1109/ROBOT.2003.1242165.