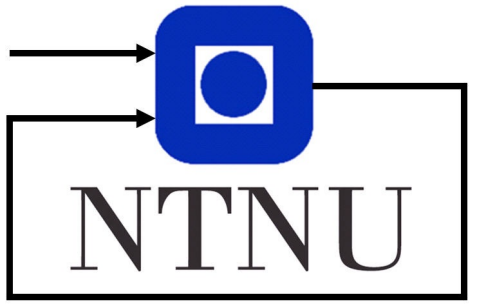
Robust dynamic walking in a sprawling quadrupedal robot



Author:
Mads Erlend Bøe Lysø

Supervisors:
Prof. Kristin Ytterstad Pettersen & Dr. Esten Ingar Grøtli

Specialization project
Department of Engineering Cybernetics
Norwegian University of Science and Technology

January 10, 2022

Preface

The following thesis is submitted as a specialization project report in the department of Engineering Cybernetics at the Norwegian University of Science and Technology. The thesis delves into the field of quadrupedal robotics and is inspired by the many challenges faced by the field and by autonomy in general.

I would like to express my gratitude to my academic supervisors Prof. Kristin Ytterstad Pettersen and Dr. Esten Ingar Grøtli for their time and invaluable guidance, expertise and feedback.

Abstract

Legged robotics has been a research topic of great interest for the last half-century. On one hand, this is due to the field's potential applications across a wide variety of domains both military, commercial and humanitarian in nature.

On the other hand, the topic is of great academic interest as it involves a number of substantial theoretical and application-related challenges. As the field of legged locomotion generally and quadrupedal locomotion in particular matures, the focus has to a large extent been on quadrupeds with a mammalian leg configuration.

However, there exist other possible leg configurations which may have distinct advantages but which have not been thoroughly explored in the existing literature. One of these is the sprawling leg configuration, exemplified in nature in certain reptiles and arachnids.

In this thesis, a method using a simplified single rigid body dynamics system model and convex Model Predictive Control is presented to achieve dynamic walking in the sprawling quadrupedal robot ASTRo. The system dynamics are reduced to a rigid body model, using the standard assumption of approximately mass-less legs.

The resulting dynamics are linearized with respect to roll and pitch, and a precalculated reference in yaw is substituted to yield an approximate, Linear Time Varying model for the system to be used for Model Predictive Control.

Another particularly interesting method utilizing the full hybrid dynamics of the system is examined and given a thorough presentation as well, though not implemented.

The implemented control algorithm performed well on a trotting stationary and forward gait. An ambling gait was also examined, although slightly degraded performance was achieved.

The results suggest that the method in question shows promise for producing dynamic locomotion in sprawling quadrupeds. However, further modifications to the method may be required to achieve truly robust locomotion for a wide variety of dynamic gaits.

Table of Contents

List of Tables

List of Figures

1

Introduction

The field of robotics has for decades increased efficiency and reduced the need for human labor by aiding human workers in carrying out certain tasks, while facilitating the automating away of others. This automation is expected to continue and gain momentum as we enter what has at times been referred to as the fourth industrial revolution.

However, a large portion of jobs requires interaction with and locomotion through unstructured or human-centered environments. Such task may include load-bearing in manufacturing or construction, search-and-rescue, goods delivery, transportation, etc. Thus, robots which are able to traverse such environments are required. In this context, legged robots are often considered the solution: They have the abilities to traverse non-smooth terrains which lack in wheeled robots, while exhibiting greater load-bearing capacity for less energy output compared to drones.

The problem of robotic walking is a fairly complex one. Firstly, the dynamics of a robot with any revolute joints are highly nonlinear and do not lend themselves to linearization if the robot is to traverse any significant part of its configuration space. Walking robots are also necessarily underactuated: The force used for locomotion is the ground reaction forces from the feet.

Firstly, this prohibits any sort of thrust downwards apart from gravity. Secondly, as the static friction force is proportional to the normal force, the force which can be applied in the surface's tangent plane is always limited, to a lesser or greater extent. Walking robots are in addition hybrid systems: Any time a foot is lifted or put down the system inputs change in a discrete manner, and impacts occur any time a foot lands. These factors make control of walking robots a challenging problem, possibly requiring novel methods for control.

In the last decade or so, dynamic bipedal and quadrupedal walking has received increased public interest as it has started to take the leap from the strictly academic into the commercial arena. Where quadrupedal robots are concerned, most research so far has been

conducted on robots with a mammalian joint configuration. A mammalian joint configuration entails that the first joint of each leg – the hip joint – has a rotational axis which is aligned with the roll axis of the body coordinate frame, see Figure 1.1a. Examples of such robots are Boston Dynamic's Spot, Anybotics' ANYmal (Hutter et al. (2016)), Ghost robotics' Vision 60, as well as the series of Cheetah-robots developed at MIT (Bledt et al. (2018); Katz et al. (2019)).

On the other hand, research on robots with alternative joint configurations, such as a sprawling configuration, is relatively sparse. A sprawling joint configuration entails that the hip joint's rotational axis is aligned with the yaw axis of the body coordinate frame, see Figure 1.1b. Sprawling quadrupeds may have certain advantages over mammalian quadrupeds: They have the ability to widen their stance compared to their mammalian counterpart. This widens the support polygon of the robot, making it less prone to falling over and thus more robust. This becomes even more important for rugged terrain, or when legs hit unforeseen obstacles.

It is thus of interest to examine whether methods for walking developed for and tested on mammalian-inspired robots are easy to transfer and use with sprawling robots. The success of a locomotion method may be evaluated in a number of ways, and the following questions will be considered:

- Does the method produces gaits which remain stable over a long period of time?
- Is the actuator torque expenditure kept reasonably low?
- Is the method flexible, i.e., is it able to produce various gaits as opposed to being tailored for one specific gait?

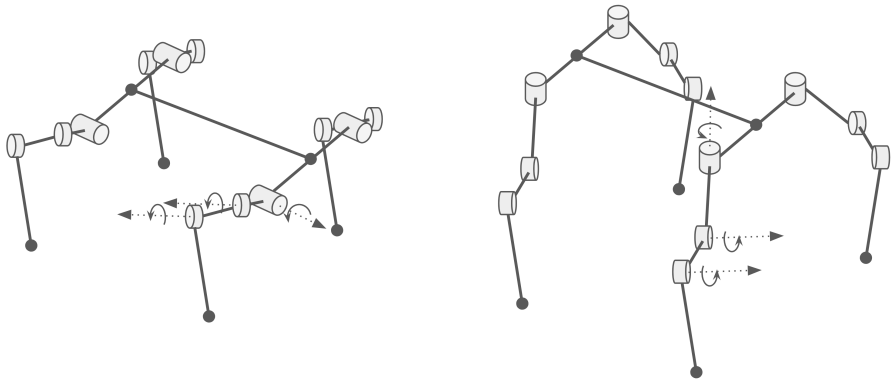
This thesis will carry out the following goals:

- A literature survey will be performed, looking at the existing body of research on the topic of dynamic locomotion in mammalian quadrupeds.
- Based on the literature review, two methods will be selected to be elaborated further on.
- One of the two methods will be chosen for implementation for use with the sprawling tetrapod robot developed in Ghansah and Thorseth (2021).
- The implementation will be tested in the simulation framework Gazebo and the results will be evaluated.

The contribution of this thesis will be an examination of whether methods which have previously been successfully applied for dynamic locomotion in mammalian quadrupeds will be successful when applied for dynamic locomotion in sprawling quadrupeds, with implementation and testing of one.

1.0.1 Thesis overview

The thesis begins with a literature review in chapter 2 in order to give an overview of the state of the field of mammalian quadrupedal locomotion and examine potentially promis-



(a) Mammal-like configuration **(b)** Sprawling configuration

Figure 1.1: Illustration of different quadruped leg configurations

ing methods use for sprawling robots. Following that, two methods which were found to be of particular interest will be described and explained in greater detail in chapter 4, and the overarching architecture of the system as implemented is presented. In chapter 5 the results from the method tested in simulation are presented and the method's performance and further implications are discussed. Finally, concluding remarks and suggestions for further work are presented in chapter 6.

2

Literature Review

Several methods have previously been proposed for achieving robust dynamic walking and running in mammalian quadrupeds. In what follows, a few of these efforts are listed and described in order to give the reader an overview of the state of the field of quadrupedal robotic locomotion.

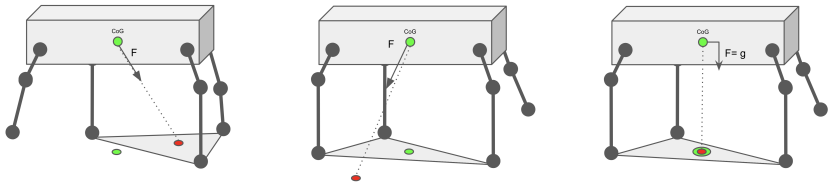
2.1 Zero Moment Point

The Zero Moment Point concept is one of the older, more acclaimed concepts in planning and controlling dynamic gait in both bipedal and quadrupedal robots. In Vukobratovic and Borovac (2004), Vukobratovic and Borovac give an account of the concept and its role in the history of legged locomotion.

In a sense, the task of achieving any sort of walking in a robot is the task of achieving some motion which propels the robot forward while ensuring that it doesn't fall over. Controlling the robot's motion is done indirectly, through the reaction forces from the ground on the robot.

The support polygon is the convex hull spanned by all the contact points (in the case of point feet). It is, in other words, the part of the ground in which ground reaction forces act on the robot. If the robot does not already have angular momentum so as to result in a tilt, then a lack of applied moment will ensure that it continues to not tilt. There is one axis along which the sum of all moments and forces acting on the robot can be replaced by a single force.

Consider the point at which this axis intersects the ground. If this point – known as the Zero Moment Point – lies within the support polygon, then the sum of ground reaction forces can be considered also as one single resultant force acting on that point. Under assumptions of no sliding, this resultant reaction force balances out all other forces acting on the mechanism.



(a) CoG outside of support polygon but ZMP inside **(b)** CoG inside of support polygon but ZMP outside **(c)** CoG corresponds with ZMP (stationary case)

Figure 2.1: Illustration of the relation between CoG projection and ZMP

However, if the intersection of this axis with the ground falls outside of the support polygon, this resultant force – applied at one point in the ground plane – and the ground reaction force – applied at another point – will not balance out but rather generate moments, potentially tilting the robot over. With this in mind, the goal then becomes to plan a combination of footholds and trajectories which leads the ZMP to be within the support polygon of the robot at all times, what is known as the ZMP condition.

It may be interesting to note that, in the stationary case where the only non-ground force affecting the robot is gravity, the ZMP coincides with the projection of the center of gravity onto the ground. Please see Figure 2.1 for an illustration of the relation between the projection of center of gravity and the ZMP.

While the ZMP concept has been widely used in the existing literature, it is somewhat limited. The concept rests on a balance of moments, and does not consider angular momentum or velocity. Especially for gaits involving higher angular velocities, i.e. more dynamic gaits, this can lead the robot to tip over even while fulfilling the ZMP condition (Yeom and Bae (2021)).

2.2 Vertical Impulse Scaling

In Park and Kim (2015), a bioinspired controller is introduced based on a limit cycle approach.

The initial observation is that in animals, swing phases remain mostly constant while stance phases may change to alter gait period. As the gait frequency increases and stance phase decreases, the magnitude of the ground reaction force profiles increases to keep making up for the vertical momentum lost to gravity.

Initially, a nonlinear optimization problem over the reaction force profiles and the timing and duration of stances is solved offline for one desired forward velocity in order to obtain a limit cycle. Having obtained this limit cycle and the corresponding ground reaction forces and timings, the designed gait is converted to state-feedback control of ground reaction forces, using the angle between the respective stance leg and the ground as phase variable.

On top of this feed-forward signal a virtual spring and damper is added in the body height and pitch, in order to stabilize the limit cycle. Lastly, additional horizontal force is added to control speed.

Having calculated the desired ground reaction force, it is transformed to a signal in torques using a simplified version of the inverse dynamics, ignoring leg inertia as well as coriolis effects. In order to achieve higher speed gaits while respecting friction cone considerations, the stance phase is reduced and the vertical impulse is scaled commensurately, so that the area under the reaction force curve is constant. This allows for greater horizontal force to be applied to the ground without slipping, resulting in higher speed gaits.

Though the method seems to allow for highly dynamic locomotion – galloping – both the simulation and the experiments carried out has the robot constrained to a plane. It is thus not given that the method will hold – that is, produce feasible stable locomotion – for the fully 3D case.

2.3 Contact Time Modulation

Yeom and Bae (2021) introduces the Contact Time Modulation method for stabilization of various quadrupedal gaits. The method builds on observations about typical properties of walking in animals. Firstly, the duration of the swing phases of a gait tend to be constant regardless of the gait period. Thus, any variation in gait period is accounted for by a shrinking or growing of the stance phase of each leg.

Further, the ground force profile of each leg in animals has been shown to be well-approximated by a quadratic function. The impact applied to the ground by each foot is also assumed to be identical. Lastly, while one cannot control the exact time at which a foot makes contact with the ground, one can control when a foot lifts off. Thus, one can control the stance time of each foot (within limits).

Using conservation of momentum and assuming constant height across gait cycles, one can thus calculate the nominal stance foot force profile.

A closed form approximation of the pitch after one gait period is derived as a function of the length of the stance phase of each foot. By imposing that the robot should attain zero pitch after a full cycle has passed, one can calculate the desirable stance phase – and thus liftoff time – of each foot.

The controller was proven to be stable in the sense of Lyapunov. As the closed form expression for pitch after a period was an approximation, a disturbance term was included in the analysis to account for modeling errors. By keeping the stance duration within a given disturbance-dependent interval, the pitch of the closed-loop system was shown to be stable.

Furthermore, a nonlinear disturbance observer was employed to estimate the disturbance and modeling error. The method was shown to outperform the vertical impulse scaling method as well as the Zero Moment Point method across a range of metrics, for instance robustness to disturbances and different ground stiffness.

The critique about planar motion is relevant for this paper as well as the robot's roll angle is not modeled or attempted controlled. Extension to roll stabilization is listed as further work.

3

System modeling

3.1 Floating base kinematics

The kinematics of ASTRo can be described by a set of generalized coordinates $\mathbf{q} = [\mathbf{q}_b^\top \ \mathbf{q}_r^\top]^\top$ where $\mathbf{q}_b \in \mathbb{R}^3 \times \mathrm{SO}(3)$ is the pose of the base, while $\mathbf{q}_r \in \mathbb{R}^{n_r}$ are the generalized coordinates describing the robot in the base frame, in this case consisting of the collection of joint angles. The orientation of the base is chosen to be parametrized by intrinsic X-Y-Z Tait-Bryan angles (ψ, θ, ϕ) for the system model. While this description has a singularity at $\theta = \frac{\pi}{2}$, this configuration is outside of the planned operational range of the robot. There are three joints for each of the four legs, so $n_r = 12$ and $n = n_b + n_r = 18$.

The position of the robot base in the world frame is denoted $\mathbf{p}_{w\to b}^{w}$. The orientation of the robot base as seen from the world frame will be denoted as R_b^w, and from our choice of parametrization we get $\mathbf{R}_b^w(\psi, \theta, \phi) = \mathbf{R}_x(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_z(\phi)$. The homogeneous transformation matrix describing the base frame as seen in the world frame is denoted $\mathbf{T}_b^w$, and

$$\mathbf{T}_b^w(\mathbf{q}_b) = \begin{bmatrix} \mathbf{R}_b^w(\psi, \theta, \phi) & \mathbf{p}_{w\to b}^{w} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.1}$$

Each leg is described by a transformation from the base frame to each hip frame, and can further be described by a set of Denavit-Hartenberg parameters, a compact form for describing robotic systems. The transform to each hip frame is a pure translation, $\mathbf{p}_{b\to f}^{b}$ with $f \in \{fl, fr, rl, rr\}$ being the foot in question. Each leg consists of three joints and three corresponding links. The innermost joint is the hip yaw joint, which is centered at the hip frame and followed by the hip link. Second is the hip pitch joint, which is followed by the thigh link. Last is the knee pitch joint, which is followed by the calf link. The end point of the calf link is the placement of the foot of each leg.

i	a_i	α_i	d_i	θ_i
1	l_1	$\pm\frac{\pi}{2}$	0	$\theta_{h,y}$
2	l_2	0	0	$\theta_{h,p}$
3	l_3	0	0	$\theta_{k,p}$

Table 3.1: DH parameters for a single leg. α_1 is positive for front right and rear left legs, negative for front left and rear right legs

The length of the hip link is denoted as l_1, and the length from the hip yaw joint to the hip link's center of mass is denoted l_{c1}. Similar denotation structure holds for the thigh and calf links, with lengths l_2, l_{c2} and l_3, l_{c3} respectively. The DH parameters for a single leg are found in Table 3.1 and together they give a description of the kinematics of the robot. In order to calculate the translation to any link's center of mass the last link length l_i will be substituted for the corresponding l_{ci}.

3.1.1 Model assumptions

The derived kinematics and dynamical system rests on a few key assumptions with respect to the robot feet:

- The robot has point feet which have both zero size, zero mass and zero momentum. These are simply located at the exact end of each calf link.
- The point feet are subject to friction forces when in contact with the ground as any object with finite nonzero extension would.
- The point feet are not subject to moments in the tangent plane when in contact with the ground
- The point feet are not subject to rotational friction when in contact with the ground.

3.1.2 System dynamics

The system dynamics can be derived from the robot kinematics as well as the masses and inertias of each rigid body by the use of the constrained Euler-Lagrange equation:

$$\begin{aligned} \frac{\partial L}{\partial \dot{\mathbf{q}}}^\top - \frac{\partial L}{\partial \mathbf{q}}^\top &= \frac{\partial \mathbf{g}}{\partial \mathbf{q}}^\top \boldsymbol{\lambda} \\ \mathbf{g}_c(\mathbf{q}) &= \mathbf{0} \end{aligned} \tag{3.2}$$

where $L(\mathbf{q}, \dot{\mathbf{q}}) = K - P$, is the Lagrangian of the system, K is the kinetic energy which can be described as:

$$\frac{1}{2}\sum_{i=1}^{n_l}\left(m_i \cdot \dot{\mathbf{q}}^\top \frac{\partial \mathbf{p}_{l_i}^w}{\partial \mathbf{q}}^\top \frac{\partial \mathbf{p}_{l_i}^w}{\partial \mathbf{q}}\dot{\mathbf{q}}\right) + \frac{1}{2}\sum_{i=1}^{n_l}\left({\boldsymbol{\omega}^{l_i}}_{w\to l_i}^\top \mathbf{I}_i^{l_i} \boldsymbol{\omega}_{w\to l_i}^{l_i}\right) \tag{3.3}$$

with n_l being the number of links, m_i being each link's mass and $\mathbf{I}_i^{l_i}$ being its inertia in the link frame. $\mathbf{p}_{l_i}^w$ is link i's position in the world frame, while $\boldsymbol{\omega}_{w \to l_i}^{l_i}$ is its angular velocity in the link frame.

P is the potential energy of the system, given by

$$\sum_{i=1}^{n_l} g \cdot m_i \cdot p_{i_z}^w \tag{3.4}$$

where $p_{i_z}^w$ is the vertical position of the ith link in the world frame.

The link frame is chosen for the inertia matrices, as these tend to be constant in the link frame and nonconstant in the world frame. $\mathbf{g}_c$ is the constraint function defining the holonomic constraints of the system. In this system, $\mathbf{g}_c$ is a concatenation of the difference between initial and current position of all stance legs of the system.

4

Methods

4.1 Hybrid Zero Dynamics

The framework of Hybrid Zero Dynamics is a method in which a controller considering the full state and full-order dynamics of the robot is realized. This sets it apart from most other control efforts for quadrupeds, in which clever model simplification and heuristics have been more prominent. This method has been applied successfully in numerous bipedal locomotion applications, and was extended to use on quadrupedal robots in Ma et al. (2019). The method was chosen to be elaborated further on due to the generality of the framework as well as the theoretically interesting aspect of producing controllers considering the full order of the system, making it stand out from most other methods in the field of quadrupedal locomotion.

This is an Input-output feedback linearization approach in which the dynamics of a higher-dimensional system is forced to evolve on a lower-dimensional manifold which is defined by a set of virtual constraints, or outputs. The dynamics as they evolve once constrained to the lower-dimensional manifold is known as the zero-dynamics of the system (Isidori (2011)).

The framework of HZD extends the notion of zero-dynamics framework to hybrid systems, i.e. systems which exhibit both continuous and discrete dynamics. An example of such a system is a legged robot, in which foot lift-off and impact represent discrete jumps in state and transitions between different continuous domains.

4.1.1 Hybrid system formulation

Adhering to the formulation in Hamed et al. (2018), the hybrid system of quadruped locomotion can be given as the tuple $\mathcal{H} = (\Lambda, \mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{D}, \Delta, FG)$. Here, Λ represents a directed cyclic graph $\Lambda = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices, while $\mathcal{E}$ is the set of edges.

We define $\mu : \mathcal{V} \longrightarrow \mathcal{V}$ to be the function mapping each vertex to its succeeding vertex. In this way, we also have that $\mathcal{E} = \{(v \rightarrow \mu(v))\}_{v \in \mathcal{V}}$. Each vertex here represents a continuous dynamical subsystem, while each edge represents a discrete instantaneous transition between two such dynamical systems.

Each vertex evolves according to an ordinary differential equation, arising from the Lagrangian dynamics. The discrete transitions in the case of a quadrupedal robot represent changes in the number and placement of stance legs, a change which is modeled as instantaneous. Thus, each lifting or landing of a foot is represented by such a transition between two vertices.

$\mathcal{X}$ is the set of state manifolds for the vertices, i.e. $\mathcal{X} = \{\mathcal{X}_v\}_{v \in \mathcal{V}}$. Likewise, the set of admissible control inputs is $\mathcal{U} = \{\mathcal{U}_v\}_{v \in \mathcal{V}}$. $\mathcal{D} = \{\mathcal{D}_v\}_{v \in \mathcal{V}}$ is the set of domains of admissibility for each continuous dynamical system, so that $\mathcal{D}_v \subseteq \mathcal{X}_v \times \mathcal{U}_v$. $FG = \{(f_v, g_v)\}_{v \in \mathcal{V}}$ is the set of control systems for each domain.

We assume (f_v, g_v) to be control affine $\forall v \in \mathcal{V}$ so that $\dot{x}_v = f_v(x) + g_v(x)u$ for $(x, u) \in \mathcal{D}_v$. $\mathcal{S} = \{S_{v \rightarrow \mu(v)}\}_{v \in \mathcal{V}}$ is the set of guards so that the instantaneous transition from one domain to the next occurs when the state and control input (x, u) intersects the guard $\mathcal{S}_{v \rightarrow \mu(v)}$. $\Delta = \{\Delta_{v \rightarrow \mu(v)}\}_{v \in \mathcal{V}}$ is then the set of reset laws which relate the end state in one domain to the initial state in the next, so that $x^+ = \Delta_{v \rightarrow \mu(v)}(x^-)$ where x^- is the state the instant before impact, while x^+ is the state the instant after impact.

4.1.2 Continuous dynamics

In the continuous-time domain $\mathcal{D}_v$, the evolution of the state $\mathbf{x} = [\mathbf{q}^\top, \dot{\mathbf{q}}^\top]^\top$ is given by the control system (f_v, g_v), subject to holonomic constraints $\boldsymbol{\eta}_v(\mathbf{q}) = \mathbf{0}$ resulting from static friction on the stance feet. The derivative of holonomic constraints is then $\mathbf{J}_v(\mathbf{q})\dot{\mathbf{q}}$. The system dynamics can be expressed as an implicit second-order ODE resulting from the Euler-Lagrange equations listed in Equation 4.1:

$$\begin{aligned} \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) &= \mathbf{B}\mathbf{u} + \mathbf{J}_v^\top(\mathbf{q})\boldsymbol{\lambda} \\ \mathbf{J}_v^\top(\mathbf{q})\ddot{\mathbf{q}} + \frac{\partial}{\partial \mathbf{q}}\left(\mathbf{J}_v\dot{\mathbf{q}}\right)\dot{\mathbf{q}} &= 0 \end{aligned} \tag{4.1}$$

where $\mathbf{D}(\mathbf{q})$ is the generalized mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ pertains to the coriolis and centrifugal terms and $\mathbf{G}(\mathbf{q})$ represents the potential/gravitational terms. $\mathbf{B}$ is the input matrix, and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers relating to the constraint forces. The constraint forces can be solved for explicitly. As $\mathbf{D}$ is assumed invertible for well-posed models of mechanical systems (all generalized coordinates are associated with non-zero generalized mass), the system can be posed on an explicit control-affine form if desired.

4.1.3 Zero dynamics

The system is to be controlled using a IO feedback linearization controller, in which a control signal is design so as to drive a set of virtual holonomic constraints, or "outputs", to

zero. As described in Hereid et al. (2018); Hamed et al. (2018), these outputs are typically the deviation in some given state, or combination of states, from a desired setpoint or time-parametrized trajectory:

$$\mathbf{y}_v(t) = \mathbf{y}_{v,a}(t) - \mathbf{y}_{v,d}(t) \tag{4.2}$$

where $\mathbf{y}_{v,a}(t)$ is referred to as the actual output, a combination of system states, while $\mathbf{y}_{v,d}$ is referred to as the desired output. Here, each $\mathbf{y}_{v,d}$ is parametrized as a Bézier curve, with corresponding parameter vector $\boldsymbol{\alpha}_v$, i.e. $\mathbf{y}_{v,d} = \mathbf{y}_{v,d}(t, \boldsymbol{\alpha}_v)$. By driving these outputs to zero, the dynamics is forced to lie on a lower-dimensional manifold of the system dynamics. For the case of walking, this lower-dimensional manifold should contain an orbitally stable or stabilizable periodic behavior, which is the gait. The driving of system dynamics to the zero-dynamics is realized by designing $\mathbf{u}$ so that

$$\begin{aligned} \dot{\mathbf{y}}_{1,v}(t) &= -\epsilon \mathbf{y}_{1,v}(t) \\ \ddot{\mathbf{y}}_{2,v}(t) &= -2\epsilon \dot{\mathbf{y}}_{2,v}(t) - \epsilon^2 \mathbf{y}_{2,v}(t) \end{aligned} \tag{4.3}$$

where $\mathbf{y}_{1,v}(t)$ and $\mathbf{y}_{2,v}(t)$ are outputs with relative degree 1 and 2 respectively, and ϵ is a freely chosen tuning parameter. Roughly speaking, outputs with relative degree 2 correspond to virtual holonomic constraints (as they depend only on configuration) while outputs with relative degree 1 correspond to virtual nonholonomic constraints, and may enforce constraints on the derivatives of the configuration variables.

Now, for a set of chosen outputs consider $\mathbf{y} = [\mathbf{y}_1^\top \quad \mathbf{y}_2^\top]^\top$. It is important to choose the outputs in such a way that the map from $\mathbf{q}$ to $\mathbf{y}$ is a diffeomorphism. This is to say that the map is invertible, and that both the map and its inverse are differentiable. If and only if $rank\left(\frac{\partial \mathbf{y}}{\partial \mathbf{q}}\Big|_{\mathbf{q}_0}\right) = n$ (n being the dimension of the configuration space) the map is locally diffeomorphic in a neighborhood around $\mathbf{q}_0$ ((Khalil, 2002, p. 508)). This map being diffeomorphic further implies that the matrix given in Equation 4.4 – called the decoupling matrix – is invertible (Ames (2014)).

$$\mathcal{A} = \begin{bmatrix} L_g \mathbf{y}_1 \\ L_g L_f \mathbf{y}_2 \end{bmatrix} \tag{4.4}$$

where L_f, L_g are the Lie derivatives with respect to f and g respectively, as introduced in subsection 4.1.1. In this case, choosing $\mathbf{u}$ as in Equation 4.5 ensures the desired output dynamics as shown in Equation 4.3.

$$\mathbf{u}_v = -\mathcal{A}_v^{-1} \left(\begin{bmatrix} L_{f_v} \mathbf{y}_{1,v} \\ L_{f_v}^2 \mathbf{y}_{2,v} \end{bmatrix} + \begin{bmatrix} \epsilon \mathbf{y}_{1,v} \\ 2\epsilon \dot{\mathbf{y}}_{2,v} + \epsilon^2 \mathbf{y}_{2,v} \end{bmatrix} \right) \tag{4.5}$$

We hereby refer to the stack of Equation 4.3 and Equation 4.1 as $\mathbf{F}_v(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\alpha}_v)$.

4.1.4 Domain specific inequality constraints

For each vertex v, there exist an admissible domain $\mathcal{D}_v \subseteq \mathbb{R}^n$. The admissible domain can be constrained by a set of equalities or inequalities imposed on variables. Here, we consider two types of inequalities which define the boundaries of the admissible domain:

First, there are constraints imposed on the contact wrenches (or in the case of point feet, as here, contact forces) of the stance feet, denoted as $\mathbf{v}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v)\boldsymbol{\lambda}_v \geq \mathbf{0}$. These amount to 1) requiring the normal force from the ground on the robot to be positive and 2) requiring the tangential forces to be within the friction cone – or a linearization of it – of the foot, as the model assumes no slipping of the feet:

$$\begin{aligned} \lambda^{f_z} &\geq 0 \\ \|\lambda^{f_x}\|_1 &\leq \mu\lambda^{f_z} \\ \|\lambda^{f_y}\|_1 &\leq \mu\lambda^{f_z} \end{aligned} \tag{4.6}$$

It is here assumed that the robot is walking on flat ground, so that the normal force is identical to the force in the z-direction.

Secondly, there are constraints relating to the state of the system other than the contact forces, denoted $\mathbf{h}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v) \geq \mathbf{0}$. Here, we require that the swing feet are above the ground:

$$z_{nsf} \geq 0 \tag{4.7}$$

for all swing feet.

We summarize these constraints as

$$\mathbf{A}_v = \begin{bmatrix} \mathbf{v}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v)\boldsymbol{\lambda}_v \\ \mathbf{h}_v(\mathbf{q}_v) \end{bmatrix} \geq \mathbf{0} \tag{4.8}$$

4.1.5 Guards

Guards are associated with the transition of the HDS from one continuous domain to another along an edge of the DAG. A guard $\mathcal{S}_{v\to\mu(v)}$ is defined as a proper subset of the boundary of the domain, that is, $\mathcal{S}_{v\to\mu(v)} \subset \partial\mathcal{D}_v$.

For a discrete transition to occur, the state must be about to exit $\mathcal{D}_v$ through a guard. Thus, for some chosen element H_e of Equation 4.8 we can define a corresponding guard as $\mathcal{S}_e = \{(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) \,|\, H_e = 0,\ \dot{H}_e < 0\}$ In this paper no-slip contact between the stance feet and the ground are always assumed. Thus, there are two classes of guard conditions.

Firstly, if a swing foot hits the ground, part of the boundary associated with $\mathbf{h}$ is reached, and the system undergoes a discrete jump to a continuous system in which the previous swing foot is now a stance foot. This can be stated as

$$\begin{aligned} z_{nsf} &= 0 \\ \dot{z}_{nsf} &< 0 \end{aligned} \tag{4.9}$$

for any swing foot.

Secondly, if the normal force of any stance foot becomes zero with a negative derivative, this indicates that the associated foot is about to lift from the ground and become a swing foot. This can be stated as

$$\begin{aligned} \lambda^{f_z} &= 0 \\ \dot{\lambda}^{f_z} &< 0 \end{aligned} \tag{4.10}$$

for any stance foot.

4.1.6 Discrete dynamics

The hybrid system transitions between continuous domains as dictated by its discrete dynamics, Δ. The discrete dynamics are distinctly different for the two abovementioned cases of stance-foot liftoff and swing-foot landing. In the case of lift-off, we assume no instantaneous change in state. Thus, the reset-law $\Delta_{v\to\mu(v)}$ simply becomes the identity map.

In the case of landing, we assume a perfectly plastic impact. We also assume no discontinuous changes in the generalized coordinates $\mathbf{q}$, although the generalized velocities $\dot{\mathbf{q}}$ may have discontinuous jumps during impact. From conservation of generalized momentum, we get

$$\begin{aligned} \mathbf{D}\dot{\mathbf{q}}^+ - \mathbf{D}\dot{\mathbf{q}}^- &= \mathbf{J}_{\mu(v)}\boldsymbol{\lambda}_{impulse} \\ \implies \dot{\mathbf{q}}^+ &= \dot{\mathbf{q}}^- + \mathbf{D}^{-1}\mathbf{J}_{\mu(v)}\boldsymbol{\lambda} \end{aligned} \tag{4.11}$$

where $\boldsymbol{\lambda}_{impulse}$ is the intensity of the impulsive contact forces occurring upon impact. These are determined by the generalized momentum of the system before impact, along with the holonomic constraints on the stance feet after impact (velocity of stance feet must be identically zero).

Alongside the assumptions of no abrupt changes to generalized coordinates, i.e. $\mathbf{q}^+ = \mathbf{q}^-$, this determines the reset map $\Delta_{v\to\mu(v)}$ in the cases of impact.

4.1.7 Optimization problem

A large difficulty in designing this type of controllers for walking is the gait design, which relates to finding well-suited specific trajectories $\mathbf{y}_{v,d}(t, \boldsymbol{\alpha}_v)$. These trajectories should

result in gaits that are both feasible and stable, and at the same time the ones which are energetically efficient.

While hand-crafting these desired trajectories is a possibility, the more common approach is to find these feasible trajectories and corresponding inputs by the use of optimization methods, a method known as optimal control. The framework of nonlinear optimization is both powerful and flexible enough to incorporate holonomic constraints, virtual holonomic and nonholonomic constraints, state inequality constraints and bounds, while searching for solutions that minimize either energy expenditure, average torque or other desired objectives.

In the approach suggested in Hereid et al. (2018) and later in Ma et al. (2019), direct collocation methods are used. Direct collocation allows for greater accuracy at longer time steps, decreasing the number of decision variables needed. In this formulation, both the state at interior nodes and the slope at cardinal nodes, as well as all constraint forces, are introduced explicitly as defect variables as opposed to being calculated in closed form. Although this increases the size of the NLP, it is done to help the convergence properties of the problem (Hereid et al. (2018)). It also avoids matrix inversion, which is desirable for a few reasons:

Firstly, there are concerns of numerical stability and accuracy when inverting poorly conditioned matrices. Secondly, matrix inversion is a highly costly operation with time complexity which scales poorly in the number of variables.

The collocation constraints are a set of constraints that forces the state and slope $\mathbf{x}^{(i)} = (\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)})$ and $\dot{\mathbf{x}}^{(i)} = (\dot{\mathbf{q}}^{(i)}, \ddot{\mathbf{q}}^{(i)})$ at the interior nodes to adhere to some interpolation of the states at adjacent cardinal nodes. In Hermite-Simpson collocation, these constraints equal state and slope to the cubic interpolation of state and quadratic interpolation of slope calculated from the previous and following cardinal nodes. The collocation constraint on state is $\boldsymbol{\delta}^{(i)}$ while the constraint on slope is $\boldsymbol{\zeta}^{(i)}$ for the ith node:

$$\begin{aligned} \boldsymbol{\delta}^{(i)} &= \mathbf{x}^{(i)} - \frac{1}{2}\left(\mathbf{x}^{(i+1)} + \mathbf{x}^{(i-1)}\right) - \frac{\Delta t^{(i)}}{8}\left(\dot{\mathbf{x}}^{(i-1)} - \dot{\mathbf{x}}^{(i+1)}\right) \\ \boldsymbol{\zeta}^{(i)} &= \dot{\mathbf{x}}^{(i)} - \frac{3}{2\Delta t^{(i)}}\left(\mathbf{x}^{(i+1)} - \mathbf{x}^{(i-1)}\right) + \frac{1}{4}\left(\dot{\mathbf{x}}^{(i-1)} + \dot{\mathbf{x}}^{(i+1)}\right) \end{aligned} \tag{4.12}$$

with $\Delta t^{(i)} = t^{(i+1)} - t^{(i-1)}$, for $i = 1, 3, 5, \ldots$

The decision variables are, in addition to the states, slopes and input at each node, parameters for the desired output trajectories as well as the placement in time of each cardinal node, $t^{(i)}$ for $i = 0, 2, 4, \ldots$

As the feedback control law is included in the optimization problem, the optimal trajectory parameters yielded in the optimization problem are directly compatible with the described feedback control law. This is distinct from typical optimal control, where an open-loop optimal control signal which is assumed to be piecewise linear or constant, is calculated.

Let $\boldsymbol{\delta}_v$ and $\boldsymbol{\zeta}_v$ be the stacked vectors of $\{\boldsymbol{\delta}^{(i)}\}, \{\boldsymbol{\zeta}^{(i)}\}$ respectively. Also, let $\mathbf{F}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \boldsymbol{\lambda}_v, \boldsymbol{\alpha}_v)$ be the stacked vector of $\{\mathbf{F}_v(\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)}, \ddot{\mathbf{q}}^{(i)}, \boldsymbol{\lambda}^{(i)}), \boldsymbol{\alpha}_v\}$. Then, the full NLP can be stated as follows:

$$\begin{aligned} \min_{z} \quad & \mathrm{J}(\mathbf{z}) \quad \text{s.t.} \\ \mathbf{F}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \boldsymbol{\lambda}_v, \boldsymbol{\alpha}_v) &= \mathbf{0} \\ \boldsymbol{\delta}_v &= \mathbf{0} \\ \boldsymbol{\zeta}_v &= \mathbf{0} \\ (\mathbf{q}_v^{(\mathcal{N}_v)}(t^{(\mathcal{N}_v)}), \dot{\mathbf{q}}_v^{(\mathcal{N}_v)}(t^{(\mathcal{N}_v)}), \mathbf{u}_v(t^{(\mathcal{N}_v)})) &\in \mathcal{S}_{v\to\mu(v)} \\ (\mathbf{q}_{\mu(v)}^{(0)}(t^{(0)}), \dot{\mathbf{q}}_{\mu(v)}^{(0)}(t^{(0)}), \mathbf{u}_{\mu(v)}(t^{(0)})) &\in \mathcal{S}_{v\to\mu(v)} \\ \mathbf{A}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \boldsymbol{\lambda}_v) &\geq \mathbf{0} \\ \forall v \in \mathcal{V} & \end{aligned} \tag{4.13}$$

Where $\mathbf{z}$ is the entire vector of decision variables, and $\mathrm{J}(\mathbf{z})$ is an objective function of choice. Two typical choices might either be an integral cost over the square of expended torque:

$$J(\mathbf{z}) = \sum_{v\in\mathcal{V}} \int_{t^{(0)}}^{t^{(\mathcal{N}_v)}} \mathbf{u}_v(\tau) d\tau \tag{4.14}$$

or expended torque divided by the time interval:

$$J(\mathbf{z}) = \sum_{v\in\mathcal{V}} \frac{1}{t^{(\mathcal{N}_v)} - t^{(0)}} \int_{t^{(0)}}^{t^{(\mathcal{N}_v)}} \mathbf{u}_v(\tau) d\tau \tag{4.15}$$

4.2 Convex MPC over ground forces

Finding feasible and optimal trajectories for the full-order quadrupedal system is a highly nonlinear, non-convex optimization problem. An alternative to finding full state trajectories through optimization over the full-state nonlinear system is using model simplifications to yield models which are simpler, and which have greater chances or even theoretical guarantees of convergence for the corresponding optimization problem. This might also allow for online recurrent solving, yielding control signals which are more responsive to the robot's current situation.

Here, we consider a method using a Single Rigid Body Dynamics approach. In SRBD, only the dynamics of the robot base itself are considered, as acted upon by forces and torques – in this case, ground reaction forces. For this approximation to work well, one assumes that the mass and inertia of the robot legs are so small relative to the base that they can be ignored without incurring too great costs in accuracy to be detrimental.

Di Carlo et al. (2018) describes a method in which the SRBD model is simplified further in order to yield a model which is Linear Time Varying, making it quick enough to solve to lend itself to use in Model Predictive Control. The method was chosen due to its leg-configuration agnostic nature (the simplified model used during MPC does not depend on leg kinematics), as well as its stated ability to produce and stabilize various types of gaits.

4.2.1 Model description

As in Di Carlo et al. (2018), we consider the rigid-body dynamics of the robot base, controlled by the reaction forces from the stance feet at any given time. The orientation is parametrized by extrinsic X-Y-Z Tait-Bryan angles (ϕ, θ, ψ), rendering a 12-dimensional description of pose and twist:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{p}(t) \\ \phi(t) \\ \theta(t) \\ \psi(t) \\ \dot{\mathbf{p}}(t) \\ \boldsymbol{\omega}(t) \end{bmatrix}$$

The translational dynamics are given by

$$\mathbf{a}^w = \frac{1}{m}\left(\sum_i \mathbf{f}^w_{r,i} + \mathbf{g}^w\right) \tag{4.16}$$

Where $\{\mathbf{f^w}_{r,i}\}_i$ are the ground reaction forces from the stance feet.

The rotational dynamics are given by

$$\frac{d}{dt}(\mathbf{I^w}\boldsymbol{\omega}^\mathbf{w}) = \mathbf{I^w}\dot{\boldsymbol{\omega}}^w + \boldsymbol{\omega}^\mathbf{w} \times (\mathbf{I^w}\boldsymbol{\omega}^\mathbf{w}) = \sum_i \mathbf{r}^w_i \times \mathbf{f}^w_{r,i} \tag{4.17}$$

$$\begin{aligned} \boldsymbol{\omega} &= \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi & 0 \\ \cos\theta\sin\psi & \cos\psi & 0 \\ -\sin\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ \implies \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} \frac{\cos\psi}{\cos\theta} & \frac{\sin\psi}{\cos\theta} & 0 \\ -\sin\psi & \cos\psi & 0 \\ \cos\psi\tan\theta & \sin\psi\tan\theta & 1 \end{bmatrix} \boldsymbol{\omega} \quad \forall\theta \neq \pm\pi \end{aligned} \tag{4.18}$$

$$\dot{\mathbf{R}} = [\boldsymbol{\omega}^\mathbf{w}]_\times \mathbf{R} \tag{4.19}$$

where $\{\mathbf{r}_i^w(t)\}$ are the vectors from the body to the stance feet. For small $\boldsymbol{\omega}$, $\mathbf{I^w}\dot{\boldsymbol{\omega}}^w + \boldsymbol{\omega}^\mathbf{w} \times (\mathbf{I^w}\boldsymbol{\omega}^\mathbf{w})$ is well approximated by $\mathbf{I^w}\dot{\boldsymbol{\omega}}^w$. Furthermore, for $\phi, \theta \approx 0$, Equation 4.18 can be simplified to

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\omega} = \mathbf{R}_z^\top(\psi)\boldsymbol{\omega} \tag{4.20}$$

The simplified model then becomes

$$\dot{\mathbf{x}}(t) \approx \mathbf{A}(\psi(t))\mathbf{x}(t) + \mathbf{B}(\psi, \{\mathbf{r}_i(t)\}_i)\mathbf{u}(t) + \mathbf{g}^+ \tag{4.21}$$

with

$$\begin{aligned} \mathbf{A}(\psi) &= \begin{bmatrix} \mathbf{0_3} & \mathbf{0_3} & \boldsymbol{I_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{R}^\top(\psi) \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \end{bmatrix} \\ \mathbf{B}(\psi, \{\mathbf{r}_i(t)\}_i) &= \begin{bmatrix} \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \frac{1}{m}\boldsymbol{I_3} & \frac{1}{m}\boldsymbol{I_3} & \frac{1}{m}\boldsymbol{I_3} & \frac{1}{m}\boldsymbol{I_3} \\ \mathbf{I}^{w^{-1}}[\mathbf{r_1^w}]_\times & \mathbf{I}^{w^{-1}}[\mathbf{r_2^w}]_\times & \mathbf{I}^{w^{-1}}[\mathbf{r_3^w}]_\times & \mathbf{I}^{w^{-1}}[\mathbf{r_4^w}]_\times \end{bmatrix} \end{aligned} \tag{4.22}$$

and

$$\mathbf{u} = \begin{bmatrix} \mathbf{f_{r,1}^w} \\ \mathbf{f_{r,2}^w} \\ \mathbf{f_{r,3}^w} \\ \mathbf{f_{r,4}^w} \end{bmatrix}, \quad \mathbf{g}^+ = \begin{bmatrix} \mathbf{0_{6\times 1}} \\ \mathbf{g} \\ \mathbf{0_{3\times 1}} \end{bmatrix} \tag{4.23}$$

The system matrices depend only on ψ and $\{\mathbf{r}_i^w\}_i$. If these can be calculated beforehand, the system becomes linear time-varying. A desired reference trajectory $\mathbf{x}_{ref}(t)$ can be calculated ahead of time.

Assuming that $\mathbf{x}(t) \approx \mathbf{x}_{ref}(t)$, and alongside a heuristic for placing new footholds upon foot landing, this can be used to find approximate values for both $\psi(t)$ and $\{\mathbf{r}_i^w(t)\}_i$. The footstep placement is done using the Raibert Heuristic, as discussed in subsection 4.3.1. Thus, we get an LTV approximation of the system dynamics with $\boldsymbol{A}(t) \approx \boldsymbol{A}(\psi_{ref}(t)), \boldsymbol{B}(t) \approx \boldsymbol{B}(\psi_{ref}(t), \{\mathbf{r}_{ref,i}(t)\}_i)$.

4.2.2 Optimization problem

This simplified model lays the foundation for formulating an MPC controller. As in Di Carlo et al. (2018) a zero-order-hold integration scheme is employed for discretizing

the system, yielding the discrete-time dynamical system

$$\mathbf{x}[k+1] = \mathbf{A}_{k,d}\mathbf{x}[k] + \mathbf{B}_{k,d}\mathbf{u}[k] + \mathbf{g}_d^+ \tag{4.24}$$

with

$$\mathbf{A}_{k,d} = e^{\boldsymbol{A}(k\Delta t)\Delta t}, \quad \mathbf{B}_{k,d} = \left(\int_0^{\Delta t} e^{\boldsymbol{A}(k\Delta t)\tau} d\tau\right) \boldsymbol{B}(k\Delta t), \quad \mathbf{g}_d^+ = \Delta t \cdot \mathbf{g}^+ \tag{4.25}$$

It may be noted that due to the required time discretization of the system, and the control signal being generated by considering the system behavior over a time interval, the hybrid nature of the system is handled elegantly, simply by changing the corresponding system matrices through time in accordance with the gait pattern.

For the ground reaction forces to be physically realizable, they must obey certain constraints. Firstly, the normal component of the reaction force must be upward-pointing, as the foot cannot pull the ground closer. Secondly, the force must stay within the friction cone, parametrized by the static friction coefficient μ. The second corresponds to a nonlinear inequality constraint (friction cone) which is then linearized (friction pyramid). Lastly, all non-stance feet must generate zero force. These constraints are summarized as

$$\begin{aligned} \mathbf{C}_f\mathbf{f}_i[k] \leq \mathbf{0} \quad & \forall i \in \mathcal{S}_{st,k} \\ \mathbf{f}_i[k] = \mathbf{0} \quad & \forall i \in \mathcal{S}_{nst,k} \\ \forall k \in \{1, \ldots, N\} & \end{aligned} \tag{4.26}$$

where

$$\mathbf{C}_f = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & -\mu \\ -1 & 0 & -\mu \\ 0 & 1 & -\mu \\ 0 & -1 & -\mu \end{bmatrix} \tag{4.27}$$

$\mathcal{S}_{st,k}$ and $\mathcal{S}_{nst,k}$ are the index sets for the stance feet and non-stance feet respectively at timestep k. The system dynamics are imposed as equality constraints between time-adjacent system states.

The reference trajectory, being calculated simply from initial pose and desired velocities, may not always be a feasible trajectory. Thus, in stead of imposing equality constraints between the real and desired state trajectory, a penalty is imposed on the distance between the two in the objective function. Alongside an imposed cost on the squared norm of the reaction forces, one can summarize the optimization problem as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{f}} \quad & \|\mathbf{f}\|_2^2 + c2\|\mathbf{x} - \mathbf{x}_{ref}\|_2^2 \\ \text{s.t.} \quad & \mathbf{x}[k+1] = \mathbf{A}_{k,d}\mathbf{x}[k] + \mathbf{B}_{k,d}\mathbf{u}[k] + \mathbf{g}_d \\ & \mathbf{x}[1] = \mathbf{x}_0 \\ & \mathbf{C}_f\mathbf{f}_i[k] \leq \mathbf{0} \quad \forall i \in \mathcal{S}_{st,k} \\ & \mathbf{f}_i[k] = \mathbf{0} \quad \forall i \in \mathcal{S}_{nst,k} \\ & \forall k \in \{1, \ldots, N\} \end{aligned} \tag{4.28}$$

where

$$\begin{aligned} \mathbf{u}[k] &= \begin{bmatrix} \mathbf{f}_1[k]^\top & \mathbf{f}_2[k]^\top & \mathbf{f}_3[k]^\top & \mathbf{f}_4[k]^\top \end{bmatrix}^\top \\ \mathbf{x} &= \begin{bmatrix} \mathbf{x}[1]^\top & \ldots & \mathbf{x}[N]^\top \end{bmatrix}^\top \\ \mathbf{f} &= \begin{bmatrix} \mathbf{f}_1[1]^\top & \mathbf{f}_2[1]^\top & \mathbf{f}_3[1]^\top & \mathbf{f}_4[1]^\top & \mathbf{f}_1[2]^\top & \ldots & \mathbf{f}_4[N]^\top \end{bmatrix}^\top \end{aligned} \tag{4.29}$$

This optimization problem is solved recurrently, and output resulting ground reaction forces. As opposed to in Di Carlo et al. (2018), we choose here to use the corresponding acceleration as input to a whole-body-controller, in stead of converting the forces to torque directly. This choice will be elaborated upon in section 4.6. Solve times typically lie between $0.01s - 0.05s$, so reaction forces are updated quickly in order to respond both to unknown disturbances and to effects from modeling errors and simplifications.

The reference trajectory is calculated by considering the current position and yaw, and having these evolve by a given constant xy velocity and yaw rate. Desired body height is constant, and desired roll and pitch angles are always 0.

4.3 Swing leg control

4.3.1 Leg placement heuristic

The heuristic used to find new footholds for each foot is the so-called Raibert heuristic (Bledt et al. (2018)) and can be seen in Equation 4.30:

$$\mathbf{p}^b_{des} = \mathbf{p}^b_{ref} + \frac{T_{stance}}{2}\mathbf{v}_{des} \tag{4.30}$$

where $\mathbf{p}_{ref}$ is the reference (shoulder) position of the leg, $\mathbf{v}_{des}$ is the desired velocity and T_{stance} is the foot stance period.

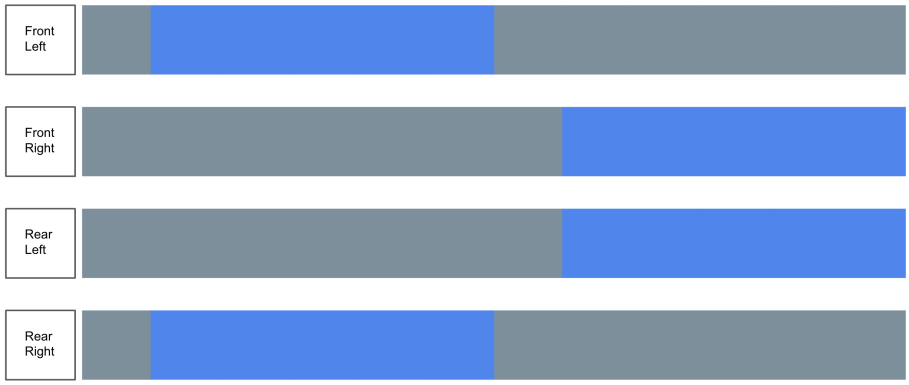


Figure 4.1: Gait graph for trotting gait. Swing phases are in blue, while stance phases are in gray

4.3.2 Swing leg trajectory design

The swing leg trajectory design is fairly simple: A linear interpolation is done between the old foothold and the new desired foothold in the body frame. Then, a sinusoidal term is added to the z-component in order to get the desired elevation in the step:

$$\mathbf{p}(\tau)^b = \mathbf{p}^b_{old} + (\mathbf{p}^b_{des} - \mathbf{p}^b_{old})\tau + \begin{pmatrix} 0 \\ 0 \\ h \cdot \sin(\pi\tau) \end{pmatrix} \tag{4.31}$$

4.4 Gaits

Three different gaits will be tested with the controller: Standing, trotting and ambling. Each gait is characterized by its gait graph, which is a diagram illustrating the placement of stance and swing phases for each leg through the gait cycle.

4.4.1 Standing

In the standing gait, all four legs are in stance phase for the duration of the gait period.

4.4.2 Trotting

In trot, diagonal legs share the same sequence of stance and swing phases. See Figure 4.1.

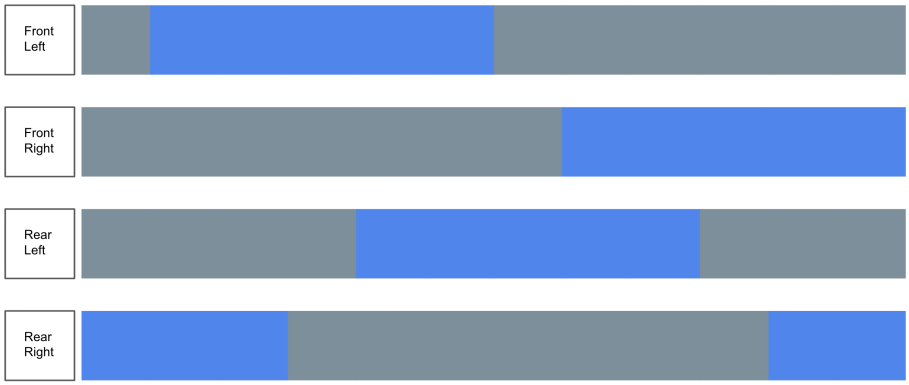


Figure 4.2: Gait graph for trotting gait. Swing phases are in blue, while stance phases are in gray

4.4.3 Ambling

In amble, the fore and rear legs switch stance and swing leg in an alternating fashion. See Figure 4.2.

4.4.4 Static and dynamic gaits

Gaits are typically split into two categories: Those which are statically stable, and those which are dynamically stable. Statically stable gaits are gaits where any configuration realized during a gait cycle is a feasible standing configuration, i.e. the walker can freeze in place at any point and not fall over. This corresponds to the robot's center of mass consistently residing within the convex hull spanned by its feet, referred to as the support polygon. For a stationary robot, this translates to a situation in which the ground's resultant normal force is able to completely cancel out the gravitational force without any resulting moment acting on the robot.

Dynamically stable gaits, on the other hand, are gaits in which this is not satisfied for portions of the gait cycle. This might be due to the center of mass simply being outside of the support polygon for parts of the gait, or the support polygon may be degenerate for parts of the gait – that is, it might be spanned by fewer than three feet, resulting in zero area.

While it is theoretically possible for the robot's center of mass to coincide with a given line or point, any real-world scenario or simulation utilizing numerical calculations will introduce errors or disturbances that makes this impossible as a matter of practicality. Thus, the robot is for portions of the gait "falling forwards" and requires the controller to make continual adjustments in order to not fall over. Dynamic gaits are thus considered to be more difficult to achieve, as well as less robust than static gaits. On the other hand, they are typically more energy efficient, and any sort of fast walking or running will be using a dynamic gait.

Both the trot and the ambling gait are examples of dynamic gaits. However, the amble may be more difficult to achieve: In the double support phase of the trot, the stance feet are always diagonal pairs, see Figure 4.1. Thus, the support line is fairly close to the center of mass, and the robot is as such closer to an unstable equillibrium. If the robot were to stop in the middle of such a double support phase, it would resemble a sort of pendulum close to its unstable upward equillibrium: While unstable, the repulsive force acting on the pendulum is fairly modest.

During parts of the ambling gait, however, there are double support phases in which the stance legs are on the same side, see Figure 4.2. In these phases, the robot more resembles a pendulum in its sideways position, with close to the maximal amount of torque acting on it. This might make the ambling gait slightly more difficult to realize.

4.5 Implementation notes

4.5.1 Code base

The source code for this project is written in C++ and builds and expands upon the code base developed in Ghansah and Thorseth (2021). The full code base is open source and hosted in a git repository on Github[1]. The implementation done in this thesis can be found in the `mpc_control_feature` branch, and is located in the catkin workspace structure in the `convex_mpc_controller` ROS package.

4.5.2 ROS

The control system is written as c++ ROS-nodes. ROS (Robot Operating System) is a powerful, open source tool for writing robotic software, and provides a collection of tools, libraries and conventions for facilitating the development of complex robotic systems.

4.5.3 Simulation

The robot is simulated using the open source simulation environment Gazebo. Gazebo was chosen for its versatility as well as for its integration with ROS.

4.6 Control system overview

The method used for controlling the robot is based on work done in Di Carlo et al. (2018). There, ground reaction forces are taken directly from the solution from the MPC, converted to joint torques and given as input to the stance legs. A different controller is used to control swing legs.

However, swing legs also generate torque when moving, which is not accounted for when calculating reaction forces in the MPC. Thus, we have decided to utilize a whole-body

[1] `https://github.com/Norwegian-Legged-Lab/Tetrapod-Robot/tree/mpc_control_feature`

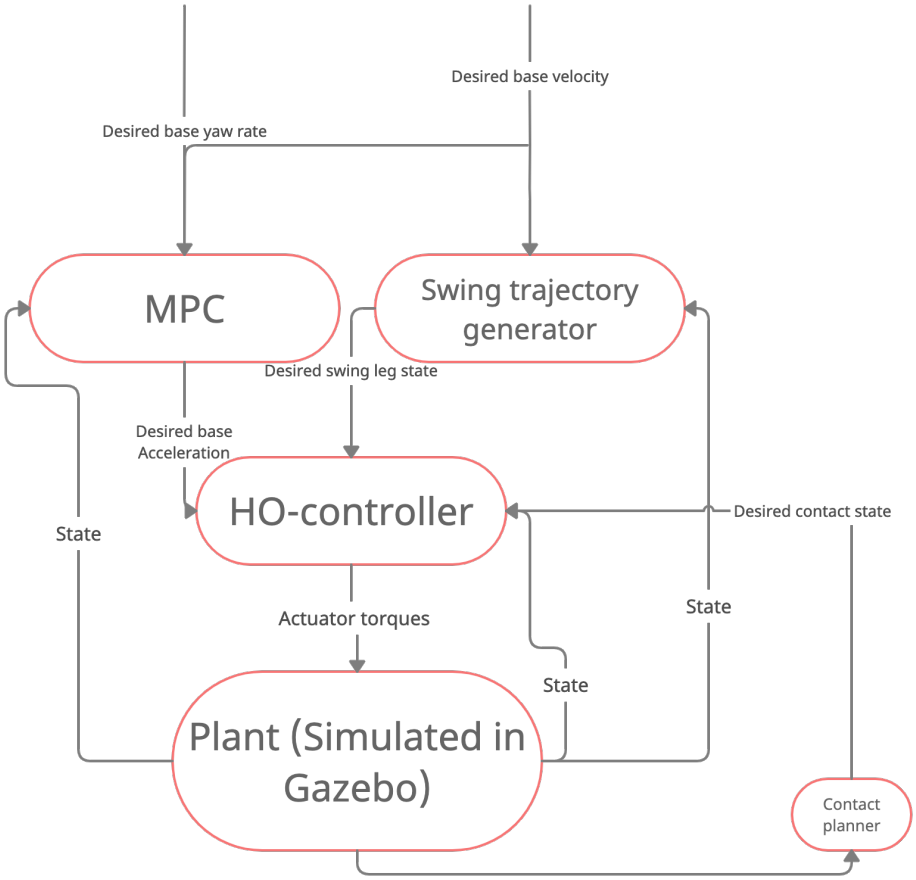


Figure 4.3: Block diagram of control system

controller, which is given desired base acceleration as well as the desired states of all feet.

Calculating the desired base acceleration using MPC gives a desired acceleration which is realizable with the current number and placement of stance feet, while also resulting in a feasible trajectory given future footholds.

On the other hand, the use of a whole-body controller to achieve the desired acceleration makes it possible to take into account and compensate for the torque generated by swing-legs. The whole-body controller used here was described in Dario Bellicoso et al. (2016) and implemented for ASTRo in Ghansah and Thorseth (2021). For a block diagram of the full control system, see Figure 4.3

5

Results and discussion

5.1 Convex MPC

5.1.1 Model parameters

Parameters related to the gait can be seen in Table 5.1. X-velocities relate to the forward-moving scenarios for the gaits. Parameters related to the optimization problem can be seen in Table 5.2. Costs are given for one time step of the discretization, and are identical across time steps. All force terms have the same cost associated, while the costs associated with reference error are the diagonal elements of the diagonal reference error cost matrix.

5.1.2 Simulated body trajectories

In what follows, the full control pipeline is used to control a simulated ASTRo. Simulation is performed in the open-source simulation framework Gazebo. The controller is tested with standing, as well as a trotting and an ambling gait. For the trotting and ambling gaits, both a walking-in-place and a forward walking scenario is tested.

The controller exhibited some performance issues when simulating at real-time. Thus, all simulations are performed at half of real-time speed, except for the forward amble, which is performed at 30% of real-time speed. Fairly long time series are shown in order to demonstrate the ability to produce stable walking over longer periods. All gait periods are 0.5 seconds in simulated time.

$\mathrm{h}_{swing}[m]$	$\mathrm{T}_{gait}[s]$	$\mathrm{v}_x^{ref,trot}[m/s]$	$\mathrm{v}_x^{ref,amble}[m/s]$	$\mathrm{v}_y^{ref}[m/s]$	$\dot{\phi}^{ref}[rad/s]$
0.02	0.5	0.4	0.2	0	0

Table 5.1: Parameters determining the gaits

Δt[s]	0.05
T_{sim}[s]	0.5
$cost_f$	1
$cost_x$	$5 \cdot 10^5$
$cost_y$	$5 \cdot 10^5$
$cost_z$	$5 \cdot 10^6$
$cost_\psi$	$5 \cdot 10^6$
$cost_\theta$	$5 \cdot 10^5$
$cost_\phi$	$5 \cdot 10^5$
$cost_{\dot{x}}$	$5 \cdot 10^5$
$cost_{\dot{y}}$	$5 \cdot 10^5$
$cost_{\dot{z}}$	$5 \cdot 10^5$
$cost_{\omega^b_x}$	$5 \cdot 10^4$
$cost_{\omega^b_y}$	$5 \cdot 10^4$
$cost_{\omega^b_z}$	$5 \cdot 10^4$

Table 5.2: Parameters configuring the optimization problem

Base height, roll and pitch are the states which are important to keeping the robot's posture stable, as the robot's dynamics are not invariant to a shift along these dimensions. On the other hand, the horizontal position of the base as well as its yaw are dimensions for which a constant shift does not affect the robot's dynamics. Thus, roll, pitch and height values are plotted against references to indicate the controller's ability to keep the gait stable (in the sense of not falling over).

For movement in the xy-plane, x-velocity is the only non-zero reference given to the controller. Thus, x-velocity is plotted against its reference value. Torque exertion is important for real-world feasibility: Inordinately high torque values might indicate that while the controller works in simulation, physical limitations would likely bar it from working on hardware.

Standing

As expected, standing is handled fairly well. As seen in Figure 5.1 there is a very slight constant deviation in the vertical position of the body of roughly $0.6mm$ RMSE, see Table 5.3. This is likely due to the trade-off done in the MPC between minimizing ground reaction forces – in particular normal force – and minimizing the distance to the nominal trajectory. Roll and pitch errors are practically zero with the exception of some transients. These are possibly due to higher solve times for the MPC. Already when standing, hip pitch torque exertion is moderately high, see Figure 5.2.

This illustrates one of the potential shortcomings of sprawling quadrupeds: A wide stance implies a large component of the distance from actuators to footholds lying in the transverse plane. This again implies a large cross product between the arm and the normal force, resulting in a larger torque exertion for a given produced normal ground force. This is more prominent for the hip pitch than for the knee pitch, as the foot-to-actuator distance

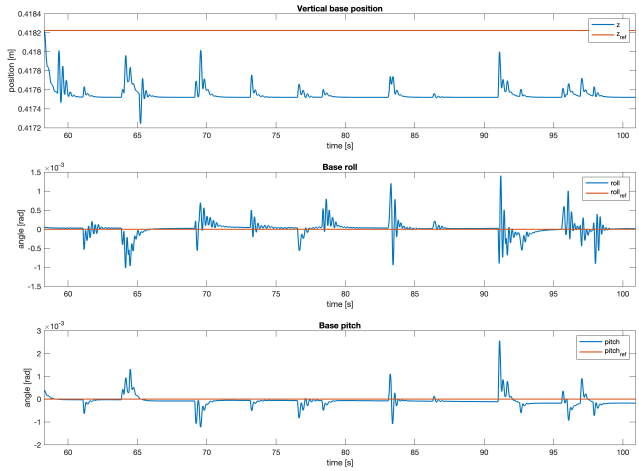


Figure 5.1: Selected state trajectories for standing

	z_{RMSE}[m]	ψ_{RMSE}[rad]	θ_{RMSE}[rad]	$\dot{x}_{RMSE}$[m/s]
Stand	$6.731 \cdot 10^{-4}$	$1.961 \cdot 10^{-4}$	$2.584 \cdot 10^{-4}$	$5.669 \cdot 10^{-4}$
Stat. trot	$1.292 \cdot 10^{-3}$	$1.340 \cdot 10^{-3}$	$2.950 \cdot 10^{-4}$	$2.842 \cdot 10^{-3}$
Forward trot	$1.630 \cdot 10^{-3}$	$3.765 \cdot 10^{-3}$	$5.867 \cdot 10^{-3}$	$2.058 \cdot 10^{-2}$
Stat. amble	$3.210 \cdot 10^{-2}$	$1.237 \cdot 10^{-1}$	$2.404 \cdot 10^{-2}$	$1.318 \cdot 10^{-1}$
Forward amble	$1.501 \cdot 10^{-2}$	$2.390 \cdot 10^{-2}$	$1.598 \cdot 10^{-2}$	$6.365 \cdot 10^{-2}$

Table 5.3: RMSE values for selected system states

is greater.

Trotting

For trotting, there is one stationary and one forward-moving scenario.

As seen in Figure 5.3, there are slight periodic oscillations affecting all states. This is most likely explained by the impacts of feet against the floor when establishing a new foothold, causing a periodic impulse-like disturbance. These effects – while modeled in the Gazebo simulator – are not in the model-based HO controller and is thus not compensated for.

However, there also seems to be more slowly-varying deviations in the base roll. These might stem from non-optimal tuning of the MPC, so that aggressive corrections towards the roll reference are negated by a 0-reference in angular velocity in the transverse plane. However, the order of magnitude is sufficiently small for these oscillations to not be of great concern. Base x-velocity is maintained fairly well at its 0 reference, with slight oscillations here as well. As seen in Figure 5.4 actuator torques are still reasonable, mostly

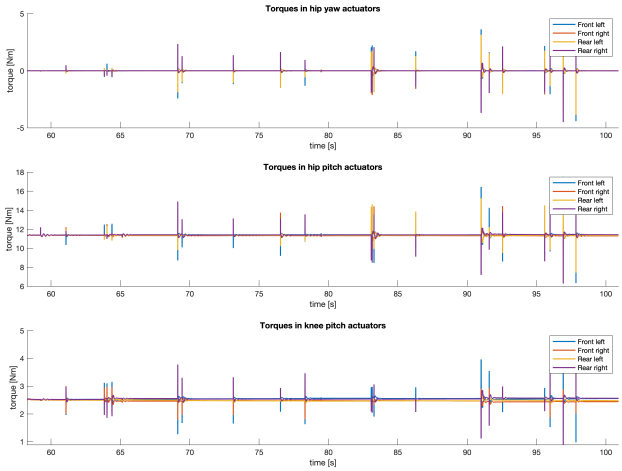


Figure 5.2: Actuator torques for standing

staying below $20Nm$.

In the forward-moving scenario, we see from Figure 5.5 that both vertical position and roll continues to lie close to their references, apart from the aforementioned slight periodic oscillations. What is notable here, though, is that the base pitch lies constantly slightly above its 0 reference, corresponding to the robot leaning slightly forward when moving forward.

This might be partially explained by the opposing torque applied to the base during the swing phase of each leg. The controller tracks the reference fairly closely also in this case. As seen in Table 5.3, both roll and height RMSE increase roughly one order of magnitude from the standing case to both trotting cases. This is expected both due to the more dynamic nature of the movement, as well as the frequent disturbances from foot-ground impacts.

Ambling

For ambling there are two scenarios as well, one in place and one moving forward. Already for stationary ambling, we see degradation in controller performance. The periodic oscillations have a larger amplitude, and the roll angle is consistently below 0, indicating some stationary error superimposed on the oscillations seen in previous plots. Particularly the stationary error in roll is indicative of poorer controller performance, as the robot is unable to stay level in the roll direction but leans consistently to one side. Oscillatory errors in x-velocity are also roughly one order of magnitude greater compared to those in Figure 5.3. At the end of the time series in Figure 5.9, it can be seen that the controller fails to produce a stable gait, so that the robot falls over and collapses.

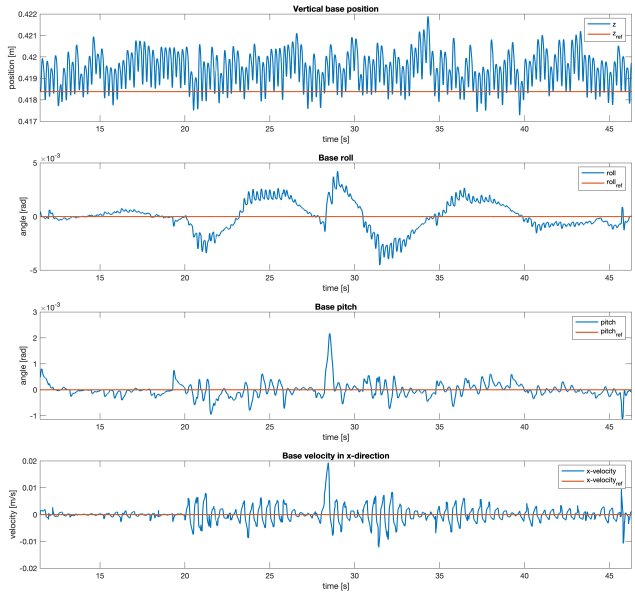


Figure 5.3: Selected state trajectories for trotting in place

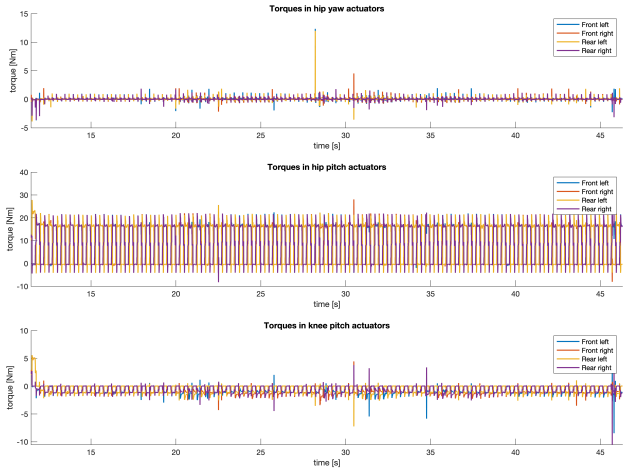


Figure 5.4: Actuator torques for trotting in place

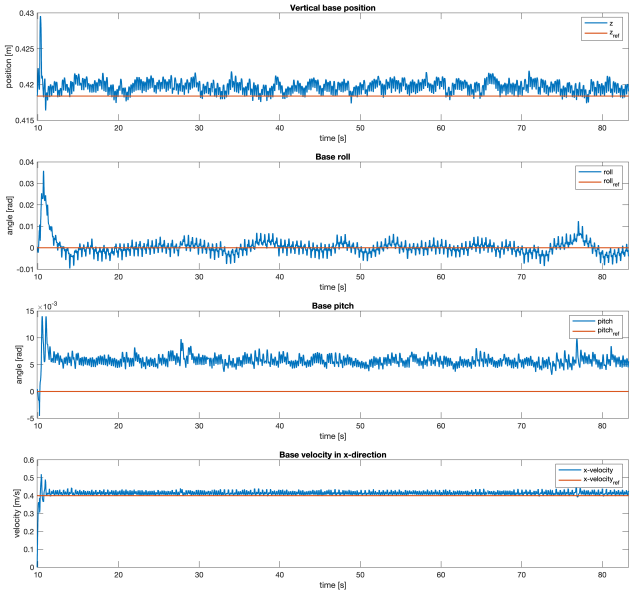


Figure 5.5: Selected state trajectories for forward trotting

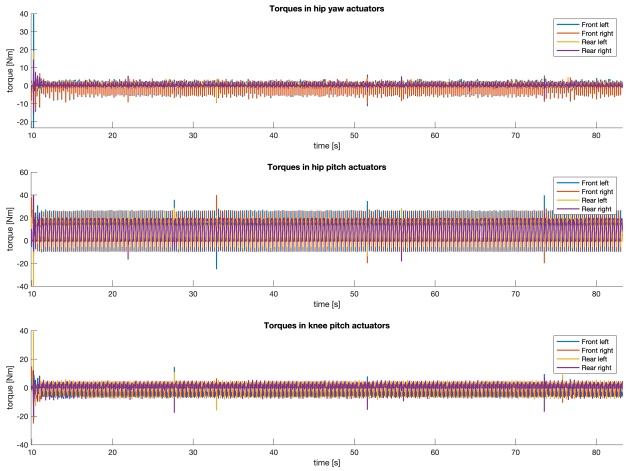


Figure 5.6: Actuator torques for forward trotting

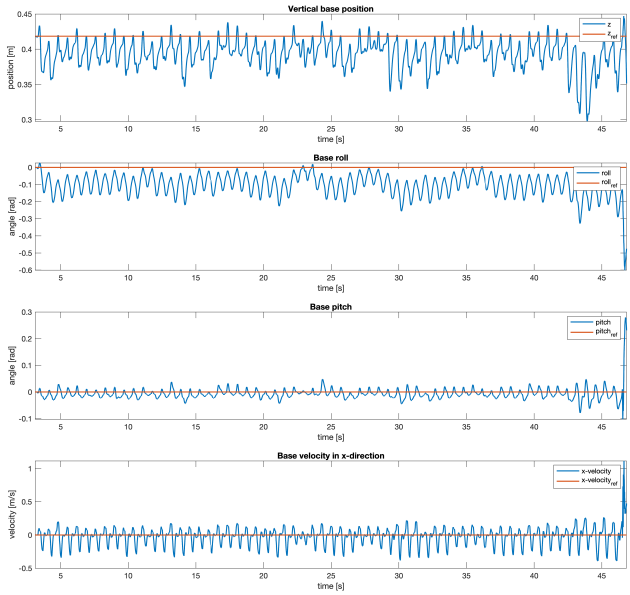


Figure 5.7: Selected state trajectories for ambling in place

Additionally, we see from Figure 5.8 that the torque limits of $40Nm$ are saturated at several points. This likely contributes to the degraded controller performance, as the inequality constraints on torque limit the controller's ability to fulfill lower-priority tasks such as foot and base reference tracking. RMSE consistently increase roughly one order of magnitude from the trotting scenarios, see Table 5.3. This demonstrates the hypothesized difference in difficulty in producing a trotting and ambling gait put forth in **??**.

For the forward moving scenario, the undesirable tendencies seen in the stationary case are still present. While magnitude of both oscillatory and stationary errors are slightly smaller, this has to be seen in light of the simulation happening at a lower speed, effectively giving the controller a higher operating frequency than in previous cases. This further reduction in simulator speed was due to the inability of the controller to produce a forward ambling gait at the same real time factor as for the other simulations.

As seen in Figure 5.10 the body torques saturate for some actuators for large portions of the simulation. Here as well, this might account for some of the degraded performance of the controller.

The tendency for the robot to veer to one side for more demanding scenarios were also present in the trotting scenarios at real-time simulation. It is possible that this betrays some flaw in either the formulation or the current implementation of the MPC.

It is possible that both the periodic oscillatory errors and the saturation of torque could

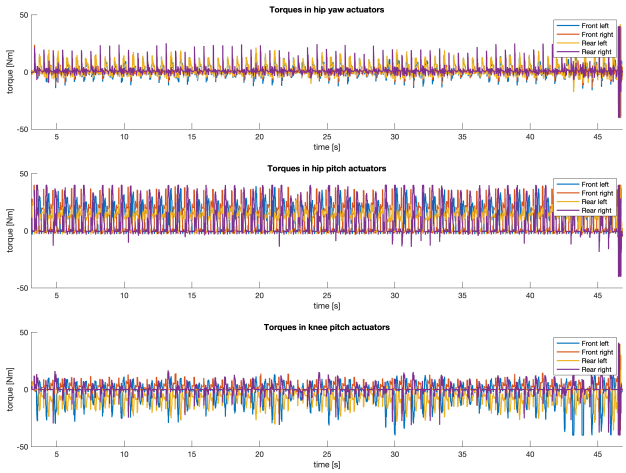


Figure 5.8: Actuator torques for ambling in place

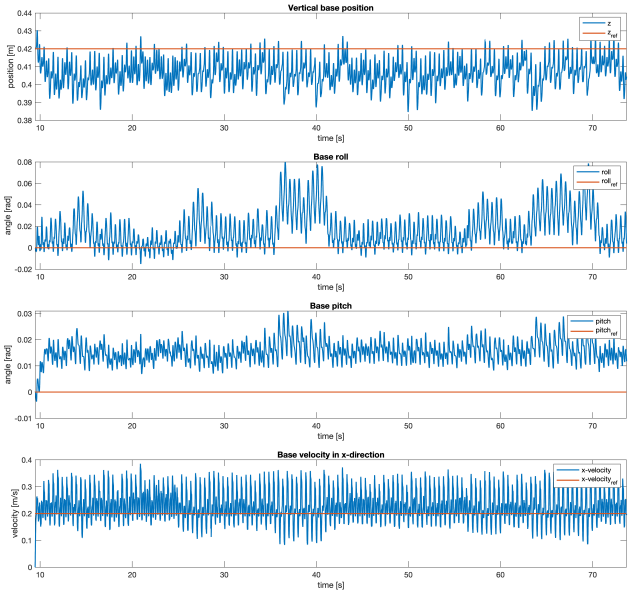


Figure 5.9: Selected state trajectories for forward ambling

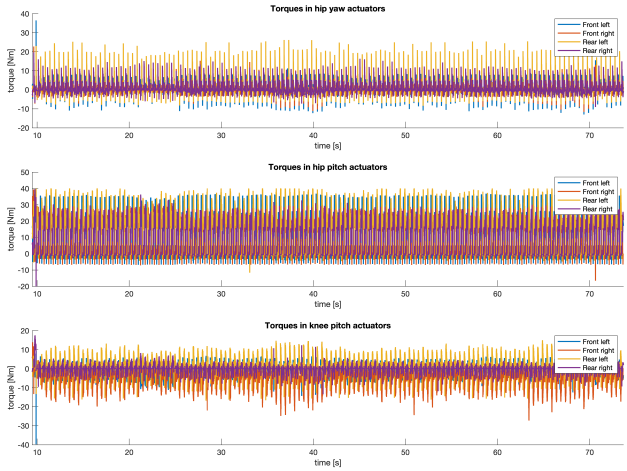


Figure 5.10: Actuator torques for forward ambling

both be amended by a more carefully designed swing leg trajectory, where acceleration and velocity as well as position were continuous. One way to achieve this could be by using Bézier curves, which are extensively used in the field of trajectory planning (Zhang et al. (2019); Askari et al. (2016); Ingersoll et al. (2016)).

6

Conclusion and further work

6.1 Conclusion

The topic of quadrupedal legged locomotion has been a hot research topic, and in the last few decades we have seen a substantial body of research produced in the field. Furthermore, the last five years has seen the emergence of the first quadrupedal robots in the commercial sphere, with Boston Dynamics' Spot having been made available for sale and other companies following suit. This seems to indicate that legged robotics as a field is in the process of escaping the purely academic sphere and is starting to mature enough for serious commercial applications.

As most research has been focused on mammalian quadrupeds, an examination is warranted into whether the methods yielded are applicable to quadrupedal robots with other leg configurations, which may have their distinct set of advantages. Therefore, in this paper, the question has been posed of whether methods from this rich body of research is straightforwardly transferable to quadrupeds with a sprawling leg configuration.

In this paper, we examined a number of methods used for dynamic gait generation in mammalian quadrupeds. Two such methods were selected for further elaboration, and one was chosen for implementation and testing through simulation. A method utilizing a single body rigid dynamics model simplification and a convex MPC formulation in series with a HO controller was implemented and tested on the sprawling quadruped robot ASTRo, developed and built in Ghansah and Thorseth (2021). The simulation was performed in the robot simulation framework Gazebo. The simulation was performed at half of real time speed for controller performance reasons.

For a trotting gait the results were promising, as the controller managed to stabilize the gait over longer periods of time and to follow a given reference velocity with a low error (0.021 m/s RMSE). However, the simulations performed with an ambling gait were more ambiguous, with an RMSE of 0.12 radians in the roll axis for the stationary case. A further

reduction in simulation speed was necessary to produce a successful forward ambling gait. As the controller could not stabilize the gait at real time, implementation on hardware is infeasible for now. Overall, the results show promise for the use of the method for sprawling locomotion. However, the difficulties in stabilizing more dynamic gaits suggests that the transfer of this type of controller may not be completely straightforward, but may in stead require further modification to successfully produce a wide range of gaits.

Additionally, another control method relying on full-order hybrid models of the system was examined. This method relies on an extension of the concept of zero dynamics to apply for hybrid systems, and utilizes IO-linearization approaches to control the full-order dynamics to a zero-dynamics manifold. The desired output trajectories are found through posing a nonlinear optimization problem utilizing the complete dynamics of the system and a modified Hermite-Simpson collocation method.

Due to the generality of the framework and the lack of model simplifications, it is possible that this method may prove more successful at capturing the system dynamics and controlling the system where methods utilizing substantial simplifications may fall short.

6.2 Further work

Looking forward, there are some ways to address the limitations of the implementation chosen for this thesis. Firstly, while the MPC manages to recalculate appropriate ground reaction forces recurrently in order to stabilize the robot gait, both foot placement and gait scheduling is static and done in a fairly simplistic manner.

In order to ensure a robust, flexible and efficient gait, animals adapt both ground reaction forces, footstep placement and gait scheduling to unexpected obstacles or disturbance. In the future, it would therefore be interesting to attempt to merge a method such as the one implemented here with reactive gait scheduling methods such as the contact time modulation method introduced in Yeom and Bae (2021) and discussed in section 2.3.

Additionally, one could argue that the challenges faced in certain highly dynamic gaits warrants a look at more sophisticated, full-model methods such as the HZD approach examined in section 4.1. Especially higher-order effects such as coriolis and centrifugal terms of the body and torque generated by leg movement start to become relevant for higher velocities, and simplified models which do not account for these effects may simply not be the best choice of method in these cases.

However, methods such as the last one rely on the solving of complex high-dimensional NLPs. Hence they have the drawback that an essential part of the method consists of offline trajectory generation, while the online control aspect merely consists of stabilizing the system in some vicinity of such a pre-defined trajectories. Such methods will likely be more vulnerable to large disturbances which perturb the system away from the orbitally stable vicinity around such a trajectory. Thus, in order to achieve flexible and robust control of locomotion, further development of fully online methods such as the one implemented here may be the more interesting future direction of research.

All in all, we strongly believe that the coming decades will witness the widespread adop-

tion of legged robotics in the commercial sector, and that there are niches which are best suited for mammalian and sprawling quadrupeds respectively.

Bibliography

Ames, A.D., 2014. Human-Inspired Control of Bipedal Walking Robots. IEEE Transactions on Automatic Control 59, 1115–1130. doi:10.1109/TAC.2014.2299342. conference Name: IEEE Transactions on Automatic Control.

Askari, A., Mortazavi, M., Talebi, H., Motamedi, A., 2016. A new approach in UAV path planning using Bezier–Dubins continuous curvature path. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 230, 1103–1113. URL: https://doi.org/10.1177/0954410015603415, doi:10.1177/0954410015603415. publisher: IMECHE.

Bledt, G., Powell, M.J., Katz, B., Di Carlo, J., Wensing, P.M., Kim, S., 2018. MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2245–2252. doi:10.1109/IROS.2018.8593885. iSSN: 2153-0866.

Dario Bellicoso, C., Gehring, C., Hwangbo, J., Fankhauser, P., Hutter, M., 2016. Perception-less terrain adaptation through whole body control and hierarchical optimization, in: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp. 558–564. doi:10.1109/HUMANOIDS.2016.7803330. iSSN: 2164-0580.

Di Carlo, J., Wensing, P.M., Katz, B., Bledt, G., Kim, S., 2018. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1–9. doi:10.1109/IROS.2018.8594448. iSSN: 2153-0866.

Ghansah, A., Thorseth, P., 2021. Design and Control of a Torque Controllable Quadrupedal Robot - A study on the development of ASTRo. Ph.D. thesis. doi:10.13140/RG.2.2.11840.66569.

Hamed, K.A., Wen-Loong, Ames, A.D., 2018. Dynamically Stable 3D Quadrupedal Walking with Multi-Domain Hybrid System Models and Virtual Constraint Controllers.

arXiv:1810.06697 [math] URL: http://arxiv.org/abs/1810.06697. arXiv: 1810.06697.

Hereid, A., Hubicki, C.M., Cousineau, E.A., Ames, A.D., 2018. Dynamic Humanoid Locomotion: A Scalable Formulation for HZD Gait Optimization. IEEE Transactions on Robotics 34, 370–387. doi:10.1109/TRO.2017.2783371. conference Name: IEEE Transactions on Robotics.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C.D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A., Hoepflinger, M., 2016. ANYmal - a highly mobile and dynamic quadrupedal robot, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 38–44. doi:10.1109/IROS.2016.7758092. iSSN: 2153-0866.

Ingersoll, B., Ingersoll, J., DeFranco, P., Ning, A., 2016. UAV Path-Planning using Bezier Curves and a Receding Horizon Approach. doi:10.2514/6.2016-3675.

Isidori, A., 2011. The zero dynamics of a nonlinear system: FrOm The Origin To the latest progresses of a long successful story, in: Proceedings of the 30th Chinese Control Conference, pp. 18–25. ISSN: 2161-2927.

Katz, B., Carlo, J.D., Kim, S., 2019. Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control, in: 2019 International Conference on Robotics and Automation (ICRA), pp. 6295–6301. doi:10.1109/ICRA.2019.8793865. iSSN: 2577-087X.

Khalil, H.K., 2002. Nonlinear Systems. 3 ed., Prentice Hall. Google-Books-ID: t_d1QgAACAAJ.

Ma, W.L., Hamed, K.A., Ames, A.D., 2019. First Steps Towards Full Model Based Motion Planning and Control of Quadrupeds: A Hybrid Zero Dynamics Approach, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5498–5503. doi:10.1109/IROS40897.2019.8968189. iSSN: 2153-0866.

Park, H.W., Kim, S., 2015. Quadrupedal galloping control for a wide range of speed via vertical impulse scaling. Bioinspiration & Biomimetics 10, 025003. URL: https://iopscience.iop.org/article/10.1088/1748-3190/10/2/025003/meta, doi:10.1088/1748-3190/10/2/025003. publisher: IOP Publishing.

Vukobratovic, M., Borovac, B., 2004. Zero-Moment Point - Thirty Five Years of its Life. I. J. Humanoid Robotics 1, 157–173. doi:10.1142/S0219843604000083.

Yeom, H., Bae, J., 2021. A dynamic gait stabilization algorithm for quadrupedal locomotion through contact time modulation. Nonlinear Dynamics 104, 2275–2289. URL: https://doi.org/10.1007/s11071-021-06376-5, doi:10.1007/s11071-021-06376-5.

Zhang, C., An, H., Wei, Q., Ma, H., 2019. Foot Trajectory Planning Method with Adjustable Parameters for Complex Environment, in: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1151–1157. doi:10.1109/ROBIO49542.2019.8961767.