

# UNIVERSITY OF TWENTE.

Biomechanical Engineering  
Faculty of Engineering Technology

## MOMENTUM-BASED CONTROL FOR COMPLIANT LINK QUADRUPEDS

Sander W.S. Oosterveld  
Master's Thesis  
23-02-2023

DOCUMENT NUMBER: BE-914

---

**Assessment Committee:**

prof. dr. ir. Herman van der Kooij  
*Chairman*

dr. ir. Ronald G.K.M. Aarts  
*External member*

dr. ir. Arvid Q.L. Keemink  
*Daily supervisor*

Biomechanical Engineering  
Faculty of Engineering Technology  
University of Twente  
P.O. Box 217  
7500 AE Enschede  
The Netherlands

---



## Preface

Dear reader,

This is my Master’s thesis titled “Momentum-based Control for Compliant Link Quadruped”. It was written to fulfil the graduation requirements for Mechanical Engineering at the University of Twente in Enschede, The Netherlands. I worked on it for one year, from February 2022 to February 2023, and my defence will be held on February 23, 2023.

I am optimistic about the future of technology and believe that robots can make a significant impact in improving our lives by taking over mundane or dangerous tasks. If this thesis contributes to that goal even slightly, I will be content. The journey to complete this thesis was both rewarding and challenging, and I am grateful for the support I received.

I would like to express my gratitude to my supervisor, Dr. Ir. Arvid Keemink, for his continuous support in both technical and academic fields. He challenged me, as I expected, and I am grateful for his help. I would also like to thank Ir. Ander Vallinas Prieto, for the many whiteboards we filled together, which inspired parts of this thesis. Finally, I would like to thank current and former members of the Student Balance Control group for their inspiring presentations and critical questions during our monthly meetings.

My family and friends got me through this thesis, both in the easy and the hard times, for which I am forever grateful. My sister, Wija, deserves a special mention for proofreading my thesis more often than I did myself. And even though our daily morning calls felt a little early on some days, they made sure I got up and got to work every day. I would also like to thank the people of *Campuslaan 25*, *Solar Boat Twente*, *V.V. Harambee*, *H.V. Ockham*, *the PCV Group*, and *Sport Umbrella Twente* for giving me the greatest 6 years of my life this far.

Enjoy reading this thesis.

Sander Oosterveld  
Enschede, February 8, 2023

*“Nature has given us all the pieces required to achieve exceptional wellness and health, but has left it to us to put these pieces together”*

Diane McLaren

## Abstract

This thesis examines the implications of incorporating compliant links in two critical areas of legged robot control: state estimation and momentum-based control. To this end, we consider a planar quadruped with a compliant spine. In the first part, we investigate the influence of the compliant spine on state estimation, an essential component of a model-based controller. We find that the unmodelled compliant spine degrades the state estimation. Fortunately, the state estimator can be improved by including the compliant link without the need for additional sensors. We demonstrate that the state estimator is more accurate in estimating the flexible coordinate and the center of mass position when it considers two or more flexible modes.

In the second part, we combine this state estimator with a momentum-based control. This controller is designed to prevent the robot from falling when it lifts the legs to move around. We show that neglecting the compliant spine has a detrimental effect on the controller, eventually causing the robot to fall. This behaviour can be avoided when the momentum-based controller considers one flexible mode. However, the controller is not robust when more flexible modes are added or when large parameter variations are induced. This highlights the importance of including the compliant spine in the controller, but also the need for further development to ensure robustness.

# Table of Contents

Preface	ii
Abstract	iii
Table of Contents	iv
<b>I Introduction</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Research Goal . . . . .	2
1.2 Structure of Thesis . . . . .	3
1.3 Statement of Use of Artificial Intelligence . . . . .	3
<b>2 Literature Review</b>	<b>4</b>
2.1 State Estimation . . . . .	4
2.1.1 State Estimation of Flexible Manipulators . . . . .	4
2.1.2 State Estimation of Legged Robots . . . . .	5
2.2 Control of (Compliant) Legged Robots . . . . .	5
2.3 Modelling of Flexible Multibody Systems . . . . .	6
<b>3 Preliminaries</b>	<b>8</b>
3.1 Modelling in the floating frame formulation . . . . .	8
3.1.1 Beams in the floating frame . . . . .	8
3.1.2 Combining floating frame elements . . . . .	11
3.1.3 Actuation in the Floating Frame Formulation . . . . .	12
3.1.4 Validation of the Floating Frame Formulation . . . . .	13
3.2 Null spaces . . . . .	15
3.2.1 Null space for underactuated systems . . . . .	15
3.2.2 Null space for Hierarchical Control . . . . .	16
<b>II Papers</b>	<b>18</b>
<b>4 State observation of Compliant Link Floating Base Robots</b>	<b>19</b>
<b>5 Momentum-Based Control for a Quadruped with a Compliant Spine</b>	<b>27</b>
<b>III Conclusion</b>	<b>38</b>
<b>6 Conclusion</b>	<b>39</b>
<b>7 Future work</b>	<b>40</b>

<b>Bibliography</b>	<b>41</b>
<b>A Extending the Floating Frame Beam Element</b>	<b>47</b>
A.1 Add damping . . . . .	47
A.2 Split bodies . . . . .	48
<b>B Ground Contact Model in the Floating Frame Formulation</b>	<b>49</b>

# Part I

## Introduction

# Chapter 1

## Introduction

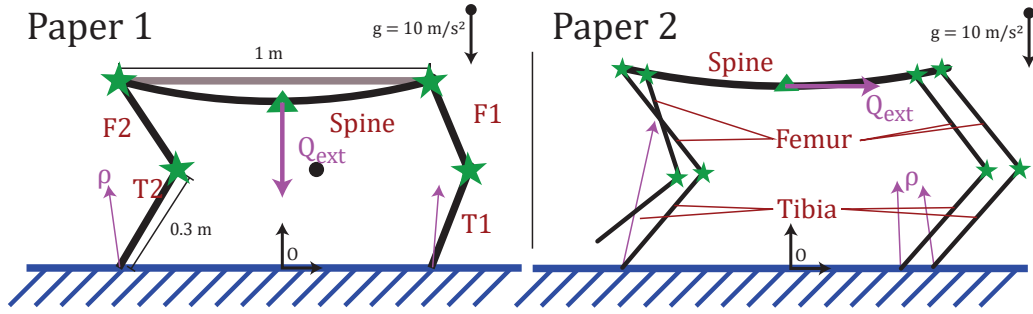
### 1.1 Research Goal

Staff shortages will continue to increase in the coming years, causing significant issues as early as 2030 [1]. Not only will the economy grow, but the fertility rate will also diminish due to the ageing population [2], making it essential to ensure that people reach their full potential. Automation can help maximize potential, as the same job can then require fewer workers. The second and third industrial revolutions have demonstrated that when more work is done by machines, the efficiency of the process improves, the number of workers decreases, and the quality of life increases [3].

For a robot to conduct more tasks, it needs to be almost as versatile as a human. The current infrastructure is designed for humans, with stairs, ladders, and uneven ground. This is especially true in places where people do manual labour, such as construction and road works. For a robot to assist in these locations, it would have to be able to traverse the same terrain. *Wieber et al.* suggest that a legged robot is the most versatile for navigating irregular terrain, as each leg can move independently. This independence of the legs, however, presents unique control challenges for legged robots. One such challenge is the redundancy of the actuators of a legged robot, similar to the abundance of control options available for human motion. This leads to an infinite number of possibilities for controlling the robot [4]. Additionally, a legged robot can only move when a leg disconnects from the ground, resulting in varying numbers of contact points during a walking cycle.

Currently, most legged robots use purely rigid links, as much of the analysis in legged robots is done for rigid systems. Flexible links, however, can benefit the performance and cost of the robot. Flexible links can use less material or cheaper materials, directly reducing costs. A lighter robot also requires less motor force, allowing either cheaper motors or faster robot speeds. Link flexibility also results in a lower impact force (compared to joint flexibility) which makes the robot safer in cases of human-robot interaction [5]. Finally, compliant links can improve the performance, for example, animals such as cheetahs have a flexible spine to increase running efficiency and speed. A similar design could be used in legged robots to move faster and more energy efficient[6].

This thesis focuses on the latter use-case, where a quadruped has a compliant spine. The problem is simplified by applying the theory to a planar version of a quadruped where only the spine is a flexible link, as shown in Fig. 1.1. Moreover, this planar robot knows the position of one point on the flexible spine as well as the global ground reaction forces. Such a position sensor would normally require the integration of different sensors such as Inertial Measurement Units (IMUs) and cameras [7], while the global ground reaction forces would require a coordinate transformation [8]. The system, however, will not have additional sensors or actuators to control



**Figure 1.1:** The models used in the different papers, both, represent a planar quadruped with a flexible spine.

the flexible modes.

The challenge of controlling this quadruped with a compliant spine is split into two parts. First, the (flexible) states of the robot must be estimated with only the sensors, as shown in Fig. 1.1. These states should then be used in the control of the robot. For legged robots, momentum-based control has been successfully applied to control the balance of rigid legged bipeds and quadrupeds [9–13]. Thus, the second challenge is to apply this momentum-based control to the simplified quadruped, and the research objective can be split into two parts, each with three research questions:

1. Develop a state observer for a quadruped with a flexible spine.
  - (a) What are current advances in state estimation of (compliant) legged robots?
  - (b) How does a compliant spine influence the quality of the state estimation?
  - (c) How should a state estimator be modified to include a compliant spine for a planar quadruped?
2. Develop a momentum-based control framework for a planar quadruped with a compliant spine.
  - (a) What is momentum-based control, and why is it relevant?
  - (b) What modifications should be done to a momentum-based controller to apply it to a planar quadruped with a compliant spine?
  - (c) How robust is the momentum-based controller when applied to a planar quadruped with a compliant spine?

## 1.2 Structure of Thesis

This thesis is divided into three parts, the research questions will be answered in the two papers found in Part II. But, before the papers there are more two chapters in this part. Chapter 2 will present existing literature relevant to the papers. This chapter answers the first subquestion for each of the research questions. In addition, Chapter 2 defends the modelling method used by the papers in this thesis. Chapter 3 present the preliminaries required to understand both papers. Even though the papers should be understandable without this knowledge, this chapter clarifies the work in the papers. The final part presents the combined conclusions from Chapter 2 and the papers in Part II, as well as propose future work.

## 1.3 Statement of Use of Artificial Intelligence

A large language model, ChatGPT (version January 2023) [14], was utilized for providing suggestions in restructuring text and for performing grammar and spelling checks. The output was thoroughly reviewed to ensure the author’s accountability for the content in this thesis.



# Chapter 2

## Literature Review

This chapter presents a literature review of three different subjects. The first section discusses State Estimation and is related to the subquestion: *What are current advances in state estimation of (compliant) legged robots?*. The second section goes into the control of legged robots and how this relates to the momentum-based control used in the papers, addressing the subquestion: *What is momentum-based control, and why is it relevant?*. The final section is not related to research questions, instead focusing on different formulations for flexible multibody systems.

### 2.1 State Estimation

States are an essential part of model-based control, as it allows us to predict the motion of a robot. However, these states often cannot be measured directly. For many mechanical systems, the state vector is a combination of the velocity and position of a body, while most sensors only measure positions, such as visual detection, strain gauges, or rotary encoders. Numerical differentiation can be used to compute the velocity from the position, but this generally results in an amplified noise or time delay [15].

State estimation is the field of combining the information from different sensors and possibly a dynamic model to generate the best possible estimation of the state vector. Proposed in 1997 by *Julier*, the Extended Kalman filter is still the most widely used optimal state estimator [16]. However, there are other options such as the Unscented Kalman Filter [17], the Particle Filters [18], and the  $H_\infty$  Extended Kalman Filter [19]. Each of these extensions demonstrate improved robustness to modelling errors and better estimation of non-linear dynamics. This is only a small selection of improved filters, as shown in various review articles [20][21].

Therefore, this section does not aim to provide an exhaustive overview of all possible state estimators. Instead, the focus is on the application of state estimators to different systems, firstly flexible multibody manipulators and secondly legged robotic systems. This comparison aims to show how these two application fields overlap.

#### 2.1.1 State Estimation of Flexible Manipulators

The first step in state estimation is to determine whether the state is observable, i.e. whether all the states influence the measurements. *Balas* executed the required observability check in 1978 and concluded that one measurement on a flexible beam is sufficient to measure all the flexible modes [22] when the multiplicity is 1, i.e. there are no eigenmodes with the same eigenfrequency<sup>1</sup>.

---

<sup>1</sup>This is not always the case, for example, a robot which is not attached to the fixed world has six rigid body modes with an eigenfrequency of zero. Such robot a robot always requires at least six sensors to measure these rigid body modes[8]

*Hughes and Skelton* extended this theory by introducing the notion that the sensor should not be in a nullspace of a mode [23]. This happens, for example, when a rotary encoder tries to measure an axial mode shape. It is important to note that eigenfrequencies that are multiples are harder to observe with one sensor [24].

Despite the fact that a flexible beam structure should be observable, many researchers use multiple sensors to improve the observation. For example, *Li et al.* observed the flexible state based only on rotary encoders [25, 26]. However, current research is more focused on applying additional sensors such as strain gauges [27–29], vision [30–32], inertial sensors [33, 34], or end-effector location [35, 36]. Most of these researchers apply a Kalman filter, such as the simple Kalman Filter [25, 31, 33, 36], the Extended Kalman Filter [28, 35] or the Unscented Kalman Filter [30, 32, 34]. A few have designed their own observers based on the Kalman Filter to include flexible uncertainty [27], or have designed custom observers to estimate partial differential equations [26, 29].

### 2.1.2 State Estimation of Legged Robots

Where flexible manipulators, as described in the previous section, are observable due to their connection to the real world, this is not the case for legged robots. Classic sensors for a robot include rotary encoders on all the joints and an Inertial Measurement Unit (IMU) on the body of the robot; however, using only these sensors, the exact location of the body and the yaw angle are unobservable [8].

One solution for this is to use kinematic state observation. Here, the leg kinematics and ground reaction forces can be used to estimate the base location, as described extensively in a review article by *Masuya and Ayusawa* [37]. Alternatively, *simultaneous localization and mapping* (SLAM) algorithms can detect the position of a body based on non-kinematics sensors such as cameras, thermals, or LiDaR [7, 38]. The combination of kinematics state-observation with non-kinematic sensors has shown promising results and has been applied successfully with the momentum-based framework by *Koolen et al.* [11, 39].

Compliant elements have also been used in the state estimation for legged robots. *Koolen et al.* note that both link and joint compliance can degrade the quality of the centre of mass estimation [11]. To address this, they propose including a torsion spring in the ankle of the model, with the stiffness following from an optimization problem. Other research has focused on state estimation when the robot has an elastic transmission [40–42]. *Vigne et al.* simplify all internal flexibilities as punctual flexibilities [43], again modelling these flexibilities as torsion springs.

To the best of the author’s knowledge, there are no examples of link compliance in any legged robot state estimation. Examples of state estimation of compliant links do exist for either manipulators or satellites [44]; however, for legged robots, link compliance is rarely considered. When it is, the link compliance is usually simplified to a torsional spring.

## 2.2 Control of (Compliant) Legged Robots

The control of legged robots is a heavily studied field, and thus it is not feasible to examine the entire field in depth. A search of *Scopus* using the query ‘Control AND (“Legged Robot” OR biped OR quadruped)’ yields over 12000 articles. This section will provide an overview of the different areas of control, and then focus on the momentum-based controller, which is the focus of the second paper.

The control of legged robots can be divided into three distinct sections: motion generation, trajectory regularization, and actuator control. Motion generation involves deciding when, where,

and how to make a step, as well as the corresponding movement of the center of mass. Trajectory regularization ensures that steps can be taken in the presence of external disturbances. Finally, actuator control (also known as low-level control) ensures that the actuators deliver the required torque.

There are two general approaches to motion generation: full-body motion generation and simplified model gait generation. Early simplified models focused on slow gait generation and included the Linear Inverted Pendulum Model (LIPM) [45], Inverted Pendulum Model (IPM) [46], and Linear Pendulum Model (LPM) [47]. Later research focused on dynamic gait, resulting in the Spring Loaded Inverted Pendulum (SLIP) model [48], which models the legs as a set of springs. Alternatively, offline gait trajectories can be generated using a non-linear optimizer on the full-body model [49–51] or by estimating human gait [52]. There is also research into trajectory generation through a neural network, where the learned trajectory can then be used online [53].

The most common method to regularize a trajectory is using a Proportional Derivative controller on the desired joint angles or state feedback [54]. However, this does not utilize the knowledge of the dynamic model. *Hwangbo et al.* use a neural network to learn the actuator and robot dynamics to generate one policy which performs both regularization and trajectory planning [53]. Alternatively, Whole Body Control (WBC) [55] uses task prioritization to regularize motions. When one of these whole body control tasks includes the momentum, it is generally called momentum-based control (MBC). Central pattern generators (CPGs) also use a method similar to the human body to control the torques in a legged robot [56], combining control sequences with reflex loops to generate robust gaits. CPG has also been successful in control of a compliant spine [57].

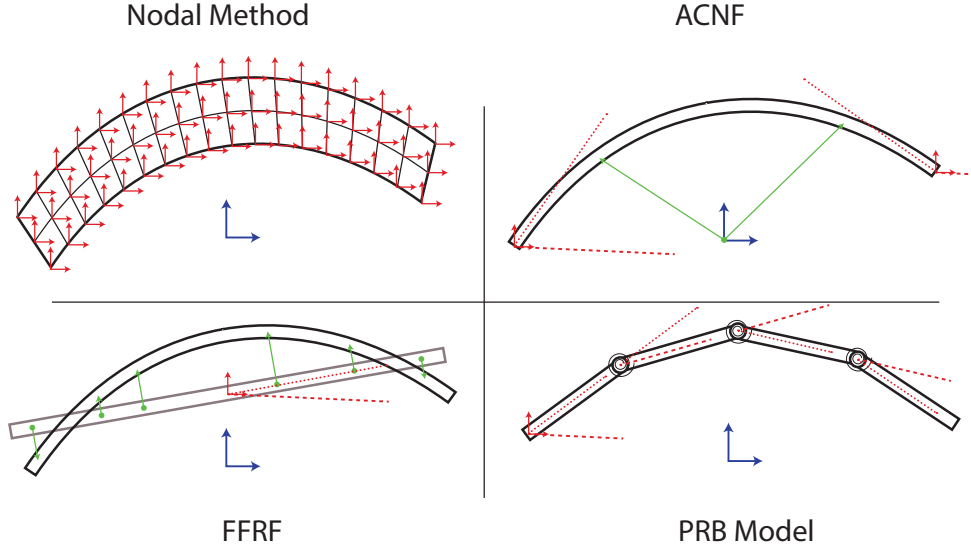
Humans control their (angular) momentum when walking; and *Abdallah and Goswami* were the first to apply this concept to the control of legged robots. This momentum-based control has since been shown to be successful in balancing a robot, both in simulation [58]–[62] and on real robots [9]–[12]. It is also possible to control robots with wheeled legs [63] based on the momentum. The downside of these implementations of momentum-based controllers is the separation between motion generation and the momentum-based controller. This separation can cause the planned motion to become infeasible when there are large disturbances [64].

## 2.3 Modelling of Flexible Multibody Systems

Flexible multi-body systems have been extensively studied, and there are various ways to model them [65]. This section focuses on three strategies for modelling flexible bodies: the *Floating Frame of Reference Formulation* (FFRF), the *Absolute Nodal Coordinate Formulation* (ACNF) [66], and the *Pseudo-Rigid Body* (PRB) model [67]. Fig. 2.1 illustrates these three formulations. Nodal model methods, which split the beam into multiple elements, are not discussed in this review, as they are computationally expensive for control problems [68].

The PRB model transforms a flexible beam into multiple rigid beams connected by torsion springs [67]. An algorithm optimizes the orientation and length of the different beams, as well as the stiffness of the torsion springs, such that the PRB model behaves similarly to a simulated flexible beam. This model consists of four rigid beam elements and three torsion springs [69], which gives the optimizer 10 parameters to optimize. Compared to a rigid beam element, this model offers 3 additional degrees of freedom, and accurately follows the deflection of the tip of large finite element models [70].

However, current research into the PRB model focuses primarily on (quasi-)static tip deflection of a beam fixed to a wall [70–73]. In addition, the force and moments used in the optimization are always located at the tip of the beams, making it unclear how this model formulation performs



**Figure 2.1:** The four different flexible multibody formulations discussed in the literature review. The Nodal method is not explained due to the large number of degrees of freedom (3 per node).

in dynamic cases or when the beam is not fixed to a base. Furthermore, in spatial beams, the parameter optimization can become complex due to an increase in parameters [74].

The ANCF uses the slopes of the beam at all the joints as coordinates [75]. The rigid body and flexible body motions are coupled in this information through shape functions. These shape functions form the basis of the mass and stiffness of the element. In this formulation, the stiffness energy is defined by a nonlinear combination of the coordinates, which allows for large deformation in the element.

Recent review articles show that ACNF is still a highly researched field [76, 77]. Recent advances focus on the challenges of multi-body simulation with external forces and (imperfect) actuators [77]. However, one disadvantage of the ACNF is that rigid and flexible motions are connected, which means that rigid elements require more nodes than minimal, making a rigid-flexible model less efficient.

Finally, the FFRF is an extension of rigid body dynamics. The rigid body motion and flexible motion are decoupled using flexible mode shapes, which can be determined from an experiment [78] or finite element simulations. Since only a limited number of flexible modes are considered, the compliant element can be complex and the simulation with the complex structure only needs to be done once to find the Eigenvalues and Eigenmodes.

The largest disadvantage of the FFRF is that it can only use linear elements, which are not valid for large deformations. To solve this problem, beams can be divided into multiple elements to make the results similar to the non-linear ACNF [79, 80]. Furthermore, the constraint equations to connect the different elements are a non-linear combination of the coordinates. *Ellenbroek and Schilder* solve this problem by defining super elements to represent an element only in its interface points [81].

Based on the comparison of these three formulations, FFRF is best suited for the application in this thesis. The largest strength of FFRF is that it allows a modal reduction, which greatly improves the algorithm when the spine has a complicated shape and mass distribution. After an accurate but time-consuming finite element method simulates the flexibility of a spine, the fast FFRF only uses the relevant mode shapes. Besides the modal reduction, the FFRF is the most similar to rigid body dynamics, which simplifies the comparison of the flexible simulation results to rigid simulations in the literature.

# Chapter 3

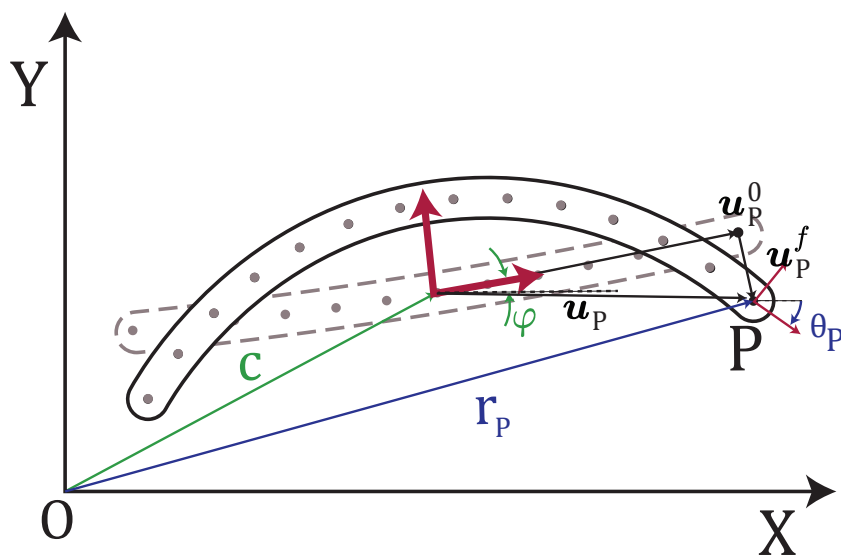
## Preliminaries

The information in this section complements the information given in the papers in Part II. Naturally, there will be some overlap in information to allow the papers to be published separately. Where the papers focus on what was done, these section focus on why it was done. This hopefully makes the papers clear, even for engineers who do not work in the field of flexible multibody dynamics and balance control.

### 3.1 Modelling in the floating frame formulation

The main issue addressed in this thesis is the addition of a compliant link. This section explains the modelling of such flexible links in the floating frame of reference method.

#### 3.1.1 Beams in the floating frame



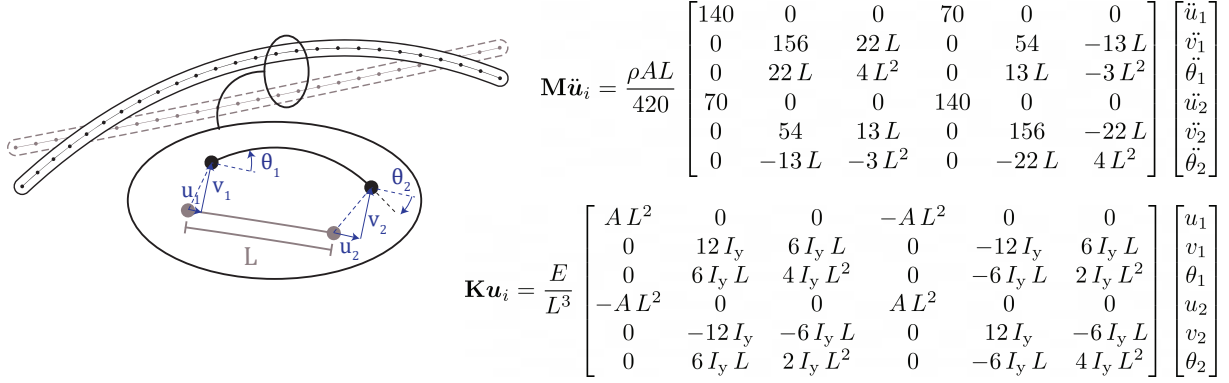
**Figure 3.1:** One beam in the floating frame representation. The dots represent the nodes in the nodal dynamics, the dashed gray body is the undeformed body. The green lines represent the rigid body motions in the global frame, while the black lines are local definition. The blue line gives the global position and orientation of a point  $P$  on the deformed body.

This explanation starts with the floating frame formulation for one beam. This element is defined in Fig. 3.1 where the dynamics of this beam is split into nodal dynamics and frame dynamics. The nodal dynamics requires solving the displacements of all the nodes,  $\mathbf{u}$  based on, for example,

a linear finite element model given as

$$\mathbf{M}_{FEM}\ddot{\mathbf{u}}^f + \mathbf{K}_{FEM}\mathbf{u}^f = \mathbf{Q}_{FEM} \quad (3.1)$$

where  $\mathbf{M}_{FEM}$  and  $\mathbf{K}_{FEM}$  are, often sparse, mass and stiffness matrices. The contents of these matrices depend on the number of nodes, node locations, beam shape, beam type and the material properties. The simulations in this thesis use a 1D beam with a local consistent mass matrix and an Euler-Bernoulli beam element, as shown in Fig. 3.2.



$$\mathbf{M}\ddot{\mathbf{u}}_i = \frac{\rho AL}{420} \begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2 \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2 \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{v}_1 \\ \ddot{\theta}_1 \\ \ddot{u}_2 \\ \ddot{v}_2 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$\mathbf{K}\mathbf{u}_i = \frac{E}{L^3} \begin{bmatrix} AL^2 & 0 & 0 & -AL^2 & 0 & 0 \\ 0 & 12I_y & 6I_y L & 0 & -12I_y & 6I_y L \\ 0 & 6I_y L & 4I_y L^2 & 0 & -6I_y L & 2I_y L^2 \\ -AL^2 & 0 & 0 & AL^2 & 0 & 0 \\ 0 & -12I_y & -6I_y L & 0 & 12I_y & -6I_y L \\ 0 & 6I_y L & 2I_y L^2 & 0 & -6I_y L & 4I_y L^2 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{bmatrix}$$

**Figure 3.2:** One beam element with its mass and stiffness matrix. Here  $\rho$  is the density,  $A$  is the area,  $L$  is the length of one element,  $E$  is the young's modulus, and  $I_y$  is the second moment of area. Since the beam is rectangular, this equals  $I_y = \frac{bh^3}{12}$ , where  $b$  is the width and  $h$  is the height of the beam.

The full beam in Fig. 3.2 has  $9 \cdot 3 = 27$  degrees of freedom. A modal reduction reduces the number of coordinates by only considering specific mode shapes. There are multiple options for modal reduction. This thesis does the modal reduction based on the eigenmodes of the system, which are shown in Fig. 3.3. The eigenmodes with the lowest eigenfrequency will store most of the energy and therefore influence the system the most. The global position and orientation of the nodes, as given in, Fig. 3.1 will then be given by

$$\mathbf{r}_P = \mathbf{c} + \mathbb{R}(\varphi)(\mathbf{u}) = \mathbf{c} + \mathbb{R}(\varphi)(\mathbf{u}_P^0 + \mathbf{u}_P^f) = \mathbf{c} + \mathbb{R}(\varphi)(\mathbf{u}^0 + \Phi^f \mathbf{q}^f) \quad (3.2)$$

$$\boldsymbol{\theta}_P = \varphi + \Phi_{\theta,P}^f \mathbf{q}^f \quad (3.3)$$

where  $\mathbb{R}(\varphi)$  is a rotation matrix and  $\Phi^f$  is the matrix with the used mode shapes and  $\mathbf{q}^f$  are scalars for the different mode shapes. Furthermore,  $(\cdot)_P$  are the rows related to the point  $P$ , similarly  $(\cdot)_{\theta,P}$  are the rows related to the angle of the beam at point  $P$ .

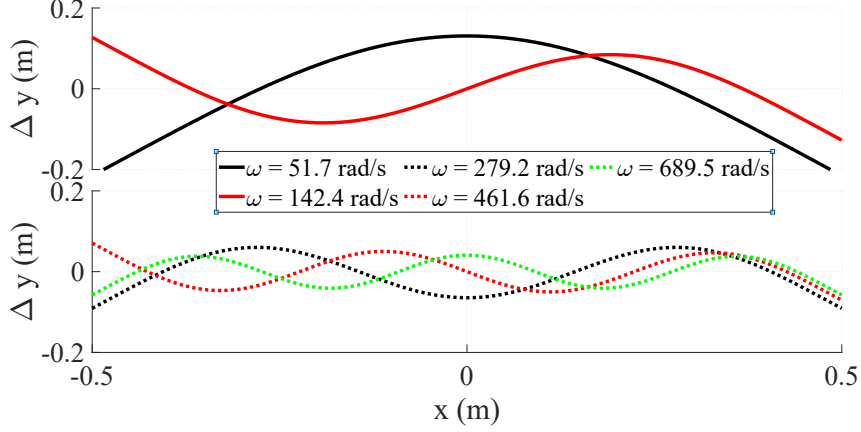
The step from the modal coordinates and the frame coordinates follows from the mode shapes. Ignoring the insignificant quadratic velocity terms<sup>1</sup>. In this case,

$$\ddot{\mathbf{u}}^f = \begin{bmatrix} \Phi_r & \Phi_f \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{c}} \\ \ddot{\varphi} \\ \ddot{\mathbf{q}}^f \end{bmatrix}, \quad \mathbf{u}^f = \begin{bmatrix} \mathbf{0} & \Phi_f \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \varphi \\ \mathbf{q}^f \end{bmatrix} \quad (3.4)$$

where  $\Phi_r$  are the rigid body modes, which are the displacements of the nodes when the center of mass changes. Since, flexible coordinates are relative to center of mass frame, these rigid modes are 0 for the positions. Filling this into Eq. (3.1) and premultiplying with the transpose of the modes

$$\begin{bmatrix} \Phi_r^T \\ \Phi_f^T \end{bmatrix} \mathbf{M}_{FEM} \begin{bmatrix} \Phi_r & \Phi_f \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{c}} \\ \ddot{\varphi} \\ \ddot{\mathbf{q}}^f \end{bmatrix} + \begin{bmatrix} \Phi_r^T \\ \Phi_f^T \end{bmatrix} \mathbf{K}_{FEM} \begin{bmatrix} \mathbf{0} & \Phi_f \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \varphi \\ \mathbf{q}^f \end{bmatrix} = \begin{bmatrix} \Phi_r^T \\ \Phi_f^T \end{bmatrix} \mathbf{Q}_{FEM} \quad (3.5)$$

<sup>1</sup>These terms become apparent only when working with high angular velocities or when bending becomes very large.



**Figure 3.3:** Mode shapes of the spine used by paper 1 and paper 2. The top figure gives the first two mode shapes, which are also included in the simulation in paper 1. The dotted mode shapes in the bottom figure are residual mode shapes; they are not used in the controller or observer model. It also gives the natural frequencies  $\omega_n$  for the different modes.

calculating this gives

$$\mathbf{M}_{fr} \begin{bmatrix} \ddot{\mathbf{c}} \\ \ddot{\varphi} \\ \ddot{\mathbf{q}}^f \end{bmatrix} + \mathbf{K}_{fr} \begin{bmatrix} \mathbf{c} \\ \varphi \\ \mathbf{q}^f \end{bmatrix} = \begin{bmatrix} \mathbf{M}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{c}} \\ \ddot{\varphi} \\ \ddot{\mathbf{q}}^f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \varphi \\ \mathbf{q}^f \end{bmatrix} \quad (3.6)$$

where  $\mathbf{M}_R = \text{diag}(m, m, I)$  where  $m$  is the total mass and  $I$  is the moment of inertia.  $\mathbf{M}_f$  is a diagonal matrix with the flexible inertias,  $\mathbf{K}_f$  are the flexible stiffnesses. These matrices are often scaled such that  $\mathbf{M}_f$  is identity and  $\mathbf{K}_f$  is a diagonal matrix with the  $\omega_n^2$ . Appendix A.1 explains how damping can be added to this system.

So, the frame dynamics depend on the position of the frame, orientation of the frame and the flexible mode scalers

$$\mathbf{M}_{fr} \ddot{\mathbf{q}}_{fr} + \mathbf{K}_{fr} \mathbf{q}_{fr} = \mathbf{M}_{fr} \begin{bmatrix} \ddot{\mathbf{c}} \\ \ddot{\varphi} \\ \ddot{\mathbf{q}}^f \end{bmatrix} + \mathbf{K}_{fr} \begin{bmatrix} \mathbf{c} \\ \varphi \\ \mathbf{q}^f \end{bmatrix} = \mathbf{Q}_{fr} \quad (3.7)$$

where  $\mathbf{Q}$  are the generalized force on the body, these are derived from the Jacobian of the (angular) velocities of point  $P$  as

$$\dot{\mathbf{r}}_P = \dot{\mathbf{c}} + \dot{\mathbb{R}}(\varphi)(\mathbf{u}_P^0 + \Phi_P^f \mathbf{q}^f) + \mathbb{R}(\varphi) \Phi_P^f \dot{\mathbf{q}}^f = \begin{bmatrix} \mathbf{I} & \mathbb{R}(\varphi)(\mathbf{u}_P^0 + \Phi_P^f \mathbf{q}^f)^\times & \mathbb{R}(\varphi) \Phi_P^f \end{bmatrix} \begin{bmatrix} \dot{\mathbf{c}} \\ \dot{\varphi} \\ \dot{\mathbf{q}}^f \end{bmatrix} \quad (3.8)$$

$$\dot{\theta}_P = \dot{\varphi} + \Phi_{P,\theta}^f \dot{\mathbf{q}}^f = \begin{bmatrix} 0 & 1 & \Phi_{P,\theta}^f \end{bmatrix} \begin{bmatrix} \dot{\mathbf{c}} \\ \dot{\varphi} \\ \dot{\mathbf{q}}^f \end{bmatrix} \quad (3.9)$$

Where,  $\mathbf{r}^\times \equiv \begin{bmatrix} -p_y \\ p_x \end{bmatrix}$ , which follows from  $\dot{\mathbb{R}}(x) = \mathbb{R}(x) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \dot{x}$ . Based on Eq. (3.8) and

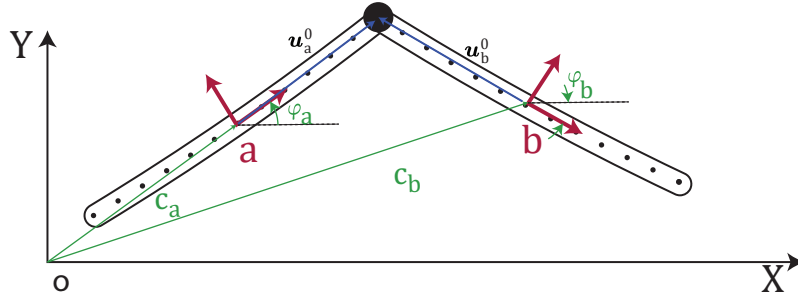
Eq. (3.9) the Jacobian and its derivative for a point  $P$  equal

$$\begin{bmatrix} \dot{\mathbf{r}}_P \\ \dot{\theta}_P \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbb{R}(\varphi)(\mathbf{u}_P^0 + \Phi_P^f \mathbf{q}^f)^\times & \mathbb{R}(\varphi)\Phi_P^f \\ 0 & 1 & \Phi_{P,\theta}^f \end{bmatrix} \begin{bmatrix} \dot{\mathbf{c}} \\ \dot{\varphi} \\ \dot{\mathbf{q}}^f \end{bmatrix} \equiv \mathbf{J}_P \dot{\mathbf{q}}_{fr}, \quad (3.10)$$

$$\begin{bmatrix} \ddot{\mathbf{r}}_P \\ \ddot{\theta}_P \end{bmatrix} = \mathbf{J}_P \ddot{\mathbf{q}}_{fr} + \begin{bmatrix} \mathbf{0} & -\mathbb{R}(\varphi)(\mathbf{u}_P^0 + \Phi_P^f \mathbf{q}^f)\dot{\varphi} + 2\mathbb{R}(\varphi)(\Phi_P^f \dot{\mathbf{q}}^f)^\times & \mathbf{0} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{c}} \\ \dot{\varphi} \\ \dot{\mathbf{q}}^f \end{bmatrix} \equiv \mathbf{J}_P \ddot{\mathbf{q}}_{fr} + \dot{\mathbf{J}}_P \dot{\mathbf{q}}_{fr}. \quad (3.11)$$

Through the conservation of power, the relation between a global force on a node  $P$  called  $\mathbf{Q}_{P,FE}$  and the generalized force in Eq. (3.7) equals  $\mathbf{Q}_{fr} = \mathbf{J}_P^T \mathbf{Q}_{P,FE}$ .

### 3.1.2 Combining floating frame elements



**Figure 3.4:** Two rigid beams with a revolute joint.

The connection of different beam elements is performed through the nonlinear holonomic constraints. These holonomic constraints can always be represented as

$$\mathbf{f}_a(\mathbf{q}_a) - \mathbf{f}_b(\mathbf{q}_b) = 0 \quad (3.12)$$

where  $\mathbf{f}_a(\mathbf{q}_a)$  and  $\mathbf{f}_b(\mathbf{q}_b)$  are a nonlinear function based on the coordinates of a body  $a$  and body  $b$ , respectively. One of these bodies can also be the world if a body is connected to the world. The most common connection between bodies is a revolute joint, as shown in Fig. 3.4. Alternatively, constraints can be used to split one beam up allowing also nonlinear flexibility, as shown in Appendix A.2. For a revolute joint between two rigid beams, the constraint equations are then given by

$$\begin{aligned} \mathbf{c}_a + \mathbb{R}(\varphi_a)\mathbf{u}_a^0 - (\mathbf{c}_b + \mathbb{R}(\varphi_b)\mathbf{u}_b^0) &= \mathbf{c}_a + \mathbb{R}(\varphi_a) \begin{bmatrix} \frac{L_a}{2} \\ 0 \end{bmatrix} - \left( \mathbf{c}_b + \mathbb{R}(\varphi_b) \begin{bmatrix} \frac{-L_b}{2} \\ 0 \end{bmatrix} \right) = \\ \begin{bmatrix} \mathbf{c}_{a,x} - \mathbf{c}_{b,x} + \cos(\varphi_a)\frac{L_a}{2} + \cos(\varphi_b)\frac{L_b}{2} \\ \mathbf{c}_{a,y} - \mathbf{c}_{b,y} - \sin(\varphi_a)\frac{L_a}{2} - \sin(\varphi_b)\frac{L_b}{2} \end{bmatrix} &= \mathbf{0}. \end{aligned} \quad (3.13)$$

For a flexible beam,  $\mathbf{u}_a^0$  should be replaced by  $\mathbf{u}_a$ , which is a combination of the original position and a flexible component to get

$$\mathbf{c}_a + \mathbb{R}(\varphi_a) \left( \begin{bmatrix} \frac{L_a}{2} \\ 0 \end{bmatrix} + \Phi_a^f \mathbf{q}_a^f \right) - \left( \mathbf{c}_b + \mathbb{R}(\varphi_b) \left( \begin{bmatrix} \frac{-L_b}{2} \\ 0 \end{bmatrix} + \Phi_b^f \mathbf{q}_b^f \right) \right) = \mathbf{0}, \quad (3.14)$$



where  $\Phi_a^f$  are the rows of the mode shape matrix corresponding to the node with the revolution joint for the body  $a$ . The holonomic constraint matrix,  $\mathcal{C}$ , combines the constraints of all joints such that

$$\mathcal{C}(\mathbf{q}) = \mathbf{0}. \quad (3.15)$$

Also the derivative of this matrix must be 0, these can be computed using the chain rule since  $\mathcal{C}$  is a function of  $\mathbf{q}$ ,

$$\dot{\mathcal{C}} = [\mathcal{C}]_{\mathbf{q}} \dot{\mathbf{q}} = 0, \quad (3.16)$$

where  $[\cdot]_{\mathbf{q}}$  denotes the Jacobian with respect to  $\mathbf{q}$ . Taking the derivative again gives

$$\ddot{\mathcal{C}} = [\mathcal{C}]_{\mathbf{q}} \ddot{\mathbf{q}} + [\mathcal{C}]_{\mathbf{q}\mathbf{q}} \dot{\mathbf{q}} \dot{\mathbf{q}} = [\mathcal{C}]_{\mathbf{q}} \ddot{\mathbf{q}} + [[\mathcal{C}]_{\mathbf{q}} \dot{\mathbf{q}}]_{\mathbf{q}} \dot{\mathbf{q}} = 0, \quad (3.17)$$

where  $[\mathcal{C}]_{\mathbf{q}\mathbf{q}}$  is the Hessian of  $\mathcal{C}$ . We prevent the computation of the Hessian since  $\dot{\mathbf{q}}$  is not a function of  $\mathbf{q}$  so  $[\mathcal{C}]_{\mathbf{q}\mathbf{q}} \dot{\mathbf{q}} = [[\mathcal{C}]_{\mathbf{q}} \dot{\mathbf{q}}]_{\mathbf{q}}$ .

The dynamics of all the bodies combined is a combination of the separate equations of motions given in Eq. (3.7)

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{K} \mathbf{q} = \mathbf{Q} \quad (3.18)$$

where  $\mathbf{q} = [\mathbf{q}_{fr,a}^T, \mathbf{q}_{fr,b}^T, \dots, \mathbf{q}_{fr,Nb}^T]^T$ . The mass and stiffness matrices are block diagonal matrices given by, for example,  $\mathbf{M} = \text{blkdiag}(\mathbf{M}_{fr,a}, \mathbf{M}_{fr,b}, \dots, \mathbf{M}_{fr,Nb})$ . The generalized force,  $\mathbf{Q}$  are split into actuation forces  $\mathbf{Q}_a$  and constraint forces  $\mathbf{Q}_c$ .

The constraint forces can be removed from the equations of motions using the d'Alembert principle. The *kinematic admissible displacement field* follows from the holonomic constraint matrix,

$$\mathcal{C} = 0 \rightarrow \delta \mathcal{C} = 0 \rightarrow [\mathcal{C}]_{\mathbf{q}} \delta \mathbf{q} = 0, \quad (3.19)$$

where  $\delta(\cdot)$  is the variation of a variable. This constraint on the *kinematic admissible displacement field* results in Lagrange multipliers such that the equation of motion becomes

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{K} \mathbf{q} - ([\mathcal{C}]_{\mathbf{q}})^T \boldsymbol{\lambda} = \mathbf{Q}_a \quad (3.20)$$

This system is not solvable since both  $\ddot{\mathbf{q}}$  and  $\boldsymbol{\lambda}$  are unknown. Therefore, the system is augmented using the constraints. Using Eq. (3.17) the system of equations becomes,

$$\begin{bmatrix} \mathbf{M} & ([\mathcal{C}]_{\mathbf{q}})^T \\ [\mathcal{C}]_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a - \mathbf{K} \mathbf{q} \\ -[[\mathcal{C}]_{\mathbf{q}} \dot{\mathbf{q}}]_{\mathbf{q}} \dot{\mathbf{q}} \end{bmatrix}. \quad (3.21)$$

which is called the *Constrained Equations of Motion in Augmented Form*.

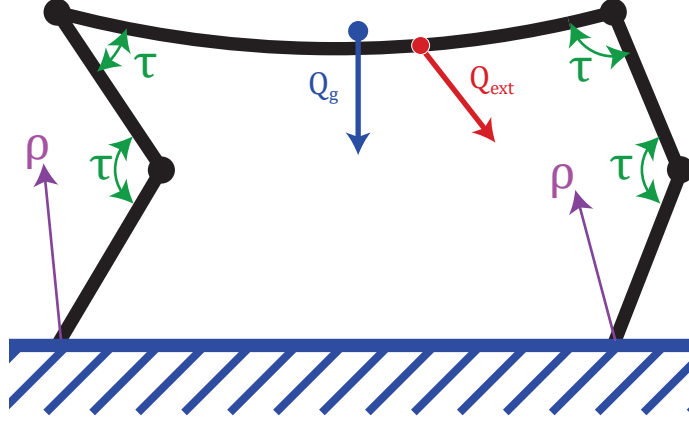
### 3.1.3 Actuation in the Floating Frame Formulation

The quadruped experiences four distinct types of external forces: actuator forces, disturbance forces, ground reaction forces, and the force of gravity, as shown in Fig. 3.5. Using these forces, the actuation force equals

$$\mathbf{Q}_a = \mathbf{S}_{\tau}^T \boldsymbol{\tau} + \mathbf{J}_{ext}^T \mathbf{W}_{ext} + \mathbf{J}_{\rho}^T \boldsymbol{\rho} + \mathbf{Q}_g, \quad (3.22)$$

where this chapter explains how to get the  $\mathbf{S}_{\tau}$ ,  $\mathbf{J}_{ext}$ ,  $\mathbf{J}_{\rho}$ , and  $\mathbf{Q}_g$ . The designer determines the external wrench,  $\mathbf{W}_{ext}$ , the controller defines the torques,  $\boldsymbol{\tau}$ , and the contact model in Appendix B determines the ground reaction forces,  $\boldsymbol{\rho}$ .

Since  $\mathbf{J}_{ext}$  and  $\mathbf{J}_{\rho}$  are simply Jacobians, they can be calculated using Equation (1) (Jacobian). These forces only result in a generalized force  $\mathbf{Q}_a$  on the body they act upon. Therefore, these



**Figure 3.5:** All applied forces on a quadruped,  $\rho$  are the ground reaction forces,  $\tau$  are the motor torques,  $Q_g$  is the force due to gravity and  $Q_{ext}$  is an external force

Jacobians only have non-zero values for the coordinates associated with the actuated body, and are zero for all other coordinates. For ground reaction forces, the Jacobian for each foot location should be stacked. For external forces, the Jacobian depends on the direction and type of force, and should correspond to the velocity in the same direction as the force. This can also be an angular velocity in when the external force is a moment.

The actuator matrix is similar to the Jacobians. However, due to Newton's third law, each torque results in an equal and opposite torque in the other direction. So for a torque between two rigid bodies  $a$  and  $b$  in Fig. 3.4

$$\mathbf{S}_\tau = \begin{bmatrix} -\mathbf{J}_{\theta,a} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{J}_{\theta,b} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

where  $\mathbf{J}_{\theta,a}$  and  $\mathbf{J}_{\theta,b}$  are the Jacobians corresponding to the angular velocity of the beam at the revolution joint for body  $a$  and  $b$ , respectively. For a flexible beam, these Jacobians equals the last row in Eq. (3.10).

Finally, the gravity only acts on the rigid body motion of the bodies. This gives

$$\mathbf{Q}_g = \sum_{i=1}^{Nb} -g \mathcal{S}_{y,i}^T m_i, \quad (3.24)$$

where  $g$  is the gravitation constant,  $m_i$  is the mass of the body  $i$ , and  $Nb$  is the number of bodies.  $\mathcal{S}_{y,i}$  is a selection matrix for the rigid body motion in the  $y$ -direction of the body  $i$ . For body  $b$  in Fig. 3.4 this becomes

$$\mathcal{S}_{y,b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.25)$$

So for the rigid body elements in Fig. 3.4

$$\mathbf{Q}_g = \begin{bmatrix} 0 & -m_a g & 0 & 0 & -m_b g & 0 \end{bmatrix}^T, \quad (3.26)$$

where  $m_a$  and  $m_b$  are the mass of body  $a$  and  $b$ .

### 3.1.4 Validation of the Floating Frame Formulation

The validity of the flexible beam is proven in Appendix A.2 where the beam is split into multiple beams to see the convergence. Two more principles can prove if the simulation is correct, conservation of momentum and conservation of energy. The conservation of energy will be checked

based on a flexible 2 link pendulum, while the conservation of momentum is checked on the complete quadruped.

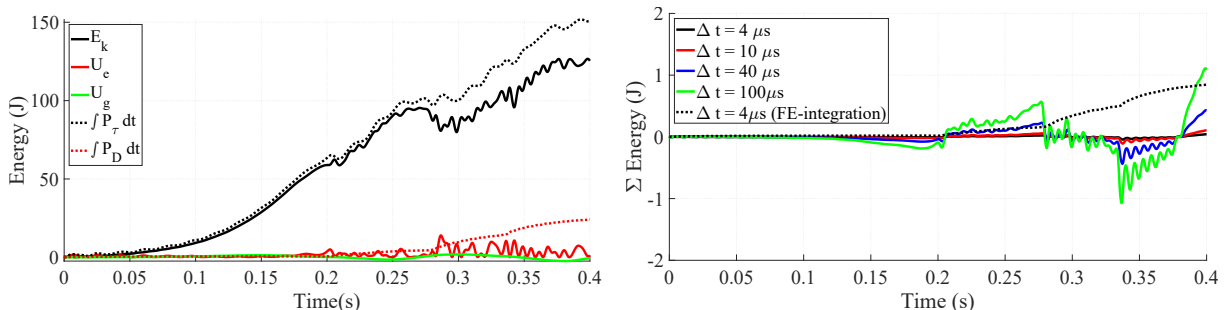
The total energy in the system can be split into three different categories: kinetic energy,  $E_k$ , potential elastic energy,  $U_e$  and potential gravitational energy,  $U_g$ . The actuators and the internal damping change the amount of energy in the system such that

$$E_k + U_e + U_g - \int P_\tau dt + \int P_D dt = \text{const}, \quad (3.27)$$

where  $P_\tau = \boldsymbol{\tau}^T \mathbf{S}\dot{\mathbf{q}}$ , and  $P_D = \dot{\mathbf{q}}^T \mathbf{D}\dot{\mathbf{q}}$ . The energies can be computed as

$$E_k = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}}, \quad U_e = \frac{1}{2} \mathbf{q}^T \mathbf{K}\mathbf{q}, \quad \text{and} \quad U_g = -\mathbf{Q}_g^T (\mathbf{q} - \mathbf{q}_0), \quad (3.28)$$

where  $\mathbf{q}_0$  are the initial coordinates. Plotting these values in Fig. 3.6 shows that the deviation in the energy depends greatly on the integration time step and the used integrator. For the same time step, the symplectic Euler integration scheme [82] decreases the error from 0.84 J to 0.02 J compared to the Forward Euler scheme.



**Figure 3.6:** Checking the energy balance for a 2 link pendulum with two compliant links. The left figure splits up the different energies for a time step of  $40 \mu\text{s}$ . The right figure shows the change in the energy for different integration time steps when the Symplectic Euler Scheme is used. Furthermore, one simulation is run with the standard Forward Euler scheme.

The conservation of momentum follows from Newton's second law

$$\dot{\mathbf{h}}_G = \mathbf{W}_G, \quad (3.29)$$

where  $\mathbf{h}_G = [l_G, \mathbf{k}_G^T]^T$  is the centroidal momentum, where  $l_G$  is the total angular momentum relative to the center of mass and  $\mathbf{k}_G$  is the total linear momentum. The centroidal Wrench,  $\mathbf{W}_G = [\tau_G, \mathbf{F}_g^T]^T$ , is composed of all the external torque around the center of mass,  $\tau_G$  and the external forces,  $\mathbf{F}_G$ .

The centroidal momentum is computed using the centroidal momentum matrix,  $\mathbf{A}_G$ ,

$$\dot{\mathbf{h}}_G = \mathbf{A}_G \ddot{\mathbf{q}} + \dot{\mathbf{A}}_G \dot{\mathbf{q}}. \quad (3.30)$$

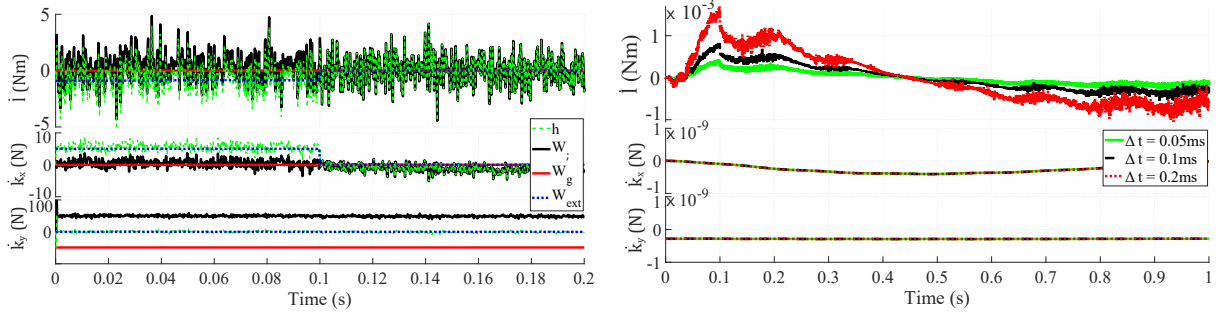
Both papers in Part II discuss how to get  $\mathbf{A}_G$  and  $\dot{\mathbf{A}}_G$ . This derivation will not be repeated here. For the centroidal wrench, there are 3 external wrenches acting on the center of mass: the ground reaction forces,  $\mathbf{W}_\rho$ , the gravity,  $\mathbf{W}_g$ , and the external disturbance force  $\mathbf{W}_{ext}$ . Note that the actuators do not result in a wrench on the center of mass.

By definition, the gravity does not result in a torque. The ground reaction forces and disturbance force do result in a torque, since these forces do not act on the center of mass. The torque on the center of mass,  $\tau_G$  equals

$$\begin{bmatrix} 0 \\ 0 \\ \tau_G \end{bmatrix} = (\mathbf{p}_F - \mathbf{c}_R) \times \mathbf{F}, \quad (3.31)$$

where  $\mathbf{p}_F$  is the position the force  $\mathbf{F}$  acts,  $\mathbf{c}_R$  is the center of mass position of the robot, and  $\times$  is the cross product. The total external force for these wrenches is the sum of forces.

The conservation of momentum is checked for the quadruped in Paper 2 in Fig. 3.7. Similar to the energy, there is an error in the angular momentum, which decreases linearly with a smaller time step. This indicates that the physics in the simulation is correct, and the error is due to the truncation in the integration.

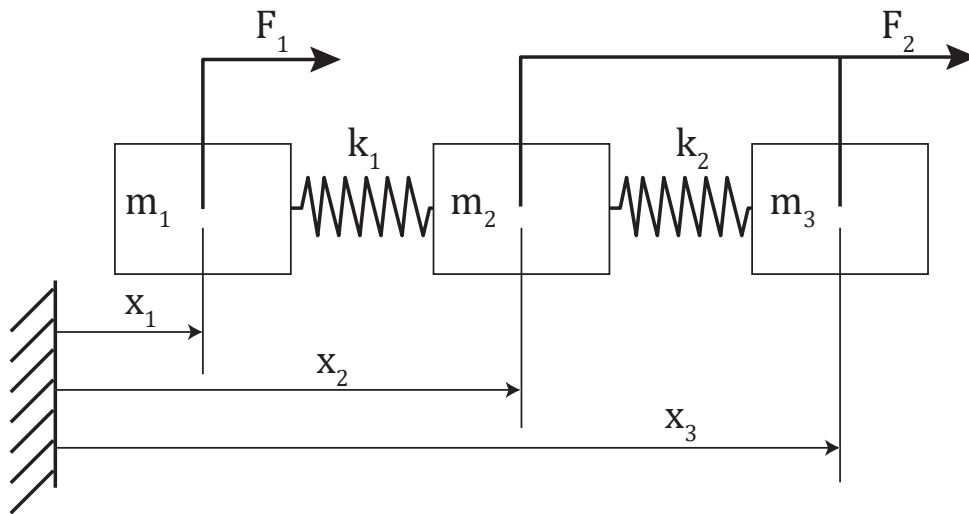


**Figure 3.7:** Checking the momentum balance for the simulated quadruped in Paper 2. The left figure splits up the different momentum for a time step of 0.1 ms, the same as used in the simulation. The right figure shows the change in the energy for different integration time steps when the Symplectic Euler Scheme is used.

## 3.2 Null spaces

### 3.2.1 Null space for underactuated systems

All legged robots are underactuated, meaning there are fewer actuators than degrees of freedom. This means that not all accelerations are possible. Including compliant link elements in the robot further increases the degrees of freedom without adding more actuators. This limited number of actuators cannot achieve all combinations of generalized forces. The momentum-based controller finds the optimal accelerations based on a (hierarchical) quadratic problem. Therefore, the first step in the (hierarchical) quadratic problem must ensure that these optimal accelerations are possible.



**Figure 3.8:** Underactuated linear mass-spring system. The masses all equal 1 kg and the springs both are 10 N/m

This section explains the steps based on a simplified example shown in Fig. 3.1 for which the

equations of motions are

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{S}^T \mathbf{F}^T, \quad (3.32)$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{F}^T = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} -10 & 10 & 0 \\ 10 & -20 & 10 \\ 0 & 10 & -10 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}. \quad (3.33)$$

This system is clearly underactuated with three degrees of freedom and only two actuators. When  $\mathbf{x} = 0$ , the acceleration of  $x_2$  and  $x_3$  must be the same since they both have the same force.

The singular value decomposition splits the matrix  $\mathbf{S}^T$  such that  $\mathbf{S}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ . For the  $\mathbf{S}^T$  in the simplified problem the values of these matrices are

$$\mathbf{U} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (3.34)$$

for a matrix with more rows than columns, e.g.  $\mathbf{S}^T$  of an underactuated problem, the matrices can be split into a non-zero part and an zero part

$$\mathbf{S}^T = \begin{bmatrix} \mathbf{U}_C & \mathbf{U}_N \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_C \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T, \quad (3.35)$$

where  $\mathbf{U}_C$  is called the *Orthonormal Basis of the Column Space* and  $\mathbf{U}_N$  is called the *Orthonormal Basis of the Left-Null Space*<sup>2</sup>. Since all the vectors in  $\mathbf{U}$  are independent, premultiplying  $\mathbf{S}^T$  with  $\mathbf{U}_N^T$  always equals  $\mathbf{0}$ .  $\mathbf{U}_N^T$  is denoted as  $\tilde{\mathbf{Z}}$  in the paper.

Applying this theory to Eq. (3.32) results in a constraint on the accelerations:

$$\mathbf{U}_N^T \mathbf{M}\ddot{\mathbf{x}} + \mathbf{U}_N^T \mathbf{K}\mathbf{x} = \mathbf{U}_N^T \mathbf{S}^T \mathbf{F}^T = \mathbf{0} \quad (3.36)$$

where  $\ddot{\mathbf{x}}$  are the desired accelerations following, for example, from an quadratic program. If this holds, then the forces can be recovered by inverting the relevant columns of the singular value decompositions to give

$$\mathbf{F} = \mathbf{V} (\mathbf{\Sigma}_C)^{-1} \mathbf{U}_C^T (\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x}) \quad (3.37)$$

which follows from the general inverse of a singular value problem

$$(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^{-1} = \mathbf{V} (\mathbf{\Sigma})^{-1} \mathbf{U}^T \quad (3.38)$$

### 3.2.2 Null space for Hierarchical Control

In hierarchical control, the optimization is split into different hierarchies. The lower priorities should optimize the problem without losing the optimality of lower priorities. Going back to the simplified example in Fig. 3.1. If the first priority is related to the rigid body acceleration of the complete system, the Jacobian for this problem equals,

$$\mathbf{a}_R^1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \ddot{x}_1^1 \\ \ddot{x}_2^1 \\ \ddot{x}_3^1 \end{bmatrix} = \mathbf{J}^1 \ddot{\mathbf{x}}^1 \quad (3.39)$$

<sup>2</sup>In MATLAB this can be found using `null(AT)` for a matrix  $\mathbf{A}$

where  $a_{R,1}$  is the rigid body motion of the complete system after solving the first priority,  $\ddot{\mathbf{x}}$  is the acceleration of the different bodies. When this has been optimized to some value, the lower optimization step is not allowed to change this motion so

$$a_R^2 = a_R^2 \rightarrow a_R^2 - a_R^1 = 0 \rightarrow \mathbf{J}^1 \ddot{\mathbf{x}}^2 - \mathbf{J}_1 \ddot{\mathbf{x}}^1 = 0 \rightarrow \mathbf{J}^1 (\ddot{\mathbf{x}}^2 - \ddot{\mathbf{x}}^1) = 0 \quad (3.40)$$

The trivial solution would be to let  $(\ddot{\mathbf{x}}^2 - \ddot{\mathbf{x}}^1) = 0$ . The other solution would be to only solve  $\mathbf{v}$  for which  $\mathbf{J}^1 \mathbf{v} = 0$ . The vectors for which this is the case are spanned by the null space of  $\mathbf{J}^1$ . The vectors that span the null space are called the *Orthonormal Basis of the Null Space*<sup>3</sup> and are denoted by  $\mathbf{Z}^1$ . Then  $\mathbf{v} = \mathbf{Z}^1 \mathbf{u}^2$ ,  $\mathbf{u}^2 \in \mathbb{R}^2$ , since  $\mathbf{J}$  is a column vector with 3 values, the null space is spanned using two scalar vectors. Next, since

$$\mathbf{J}^1 \mathbf{v} = 0 \rightarrow (\ddot{\mathbf{x}}^2 - \ddot{\mathbf{x}}^1) = \mathbf{v} \rightarrow \ddot{\mathbf{x}}^2 = \ddot{\mathbf{x}}^1 + \mathbf{Z}_J \mathbf{u}^2, \quad (3.41)$$

it is possible to solve an optimization problem for  $\mathbf{u}^2$  instead of  $\ddot{\mathbf{x}}^2$ . This ensures that the solution of the second hierarchy will stay to keep the optimality of the rigid body motion.

There is however also a limitation due to Eq. (3.36) which limits the possible rigid body acceleration based on the forces. The value of the acceleration, based on Eq. (3.36) equals

$$\mathbf{U}_N^T \mathbf{M} \ddot{\mathbf{x}}^0 = -\mathbf{U}_N^T \mathbf{K} \mathbf{x} \rightarrow \ddot{\mathbf{x}}^0 = -\mathbf{M}^{-1} \mathbf{U}_N^T \mathbf{U}_N^T \mathbf{K} \mathbf{x}, \quad (3.42)$$

using the fact that for all the vectors in the singular value decomposition  $\mathbf{U} \mathbf{U}^T = \mathbf{I}$  where  $\mathbf{I}$  is the identity matrix. Similar to Eq. (3.40) the acceleration after the first optimization, must hold this constraint

$$\mathbf{U}_N^T \mathbf{M} \ddot{\mathbf{x}}^0 = \mathbf{U}_N^T \mathbf{M} \ddot{\mathbf{x}}^1 = -\mathbf{U}_N^T \mathbf{K} \mathbf{x} \rightarrow \mathbf{U}_N^T \mathbf{M} \ddot{\mathbf{x}}^1 - \mathbf{U}_N^T \mathbf{M} \ddot{\mathbf{x}}^0 = \mathbf{0} \rightarrow \mathbf{U}_N^T \mathbf{M} (\ddot{\mathbf{x}}^1 - \ddot{\mathbf{x}}^0) = \mathbf{0}. \quad (3.43)$$

So instead of optimizing over  $\ddot{\mathbf{x}}$ , the value  $\mathbf{u}^1$  is optimized where

$$a_R^1 = \mathbf{J}^1 \ddot{\mathbf{x}}^1 = \mathbf{J}^1 (\ddot{\mathbf{x}}^0 + \mathbf{Z}^0 \mathbf{u}^1) = \mathbf{J}^1 \mathbf{Z}^0 \mathbf{u}^1 + \mathbf{J}^1 \ddot{\mathbf{x}}^0 \quad (3.44)$$

where  $\mathbf{Z}^0$  is the *Orthonormal Basis of the Null Space* of  $\mathbf{U}_N^T \mathbf{M}$ . In this case,  $\mathbf{J}^1 \ddot{\mathbf{x}}^0$  is a bias on the optimization problem. Note that for the second priority, we have to solve in the null space of the null space of the constraints so:

$$\ddot{\mathbf{x}}^2 = \ddot{\mathbf{x}}^0 + \mathbf{Z}^0 (\mathbf{u}^1 + \mathbf{Z}^1 \mathbf{u}^2) = \ddot{\mathbf{x}}^0 + \mathbf{Z}^0 \mathbf{u}^1 + \mathbf{Z}^0 \mathbf{Z}^1 \mathbf{u}^2 = \ddot{\mathbf{x}}^1 + \mathbf{Z}^0 \mathbf{Z}^1 \mathbf{u}^2 \quad (3.45)$$

where  $\mathbf{Z}^1$  is the *Orthonormal Basis of the Null Space* of  $\mathbf{J}^1 \mathbf{Z}^0$ . This combination of null spaces  $\mathbf{Z}^0 \mathbf{Z}^1$  is denoted using  $\tilde{\mathbf{Z}}^1$ .

This can be generalized for any priority  $p$ , where the acceleration at priority  $p$  has to equal

$$\ddot{\mathbf{x}}^p = \ddot{\mathbf{x}}^{p-1} + \tilde{\mathbf{Z}}^{p-1} \mathbf{u}^p. \quad (3.46)$$

Here  $\ddot{\mathbf{x}}^{p-1}$  is the solution of the higher priority,  $\mathbf{u}^p$  are the scalars which can be optimized, and  $\tilde{\mathbf{Z}}^{p-1}$  is the mapping into the null-space of all the higher priorities so

$$\tilde{\mathbf{Z}}^{p-1} = \prod_{i=0}^{p-1} \mathbf{Z}^i \equiv \tilde{\mathbf{Z}}^{p-2} \mathbf{Z}^{p-1}. \quad (3.47)$$

with a goal of an optimizer is then to find  $\mathbf{u}^p$  to minimize

$$\|\mathbf{J}^p \ddot{\mathbf{x}}^p - \mathbf{a}^p\| = \left\| \mathbf{J}^p \left( \ddot{\mathbf{x}}^{p-1} + \tilde{\mathbf{Z}}^{p-1} \mathbf{u}^p \right) - \mathbf{a}^p \right\| = \left\| \mathbf{J}^p \tilde{\mathbf{Z}}^{p-1} \mathbf{u}^p + \mathbf{J}^p \ddot{\mathbf{x}}^{p-1} - \mathbf{a}^p \right\| \quad (3.48)$$

where  $\mathbf{a}^p$  is some target which should be a linear combination of  $\mathbf{x}$ . This linear combination is given by  $\mathbf{J}^p$ .

---

<sup>3</sup>In this case it is the ‘regular null-space’

Part II

Papers

# State Observation of Compliant Link Floating Base Robots

Sander W.S. Oosterveld, BSc and dr. ir. Arvid Q.L. Keemink

**Abstract**—This paper presents an approach using the Extended Kalman Filter (EKF) to estimate the states of a robot that is free to move and has a compliant link, while it only has rotary encoders on the joints. The dynamics of the robot are split into the movement of the body's center of mass and the flexible modes, both of which are estimated using the EKF. This information is used to calculate parameters that are crucial for balancing the robot, such as the centroidal momentum and full-body center of mass location. To demonstrate the significance of considering flexibility, the algorithm is applied to a planar quadruped with a compliant spine. The results indicate an improvement when it assumes a flexible spine, as opposed to an estimation that assumes only rigid links.

**Index Terms**—Extended Kalman Filter, Flexible Multibody systems, Floating base robots,

## I. INTRODUCTION

Robotic automation has led to one of the largest industrial revolutions in the past 20 years, with robots performing manual tasks instead of humans. However, currently, the majority of benefits from robotic automation are limited to factories or controlled environments. For a broader application of robots, they must be versatile, safe, and cost-effective. In addition, a robot should have a floating base, which allows it to move around [1.1].

One way to achieve these requirements is by incorporating compliant links. Compliance assists robots in grasping objects with a soft touch, making them more versatile [1.2]. Furthermore, research has shown that compliance improves safety in human-robot interactions, as link-compliance results in a lower impact force compared to joint compliance [1.3]. Additionally, compliant links can make robots cheaper and lighter than rigid robots, as they require less stiff or less material.

However, compliant links also present control challenges. The flexible links increase the number of degrees of freedom (DOFs) of the system, complicating both simulation and introducing hidden variables. Sensors can be added to the system to measure the additional DOFs, but this can be expensive [1.4]. An alternative solution is to use state estimation to estimate the flexible state.

In addition to flexible DOFs, a robot also requires an estimation of the whole body center of mass, body orientation, body velocities, and end effector positions. When, the robot moves using legs, control methods also depends on the centroidal momentum [1.5]. Typical sensors in robots (e.g., encoders) and accelerometers cannot directly measure these parameters.

State estimation allows a controller to estimate these parameters by combining measurement data and system dynamics. State estimation is a well-studied field that has been applied to various sectors such as robots, and navigation [1.6]. However, it has rarely been used to estimate the states of compliant link floating base robots. Therefore, the goal of this paper is to show, that state estimation can be applied to planar quadruped.

### A. Related work

The state estimation of quadruped and bipeds is a well-studied field [1.7]- [1.9]. In all these applications, the information of one or multiple inertial measurements units (IMUs) is combined with the joint angles and the feet location to estimate the states of the robot. All these papers focus on the challenges of spatial estimation, such as yaw-angle drift. The effect of flexible links is not discussed.

Even though the flexible states of a compliant beam are observable with only one sensor [1.10], they are susceptible to modelling errors. This makes estimation of the flexible modes using one sensor challenging [1.11]. Therefore, estimations of flexible multi body systems is often done using additional measurements such as strain gauges [1.12], vision [1.13] [1.14], inertial sensors [1.15], or end-effector location [1.16]. These sensors improve the observation of the flexible links.

In a cost-effective quadruped, it is not desirable to include additional sensors for the flexible spine. This paper therefore investigates if it is possible to estimate the state of a quadruped with a flexible spine without additional sensors. The focus is on the estimation of parameter essential to control legged robots, such as center of mass position and centroidal momentum. The problem is simplified by assuming the location of the



spine is exactly known, and the ground reaction forces can be measured.

### B. Contributions

The main contributions of this paper are giving a coherent integration of the following methods:

- 1) Present a quick method of computing the centroidal momentum and center of mass position based on the floating frame of reference modelling method
- 2) Apply state estimation to a robot with one flexible link and show an improved performance compared to state estimation with rigid links regarding momentum and center of mass estimation
- 3) Use modal expansion for the flexible link and show that taking modes with a lower eigenfrequency is sufficient in estimating the flexibility.

### C. Outline of the Paper

The goal of this paper is to show how a state estimator can estimate the states for a floating base flexible robot. Section II explains the modelling method for both the simulation model and the estimation model. Section III defines the used state estimation method. The final simulation parameters are described in Section IV. This section also includes the location of the sensors, the observers parameters and the chosen performance metrics. Section V presents these performance metrics as results and Section VI discusses the results. The final section presents the conclusions of this paper.

## II. MODEL OF FLOATING BASE FLEXIBLE LINK ROBOTS

In the modeling of flexible multibody systems, there are three commonly utilized approaches [1.17]. The first method is the finite element method, which divides the system into multiple elements, with the coordinate vector containing all the nodes of these elements. While this approach yields accurate results, it is often accompanied by long simulation times, making it impractical for real-time simulations of dynamic systems. In contrast, the second method, Absolute Nodal Coordinate Formulation (ANCF), utilizes only the position and slope of the nodes between the bodies, allowing for non-linear flexibility and the ability to handle large deformations while maintaining computational efficiency [1.18]. The third and applied method is the Floating Frame of Reference Formulation (FFRF), which separates the dynamics into rigid body motion and local flexible deformation, with the flexible deformation being based on linear mode shapes. However,

this approach is limited in its ability to model large deformations.

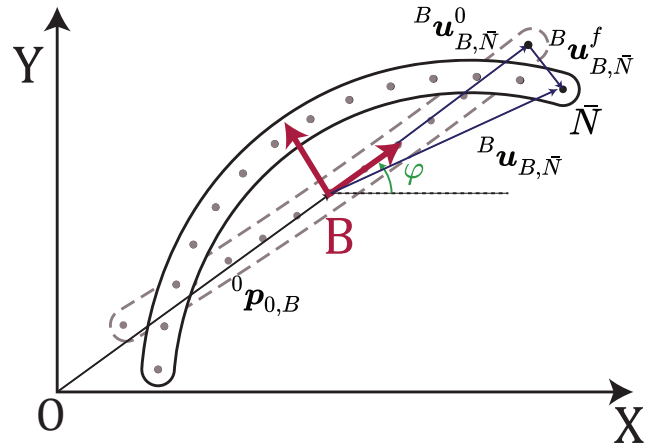


Fig. 1. The coordinates of flexible body B. Here,  ${}^x(\cdot)_{a,b}$  means the position vector in frame  $x$  from  $a$  to  $b$ .  $\varphi$  is the rotation regarding the origin.  $\bar{N}$  is the number of nodes in the link, where the first node is number 1 and the final node is numbered  $\bar{N}$ . The dashed light-gray line shows the undeformed body. The dots are the nodes in the body

Fig. 1 shows the coordinates for an FFRF body. The rigid body coordinates of body B,  $q_B^r$ , are a combination of the absolute position  ${}^0p_{0,B}$  and the orientation of the body  $\varphi$ . The location of a point,  $\bar{N}$  on the link, is given by a combination of a constant vector  ${}^B u_{B,\bar{N}}^0$  and a local flexible deformation  ${}^B u_{B,\bar{N}}^f$ . For clarity, a local body coordinate is denoted using a bar, e.g.  ${}^B u_{B,\bar{N}}^0$  becomes  $\bar{u}_{\bar{N}}^0$ . Also, the  $O$  in global coordinates will be omitted so  ${}^0p_{0,B} \equiv p_B$

The flexible coordinates define the amount of deformation of one body compared to the center of mass position. The relation between the flexible coordinate and the deformation is called the mode shape. This mode shape is free to choose, one method are the free-free modes. These modes are based on the eigenfrequencies of the linear elastic model given by

$$\bar{M}\ddot{\bar{u}}^f + \bar{K}\bar{u}^f = \bar{Q}, \quad (1.1)$$

where  $\bar{u}^f$  are the local node displacements (position and rotation),  $\bar{M}$  and  $\bar{K}$  are the full mass and stiffness matrix of one beam, and  $\bar{Q}$  are the nodal forces. The free-free modes are the vectors,  $\phi_\omega$ , which satisfy

$$\bar{M}\phi_{\omega_i} = \omega_i^2 \bar{K}\phi_{\omega_i}, \quad (1.2)$$

where  $\omega_i$  is the eigenfrequency corresponding to mode shape  $\phi_\omega$ . One property of free-free modes is that these modes do not change the location of the center of mass.

The strength of the floating frame of reference formulation is that it then only uses the bending modes with the lowest eigenfrequency. These modes contain the largest energy in the system and have the most effect. Using more or all mode shapes makes the

simulation more accurate, but it also slows down the computation. The mode shape matrix,  $\Phi_i$ , contains both the rigid body modes and the desired number of flexible modes such that

$$\bar{\mathbf{u}}_J^f = \begin{bmatrix} \phi_{\omega_1} & \dots & \phi_{\omega_{N_m}} \end{bmatrix} \begin{bmatrix} \mathbf{q}_J^{f,1} \\ \vdots \\ \mathbf{q}_J^{f,N_m} \end{bmatrix} = \Phi_J^f \mathbf{q}_J, \quad (1.3)$$

where  $\mathbf{q}_J$  are the coordinates corresponding to body  $J$ , and  $N_m$  is the number of flexible modes. This allows a modal expansion which reduces the number of coordinates based on the number of considered flexible modes to give the body mass and stiffness matrix as

$$\mathbf{M}_J \ddot{\mathbf{q}}_J + \mathbf{K}_J \mathbf{q}_J = \Phi_J^T \bar{\mathbf{Q}}, \quad (1.4)$$

where

$$\mathbf{M}_J = \Phi_J^T \bar{\mathbf{M}} \Phi_J, \text{ and } \mathbf{K}_J = \Phi_J^T \bar{\mathbf{K}} \Phi_J, \quad (1.5)$$

where

$$\Phi_J = \begin{bmatrix} \Phi_J^r & \Phi_J^f \end{bmatrix} \quad (1.6)$$

Here  $\Phi_J^r$  are the rigid body modes, which is a constant matrix that defines the velocity of all the nodes based on the velocity of the rigid body.

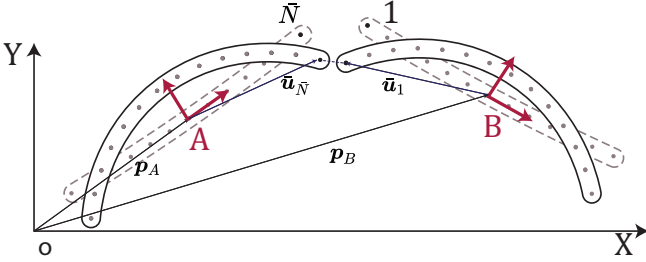


Fig. 2. Schematic of a rotational joint between bodies A and B.

The next step is combining the different bodies using the holonomic constraint matrix. Each connection of a body results in one or multiple holonomic constraints in the form:

$$f_A(\mathbf{q}_A) - f_B(\mathbf{q}_B) = 0, \quad (1.7)$$

where  $\mathbf{q}_A$ ,  $\mathbf{q}_B$  are the coordinates of body A and B, respectively, and  $f_A$ ,  $f_B$  map these coordinates to the constraint space. In a planar revolution joint, the absolute position of the joint should be the same when expressed from both body frames. In Fig. 2, the constraint equation is given by:

$$\mathbf{p}_A + {}^0\mathbf{R}_A \bar{\mathbf{u}}_{N} - (\mathbf{p}_B + {}^0\mathbf{R}_B \bar{\mathbf{u}}_1) = 0. \quad (1.8)$$

Here  ${}^0\mathbf{R}_A$  is the rotation matrix from body A to the origin depending on the orientation of body A, and  $\bar{\mathbf{u}}_{N} = {}^A\mathbf{u}_{A,N} = {}^A\mathbf{u}_{A,N}^0 + {}^A\mathbf{u}_{A,N}^f$  where

$${}^A\mathbf{u}_{A,N}^f = \Phi_{N}^f \mathbf{q}_A^f. \quad (1.9)$$

Here  $\Phi_{N}^f$  are the rows of the mode shape corresponding to the node  $\bar{N}$ . Since  ${}^A\mathbf{u}_{A,N}^0$  is constant, this part of the constraint depends solely on the coordinates of body A. A similar derivation can be done for body B. The holonomic constraint matrix,  $\mathcal{C}(\mathbf{q})$ , combines all the constraints of all the bodies on top of each other such that

$$\mathcal{C}(\mathbf{q}) = \mathbf{0}. \quad (1.10)$$

The equation of motion for the floating frame of reference formulation is given by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{Q}, \quad (1.11)$$

where  $\mathbf{q}$  is the combination of all the body coordinates and flexible coordinates

$$\mathbf{q} = \left[ (\mathbf{q}_A)^T \quad (\mathbf{q}_B)^T \quad \dots \quad (\mathbf{q}_{NB})^T \right]^T, \quad (1.12)$$

where, for example,

$$\mathbf{q}_A = \left[ ({}^0\mathbf{p}_{0,A})^T \quad \phi_A \quad (\mathbf{q}_A^f)^T \right]^T. \quad (1.13)$$

Both  $\mathbf{M}$  and  $\mathbf{K}$  are composed of the body mass and stiffness matrices, for example  $\mathbf{M} = \text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{Nb})$ .  $\mathbf{Q}$  are all the generalized forces acting on the bodies. These forces contain both forces due to the constraints forces between bodies,  $\mathbf{Q}_c$  and the applied forces,  $\mathbf{Q}_a$  due to gravity, actuators, and ground reaction forces.  $\mathbf{q}$  is a stack of all the body DOFs.

The d'Alemberts principle of virtual work can remove the constraint forces. A small variation of  $\mathbf{q}$  must correspond to a small variation in the constraint forces, which must be 0 as

$$\delta\mathcal{C} = [\mathcal{C}]_q \delta\mathbf{q} = 0. \quad (1.14)$$

Here  $\delta(\cdot)$  is a small variation of a variable and  $[(\cdot)]_q$  is the partial derivative with respect to  $\mathbf{q}$ . Since this must hold, 1.11 changes to

$$\mathbf{M}\ddot{\mathbf{q}} + [\mathcal{C}]_q^T \lambda = \mathbf{Q}_a - \mathbf{K}\ddot{\mathbf{q}}, \quad (1.15)$$

where  $\lambda$  are the lagrange multipliers, which ensure 1.14 holds. The constraint matrix also constrains  $\ddot{\mathbf{q}}$  since

$$\ddot{\mathcal{C}} = 0 \rightarrow [\mathcal{C}]_q \ddot{\mathbf{q}} + [[\mathcal{C}]_q \dot{\mathbf{q}}]_q \dot{\mathbf{q}} = 0. \quad (1.16)$$

Combining Eq. (1.15) and Eq. (1.16) gives the constrained equations of motion in augmented form

$$\begin{bmatrix} \mathbf{M} & [\mathcal{C}]_q^T \\ [\mathcal{C}]_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a - \mathbf{K}\ddot{\mathbf{q}} \\ -[[\mathcal{C}]_q \dot{\mathbf{q}}]_q \dot{\mathbf{q}} \end{bmatrix}. \quad (1.17)$$

Here the applied forces can be split up as

$$\mathbf{Q}_a = \mathbf{S}_\tau^T \boldsymbol{\tau} + \mathbf{J}_\rho \boldsymbol{\rho} + \mathbf{J}_{ext} \mathbf{Q}_{ext} + \mathbf{Q}_g, \quad (1.18)$$

where  $\boldsymbol{\tau}$  are the actuator torques,  $\boldsymbol{\rho}$  are the global ground reaction forces and  $\mathbf{Q}_{ext}$  is an external force on the system in the global frame.  $\mathbf{S}_\tau$ ,  $\mathbf{J}_f$ , and  $\mathbf{J}_{ext}$  map these force to the generalized coordinates.  $\mathbf{Q}_g$  is the generalized gravity force.

A differential equation in the form of equation (1.17) is called a differential algebraic equation. Only a limited set of state observers, called indirect Kalman filters, can directly estimate these types of equations [1.19]. However, it is possible to transform equation (1.17) into the form of an ordinary differential equation, as the full system dynamics depends uniquely on a smaller set of coordinates called the independent coordinates,  $\mathbf{q}^i$ . A designer has the freedom to choose any set of independent coordinates as long as they are independent; all remaining coordinates are called dependent coordinates,  $\mathbf{q}^d$ . For floating base robots, the standard choice is body rotation, flexible coordinates, and the position of one body. The acceleration of all coordinates can be determined from the independent coordinates using

$$\ddot{\mathbf{q}} = \Lambda^T \begin{bmatrix} \ddot{\mathbf{q}}^d \\ \ddot{\mathbf{q}}^i \end{bmatrix} = \mathbf{B}\mathbf{q}^i + \mathbf{b}, \quad (1.19)$$

where  $\Lambda$  is a perturbation matrix to reorder  $\mathbf{q}$  into the dependent and independent coordinates,

$$\mathbf{B} = \Lambda^T \begin{bmatrix} -([\mathcal{C}]_q^d)^{-1} ([\mathcal{C}]_q^i) \\ \mathbf{I} \end{bmatrix}, \quad (1.20)$$

$$\mathbf{b} = \Lambda^T \begin{bmatrix} -([\mathcal{C}]_q^d)^{-1} [[\mathcal{C}]_q \ddot{\mathbf{q}}]_q \dot{\mathbf{q}} \\ \mathbf{0} \end{bmatrix}. \quad (1.21)$$

Here  $(\cdot)^i$ ,  $(\cdot)^d$  are selections of matrices for the rows related to the (in)dependent coordinates,  $\mathbf{I}$  is an identity matrix. Using this transform, the equation of motion becomes

$$\mathbf{B}^T \mathbf{M} \mathbf{B} \ddot{\mathbf{q}}^i = \mathbf{B}^T (\mathbf{Q}_a - (\mathbf{K})^i \mathbf{q}^i - \mathbf{M} \mathbf{b}). \quad (1.22)$$

After integrating 1.22, the dependent coordinates follow from solving the holonomic constraint equation matrix.

### A. Centroidal parameters

Essential properties for control of legged robots are the location of the center of mass of the robot,  $c_R$ , and the centroidal momentum,  $h_R$ . These values can be computed from the DOFs of the system using

$$\mathbf{c}_R = \mathbf{H}_G \mathbf{q}, \quad (1.23)$$

where  $\mathbf{H}_G$  is the center of mass matrix. Since the degrees of freedom matrix contains the absolute positions

of the center of mass of all bodies  $\mathbf{H}_G$  is a constant matrix equal to

$$\mathbf{H}_G = \frac{1}{m_{tot}} \sum_i^{N_b} m_i \mathcal{S}_{c,i}. \quad (1.24)$$

Here  $m_{tot}$  is the total mass,  $m_i$  is the mass of the body  $i$ , and  $\mathcal{S}_{v,i}$  is a selection matrix for the center of mass coordinates of this body. Since the flexible modes do not change the center of mass of a body,  $\mathbf{H}_G$  depends solely on the rigid coordinates and is constant.

The centroidal linear momentum of the robot,  $\mathbf{k}_G$ , is the sum of the linear momentum of each body given by

$$\mathbf{k}_G = \mathbf{A}_{G,k} \dot{\mathbf{q}} = \sum_i^{N_b} m_i \dot{\mathbf{c}}_i = \left( \sum_i^{N_b} m_i \mathcal{S}_{c,i} \right) \dot{\mathbf{q}}. \quad (1.25)$$

This is similar to the Center of Mass matrix. The total centroidal angular momentum is given by the sum of the body angular momentum relative to the center of mass of the robot

$$\mathbf{l}_G = \sum_i^{N_b} ((\mathbf{c}_i - \mathbf{c}_R) \times \mathbf{k}_i + I_i \mathcal{S}_{\theta,i}), \quad (1.26)$$

where  $\mathcal{S}_{\theta,i}$  is a selection matrix of the rotation of the center of mass of body  $i$ . Substituting Eq. (1.25) and Eq. (1.24) gives

$$\mathbf{l}_G = \mathbf{q}^T \sum_i^{N_b} \left( \frac{m_i^2}{m_{tot}} \tilde{\mathcal{S}}_{c,i} \mathcal{S}_{c,i} - m_i \tilde{\mathbf{H}}_G \mathcal{S}_{c,i} \right) \dot{\mathbf{q}} + \sum_i^{N_b} I_i \mathcal{S}_{\theta,i} \dot{\mathbf{q}}, \quad (1.27)$$

where  $\tilde{(\cdot)}$  for a  $2 \times N$  matrix equals

$$\tilde{(\cdot)} = (\cdot)^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (1.28)$$

Finally Eq. (1.25) and Eq. (1.27) combine to become the centroidal momentum matrix,  $\mathbf{A}_G$ ,

$$\mathbf{h}_G = \begin{bmatrix} \mathbf{l}_G \\ \mathbf{k}_G \end{bmatrix} = \mathbf{A}_G \dot{\mathbf{q}}. \quad (1.29)$$

## III. STATE OBSERVING FLEXIBLE MULTI BODY SYSTEM

For flexible multibody systems, the state vector contains the independent coordinates and the first derivative of these coordinates, which define the dynamics of the robot and are essential for full-body control. However, sensors in a robotic system are not able to measure all of these states directly. A state observer combines sensor information with the system dynamics to estimate the state vector.

The most well-known state observer is the Kalman filter, which was first introduced by Kalman in 1960 [1.20]. However, the original filter is only suitable for linear systems and must be modified for use in non-linear systems. One such modification is the Extended Kalman Filter (EKF) developed by Julier [1.21], which uses local linearization to estimate the states.

The system for the extended kalman filter is given as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) \quad (1.30)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) \quad (1.31)$$

where  $\mathbf{x}_k = [(\mathbf{q}^i)^T, (\dot{\mathbf{q}}^i)^T]^T$ ,  $\mathbf{u}_k$  are the input torques, and  $\mathbf{y}_k$  are the measured outputs of the system, such as encoder angles or the orientation of an IMU.  $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$  follows from combining 1.22 with an integration scheme, and  $\mathbf{h}(\mathbf{x}_k)$  follows from relative rotations at joints based on the rigid body rotation and the flexible rotation.

The EKF starts with an initial estimation of the state,  $\hat{\mathbf{x}}_0$ , and its estimated covariance,  $\mathcal{X}_0$ . Based on these values, the current state and its covariance are estimated using

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k), \quad (1.32)$$

$$\mathcal{X}_k^- = ([\mathbf{f}]_{\hat{\mathbf{x}}_k^-}) \mathcal{X}_{k-1} ([\mathbf{f}]_{\hat{\mathbf{x}}_k^-})^T + \mathbf{Q}_k, \quad (1.33)$$

where  $(\cdot)^-$  is the a priori estimate, so before taking the current measurement into account.  $\hat{\mathbf{x}}_k$  is the estimated state at time step k.  $\mathbf{Q}_k$  is the estimated process noise on the system, a designer has to choose this value. Finally,  $[\mathbf{f}]_{\hat{\mathbf{x}}_k^-}$  is the numerical Jacobian of the state equation given by

$$[\mathbf{f}]_{\hat{\mathbf{x}}_k^-} = \left. \frac{\partial \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k^-}, \quad (1.34)$$

$$([\mathbf{f}]_{\hat{\mathbf{x}}_k^-})_{ij} = \frac{f_i(\hat{\mathbf{x}}_k^- + \Delta x \mathbf{e}_j, \mathbf{u}_k) - f_i(\hat{\mathbf{x}}_k^- - \Delta x \mathbf{e}_j, \mathbf{u}_k)}{2\Delta x} \quad (1.35)$$

where  $\mathbf{e}_j$  is a base vector with all zeros except at location  $j$ ,  $\Delta x$  is a design parameter set to  $10^{-3}$ .

In addition to the covariance of the estimated state, EKF also requires the estimated value and covariance of the measurements. Given as,

$$\hat{\mathbf{y}}_k^- = \mathbf{h}(\hat{\mathbf{x}}_k^-), \quad (1.36)$$

$$\mathcal{Y}_k^- = (\mathbf{H}_k^-) \mathcal{X}_k^- (\mathbf{H}_k^-)^T + \mathbf{R}_k, \quad (1.37)$$

where  $\mathbf{R}_k$  is the expected covariance in the measurements and  $\mathbf{H}_k^-$  is given by

$$\mathbf{H}_k^- = \left. \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k^-}, \quad (1.38)$$

and computed as in 1.35. Using these two covariance estimates, the Kalman gain is estimated to,

$$\mathbf{K}_k = \mathcal{X}_k^- (\mathbf{H}_k^-)^T (\mathcal{Y}_k^-)^{-1} \quad (1.39)$$

which determines how much to trust the state estimation. The state estimation is updated using the Kalman gain as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{y}}_k^-), \quad (1.40)$$

where  $\mathbf{z}_k$  are the real measurements. Finally, the estimated covariance is corrected using the Kalman gain:

$$\mathcal{X}_k = \mathcal{X}_k^- + \mathbf{K}_k \mathbf{H}_k^- \mathcal{X}_k^- \quad (1.41)$$

This paper applies a basic implementation of the extended kalman filter to focus on the required adaptations of the estimation.

## IV. METHOD

### A. Simulation of the Robot

The simulation presented in this paper is based on a half-quadraped, as illustrated in Fig. 3. This model is a planar version of a quadraped robot. The body of the quadraped in the simulation is flexible, with 5 flexible modes corresponding to the lowest eigenfrequencies. The material properties of the different links are provided in Table I.

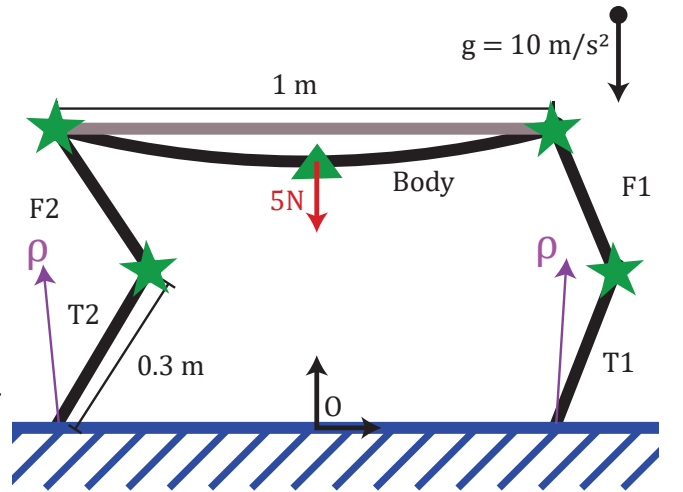


Fig. 3. Schematic of the half quadraped with location of the origin. The stars depict the location of the collocated encoders and actuators. The triangle shows the location of the position IMU.

The contact forces of a multi-contact robot cannot be computed directly and require an iterative method. *Hwangbo et al.* propose a method for quick multi-contact computation based on the impulse of the dynamics [1.22]. This method requires all forces acting on the system, including the constraint forces. Consequently, Eq. (1.17) is used to solve the simulation model, since it also yields the constraint forces. This simulation runs with a time step of  $10^{-4}$ s.

TABLE I  
PROPERTIES OF THE SIMULATION MODEL AND THE PLANT  
MODEL

Quantity	T1, T2, F1, F2 <sup>a</sup>	Plant Spine	Model Spine
Length	0.3 m	1 m	1 m
Width	0.15 m	0.15 m	0.15 m
Height	10 mm	8 mm	8 mm
Density	1 000 kg/m <sup>3</sup>	1 000 kg/m <sup>3</sup>	1 030 kg/m <sup>3</sup>
Young's Modulus	-	1 GPa	1 GPa
Modes	-	[8.2, 22.7, 44.5, 73.9, 110.9] rad/s	[8.2, 22.6] rad/s

<sup>a</sup> Identical for both the plant and the simulation model

<sup>b</sup> When the robot is flexible and no different number of modes is mentioned

### B. Measurement and Actuation of the Robot

The simulation model in this study includes three types of sensor output that the state estimation can utilize. All the joints have encoders that measure the local relative angle. This angle is a combination of the angle due to body rotation and the angle due to the flexible modes. This measurement includes a noise  $\mathcal{N}(0, 10^{-3} \text{ rad})$ , where  $\mathcal{N}(\mu, \sigma)$  defines Gaussian noise, where  $\mu$  is the mean value of the added noise and  $\sigma$  is the standard deviation of the noise. The body has a specialized IMU that provides the location and orientation of a point on the body with noise  $\mathcal{N}(0, 10^{-3} \text{ m}, 10^{-3} \text{ rad})$ . Additionally, the robot has a force sensor that measures the ground reaction forces with noise  $\mathcal{N}(0, 10^{-1} \text{ N})$ . The velocity of the feet is also included when there is a contact force, which is assumed to be zero with no noise throughout the simulation.

The actuators are co-located with the encoders, allowing for actuation at every joint. The controller's task is to maintain the original encoder angle using a proportional controller with gain  $K_p = 100 \text{ Nm/rad}$  at a frequency of 1kHz. The state estimator uses the output of the proportional controller as the torque acting on the system. However, an additional disturbance is added to the actuators to simulate unknown effects, which equals  $\mathcal{N}(0, 10^{-1} \text{ Nm})$ .

Lastly, a load causes an additional force,  $N_{load}$ , on the middle node at the back of the robot, which is not known to the state estimator. The state observer uses the dynamics of the estimation model with the measurements of the simulation model to estimate the state of the robot. The estimation model should have similar properties to the state of the simulation model, but it may differ in the number of modes taken into account for the body of the robot, as well as the density

of the links to account for modelling errors, as shown in Table I.

### C. Performance Metrics

The performance metrics in this study evaluate the performance of the algorithm. The first performance metric is based on the method presented in [1.23], where *Koolen et al.* demonstrate the robot's flexibility by analysing the estimated foot position. In a real robot, the feet remain fixed in position, but due to flexibility, the feet location varies relative to the original position. This comparison allows for examining of the differences between a rigid estimation model and a flexible estimation model.

The second performance metric is based on the centroidal parameters. Both the center of mass position and momentum are critical properties for a robot's balance [1.24]. This metric compares the differences between using flexible links and not using flexible links in the estimation model.

The final performance metric is the potential energy stored in the flexible link. Since the estimation model and simulation model use a different number of modes, this metric compares the effect of flexibility on actuator forces.

## V. RESULTS

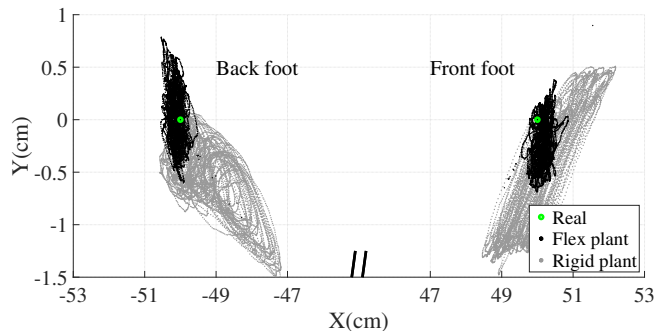


Fig. 4. Location of the two feet of the robot over a simulation of 10 seconds. A larger deviation from the real plant foot location means a worse estimation of the model.

Fig. 4 shows that the model with the compliant spine results in a smaller deviation in the foot position. When the plant model is rigid, the feet move with a standard deviation in the main principle axis of 0.93 cm and 0.84 cm for the front and back feet, respectively. If the plant model includes a compliant spine, this variation decreases to 0.23 cm and 0.26 cm, respectively.

Fig. 5 demonstrate that the estimation of the center of mass position is more accurate when flexibility is considered in the plant. The root-mean-square errors between the estimation and the real center of mass position are 0.22 cm and 0.31 cm for the rigid robot,

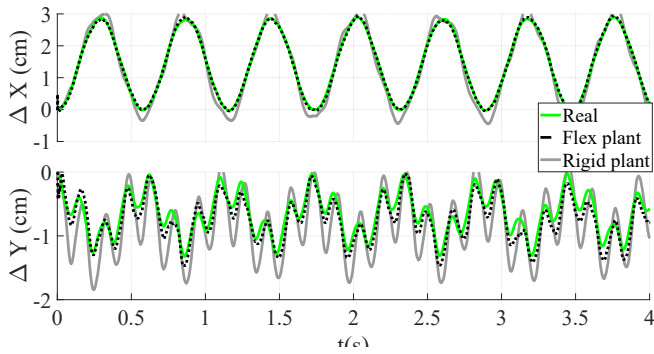


Fig. 5. The deviation of the center of mass using  $c_R = H_G \mathbf{q}$  over 10 seconds. The variation is caused by swaying due to active control of the robot.

and 0.04 cm and 0.13 cm for the flexible robot for the x and y direction, respectively.

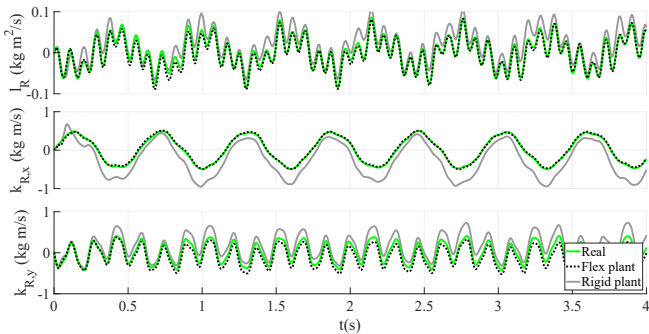


Fig. 6. The centroidal momentum of the robot split into the angular momentum  $l_R$  in the top plot and the linear momentum in the bottom two plots,  $k_{R,x}$  and  $k_{R,y}$ . This momentum is calculated using  $\mathbf{h} = [l_R \ k_{R,x} \ k_{R,y}]^T = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}}$

In the estimation of the momentum, the rigid plant exhibits an offset in the momentum, as shown in Fig. 6. This offset is caused by energy stored in the compliant spine of the robot, which negatively impacts the balance of the robot. Specifically, controlling based on this offset momentum will result in a constant positive momentum on the real robot.

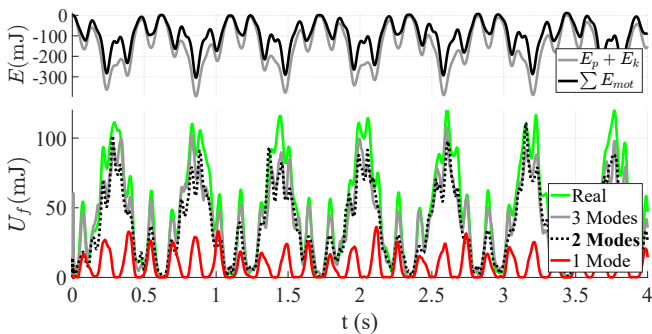


Fig. 7. The energy stored in the flexible beam relative to the number of modes in the plant, compared to the other energy sources in the robot. The potential energy,  $E_p$  equals  $\mathbf{Q}_g^T \Delta \mathbf{q}$ , the kinetic energy,  $E_k$  equals  $(\mathbf{q}^r)^T \mathbf{M}^r \mathbf{q}^r$ . The motor energy is computed by integrating the total supplied power  $P_{mot} = \boldsymbol{\tau}^T \mathbf{S}_r \dot{\mathbf{q}}$

The energy of the actuators is stored in the bending of the compliant spine, as shown by Fig. 7. In the real robot, the flexible beam element has 5 modes which store 40% of the supplied energy. The error in the energy stored decreases drastically when the model uses more than 1 mode. For one mode, the root-mean-square error is 50 mJ, while for 2 modes it has decreased to 16.3 mJ. Using 3 modes, only, decreases the error to 13.3 mJ. This indicates that the higher frequency store less energy and can be neglected in the model.

## VI. DISCUSSION

This study is simulation-based and has not been validated with real-world examples. The Extended Kalman Filter can use measurements that are not typically known, such as the location and orientation of the body node. Additionally, the simulation assumes all noise in the system is Gaussian and without an offset. This assumption is simulated by introducing a small error in body density, leading to a small offset in the total gravity vector and the ground reaction forces working to counteract this.

Even though the plant uses fewer mode shapes, the used mode shapes are still similar to the simulation model. For a more accurate comparison, the simulation model should use different mode shapes, such as Craig Bampton modes, or employ a different simulation method that allows for larger deformations. The main difference between the plant and plant model simulation is that the simulation model solves the *Constrained Equation of Motion in Augmented Form*, in Eq. (1.17). While the state observer simulates the model in the form of reduced form in Eq. (1.22).

The state estimation in this paper is separated from the control of the robot, which may result in potential instability not being shown. Additionally, there is no offset in the plant orientation at the start of the simulation, which can often cause instability for an Extended Kalman Filter. To address these issues, the Kalman filter can be modified to be more robust against initial offset, for example, by using the Unscented Kalman Filter (UKF), the Error-state extended Kalman filter (errorEKF), or the indirect Kalman Filter [1.25]. Additionally, the process noise could be better estimated by integrating it using Van Loan's method [1.25].

Finally, the excitation in this paper is limited as there is only a proportional controller on the relative angles. The addition of a balance controller enables the robot to begin walking, resulting in greater excitation with larger forces, a greater movement of the center of mass, and larger variations in momentum.

## VII. CONCLUSION

This paper demonstrates the feasibility of state estimation for a floating frame robot with a flexible spine by simplifying the system by only considering the planar case and by incorporating sensors for the position of the body and ground reaction forces. With these sensors, state estimation using an Extended Kalman Filter for a flexible link in the plant results in improved estimation of foot position, center of mass, and momentum. The observer's robustness is evaluated by running the observer at a lower frequency, varying the link density, and using fewer mode shapes.

In future work, it would be beneficial to apply this algorithm to a physical robot to test the observer's robustness. This means replacing the magical position IMU with a regular IMU and measuring the ground reaction forces using force sensors. As an alternative, the simulation of the robot can be performed using a different simulation method, such as the Absolute Nodal Coordinate Formulation, to gain a better understanding of the observer's robustness. Furthermore, it would be worthwhile to integrate the state observer into a controller for further testing.

## REFERENCES

- [1.1] C. Heyer, "Human-robot interaction and future industrial robotics applications," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4749–4754, 2010.
- [1.2] J. Becedas, I. Payo, and V. Feliu, "Two-flexible-fingers gripper force feedback control system for its application as end effector on a 6-dof manipulator," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 599–615, 2011.
- [1.3] Y. She, D. Meng, J. Cui, and H.-J. Su, "On the impact force of human-robot interaction: Joint compliance vs. link compliance," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6718–6723, 2017.
- [1.4] K. Gurses, B. J. Buckman, and E. J. Park, "Vibration Control of a Flexible Link Manipulator Using an Array of Fiber-Optic Curvature Sensors and Piezoelectric Actuators," *ASME International Mechanical Engineering Congress and Exposition*, vol. Volume 10: Mechanics of Solids and Structures, Parts A and B, pp. 417–423, 11 2007.
- [1.5] M. Abdallah and A. Goswami, "A biomechanically motivated two-phase strategy for biped upright balance control," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1996–2001.
- [1.6] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orłowska-Kowalska, and S. Katsura, "Industrial applications of the kalman filter: A review," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5458–5471, 2013.
- [1.7] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, "Pronto: A multi-sensor state estimator for legged robots in real-world scenarios," *Frontiers in Robotics and AI*, vol. 7, 2020.
- [1.8] N. Rotella, M. Blösch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," *CoRR*, vol. abs/1402.5450, 2014.
- [1.9] J. Li, Z. Yuan, S. Dong, J. Zhang, and J. Zhang, "A nonlinear state estimation framework for humanoid robots," *Robotics and Autonomous Systems*, vol. 153, p. 104100, 2022.
- [1.10] M. Balas, "Feedback control of flexible systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 673–679, 1978.
- [1.11] K. Cho, N. Hori, and J. Angeles, "On the controllability and observability of flexible beams under rigid-body motion," in *Proceedings IECON '91: 1991 International Conference on Industrial Electronics, Control and Instrumentation*, 1991, pp. 455–460 vol.1.
- [1.12] J. Lin and F. L. Lewis, "Enhanced measurement and estimation methodology for flexible link arm control," *Journal of robotic systems*, vol. 11, no. 5, pp. 367–385, 1994.
- [1.13] M. Mohammadi, Y. Shabbouei Hagh, X. Yu, H. Handroos, and A. Mikkola, "Determining the state of a nonlinear flexible multibody system using an unscented kalman filter," *IEEE Access*, vol. 10, pp. 40 237–40 248, 2022.
- [1.14] M. T. Hussein and D. Söffker, "State variables estimation of flexible link robot using vision sensor data," *IFAC Proceedings Volumes*, vol. 45, no. 2, pp. 193–198, 2012.
- [1.15] P. Mäkinen, T. Mononen, and J. Mattila, "Inertial sensor-based state estimation of flexible links subject to bending and torsion," in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2018, pp. 1–8.
- [1.16] V. Lertpiriyasuwat, M. C. Berg, and K. W. Buffinton, "Extended kalman filtering applied to a two-axis robotic arm with flexible links," *The International Journal of Robotics Research*, vol. 19, no. 3, pp. 254–270, 2000.
- [1.17] A. A. Shabana, *Dynamics of Multibody Systems*, 4th ed. Cambridge University Press, 2013.
- [1.18] M. Dibold, J. Gerstmayr, and H. Irschik, "A Detailed Comparison of the Absolute Nodal Coordinate and the Floating Frame of Reference Formulation in Deformable Multibody Systems," *Journal of Computational and Nonlinear Dynamics*, vol. 4, no. 2, 03 2009, 021006.
- [1.19] E. Sanjurjo, D. Dopico, A. Luaces, and M. Ángel Naya, "State and force observers based on multibody models and the indirect kalman filter," *Mechanical Systems and Signal Processing*, vol. 106, pp. 210–228, 2018.
- [1.20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [1.21] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.
- [1.22] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.
- [1.23] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Engelsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, pp. 1 650 007–1, 03 2016.
- [1.24] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 10 2013.
- [1.25] E. Sanjurjo, M. A. Naya, J. L. Blanco-Claraco, J. L. Torres-Moreno, and A. Giménez-Fernández, "Accuracy and efficiency comparison of various nonlinear kalman filters applied to multibody models," *Nonlinear Dynamics*, vol. 88, no. 3, pp. 1935–1951, 2017.

# Momentum-Based Control for Quadrupeds with a Compliant Spine

Sander W.S. Oosterveld, BSc and dr. ir. Arvid Q.L. Keemink

**Abstract**—This paper presents a method to balance quadrupeds with a compliant spine. Robots with compliant spines have been researched for their energy efficiency in walking, however, the control of these quadrupeds is still a challenge. We approach this challenge based on a framework to control rigid bipeds and quadrupeds, called the momentum-based controller. We show that this framework can be modified for robots with compliant elements and successfully apply quadruped in simulation. Finally, we check the robustness of the framework to modelling errors.

**Index Terms**—Momentum-based control, flexible multibody systems, quadrupeds

## I. INTRODUCTION

Two possible ways for robots to move around are wheeled motion and legged motion. While wheeled robots are easier to control, legged motion provides the ability to move over rougher terrain. In legged motion, the motion of each leg is independent. This allows legged robots to move over uneven terrain, which is not possible for wheeled robots.

However, this independence of the legs poses specific control challenges for legged robots. One such challenge is the redundancy in the actuators of a legged robot, similar to the abundance of control options available for human motion [2.1]. This results in infinite possibilities for control to perform a task. Additionally, a legged robot can only move around in an open space when a leg disconnects from the ground. This results in a changing number of contact points during a walking cycle.

The robot should maintain its balance throughout the entire walking cycle. Researchers have been attempting to solve this challenge since 1980. Initially, the focus was on full-body trajectory optimization, but this method is still infeasible for online balance control due to non-linearities. Therefore, the focus shifted to simplified models of the robot dynamics using the Linear Inverted Pendulum Model (LIPM) [2.2], Inverted Pendulum Model (IPM) [2.3], Linear Pendulum Model (LPM) [2.4], or Hybrid Zero Dynamics (HZD) [2.5]. *Abdallah and Goswami* show that humans control (angular) momentum to balance, and proposed using only the centroidal dynamics as a simplified system [2.6].

The application of momentum to balance a robot is called momentum-based control. This momentum-based control has shown good performance in balancing the robot, both in simulation [2.7]- [2.11] and on physical robots [2.12]- [2.15]. The robotic model in all these applications only uses rigid links to describe the robot. *Koolen et al.* notes that the physical robot has some link compliance, but simplifies this to joint compliance [2.14].

The design freedom of a robotic system will increase when the momentum-based framework also considers compliant links. The compliant links can improve the performance of a system by storing energy while walking, for example, in the spine [2.16]. Compliance can also result from cheaper production methods, such as 3D-printed plastics [2.17]. Therefore, link compliance can lead to better-performing and cheaper robots.

## A. Contributions

Flexibility in robotic systems have been applied, especially in the spines of different quadrupeds. In passive compliant spines, the robots are often controlled either through open-loop trajectory-based control [2.18] [2.19], energy-based control [2.20], or no control at all [2.21]. Alternatively, when there is a form of closed loop control, it often contains non-linear model predictive control [2.22]- [2.24]. Even though momentum-based control is applied to quadrupeds [2.25], there is no information on the effect of a compliant spine.

An unactuated compliant spine results in a further under-actuation of a quadruped robot. This paper applies momentum-based control to a robot with a compliant spine to show the effects of the compliant spine on the control. The contributions of this paper are:

- Extend the momentum-based control framework to include flexible links, resulting in a more general application of momentum-based control.
- Show that a compliant spine cannot be neglected in the controller design when using momentum-based control.
- Define the challenges of applying momentum-based control on a robot with a flexible spine.



## B. Outline of the Paper

The goal of the paper is to apply momentum-based control to a quadruped with a flexible spine. Section II explains the method for modelling a floating base robot with a flexible element. Section III explains hierarchical momentum-based control and how this can be modified to account for flexible elements. Based on the model and the momentum-based controller, Section IV shows how to apply this to a planar quadruped with a flexible spine. Section V shows the results of this application and Section VI discusses these results. The final section presents the conclusions of this paper.

## II. MODELLING OF COMPLIANT LINK ROBOTS

There are multiple ways to simulate compliant link robots [2.26]. This paper uses the floating frame of reference formulation. In this formulation, the flexible coordinates are separated from the rigid body coordinates, which allows the flexible coordinates to use linear flexibility theory. This speeds up the computation and allows the use of modal expansion to decrease the number of degrees of freedom.

For a planar robot, the coordinates of each body are a combination of rigid body coordinates and flexible coordinates, for a body  $A$

$$\mathbf{q}_A^T = \begin{bmatrix} \mathbf{c}_A^T & \varphi_A & (\mathbf{q}_A^f)^T \end{bmatrix} \quad (2.1)$$

where,  $\mathbf{q}$  is the vector of coordinates,  $\mathbf{c}_A$  is the center of mass position,  $\varphi_A$  is the orientation, and  $\mathbf{q}_A^f$  are the modal coordinates. These modal coordinates scale the deformation modes shapes to find displacement of each node in a body

$$\bar{\mathbf{u}} = \begin{bmatrix} \phi_1^f & \phi_2^f & \dots & \phi_{N_f}^f \end{bmatrix} \begin{bmatrix} q_1^f \\ q_2^f \\ \vdots \\ q_{N_f}^f \end{bmatrix} = \Phi^f \mathbf{q}^f, \quad (2.2)$$

where  $\bar{\mathbf{u}}$  are the nodal displacements and  $\phi_1^f$  are constant mode shapes.

The type of deformation mode shape is a design parameter in the simulation. One example, is to use an eigenfrequency analysis to create the so-called free-free modes. These are the modes with the lowest eigenfrequency for the linear elastic model given by

$$\bar{\mathbf{M}}\ddot{\bar{\mathbf{u}}} + \bar{\mathbf{K}}\bar{\mathbf{u}} = \bar{\mathbf{Q}}. \quad (2.3)$$

where  $\bar{\mathbf{M}}$ ,  $\bar{\mathbf{K}}$ , and  $\bar{\mathbf{Q}}$  are the finite element mass matrix, stiffness matrix and generalized forces. The eigenmodes are such that

$$(\bar{\mathbf{K}} - \omega_i^2 \bar{\mathbf{M}})\phi_i = \mathbf{0} \quad (2.4)$$

The first three modes always correspond to the rigid body modes,  $\Phi^r$ , i.e.

$$\bar{\mathbf{K}}\phi_i = \mathbf{0} \quad (2.5)$$

which is the motion of the center of mass of the body without exciting the flexibility and have an eigenfrequency of zero. Since all modes are linearly independent, all remaining modes, the flexible modes,  $\Phi^f$ , do not move the center of mass location. The flexible modes with a low eigenfrequency contain most of the energy, so often only those are used in simulation by only selected the first columns of  $\Phi^f$  [2.26].

Using these modes the dynamics for one body are defined by

$$\mathbf{M}_A \ddot{\mathbf{q}}_A + \mathbf{K}_A \mathbf{q} = \mathbf{Q}_c + \mathbf{Q}_a \quad (2.6)$$

where  $\mathbf{Q}_c$  are the generalized force due to the constraints and  $\mathbf{Q}_a$  are applied forces from actuators, gravity, or the ground reaction forces.  $\mathbf{M}_A$  and  $\mathbf{K}_A$  are the body mass and stiffness matrix found using the rigid and flexible modes, for example,

$$\mathbf{M}_A = \begin{bmatrix} \Phi^r & \Phi^f \end{bmatrix}^T \bar{\mathbf{M}} \begin{bmatrix} \Phi^r & \Phi^f \end{bmatrix} \quad (2.7)$$

These different equations of motions can be stacked to form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{Q}_c + \mathbf{Q}_a, \quad (2.8)$$

where  $\mathbf{q} = [\mathbf{q}_A^T, \mathbf{q}_B^T, \dots]^T$ , and  $\mathbf{M}$ ,  $\mathbf{K}$  are block diagonal matrices with body matrices

$$\mathbf{M} = \text{blkdiag} \{ \mathbf{M}_A, \mathbf{M}_B, \dots, \mathbf{M}_{N_b} \} \quad (2.9)$$

The constraint forces and applied forces are also the stacked versions of the forces on each body.

The actuated forces for the whole system are given by

$$\mathbf{Q}_a = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{Q}_g + \mathbf{J}_f^T \boldsymbol{\rho} + \mathbf{J}_{ext}^T \mathbf{W}_{ext}. \quad (2.10)$$

Here  $\boldsymbol{\tau}$  are the actuator torques,  $\boldsymbol{\rho}$  are the ground reaction forces in the global frame and  $\mathbf{W}_{ext}$  is an external wrench on the system in the global frame.  $\mathbf{S}$ ,  $\mathbf{J}_f$ , and  $\mathbf{J}_{ext}$  map these force to the generalized coordinates.  $\mathbf{Q}_g$  is the generalized gravity force.

Different bodies are connected using holonomic constraints given as

$$\mathcal{C}(\mathbf{q}) = \mathbf{0} \quad (2.11)$$

where,  $\mathcal{C}$ , contains all the constraints. Each revolute joint results in two holonomic constraints, reducing the number of degrees of freedom by two. Using d'Alembert principle Eq. (2.6) with an admissible field equal to

$$\delta \mathcal{C}(\mathbf{q}) = [\mathcal{C}]_q \delta \mathbf{q} = \mathbf{0} \quad (2.12)$$

where  $[(\cdot)]_q$  is the Jacobian with respect to  $q$ , gives

$$\mathbf{M}\ddot{q} + \mathbf{K}q = ([\mathcal{C}]_q)^T \lambda + Q_a \quad (2.13)$$

where  $\lambda$  are Lagrange multipliers. This equation is combined with the fact that

$$\ddot{C} = \mathbf{0} \rightarrow [\mathcal{C}]_q \ddot{q} + [[\mathcal{C}]_q \dot{q}]_q \dot{q} = 0 \quad (2.14)$$

to give the *constrained equations of motions in augmented form*

$$\begin{bmatrix} \mathbf{M} & ([\mathcal{C}]_q)^T \\ [\mathcal{C}]_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_a - \mathbf{K}q \\ - [[\mathcal{C}]_q \dot{q}]_q \dot{q} \end{bmatrix}. \quad (2.15)$$

Solving Eq. (2.15) requires inverting a large matrix, since it computes both the Lagrange multipliers and all the generalized coordinates. It is, however, possible to split all the coordinates into dependent,  $q^d$  and independent coordinates,  $q^i$ . The independent coordinates are sufficient to describe the dynamics of the system. For floating base robots, the standard choice is to use the body rotations, the flexible coordinates, and the position of one body

$$q^i = \begin{bmatrix} \varphi_A & \varphi_B & \dots & \varphi_{Nb-1} & (q_{Nb})^T & (q^f)^T \end{bmatrix}^T. \quad (2.16)$$

The acceleration of all coordinates can be determined from the independent coordinates using

$$\ddot{q} = \Lambda^T \begin{bmatrix} \ddot{q}^d \\ \ddot{q}^i \end{bmatrix} = \mathbf{B}\ddot{q}^i + b, \quad (2.17)$$

where  $\Lambda$  is a constant perturbation matrix to reorder  $q$  into the dependent and independent coordinates,

$$\mathbf{B} = \Lambda^T \begin{bmatrix} -(\mathcal{S}_{q^d}([\mathcal{C}]_q))^{-1} (\mathcal{S}_{q^i}[\mathcal{C}]_q) \\ \mathbf{I} \end{bmatrix}, \quad (2.18)$$

$$b = \Lambda^T \begin{bmatrix} -(\mathcal{S}_{q^d}(\mathcal{C}_q))^{-1} [[\mathcal{C}]_q \dot{q}]_q \dot{q} \\ \mathbf{0} \end{bmatrix}. \quad (2.19)$$

Here  $\mathcal{S}_x$  are selection matrices to select the rows related to  $x$ . For clarity all the parameter transformed to the independent coordinates are given a tilde. Using this transform, the equation of motion becomes

$$\mathbf{B}^T \mathbf{M} \mathbf{B} \ddot{q}^i = \mathbf{B}^T (Q_a - \mathcal{S}_{q^i} \mathbf{K} q^i - \mathbf{M} b). \quad (2.20)$$

when using the tilde form this equals

$$\tilde{\mathbf{M}} \ddot{q}^i = \tilde{\mathbf{S}}^T \tau + \tilde{\mathbf{J}}_f^T \rho + \tilde{\mathbf{Q}}^i(q, \dot{q}) \quad (2.21)$$

where

$$\tilde{\mathbf{Q}}^i(q, \dot{q}) = \mathbf{B}^T Q_g + \mathbf{B}^T \mathbf{J}_{ext}^T \mathbf{W}_{ext} - \tilde{\mathbf{K}} q^i - \mathbf{B}^T \mathbf{M} b, \quad (2.22)$$

and

$$\tilde{\mathbf{J}}_f^T = \mathbf{B}^T \mathbf{J}_f^T \quad (2.23)$$

After integrating 2.20, the dependent coordinates follow from the roots of the holonomic constraint equation matrix for the given independent coordinates. These roots can be found, for example, using the Newton–Raphson method [2.27].

### III. HIERARCHICAL MOMENTUM-BASED CONTROL

This section explains the momentum-based controller. It begins by introducing the basics of a quadratic program, followed by a description of the regular momentum-based controller. Finally, this section explains the concept of hierarchical momentum-based control, and concludes with the required modifications to apply it to a robot with compliant links.

#### A. Quadratic optimization

A quadratic program is an optimization of a problem in the form

$$x^* = \underset{x}{\operatorname{argmin}} \frac{1}{2} x^T \mathbf{H} x + f^T x \quad (2.24)$$

subject to

$$\begin{aligned} \mathbf{A} x &\leq b \\ \mathbf{A}_{eq} x &= b_{eq} \end{aligned} \quad (2.25)$$

where  $x^*$  is the output of the optimization.

The advantage of a problem in this formulation is that the problem is linear in and convex in the optimization parameters  $x$ . This makes solving this problem fast. This paper uses the Matlab `quadprog` function to solve a quadratic program.

In robotics,  $x$  often equals the vector of acceleration and the torques, so  $x = [\ddot{q}^T, \tau^T]^T$ . If the goal is to get an acceleration at point  $A$  to  $a_A^{des}$  then the algorithm would look like follows. This acceleration is given based on  $\ddot{q}$  as

$$a_p = \mathbf{J}_A \ddot{q} + \dot{\mathbf{J}}_A \dot{q} = \mathbf{J} x + d, \quad (2.26)$$

where  $d$  is called the bias and the Jacobian equals

$$\mathbf{J}_A = [p_A]_q, \quad \dot{\mathbf{J}}_A = [[p_A]_q \dot{q}]_q \quad (2.27)$$

the goal is then to minimize the absolute value of  $a_A - a_A^{des}$ . Using the L2 norm to minimise this using a weighting matrix,  $\mathcal{W}$ , is equivalent to

$$\min_x (a_A - a_A^{des})^T \mathcal{W} (a_A - a_A^{des}) \quad (2.28)$$

$$\min_x (\mathbf{J} x + d - a_A^{des})^T \mathcal{W} (\mathbf{J} x + d - a_A^{des}) \quad (2.29)$$

$$\min_x (x^T \mathbf{J}^T \mathcal{W} \mathbf{J} x + 2(d - a_A^{des})^T \mathcal{W} \mathbf{J} x) \quad (2.30)$$

with a residual part,  $r$ , not depending on  $x$  equal to

$$r = (b - a_A^{des})^T \mathcal{W} (b - a_A^{des}). \quad (2.31)$$

In addition to this optimization, there can also be a constraint on the system. There can be an equality constraint based on the dynamics given as  $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \boldsymbol{\tau}$  results in

$$\begin{bmatrix} \mathbf{M} & -\mathbf{S}^T \end{bmatrix} \mathbf{x} = -\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}). \quad (2.32)$$

Alternatively there can be an inequality constraint limiting the torques, for example

$$\begin{bmatrix} \mathbf{0}^{N_\tau \times N_q} & \mathbf{I}^{N_\tau} \\ \mathbf{0}^{N_\tau \times N_q} & -\mathbf{I}^{N_\tau} \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} \boldsymbol{\tau}_{\max} \\ -\boldsymbol{\tau}_{\max} \end{bmatrix}, \quad (2.33)$$

where  $\mathbf{0}^{x \times y}$  is a zero matrix with  $x$  rows and  $y$  columns. Similarly,  $\mathbf{I}^x$  is a square identity matrix with  $x$  rows and columns.  $N_\tau$  and  $N_q$  are the number of torques and generalized coordinates, respectively.

Rewriting these equations in the form of Eq. (2.24) gives

$$\begin{aligned} \mathbf{H} &= 2\mathbf{J}^T \mathcal{W} \mathbf{J}, & \mathbf{f}^T &= 2 \left( \mathbf{d} - \mathbf{a}_A^{des} \right) \mathcal{W} \mathbf{J}, \\ \mathbf{A}_{eq} &= \begin{bmatrix} \mathbf{M} & -\mathbf{S}^T \end{bmatrix}, & \mathbf{b}_{eq} &= -\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}), \\ \mathbf{A} &= \begin{bmatrix} \mathbf{0}^{N_\tau \times N_q} & \mathbf{I}^{N_\tau} \\ \mathbf{0}^{N_\tau \times N_q} & -\mathbf{I}^{N_\tau} \end{bmatrix}, & \mathbf{b} &= \begin{bmatrix} \boldsymbol{\tau}_{\max} \\ -\boldsymbol{\tau}_{\max} \end{bmatrix}. \end{aligned}$$

The Jacobian and biases of different optimization criteria stack for more complicated quadratic programs. The minimal value of the optimization reflects the quality of the optimization. Adding the residual,  $r$ , to this minimal value gives L2 norm of the error between the optimized acceleration and the target acceleration.

### B. Momentum-Based Control

*Koolen et al.* present the application of momentum-based control on the Atlas robot [2.14]. In the momentum-based control formulation, the optimization variables consist of only the acceleration,  $\ddot{\mathbf{q}}$  and the ground reaction forces,  $\boldsymbol{\rho}$ . The torques are computed based on the inverse dynamics, which is discussed in Section III-C. A momentum-based controller optimizes the rate of change of the centroidal momentum,  $\dot{\mathbf{h}}_G$ , based on the desired rate of change of the centroidal momentum,  $\dot{\mathbf{h}}_G^{des}$ . This results in

$$\underset{\mathbf{x}}{\operatorname{argmin}} \left( \dot{\mathbf{h}}_G - \dot{\mathbf{h}}_G^{des} \right)^T \mathcal{W}_h \left( \dot{\mathbf{h}}_G - \dot{\mathbf{h}}_G^{des} \right) + \mathbf{x}^T \mathcal{W}_r \mathbf{x} \quad (2.34)$$

where  $\mathcal{W}_h$  is the weighting matrix on the centroidal momentum and  $\mathcal{W}_r$  is a regulating term on the accelerations and the ground reaction forces. With the constraints,

$$\dot{\mathbf{h}}_G = \mathbf{J}_{c,f} \boldsymbol{\rho} + \mathbf{W}_g + \sum_i \mathbf{W}_{ext,i} \quad (2.35)$$

where  $\mathbf{J}_{c,f}$  is the jacobian of the centre of mass location to the feet location.  $\mathbf{W}_g$  and  $\mathbf{W}_{ext,i}$  are wrenches

on the center of mass of the gravity and external forces, respectively. Finally, *Koolen et al.* limits the ground reaction forces similar as done in Eq. (2.33) for the torques.

The centroidal momentum is essential in this framework, and based the definition from *Orin and Goswami* [2.28]

$$\dot{\mathbf{h}}_G = \mathbf{A}_G \ddot{\mathbf{q}} + \dot{\mathbf{A}}_G \dot{\mathbf{q}} \quad (2.36)$$

where  $\mathbf{A}_G$  is called the centroidal momentum matrix (CMM), which can be computed efficiently as shown in the Appendix. The desired change in centroidal linear momentum,  $\dot{\mathbf{k}}^{des}$  is based on PD control of the center of mass, the desired change of angular momentum,  $\dot{\mathbf{l}}^{des}$ , is a P control on a reference angular momentum,  $\mathbf{l}^{ref}$ . Both terms also include a feedforward reference term which combines to give

$$\dot{\mathbf{h}}_G^{des} = \begin{bmatrix} \mathcal{K}_{p,l} \left( \mathbf{l}_G^{ref} - \mathbf{A}_G \dot{\mathbf{q}} \right) + \dot{\mathbf{l}}_G^{ref} \\ \mathcal{K}_{p,k} \left( \mathbf{c}_R^{ref} - \mathbf{c}_R \right) + \mathcal{K}_{d,k} \left( \dot{\mathbf{c}}_R^{ref} - \dot{\mathbf{c}}_R \right) + \dot{\mathbf{k}}_G^{ref} \end{bmatrix} \quad (2.37)$$

where  $\mathcal{K}_{p,k}$ ,  $\mathcal{K}_{d,k}$  are the PD gains,  $\mathbf{c}_R$  is the center of mass position of the robot.  $\mathbf{c}_R^{ref}$  is the reference center of mass position of the robot. Finally,  $\mathbf{l}^{ref}$  is the reference angular momentum. Both reference values follow from the gait planner, the reference angular velocity is often equal to 0 [2.13].

### C. Hierarchical Momentum-Based Control

In Eq. (2.34) only the weighting between the momentum optimum and the regularization is important. There are, however, more tasks to optimize such as end-effector/feet location tracking, body orientations, contact forces, and the center of pressure location of the ground reaction forces [2.13].

The prioritization between these tasks is achieved through the chosen weight matrices for each parameter in a standard quadratic program. Often, one specific task has a higher priority over other tasks; the feet position tracking has a higher priority compared to body orientation. In this case, a hierarchical quadratic program can solve a quadratic program for each task priority separately [2.29].

The task priorities use the number 0 to N where the task with priority 0 has the highest priority. A task with priority,  $p$ , has a task Jacobian  $\mathbf{J}_p$ , bias  $\mathbf{d}_p$  and target  $\mathbf{t}_p$  such that the goal is

$$\min_x \|\mathbf{J}_p \mathbf{x} + \mathbf{d}_p - \mathbf{t}_p\| \quad (2.38)$$

subject to the same constraints as in Eq. (2.25). This task should remain optimal when solving the lower priority tasks. This is possible by limiting the solution space of the lower priority tasks to the null space of the

current task. So instead of optimizing  $\mathbf{x}$ , we optimize a variable,  $\mathbf{u}$  which parametrizes the nullspace

$$\mathbf{x}_p^* = \mathbf{x}_{p-1}^* + \mathbf{Z}_{p-1}\mathbf{u}_p^* \quad (2.39)$$

where  $\mathbf{x}_p^*$  is the  $\mathbf{x}$  after optimizing over the task with priority  $p$ .  $\mathbf{Z}_p$  is the null space basis of the task of priority  $p$ . These are the vectors corresponding to a zero value in a singular value deposition of the Jacobian at task  $p$ .

Substituting 2.39 into 2.38 gives an optimization task in the form

$$\mathbf{u}_p^* = \underset{\mathbf{u}_p}{\operatorname{argmin}} \|\mathbf{J}_p\mathbf{Z}_{p-1}\mathbf{u}_p + \mathbf{d}_p - \mathbf{t}_p + \mathbf{J}_p\mathbf{x}_{p-1}^*\|. \quad (2.40)$$

Also the (inequality) matrices change to

$$\begin{aligned} \mathbf{A}\mathbf{Z}_{p-1}\mathbf{u}_p &\leq \mathbf{b} - \mathbf{A}\mathbf{x}_{p-1}^* \\ \mathbf{A}_{eq}\mathbf{Z}_{p-1}\mathbf{u}_p &= \mathbf{b}_{eq} - \mathbf{A}_{eq}\mathbf{x}_{p-1}^* \end{aligned} \quad (2.41)$$

due to the substitution of Eq. (2.39) into Eq. (2.25). The null space for the next priority equals:

$$\mathbf{Z}_p = \mathbf{Z}_{p-1} \operatorname{null}(\mathbf{J}_p\mathbf{Z}_{p-1}) \quad (2.42)$$

#### D. Solving the zeroth Hierarchy for Compliant Link Robots

When *Herzog et al.* presents their Hierarchical Momentum-Based Controller, the first task ensures the resulting combination of contact forces and accelerations is feasible [2.13] with given motor locations. It is not possible for the center of mass of the robot to move up without ground reaction forces.

When the equation of motion is in the form,

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}}^i - \left(\tilde{\mathbf{J}}_\rho^T \boldsymbol{\rho} + \tilde{\mathbf{Q}}(\mathbf{q}, \dot{\mathbf{q}})\right) = \tilde{\mathbf{S}}^T \boldsymbol{\tau} \quad (2.43)$$

This system is under actuated since  $\operatorname{rank}(\tilde{\mathbf{S}}) = N_q - 3 - N_f$ . So, the force in some direction has to equal 0, these directions correspond to the *basis of the left null space* of  $\tilde{\mathbf{S}}^T$ , given the letter  $\hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}$ . This gives

$$\hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\tilde{\mathbf{M}}\ddot{\mathbf{q}}^i - \hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\mathbf{B}^T\mathbf{J}_\rho^T\boldsymbol{\rho} = \hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}). \quad (2.44)$$

For a rigid system,

$$\hat{\mathbf{Z}}_{\tilde{\mathbf{S}}} = \begin{bmatrix} \mathbf{I}^3 & \mathbf{0}^{3 \times N_q - 3} \end{bmatrix} \Lambda_b \quad (2.45)$$

where  $\Lambda_b$  is a perturbation matrix to reorder the coordinates into first the floating base coordinates and then all the other coordinates. The formulation in Eq. (2.44) can be applied to any under actuated system.

The solution of the first hierarchy solves Eq. (2.44) when the ground reaction forces are zero,

$$\mathbf{x}_0^* = \left[ \left(\hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\tilde{\mathbf{M}}\right)^\dagger \hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) \quad \mathbf{0} \right]. \quad (2.46)$$

where  $\left(\hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\tilde{\mathbf{M}}\right)^\dagger = \tilde{\mathbf{M}}^{-1}\hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}^T$  the corresponding null space basis at priority 0 then equals

$$\mathbf{Z}_0 = \operatorname{null} \left( \begin{bmatrix} \hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\tilde{\mathbf{M}} & \hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}\mathbf{B}^T\mathbf{J}_\rho^T \end{bmatrix} \right) \quad (2.47)$$

After solving the different priorities, the momentum-based controller returns the desired accelerations and ground reaction forces. These cannot be applied directly to the robot and need to be transformed to torques. This requires the singular value deposition of  $\tilde{\mathbf{S}}^T$ , which equals

$$\tilde{\mathbf{S}}^T = \begin{bmatrix} \mathbf{U}_C & \mathbf{U}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_C \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T, \quad (2.48)$$

where  $\mathbf{U}_N^T \equiv \hat{\mathbf{Z}}_{\tilde{\mathbf{S}}}$ . Eq. (2.44) ensures that there are no torques in the direction of  $\mathbf{U}_N$ , therefore  $\tilde{\mathbf{S}}^T \boldsymbol{\tau} = \mathbf{U}_C \boldsymbol{\Sigma}_C \mathbf{V}^T \boldsymbol{\tau}$ . The inverse dynamics then solve for  $\boldsymbol{\tau}^{des}$

$$\boldsymbol{\tau} = \mathbf{V} (\boldsymbol{\Sigma}_C)^{-1} \mathbf{U}_C^T \left( \tilde{\mathbf{M}}\ddot{\mathbf{q}}^{des} - \tilde{\mathbf{J}}_\rho^T \boldsymbol{\rho}^{des} + \tilde{\mathbf{Q}}^i(\mathbf{q}, \dot{\mathbf{q}}) \right), \quad (2.49)$$

where  $\ddot{\mathbf{q}}^{des}$  and  $\boldsymbol{\rho}^{des}$  are the desired acceleration of the independent coordinates and the desired ground reaction forces. These are the parameter optimized by the quadratic program. The matrix,  $\mathbf{V} (\boldsymbol{\Sigma}_C)^{-1} \mathbf{U}_C^T$ , is a pseudo inverse of  $\tilde{\mathbf{S}}^T$  and will be denoted with  $\left(\tilde{\mathbf{S}}^T\right)^\dagger$

#### IV. APPLYING MOMENTUM-BASED CONTROL TO A QUADRUPEL WITH A COMPLIANT SPINE

Fig. 1 shows different blocks of applying momentum-based control on a robot with a compliant spine. This section discusses the different parts of this block diagram. For clarity, the ‘plant’ refers to the simulation of a real robot, the ‘plant model’ refers to a model of the plant used by the controller and state estimator. The ‘plant model’ used for control is not perfect. Therefore, the models are similar but not the same.

##### A. Plant simulation

The goal of the simulation is to mimic a physical robot, so it should be realistic. The plant is a planar version of a quadruped, as shown in Fig. 2. The dimensions and properties of the different links are given in Table I. Only the spine of the robot is compliant.

The contact forces of a multi-contact robot cannot be computed directly and require an iterative method. *Hwangbo et al.* propose a method for quick multi-contact computation based on the impulse of the dynamics [2.30]. This method requires all forces acting on the system, including the constraint forces. Therefore, Equation 2.15 is used to solve the simulation model, as it also yields the constraint forces. The

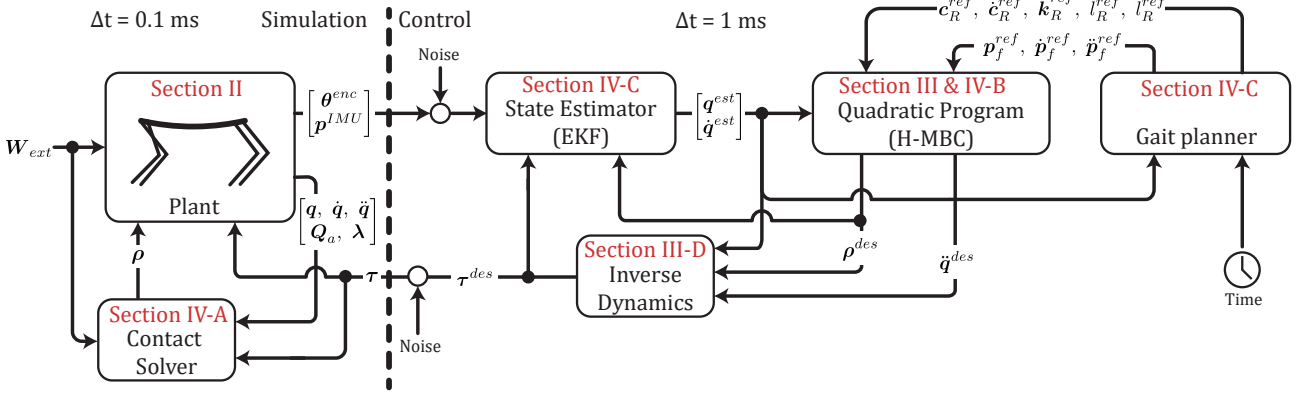


Fig. 1. Structure of the Hierarchical Momentum-Based framework as used in this paper. Including the relevant sections for each of the blocks. The simulation part uses the ‘plant’ to simulate the robot, while the control part uses the ‘plant model’.

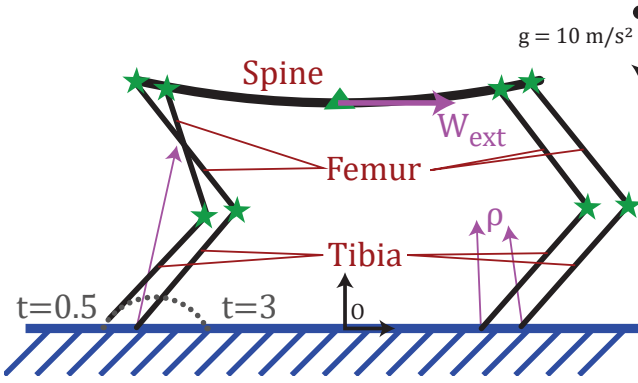


Fig. 2. Schematic of the quadruped with location of the origin. The stars depict the location of the collocated encoders and actuators. The triangle shows the location of the position IMU. The dotted line shows the desired movement of the leg.

simulation of the plant runs with a time step of 0.1 ms using the Symplectic Euler Integration Scheme [2.31].

Eight actuators control each of the legs using direct torque control. Noise is added to simulate motor dynamics, with a value of  $\mathcal{N}(0, 0.2\text{Nm})$ . Here,  $\mathcal{N}(\mu, \sigma)$  defines Gaussian noise, where  $\mu$  is the mean value of

the added noise and  $\sigma$  is the standard deviation of the noise.

In addition to the actuators, the plant has two types of sensors. Firstly, there are eight encoders on the knee and hip joints, measuring the relative local angles,  $\theta$ , including any additional angles due to bending. Additionally, there is a non-physical IMU, which measures the position and orientation of the center node on the spine,  $p^{IMU}$ , including any possible bending. The noise on these measurements is  $\mathcal{N}(0, 10^{-3}\text{rad})$  and  $\mathcal{N}(0, 10^{-3}\text{m})$ , respectively.

## B. Quadratic Program

The hierarchical quadratic program contains four priority levels, as shown in Table II. Note that both tasks in priority 2, can be absent. Section III explains how to solve these different priorities. The zeroth hierarchy is solved as shown in Section III-D. The remaining priorities are defined in this section. First, the required (in)equality matrices will be explained.

The zeroth hierarchy ensures that the dynamics hold so the (in)equality constraints only relate to the maximum torque, and allowed ground reaction forces. The maximum torques follows from rewriting Eq. (2.49)

$$\begin{aligned} (\tilde{\mathbf{S}}^T)^\dagger [\tilde{\mathbf{M}} \quad \tilde{\mathbf{J}}_\rho^T] \mathbf{x} &\leq -(\tilde{\mathbf{S}}^T)^\dagger \mathbf{q}^i + \tau_{\max} \\ -(\tilde{\mathbf{S}}^T)^\dagger [\tilde{\mathbf{M}} \quad \tilde{\mathbf{J}}_\rho^T] \mathbf{x} &\leq (\tilde{\mathbf{S}}^T)^\dagger \mathbf{q}^i - \tau_{\max} \end{aligned} \quad (2.50)$$

where  $\mathbf{x} = [\tilde{\mathbf{q}}^T, \boldsymbol{\rho}]^T$  and  $\tau_{\max}$  is the maximum torque, set to 10 Nm. For the ground reaction forces, there are two constraints. First, there can only be a ground reaction force when the feet are (supposed to be) on the ground so for foot  $i$

$$\mathbf{A}_{f,i} \boldsymbol{\rho}_i = \mathbf{I}^2 (\mathbf{y}_{f_i} \geq \epsilon_\rho) \begin{bmatrix} \rho_{x,i} \\ \rho_{y,i} \end{bmatrix} = 0 \quad (2.51)$$

TABLE I

PROPERTIES OF THE PLANT AND THE PLANT MODEL

Quantity	Tibia/Femur <sup>a</sup>	Plant Spine	Plant Model Spine
Length	0.3 m	1 m	1 m
Width	0.15 m	0.15 m	0.15 m
Height	10 mm	13 mm	13 mm
Density	1 000 kg/m <sup>3</sup>	1 000 kg/m <sup>3</sup>	1 010 kg/m <sup>3</sup>
Young's Modulus	-	1 GPa	1.02 GPa
Modes	-	[13.4, 36.9, 72.3, 120.0, 180.2] rad/s <sup>b</sup>	[13.4, 37.0] rad/s <sup>b</sup>

<sup>a</sup> Identical for both the plant and the plant model.

<sup>b</sup> Maximum number of modes to be considered.

TABLE II  
HIERARCHIES OF THE MOMENTUM-BASED CONTROLLER

Priority	Related Equations	Weight	Goal
0	Eq. (2.44)	— <sup>a</sup>	Result is feasible
1	Eq. (2.55) Eq. (2.56)	— <sup>a</sup>	Contact feet stay at contact points
2	Eq. (2.57) Eq. (2.55) Eq. (2.56)	$\mathcal{W}_{spine}$ $\mathcal{W}_f$	Torso does not rotate <sup>b</sup> Swing feet follows trajectory <sup>c</sup>
3	Eq. (2.37) Eq. (2.58) Eq. (2.59)	$\mathcal{W}_h$ $\mathcal{W}_\rho$	Balance control on momentum Regularization of GRFs

<sup>a</sup> There is no weighting required because there is only one task

<sup>b</sup> Only when 1 mode is used in the controller.

<sup>c</sup> Possibly, this does not exist when all feet should be in contact, so when all feet are on the ground, there might be only 3 priorities.

where  $y_{f,i}$  is the y-position of foot  $i$ ,  $\epsilon_\rho$  is a small value of  $10^{-2}$  m to counteract instability in the estimation. Combining these gives

$$\begin{bmatrix} \mathbf{0} & \text{blkdiag} \{ \mathbf{A}_{f,1}, \mathbf{A}_{f,2}, \mathbf{A}_{f,3}, \mathbf{A}_{f,4} \} \end{bmatrix} \mathbf{x} \leq \mathbf{0} \quad (2.52)$$

The second limitation is that the ground reaction force should be within the linearized friction cone [2.32]:  $\rho_x < \mu\rho_y$ ,  $-\rho_x > -\mu\rho_y$ , and  $\rho_y > 0$ . This gives for foot  $i$

$$\mathbf{A}_{\rho,i} \boldsymbol{\rho}_i \equiv \begin{bmatrix} 1 & -\mu \\ -1 & -\mu \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \rho_{x,i} \\ \rho_{y,i} \end{bmatrix} \leq 0, \quad (2.53)$$

where  $\mu$  is the friction coefficient and set to 1. Combining this for the four feet gives

$$\begin{bmatrix} \mathbf{0} & \text{blkdiag} \{ \mathbf{A}_{\rho,1}, \mathbf{A}_{\rho,2}, \mathbf{A}_{\rho,3}, \mathbf{A}_{\rho,4} \} \end{bmatrix} \mathbf{x} \leq \mathbf{0}, \quad (2.54)$$

First, the controller should ensure that the feet stay on the floor. When the feet release from the floor, they cannot enact a ground reaction force, while this is essential to control momentum. For each foot touching the floor, the highest priority task is a PD control on the feet location, such that

$$\ddot{\mathbf{p}}_{f,i}^{des} = \mathcal{K}_{p,f} (\mathbf{p}_{f,i} - \mathbf{p}_{f,i}^{ref}) + \mathcal{K}_{d,f} (\dot{\mathbf{p}}_{f,i}^{ref} - \dot{\mathbf{p}}_{f,i}) + \ddot{\mathbf{p}}_{f,i}^{ref} \quad (2.55)$$

where  $\mathbf{p}_{f,i}$  is the absolute position of the feet and  $\mathbf{p}_{f,i}^{ref}$  is the reference position of the feet. This acceleration is related to the vector of independent coordinates using Eq. (2.17)

$$\ddot{\mathbf{p}}_{f,i} = \mathbf{J}_{f,i} \mathbf{B} \ddot{\mathbf{q}}^i + \dot{\mathbf{J}}_{f,i} \dot{\mathbf{q}} + \mathbf{J}_{f,i} \mathbf{d}. \quad (2.56)$$

The goal is to minimize the difference between Eq. (2.55) and Eq. (2.56) for all the feet, which should be on the floor based on the gait planner. *Herzog et al.* use a constraint of the acceleration of the feet instead of a PD control [2.13], this, however, shows undesirable results due to the unknown disturbances.

The second priority ensures the robot follows the gait planner. These are PD controller on the spine body angle,  $\varphi_{Sp}$ , and the swing foot trajectory. The task for the swing feet is identical compared to the contact feet in Eq. (2.55) and Eq. (2.56). The coordinate for the spine angle is one of the states, so the Jacobian is a constant selection

$$\varphi_{Sp} = \hat{\mathbf{e}}_{11}^T \mathbf{q}^i \rightarrow \ddot{\varphi}_{Sp} = \hat{\mathbf{e}}_{11}^T \ddot{\mathbf{q}}^i \quad (2.57)$$

where  $\hat{\mathbf{e}}_x$  is a vector with zero at all places except at position  $x$ , where it is 1. The desired value of this angular acceleration is similar to Eq. (2.55) but with different scalar  $K_{p,\varphi}$  and  $K_{d,\varphi}$ .

The final priority handles the balance control of the robot by weighting between centroidal momentum and the ground reaction forces. The target for the momentum is defined in Eq. (2.37). The difference between this target and

$$\dot{\mathbf{h}}_G = \mathbf{A}_G \mathbf{B} \dot{\mathbf{q}}^i + \mathbf{A}_G \mathbf{b} + \dot{\mathbf{A}}_G \mathbf{B} \dot{\mathbf{q}} \quad (2.58)$$

should be minimized. The regularization of the ground reaction forces is done using

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{0}^{1 \times N_a} & \mathbf{1}^{1 \times N_\tau} \end{bmatrix} \mathbf{x} \right\|. \quad (2.59)$$

All the PD values are scaled to ensure the damping factor is  $1/\sqrt{2}$ , the crossover frequencies and a crossover frequency of 3.6 rad/s. For the angular momentum,  $K_{p,l} = 1$ . The weights used in the quadratic program are given in Table III.

TABLE III  
HIERARCHIES OF THE MOMENTUM-BASED CONTROLLER

Task	Weight
(Swing)Foot positions	$\mathcal{W}_f = 1$
Spine orientation	$\mathcal{W}_{spine} = 2$
Momentum	$\mathcal{W}_h = \text{diag}([10, 10, 50])$
GRF	$\mathcal{W}_{\rho,i} = \text{diag}([0.5, 0.1])$

### C. State estimation and gait planner

The encoder angles are not sufficient to define the dynamics for the plant model. The plant model uses Eq. (2.20) and therefore requires the vector of independent coordinates,  $\mathbf{q}^i, \dot{\mathbf{q}}^i$ . A state observer estimates these independent coordinates based on the measurements and the system dynamics.

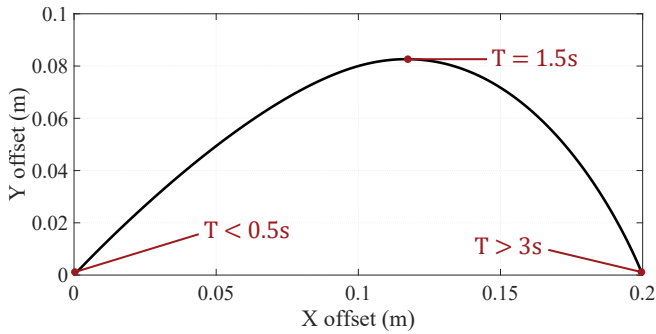


Fig. 3. Target trajectory of one of the feet during the simulation

In the presence of measurement and process noise, the Kalman Filter can optimally estimate these independent coordinates [2.33]. For the non-linear system, *Jullier* presents a linearized version named the Extended Kalman Filter (EKF). The state vector consists of  $\mathbf{q}_k^i, \dot{\mathbf{q}}_k^i$ , which is a discrete vector at step  $k$ . Using Symplectic Euler Integration [2.31], the system is then defined as

$$\begin{bmatrix} \mathbf{q}_{k+1}^i \\ \dot{\mathbf{q}}_{k+1}^i \end{bmatrix} = \begin{bmatrix} \mathbf{q}_k^i + \Delta T(\dot{\mathbf{q}}_k^i + \Delta T \ddot{\mathbf{q}}_k^i) \\ \dot{\mathbf{q}}_k^i + \Delta T \ddot{\mathbf{q}}_k^i \end{bmatrix} \quad (2.60)$$

where  $\ddot{\mathbf{q}}_k^i$  depends on the input torque, ground reaction forces and the previous state in Eq. (2.20).

The measurement used by the EKF are the encoder angles, IMU position and the desired feet velocities. These can all be found as a linear combination of the state variables. In addition, the state estimator uses the desired torques,  $\boldsymbol{\tau}^{des}$  and desired ground reaction forces,  $\mathbf{r}^{des}$ .

In addition to the states, the momentum-based controller also requires the desired momentum, feet position and spine orientation. This is not the focus of this paper; therefore, the target is to move one leg on the trajectory shown in Fig. 3. The desired center of mass position, body orientation and angular momentum are their original values.

## V. RESULTS

During movement of the foot, the robot receives two disturbance pushes of 5 N for 0.1 seconds—one push at time zero and one push when  $t = 1.5s$ . As illustrated in Figure 4, the results for the recovery of the center of mass position suggest that the compliant spine should be considered when modelling the plant. When both the plant and the plant model are similar, the controller controls the center of mass back to the desired position with the expected natural frequency ( $3.6rad/s$ ) and damping ratio ( $1/\sqrt{2}$ ).

However, when the flexibility in the models is inconsistent, the system becomes unstable. This is

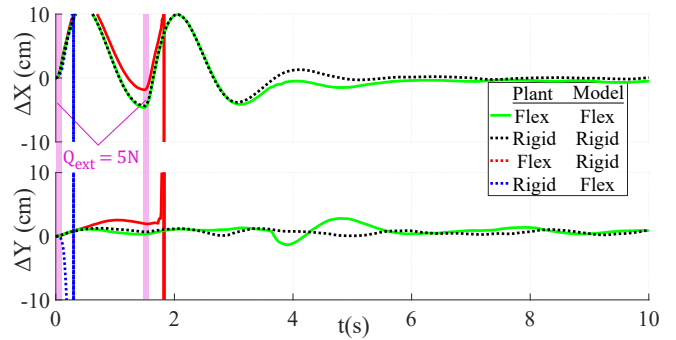


Fig. 4. Recovery off a push for different plants and plant models. When a plant (model) is flexible, it has a compliant spine. In the plant, the compliant spine is modelled with 5 flexible modes, while in the plant model the compliant spine has 2 flexible modes. The two plots show the variation in the center of mass from the original position, the goal of the controller is to keep the center of mass at the original position.

particularly pronounced when the model has a compliant spine, but the plant does not. One explanation for this instability is that a flexible model requires additional actuator torque, which would put energy into the compliant spine. If the robot does not have a compliant spine, this additional force causes the feet to detach from the floor, resulting in the robot collapsing. Similarly, when the plant has a flexible spine, but the model does not take this into account, the system is unstable.

There is always a discrepancy between a physical robot and its model, as seen in Fig. 5, which highlights the limited robustness of the algorithm. Observations of unstable trials suggest that the orientation of the spine rotates, causing one leg to overextend, which leads to the robot falling. To address this, a compensation term is proposed in Table II, with a priority of 2. However, combining this task with 2 modes results in unstable solutions. As a result, the next tests will

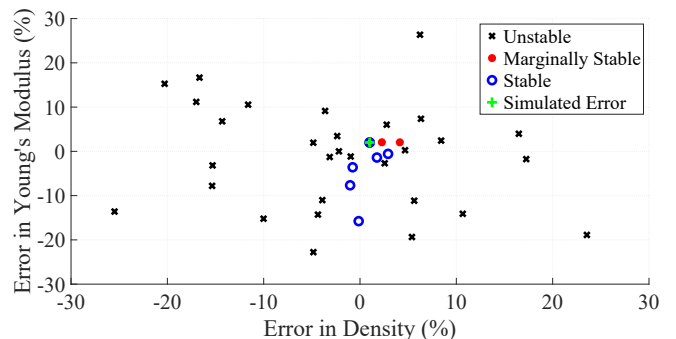


Fig. 5. Stability check related to error between plant and plant model, an error of 10% means the value in the plant model is 1.1 times the value in the plant. In the unstable cases the robot drops, in the marginally stable cases the flexible spine keeps oscillating after 10 seconds. The stable tests result in a damped vibration of the flexible spine.

be conducted using only one mode in the plant.

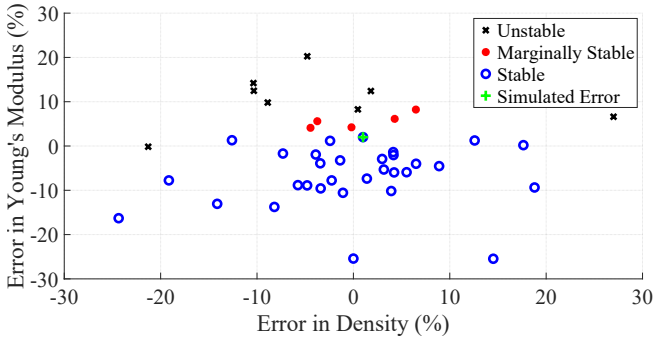


Fig. 6. Stability check related to error between plant and plant model, similar to Fig. 5. In this simulation, both the plant and the plant model only contain 1 mode shape.

Fig. 6 shows an increase in the robustness of the algorithm. Still, the stability of the algorithm deteriorates when the Young's modulus of the model is higher than that of the plant. This causes the output torque from the inverse dynamics to increase, leading to oscillations. The variation in the density seems to only have a minor influence on the stability of the solution when the plant contains one mode shape.

In a real system, there are more modes than considered in the model. Fig. 7 shows the effect of this. The algorithm can handle this difference, but a larger difference in the number of modes causes the oscillations to increase. With 5 modes in the model, the system is unstable, while with 3 modes it is (marginally) stable. The final plot in Fig. 7 shows that using two modes in the model with the spine correction causes the controller to become unstable almost instantly. Since the body orientation is required for the robustness, solving this will be left to future work.

## VI. DISCUSSION

The simulations in this paper demonstrate that it is possible to incorporate compliant links within a

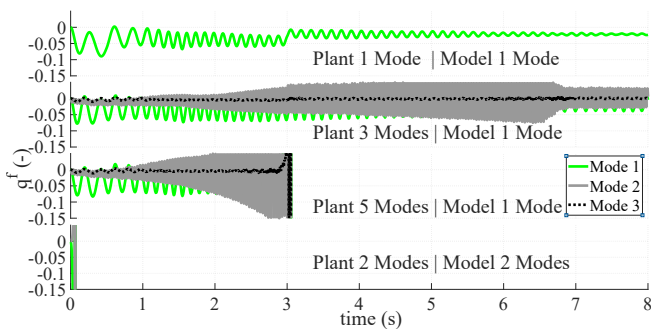


Fig. 7. The amount of excitation of the flexible modes in the robot. In the first three plots the model has one flexible mode and the plant has an increasing number of flexible modes. In the last plot, both the plant and the plant model have 2 flexible modes

momentum-based framework, but the application of this approach is limited. The framework requires an accurate model of the flexible modes, which is often difficult to obtain. An alternative approach, called dual estimation, which combines state and parameter estimation, may help to reduce the impact of modelling errors on the system [2.34].

A significant limitation of the algorithm is that it is unable to produce a robustly stable solution when the spine has more than one flexible mode. This is not due to the state estimator, as it also fails in when the robot states are used directly. The main limitation is that when there are two flexible modes in the plant, the robot is unable to control the orientation of the spine. Preliminary testing has shown that this is due to the interaction between the feet position task in priority 1 and the spine orientation task. A possible solution is to put both tasks in the same hierarchy or controlling the body orientation through the angular momentum.

The tasks in this algorithm do not depend on the flexible mode. It would be beneficial to include some control tasks based on the compliance of the spine to induce more damping. For example, the flexing of the spine could be minimized by using sliding mode control [2.35].

Finally, the robot in the simulation maintains full contact with the floor for most of the time, which greatly reduces its movement freedom, as it results in eight constraints. It would be interesting to focus on the stages when the robot does not have full contact with the floor, as it can use the swing feet to generate additional momentum. This is similar to a biped using its arms to balance [2.10].

## VII. CONCLUSION

This paper presents a hierarchical momentum-based controller applied to a planar quadruped, and investigates its performance. The results indicate that when there is one flexible mode, the controller performs well, but it cannot handle two modes. Additionally, the stability of the system is highly dependent on accurate parameter estimation. The study indicates that the system can tolerate a deviation of approximately 10% before the system becomes unstable

In future work, the issue with multiple modes in the model should be addressed, and the framework should be tested on a spatial system. In addition, the performance of this framework should be compared with a model which incorporates joint flexibility.



APPENDIX  
CENTROIDAL MOMENTUM IN FLOATING FRAME  
FORMULATION

Essential properties for control of legged robots are the location of the center of mass of the robot,  $c_R$ , and the centroidal momentum,  $h_R$ . These values can be computed from the coordinates of the system using

$$c_R = \mathbf{H}_G \mathbf{q} \quad (2.61)$$

where  $\mathbf{H}_G$  is the center of mass matrix. Since the degrees of freedom matrix contains the absolute positions of the center of mass of all bodies  $\mathbf{H}_G$  is a constant matrix equal to

$$\mathbf{H}_G = \frac{1}{m_{tot}} \sum_i^{N_b} m_i \mathcal{S}_{c,i}. \quad (2.62)$$

Here  $m_{tot}$  is the total mass,  $m_i$  is the mass of the body  $i$ , and  $\mathcal{S}_{v,i}$  is a selection matrix for the center of mass coordinates of this body. Since the flexible modes do not change the center of mass of a body,  $\mathbf{H}_G$  depends solely on the rigid coordinates and is constant.

The centroidal linear momentum of the robot,  $k_G$ , is the sum of the linear momentum of each body given by

$$k_G = \mathbf{A}_{G,lin} \dot{\mathbf{q}} = \sum_i^{N_b} m_i \dot{c}_i = \left( \sum_i^{N_b} m_i \mathcal{S}_{c,i} \right) \dot{\mathbf{q}}. \quad (2.63)$$

This is similar to the Center of Mass matrix. The total centroidal angular momentum is given by the sum of the body angular momentum relative to the center of mass of the robot

$$l_G = \sum_i^{N_b} ((c_i - c_R) \times k_i + I_i \mathcal{S}_{\theta,i}), \quad (2.64)$$

where  $\mathcal{S}_{\theta,i}$  is a selection matrix of the rotation of the center of mass of body  $i$ . Substituting Eq. (2.63) and Eq. (2.62) gives

$$l_G = \left[ \mathbf{q}^T \sum_i^{N_b} \left( \frac{m_i^2}{m_{tot}} \tilde{\mathcal{S}}_{c,i} \mathcal{S}_{c,i} - m_i \tilde{\mathbf{H}}_G \mathcal{S}_{c,i} \right) + \sum_i^{N_b} I_i \mathcal{S}_{\theta,i} \right] \dot{\mathbf{q}} \quad (2.65)$$

where  $(\tilde{\cdot})$  for a  $2 \times N$  matrix equals

$$(\tilde{\cdot}) = (\cdot)^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (2.66)$$

Finally Eq. (2.63) and Eq. (2.65) combine to become the centroidal momentum matrix,  $\mathbf{A}_G$ ,

$$h_G = \begin{bmatrix} l_G \\ k_G \end{bmatrix} = \mathbf{A}_G \dot{\mathbf{q}} \quad (2.67)$$

The derivative of  $A_G$  follows directly from Eq. (2.65) and equals

$$\dot{\mathbf{A}}_G = \begin{bmatrix} \dot{\mathbf{q}}^T \sum_i^{N_b} \left( \frac{m_i^2}{m_{tot}} \tilde{\mathcal{S}}_{c,i} \mathcal{S}_{c,i} - m_i \tilde{\mathbf{H}}_G \mathcal{S}_{c,i} \right) \\ \mathbf{0} \end{bmatrix} \quad (2.68)$$

such that

$$\dot{h} = \mathbf{A}_G \ddot{\mathbf{q}} + \dot{\mathbf{A}}_G \dot{\mathbf{q}} \quad (2.69)$$

REFERENCES

- [2.1] M. Latash, "There is no motor redundancy in human movements. there is motor abundance," *Motor Control*, vol. 4, no. 3, pp. 259 – 261, 2000.
- [2.2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, 2001, pp. 239–246 vol.1.
- [2.3] T. Sugihara, Y. Nakamura, and H. Inoue, "Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1404–1409.
- [2.4] M. Shibuya, T. Suzuki, and K. Ohnishi, "Trajectory planning of biped robot using linear pendulum mode for double support phase," in *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*. IEEE, 2006, pp. 4094–4099.
- [2.5] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.
- [2.6] M. Abdallah and A. Goswami, "A biomechanically motivated two-phase strategy for biped upright balance control," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1996–2001.
- [2.7] A. MacChietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.
- [2.8] S. H. Lee and A. Goswami, "Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, Conference Proceedings, pp. 3157–3162.
- [2.9] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-February, 2015, Conference Proceedings, pp. 295–302.
- [2.10] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, 2016.
- [2.11] L. Righetti and A. Herzog, "Momentum-centered control of contact interactions," in *Springer Tracts in Advanced Robotics*, 2017, vol. 117, pp. 339–359.
- [2.12] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Compliant locomotion using whole-body control and divergent component of motion tracking," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, 2015, Conference Proceedings, pp. 5726–5733.

- [2.13] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.
- [2.14] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Engelsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, 2016.
- [2.15] C. A. Laubscher, R. J. Farris, and J. T. Sawicki, "Angular momentum-based control of an underactuated orthotic system for crouch-to-stand motion," *Autonomous Robots*, vol. 44, no. 8, pp. 1469–1484, 2020.
- [2.16] Q. Cao and I. Poulakakis, "Passive quadrupedal bounding with a segmented flexible torso," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2484–2489.
- [2.17] Engineering Toolbox, "Young's modulus, tensile strength and yield strength values for some materials," Online, 2003, available: [https://www.engineeringtoolbox.com/young-modulus-d\\_417.html](https://www.engineeringtoolbox.com/young-modulus-d_417.html).
- [2.18] T. Takuma, M. Ikeda, and T. Masuda, "Facilitating multimodal locomotion in a quadruped robot utilizing passive oscillation of the spine structure," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4940–4945.
- [2.19] M. Khoramshahi, A. Badri-Spröwitz, A. Tuleu, M. Nili Ahmadabadi, and A. Ijspeert, "Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot," *Proceedings of 2013 IEEE International Conference on Robots and Automation*, 05 2013.
- [2.20] G. A. Folkertsma, A. J. van der Schaft, and S. Stramigioli, "Morphological computation in a fast-running quadruped with elastic spine," *IFAC-PapersOnLine*, vol. 48, no. 13, pp. 170–175, 2015, 5th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control LHMNC 2015.
- [2.21] M. H. H. Kani, M. Derafshian, H. J. Bidgoly, and M. N. Ahmadabadi, "Effect of flexible spine on stability of a passive quadruped robot: Experimental results," in *2011 IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 2793–2798.
- [2.22] A. P. Sabelhaus, A. K. Akella, Z. A. Ahmad, and V. Sun-Spiral, "Model-predictive control of a flexible spine robot," in *2017 American Control Conference (ACC)*, 2017, pp. 5051–5057.
- [2.23] A. P. Sabelhaus, A. H. Li, K. A. Sover, J. R. Madden, A. R. Barkan, A. K. Agogino, and A. M. Agogino, "Inverse statics optimization for compound tensegrity robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3982–3989, 2020.
- [2.24] S. Bhattacharya, G. K. Ananthasuresh, and A. Ghosal, "Design of a one-dimensional flexible structure for desired load-bearing capability and axial displacement," *Mechanics Based Design of Structures and Machines*, vol. 46, no. 3, pp. 376–399, 2018.
- [2.25] M. Liu, D. Qu, F. Xu, F. Zou, P. Di, and C. Tang, "Quadrupedal robots whole-body motion control; based on centroidal momentum dynamics," *Applied Sciences*, vol. 9, no. 7, p. 1335, Mar 2019.
- [2.26] A. A. Shabana, *Dynamics of multibody systems*. Cambridge university press, 2003.
- [2.27] K. Atkinson, *An introduction to numerical analysis*. John wiley & sons, 1991.
- [2.28] D. E. Orin and A. Goswami, "Centroidal momentum matrix of a humanoid robot: Structure and properties," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 653–659, 2008.
- [2.29] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.
- [2.30] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.
- [2.31] J. Niiranen, "Fast and accurate symmetric euler algorithm for electromechanical simulations," in *Electrimacs 99 (modelling and simulation of electric machines converters an& systems)*, 1999, pp. 1–71.
- [2.32] C. Klein and S. Kittivatcharapong, "Optimal force distribution for the legs of a walking machine with friction cone constraints," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 73–85, 1990.
- [2.33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [2.34] E. Wan, R. van der Merwe, and A. Nelson, "Dual estimation and the unscented transformation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999.
- [2.35] Q.-l. Hu, Z. Wang, and H. Gao, "Sliding mode and shaped input vibration control of flexible systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 503–519, 2008.

## Part III

# Conclusion

## Chapter 6

# Conclusion

This thesis presented two papers that investigate the effects of including a compliant spine in a planar quadruped. The first paper showed that the inclusion of a compliant spine improves the state estimation, even when the plant and the model are different. The second paper showed that a momentum-based controller can be applied to a quadruped with a compliant spine, but the controller is sensitive to modelling errors.

The papers in this thesis demonstrated that a compliant spine can be included in a planar quadruped, and that it improves the performance of both the state estimation and control algorithm. The literature review provided an overview of the broad subjects of state-estimation of robotic systems and control of legged robots, and revealed that there is limited research in adding compliant links to legged robots. Both in the fields of state estimation and control, the models only include joint flexibility.

The first paper applied state estimation to a quadruped with a compliant spine and revealed that the rotary encoders with the position of the body and the ground reaction forces are sufficient to estimate the flexible states. When the state estimator does not use a model including the compliant spine, the quality of the center of mass and momentum estimation decreases. However, the inclusion of a compliant spine did not require additional steps, and it improved the estimation even when the plant and the plant model are different, with the inclusion of two flexible modes.

The second paper modelled a quadruped with a compliant spine and applies the momentum-based controller. The controller used the states from the state-estimator discussed in paper 1. The first step in the quadratic program proposed by [10] must be modified to include the under-actuation of the flexible spine. With the adaptation, a momentum-based control could be applied to a quadruped with a compliant spine, but the controller was only robust when it included one mode. Additionally, the controller was sensitive to an erroneous estimation of the Young's modulus, indicating that the controller, as it was applied in the paper, was not robust to modelling errors.

## Chapter 7

# Future work

Even though the work in this thesis shows promising results regarding the implementation of compliant links in the legged robots, there is still additional research required before this can be implemented. This section will give an overview of subjects which could be interesting to investigate.

This thesis demonstrates the potential of incorporating compliant links into legged robots, and compares the results to those of a completely rigid robot. The flexible joint model is not discussed in the comparison. However, the successful application of this simplification in several studies[11, 43] suggests that further investigation into the benefits of modelling flexible link, rather than flexible joints, should be conducted.

The Extended Kalman Filter used in the state estimation and momentum-based controller can be replaced with newer techniques such as the Unscented Kalman Filter, or extended to include the flexible mode uncertainty. Furthermore, the tasks of the momentum-based controller should be modified to include compliance. The positive effects of a compliant spine can then be investigated.

When implementing compliant links into a robot's controller, the algorithm must first be improved to ensure robustness. Currently, the algorithm is not robust enough for practical applications. Adding a parameter estimator increases robustness against parameter errors, but can also lead to parameter instability. Currently, the controller is limited to include one mode robustly. Removing the spine orientation as a task eliminates the direct instability, however, does not produce a robustly stable solution. An alternative task which results in a robust solution when more than one mode is included is left as future work.

Once the robustness of the algorithm is improved, more realistic use cases should be investigated. A more complex simulation model, incorporating a nodal finite element method to estimate bending, actuator dynamics, lower level control, and the lack of position and absolute ground reaction force sensors, can better estimate the performance of the controller and state estimator. Such a model would follow the approach taken by *Bloesch et al.*[8].

Finally, an extension to 3D should be made. This not only add more degrees of freedom, but also induces additional flexible modes such as torsion. This extension requires large changes in both the model and the controller. For a spatial model, the different modelling methods should be compared again, since modelling of three-dimensional flexible beams brings some additional challenges[68]. In addition, the computation of spatial momentum should be extended to the three-dimensional case.

# Bibliography

- [1] R. Pearlman, K. Kane, and H. Euan-Smith, “The very human future of work,” *Korn ferry briefings*, 2017.
- [2] M. Dolan, “Global worker shortages may only worsen post-covid,” *reuters*, 9 2022.
- [3] J. Mokyr and R. H. Strotz, “The second industrial revolution, 1870-1914,” *Storia dell’economia Mondiale*, vol. 21945, no. 1, 1998.
- [4] M. Latash, “There is no motor redundancy in human movements. there is motor abundance,” *Motor Control*, vol. 4, no. 3, pp. 259–261, 2000.
- [5] Y. She, D. Meng, J. Cui, and H.-J. Su, “On the impact force of human-robot interaction: Joint compliance vs. link compliance,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6718–6723, 2017.
- [6] Q. Cao and I. Poulakakis, “Passive quadrupedal bounding with a segmented flexible torso,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2484–2489.
- [7] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak *et al.*, “Present and future of slam in extreme underground environments,” *arXiv preprint arXiv:2208.01787*, 2022.
- [8] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, “State estimation for legged robots on unstable and slippery terrain,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 6058–6064.
- [9] M. A. Hopkins, D. W. Hong, and A. Leonessa, “Compliant locomotion using whole-body control and divergent component of motion tracking,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, 2015, Conference Proceedings, pp. 5726–5733.
- [10] A. Herzog, N. Rotella, S. Mason, F. Grimmering, S. Schaal, and L. Righetti, “Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid,” *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.
- [11] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Engelsberger, and J. Pratt, “Design of a momentum-based control framework and application to the humanoid robot atlas,” *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1650007, 2016.
- [12] C. A. Laubscher, R. J. Farris, and J. T. Sawicki, “Angular momentum-based control of an underactuated orthotic system for crouch-to-stand motion,” *Autonomous Robots*, vol. 44, no. 8, pp. 1469–1484, 2020.
- [13] M. Liu, D. Qu, F. Xu, F. Zou, P. Di, and C. Tang, “Quadrupedal robots whole-body motion control; based on centroidal momentum dynamics,” *Applied Sciences*, vol. 9, no. 7, p. 1335, Mar 2019.

- [14] OpenAI. chatGPT (version january 2023). [Online]. Available: <https://chat.openai.com>
- [15] T. D. Barfoot, *State estimation for Robotics*. Cambridge University Press, 2022.
- [16] S. Julier and J. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” in *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.
- [17] E. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.
- [18] T. Chen, J. Morris, and E. Martin, “Particle filters for the estimation of a state space model,” in *European Symposium on Computer-Aided Process Engineering-14*, ser. Computer Aided Chemical Engineering, A. Barbosa-Póvoa and H. Matos, Eds. Elsevier, 2004, vol. 18, pp. 613–618.
- [19] J. Zhao, “Dynamic state estimation with model uncertainties using  $h_\infty$  extended kalman filter,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1099–1100, 2018.
- [20] H. Afshari, S. Gadsden, and S. Habibi, “Gaussian filters for parameter and state estimation: A general review of theory and recent trends,” *Signal Processing*, vol. 135, pp. 218–238, 2017.
- [21] E. Sanjurjo, M. A. Naya, J. L. Blanco-Claraco, J. L. Torres-Moreno, and A. Giménez-Fernández, “Accuracy and efficiency comparison of various nonlinear kalman filters applied to multibody models,” *Nonlinear Dynamics*, vol. 88, no. 3, pp. 1935–1951, 2017.
- [22] M. Balas, “Feedback control of flexible systems,” *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 673–679, 1978.
- [23] P. C. Hughes and R. E. Skelton, “Controllability and observability for flexible spacecraft,” *Journal of guidance and control*, vol. 3, no. 5, pp. 452–459, 1980.
- [24] T. Williams and X. Cheng, “Degrees of controllability and observability for close modes of flexible space structures,” *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1791–1795, 1999.
- [25] Y. Li and G. Wang, “Sensing strategies for a flexible manipulator,” *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 3, pp. 565–572, 2000.
- [26] N. G. Chalhoub and G. A. Kfoury, “Development of a robust nonlinear observer for a single-link flexible manipulator,” in *ASME International Mechanical Engineering Congress and Exposition*, vol. 47063, 2004, pp. 851–861.
- [27] J. Lin and F. L. Lewis, “Enhanced measurement and estimation methodology for flexible link arm control,” *Journal of robotic systems*, vol. 11, no. 5, pp. 367–385, 1994.
- [28] I. Palomba, D. Richiedei, and A. Trevisani, “Reduced-order observers for nonlinear state estimation in flexible multibody systems,” *Shock and Vibration*, vol. 2018, 2018.
- [29] H. Yang, J. Liu, and X. Lan, “Observer design for a flexible-link manipulator with pde model,” *Journal of Sound and Vibration*, vol. 341, pp. 237–245, 2015.
- [30] M. Mohammadi, Y. Shabbouei Hagh, X. Yu, H. Handroos, and A. Mikkola, “Determining the state of a nonlinear flexible multibody system using an unscented kalman filter,” *IEEE Access*, vol. 10, pp. 40 237–40 248, 2022.

- [31] M. T. Hussein and D. Söffker, “State variables estimation of flexible link robot using vision sensor data,” *IFAC Proceedings Volumes*, vol. 45, no. 2, pp. 193–198, 2012.
- [32] M. Morlock, C. S. Markus Burkhardt, and R. Seifried, “Nonlinear state estimation for trajectory tracking of a flexible parallel manipulator,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3449–3454, 2017, 20th IFAC World Congress.
- [33] P. Mäkinen, T. Mononen, and J. Mattila, “Inertial sensor-based state estimation of flexible links subject to bending and torsion,” in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2018, pp. 1–8.
- [34] K. Tatsis, L. Wu, P. Tiso, and E. Chatzi, “State estimation of geometrically non-linear systems using reduced-order models,” *Life Cycle Analysis and Assessment in Civil Engineering: Towards an Integrated Vision*, pp. 219–227, 2018.
- [35] V. Lertpiriyasuwat, M. C. Berg, and K. W. Buffinton, “Extended kalman filtering applied to a two-axis robotic arm with flexible links,” *The International Journal of Robotics Research*, vol. 19, no. 3, pp. 254–270, 2000.
- [36] Y. Xu and E. Ritz, “Vision based flexible beam tip point control,” in *2009 American Control Conference*, 2009, pp. 5277–5282.
- [37] K. Masuya and K. Ayusawa, “A review of state estimation of humanoid robot targeting the center of mass, base kinematics, and external wrench,” *Advanced Robotics*, vol. 34, no. 21-22, pp. 1380–1389, 2020.
- [38] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, “Online lidar-slam for legged robots with robust registration and deep-learned loop closure,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4158–4164.
- [39] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Engelsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-L. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt, “Summary of team ihmc’s virtual robotics challenge entry,” in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 307–314.
- [40] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, “Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion,” in *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 483–490.
- [41] Q. Zhu, Y. Mao, R. Xiong, and J. Wu, “Adaptive torque and position control for a legged robot based on a series elastic actuator,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, p. 26, 2016.
- [42] Z. Sun, J. Qiu, J. Zhu, and S. Li, “A composite position control of flexible lower limb exoskeleton based on second-order sliding mode,” *Nonlinear Dynamics*, pp. 1–10, 2022.
- [43] M. Vigne, A. El Khoury, F. Di Meglio, and N. Petit, “State estimation for a legged robot with multiple flexibilities using imus: A kinematic approach,” *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 195–202, 2020.
- [44] M. Ghani, N. Assadian, and R. Varatharajoo, “Attitude and deformation coupled estimation of flexible satellite using low-cost sensors,” *Advances in Space Research*, vol. 69, no. 1, pp. 677–689, 2022.



- [45] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation,” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, 2001, pp. 239–246 vol.1.
- [46] T. Sugihara, Y. Nakamura, and H. Inoue, “Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1404–1409.
- [47] M. Shibuya, T. Suzuki, and K. Ohnishi, “Trajectory planning of biped robot using linear pendulum mode for double support phase,” in *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*. IEEE, 2006, pp. 4094–4099.
- [48] G. Cavagna, N. Heglund, and C. Taylor, “Walking, running and galloping: mechanical similarities between different animals,” *Scale effects in animal locomotion*, 1976.
- [49] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [50] Y. Zhao, B. R. Fernandez, and L. Sentis, “Robust optimal planning and control of non-periodic bipedal locomotion with a centroidal momentum model,” *International Journal of Robotics Research*, vol. 36, no. 11, pp. 1211–1243, 2017.
- [51] A. Bratta, R. Orsolino, M. Focchi, V. Barasuol, G. G. Muscolo, and C. Semini, “On the hardware feasibility of nonlinear trajectory optimization for legged locomotion based on a simplified dynamics,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020, Conference Proceedings, pp. 1417–1423.
- [52] B. Koopman, E. van Asseldonk, and H. van der Kooij, “Speed-dependent reference joint trajectory generation for robotic gait support,” *Journal of Biomechanics*, vol. 47, no. 6, pp. 1447–1458, 2014.
- [53] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning agile and dynamic motor skills for legged robots,” *Science Robotics*, vol. 4, no. 26, jan 2019.
- [54] S. Park, F. B. Horak, and A. D. Kuo, “Postural feedback responses scale with biomechanical constraints in human standing,” *Experimental brain research*, vol. 154, no. 4, pp. 417–427, 2004.
- [55] L. Sentis and O. Khatib, “Synthesis of whole-body behaviors through hierarchical control of behavioral primitives,” *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.
- [56] J. He and F. Gao, “Mechanism, actuation, perception, and control of highly dynamic multilegged robots: a review,” *Chinese Journal of Mechanical Engineering*, vol. 33, no. 1, pp. 1–30, 2020.
- [57] S. Pouya, M. Khodabakhsh, A. Spröwitz, and A. Ijspeert, “Spinal joint compliance and actuation in a simulated bounding quadruped robot,” *Autonomous Robots*, vol. 41, no. 2, pp. 437–452, 2017.
- [58] A. MacChietto, V. Zordan, and C. R. Shelton, “Momentum control for balance,” *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.

- [59] S. H. Lee and A. Goswami, “Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground,” in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, Conference Proceedings, pp. 3157–3162.
- [60] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-February, 2015, Conference Proceedings, pp. 295–302.
- [61] P. M. Wensing and D. E. Orin, “Improved computation of the humanoid centroidal dynamics and application for whole-body control,” *International Journal of Humanoid Robotics*, vol. 13, no. 01, 2016.
- [62] L. Righetti and A. Herzog, “Momentum-centered control of contact interactions,” in *Springer Tracts in Advanced Robotics*. Springer, 2017, vol. 117, pp. 339–359.
- [63] W. Du, M. Fnadi, and F. Benamar, “Rolling based locomotion on rough terrain for a wheeled quadruped using centroidal dynamics,” *Mechanism and Machine Theory*, vol. 153, 2020.
- [64] J. Ahn, S. J. Jorgensen, S. H. Bang, and L. Sentis, “Versatile locomotion planning and control for humanoid robots,” *Frontiers in Robotics and AI*, vol. 8, 2021.
- [65] A. A. Shabana, *Dynamics of Multibody Systems*, 4th ed. Cambridge University Press, 2013.
- [66] —, “Definition of the slopes and the finite element absolute nodal coordinate formulation,” *Multibody system dynamics*, vol. 1, no. 3, pp. 339–348, 1997.
- [67] L. L. Howell and A. Midha, “Parametric Deflection Approximations for End-Loaded, Large-Deflection Beams in Compliant Mechanisms,” *Journal of Mechanical Design*, vol. 117, no. 1, pp. 156–165, 03 1995.
- [68] T. M. Wasfy and A. K. Noor, “Computational strategies for flexible multibody systems ,” *Applied Mechanics Reviews*, vol. 56, no. 6, pp. 553–613, 11 2003.
- [69] H.-J. Su, “A Pseudorigid-Body 3R Model for Determining Large Deflection of Cantilever Beams Subject to Tip Loads,” *Journal of Mechanisms and Robotics*, vol. 1, no. 2, 01 2009, 021008.
- [70] V. K. Venkiteswaran and H.-J. Su, “A parameter optimization framework for determining the pseudo-rigid-body model of cantilever-beams,” *Precision Engineering*, vol. 40, pp. 46–54, 2015.
- [71] G. Chen, B. Xiong, and X. Huang, “Finding the optimal characteristic parameters for 3r pseudo-rigid-body model using an improved particle swarm optimizer,” *Precision Engineering*, vol. 35, no. 3, pp. 505–511, 2011.
- [72] S. Šalinić and A. Nikolić, “A new pseudo-rigid-body model approach for modeling the quasi-static response of planar flexure-hinge mechanisms,” *Mechanism and Machine Theory*, vol. 124, pp. 150–161, 2018.
- [73] Y.-Q. Yu, Z.-L. Feng, and Q.-P. Xu, “A pseudo-rigid-body 2r model of flexural beam in compliant mechanisms,” *Mechanism and Machine Theory*, vol. 55, pp. 18–33, 2012.
- [74] P. Bilancia, G. Berselli, L. Bruzzone, and P. Fanghella, “A cad/cae integration framework for analyzing and designing spatial compliant mechanisms via pseudo-rigid-body methods,” *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 287–302, 2019.

- [75] M. Berzeri and A. Shabana, “Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation,” *Journal of Sound and Vibration*, vol. 235, no. 4, pp. 539–565, 2000.
- [76] J. Gerstmayr, H. Sugiyama, and A. Mikkola, “Review on the Absolute Nodal Coordinate Formulation for Large Deformation Analysis of Multibody Systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 8, no. 3, 03 2013, 031016.
- [77] K. Otsuka, K. Makihara, and H. Sugiyama, “Recent Advances in the Absolute Nodal Coordinate Formulation: Literature Review From 2012 to 2020,” *Journal of Computational and Nonlinear Dynamics*, vol. 17, no. 8, 04 2022, 080803.
- [78] R. G. Ross, “Synthesis of stiffness and mass matrices from experimental vibration modes,” *SAE Transactions*, vol. 80, pp. 2627–2635, 1971.
- [79] M. Dibold, J. Gerstmayr, and H. Irschik, “A Detailed Comparison of the Absolute Nodal Coordinate and the Floating Frame of Reference Formulation in Deformable Multibody Systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 4, no. 2, 03 2009, 021006.
- [80] A. A. Nada, B. A. Hussein, S. M. Megahed, and A. A. Shabana, “Use of the floating frame of reference formulation in large deformation analysis: Experimental and numerical validation,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 224, no. 1, pp. 45–58, 2010.
- [81] M. Ellenbroek and J. Schilder, “On the use of absolute interface coordinates in the floating frame of reference formulation for flexible multibody dynamics,” *Multibody system dynamics*, vol. 43, no. 3, pp. 193–208, 2018.
- [82] J. Niiranen, “Fast and accurate symmetric euler algorithm for electromechanical simulations,” in *Electrimacs 99 (modelling and simulation of electric machines converters and systems)*, 1999, pp. I–71.
- [83] J. Semblat, “Rheological interpretation of rayleigh damping,” *Journal of Sound and Vibration*, vol. 206, no. 5, pp. 741–744, oct 1997.
- [84] J. Hwangbo, J. Lee, and M. Hutter, “Per-contact iteration method for solving contact dynamics,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

# Appendix A

## Extending the Floating Frame Beam Element

### A.1 Add damping

There are two reasons to add damping to a flexible model: it makes the system more numerically stable, and it makes the system more realistic. An integration scheme can induce energy into the system, which can cause the system to explode if there are no dissipative elements such as damping. Furthermore, damping exist in most real materials, especially plastics, as nothing keeps oscillating forever.

The damping changes the floating frame equations of motions to

$$\begin{bmatrix} \mathbf{M}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{c}} \\ \ddot{\varphi} \\ \ddot{\mathbf{q}}^f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_f \end{bmatrix} \begin{bmatrix} \dot{\mathbf{c}} \\ \dot{\varphi} \\ \dot{\mathbf{q}}^f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \varphi \\ \mathbf{q}^f \end{bmatrix} = \mathbf{Q}_{fr} \quad (\text{A.1})$$

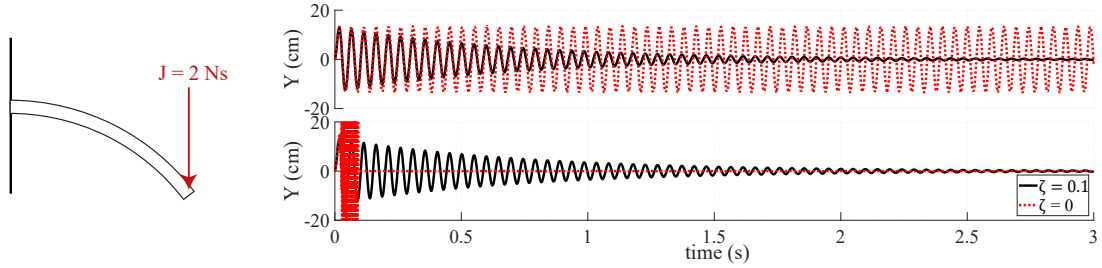
where  $\mathbf{D}_f$  is the focus of this appendix. In this thesis, the damping matrix follows from the damping ratio,  $\zeta$ . This can only be done when using the free-free modes, such that both  $\mathbf{M}_f$  and  $\mathbf{K}_f$  are diagonal matrices. Each flexible coordinate is uncoupled and a separate second-order differential equation

$$m_f \ddot{q}^f + d_f \dot{q}^f + k_f q^f \rightarrow \ddot{q}^f + 2\zeta \omega_n \dot{q}^f + \omega_n^2 q^f \quad (\text{A.2})$$

where  $m_f$ ,  $d_f$ , and  $k_f$  are the element values of the matrices,  $\omega_n$  is the natural frequency and equals  $\sqrt{\frac{k_f}{m_f}}$ . The damping ratio,  $\zeta$ , is a design parameter set to 0.1 to result in underdamped system the required damping value then equals  $d_f = 2\zeta m_f \omega_n = 2\zeta m_f \sqrt{\frac{k_f}{m_f}} = 2\zeta \sqrt{k_f m_f}$ .

Fig. A.1 shows the effect of the added damping on an input impulse. After around 3 seconds, the oscillation damps out when using the symplectic Euler scheme. The second plot shows how damping makes the system more stable by integrating using a non-energy preserving algorithm.

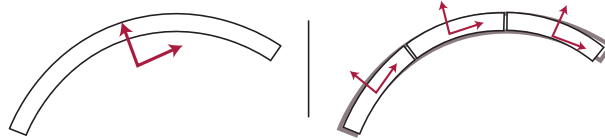
When different mode shapes are used, this damping is no longer applicable. The same counts when one beam is split to use a non-linear model. In these cases, Rayleigh damping could be used where  $\mathbf{D} = \alpha \mathbf{M} + \beta \mathbf{K}$ , where  $\alpha$  and  $\beta$  are design parameters. Even though, this damping has little physical meaning, it is a convenient way to represent damping in complex models[83]. Note, that these are the full mass and stiffness matrices, so this damping will also affect the rigid body motions. For that reason,  $\alpha$  is often set to 0.



**Figure A.1:** The tip deflection after an impulse of 200 N for 0.01 s resulting in an impulse of 2 Ns as shown on the left. The right shows the response. The top figure is integrated using the symplectic Euler method, while the beam in the bottom figure is integrated using forward euler.

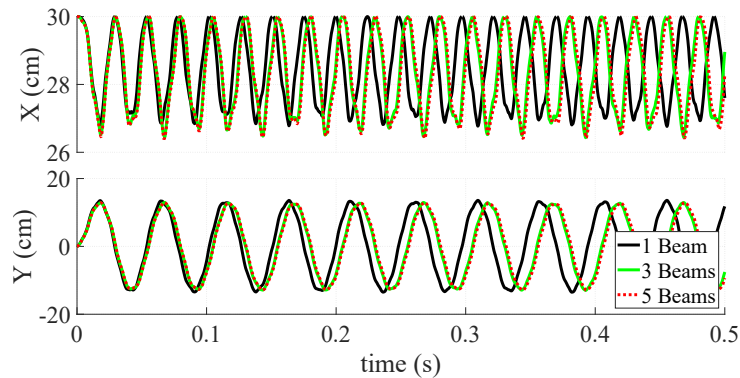
## A.2 Split bodies

One limitation of the floating frame formulation is its limitation to systems with a linear flexibility. *Nada et al.* propose to split the beam such that one beam has multiple floating frames [80]. This is shown schematically in Fig. A.2. This results in additional degrees of freedom and more constraints. Each additional body adds 3 rigid body coordinates and some flexible coordinates. Similar to a revolute joint, the position of each connection constrained, however, for these split beams also the angle of the two connected beams must be equal at the connection. So, each additional beam also results in three additional constraints.



**Figure A.2:** Schematic of changing from one single beam to 3 beams.

The response of this split system is compared to the response of the non-split beam in the same setup, as shown in Fig. A.1. Fig. A.3 shows the response is similar, however adding more beams results in a lower eigenfrequency due to the complexer motion it can make. Furthermore, a split beams results in a higher amplitude in the x-direction due to it better estimating the elasticity in that direction.



**Figure A.3:** Deflection of a beam due to an 2Ns impulse when the beam is split.

## Appendix B

# Ground Contact Model in the Floating Frame Formulation

The contact model defines the forces of the floor acting on the robot. *Hwangbo et al.* model the ground forces in the case of an inelastic contact[84]. This model considers multiple contacts and slipping using a model in the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}_\rho^T \boldsymbol{\rho} \quad (\text{B.1})$$

where  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$  are all the non-linear forces. The model used by the FFRF is

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} - ([\mathcal{C}]_q)^T \boldsymbol{\lambda} = \mathbf{S}_\tau^T \boldsymbol{\tau} + \mathbf{J}_{ext}^T \mathbf{W}_{ext} + \mathbf{J}_\rho^T \boldsymbol{\rho} + \mathbf{Q}_g \quad (\text{B.2})$$

which for this section is rewritten as

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}_\rho^T \boldsymbol{\rho} \quad (\text{B.3})$$

where  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$  are all the nonlinear forces

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{K}\mathbf{q} + \mathbf{S}_\tau^T \boldsymbol{\tau} + \mathbf{J}_{ext}^T \mathbf{W}_{ext} + \mathbf{Q}_g + ([\mathcal{C}]_q)^T \boldsymbol{\lambda}. \quad (\text{B.4})$$

This section derives the ground reaction forces when there is a non-slipping contact. The extension to multiple contact and slipping contact does not change, and is described by *Hwangbo et al.*.

In an inelastic collision, the velocity of each of the feet should equal zero after the collision. The first step is to transform Eq. (B.3) to the acceleration of foot  $i$  using  $\ddot{\mathbf{p}}_i = \mathbf{J}_{\rho,i} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\rho,i} \dot{\mathbf{q}}$ :

$$\begin{aligned} \ddot{\mathbf{q}} &= \mathbf{M}^{-1} ([\mathcal{C}]_q)^T \boldsymbol{\lambda} + \mathbf{M}^{-1} \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}^{-1} \mathbf{J}_\rho^T \boldsymbol{\rho}, \\ \mathbf{J}_{\rho,i} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\rho,i} \dot{\mathbf{q}} &= \mathbf{J}_{\rho,i} \mathbf{M}^{-1} ([\mathcal{C}]_q)^T \boldsymbol{\lambda} + \mathbf{J}_{\rho,i} \mathbf{M}^{-1} \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}_{\rho,i} \mathbf{M}^{-1} \mathbf{J}_\rho^T \boldsymbol{\rho} + \dot{\mathbf{J}}_{\rho,i} \dot{\mathbf{q}}, \\ \ddot{\mathbf{p}}_i &= \mathbf{J}_{\rho,i} \mathbf{M}^{-1} ([\mathcal{C}]_q)^T \boldsymbol{\lambda} + \mathbf{J}_{\rho,i} \mathbf{M}^{-1} \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}_{\rho,i} \mathbf{M}^{-1} \mathbf{J}_\rho^T \boldsymbol{\rho} + \dot{\mathbf{J}}_{\rho,i} \dot{\mathbf{q}}. \end{aligned}$$

Next using Euler integration  $\dot{\mathbf{p}}_i^+ = \Delta t \ddot{\mathbf{p}}_i + \dot{\mathbf{p}}_i = \Delta t \ddot{\mathbf{p}}_i + \mathbf{J}_{\rho,i} \dot{\mathbf{q}}_i$ , where  $\dot{\mathbf{p}}_i^+$  is the velocity of foot  $i$  at the next time step,  $\Delta t$ . In an inelastic contact, this velocity should be zero. This results in

$$|\dot{\mathbf{p}}_i^+ = \Delta t \mathbf{J}_{\rho,i} \mathbf{M}^{-1} \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta t \mathbf{J}_{\rho,i} \mathbf{M}^{-1} \mathbf{J}_\rho^T \boldsymbol{\rho} + \Delta t \dot{\mathbf{J}}_{\rho,i} \dot{\mathbf{q}} + \mathbf{J}_{\rho,i} \dot{\mathbf{q}}_i = 0, \quad (\text{B.5})$$

which we have to solve for the ground reaction forces,  $\boldsymbol{\rho}$  at each foot. Using the fact that

$$\mathbf{J}_\rho^T \boldsymbol{\rho} = \sum_{k=0}^{Nf} \mathbf{J}_{\rho,k}^T \boldsymbol{\rho}_k = \mathbf{J}_{\rho,i}^T \boldsymbol{\rho}_i + \sum_{k=0, k \neq i}^{Nf} \mathbf{J}_{\rho,k}^T \boldsymbol{\rho}_k \quad (\text{B.6})$$

where  $Nf$  is the number of feet. This makes it possible to rewrite Eq. (B.5) as

$$\mathbf{J}_{\rho,i}\mathbf{M}^{-1}\mathbf{J}_{\rho,i}^T\Delta t\boldsymbol{\rho}_i = \Delta t\mathbf{J}_{\rho,i}\mathbf{M}^{-1}\mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) + \Delta t\mathbf{J}_{\rho,i}\mathbf{M}^{-1}\sum_{k=0, k\neq i}^{Nf}\mathbf{J}_{\rho,k}^T\boldsymbol{\rho}_k + \Delta t\dot{\mathbf{J}}_{\rho,i}\dot{\mathbf{q}} + \mathbf{J}_{\rho,i}\dot{\mathbf{q}}. \quad (\text{B.7})$$

This can be solved using iteration for each of the ground reaction forces. *Hwangbo et al.* defines  $\Delta t\boldsymbol{\rho}_i$  as the ground reaction impulse  $\boldsymbol{\lambda}_i$ , not to be confused with the Lagrange multiplier in Eq. (B.3).

Similar to how the contact forces depend on each other, also the ground reaction forces and Lagrange multiplier influence each other. Therefore, one cannot be solved without the other. The solution is to also compute the Lagrange multiplier iteratively, as shown in Algorithm 1.

---

**Algorithm 1:** Iterative contact and Equations of motion solving

---

**Data:** *nonlinear forces:*  $\mathbf{K}\mathbf{q}$ ,  $\mathbf{S}_\tau^T\boldsymbol{\tau}$ ,  $\mathbf{J}_{ext}^T\mathbf{W}_{ext}$ ,  $\mathbf{Q}_g$ , and *Data from previous time step:*

$\mathbf{q}^-$ ,  $\dot{\mathbf{q}}^-$ ,  $\boldsymbol{\lambda}^-$

**Result:** *States for next time step:*  $\mathbf{q}^+$ ,  $\dot{\mathbf{q}}^+$ ,  $\boldsymbol{\lambda}^+$

/\* Start of Algorithm

\*/

$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}^-$

**while not converged do**

$\mathbf{h}(\mathbf{q}^-, \dot{\mathbf{q}}^-) = -\mathbf{K}\mathbf{q}^- + \mathbf{S}_\tau^T\boldsymbol{\tau} + \mathbf{J}_{ext}^T\mathbf{W}_{ext} + \mathbf{Q}_g + ([\mathcal{C}]_q)^T\boldsymbol{\lambda}$

$\boldsymbol{\rho} \leftarrow \text{Contact Solver}(\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}), \dot{\mathbf{q}}^-)$

$$\begin{bmatrix} \ddot{\mathbf{q}}^+ \\ \boldsymbol{\lambda}^+ \end{bmatrix} = \begin{bmatrix} \mathbf{M} & ([\mathcal{C}]_{q^-})^T \\ [\mathcal{C}]_{q^-} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}_\rho^T\boldsymbol{\rho} \\ -[\mathcal{C}]_{q^-}\dot{\mathbf{q}}^- \end{bmatrix}$$

$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}^+$

**end**

$\dot{\mathbf{q}}^+ \leftarrow \dot{\mathbf{q}}^- + \Delta t\ddot{\mathbf{q}}^+$

$\mathbf{q}^+ \leftarrow \mathbf{q}^- + \Delta t\dot{\mathbf{q}}^+$

/\* End of Algorithm

\*/

---