Interner Bericht

DLR-IB-RM-OP-2023-151

Simulation and Control of Running Models

Masterarbeit

Simone Stivala

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Robotik und Mechatronik

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Technical University of Munich

School of Engineering and Design

Simulation and Control of Running Models

Scientific work to obtain the degree M.Sc. in Mechatronics and Robotics at the TUM School of Engineering and Design

Topic author: Prof. Dr.-Ing. Alin Olimpiu Albu-Schäffer
TUM, Chair for Sensor Based Robotic Systems and Intelligent Assistance Systems

Supervisors: Dr. rer. nat. Alessandro Massimo Giordano
TUM, Chair for Sensor Based Robotic Systems and Intelligent Assistance Systems

Prof. Dr. Andrea Del Prete
University of Trento, Department of Industrial Engineering

Submitted by: Simone Stivala

Submitted on: 1 April 2023 in Garching bei München

Declaration of authenticity

I hereby state that I have written the thesis submitted by me independently and have not used any sources or aids other than those indicated.

Garching bei München, 01.04.2023, Simone Stirola

Place, date, signature

Acknowledgements

I would like to thank Dr. Giordano for his continuous and valuable support throughout my time spent at the German Aerospace Center (DLR).

Abstract

This work focuses on the locomotion of one-legged robots, with focus on approaches that stabilize passive limit cycles. Locomotion based on the so-called passive gaits promises to greatly reduce the actuation effort required for legged robots to move. In this work, the passive gaits of robots of varying complexity are characterized and stabilizing controllers are reviewed from the literature and newly formulated. The robots are modelled as hybrid dynamical systems and numerically simulated, thereby allowing to validate the proposed control strategies.

Firstly, the vertical control through energy regulation of a one-dimensional hopper is considered.

Secondly, the SLIP model is reviewed and then extended to the "pitching-SLIP", with the aim of characterizing its passive gaits with somersaults. Two controllers based on energy and angular momentum regulation are then formulated to stabilize passive gaits with somersaults, making the control effort converge to zero. A further extension of the SLIP template, denominated "body-SLIP", is then used to test the control approach on a more realistic model. The controllers shall be later extended to more complex cases, in which the somersaults are not necessarily present in the passive gaits.

Thirdly, the locomotion of a one-legged robot with a body link is studied. Raibert's control approach based on the foot placement algorithm is reviewed and compared to the non-dissipative touchdown controller of Hyon and Emura. The latter is then extended to be used with continuous torque profiles and to perform velocity tracking. Moreover, damping is added to the joints in order to study its effect on the controller, which was then modified to achieve stable running even in such conditions. The results obtained shall lay the foundations for a later test on hardware on DLR's quadruped *Bert*.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

In the design of legged robots, the introduction of compliant joints in the mechanical design plays a fundamental role in the realization of dynamic running gaits. In particular, the use of series elastic actuators allows to exploit the energy stored in the mechanical springs during running to perform key movements like the backward and forward swinging of legs in the forward horizontal motion or the compression and extension of the legs in the vertical jumping motion.

The motivation and main advantage of exploiting the passive dynamics of elastic elements to achieve under-actuated running gaits lie in the reduction of the actuation effort, which translates into a lower energy consumption required to keep the robot moving. Moreover, this locomotion approach can reduce the complexity of the underlying locomotion controllers, as the movements achieved by the robot are natural and determined by its passive dynamics, instead of being forced by the actuators trying to follow pre-computed trajectories.

The aim of this work is firstly to find and analyze the natural passive gaits of simple one-legged robot models with increasing complexity and secondly to formulate and test through numeric simulation different locomotion controllers, with the aim of achieving orbital stabilization of the passive running gaits. Controllers inspired by the state of the art as well as newly formulated controllers based on energy and momentum conservation have been studied. Moreover, in order to highlight the advantages and disadvantages of compliant robots, a comparison between stabilized elastic passive running and running with a traditional direct joint control approach has been made.

1.2 Literature review

Legged locomotion is a very active research field in modern robotics. Robots with legs have the primary advantage of being able to navigate in rough terrain. This is thanks to the fact that legs, unlike wheels, do not require a continuous surface of contact to provide the robot with movement and allow for stepping at different heights, for example while walking on stairs, or over obstacles. Moreover, legged robots are intrinsically closer to the animal world, which means that they can interact more naturally with environments designed for humans and animals. On the other side, the introduction of legs, with all the annexed mechanical joints, links and actuators, drastically increases the design complexity of legged robots when compared to wheeled or tracked robots. Furthermore, the task of walking or running requires the generation of complex joint trajectories which, together with the need of balancing to avoid falling during the movement, makes legged robots significantly more complex to control. [1]

Many different robot designs and control approaches have been developed in the past decades that achieve walking and running and that have been validated on hardware. Starting from simple quasi-static walking and evolving to complex dynamic walking and running, the performance of legged robots has increased through the years, even though in most cases the achieved energy efficiency remains lower with the respect to the levels achieved by the animal counterparts. Energy efficiency is indeed currently one of the main areas of interest of the research, and it is estimated that there is still great room for improvement. One of the main approaches to improve the mechanical energy efficiency of legged robotic locomotion is the introduction of elastic elements in the design and the exploitation of the system's natural motions or passive dynamics. [2] Such designs are inspired by the animal world, as animals have been shown to achieve efficient legged locomotion thanks to the use of spring-like body parts - mainly tendons - to store and retrieve energy during the movement [3].

The exploitation of passive dynamics, which cause the robotic system to move even when no input is given to the actuation motors, has been the object of study of several research works since the late 1980s, when fully passive walkers able to walk down on slopes without any actuation were designed and studied by Tad McGeer [4]. Subsequently, the passive running of a one-legged robot was studied in simulation by Thompson and Raibert in [5]. These works were followed by many attempts to apply the concept of passive locomotion to actual hardware robotic systems. A recent example of a robot achieving a very high efficiency by exploiting passive dynamics is the four-legged bipedal robot *Ranger* [6], which makes use of a hip spring and ankle springs to aid the swinging of the legs and of the feet respectively and was able to walk over 65 km on a single battery charge. At the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR), the quadruped *Bert* is being developed with focus on efficient locomotion based on the natural oscillation modes of the system. The quadruped robot has legs controlled by series elastic actuators, which include a mechanical spring between the motor and joint. These springs have the purpose of storing and releasing energy during the movement, with the aim of achieving highly dynamic gaits [7].

Many control algorithms for biped and quadruped locomotion are based on simple conceptual models, also known as templates. The study of controllers for such templates and the subsequent adaptation to more complex robot models has been proven to be a very effective approach and is the basis for the successful locomotion of many robots such as the *Scout II* quadruped robot [8], whose simplified model based on the spring-loaded inverted pendulum model (SLIP) template displays fully passive and stable running with bounding gaits. The SLIP plays an important role in the study of running of animals and robots [9], [10].

As the name suggests, the SLIP template introduces compliance in the model in the form of a linear spring. Compliance has been shown to play a fundamental role in human and animal running [3], [11] and it is a prerequisite to achieve passive locomotion gaits. The SLIP template has been object of study of several papers in the field of robotic locomotion and, while being considerably simpler than real-case robots, it possesses a series of properties that make it relevant for the study of one-legged or multi-legged robots. Examples are [8], where the SLIP's self-stabilization property is studied to understand the behaviour of the running with bounding gait of the quadruped robot *Scout II*, and [12], where the SLIP template dynamics are matched with those of the compliantly actuated legs of DLR's quadrupedal robot *Bert*.

One-legged robots are a particular case of study that can help with understanding the locomotion of multi-legged systems that run or hop. This type of robot moves forward by alternating between two states: stance, in which the foot is in contact with the ground, and flight, in which the system's center of mass moves on a ballistic trajectory. A controller not based on passive dynamics for one-legged robots was described by Raibert in [13] and validated on hardware. The robot model described in the work is equipped with a linear spring on the leg, but no polar spring is present on the hip, meaning that the leg must be swung through the effort of a hip actuator. The author describes a control approach that is divided into three decoupled parts: vertical control, body attitude control and forward velocity control. Despite its simplicity, this controller is able to achieve all the major tasks required by a hopping robot: initiate the hopping from a static position, advance with a desired forward speed, keep a desired body attitude and stop the hopping at a desired set-point in space. The flexibility of this controller distinguishes it from methods based on passive gaits stabilization, where changing the forward velocity requires switching between different passive gaits and where the body's attitude and hopping height during running is entirely determined by the system's natural motions and therefore uncontrolled. However this control approach requires a considerable energy consumption, which led Raibert and Thompson to investigate the use of a hip spring in order to save energy in [5]. In this work, they showed that a simple planar one-legged robot with a linear leg spring and a polar hip spring is capable of running without any actuation effort, even though the passive gaits were found to be unstable.

Different control approaches were proposed to stabilize the passive gaits of the planar one-legged robot described in [5]. One of them is described by Hyon and Emura in [14]. Three different adaptive controllers that achieve orbital stabilization of fully periodic passive gaits are described. This approach uses adaptation laws that change the hip stiffness, the leg stiffness or the robot's energy after each running step in order to minimize the control effort. Eventually the control effort converges to zero and a passive stable gait is reached. An advantage of this method is that the robot converges to passive gaits which are not known *a priori*, i.e. they do not need to be pre-computed.

A second approach to achieve controlled passive dynamic running of the monopod is described by Ahmadi and Buehler in [15], where the passive trajectory is pre-computed based on the desired forward velocity of the robot and a feedback controller is used to maintain the system as close as possible to the passive gait, with the goal of minimizing the energy consumption.

A third approach is presented in [16] by François and Samson. The method makes use of Poincaré maps linearized around the fixed points, which correspond to periodic gaits of the robot. This approach achieves orbital stability of the passive periodic gaits by applying a feedback controller that stabilizes the closed-loop linearized Poincaré map around the fixed points.

At the Institute of Robotics and Mechatronics at DLR, an experimental hardware setup with a single leg is used to test control algorithms for the later extension to the full quadruped robot *Bert*, as described in [17] and [18], where nonlinear normal modes are excited to achieve efficient forward hopping of the robot's leg. The excitation of normal modes, which are described for nonlinear systems such as robotic systems in [19], is an approach to exploit the system's passive dynamics.

Orbital stabilization is the primary objective of controllers that achieve locomotion based on passive gaits. The main idea behind is to make sure that the controlled robot converges to a passive gait even when starting from initial conditions that are not on the nominal gait. If the system converges to it, it is possible to remain on it with theoretically zero actuation effort, in the sense that the controller only needs to act in the event of small perturbations, for example due to numerical integration errors, to bring back the system to its achieved limit cycle.

Passivity-based approaches distinguish themselves from direct joint actuation methods, where joints are directly actuated with no elastic element between actuators and links. These types of architectures generally offer more flexibility in terms of types of terrain they can traverse and tasks they can perform, at the cost of an increased energy cost. Example of robots controlled with this approach are Honda's *ASIMO* and Boston Dynamics' *BigDog*, which both demonstrate very high versatility paired with a high energy consumption. Legged robots based on passive dynamics instead achieve great efficiency but are up to now limited to relatively flat terrains and low slopes. [6]

1.3 Methods

In this work, planar one-legged robots are modelled as nonlinear hybrid systems and the ground is considered infinitely rigid. Return maps based on Poincaré sections are used to find and study limit cycles of the robots corresponding to passive gaits of the models. Newly formulated controllers and controllers taken from the literature are used to achieve stable running gaits. Standard methods from nonlinear control theory are used to asses the stability of the locomotion in open-loop and closed-loop configurations. For the validation of the control approaches as well as the analysis of their performance, numerical simulations were implemented and run using *Simulink*.

1.4 Content overview

Section 2 briefly reviews the basic definitions of hybrid dynamical systems, Poincaré sections, return maps and limit cycle stability analysis. In section 3, the simple one-dimensional case of a hopper is studied. Section 4 reviews the SLIP model and introduces two new model, called "pitching-SLIP" and "body-SLIP", useful for the study of the interesting passive running gaits with somersaults. The passive gaits of the models are analyzed and then stabilized through two controllers based on energy and angular momentum regulation. In section 5, the case of a planar one-legged robot with body is considered. The non-dissipative touchdown control for the passive running of a monopod described by Hyon and Emura in [14] and the foot placement algorithm by Raibert [13] are reviewed and extended. Simulation results of the two approaches are compared and discussed. Finally, the effectiveness of the non-dissipative touchdown control is studied in the non-ideal case of viscous friction in the joints and the controller is extended to more realistic continuous torque actuation profiles and to perform velocity tracking. Section 6 concludes the work with a brief summary and outlook.

2 Modelling and analysis of legged robots

In this section, some theory concepts relevant to the work are reviewed. Firstly, a basic definition for hybrid dynamical systems is reported. Secondly, the definitions of Poincaré sections and return maps are revised. Then, the algorithm to search for limit cycles of nonlinear dynamical systems based on fixed points of the Poincaré return maps is described. Finally, a method to infer the stability of limit cycles based on the linearization of return maps around fixed points is discussed.

2.1 Legged robots as hybrid systems

One important decision when modelling legged robots is related to how the contact points between feet and ground are modelled. In particular, one can distinguish between compliant contact models and rigid contact models. While compliant models realistically take into account the deformation of the bodies involved in the contact, the high stiffness values involved can cause the differential equations to be numerical stiff, which can lead to difficulties in the numerical integration and analysis. Rigid contact models can simplify the theoretical and numerical analysis. [1]

Rigid contact models, which are used in this work, cause the dynamics of the system to switch when the contact state of the robot changes. Moreover, impacts occur whenever the robot touches the ground, as the contact points are not allowed to penetrate the ground. Such switching dynamics can be modelled and analysed as hybrid dynamical systems.

Hereby, a general definition for hybrid dynamical systems is reviewed. The definition and nomenclature here reported are taken from [20].

In hybrid dynamical systems, the state $\boldsymbol{x} \in \mathbb{R}^n$ can flow as in a continuous time system or jump as in a discrete time system depending on the current state. Such a system can be described as follows:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \quad \text{if } \boldsymbol{x} \in C \\ \boldsymbol{x}^+ = \boldsymbol{g}(\boldsymbol{x}) \quad \text{if } \boldsymbol{x} \in D \end{cases} \tag{1}$$

where $\boldsymbol{f}$ is denoted as *flow map* and is defined on the *flow set* C, and $\boldsymbol{g}$ is called *jump map* and is defined on the *jump set* D. When $\boldsymbol{x} \in C$ holds, the state flows according to a set of differential equations, whereas when $\boldsymbol{x} \in D$ holds, the state jumps according to a set of difference equations.

In the modelling of legged robots with rigid contact forces, the flow map represents the differential equations that govern the system during the flight and stance phases, whereas the jump map models the discontinuities in the states on the events of impacts with the ground or when a foot lifts off the ground. The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ can in general be defined differently on different subsets of C and D respectively. For example, in the case of one-legged robots later examined in this work, $\boldsymbol{f}$ describes the flight dynamics when the foot is not in contact with the ground, while it describes the stance dynamics when the contact with the ground is active. Another observation is that while in the general definition of hybrid systems multiple jumps can occur in sequence, in the models presented in this work the jumps only occur from D to C, meaning that after a jump happens the system will flow. Depending on the robot model, certain conditions on the state must hold so that several jumps do not occur in sequence. For example, for the one-legged robot described in [14] and used in this work in section 5, a condition can be derived on the states at touchdown so that the system does not switch back to flight immediately after hitting the ground. Such conditions ensure what is called normal phase switching, which is often necessary for the correct simulation of hybrid systems and allow to avoid "chattering" phenomena in which the system jumps repeatedly in short intervals of time.

2.2 Poincaré maps

2.2.1 Definitions

In this section, the concepts of Poincaré maps and sections are reviewed. Poincaré maps are a useful tool for the search and characterization of limit cycles of nonlinear hybrid systems. Their use to study the stability of limit cycles for hybrid systems is described in [21], whereas an example applied to the stability of passive gaits in robotics is in [8].

Given a nonlinear hybrid system evolving with state $\boldsymbol{x}$, one can define an hyperplane Σ, hereby referred to as Poincaré section, transversal to the state flow. A Poincaré map $\boldsymbol{P}(\boldsymbol{x})$ is a function that, given the states $\boldsymbol{x}$ of the system representing a point on the Poincaré section Σ, returns the states of the system at the next instance in which the flow crosses Σ again. The concept is illustrated in Figure 1. It is important to note that for the correct definition of $\boldsymbol{P}(\boldsymbol{x})$, the section Σ must be transversal to the flow, so that the system can not evolve while remaining on Σ. [21]

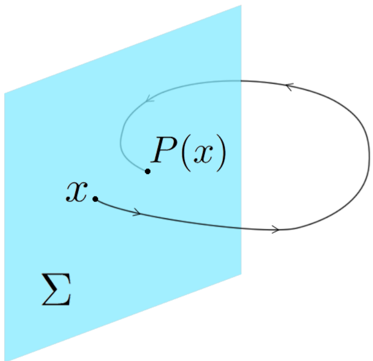


Figure 1: Illustration of the flow of the system intersecting a Poincaré section Σ at point $\boldsymbol{x}$ and then at point $\boldsymbol{P}(\boldsymbol{x})$.

Poincarè maps can be seen as discrete time functions that map the state of the system at the k-th crossing of the Poincarè section Σ with the states at the $(k+1)$-th crossing:

$$\boldsymbol{x}_{k+1} = \boldsymbol{P}(\boldsymbol{x}_k) \tag{2}$$

Fixed points of the Poincarè maps correspond to periodic solutions (limit cycles) of the system, in which the states always return at the same point on the Poincaré section Σ after a certain time period. The fixed points satisfy the condition

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \boldsymbol{P}(\boldsymbol{x}_k) - \boldsymbol{x}_k = 0 \tag{3}$$

Equation (3) offers an important tool to find limit cycles of nonlinear hybrid systems. Even though an analytical expression of the Poincarè map is usually not known, the map can be evaluated by numerically integrating the dynamics of the system from state $\boldsymbol{x}_k$ till Σ is crossed. By using an optimization algorithm to find the solutions of equation (3), the fixed points of the Poincarè map can be numerically found. In this work, the Levenberg–Marquardt algorithm (LMA) has been used for this purpose.

2.3 Stability analysis of limit cycles for hybrid systems

To study the stability of limit cycles, the stability of the corresponding Poincaré map fixed point can be studied [21], [1]. The stability of the the map can be evaluated locally by linearizing the map around a fixed point found by solving (3). The Jacobian matrix of the map can be evaluated numerically by using finite differences, since an analytical expression is usually not available. At the fixed point under study, the Poincarè map can be linearized with:

$$\Delta\boldsymbol{x}_{k+1} = J_{\boldsymbol{x}_0} \cdot \Delta\boldsymbol{x}_k = \left.\frac{\partial \boldsymbol{P}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_0} \cdot \Delta\boldsymbol{x}_k \tag{4}$$

where $\boldsymbol{x}_0$ denotes the fixed point, $J_{\boldsymbol{x}_0}$ the Jacobian matrix and $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_0$.

The eigenvalues of $J_{\boldsymbol{x}_0}$ are called characteristic (or Floquet) multipliers of the limit cycle associated with the fixed point and can be used to assess its stability. If the discrete system (4) is stable, which means that the Jacobian's eigenvalues are all inside the unit circle, then the limit cycle is also stable.

3 One-dimensional hopping robot

3.1 Model and equations of the dynamics

In this section, a simple one-dimensional problem of vertical hopping is considered. The robot model used is the one-legged hopper represented in Figure 2. The model is characterized by one prismatic joint along which a spring and a damping element act, as well as three point masses representing trunk, thigh and shank of a simple robot. The spring acts between the robot's center of mass and its foot, while the damper linearly acts against the joint velocity. Being constrained to move only along the vertical direction, the model possesses only 2 degrees of freedom which are the trunk's vertical position z_b and the prismatic joint coordinate q. An ideal force actuator can exert a force f on the prismatic joint, with positive sign aligned with the positive direction of q.

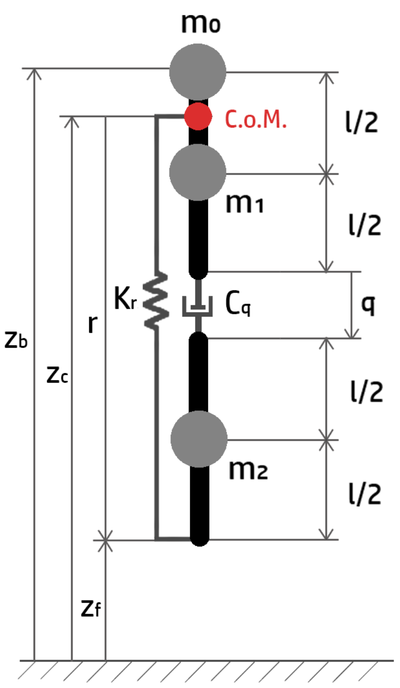


Figure 2: 1D hopper model with three point masses, a spring and a damper.

Table 1 clarifies the names of the variables and parameters of the model.

Name	Description
m_0	Trunk mass
m_1	Shank mass
m_2	Thigh mass
K_r	Radial spring stiffness
r_0	Radial spring resting length
C_q	Joint damping constant
q	Joint position
r	Distance between the center of mass and the foot
z_b	Base vertical position
z_c	Center of mass vertical position
z_f	Foot vertical position
f	Prismatic joint actuation force
l	Link length
g	Acceleration of gravity

Table 1: List of the parameters and variables of the 1D hopper with name and description.

Considering the ground as infinitely rigid, the model is hybrid and it alternates between the stance and the flight flow dynamics through the discrete events of lift-off and touchdown. The flight dynamics and the stance dynamics are described by the sets of differential equations (5) and (9) respectively.

In the flight phase, the system is governed by a set of two differential equations:

$$\begin{bmatrix} m_0+m_1+m_2 & m_2 \\ m_2 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{z}_b \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & C_q \end{bmatrix} \begin{bmatrix} \dot{z}_b \\ \dot{q} \end{bmatrix} + \begin{bmatrix} (m_0+m_1+m_2)g \\ m_2 g + J_{rq}^T K_r (r-r_0) \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}, \quad (5)$$

where the elastic spring force is realized by a spring that acts along the prismatic joint relatively to the coordinate $r = z_c - z_f$, which is the distance between the foot and the center of mass of the robot, and is at rest when $r = r_0$. The vertical coordinate of the center of mass z_c can be expressed in terms of the states z_b and q as

$$z_c = z_b - \frac{\frac{1}{2}l \cdot m_1 + (\frac{3}{2}l + q) \cdot m_2}{m_0+m_1+m_2}, \quad (6)$$

while the foot's coordinate is simply given by

$$z_f = z_b - q - 2l. \quad (7)$$

Finally, J_{rq} is expressed as

$$J_{rq} = \frac{m_0+m_1}{m_0+m_1+m_2}. \quad (8)$$

In the stance phase, the foot is constrained on the ground and the joint position q oscillates according to:

$$(m_0+m_1)\ddot{q} + C_q\dot{q} + (m_0+m_1)g + J_{rq}^T K_r (r-r_0) = f \quad (9)$$

Hereby, it is assumed that the joint damping is only present during the flight phase, so that the system is energy conservative during stance. Thus, C_q is set to zero in equation (9).

For the system to be in the flight phase, the foot's coordinate must be positive, $z_f > 0$. The transition from flight to stance happens when the foot touches the ground, $z_f = 0$ and $\dot{z}_f < 0$. During the stance phase, the foot remains fixed on the ground, $z_f = 0$. The transition from stance to flight occurs when the contact force between the foot and the ground becomes zero:

$$N_z = m_2 g - J_{rq}^T K_r (r - r_0) + f = 0 \tag{10}$$

Note that during stance the sign of the contact force N_z is always positive, as the ground constraint is unilateral, meaning that it can only push the robot upwards but can not pull it downwards.

On impact with the ground, discontinuities occur in certain state velocities of the system. In particular, the jump map is given by:

$$\begin{cases} \dot{q}^+ = \dot{z}_b^- \\ \dot{z}_b^+ = \dot{z}_b^- \end{cases} \tag{11}$$

where the superscript notations "-" and "+" refer to the time instants immediately before and after touchdown. The discontinuity only affects $\dot{q}$ and is caused by the instantaneous deceleration of the foot, whose velocity jumps to zero on impact.

The discontinuities on the velocity states also determine a loss of mechanical energy on impact which is given by the loss in kinetic energy:

$$\Delta E = E^+ - E^- = \frac{1}{2} m_2 (\dot{z}_f^-)^2 = \frac{1}{2} m_2 (\dot{q}^- - \dot{z}_b^-)^2 \tag{12}$$

Further energy losses present in the system when it is not actuated are caused by the damping element on the prismatic joint during flight.

In the next section, as simple controller based on energy regulation to achieve vertical control is described for the hopper.

3.2 Vertical control

To achieve hopping with jumps to the desired height, one possible approach is to formulate a controller that achieves energy regulation. The goal is to compensate the kinetic energy losses caused by the impacts with the ground and by the dissipating effect of the damper on the prismatic joint. A simple control law is given by 13 and is applied during the stance phases:

$$f = -\gamma_E \cdot (E - E_{des}) \cdot \dot{q} \tag{13}$$

where $\gamma_E > 0$ is a constant positive gain, E is the system's energy, E_{des} is the desired energy level and $\dot{q}$ is the joint velocity.

The energy derivative in the stance phase is given by (since $C_q = 0$):

$$\dot{E} = f\dot{q} \tag{14}$$

By substituting (13) in (14) the closed-loop dynamics of the energy are obtained:

$$\dot{E} + \gamma_E \dot{q}^2 (E - E_{des}) = 0 \tag{15}$$

Thus, the controller achieves nonlinear convergence of the energy to the desired value.

It is now assumed that the damping factor is high enough so that when the hopper reaches the apex of the jump the oscillation along the prismatic joint has been fully damped, i.e. at the apex $\dot{q} = 0$ holds. Then the amount of energy that the robot must have at the end of the stance phase in order to obtain a subsequent jump where the robot's center of mass is $z_c = z_{des}$ at the apex is given by:

$$E_{des} = (m + m_r^*) g z_{des} - m_r^* g (r_0 + \frac{m_{TO} g}{K_r}) + \frac{m_{TO}^2 g^2}{2K_r} \tag{16}$$

where:

$$m = m_0 + m_1 + m_2 \tag{17}$$

$$m_r^* = m_{TO} = \frac{m_2(m_0 + m_1 + m_2)}{m_0 + m_1} \tag{18}$$

An important consideration must be done with regards to the hybrid nature of the system and the application of the control law (13). Consider the case in which the robot is commanded to achieve a jump followed by a second jump with a higher apex. To achieve such a task, the robot must increase its energy level during the stance phase between the two jumps. Since the controller is enabled as soon as the ground is touched, this would cause the controller to pull the base down during the first half of the stance phase in order to increase the system's kinetic energy. In fact, during the first half of the stance phase the spring compresses with $\dot{q} < 0$, which, together with $E - E_{des} < 0$, causes the expression in (13) to be negative. If the magnitude of the pulling force applied is too large, it can cause a false landing as the foot is immediately pulled away from the ground at touchdown. This means that the model is not able to enter the stance phase from the flight phase. This can lead to a "chattering" phenomenon in which a succession of small jumps with false landings occur, which leads to a very fast switching between the flight and stance phases.

Given (10), the condition to avoid a false touchdown is simply given by:

$$N_z^+ \geq 0 \tag{19}$$

Since the controller described in (13) does not in general satisfy condition (19), two simple alternatives are described here.

The first option is to only allow the controller to push against the ground by introducing a saturation in the control law:

$$f = sat(-\gamma_E \cdot (E - E_{des}) \cdot \dot{q}) \tag{20}$$

where the saturation function is defined as:

$$sat(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

A second option is to allow negative values of f as long as the contact force N_z remains positive within an arbitrary margin that works as a threshold level.

$$f = \begin{cases} -\gamma_E \cdot (E - E_{des}) \cdot \dot{q} & \text{if } -\gamma_E \cdot (E - E_{des}) \cdot \dot{q} > -\beta \cdot (m_2 g - F_{el}) \\ -\beta \cdot (m_2 g - F_{el}) & \text{otherwise} \end{cases} \tag{22}$$

where $\beta \in [0, 1)$ is a parameter that determines how close the controller can bring the contact force N_z to zero by pulling the foot upwards and F_{el} is the elastic spring force. Note that the case $\beta = 0$ makes the controller act exactly as the one described in (20).

In the next section, the simulation results of the one-dimensional hopper are presented.

3.3 Simulation results

3.3.1 Model parameters

The parameters that were used for the simulation of the hopper are listed in Table 2. In the simulations, the damping was set to be active only in the flight phase.

Parameter	Value
m_0	0.957 kg
m_1	0 kg
m_2	0.05 kg
K_r	2000 N/m
C_q	15 Ns/m
r_0	0.23124 m
l	0.12 m
g	9.81 m/s^2

Table 2: List of the numerical values assigned for the simulation.

3.3.2 Uncontrolled damped hopping

Firstly, the simulation results of the case in which no controller is used to regulate the hopping are examined. In the simulation, the hopper was initialized in flight with $q = 0$ m, $\dot{q} = 0$ m/s, $z_b = 0.34$ m and $\dot{z}_b = 0$ m/s.

The state positions and velocities are reported in the plots in Figure 3. The hopper "bounces" on the ground with decreasing jumping height as energy is lost at each touchdown and is dissipated by the damper on the prismatic joint during flight. In this plot and all the next ones, the areas with gray background indicate the stance phases.

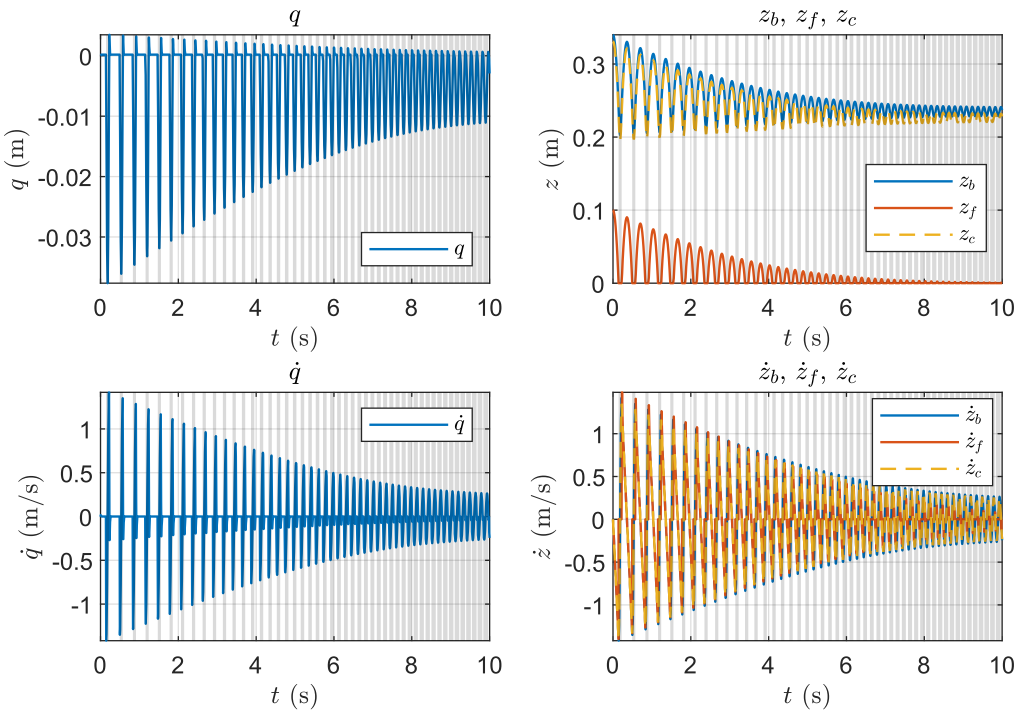


Figure 3: Joint position and vertical coordinates of base, foot and CoM. The oscillation amplitudes decrease as the robot loses energy and the stance phases become longer than the flight phases.

The plot in Figure 4 shows how the energy evolves: as expected, it starts from an initial value and then monotonically decreases due to the energy losses. The hopper bounces on the ground with decreasing jump heights until the energy decreases to a point after which no jumps are possible. The energy then remains constant and the hopper stays in stance with a residual oscillation left on the prismatic joint, since the damper is only enabled during flight.

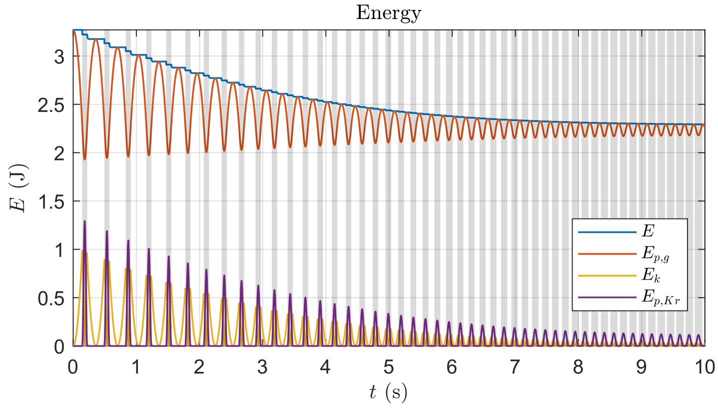


Figure 4: Time evolution of the energy of the system. E is the total energy, $E_{p,g}$ is the gravitational potential energy, E_k is the kinetic energy and E_{p,K_r} is the elastic potential energy.

In Figure 5, the intensities of the forces acting in the system are displayed.

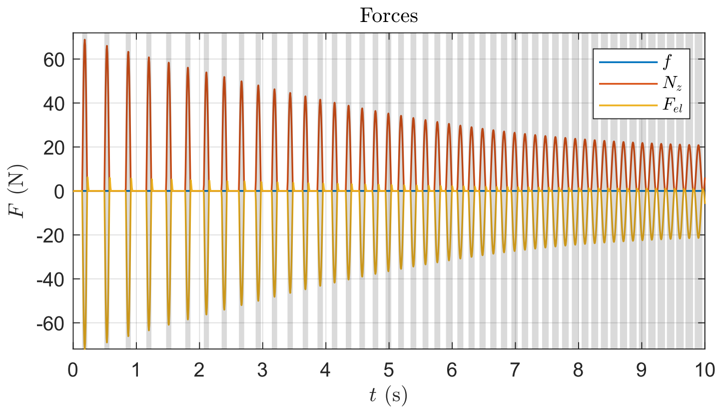


Figure 5: Forces acting in the system. In this simulation, the actuation force f is zero.

3.3.3 Controlled vertical hopping

In this section, the simulation results of the closed-loop system with the energy regulator are discussed. Figures 6 - 8 show the simulation results where the control law (20) was used with $\gamma_E = 100$ Ns/Jm. The stance phases are highlighted with a gray background in the plots.

The robot was initialized again with $q = 0$ m, $\dot{q} = 0$ m/s, $z_b = 0.34$ m and $\dot{z}_b = 0$ m/s. The desired jump height was set to three different subsequent levels in time. The robot performs two jumps with apex at $z_{des} = 0.35$ m, one jump with $z_{des} = 0.30$ m and then keeps jumping with $z_{des} = 0.40$ m.

The state positions and velocities are reported in the plots in Figure 6. After the initial transient in which the control goal is time-variant, the states evolve in a periodic way. The energy is reported in Figure 7 and the forces in Figure 8. The actuation force f is either used to inject energy in the system when the energy must be increased, as in the cases in which energy losses must be recovered or the jumping height must be increased, or to dissipate energy in the case in which the next desired jump is lower than the previous one and the current energy level is too high.

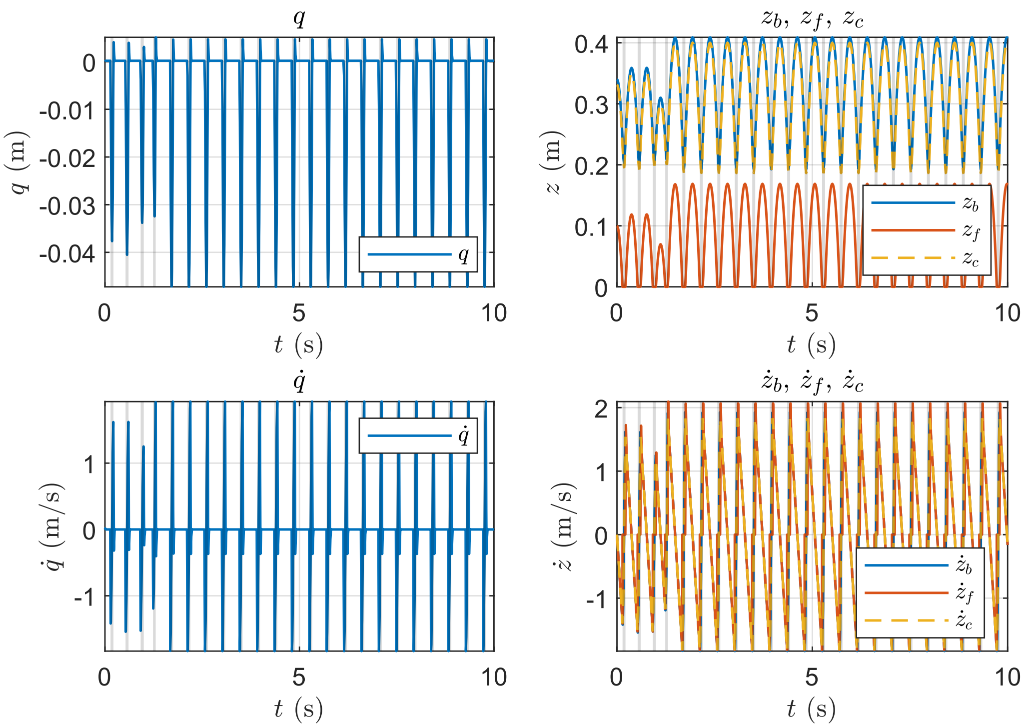


Figure 6: Joint position and vertical coordinates of base, foot and CoM.

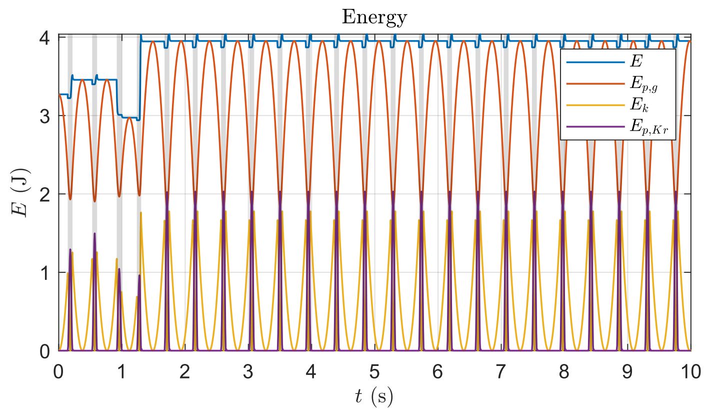


Figure 7: Time evolution of the energy of the system. E is the total energy, $E_{p.g}$ is the gravitational potential energy, E_k is the kinetic energy and E_{p,K_r} is the elastic potential energy.

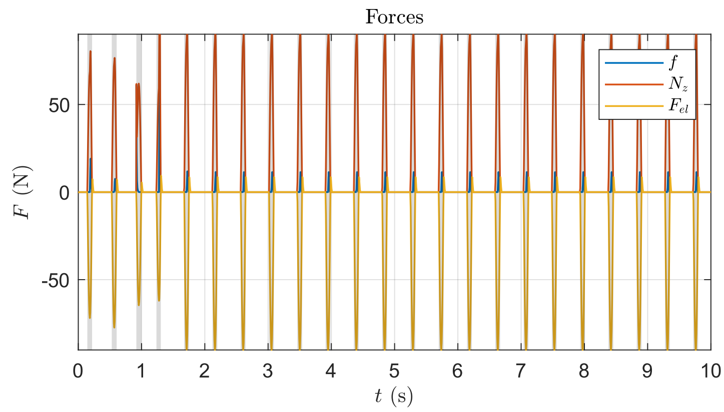


Figure 8: Forces acting in the system. The actuation force is always positive as the controller with saturation was used.

The vertical position of the center of mass together with its desired values at the apexes are shown in Figure 9. From the plot it is evident that the control goal is reached. Note that the gain γ_E can be set so that the corresponding control effort is enough to reach the desired energy levels within one stance phase. In general, with lower values of γ_E, the robot reaches the control targets by accumulating energy during more successive stance phases but requires lower peaks in the actuation forces.

Figure 10 shows again the energy, this time compared to the desired levels.

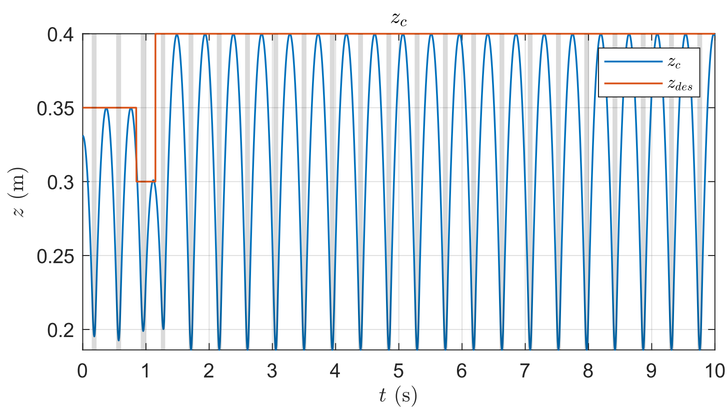


Figure 9: Time evolution of the vertical position of the center of mass together with its desired values at the apexes. In the third jump there is a small error with respect to the desired state as the controller did not manage to dissipate enough energy during the previous stance phase. This can be addressed by increasing the feedback gain γ_E.

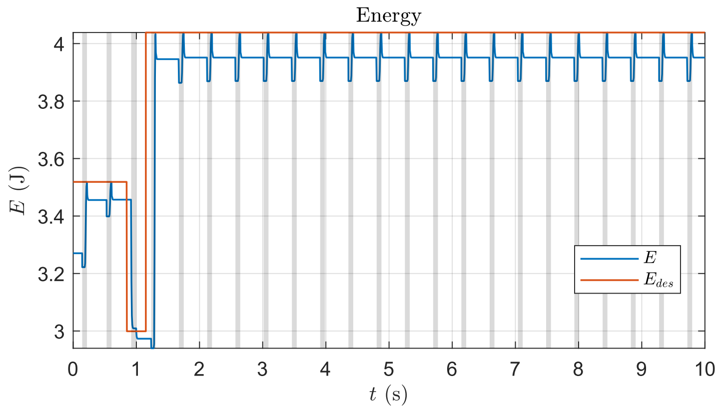


Figure 10: Time evolution of the energy compared to its desired values. The controller is active in the time-frames with shaded gray background, corresponding to the stance phases. The desired energy is changed during the flight phases with three steps.

In this section, an energy regulator to control the jumping motion of a one-dimensional robotic hopper was described and validated through numerical simulations. The controller was shown to effectively inject or dissipate energy in order to achieve the desired control goal. In the next sections, the same control law will be applied in the more complex problem of planar locomotion.

4 Planar one-legged robot without hip joint

In this chapter, the locomotion of simple planar one-legged robots is addressed. Models of increasing complexity, starting with the simple Spring Loaded Inverted Pendulum (SLIP) template and continuing by adding simple flight dynamics to the leg and subsequently an inertia to the leg, are examined.

4.1 Spring Loaded Inverted Pendulum (SLIP)

4.1.1 Model and dynamics equations

The well known Spring Loaded Inverted Pendulum (SLIP, see [9]) model is represented in Figure 11. The model comprises a single point mass to which a linear spring is attached. The spring has stiffness K_r and resting length r_0, it is massless and has no inertia. During flight, the SLIP has four degrees of freedom, which are the coordinates of the center of mass (x_c, z_c), the spring total extension r and the leg orientation angle α. During stance, the system can be described as a 2-DoF model in polar coordinates with respect to the massless extremity of the spring, hereby called foot, which is constrained to the ground.

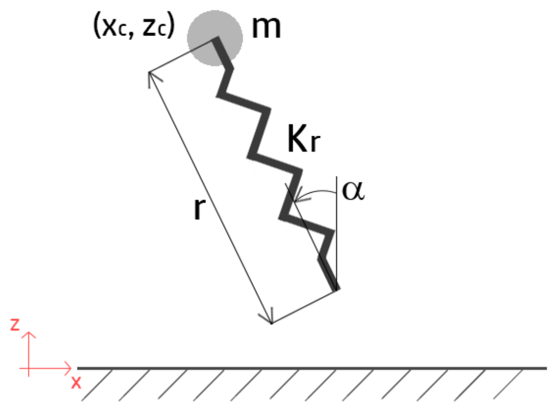


Figure 11: The SLIP model. The state variables and the model parameters are labeled in the diagram.

The equations of the model dynamics are reported in (23)-(26). During the flight phase the system evolves according to:

$$\begin{cases} \ddot{x}_c = 0 \\ \ddot{z}_c = -g \\ \alpha = \alpha_{TD,des} \\ r = r_0 \end{cases} \tag{23}$$

where g is the gravitational acceleration and $\alpha_{TD,des}$ is a constant angle arbitrarily chosen. Note that during flight the relations $\dot{\alpha} = 0$, and $\dot{r} = 0$ hold.

In the stance phase, the differential equations are:

$$\begin{cases} \ddot{\alpha} = -2\dfrac{\dot{r}}{r}\dot{\alpha} + \dfrac{g}{r}\sin(\alpha) \\ \ddot{r} = r\dot{\alpha}^2 - g\cos(\alpha) - \dfrac{K_r}{m}(r - r_0) \end{cases} \tag{24}$$

Finally, equations (25) and (26) report the lift-off and touchdown jump equations. The superscript notations "-" and "+" indicate the states respectively just before and just after the transitions.

At the lift-off event, which occurs when the spring decompresses and reaches again its resting position $r = r0$, there are discontinuities in the states of the leg, due to the zero-mass foot assumption and the arbitrarily set leg attack angle:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- \\ \dot{z}_c^+ = \dot{z}_c^- \\ \alpha^+ = \alpha_{TD,des} \\ \dot{\alpha}^+ = 0 \\ \dot{r}^+ = 0 \end{cases} \tag{25}$$

At touchdown, it holds:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- \\ \dot{z}_c^+ = \dot{z}_c^- \\ \dot{\alpha}^+ = -\dfrac{1}{r}\cos(\alpha^-)\dot{x}_c^+ - \dfrac{1}{r}\sin(\alpha^-)\dot{z}_c^+ \\ \dot{r}^+ = -\sin(\alpha^-)\dot{x}_c^+ + \cos(\alpha^-)\dot{z}_c^+ \end{cases} \tag{26}$$

It is important to note that the SLIP model is conservative, as no energy loss is present in the system neither during the flow dynamics nor at the transition jumps.

4.1.2 Simulation results

In this section, the results of a simulation of the SLIP model are reported. The values used for the model parameters are reported in Table 3.

Parameter	Value
m	1.007 kg
K_r	2000 N/m
r_0	0.23124 m
g	9.81 m/s^2

Table 3: List of the numerical values assigned for the simulation.

The model was initialized on the apex of a ballistic trajectory with $x_{c0} = 0$m, $\dot{x}_{c0} = 3$ m/s, $z_{c0} = 0.26$ m and $\dot{z}_{c0} = 0$ m/s and the leg touchdown angle was set as $\alpha_{TD,des} = 0.35$ rad. Figures 12-14 show the evolution of the model states and the ground reaction forces acting during the simulation. An important property of the SLIP highlighted by this simulation is the existence of stable limit cycles, corresponding to periodic gaits, to which the system converges when initialized with close enough initial conditions. This can be easily seen in the plots in the phase plane in Figure 13, where the states converge to closed periodic orbits. The ground reaction forces are shown in Figure 14.

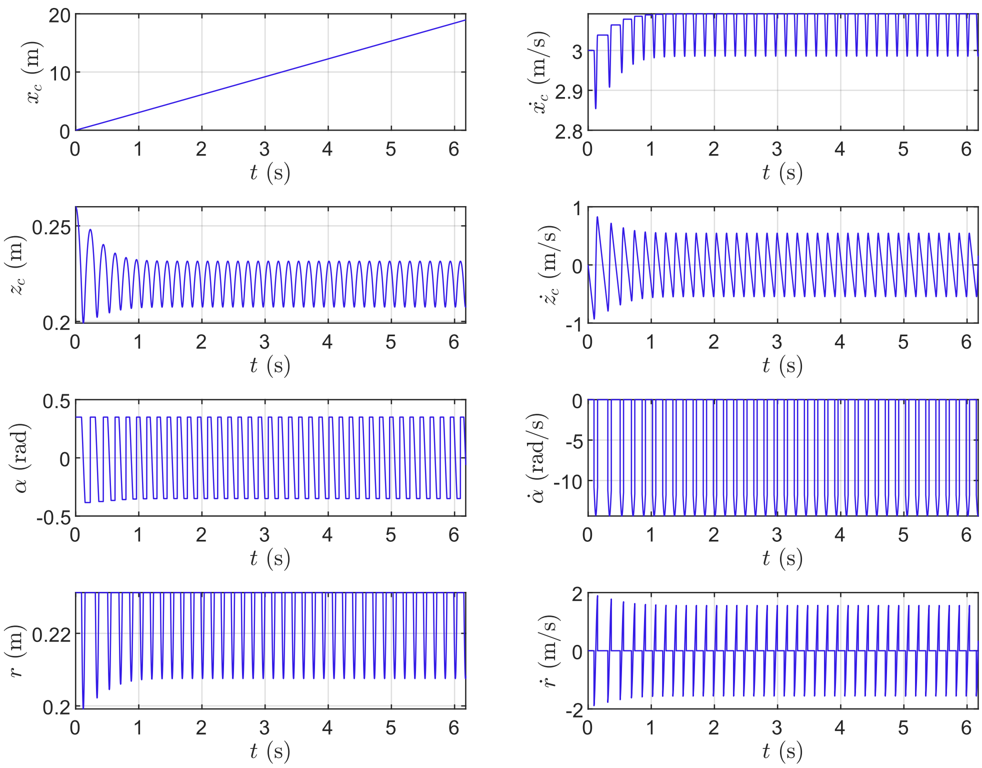


Figure 12: Plots of the state trajectories of the SLIP. The SLIP converges to a periodic gait.

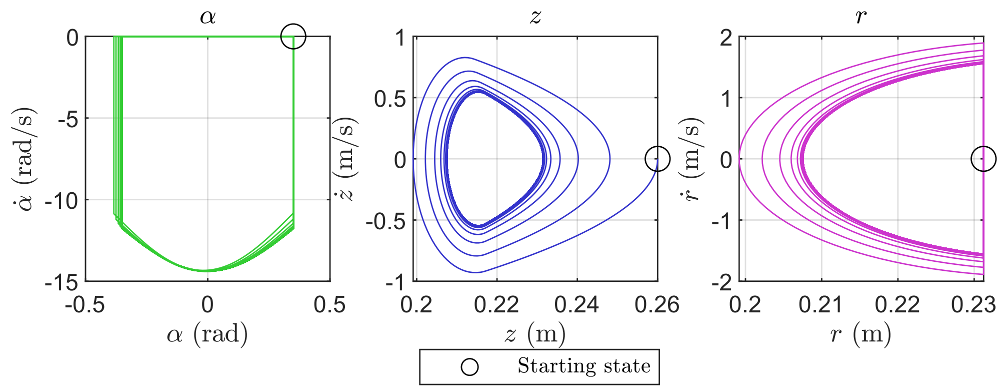


Figure 13: State trajectories of the SLIP template in the phase plane. The curves converge to closed periodic orbits, indicating that a limit cycle is reached.

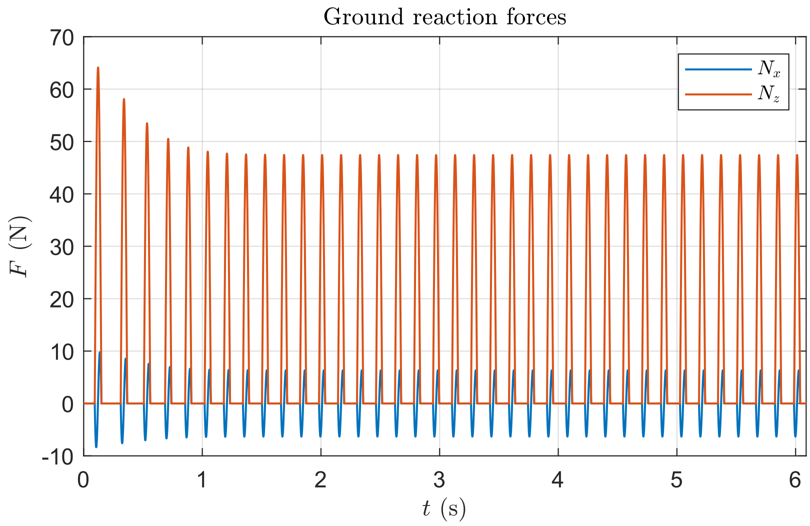


Figure 14: Ground reaction forces of the SLIP.

Figure 15 shows how the time duration of the flight and stance phases converge to fixed values as the model converges to the periodic gait. Figure 16 shows the trajectory of the SLIP after convergence to the periodic gait together with some snapshots of the model configuration. The stable periodic gaits of the SLIP are further examined in the next section.

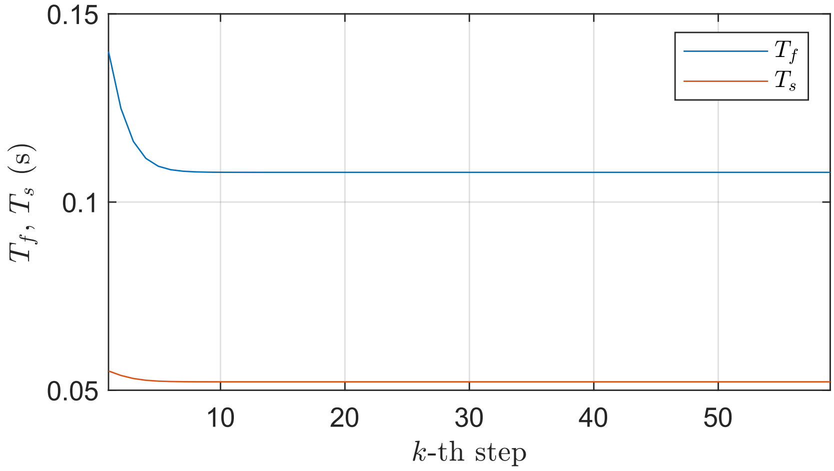


Figure 15: The flight time T_f and stance time T_s converge to constant values after a few steps.

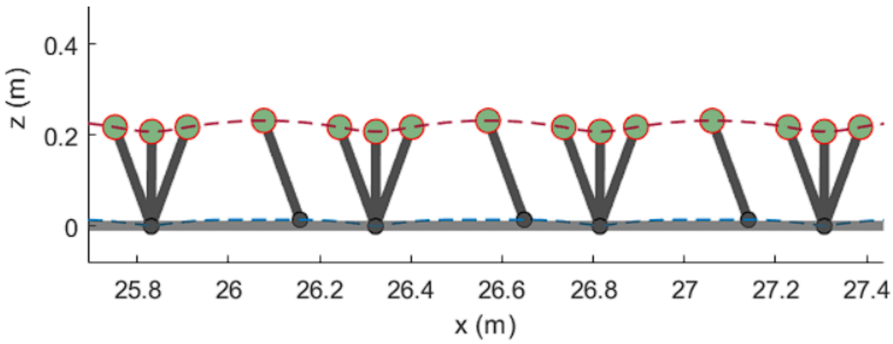
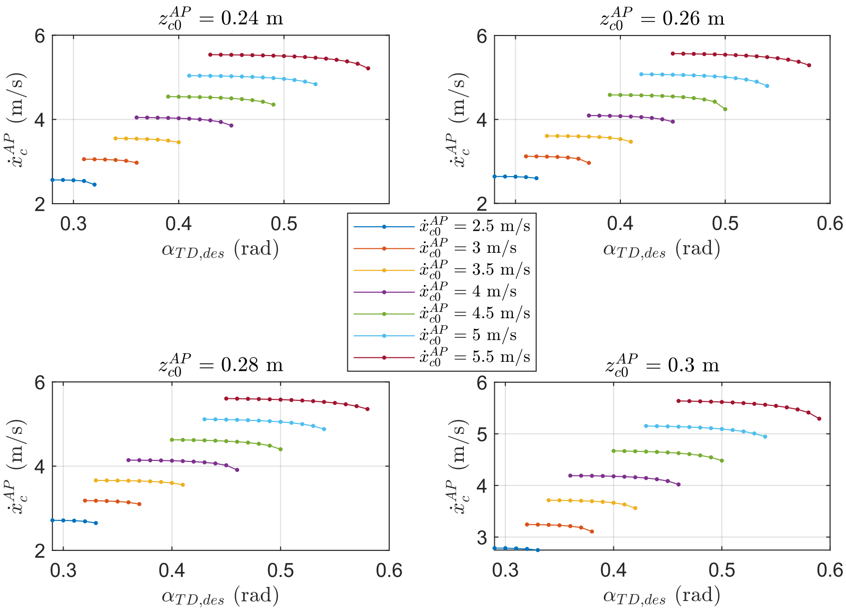


Figure 16: Periodic passive trajectory of the SLIP with some snapshots of the model configuration.

4.1.3 Auto-stabilization property of the SLIP

As highlighted in section 4.1.2, the SLIP can converge to periodic stable gaits while being initialized in their vicinity. This property of the SLIP template is described in [8] and referred to as auto-stabilization property. In Figure 17, the plots show the apex height ${z_c}^{AP}$ and the forward velocity ${\dot{x}_c}^{AP}$ after the system converges to a stable periodic gait by being initialized with different values of z_{c0}^{AP} and $\dot{x}_{c0}^{AP}$ and with different settings of the controlled leg angle $\alpha_{TD,des}$. Each point of a curve represents a simulation in which the SLIP managed to converge to a periodic gait. It is important to note that the gaits here presented are only a subsection of the possible periodic gaits of the SLIP. More precisely, these gaits are all stable, as these plots only shows limit cycles to which the model was attracted to and on which it remained indefinitely. More periodic gaits exist, generally unstable, which can be found through numerical search by looking for fixed points of a Poincaré map, as described in section 2.2. The model parameters used are the same used in the section 4.1.2.

(a) Forward velocity $\dot{x}_c^{AP}$ after convergence to a stable periodic gait



(b) Apex height z_c^{AP} after convergence to a stable periodic gait

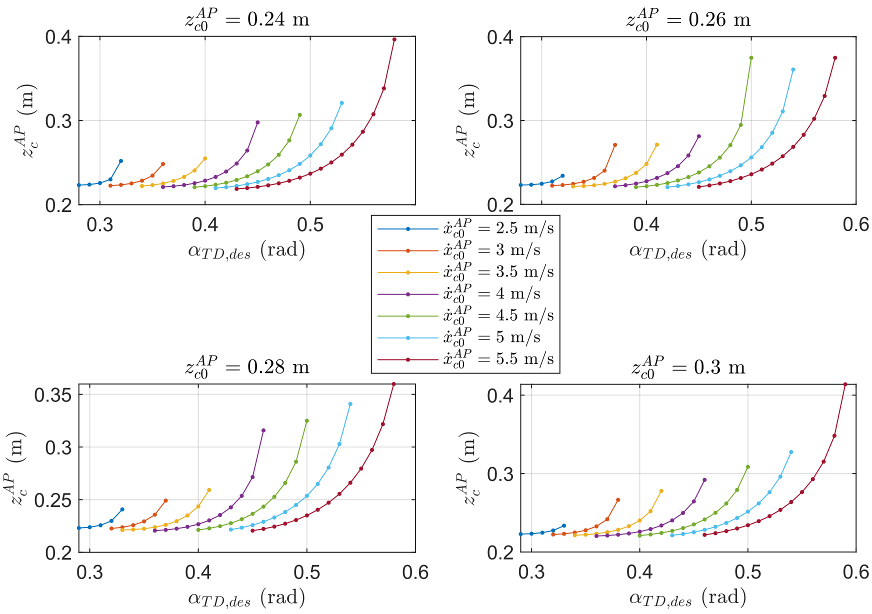


Figure 17: Self stabilization property of the SLIP template. The quantities on the vertical axes of the plots refer to the flight apexes after the SLIP has converged to a periodic gait. Each point of a curve represents a simulation in which the SLIP managed to converge to a stable periodic gait.

By collecting the data only based on the nominal flight apex states after the convergence and ignoring the initial conditions, the periodic stable gaits of the SLIP can be plotted as a surface as shown in Figure 18. Note that the fact that the periodic passive gaits are distributed on a surface, so that for a fixed horizontal velocity there are infinite gaits to choose from, is a particularity of the SLIP model and it does not hold for more complex models, as it will be shown in the next sections.

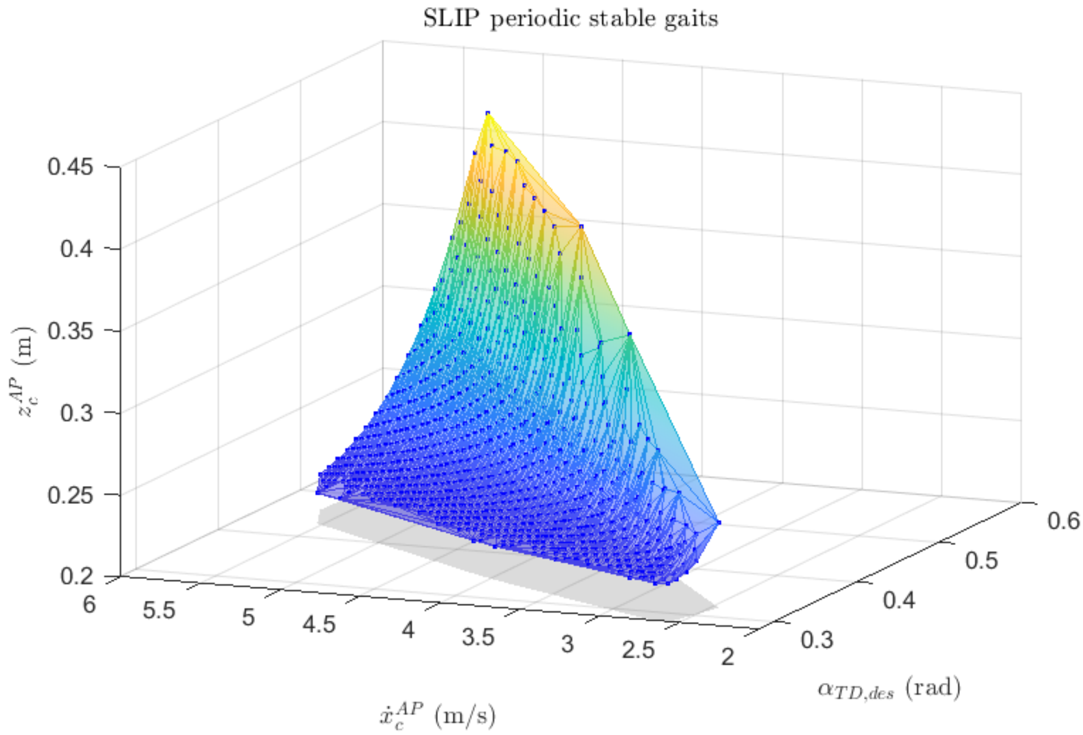


Figure 18: Slice of the surface representing the periodic stable gaits of the SLIP.

4.1.4 Simulation of an unstable periodic gait

In this section, the simulation of an unstable gait is presented. The gait was found numerically through the Poincaré maps method described in section 2.2. The Poincaré section was chosen to be on the apex of the gaits. The states of the system during flight are:

$$\boldsymbol{x} = \begin{bmatrix} x_c & \dot{x}_c & z_c & \dot{z}_c & \alpha & \dot{\alpha} & r & \dot{r} \end{bmatrix}^T \in \mathbf{X} \tag{27}$$

The desired Poincaré section can be defined as the set of states in which the vertical velocity is zero and the foot is not in contact with the ground:

$$\Sigma = \{\boldsymbol{x} \in \mathbf{X} \mid \dot{z}_c = 0 \;\cap\; z_c - r\cos(\alpha) > 0\} \tag{28}$$

Note that this section satisfies the transversality condition with respect to the flow, as the state of the robot only remains on the apex for the time instant in which $\dot{z}_c = 0$, after which the gravitational acceleration causes the velocity to become negative.

In defining the Poincaré map, a reduced state of the SLIP can be considered, as the horizontal coordinate x_c is not periodic and thus must not be included in the limit cycle analysis, and the vertical velocity, the angular velocity and the leg extension velocity are all identically null at every apex, $\dot{z}_c^{AP} = 0$ m/s, $\dot{\alpha}^{AP} = 0$ rad/s and $\dot{r}^{AP} = 0$ m/s. Finally r and α can also be removed, as during flight the leg spring is always at rest, $r^{AP} = r_0$ and the leg angle is fixed $\alpha = \alpha_{TD,des}$. The reduced state is therefore simply given by:

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_c & z_c \end{bmatrix}^T \in \hat{\mathbf{X}} \tag{29}$$

Thus a Poincaré map $P(\hat{\boldsymbol{x}})$ can be defined that returns the states at the apex $(k+1)$-th by starting from the k-th apex.

$P(\hat{\boldsymbol{x}})$ can be simply evaluated by numerically integrating the dynamics of the SLIP between two successive apexes. Moreover, fixed points of $P(\hat{\boldsymbol{x}})$ correspond to passive periodic gaits of the SLIP. In this work, the integration to evaluate the Poincaré map was done via *Simulink* and its fixed points were found by using the Levenberg-Marquardt optimization algorithm (LMA).

Figures 19 and 20 show the trajectories of the states in the phase plane and in time of a passive gait found through the fixed point method. The initial states were set to $\dot{x}_{c0} = 2.26$ m/s and $z_{c0} = 0.9169$ m and the desired leg angle to $\alpha_{TD,des} = 0.3011$ rad.

The instability of the gait eventually causes the robot to fall to the ground after ten steps. Figure 22 shows the trajectory of the robot in space. Compared to the stable gait shown in section 4.1.2, the robot jumps higher while advancing at a lower horizontal speed.

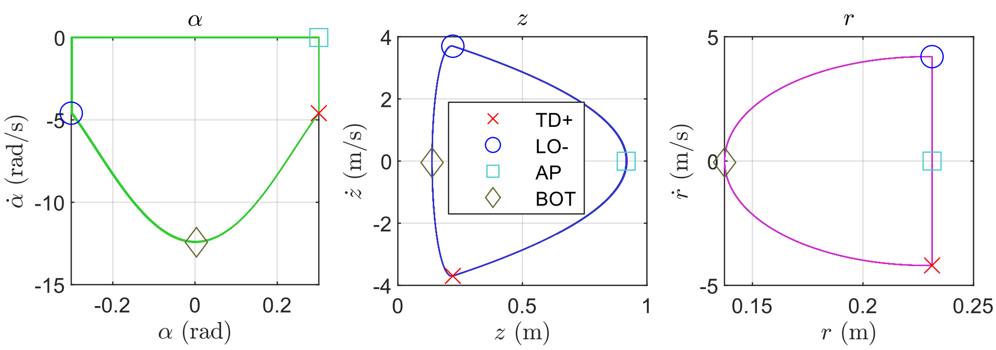


Figure 19: State trajectories of the SLIP template in the phase plane on the unstable periodic passive gait. Markers indicate the states just after touchdown “TD+” just before lift-off “LO-”, the apexes “AP” and the bottoms during stance “BOT”.

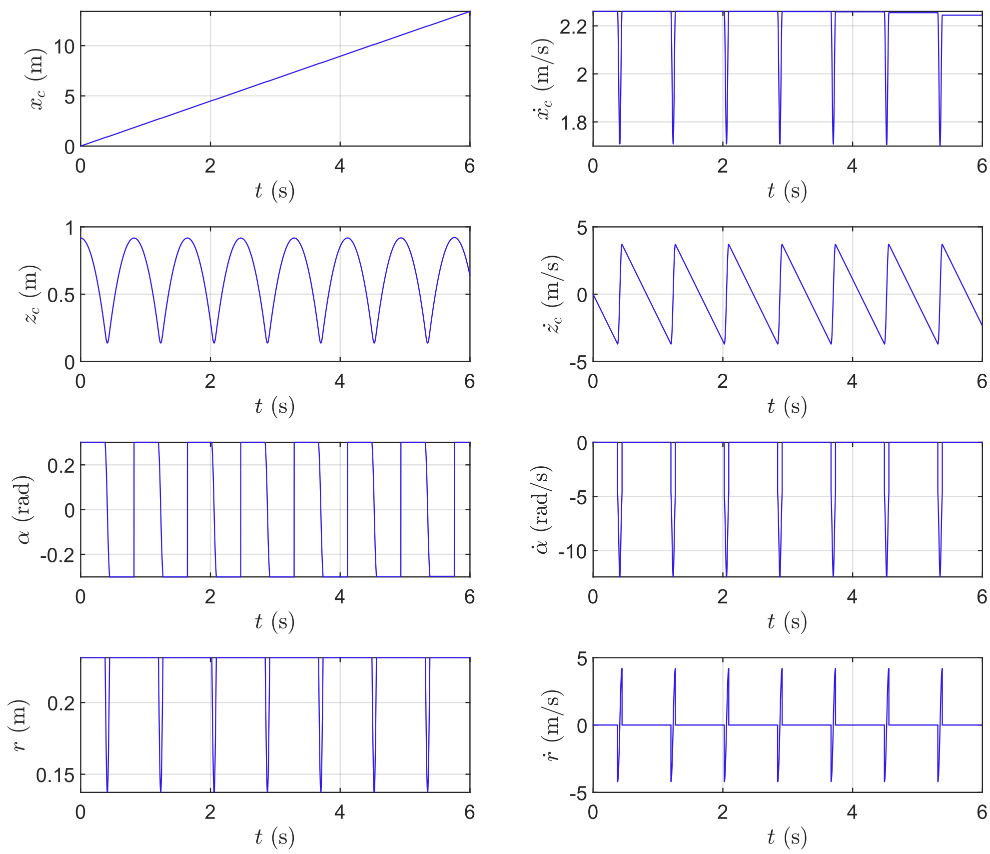


Figure 20: Plots of the state trajectories of the SLIP. The model was initialized on a periodic gait found numerically with the Poincaré maps method.

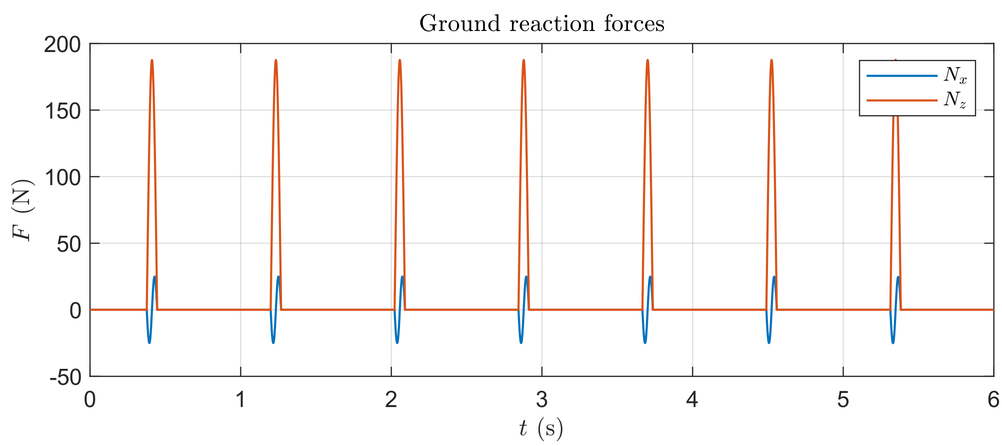


Figure 21: Ground reaction forces of the SLIP on the unstable passive gait.

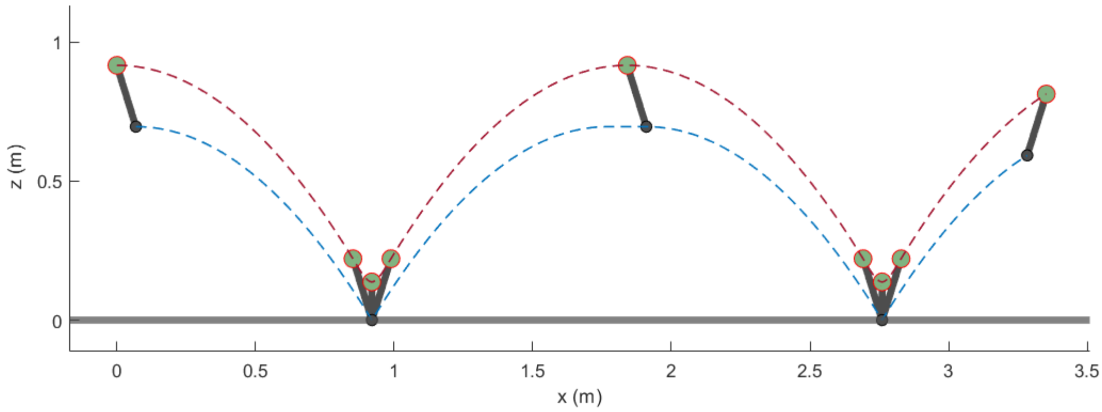


Figure 22: Unstable passive gait of the SLIP.

The fact that the periodic gait is unstable can also be confirmed by computing the characteristic multipliers of the gait, which are the eigenvalues of the linearized map $\boldsymbol{P}(\hat{\boldsymbol{x}})$ around the fixed point, as described in section 2.3. The eigenvalues of the map are plotted in Figure 23. One eigenvalue is in (1, 0i), which is caused by the fact that the model is energy conservative [8]. The other eigenvalue is unstable and located at (3.0842, 0i), meaning that the gait itself is also unstable.

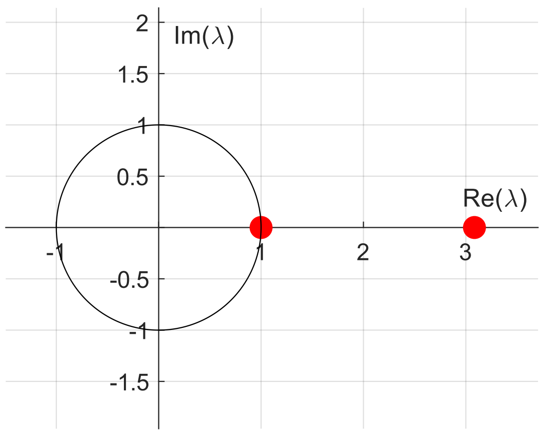


Figure 23: Eigenvalues of the unstable periodic gait of the SLIP.

4.2 Pitching-SLIP

In this section, an extension of the SLIP model, denominated "pitching-SLIP", is considered. The pitching-SLIP model extends the simple SLIP model by adding a dynamic equation that governs the pitch angle α during flight. Unlike in the SLIP model, the leg touchdown angle can not be controlled arbitrarily during flight. Instead, once the leg lifts off from the floor it keeps its angular velocity till the next touchdown event. The leg is still modelled as without inertia, which means that there is no energy loss at touchdown and the model is still conservative. However, the modified dynamics introduce a new constraint that reduces the number of possible passive periodic gaits by an infinity order.

4.2.1 Model and dynamics equations

The dynamics of the pitching-SLIP are described by very similar equations to those of the simple SLIP template.

The flight dynamics present a difference in the equation that governs α, where now the leg angular velocity $\dot{\alpha} = \text{const}$ is in general non-zero:

$$\begin{cases} \ddot{x}_c = 0 \\ \ddot{z}_c = -g \\ \ddot{\alpha} = 0 \\ r = r_0 \end{cases} \tag{30}$$

The stance phase dynamics remain unchanged:

$$\begin{cases} \ddot{\alpha} = -2\dfrac{\dot{r}}{r}\dot{\alpha} + \dfrac{g}{r}\sin(\alpha) \\ \ddot{r} = r\dot{\alpha}^2 - g\cos(\alpha) - \dfrac{K_r}{m}(r - r_0) \end{cases} \tag{31}$$

The take-off transition is given by:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- \\ \dot{z}_c^+ = \dot{z}_c^- \\ \dot{\alpha}^+ = \dot{\alpha}^- \\ r^+ = 0 \end{cases} \tag{32}$$

Finally, at touchdown, the states jump according to:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- \\ \dot{z}_c^+ = \dot{z}_c^- \\ \dot{\alpha}^+ = -\dfrac{1}{r}\cos(\alpha^-)\dot{x}_c^+ - \dfrac{1}{r}\sin(\alpha^-)\dot{z}_c^+ \\ r^+ = -\sin(\alpha^-)\dot{x}_c^+ + \cos(\alpha^-)\dot{z}_c^+ \end{cases} \tag{33}$$

4.2.2 Simulation results

In this section, the results of a simulation of the pitching-SLIP model are reported. The values used for the model parameter are reported in Table 4.

Parameter	Value
m	1.007 kg
K_r	2000 N/m
r_0	0.23124 m
g	9.81 m/s^2

Table 4: List of the numerical values assigned for the simulation of the pitching-SLIP.

Just like the SLIP, the pitching-SLIP can also exhibit passive periodic gaits. However, flying dynamics introduced in the pitching-SLIP mean that the only possible passive gaits include somersaults, since the angular velocity is preserved at lift-off. Once again, periodic gaits can be found with the fixed point method. The Poincaré section was chosen to be the apex, so the same expression as in (28), reported here for clarity, holds:

$$\Sigma = \{\boldsymbol{x} \in \mathbf{X} \mid \dot{z}_c = 0 \ \cap \ z_c - r\cos(\alpha) > 0\} \tag{34}$$

Due to the leg angle dynamics, this time the reduced state for the Poincaré map is:

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_c & z_c & \alpha & \dot{\alpha} \end{bmatrix}^T \in \hat{\mathbf{X}} \tag{35}$$

Figures 25 and 24 show the trajectories of the states in time and in the phase plane obtained in a simulation in which the pitching-SLIP was initialized on a periodic gait. The robot was initialized on an flight apex with $\alpha_0 = \pi$, $\dot{\alpha}_0 = -9.648$ rad/s, $\dot{x}_{c0} = 3.992$ m/s and $z_{c0} = 0.572$ m. Similarly to the case discussed with the SLIP, the gait here presented is unstable and eventually the robot falls to the ground. Figure 26 shows that the ground reaction forces have the same profile of the traditional SLIP model, while Figure 27 shows the trajectory of the robot with somersaults in space.

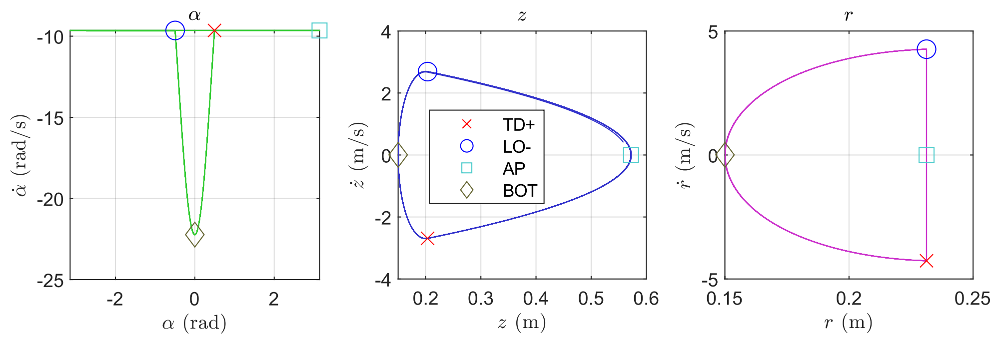


Figure 24: State trajectories of the pitching-SLIP template in the phase plane. Markers indicate the states just after touchdown “TD+” just before lift-off “LO-”, the apexes “AP” and the bottoms during stance “BOT”.

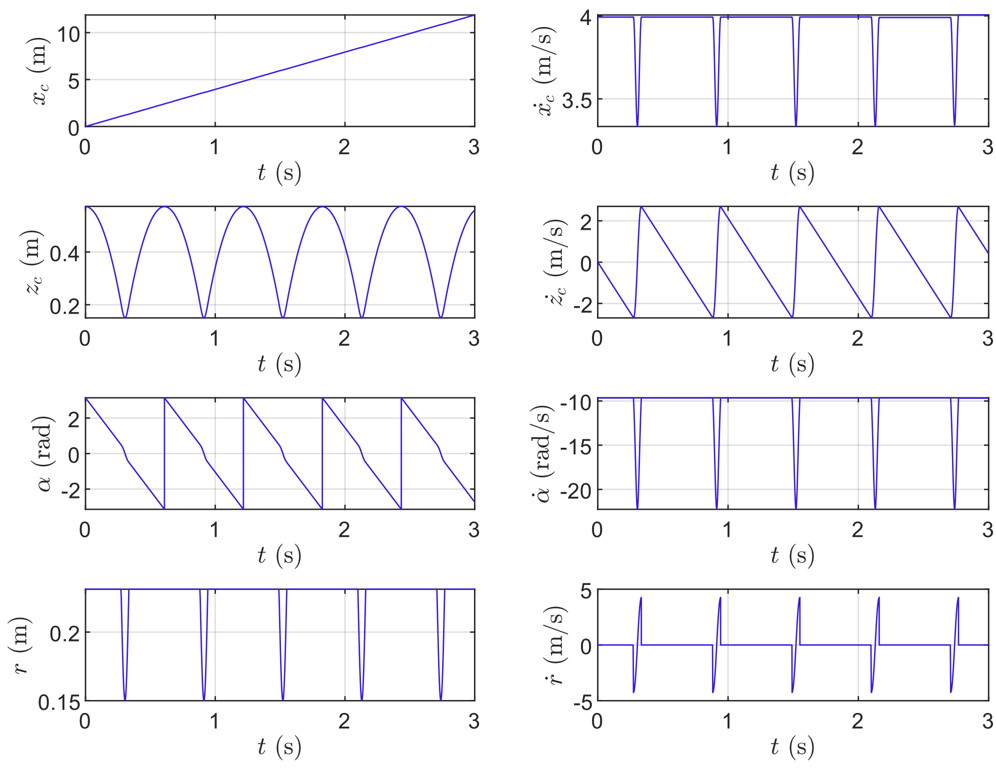


Figure 25: Plots of the state trajectories of the pitching-SLIP. The model was initialized on a periodic gait found numerically with the Poincaré maps method.

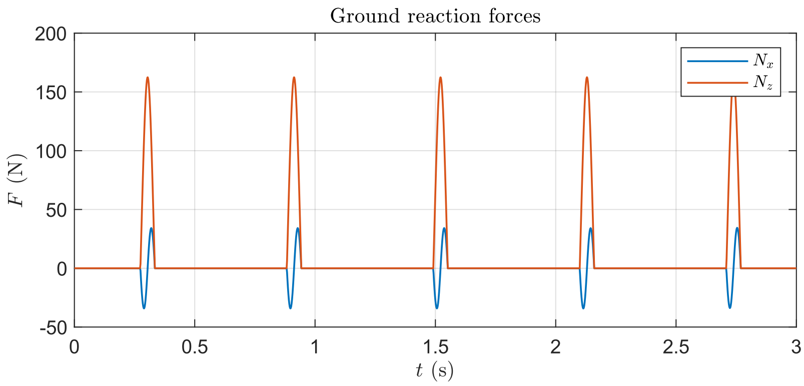


Figure 26: Ground reaction forces of the pitching-SLIP.

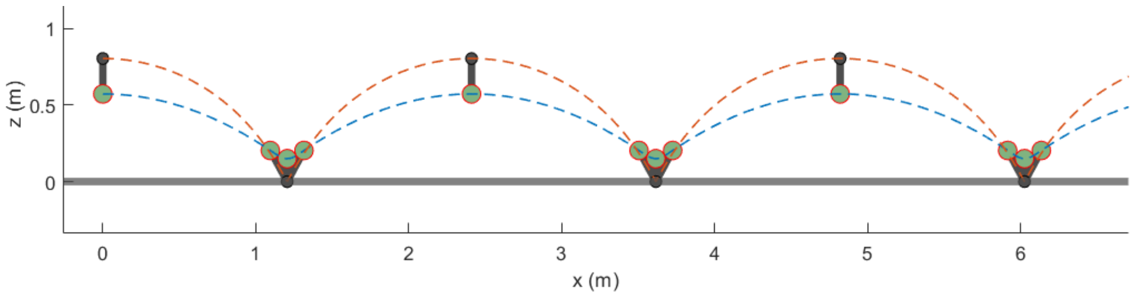


Figure 27: Unstable periodic motion of the pitching-SLIP.

The instability of the gait is again confirmed by computing the characteristic multipliers of the gait. The eigenvalues are plotted in Figure 28. Similarly to the SLIP, one eigenvalue is in (1, 0i), due to the conservative nature of the model. Another one is located in the origin, while two are located on the real axis outside the unit circle, making the gait unstable.

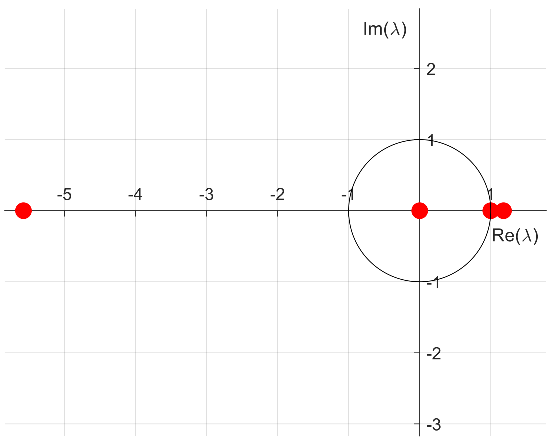


Figure 28: Eigenvalues of the unstable periodic gait of the pitching-SLIP.

4.2.3 Passive periodic gaits stability analysis

In this section, the periodic gaits of the pitching-SLIP are further analyzed. A search of the fixed points of the Poincaré map by starting from varying initial guesses made it possible to find a set of periodic gaits. When expressed in the nominal states at the apex, the periodic gaits of the pitching-SLIP take the shape of the curve displayed in Figure 29. The apex leg angle α_{nom}^{AP} is not considered in the plot, as it is common for all the gaits and set to $\alpha_{nom}^{AP} = \pi$. The stability of the gaits was inferred by linearizing the Poincaré map around the fixed points as described in section 2.3.

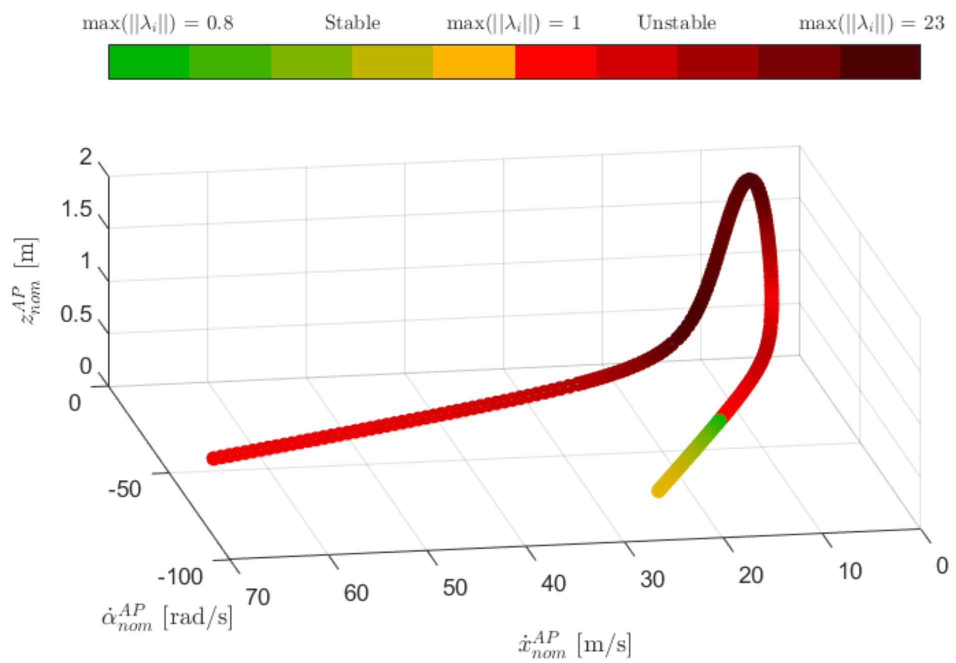


Figure 29: Geometric locus of the passive gaits of the pitching-SLIP. Every point of the curve corresponds to the apex state of a different passive gait. The colouring is based on the max $||\lambda_i||$, as specified by the colorbar on top of the graph

The curve presents two different branches that merge at the gait with the maximum apex height $z_{nom}^{AP} = 1.768$ m. The existence of two branches makes it possible to select two different gaits when fixing a parameter, for example two distinct gaits exist with $x_{nom}^{AP} = 15$ m/s.

In the graph, stable gaits are displayed with shades between green and yellow, whereas unstable gaits are depicted as red. The shading of the colour depends on largest eigenvalue magnitude associated with each gait. Interestingly, the stable gaits are all located on a single branch of the curve. Note however that one eigenvalue is always located in (1, 0i) because the system is conservative. This eigenvalue was not considered for the purpose of quantifying the stability. Moreover, the energy conservation of the system could be used to further reduce the state $\hat{\boldsymbol{x}}$ for the Poincaré map [8], which would probably make the eigenvalue in (1, 0i) disappear.

Figure 30 shows the root locus of the eigenvalues of the linearized Poincaré map around the fixed points of Figure 29. The eigenvalues were computed following the curve by starting from the left branch's and ending at the right branch's terminal points. One eigenvalue stays fixed in the (1,0i) location, while another is fixed on the origin. The third and fourth eigenvalues move along the real axis first away from the origin, then towards the unit circle until they meet and split into two conjugated complex branches that remain inside the unit circle.

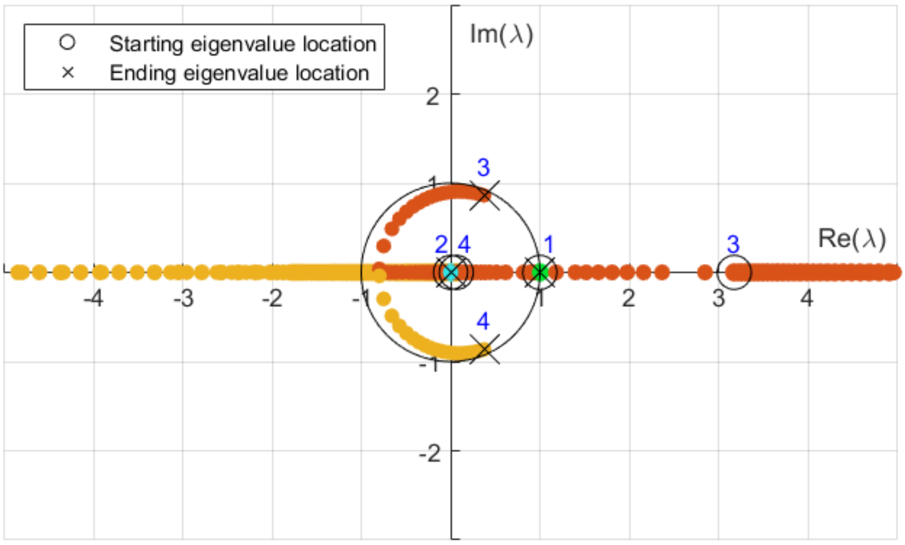


Figure 30: Root locus of the Jacobian matrix's eigenvalues. The circles indicate the starting locations and the red cross the ending locations. The eigenvalues were computed following the curve displayed in Figure 29 by starting from the left branch's and ending at the right branch's terminal points. The colours denote: the first (green), the second (cyan), the third (orange), and the fourth (yellow) eigenvalues.

The periodic gaits of the SLIP and the pitching-SLIP can be compared by plotting the following nominal states of the gaits: α_{nom}^{TD}, $\dot{x}_{c,nom}^{AP}$, $z_{c,nom}^{AP}$. As it can be seen in Figure 31, the gaits of the extended model are a subset of the gaits of the simple SLIP: the gaits of the pitching-SLIP form a curve lying on the surface that represents the gaits of the simple SLIP. The fact that the two models have a common set of periodic solution is due to the fact that they share the same stance dynamics. However, the angle flight dynamics of the pitching-SLIP introduces a new constraint: the angular velocity at lift-off must take the value that allows the leg to land after the flight phase with the same angle it had at the previous touchdown as a necessary condition for periodicity, whereas in the SLIP there is no condition on the angular velocity at touchdown as α is fixed during flight. In summary, this means that in a one-legged robot without hip in presence of a realistic flight phase, not all SLIP gaits can be realized, but only a one-dimensional subset of them corresponding to the pitching-SLIP gaits.

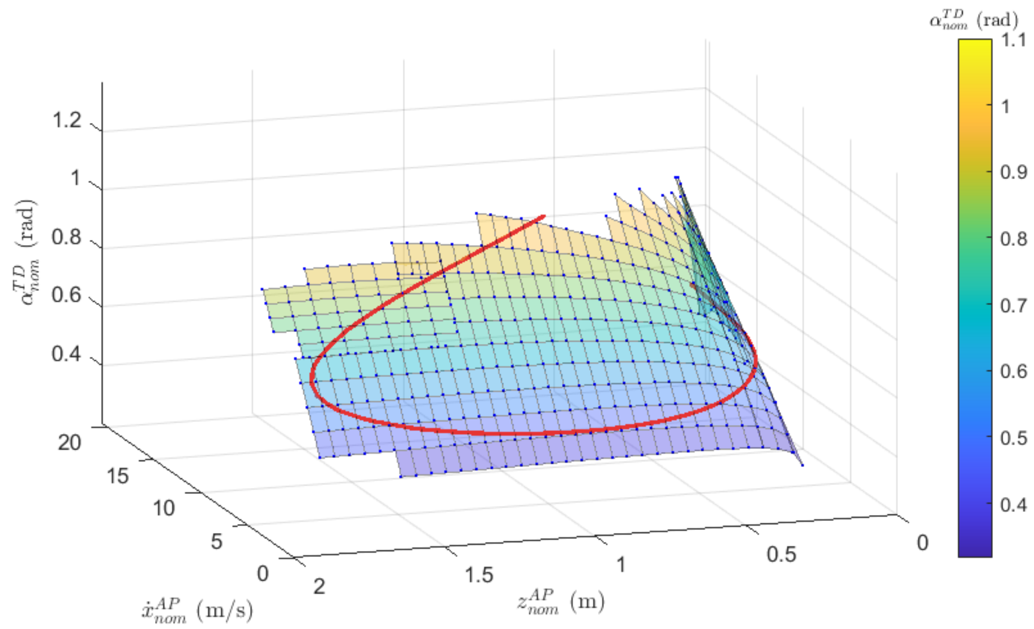


Figure 31: Periodic gaits of SLIP and pitching-SLIP compared. The surface and the red curve represent the periodic gaits of the SLIP and the pitching-SLIP respectively.

4.2.4 Energy and angular momentum control

4.2.4.1 Control goal

In this section, a controller to stabilize the passive running gaits with somersaults of the pitching-SLIP is described. The key idea is to stabilize the passive gaits so that at convergence the control effort tends to zero. First, an actuation force f acting along the linear spring and an actuation torque τ acting on the leg are added during the stance phase. The modified differential equations for the stance phase become:

$$\begin{cases} \ddot{\alpha} = -2\frac{\dot{r}}{r}\dot{\alpha} + \frac{g}{r}\sin(\alpha) + \frac{\tau}{mr^2} \\ \ddot{r} = r\dot{\alpha}^2 - g\cos(\alpha) - \frac{K_r}{m}(r - r_0) + \frac{f}{m} \end{cases} \tag{36}$$

while the flight dynamics remain unchanged.

The control approach consists in regulating the robot's energy and angular momentum around the foot so that they remain on the nominal values of a pre-computed passive gait. This idea comes from the observation that in a passive gait the energy of the system is invariant whereas the angular momentum around the foot is quasi-invariant, as it only changes during the stance phase but always returns to the same constant value at lift-off and for the subsequent flight phase.

The energy of the pitching-SLIP is given by:

$$E = mgz_c + \frac{1}{2}m\dot{r}^2 + \frac{1}{2}m\dot{r}^2\dot{\alpha}^2 + \frac{1}{2}K_r(r - r_0)^2 \tag{37}$$

The pitching-SLIP is energy conservative, so the energy can only be changed through the actuation effort, as expressed in the energy time derivative during the stance phase:

$$\dot{E} = \dot{\alpha}\tau + \dot{r}f \tag{38}$$

The angular momentum around the foot is simply computed as

$$l_f = mr^2\dot{\alpha} \tag{39}$$

and its time derivative during the stance phase, given the dynamics equations (36), is

$$\dot{l}_f = mgr\sin(\alpha) + \tau \tag{40}$$

A simple controller that regulates the energy and the angular momentum around the foot can be formulated as follows:

$$\tau = D_l \cdot (l_f - l_{f,des}), \tag{41}$$

$$f = -\gamma_E(E - E_{des})\dot{r}. \tag{42}$$

where D_l and γ_E are positive scalar gains and E_{des} and $l_{f,des}$ are the desired values for the energy and the angular momentum around the foot.

As anticipated, the desired values for the energy and angular momentum are chosen from a passive gait of the model, which can be pre-computed by searching for fixed points of a Poincaré map. By definition, the energy is invariant along a passive gait. The angular momentum is time-varying during stance but, if the gait is passive, the net tipping moment of the gravitational force is zero during each stance phase which means the angular momentum at lift-off returns to the value it had at the previous touchdown event. Thus, the angular momentum in a passive gait is invariant in all the flight phases. A simple choice for the desired set-points is then:

$$E_{des} := E_{nom}^{LO-} \tag{43}$$

$$l_{f,des} := l_{f,nom}^{LO-} \tag{44}$$

The controller is validated numerically in the next section.

4.2.4.2 Simulation results of the pitching-SLIP

To test the controller, an unstable passive gait was selected. The initial conditions on the apex are given by $\alpha_0 = \pi$, $\dot{\alpha}_0 = -44.120$ rad/s, $\dot{x}_{c0} = 13.197$ m/s and $z_{c0} = 0.200$ m.

The open loop simulation results of the passive gait are reported in Figure 32. Observe how the model slowly deviates from the nominal gait and falls to the ground after 24 steps.

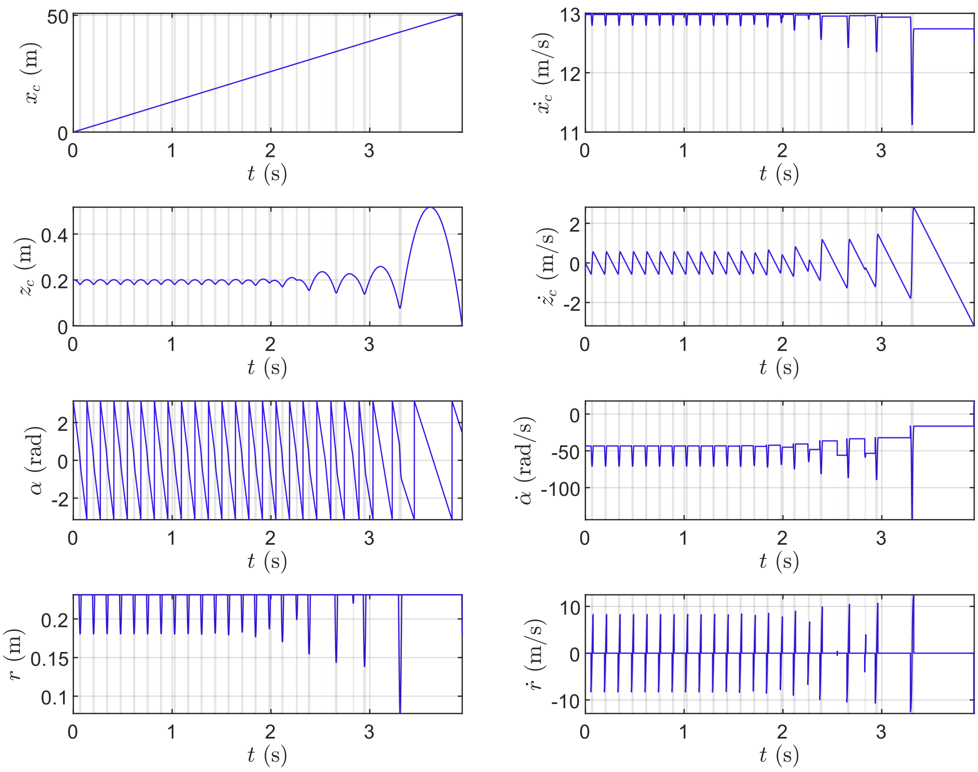


Figure 32: Open-loop simulation of an unstable passive gait of the pitching-SLIP.

The same gait was stabilized using the controller (41)-(44) with gains $D_l = 6\ \mathrm{s}^{-1}$ and $\gamma_E = 3\ \mathrm{sm}^{-2}$. Figure 33 shows that the controller stabilizes the robot bringing it to a periodic gait.

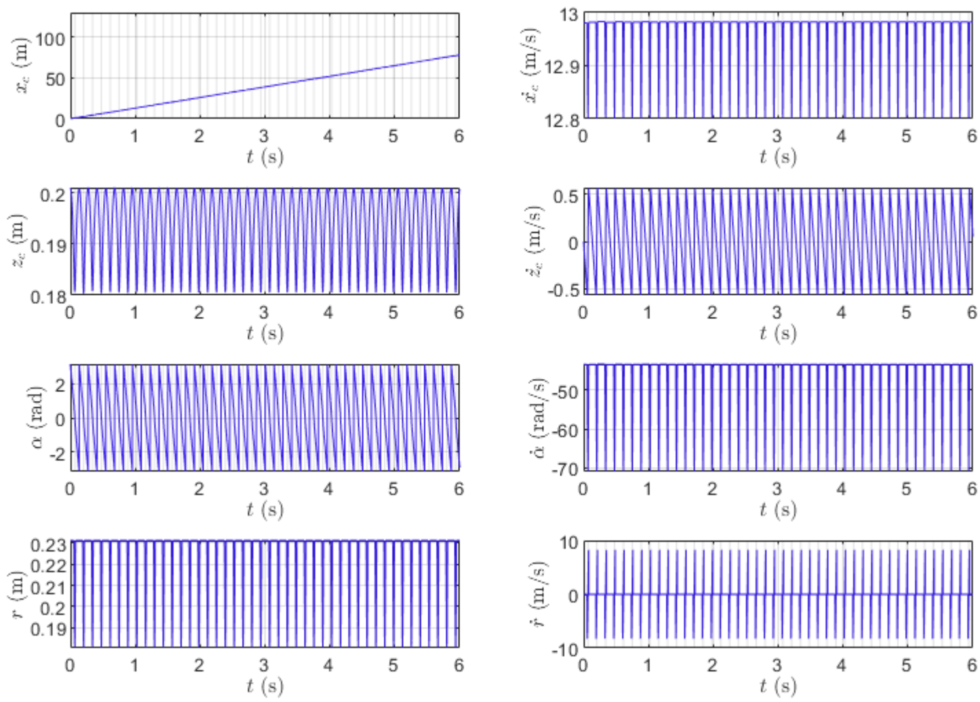


Figure 33: Closed-loop simulation of an unstable passive gait of the pitching-SLIP.

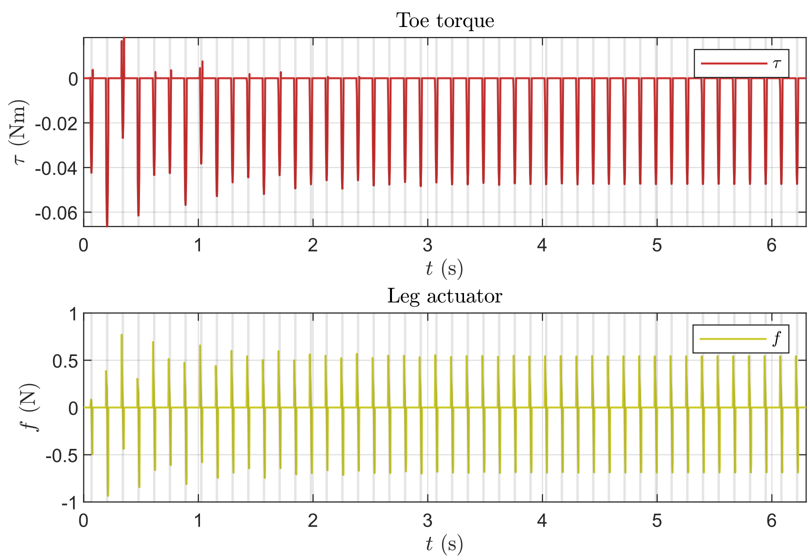


Figure 34: Actuation force and torque in the closed-loop simulation of the pitching-SLIP.

Consider now the case in which there is a disturbance on the initial horizontal velocity, $\delta \dot{x}_c = -0.3$ m/s. Figures 35 and 36 show that the controller is able to lead back the system close to the desired passive gait.

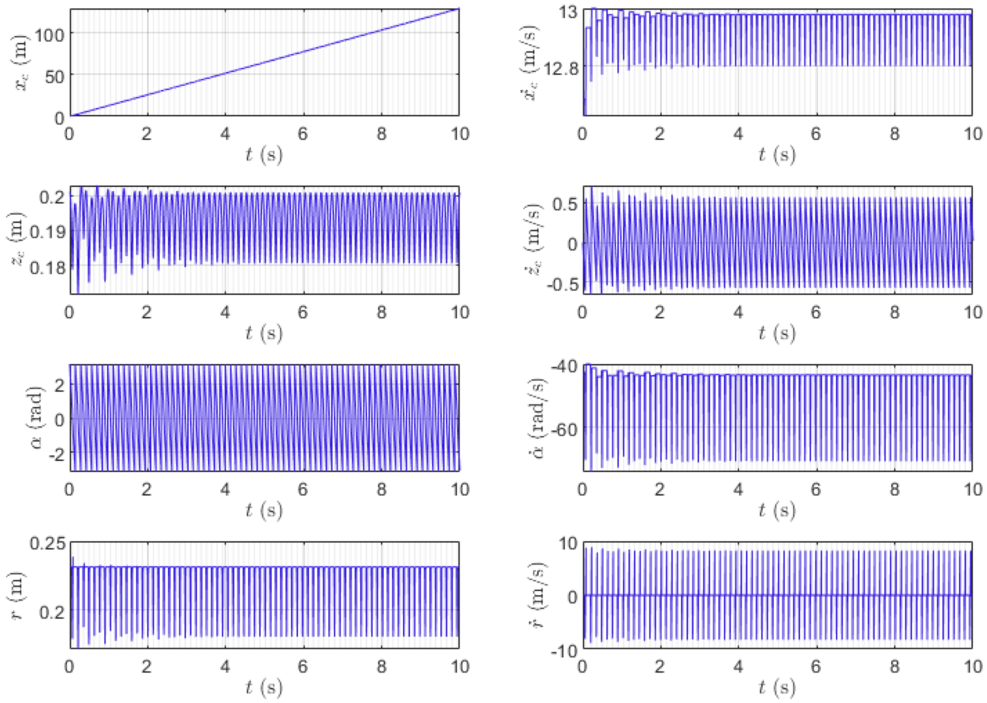


Figure 35: Closed-loop simulation of an unstable passive gait of the pitching-SLIP. The initial condition was disturbed with $\delta\dot{x}_c = -0.3$ m/s.

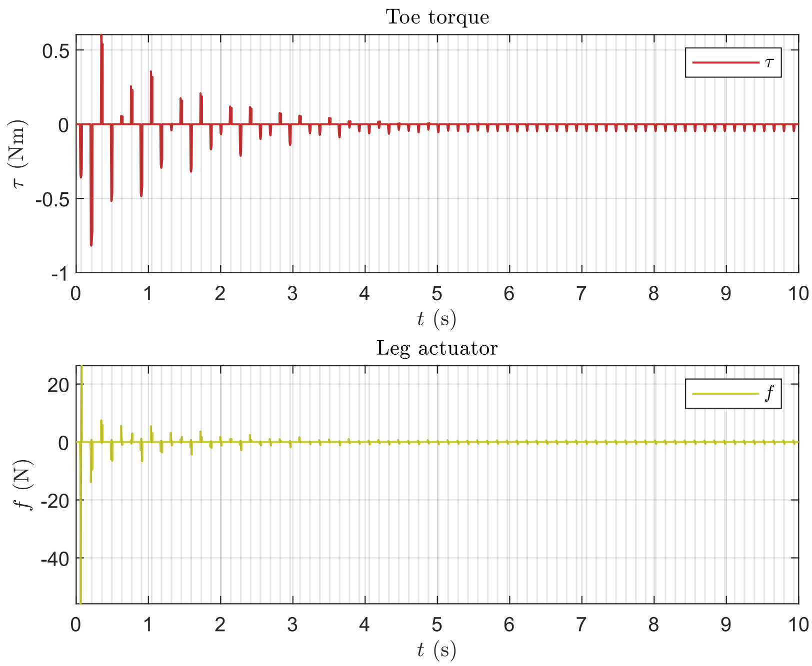


Figure 36: Actuation force and torque in the closed-loop simulation with the disturbance on the initial conditions.

4.2.4.3 Modified controller

As one may observe from Figures 34 and 36, the control input after convergence is low, but it does not converge to zero. This also means that the system does not fully converge to a passive gait, even though it is "close" to it. This is due to the gravity term in equation (40), which always causes the angular momentum to act by trying to compensate the change in l_f. The related torque actuation also induces a change in the energy derivative (38) through the term $\dot{\alpha}\tau$. In this section, a modified angular momentum regulator (41) to overcome this problem is proposed.

From (40), it is known that the angular momentum derivative during a passive gait stance phase ($\tau = 0$) evolves according to:

$$\dot{l}_{f,nom} = mgr\sin(\alpha) \tag{45}$$

By integration, on a passive gait the angular momentum evolves according to:

$$l_{f,nom}(\tilde{t}) = l_{f,nom}^{TD+} + \int_0^{\tilde{t}} mgr(\tilde{t})\sin(\alpha(\tilde{t}))d\tilde{t} \tag{46}$$

where $\tilde{t}$ indicates the time elapsed since the beginning of the stance phase and the time dependency of the variables was explicitly expressed for clarity. Since the angular momentum during passive gaits is naturally time-varying and its nominal trajectory is know (46), a time-varying desired set-point for the angular momentum can be defined instead of using a constant desired value like in (44):

$$l_{f,des}(\tilde{t}) := l_{f,nom}(\tilde{t}) = l_{f,nom}^{TD+} + \int_0^{\tilde{t}} mgr(\tilde{t})\sin(\alpha(\tilde{t}))d\tilde{t} \tag{47}$$

With the time-varying set-point (47), the angular momentum regulator (41) performs a trajectory tracking task. Note that the integral in (47) can not be explicitly integrated as an analytical solution for the stance phase is not available, but it can be numerically computed during the stance phase by measuring r and α. Also note that in a passive gait $l_{f,nom}^{TD+} = l_{f,nom}^{LO-}$, since both r and α have even symmetry with respect to time in a nominal stance phase.

The simulation results of the closed-loop system with time-varying angular momentum set-point are shown in Figures 37 and 38. The same disturbance on the initial conditions $\delta\dot{x}_c = -0.3$ m/s was applied. From the plots of the actuation force and torques, it can be seen that the trajectory tracking of the angular momentum is effective in compensating the errors previously discussed and the controller effort converges to zero.

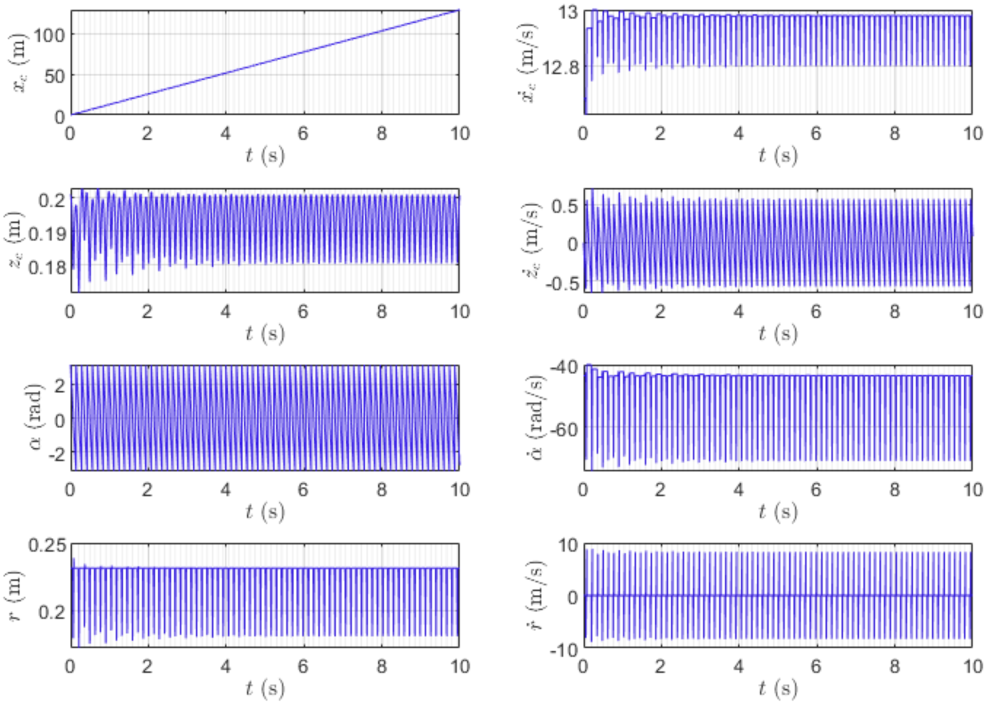


Figure 37: Closed-loop simulation of an unstable passive gait of the pitching-SLIP with the controller modified to track the time-varying angular momentum.

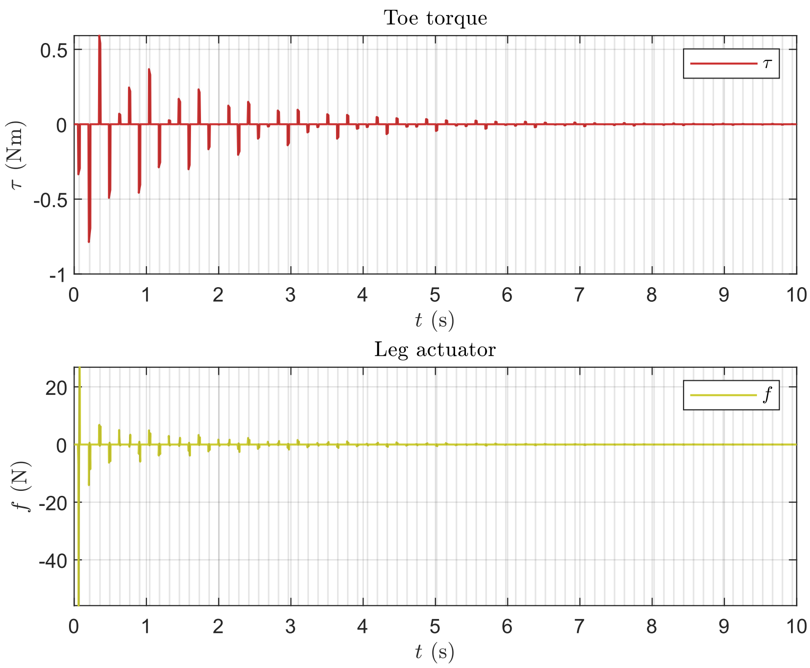


Figure 38: Actuation force and torque with the modified controller.

Thus, the controller is able to stabilize pre-computed passive gaits with a residual error after convergence in case of constant angular momentum set-point and with zero error after convergence in case of time-varying angular momentum set-point. This is reflected by the control effort after convergence.

However, by testing the controller in several scenarios, it has been concluded that it is not capable of stabilizing all the passive gaits shown in Figure 29. In particular, slower gaits required higher gain values and for the "more unstable" gaits it was not possible to tune the gains such that the closed-loop system was stable. Nonetheless, the controller is capable of stabilizing passive gaits in a wide range of velocities.

4.3 Body-SLIP

As a further extension of the pitching-SLIP model, an inertia J_c can be added to the leg. Such a model is object of study of this section and is hereby referred to as "body-SLIP".

4.3.1 Model and dynamics

Equations (48)-(51) describe the dynamics of the body-SLIP. The flight dynamics and lift-off transition equations remain unchanged, while the stance and touchdown equations are affected by the newly introduced leg inertia. A preload on the spring with force equal to the robot's weight mg was also added, so that the spring is compressed at the leg's natural length $r = r_0$ and the spring actual resting point is $r = r0 + mg/K_r$.

The flight dynamics are described by:

$$\begin{cases} \ddot{x}_c = 0 \\ \ddot{z}_c = -g \\ \ddot{\alpha} = 0 \\ r = r_0 \end{cases} \tag{48}$$

The stance dynamics are:

$$\begin{cases} \ddot{\alpha} = -\dfrac{2mr\dot{r}}{J_c + mr^2}\dot{\alpha} + \dfrac{mgr\sin(\alpha)}{J_c + mr^2} + \dfrac{\tau}{J_c + mr^2} \\ \ddot{r} = r\dot{\alpha}^2 + g(1 - \cos(\alpha)) - \dfrac{K_r}{m}(r - r_0) + \dfrac{f}{m} \end{cases} \tag{49}$$

At lift-off, the states jump according to:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- \\ \dot{z}_c^+ = \dot{z}_c^- \\ \dot{\alpha}^+ = \dot{\alpha}^- \\ r^+ = 0 \end{cases} \tag{50}$$

Finally, the touchdown transition is given by:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- - \dfrac{J_c \cos(\alpha^-)}{J_c + mr^2} \mu_{TD}^- \\ \dot{z}_c^+ = \dot{z}_c^- - \dfrac{J_c \sin(\alpha^-)}{J_c + mr^2} \mu_{TD}^- \\ \dot{\alpha}^+ = \dot{\alpha}^- - \dfrac{mr}{J_c + mr^2} \mu_{TD}^- \\ r^+ = -\sin(\alpha^-)\dot{x}_c^+ + \cos(\alpha^-)\dot{z}_c^+ \end{cases} \tag{51}$$

where μ_{TD}^- is the energy dissipation coefficient [14], defined as:

$$\mu_{TD}^- = r^- \dot{\alpha}^- + \cos(\alpha^-)\dot{x}_c^- + \sin(\alpha^-)\dot{z}_c^- \tag{52}$$

Unlike the SLIP and the pitching-SLIP, the body-SLIP is not energy conservative. In fact, energy is in general lost at touchdown. The only case in which no energy lost at touchdown is when $\mu_{TD}^- = 0$.

4.3.2 Passive Gaits of the body-SLIP

An analysis of the periodic gaits of the body-SLIP is presented in this section. For the search of the periodic gaits, the same reduced state and Poincaré section used for the pitching-SLIP as described in section 4.2.2 were used. The numerical values used for the simulations are reported in Table 5.

Parameter	Value
m	1.007 kg
K_r	2000 N/m
r_0	0.23124 m
J_c	0.001 kgm^2
g	9.81 m/s^2

Table 5: List of the numerical values assigned for the simulation of the body-SLIP.

The periodic gaits of the body-SLIP are represented in Figure 39. Like for the pitching-SLIP, the apex leg angle α_{nom}^{AP} is not considered in the plot, as it is always $\alpha_{nom}^{AP} = \pi$. The gaits found are all characterized by $\mu_{TD}^- = 0$, which is a necessary condition for the gait to be passive, since the energy of the robot must remain constant. This constraint was not manually forced but was automatically satisfied by searching for the fixed points of the Poincaré map.

By linearizing the Poincaré map around the fixed points, all gaits were found to be unstable. This was to be expected, as a perturbation of a periodic passive gait leads to a landing with $\mu_{TD}^- \neq 0$, causing a loss of energy which can not be recovered without actuation input.

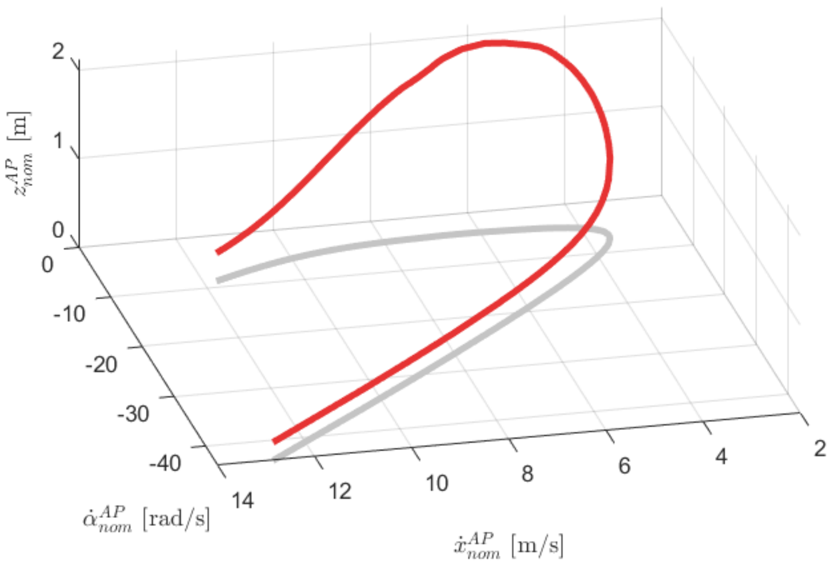


Figure 39: Geometric locus of the passive gaits of the body-SLIP. Every point of the curve corresponds to the apex state of a different passive gait. All the periodic gaits are unstable.

4.3.3 Simulation of the body-SLIP with controller

4.3.3.1 Simulation 1

In this section, the controller based on energy and angular momentum regulation described in section 4.2.4 is applied to the body-SLIP. The selected passive gait is given by the following states at the apex: $\alpha = \pi$, $\dot{\alpha} = -28.98$ rad/s, $\dot{x}_c = 8.637$ m/s, $z = 0.2303$ m.

The control gains were set to $D_l = 30$ s^{-1} and $\gamma_E = 18$ sm^{-2}. For the desired angular momentum, the time-variant set-point was used. The robot was initialized on the passive gait with an additional disturbance on the horizontal velocity, $\delta\dot{x}_c = +0.05$ m/s.

The evolution of the states of the robot is reported in Figure 40. The controller successfully stabilizes the robot. However, the control effort does not converge to zero, unlike in the simulations with the pitching-SLIP. Instead, a residual control effort remains after convergence, as shown in Figure 41. This is most likely caused by the energy losses at every touchdown. In fact, observe in Figure 42 how the energy dissipation coefficient does not converge to zero, meaning that even at steady state energy is lost at every touchdown. It was also verified that even without the initial disturbance the controller stabilizes the robot without converging to the exact passive gait and the residual effort at steady state is the same.

Finally, the ground reaction forces are displayed in Figure 43.

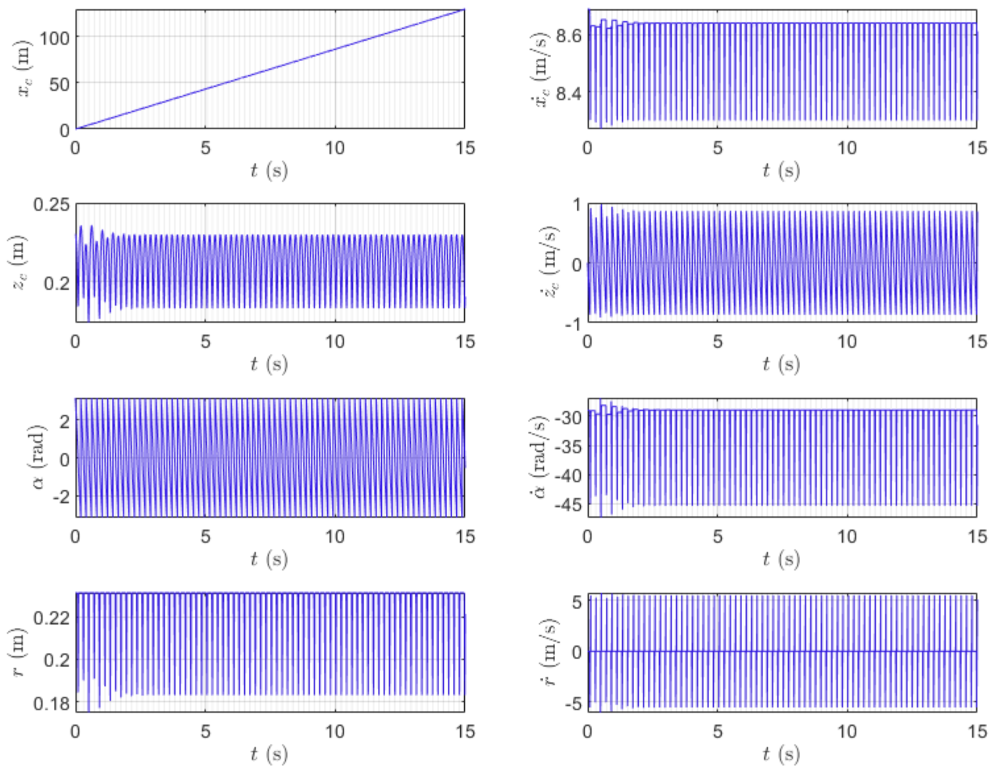


Figure 40: Closed-loop simulation of an unstable passive gait of the body-SLIP with initial disturbance $\delta\dot{x}_c = +0.05$ m/s.

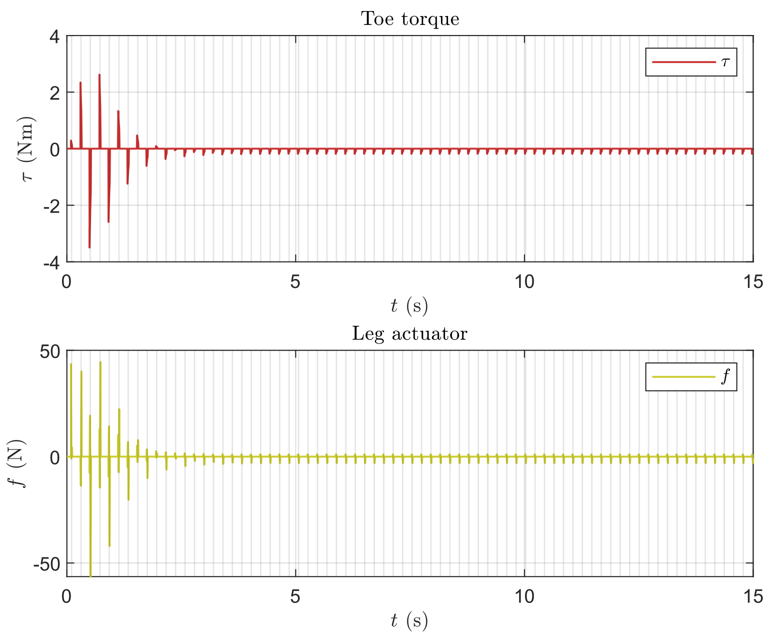


Figure 41: Actuation force and torque in the closed-loop simulation of the body-SLIP.

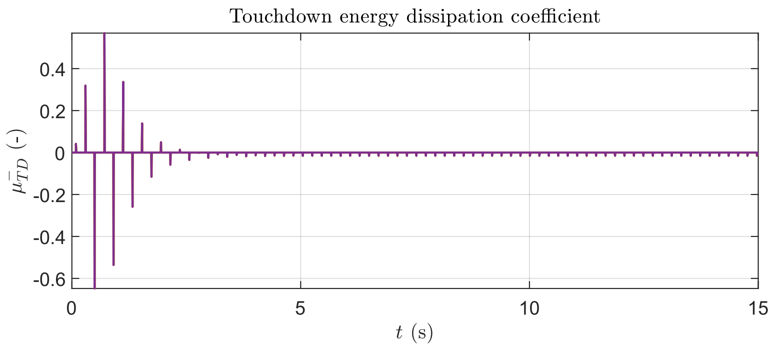


Figure 42: The energy dissipation coefficient does not converge to zero, causing energy loss at every touchdown event.

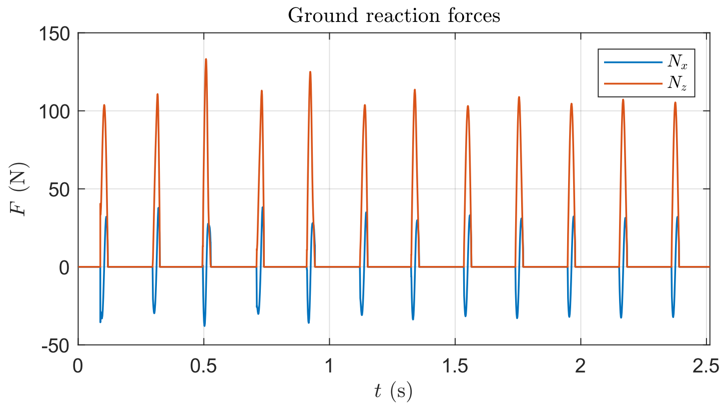


Figure 43: Detail of the ground reaction forces at the beginning of the closed-loop simulation of the body-SLIP.

4.3.3.2 Simulation 2

For completeness, another simulation of the body-SLIP is reported in this section. The passive gait was selected to be closer to the one used for the pitching-SLIP in section 4.2.4.3, with $\alpha = \pi$, $\dot{\alpha} = -43.914$ rad/s, $\dot{x}_c = 12.864$ m/s, $z = 0.2046$ m. This gait is "less unstable" than the one used in the previous section and it was possible to stabilize it with lower control gains: $D_l = 6\ \mathrm{s}^{-1}$ and $\gamma_E = 3\ \mathrm{sm}^{-2}$. Again, the robot was initialized on the passive gait with an additional disturbance on the horizontal velocity, $\delta\dot{x}_c = +0.1$ m/s. Figures 44 and 45 show the states and the actuation effort of the robot.

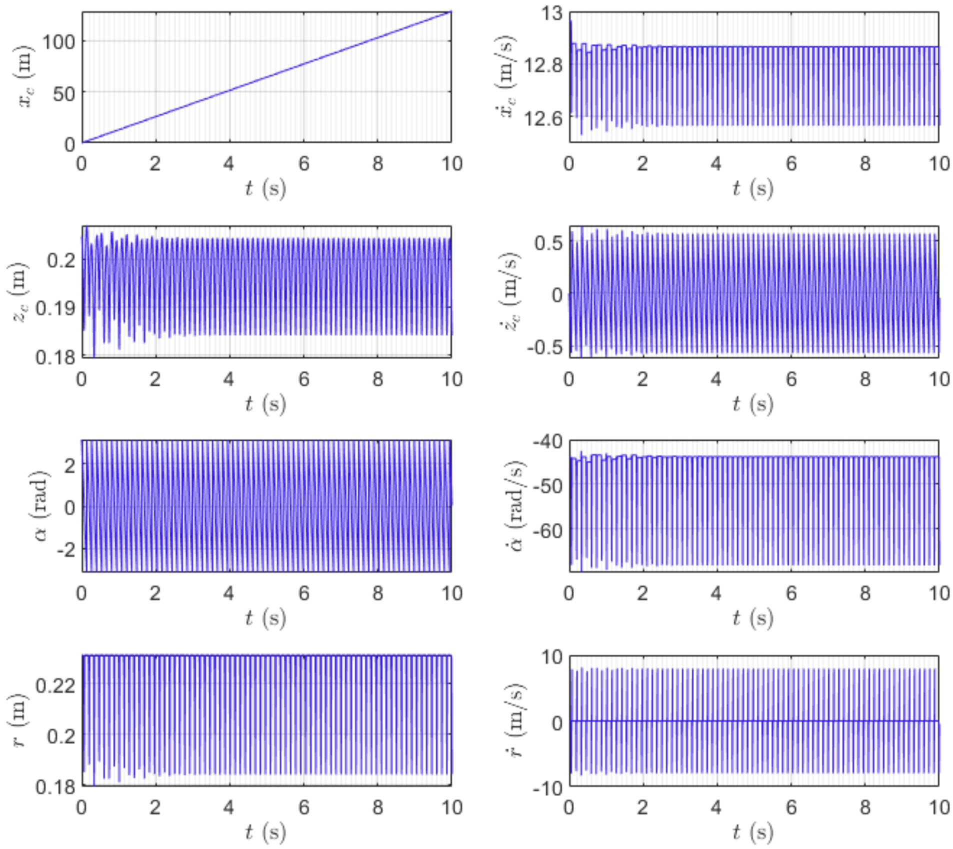


Figure 44: Closed-loop simulation of an unstable passive gait of the body-SLIP with initial disturbance $\delta \dot{x}_c = +0.1$ m/s.

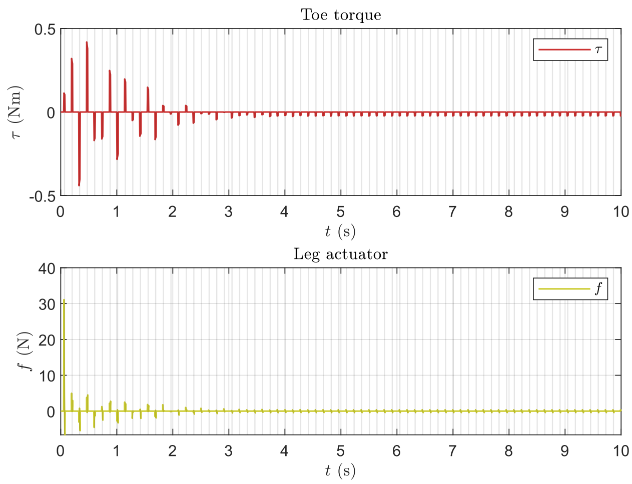


Figure 45: Actuation force and torque in the second closed-loop simulation of the body-SLIP.

In conclusion, the proposed control algorithm is able to stabilize the otherwise unstable body-SLIP by using the robot's passive gaits as reference trajectories. Even though the closed-loop simulations do not converge to exact passive gaits, the resulting control effort at steady state is low. Like in the case of the pitching-SLIP, there are limitations in the control approach. The approach relies on pre-computed periodic gait of the robots in order to use their energy and angular momentum as reference. However, such gaits do not appear to exist or it was not possible to find them at lower horizontal velocities. Moreover, the controller was able to stabilize only a subset of the periodic gaits obtained numerically.

5 Planar one-legged robot with hip

5.1 Model of the robot

In this chapter, two different approaches to control the locomotion of a planar one-legged robot with a body connected through a hip revolute joint to the leg are discussed. The robot model considered is the one depicted in Figure 46. This model was studied by Thompson and Raibert in [5], where they showed that it is capable of running passively. Further studies were conducted in [16], [15], [22] and [14], all of which focus on formulating controllers capable of stabilizing the passive running gaits of the one-legged robot.

In this section, the equations of the dynamics are briefly reviewed. Hereby, the hybrid nonlinear formulation of the model as described by Hyon and Emura in [14] is considered.

The hopper model extends the ones previously discussed by adding a body link with mass m_b and inertia J_b, which is connected to the leg via a revolute joint. The leg also has its own mass m_l and inertia J_l. The inertias J_b and J_l are both referred to the hip revolute joint. The center of mass of the robot is located at the hip revolute joint. Between the body link and the leg, a rotational spring with stiffness K_h is inserted. The foot's mass is considered to be negligible and the foot is therefore modelled as massless. No damping or friction is present in the system. Finally, a linear force actuator can act with force f along the leg prismatic joint, whereas a torque motor can exert a torque τ between the body link and the leg.

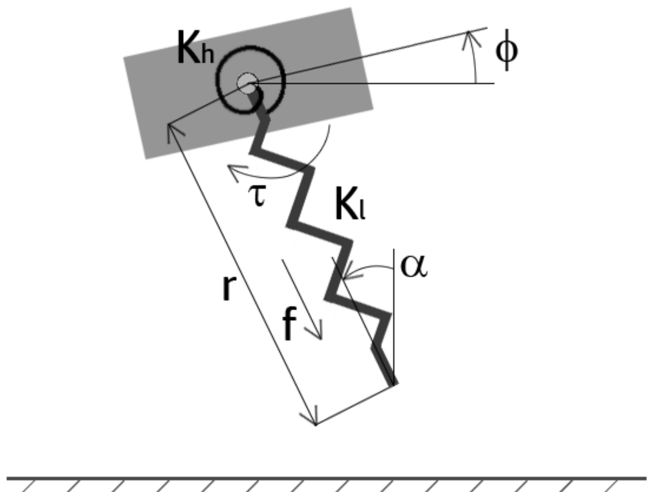


Figure 46: Planar one-legged hopper model with a body attached to the hip revolute joint.

A description of the parameters and variables of the model and the values used for its simulation are summed up in the tables 6 and 7.

Parameter	Description	Value
m	Total mass of body and leg	1.007 kg
J_l	Leg's inertia	0.002 kgm^2
J_b	Body's inertia	0.008 kgm^2
K_r	Leg stiffness	2000 N/m
K_h	Hip stiffness	1 Nm/rad
r_0	Leg resting length	0.23124 m
g	Acceleration of gravity	9.81 m/s^2

Table 6: List of the numerical values assigned for the simulation of the robot.

Variable	Description
x_c, z_c	Center of mass coordinates in the inertial reference system
x_f, z_f	Foot's coordinates in the inertial reference system
ϕ	Absolute body's orientation angle
α	Absolute leg's orientation angle
r	Leg length
f	Leg actuation force
τ	Hip actuation torque

Table 7: Description of the varaibles of the planar one-legged robot with body link.

As in the previous models, the ground is modeled as infinitely rigid and the friction coefficients are assumed to be big enough so that the ground's reaction force is always contained in the friction cone. The equations (53)-(54) describe the flow dynamics of the flight and stance phases and equations (55)-(56) describe the transitions of the states at the events of lift-off and touchdown.

During flight, the robot's center of mass follows a ballistic trajectory and the angular momentum of the system is preserved. There is no oscillation along the leg prismatic joint thanks to the hypothesis of massless foot.

$$\begin{cases} \ddot{x}_c = 0 \\ \ddot{z}_c = -g \\ J_l\ddot{\alpha} + J_b\ddot{\phi} = 0 \\ J_b\ddot{\phi} + K_h(\phi - \alpha) = \tau \\ r = r_0 \end{cases} \tag{53}$$

During the stance phase, the system evolves according to (54). In this model, the linear spring on the leg is preloaded with a force equal to the robot's weight.

$$\begin{cases} m\ddot{r} + K_r(r - r_0) - mr\dot{\alpha}^2 = f + mg(1 - \cos(\alpha)) \\ (J_l + mr^2)\ddot{\alpha} + J_b\ddot{\phi} + 2mr\dot{r}\dot{\alpha} = rmg\sin(\alpha) \\ J_b\ddot{\phi} + K_h(\phi - \alpha) = \tau \end{cases} \tag{54}$$

In the equations (55) and (56) describing the phase transitions, the subscript notations "-" and "+" indicate respectively the states before and after the phase is switched. At the lift-off transitions, the states velocities remain unchanged with the exception of $\dot{r}$.

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- \\ \dot{z}_c^+ = \dot{z}_c^- \\ \dot{\alpha}^+ = \dot{\alpha}^- \\ \dot{\phi}^+ = \dot{\phi}^- \\ \dot{r}^+ = 0 \end{cases} \tag{55}$$

At the touchdown transition, the states change according to the hypothesis of inelastic impulsive impact:

$$\begin{cases} \dot{x}_c^+ = \dot{x}_c^- - \dfrac{J_b\cos(\alpha^-)}{J_b + mr^2}\mu_{TD}^- \\ \dot{z}_c^+ = \dot{z}_c^- - \dfrac{J_b\sin(\alpha^-)}{J_b + mr^2}\mu_{TD}^- \\ \dot{\alpha}^+ = \dot{\alpha}^- - \dfrac{mr}{J_b + mr^2}\mu_{TD}^- \\ \dot{\phi}^+ = \dot{\phi}^- \\ \dot{r}^+ = -\sin(\alpha^-)\dot{x}_c^+ + \cos(\alpha^-)\dot{z}_c^+ \end{cases} \tag{56}$$

where μ_{TD}^- is defined as:

$$\mu_{TD}^- = r^-\dot{\alpha}^- + \cos(\alpha^-)\dot{x}_c^- + \sin(\alpha^-)\dot{z}_c^- \tag{57}$$

The coefficient μ_{TD}^- is denominated energy dissipation coefficient and it can be shown that the energy loss at touchdown is proportional to its squared value:

$$\Delta E_{TD} = E^+ - E^- = -\frac{mJ_l}{2(J_l + mr_0^2)}\mu_{TD}^2 \tag{58}$$

As a consequence of (58), a necessary condition to achieve energy-preserving passive gaits is:

$$\mu_{TD}^- = 0 \tag{59}$$

The lift-off transition occurs during stance when the leg extends back to its resting length r_0, whereas the touchdown transition occurs during flight when the foot's vertical position reaches zero. Note that the transitions not only determine a change of the dynamics but also cause jumps in some of the state velocities. In particular, at lift-off there is a jump in the leg radial extension velocity, which is always zero during flight due to the assumption of massless foot, while at touchdown all the velocities are subject to discontinuities with the exception of the energy lossless impact case, in which the energy dissipation coefficient is zero (59).

For the forward hopping control of the planar robot, two different approaches are described in the following sections. One approach is based on passive dynamics and exploits the stiffness of the hip, while the other directly applies torques to the hip and only uses the spring on the prismatic joint to store energy.

5.2 Forward hopping with body attitude control

5.2.1 Hopping controller based on the foot placement algorithm

The controller here described is inspired by the work of Raibert in [13]. Changes to the equations have been made in order to adapt it to the planar one-legged robot described in section 5.1, which is simpler than the one used in the original paper in which the controller was described. Firstly, the hypothesis of having the robot's center of mass located on the hip revolute joint simplifies the calculations necessary for the foot placement algorithm. Secondly, the actuation of the leg is here modeled as a direct force actuation in parallel with the spring, instead of modelling a spring displacement actuation.

In the entirety of section 5.2, the hip spring stiffness is set to zero, $K_h = 0$ Nm/rad. Instead of exploiting the spring's passive dynamics to swing the leg back and forth, the approach here described directly applies a torque to the leg through the hip actuator both during the flight phase and the stance phase.

The controller is decomposed into three separate tasks:

1. An energy regulator that achieves the desired hopping height by adjusting the linear spring actuator on the leg during the stance phase.

2. A controller active during the stance phase that uses the hip actuator to achieve a desired body attitude.

3. A controller active during the flight phase that uses the hip actuator to swing the leg in order to adjust the foot placement with respect to the center of gravity (CG) print[1] at touchdown. The purpose of this part is to regulate the forward horizontal velocity of the center of mass of the robot by exploiting the gravity's net tipping moment during the subsequent stance phase.

The vertical control task can be achieved in the same way as described for the one-dimensional case described in section 3. However, instead of considering the total energy of the system for the purpose of energy regulation, only the "vertical energy"

[1] The terminology CG print hereby indicates the projection of the center of gravity on the ground, as described in [13].

E_v is used. This choice is based on the fact that the horizontal kinetic energy remains unchanged from lift-off to the apex and does not affect the apex height and on the simplifying assumption that the rotational kinetic energies are negligible, which is especially accurate in the case of movements that are primarily vertical.

Thus, the force to be applied by the linear leg actuator during stance is given by:

$$f = -\gamma_E \cdot (E_v - E_{v,des}) \cdot \dot{r} \tag{60}$$

where γ_E is a positive gain and the vertical energy E_v is computed as:

$$E_v = mgz_c + \frac{1}{2}m\dot{z}_c^2 + \frac{1}{2}K_r\left(r - r_0 - \frac{mg}{K_r}\right)^2 \tag{61}$$

where in the computation of the elastic energy the spring preload is considered. Similarly to the one-dimensional case described in section 3, the vertical controller (60) can be modified with a saturation.

The desired vertical energy is simply given by the gravitational potential energy at the desired apex height:

$$E_{v,des} = mgz_{c,des} \tag{62}$$

The second task controls the body's attitude during stance by applying a simple proportional-derivative feedback torque control on the hip actuator to lead the body angle ϕ to a desired value ϕ_{des}:

$$\tau = K_\phi \cdot (\phi_{des} - \phi) - K_{\dot{\phi}} \cdot (\dot{\phi}) \tag{63}$$

where K_ϕ and $K_{\dot{\phi}}$ are positive gains.

Finally, the forward velocity control is obtained by placing the foot with respect to center of the CG print, also denoted as *neutral point*. Firstly, the desired foot position $\bar{x}_{des}$ with respect to the robot's center of mass can be computed as:

$$\bar{x}_{des} = \bar{x}_{err} + \bar{x}_{CG} = K_{\dot{x}} \cdot (\dot{x}_c - \dot{x}_{c,des}) + \bar{x}_{CG} \tag{64}$$

where $\bar{x}_{err}$ represents the deviation or error from the neutral point $\bar{x}_{CG}$ and it is computed with a proportional feedback term on the desired horizontal velocity of the center of mass of the robot $\dot{x}_{c,des}$ through the positive gain $K_{\dot{x}}$.

The center of the CG print in (64) can be estimated by measuring the time duration of the previous stance phase and the horizontal velocity during flight with the following expression:

$$\bar{x}_{CG} = \frac{1}{2}\dot{x}_c T_s \tag{65}$$

Note that in the equations (64) and (65), the bar notation on the variables $\bar{x}_{CG}$ and $\bar{x}_{des}$ signifies that the coordinates are expressed with respect to the robot's center of mass (i.e. $\bar{x}_{CG} = x_{CG} - x_c$ and $\bar{x}_{des} = x_{des} - x_c$).

By applying the inverse kinematics of the robot, the desired leg angle at touchdown can be easily computed as:

$$\alpha_{TD,des} = \arcsin\left(\frac{\bar{x}_{des}}{r}\right) \tag{66}$$

The leg is swung forward or backward during flight through a simple proportional-derivative feedback torque control on the hip actuator to lead the leg angle α to the value $\alpha_{TD,des}$ computed in (66):

$$\tau = K_\alpha \cdot (\alpha_{TD,des} - \alpha) - K_{\dot{\alpha}} \cdot (\dot{\alpha}) \tag{67}$$

where K_α and $K_{\dot{\alpha}}$ are positive gains.
Figure 47 clarifies the positioning of the foot with respect to the CG print.

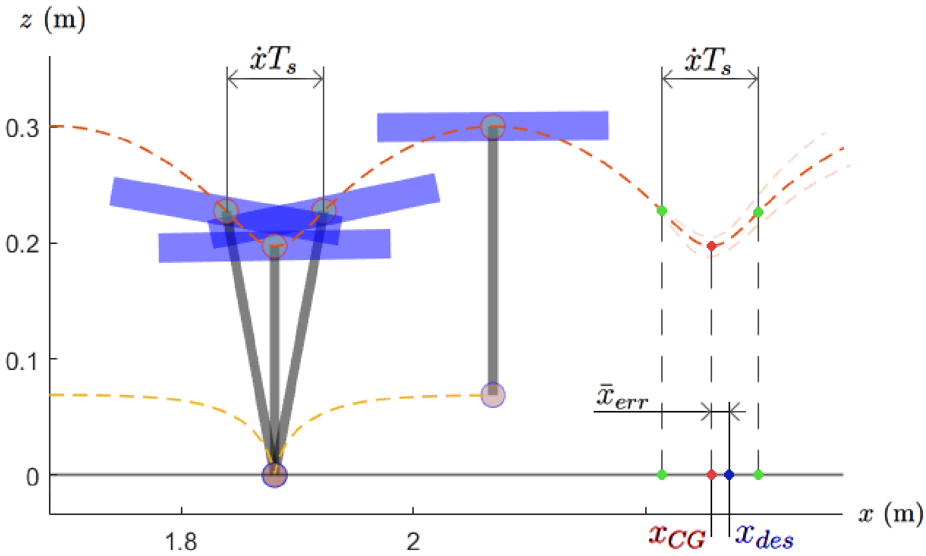


Figure 47: Drawing showing the variables used for the foot placement algorithm. The CG print is the segment of the ground between the two green dots and its middle point is denoted with x_{CG}.

5.2.2 Simulation results

5.2.2.1 Hopping at constant velocity

In this section, the simulation results of hopping with constant velocity are reported. The gains for the controller and the desired quantities were set as reported in Table 8, and the robot was initialized in stance phase with the initial conditions reported in Table 9.

Variable	Description
γ_E	30 Ns/Jm
$K_{\dot{x}}$	0.008 s
K_ϕ	6 Nm/rad
$K_{\dot{\phi}}$	0.9 Nms/rad
K_α	50 Nm/rad
$K_{\dot{\alpha}}$	3 Nms/rad
$z_{c,des}$	0.3 m
ϕ_{des}	0 rad
$\dot{x}_{c,des}$	1.3 m/s

Table 8: List of numerical values assigned to the feedback gains.

Variable	Description
α_0	0.1672 rad
$\dot{\alpha}_0$	-4.2451 rad/s
ϕ_0	-0.2363 rad
$\dot{\phi}_0$	-0.0023 rad/s
r_0	0.2312m
$\dot{r}_0$	-1.3493 m/s

Table 9: Initial conditions for the simulation of hopping at constant speed.

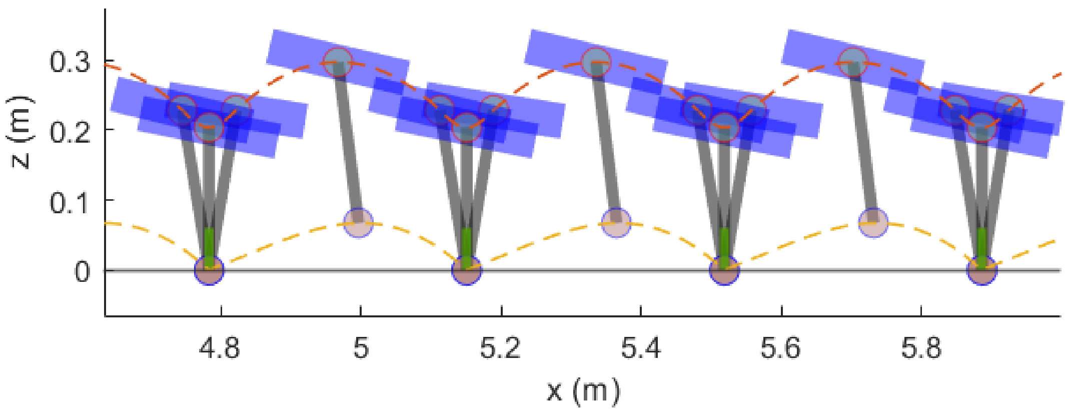


Figure 48: Trajectory of the robot performing running at constant speed with the foot placement algorithm described by Raibert.

Figure 48 reports the trajectory performed by the robot in space along with some snapshots of the model configuration. From the trajectories the asymmetry of the movement is evident: the body remains tilted forward during the locomotion and the leg is not vertical at the apexes. In this example, the desired apex height of $z_{c,des} = 0.3$ m and forward velocity $\dot{x}_{c,des} = 1.3$ m/s are reached, but the body attitude during stance is not exactly zero as prescribed. Different trajectories can be obtained by changing the gains and the desired set-points.

Figure 49 shows the plots of the states of the model. The areas with gray background indicate the stance phases. Discontinuities are present in the velocities $\dot{x}_c$, $\dot{z}_c$, $\dot{\alpha}$ and $\dot{\phi}$ at every touchdown event as the impacts occur with a non-zero energy dissipation coefficient $\mu_{TD}^{-} \neq 0$.

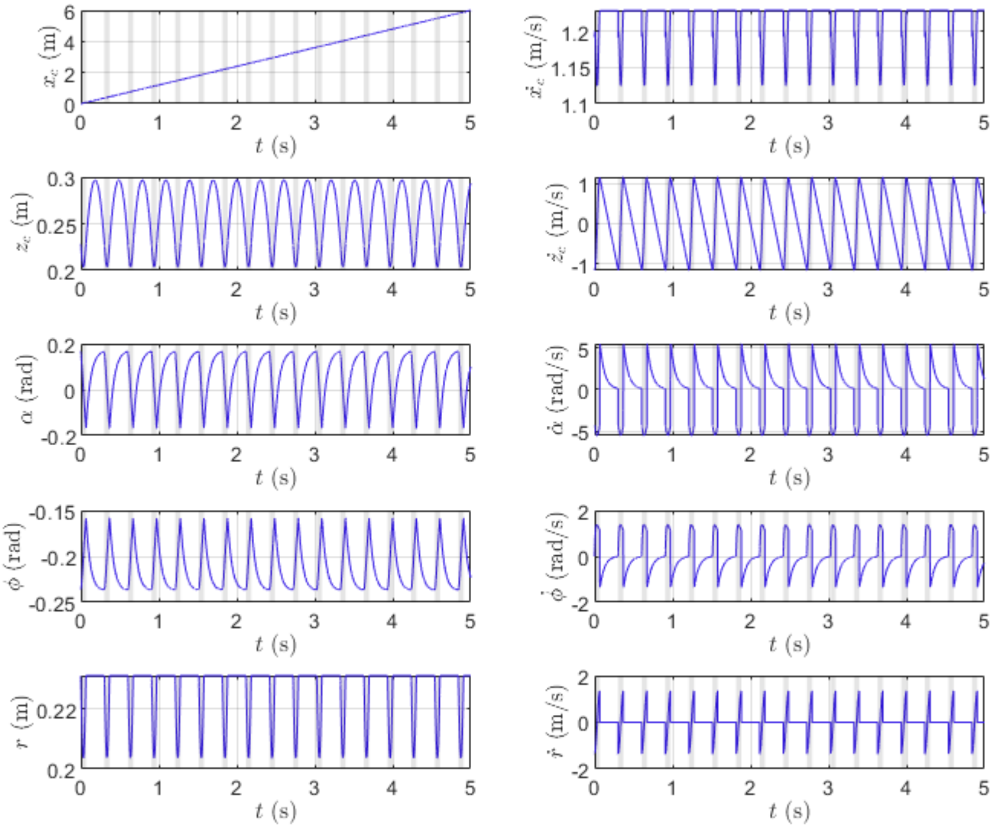


Figure 49: States of the robot while running at speed $\dot{x}_{c,des} = 1.3$ m/s. The desired apex height of $z_{c,des} = 0.3$ m and forward velocity $\dot{x}_{c,des} = 1.3$ m/s are reached. The body attitude ϕ does not reach the desired set-point of zero but remains negative instead.

The model was initialized to be already on a converged closed-loop gait. The periodicity of the gait is highlighted in Figure 50, where the states are plotted in the phase plane. The orbits are closed as the states return to their previous value after the completion of a step period. Figure 51 shows the leg and body angles at touchdown and lift-off. The angles remain at fixed points because the simulated closed-loop gait is periodic.

The actuation force and torque are shown in Figure 52. The force f was limited to positive values through a saturation in the controller. The plot highlights the presence of high peaks in the torque control during stance.

Figure 53 shows the ground reaction forces during the hopping at constant speed, while Figure 54 shows the time evolution of the energy in the system.

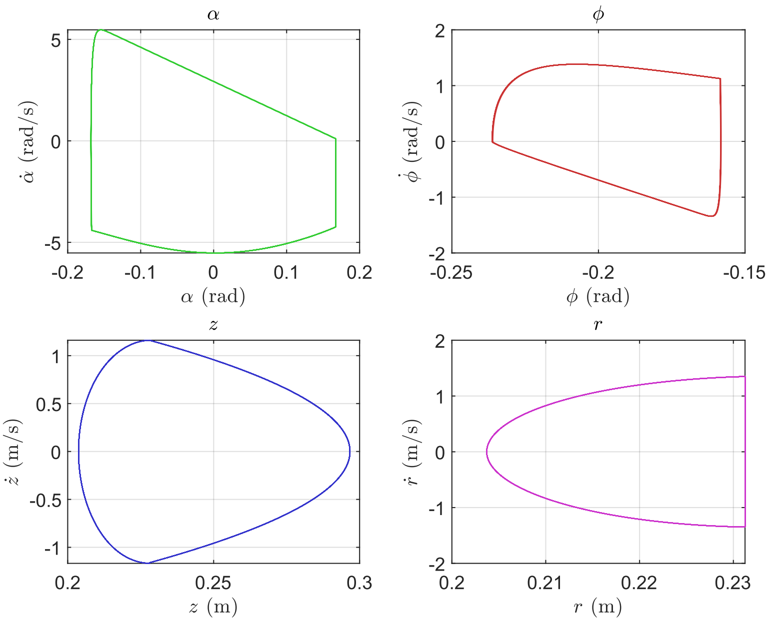


Figure 50: States plotted in the phase plane. The orbits are closed as the states return to their previous value after the completion of a step period. The pair $(x_c,\ \dot{x_c})$ was omitted as x_c is monotonically increasing and thus non-periodic.

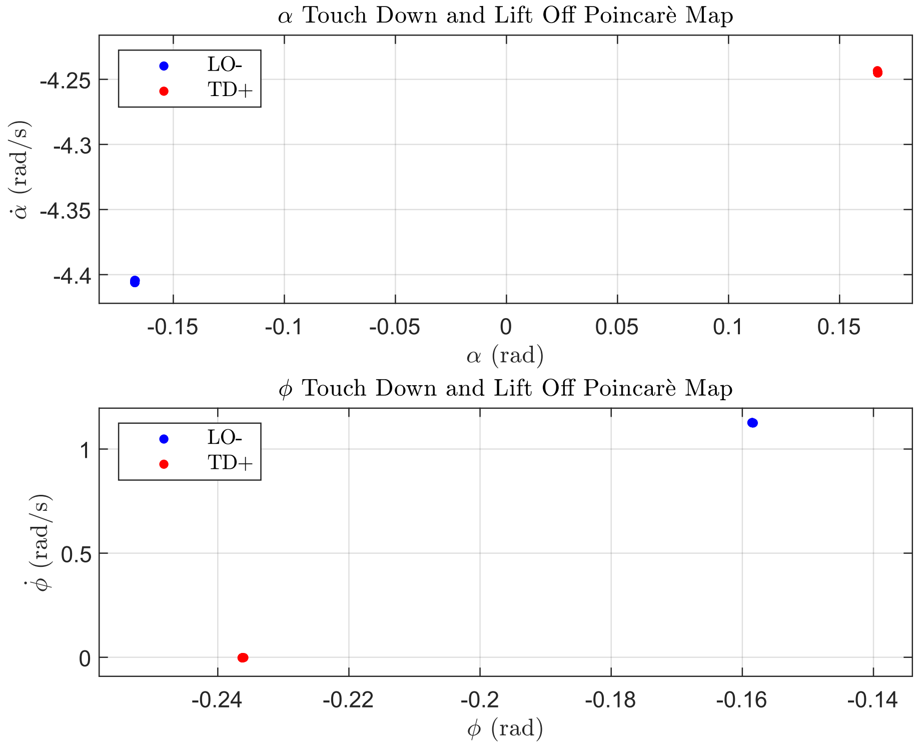


Figure 51: Leg and body angles at touchdown and lift-off. The angles remain at fixed points because the simulated closed-loop gait is periodic.

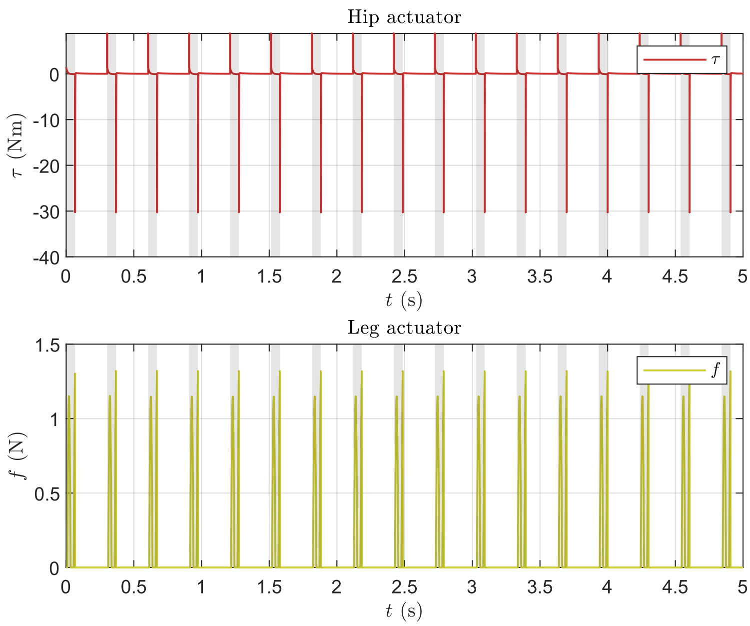


Figure 52: Actuation force and torque. The force f was limited to positive values through a saturation in the controller. The body attitude controller requires high peaks at the beginning and ending of each stance phase.

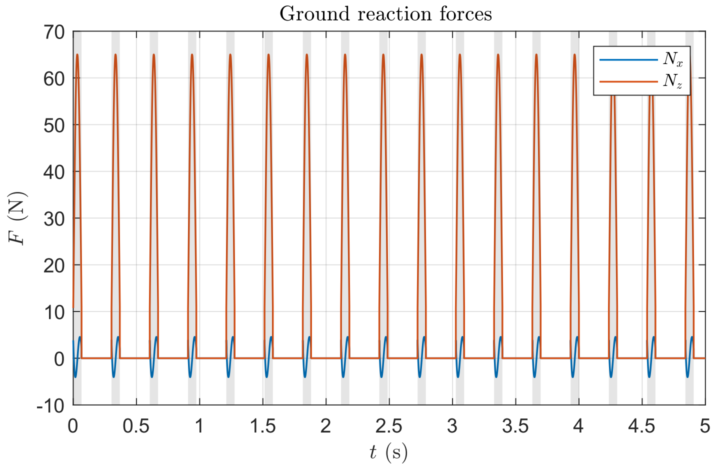


Figure 53: Ground reaction forces acting during the hopping at constant speed with the foot placement algorithm.

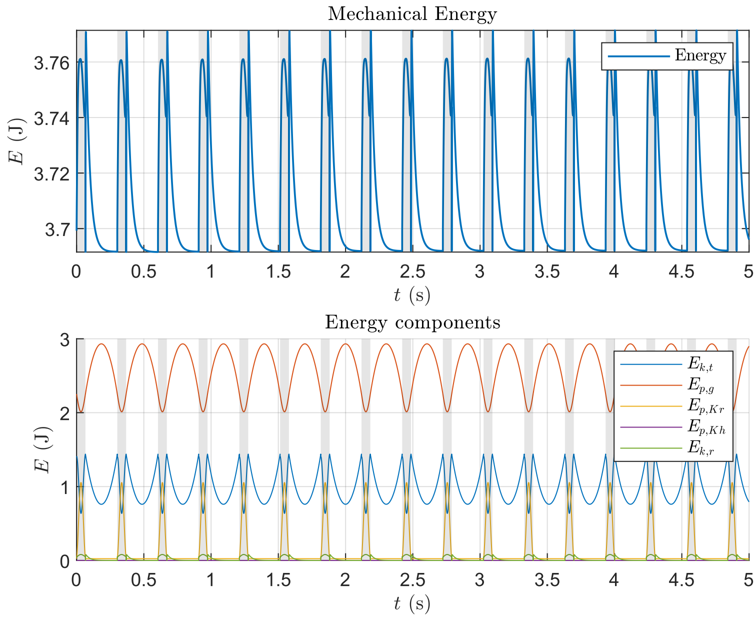


Figure 54: Energy time evolution during the hopping at constant speed with the foot placement algorithm. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy. Note that $E_{p,Kh} = 0$ as K_h is set to zero in this section.

5.2.2.2 Hopping from point A to point B

In this section, the results of a simulation demonstrating the versatility of the locomotion controller based on the foot placement algorithm are shown. In the simulation, the robot is initialized from a standing position. It is then commanded to go reach the point $x_{c,des} = 1$ m. A control in the position can be achieved by setting the desired velocity depending on the distance from the desired point and saturating it with a maximum velocity value $\dot{x}_{c,max}$:

$$\dot{x}_{c,des} = \min(K_x \cdot (x_c - x_{c,des}),\ \dot{x}_{c,max}) \tag{68}$$

where K_x is a positive gain, as done in [13]. For the simulation, the control gains used were the same of Table 9 together with $K_x = 0.2s^{-1}$ and $\dot{x}_{c,max} = 1$ m/s. Higher values for K_x and $\dot{x}_{c,max}$ will lead to a faster approach to the desired set-point but can also cause overshoot.

Figure 55 shows how the states of the robot evolved during the simulation. The robot initiates hopping by starting from a resting position, accelerates till reaching a velocity of about 0.13 m/s and then decelerates to stop at $x = 1$ m, where it keeps hopping vertically.

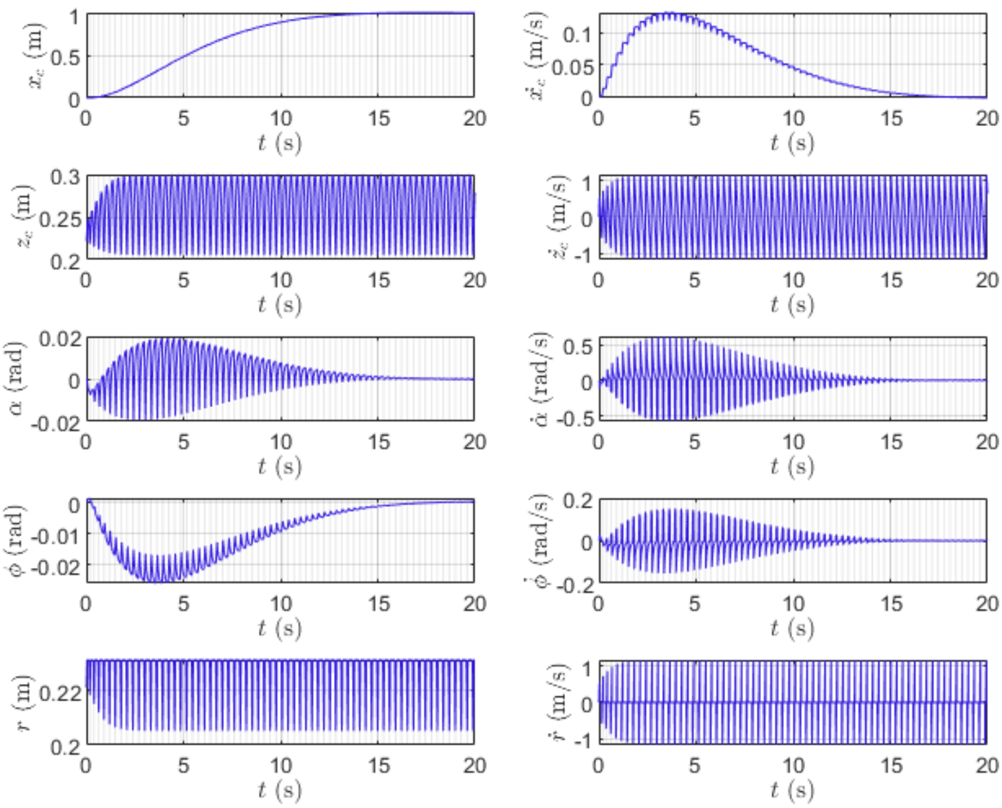


Figure 55: States of the robot as it travels from $x = 0$ m to $x = 1$ m.

Figure 56 shows the states in the phase plane, while Figure 57 shows the leg and body angles at touchdown and lift-off. The angles converge to fixed points in the origin after the robot reaches the desired set-point and hops vertically with $\alpha = 0$ and $\phi = 0$.

The actuation force and torque are shown in Figure 58. The hopping initialization requires relatively high actuation forces f. After the robot reaches $x = 1$ m, f converges to zero as no further energy losses are present in the vertical hopping. The hip torque reaches values of more than 3 Nm during movement, even though the maximum speed was only about 0.13 m/s.

Figure 59 shows the energy of the robot. It rises from a minimum corresponding to the resting position and converges to a constant value when the robot reaches $x = 1$ m.

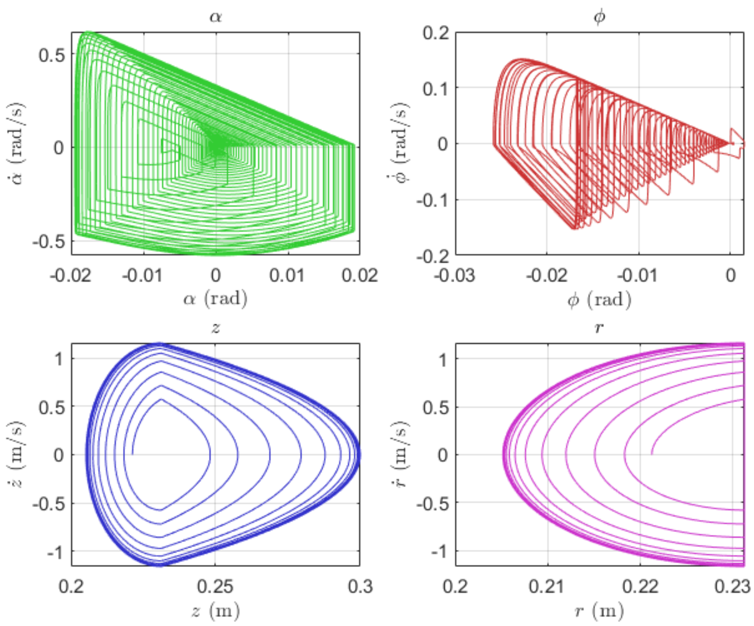


Figure 56: States of the robot in the phase plane as it travels from $x = 0$ m to $x = 1$ m.

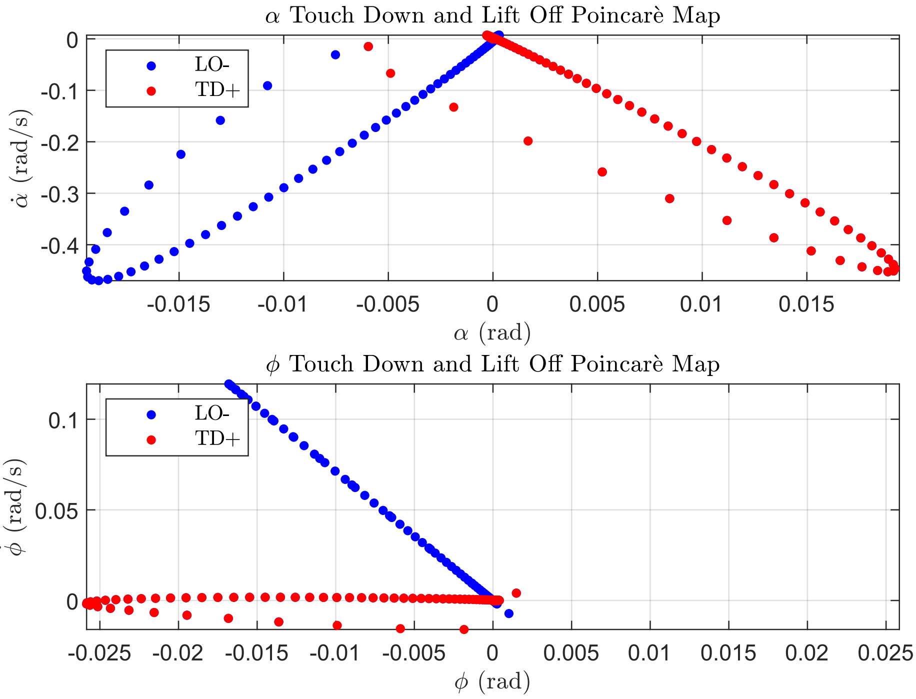


Figure 57: Leg and body angles at touchdown and lift-off. The angles converge to fixed points in the origin after the robot reaches the desired set-point and hops vertically.

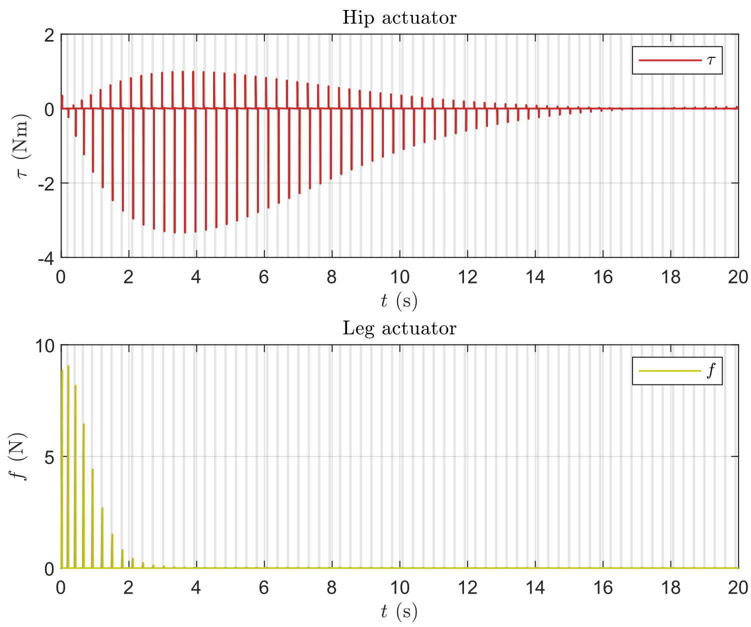


Figure 58: Actuation force and torque. The force f was limited to positive values through a saturation in the controller. The hopping initialization requires relatively high actuation forces f.

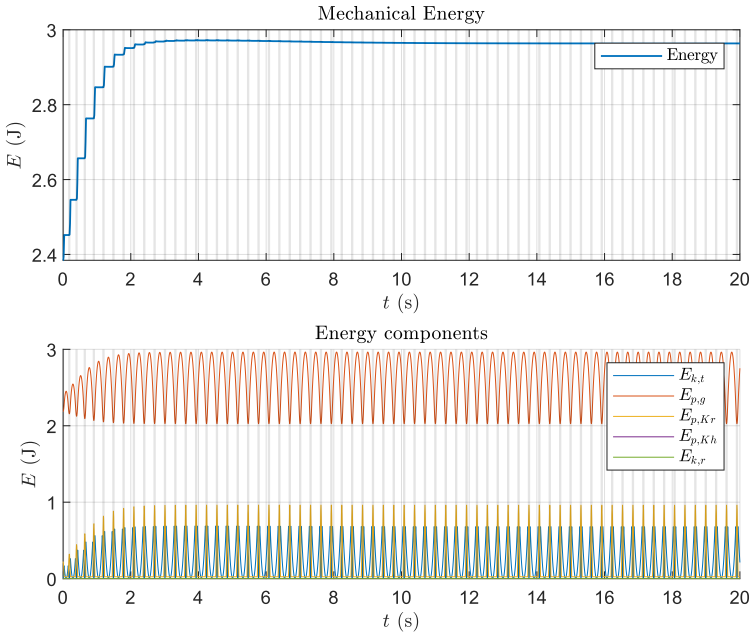


Figure 59: Energy of the robot. It converges to a constant value when the robot reaches. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy. Note that $E_{p,Kh} = 0$ as K_h is set to zero in this section.

5.3 Stabilized passive running

In this section, a control approach to stabilize the passive running of the planar one-legged robot with hip and body link is described. First, the passive gaits of the robot found through the search of fixed points of the Poincaré map are briefly analyzed. Then, the control approach, which is based on [14], is reviewed and extended with continuous actuation torques, to perform velocity tracking and to stabilize the robot in the presence of damping in the joints. Simulation results are presented and commented.

5.3.1 Initial conditions for passive running

The model of the robot here considered is the one described in section 5.1. Unlike in section 5.2, the hip compliance is enabled with $K_h = 1$ Nm/rad.

For the hopper to achieve passive running, it must be initialized with appropriate initial conditions. Hereby, the approach described for the initialization in order to obtain approximated periodic solution is derived from Hyon's and Emura's work [14].

Consider the case in which the robot is initialized immediately after the touchdown event, "TD+". The initial conditions can be parameterized with respect to an arbitrary starting leg angle α_0 and horizontal speed $\dot{x}_{c0}$. The equations used to initialize the model in this section are an approximation of periodic passive gaits. Moreover, even though a pair of initial states $(\dot{x}_{c0}, \alpha_0)$ is used as free parameters, there is only one value of α_0 that achieves a quasi-periodic gait and minimizes the state differences between the step k and $k+1$ for a given value of $\dot{x}_{c0}$, as it will be discussed in the last part of this section.

First of all, the initial leg length is equal to its resting length, r_0. Let now Ω_h be the natural oscillation frequency of the hip spring defined as:

$$\Omega_h = \sqrt{K_h \left(\frac{1}{J_b} + \frac{1}{J_l} \right)} \tag{69}$$

Ω_h is the frequency, expressed in rad/s, at which body and leg counter-oscillate in their non-trivial mode of vibration (the trivial case is when body and leg rotate together at the same constant angular speed).

It is now assumed that the period of a whole step is equal to the period of the hip natural oscillation, so that the leg swing can be synchronized with each step while the body counter-oscillates. This hypothesis yields:

$$\Omega_h \cdot T_{step} = \Omega_h \cdot (T_f + T_s) = 2\pi \tag{70}$$

where T_{step} is the step period and T_f and T_s are the flight time duration and the stance time duration respectively.

The stance duration can be approximated with half the period of the natural spring oscillation along the prismatic joint of the leg when the robot is in stance, which is accurate in the case near the vertical hopping movement, where $\alpha \approx 0$:

$$T_s \approx \frac{\pi}{\Omega_r} = \frac{\pi}{\sqrt{K_r/m}} \tag{71}$$

where Ω_r is the natural frequency of the linear mass-spring oscillator of the leg spring and the robot's mass.

From 70 the time of flight can be computed as:

$$T_f = \frac{2\pi}{\Omega_h} - T_s = \pi \left(\frac{2}{\Omega_h} - \frac{1}{\Omega_r} \right) \tag{72}$$

Since in the passive gaits the vertical velocity at touchdown is assumed to be equal in module and opposite in sign with respect the one at lift-off, the center of mass of the robot travels along a symmetric arc of a parabola during flight and T_f can be also computed as:

$$T_f = 2 \cdot \frac{\dot{z}_{lo}}{g} = -2 \cdot \frac{\dot{z}_{td}}{g} = -2 \cdot \frac{\dot{z}_{c0}}{g} \tag{73}$$

Combining (72) and (73), the initial vertical velocity at touchdown can be computed as:

$$\dot{z}_{c0} = -\frac{g}{2} \cdot T_f = -g\frac{\pi}{2} \left(\frac{2}{\Omega_h} - \frac{1}{\Omega_r} \right) \tag{74}$$

Imposing now the condition of energy lossless impact at touchdown (59), the initial angular velocity of the leg can be computed:

$$\dot{\alpha}_0 = -\frac{1}{r_0}(\dot{x}_0 \cos(\alpha_0) + \dot{z}_0 \sin(\alpha_0)) \tag{75}$$

The body and the leg counter-oscillate with the hip natural frequency Ω_h, so the initial body orientation can be computed from α_0 as:

$$\phi_0 = -\frac{J_l}{J_b}\alpha_0 \tag{76}$$

Similarly, for the angular velocities:

$$\dot{\phi}_0 = -\frac{J_l}{J_b}\dot{\alpha}_0 \tag{77}$$

The condition (77) also implies a zero total angular momentum during flight.

The remaining initial state to be computed is the initial radial velocity, which is simply given by:

$$\dot{r}_0 = \dot{z}_{c0} \cos(\alpha_0) - \dot{x}_{c0} \sin(\alpha_0). \tag{78}$$

It is important to note that due to the approximation of the stance duration in (71), the initialization of the states described in this section does not give exact solutions of passive periodic gaits. This was also empirically confirmed by searching numerically for couples of ($\dot{x}_{c0}$, α_0) that yield exact periodic solutions with the above approximated parametrization: no exact fixed points of the Poincaré map could be found. Nonetheless, this method to compute approximated periodic solutions is very useful to initialize the model close to actual periodic without needing to search for periodic gaits numerically, so that a convergence to the actual periodic solution can be achieved through the controller described in section 5.3.2.

Figure 60 shows the pairs of ($\dot{x}_{c0}$, α_0) that are the closest to periodic gaits with the initial conditions parametrization described in this section, together with the error on the states after one step. The values are also compared to those obtained by numerically searching for periodic gait by optimizing on the ($\dot{x}_{c0}$, α_0, $\dot{z}_{c0}$) variable set and using the parametrization described in this section for the remaining variables. The touchdown event was selected as the Poincaré section to find fixed points.

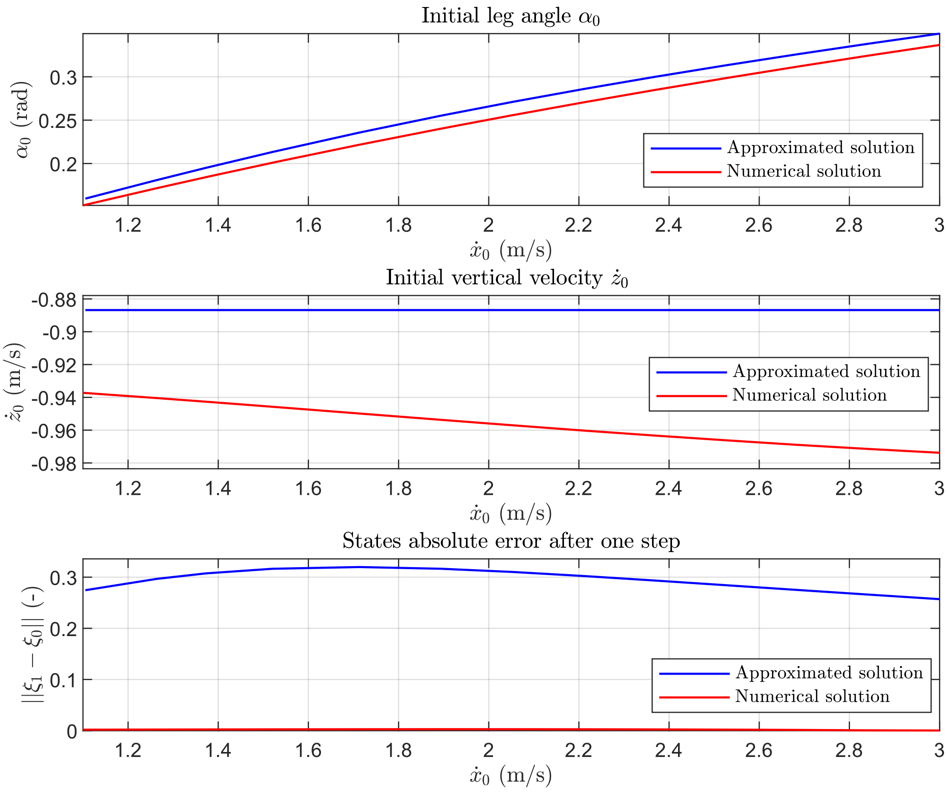


Figure 60: Comparison between the initial conditions using the gait search algorithm based on Poincaré maps: the blue curves were obtained by optimizing on the free parameters ($\dot{x}_{c0}$, α_0), whereas the red curves are the result of the optimization on ($\dot{x}_{c0}$, α_0, $\dot{z}_{c0}$). In the lower plot, the term $||\xi_1 - \xi_0||$ indicates the norm of the difference between the states at the initial touchdown event and the subsequent one. Observe how the numerical solutions without the approximation lead to a much smaller (almost zero) error in the states after one step.

The numerical solutions allow to avoid the approximation introduced in (71) and lead to significantly smaller errors in the states after one step, as it can be seen in the lower plot of Figure 60. This is also confirmed by simulating the gaits: those obtained with the approximated method fall after a few steps. Figure 61 compares two passive gaits with almost equivalent running speed, one obtained with the approximated method ($\dot{x}_{c0}$ = 1.520 m/s, α_0 = 0.213 rad, $\dot{z}_{c0}$ = −0.887 m/s) and the other numerically ($\dot{x}_0$ = 1.530 m/s, α_0 = 0.202 rad, $\dot{z}_{c0}$ = −0.946 m/s). Observe how the approximated solution falls after only five steps. Note that also the more accurate passive gait eventually falls due the disturbances caused by numerical integration and small errors in the initial conditions, however this only happens after 27 steps. The fall is caused by the inherent instability of the system.

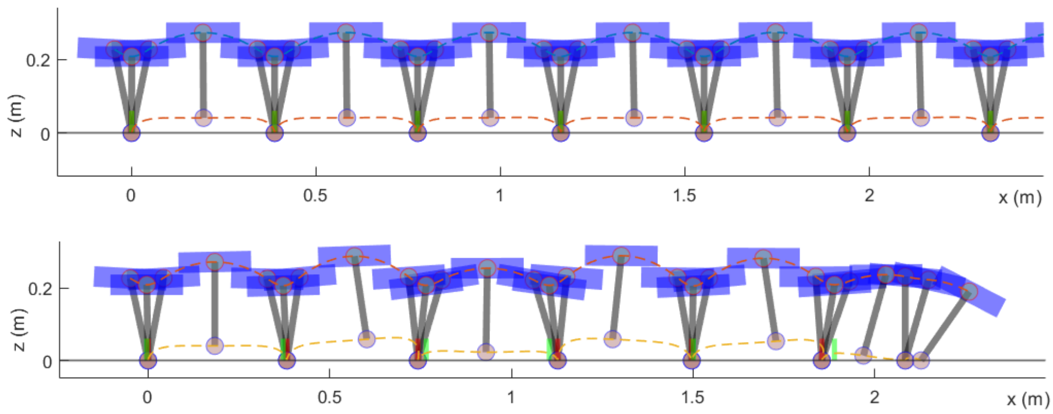


Figure 61: Comparison between an approximated gait (bottom drawing) and one obtained through numerical optimization (upper drawing).

The stability of the periodic gaits can be assessed by computing the eigenvalues of the linearized Poincaré map. Figure 62 shows the root loci of the characteristic multipliers of the periodic gaits with horizontal flight velocity going from 0 m/s to 3 m/s. At least one of the eigenvalues is always outside the unit circle, meaning that the periodic gaits are not stable.

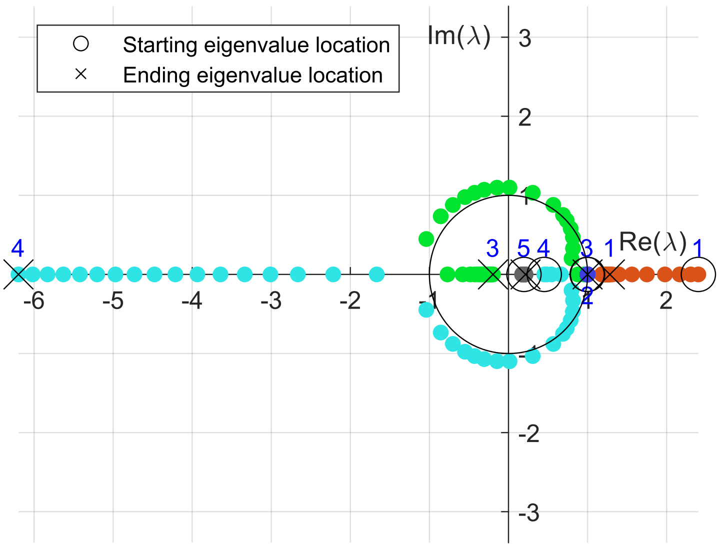


Figure 62: Root loci of the characteristic multipliers of the periodic gaits with horizontal flight velocity going from 0 m/s to 3 m/s. At least one of the eigenvalues is always outside the unit circle.

In the next section, a controller capable of stabilizing such passive gaits is described.

5.3.2 Non-dissipative touchdown control

The controller chosen for this study is the non-dissipative touchdown control described by Hyon and Emura in [14]. It applies a piece-wise torque at the hip motor of the robot during flight to achieve a desired touchdown leg angle $\alpha_{td,des}$ and angular velocity $\dot{\alpha}_{td,des}$. The main idea of the controller is to compute the leg touchdown angles based on the previous lift-off angles. This is achieved with equation (79), which computes the average of two past lift-off angles. The positive integer variable k is here used to refer to the k-th step in the running of the robot, whereas the positive integer parameter p is used to select different steps in the past for the averaging of the lift-off angles.

$$\alpha_{td,des}(k) = \begin{cases} -\dfrac{1}{2}(\alpha_{lo}(k) + \alpha_{lo}(k-p)), & \text{if } k > p \\ -\alpha_{lo}(k) \quad \text{otherwise} \end{cases} \tag{79}$$

Three interesting cases of equation (79) can be obtained by changing the value of the parameter p:

1. $p = 0$: The desired touchdown angle is equal to the mirrored previous lift-off angle, $\alpha_{td,des}(k) = -\alpha_{lo}(k)$.
2. $p = 1$: The desired touchdown angle is the average of the previous two lift-off angles. This is equivalent to applying the bisection method and makes the system converge to periodic solutions, as it will be shown in section 5.3.3.
3. $p = 2$: The desired touchdown angle is the average of the previous lift-off angle and the one of two jumps earlier. This approach converges to quasi-periodic gaits in which the steps lengths alternate between two values and the pseudo-period consists of two running steps instead of one (also called two-periodic gaits or 'wobbling' gaits).

Figure 63 clarifies the notation with the index k.

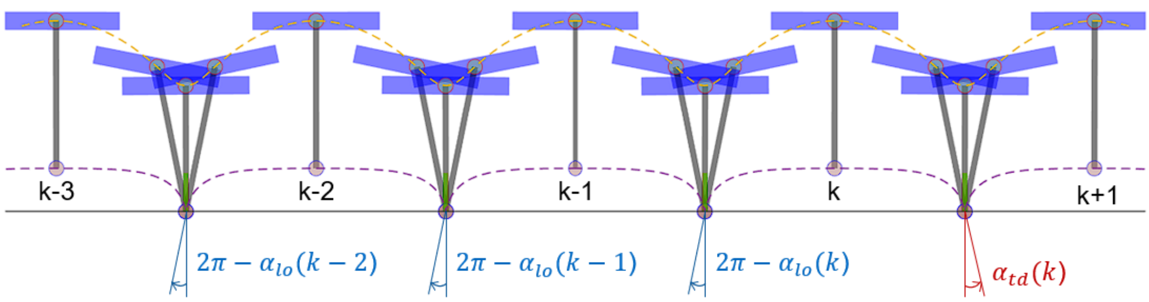


Figure 63: Snapshots taken from a running gait of the one-legged robot. As it can be seen in the drawing, the angles $\alpha_{lo}(k)$ and $\alpha_{td}(k)$ refers to the lift-off immediately before and the touchdown immediately after the k-th jump.

Once the desired touchdown angle is selected, the desired leg angular velocity can be computed so that the energy lost at touchdown is zero, which is a necessary condition to achieve passive running. To do so, the energy dissipation coefficient must be zero:

$$\mu_{TD}^{-} = 0 \tag{80}$$

Which implies:

$$\dot{\alpha}_{td,des} = -\frac{1}{r_0}(\dot{x}_{td-}\cos\alpha_{td} + \dot{z}_{td-}\sin\alpha_{td}) \tag{81}$$

Now two cases will be analyzed: the simple case in which $\alpha_{td,des}(k) = -\alpha_{lo}(k)$ ($p = 0$ or $p = 1$ after convergence), in which the flight phase is symmetrical and the general case in which $\alpha_{td,des}(k) \neq -\alpha_{lo}(k)$ and the flight phase is asymmetrical. Note that the case $\alpha_{td,des}(k) \neq -\alpha_{lo}(k)$ is only relevant when the robot has not converged yet to a periodic gait and is still adapting the desired touchdown angles by computing averages over the previous lift-off angles.

The equations here described assume that there is enough clearance to swing the leg forward without causing the foot to stub on the ground.

In the case in which the flight phase is symmetrical, the time of flight of a step is given by:

$$T_f = 2 \cdot \frac{\dot{z}_{lo}}{g} \tag{82}$$

and the following relations hold:

$$\dot{z}_{td-} = -\dot{z}_{lo}, \quad \dot{x}_{td-} = \dot{x}_{lo} \tag{83}$$

By substituting the relations (83) in (81), the desired leg angular velocity for the symmetrical flight case at touchdown is equal to the one at lift-off:

$$\dot{\alpha}_{td,des} = \dot{\alpha}_{lo} \tag{84}$$

In the more general case $\alpha_{td,des}(k) \neq -\alpha_{lo}(k)$, the time of flight can be computed by adding a Δt with respect to the expression in (82):

$$T_f^* = 2 \cdot \frac{\dot{z}_{lo}}{g} + \Delta t \tag{85}$$

where the quantities in the asymmetric case are denoted with the "*" superscript. Δt is computed as:

$$\begin{aligned} \Delta t = \frac{1}{g}\left(-\dot{z}_{lo} + \sqrt{\dot{z}_{lo}^2 - 2g\Delta z_c}\right) = \\ = \frac{1}{g}\left(-\dot{z}_{lo} + \sqrt{\dot{z}_{lo}^2 - 2gr_0(\cos(\alpha_{td,des}) - \cos(\alpha_{lo}))}\right) \end{aligned} \tag{86}$$

Here, Δt and Δz are the differences in time and fallen height with respect to the symmetric flight case, as shown in Figure 64.

In the general case:

$$\dot{z}_{td-}^* = -\dot{z}_{lo} - g\Delta t, \quad \dot{x}_{td-}^* = \dot{x}_{lo} \tag{87}$$

which, when substituted in (81), yields the general equation to compute the angular velocity from the states at lift-off:

$$\dot{\alpha}^*_{td,des} = -\frac{1}{r_0}(\dot{x}_{lo}\cos\alpha_{td,des} - (\dot{z}_{lo} + g\Delta t)\sin\alpha_{td,des}) \tag{88}$$

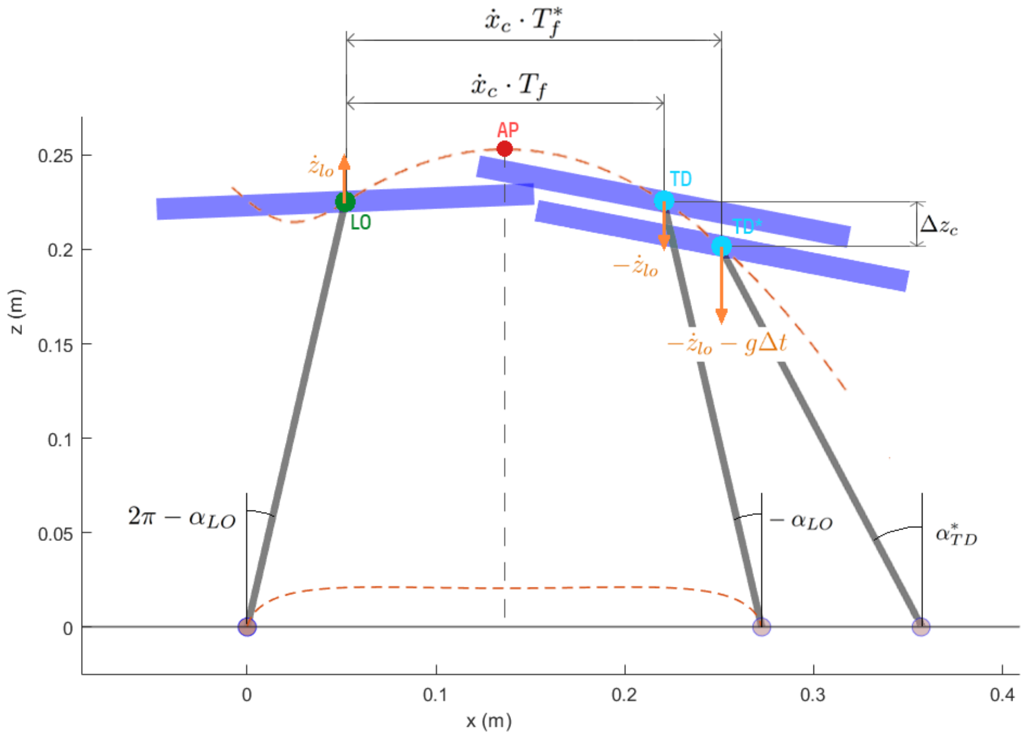


Figure 64: Comparison between the symmetric case $\alpha_{td,des}(k) = -\alpha_{lo}(k)$ and the asymmetric case $\alpha_{td,des}(k) \neq -\alpha_{lo}(k)$.

In practice, in the controller the equation (88) can be always used to compute the desired angular velocity while the expression (85) can be used to compute the flight time at lift-off. Once the desired touchdown leg angle and angular velocity are determined at lift-off, exact non linear dynamics can be used during flight to achieve them through a "bang-bang" piece-wise torque control. A constant torque τ_1 is applied for half of the flight time and then a constant torque τ_2 is applied for the remaining part of the flight. The torque profile used during flight is represented in Figure 65. The values for τ_1 and τ_2 can be computed through the formulas reported in the Appendix B.1.

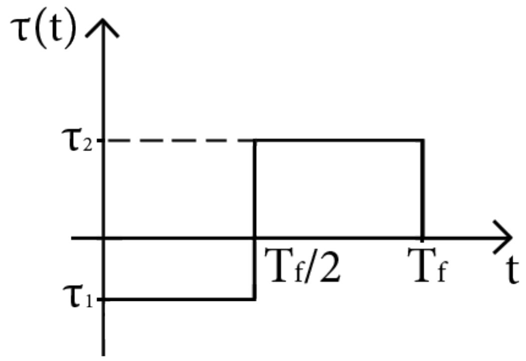


Figure 65: Piece-wise constant torque profile applied during flight. τ_1 and τ_2 can both be either positive or negative.

The equations described so far allow to achieve periodic running gaits, in particular by using (79) and (88) and applying the necessary torques during flight. The gaits thus obtained are in general not passive. In order to obtain a passive gait, the control effort must converge to zero. One possible approach is to adapt the hip natural oscillation frequency:

$$\Omega_h(k) = \begin{cases} \Omega_h(k-1) + \gamma_\Omega \cdot \Delta\tau(k-1) & \text{if } k > 1 \\ \sqrt{K_h \left(\dfrac{1}{J_b} + \dfrac{1}{J_l}\right)} & \text{otherwise} \end{cases} \tag{89}$$

where γ_Ω is a positive gain and $\Delta\tau(k)$ is computed as:

$$\Delta\tau(k) = \tau_2(k) - \tau_1(k) \tag{90}$$

Equation (89) can be implemented in practice by acting on J_b, J_l and/or K_h. For example, the stiffness K_h could be changed through the use of a variable stiffness spring, while the leg inertia J_l could be changed by stretching an eventual knee on the leg. In the simulations discussed hereafter, it is assumed that K_h can be arbitrarily controlled and thus the hip natural oscillation frequency can be adapted.

As an alternative to (89), an adaptive control can be implemented through a force on the linear actuator during the stance phases:

$$f(k) = \gamma_f \cdot sign(\Delta\tau(k-1)) \cdot \Delta\tau(k-1) \tag{91}$$

where γ_f is a positive gain.

5.3.3 Simulation results

The physical parameters of the robot were set according to Table 6.

5.3.3.1 Actuated periodic gait with p = 1

In this section, the simulation results without the spring adaptation law and with $p = 1$ are reported. The robot was initialized at touchdown with the parameterized initial conditions as described in section 5.3.1 with $\alpha_0 = 0.155$ rad and $\dot{x}_{c0} = 1.3$ m/s. The results are shown and commented in Figures 66 - 71. As highlighted by the Poincaré map images at touchdown and lift-off of Figure 69 and the states in the phase plane in Figure 68, the robot converges to a periodic gait. However such gait is not passive as it can be deduced by the non-zero actuation effort in Figure 70 and the non-constant energy of the robot in Figure 67.

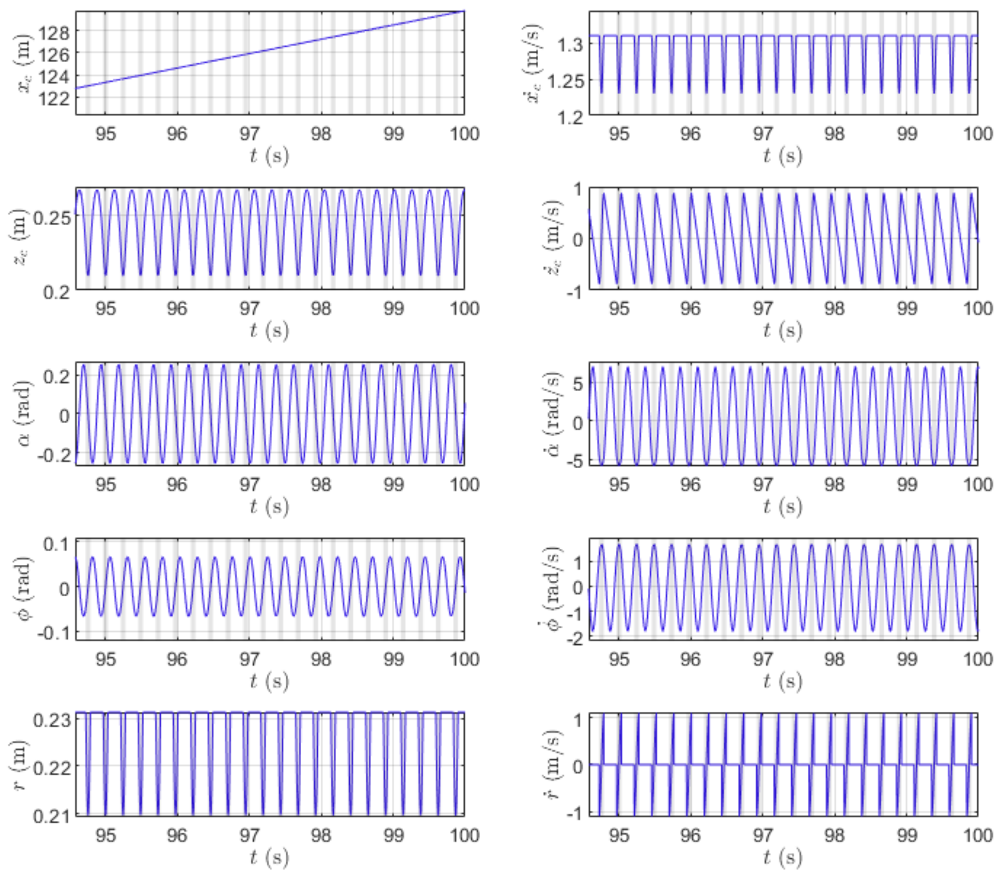


Figure 66: States of the robot during the last part of the simulation. In this time frame the robot has already converged to a periodic actuated gait.

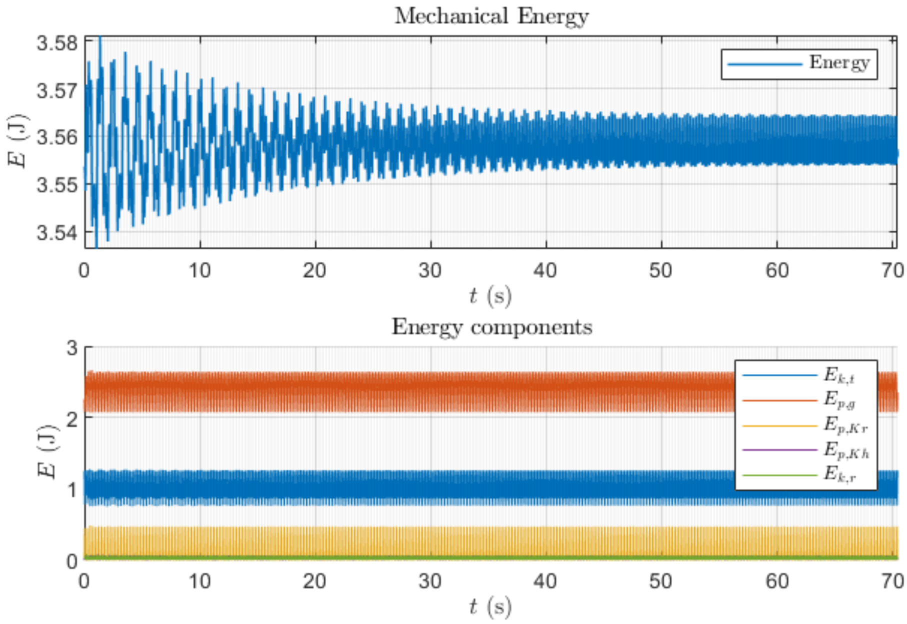


Figure 67: Energy of the robot. The total energy oscillates after convergence to a periodic actuated gait. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy.

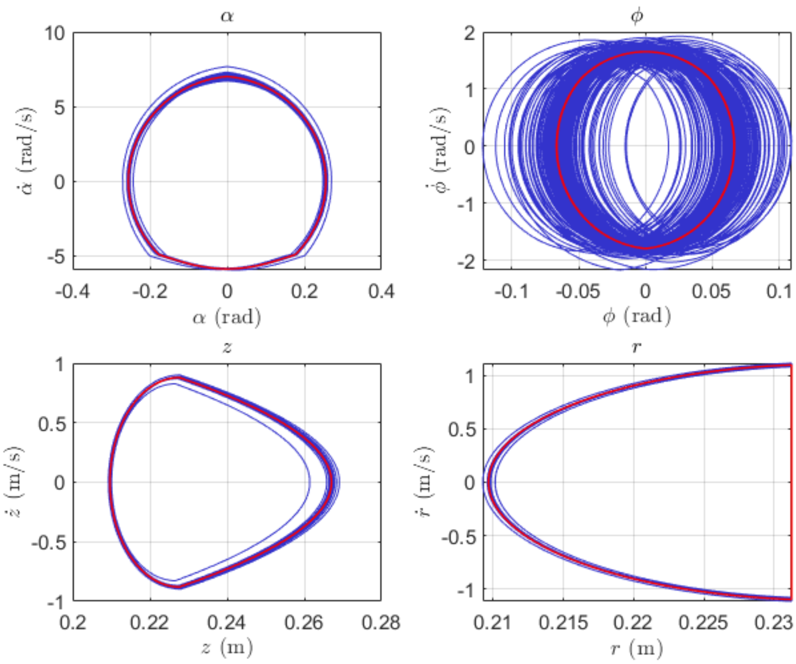


Figure 68: State trajectories in the phase plane. The red lines indicate the orbits after convergence to a periodic gait.

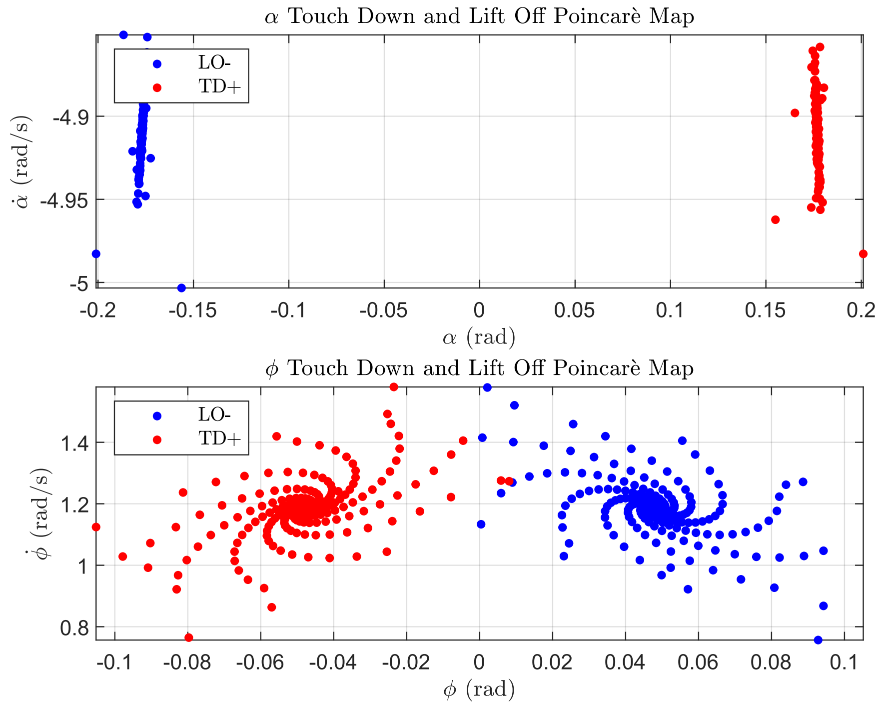


Figure 69: Poincaré maps of the robot angles at the events of touchdown and lift-off. The angles converge to fixed points.

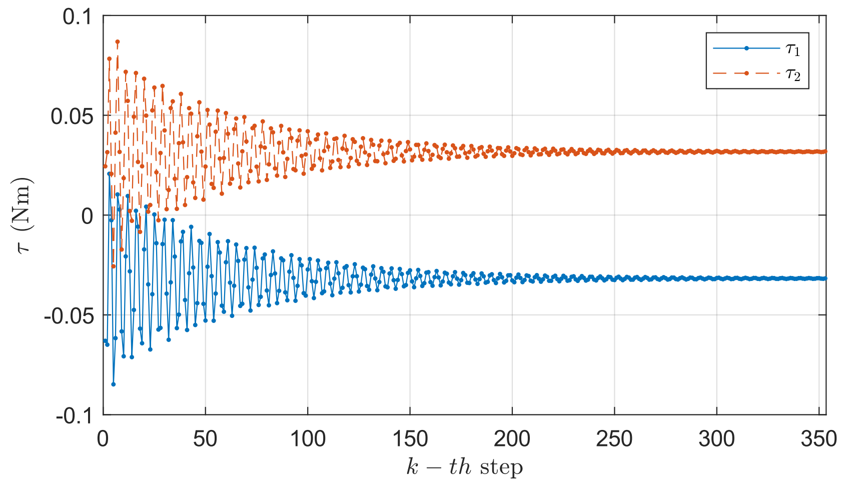


Figure 70: Control effort represented by τ_1 and τ_2. Both values eventually converge to a constant.

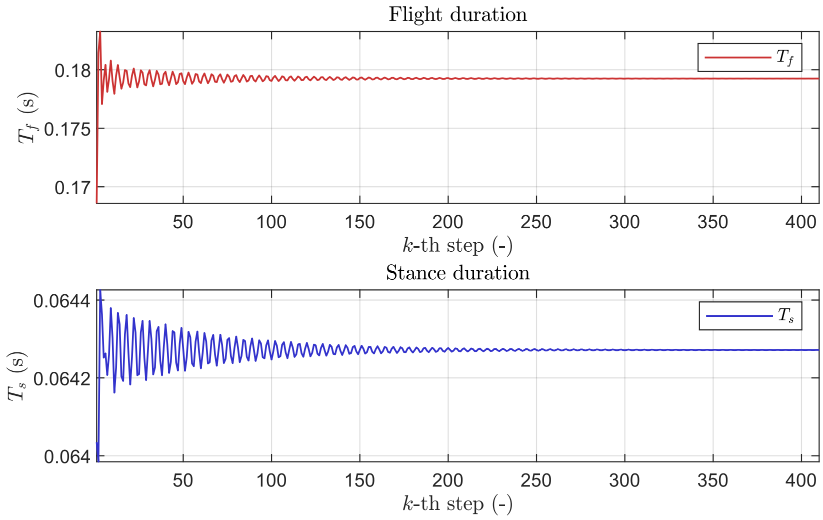


Figure 71: Stance and flight times. Both values eventually converge to a constant.

5.3.3.2 Actuated two-periodic gait with p = 2

In this section, the simulation results without the spring adaptation law and with p = 2 are reported. The robot was initialized at touchdown with the parameterized initial conditions as described in section 5.3.1 with $\alpha_0 = 0.155$ rad and $\dot{x}_{c0} = 1.5$ m/s. The results are shown and commented in Figures 72 - 78. The Poincaré map images at touchdown and lift-off in Figure 76 show how the angles converge to two fixed points as the robot converges to a two-periodic gait. In the phase plane in Figure 75, the states converge to orbits that circle around the origin two times before closing themselves.

The gait is not passive and both τ_1 and τ_2 oscillate between two values after the two-periodic gait is reached, see Figure 78. Interestingly, Figure 78 shows how the stance time converges to a fixed value while the flight time oscillates between two.

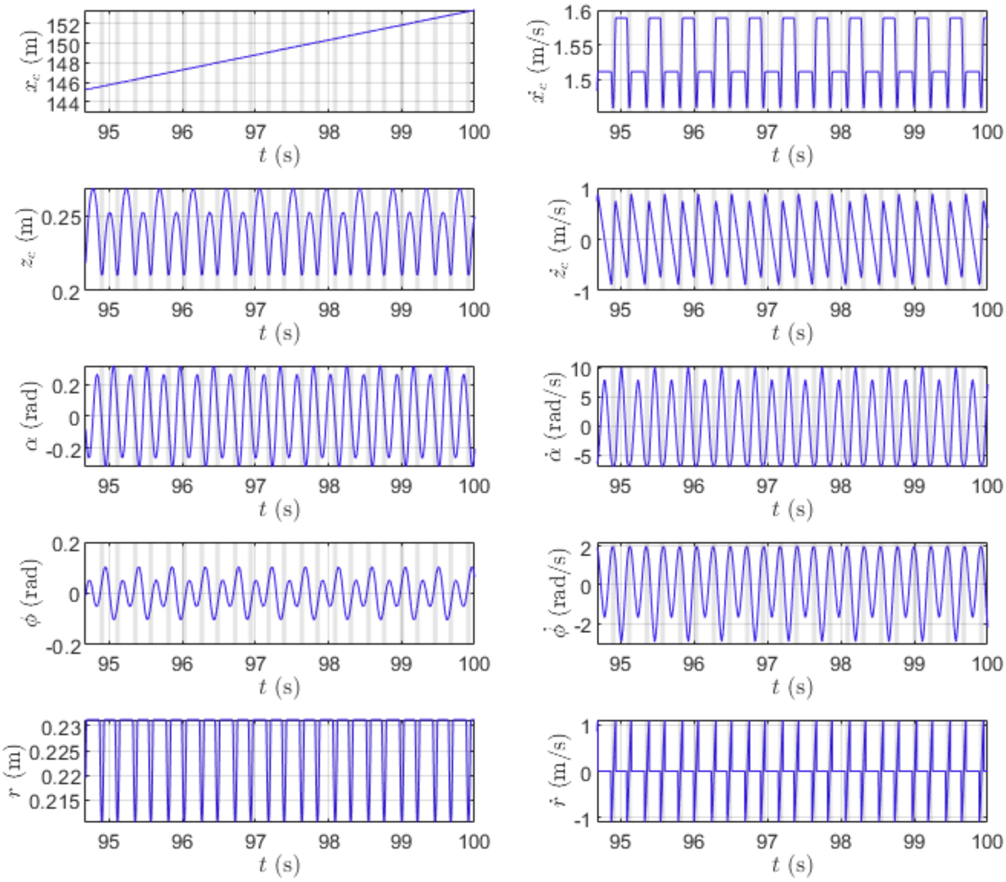


Figure 72: Time evolution of the states of the robot at steady state.

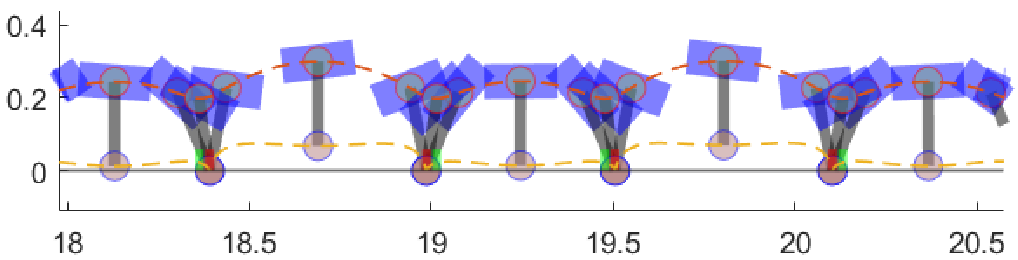


Figure 73: Trajectory of the planar one-legged robot highlighting the alternation of steps with different apex heights.

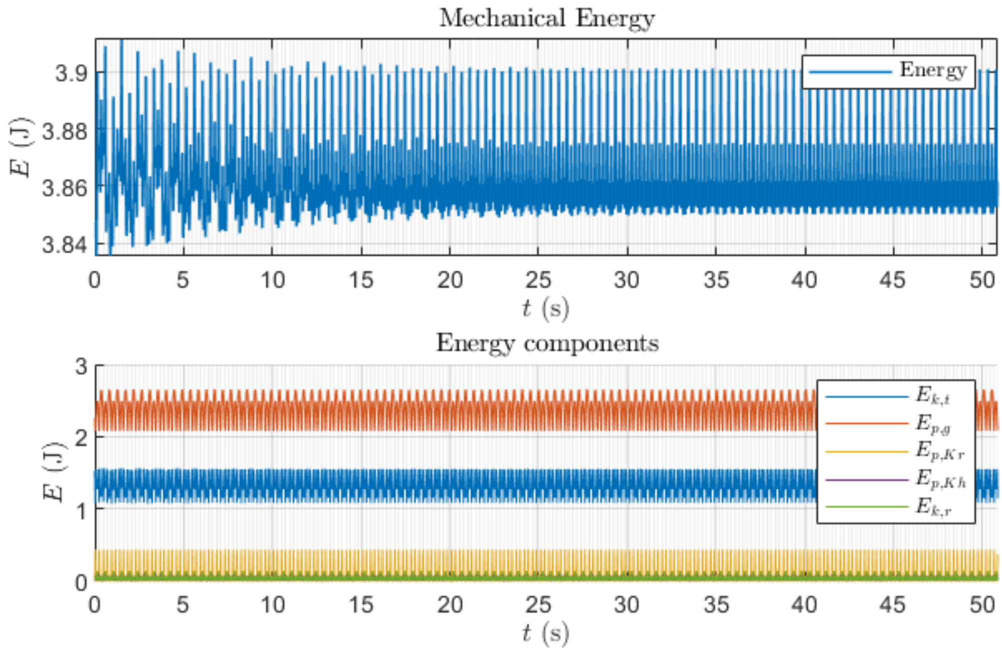


Figure 74: Energy of the robot. The total energy oscillates after convergence to a two-periodic actuated gait. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy.

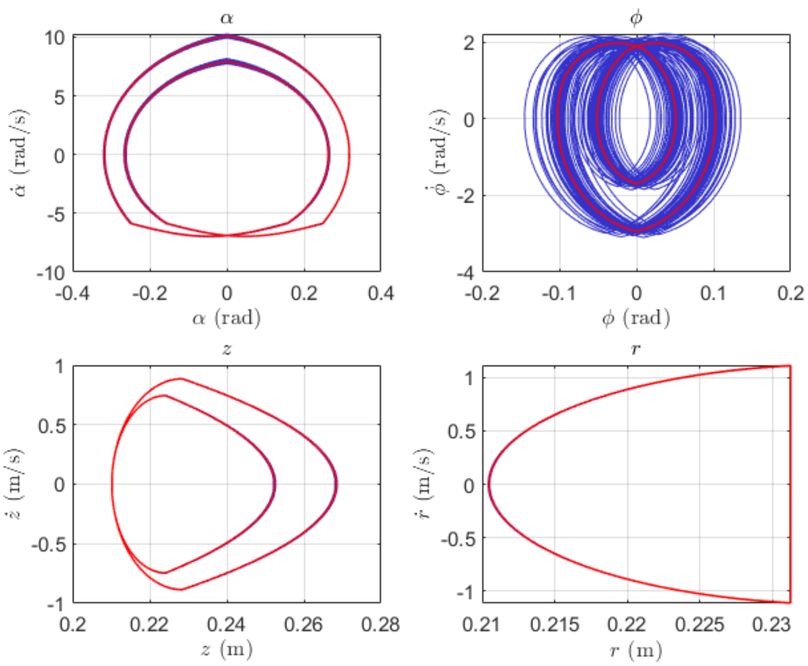


Figure 75: State trajectories in the phase plane. The red lines indicate the orbits after convergence to a periodic gait.

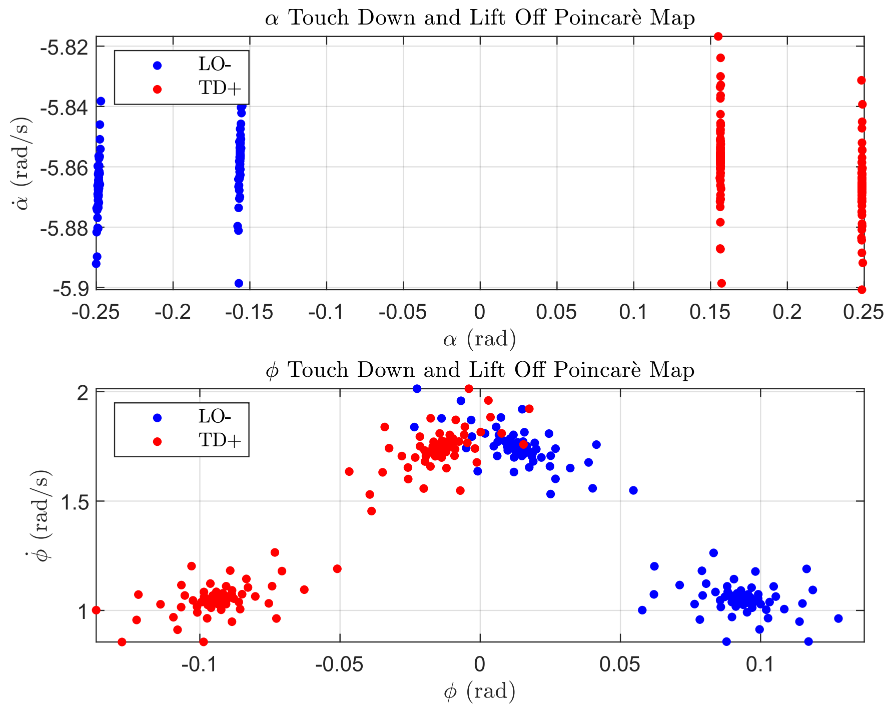


Figure 76: Poincaré maps of the system angles at the events of touchdown and lift-off. The plot highlights the presence of alternating couples of fixed points.

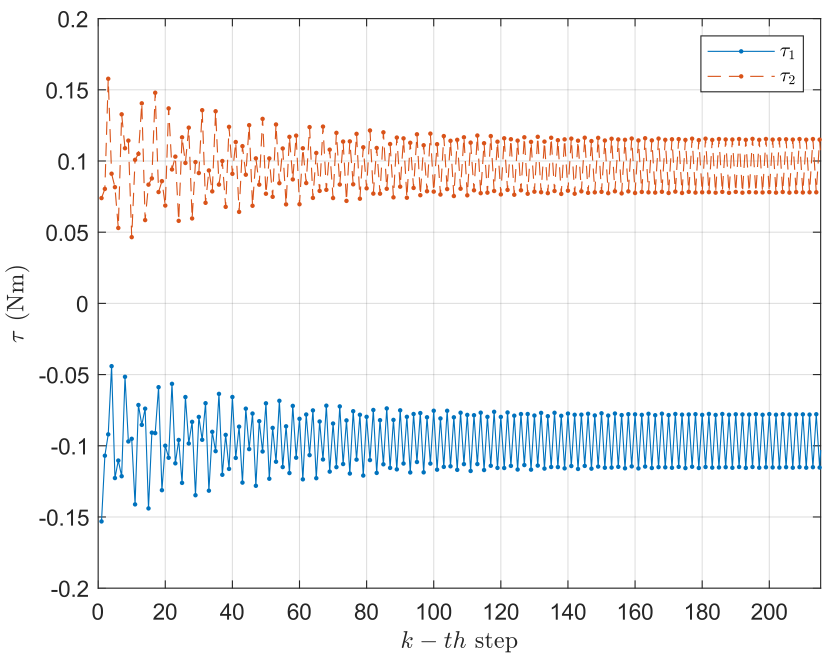


Figure 77: Control effort. Both τ_1 and τ_2 oscillate between two values after the two-periodic gait is reached.

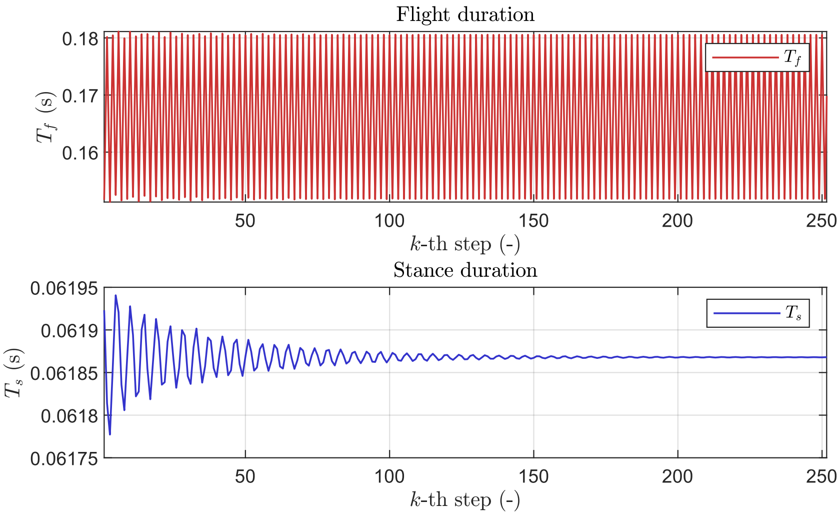


Figure 78: With $p = 2$, the stance time converges to a fixed value while the flight time oscillates between two.

5.3.3.3 Stabilized passive gait with p = 1 and hip stiffness adaptation

In this section, the simulation results with the spring adaptation law and with p = 1 are reported. The gain for the hip stiffness adaptation was set to $\gamma_\Omega = 0.5$ rad $N^{-1}m^{-1}s^{-1}$

The robot was initialized at touchdown with the parameterized initial conditions as described in section 5.3.1 with $\alpha_0 = 0.155$ rad and $\dot{x}_{c0} = 1.3$ m/s, the same as in the simulation of section 5.3.3.1. The results are shown and commented in Figures 79 - 87.

The states of the robot in the last part of the simulation are shown in Figure 79. The Poincaré map images at touchdown and lift-off in Figure 76 show how the angles converge to fixed points as the robot converges to a passive periodic gait. In fact, the control effort converges to zero and the energy settles to a constant value, as shown in Figures 86 and 81 respectively. The time evolution of the hip stiffness is shown in Figure 83. Observe how it converges to a constant value of about 1.11 Nm/rad.

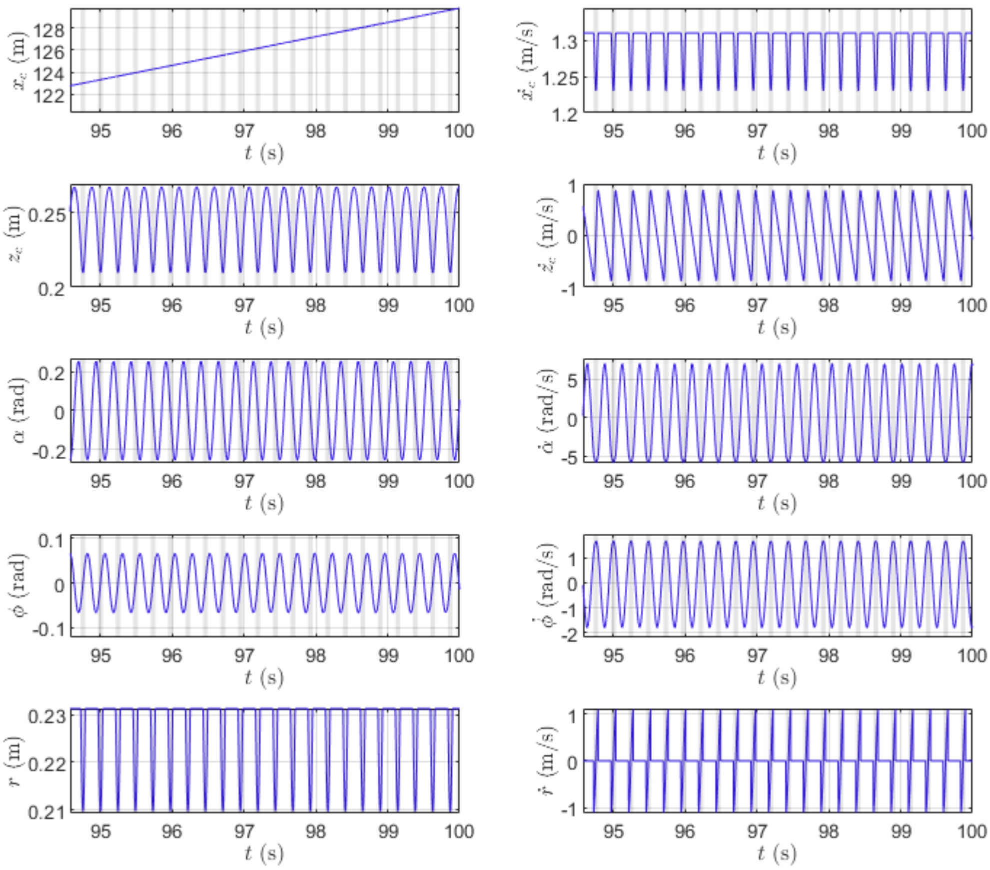


Figure 79: States of the robot during the last part of the simulation. In this time frame the robot has already converged to a passive periodic gait.

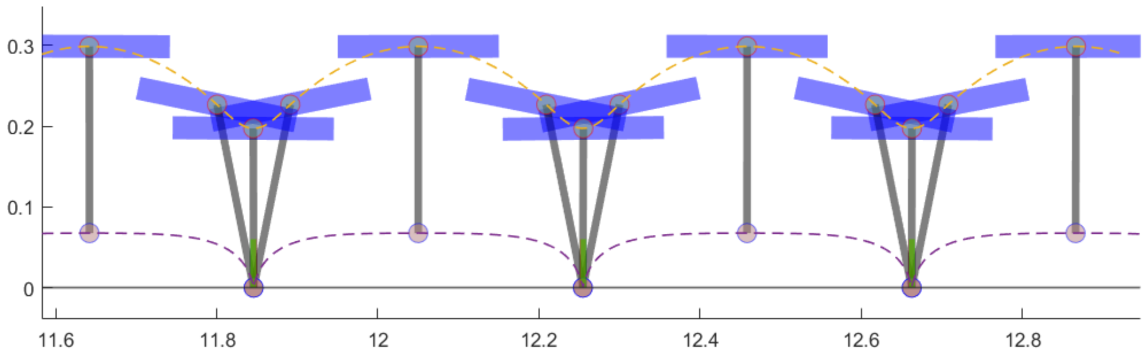


Figure 80: Trajectory of the planar one-legged robot after convergence to a passive gait.

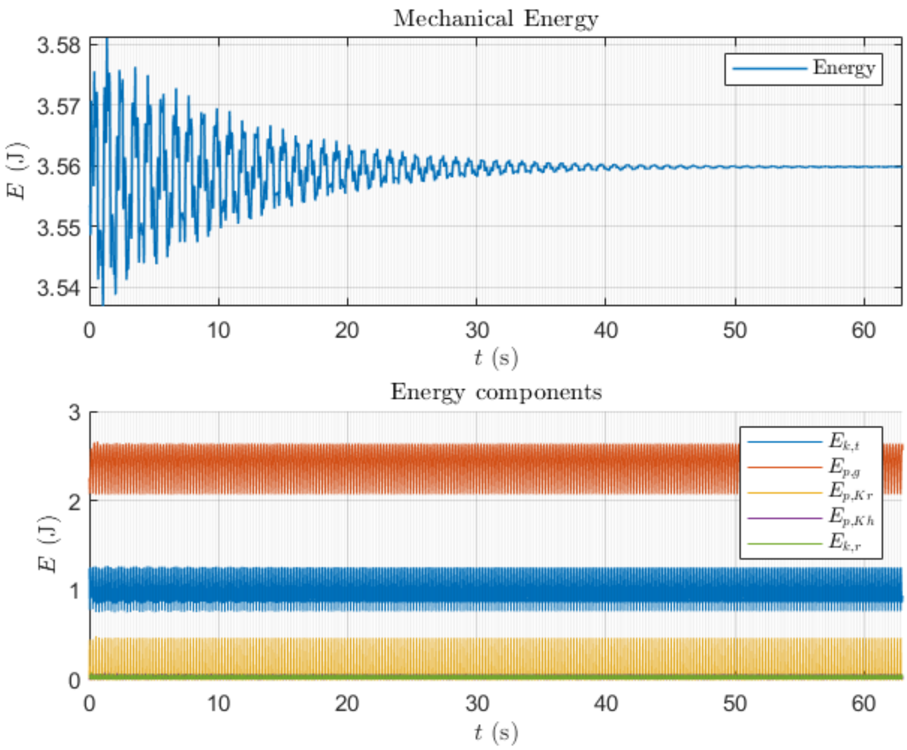


Figure 81: The energy settles to a constant value thanks to the adaptation of the hip stiffness. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy.

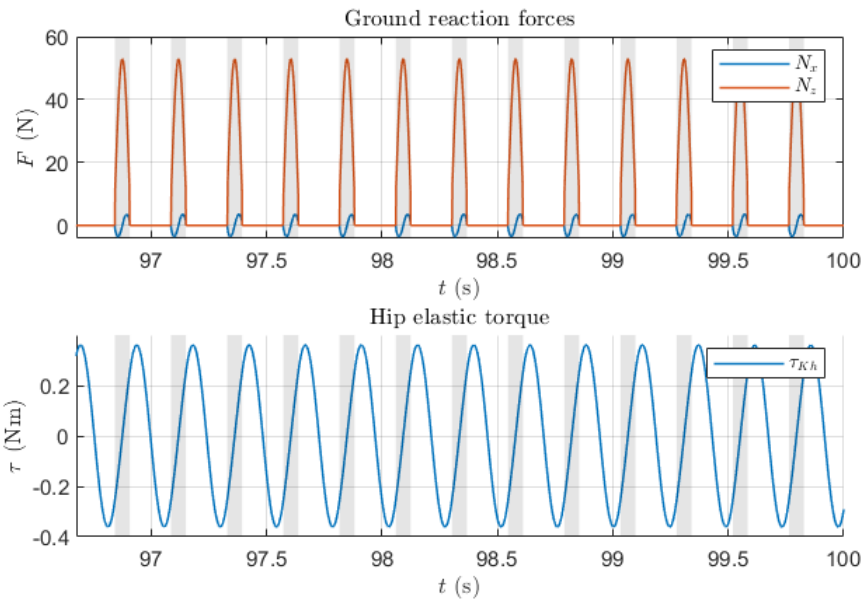


Figure 82: Detail showing the ground reaction forces and elastic hip torque at the end of the simulation.

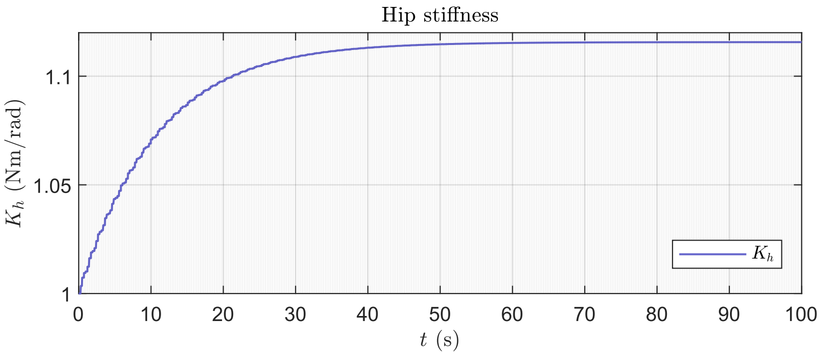


Figure 83: Time evolution of the hip stiffness. It converges to a constant value as a passive gait is reached.

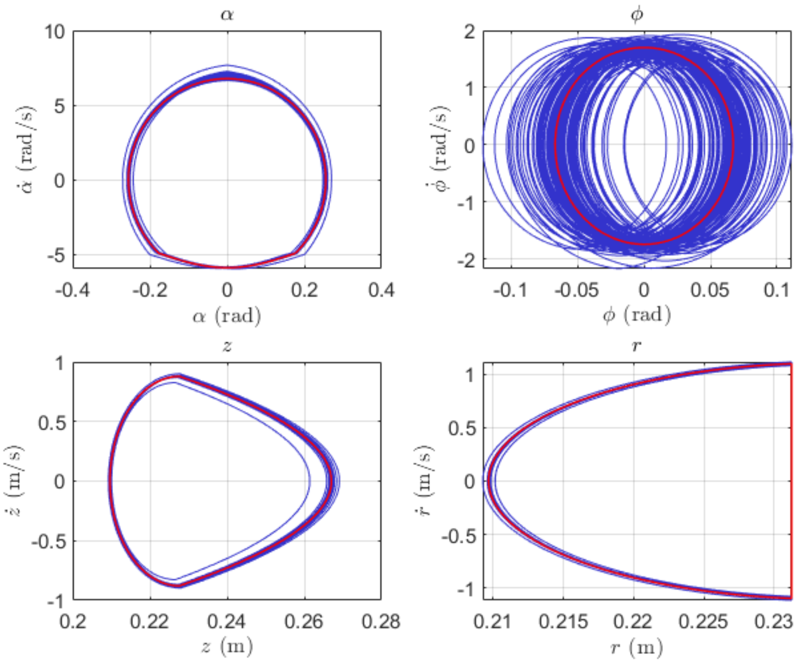


Figure 84: State trajectories in the phase plane. The red lines indicate the orbits after convergence to a periodic gait.

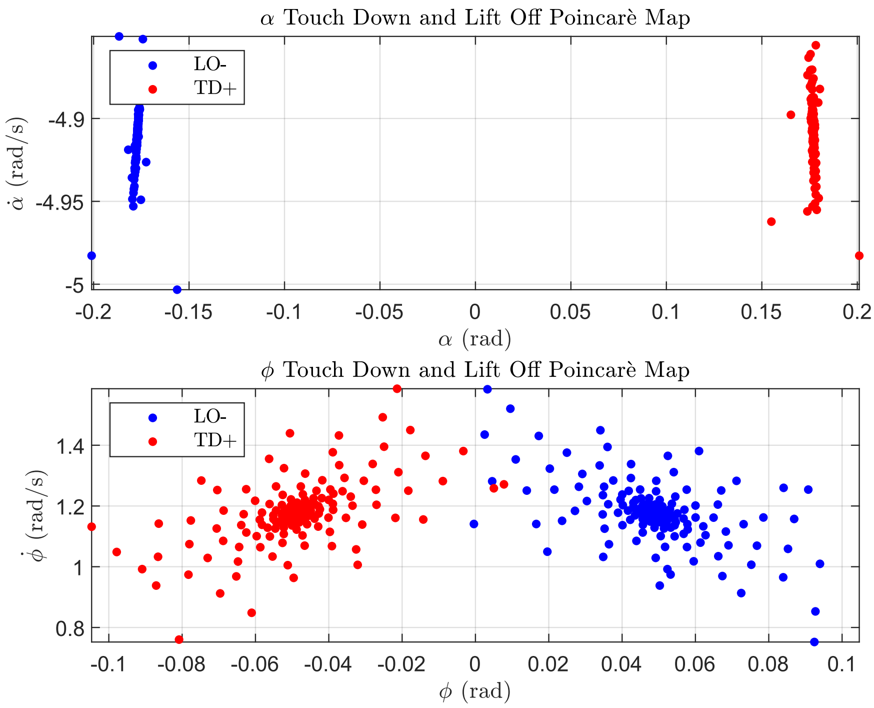


Figure 85: Poincaré maps of the robot angles at the events of touchdown and lift-off. The angles converge to fixed points.

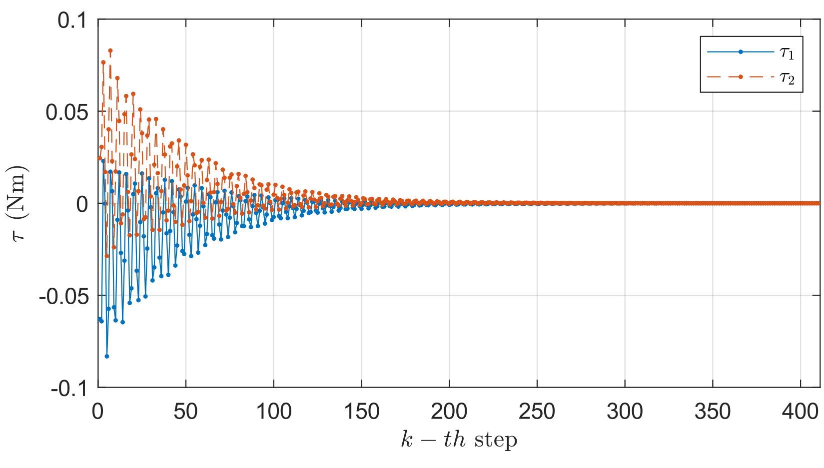


Figure 86: Control effort represented by τ_1 and τ_2. Both values eventually converge to a zero, making the gait passive.

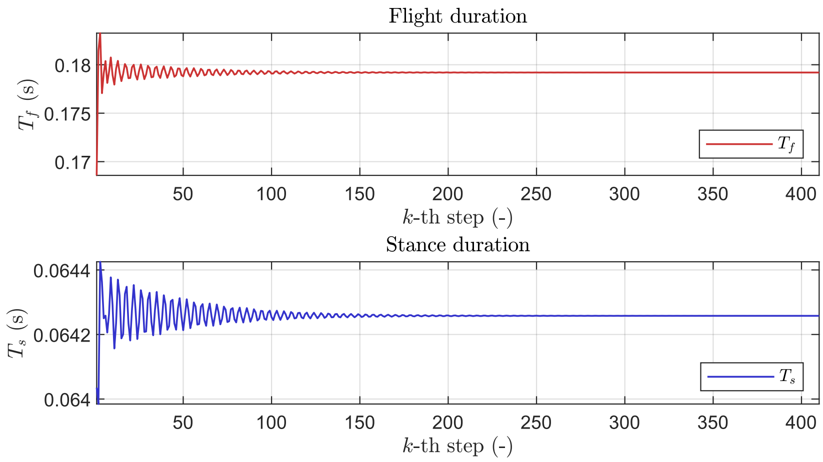


Figure 87: Stance and flight times. Both converge to constant values.

5.3.4 Extension to continuous torque profiles

The commanding of discontinuous torques, such as the piece-wise constant profile used in the non-dissipative touchdown control described in 5.3.2, is generally undesired on real motors. The aim of this section is to examine the possibility of using continuous profiles for the torque control during flight, while keeping the general control structure unchanged.

The first option analyzed is to use triangular profiles defined as follows:

$$\tau(t) = \begin{cases} \dfrac{4\tau_A}{T_f}t & \text{if } \; t \leq \dfrac{T_f}{4} \\ \tau_A - \dfrac{4\tau_A}{T_f}(t - \dfrac{T_f}{4}) & \text{if } \; \dfrac{T_f}{4} < t \leq \dfrac{T_f}{2} \\ \dfrac{4\tau_B}{T_f}(t - \dfrac{T_f}{2}) & \text{if } \; \dfrac{T_f}{2} < t \leq \dfrac{3T_f}{4} \\ \tau_B - \dfrac{4\tau_B}{T_f}(t - \dfrac{3T_f}{4}) & \text{if } \; \dfrac{3T_f}{4} < t \leq T_f \end{cases} \tag{92}$$

where τ_A and τ_B are two constants used to parameterize the torque profile. To compute τ_A and τ_B so that the control input achieves the desired $\alpha_{TD,des}$ and $\dot{\alpha}_{TD,des}$ in time T_f, remember that the differential equations governing the angles α and ϕ during flight are linear:

$$\begin{cases} J_l\ddot{\alpha} + J_b\ddot{\phi} = 0 \\ J_b\ddot{\phi} + K_h(\phi - \alpha) = \tau(t) \end{cases} \tag{93}$$

By integrating (93) from time 0 to T_f with (92) as input and considering the initial conditions to be known (they are the angles and angular velocities at lift-off), one obtains two equations in two variables. The system of equations can then be solved for τ_A and τ_B. The expressions obtained are relatively long, so they are reported in the Appendix B.2.

Before looking at the simulation results, a second option for continuous torque profile is presented. This option uses the Fourier series to approximate the piecewise constant torque actuation with a sum of sines. In particular, each constant piece of the signal can be seen as half the period of a square wave with amplitude τ_1 or τ_2. By using the well know Fourier series of a square wave, the torque actuation during flight can be defined as:

$$\tau(t) = \begin{cases} \dfrac{4\tau_1}{\pi} \displaystyle\sum_{n=0}^{N} \frac{1}{2n+1} \sin\left((2n+1)\frac{2\pi}{T_f}t\right) & \text{if } t \leq \dfrac{T_f}{2} \\ -\dfrac{4\tau_2}{\pi} \displaystyle\sum_{n=0}^{N} \frac{1}{2n+1} \sin\left((2n+1)\frac{2\pi}{T_f}t\right) & \text{if } \dfrac{T_f}{2} < t \leq T_f \end{cases} \tag{94}$$

where N is the desired Fourier expansion order and τ_1, τ_2 are the torque values computed with the method described in 5.3.2. Figure 88 shows a comparison of the piecewise constant profile, its Fourier series approximation and the triangular torque profile.

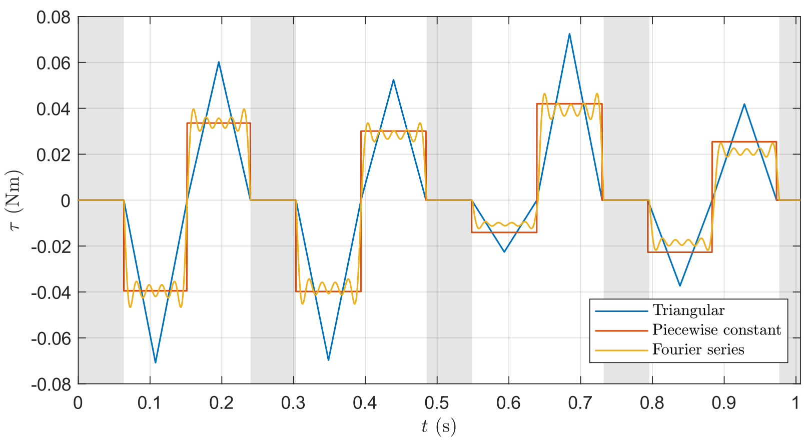


Figure 88: Comparison of the piecewise constant profile, its Fourier series approximation and the triangular torque profile. The Fourier expansion was truncated at $N = 4$.

Hereby the simulation results of using the three different torque controls are compared. The robot was initialized on an approximated passive gait with $\alpha_0 = 0.18$ rad and $\dot{x}_0 = 1.4$ m/s. The hip stiffness adaptation gain was set to $\gamma_\Omega = 0.5$ rad $\mathrm{N}^{-1}\mathrm{m}^{-1}\mathrm{s}^{-1}$ and for the triangular profile τ_A and τ_B were used to compute $\Delta\tau$ for equation (89) instead of τ_1 and τ_2. Figures 89 and 90 compare the time evolution of the hip stiffness and the control torque respectively. Interestingly, the simulations with triangular and piecewise constant profiles converge to the same value of K_h. The faster convergence with the triangular torques is due to the fact that with the same adaptation gain was set for γ_Ω, and the $\Delta\tau$ computed with the triangular profile is always larger in module, which makes the adaptation of the spring faster in (91). As expected, the simulation with the Fourier series approximation does not converge to a limit cycle with zero control effort. This is because this method, unlike the other two, does not use exact dynamics integration and thus there is an error at every step between the desired and the realized α_{td} and $\dot{\alpha}_{td}$. Nonetheless, the residual steady state control effort is very low.

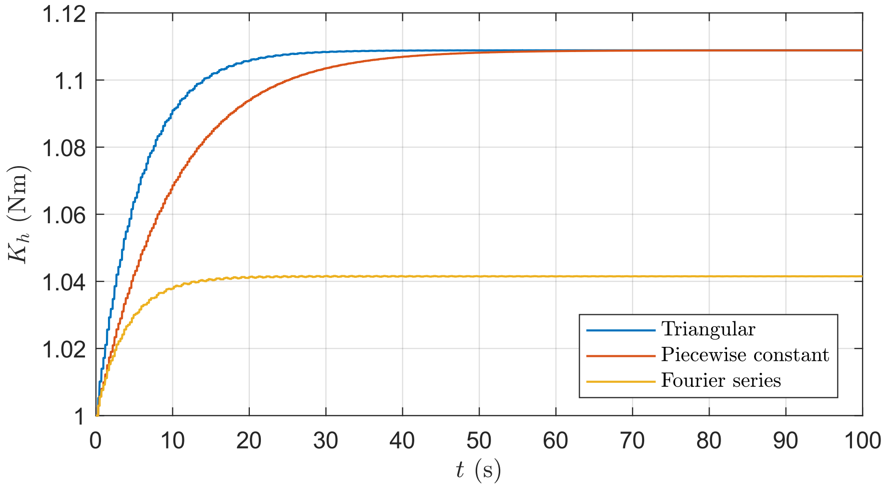


Figure 89: Comparison of hip stiffness adaptation with different torque profiles.

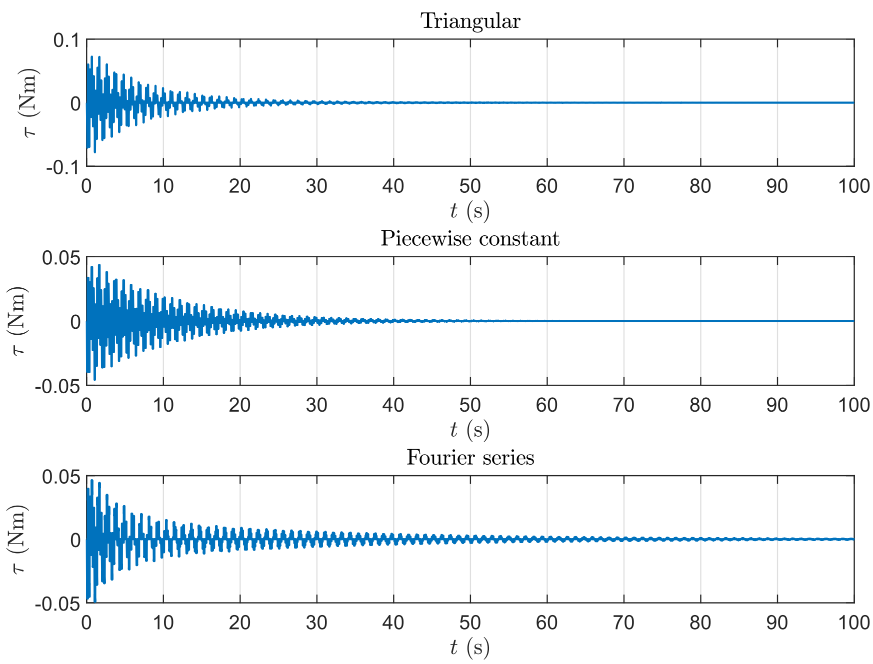


Figure 90: Comparison of the torque actuation with torque profiles.

5.3.5 Extension to perform velocity control

The aim of this section is to demonstrate that the non-dissipative touchdown control can be easily extended with the aim of controlling the horizontal velocity of the robot. The approach proposed is inspired by Raibert's foot placement algorithm, which makes use of the gravity moment during stance to accelerate or decelerate the robot. The idea is to keep the control structure unchanged with the exception of the desired leg angle, to which a deviation α_{err} is added. The new desired angle is:

$$\hat{\alpha}_{td,des}(k) = \alpha_{td,des}(k) + \alpha_{err} \tag{95}$$

where $\alpha_{td,des}(k)$ is computed as before with (79) by averaging of previous lift-off angles. The computation of the other quantities remains unchanged, except from the fact that $\hat{\alpha}_{td,des}(k)$ is used instead of $\alpha_{td,des}(k)$.

The idea comes from the fact that with $p = 1$ and the stiffness adaptation law active, the desired angle $\alpha_{td,des}(k)$ tends to the angle that puts the foot in the neutral point of the stance. Thus, a negative α_{err} will cause the foot to be placed slightly before the neutral point, causing the robot to accelerate, whereas a positive α_{err} will cause the robot to decelerate.

This was tested by using the following signal for α_{err}:

$$\alpha_{err} = \begin{cases} 0 \text{ rad if t} < 10\text{s} \\ -0.001 \text{ rad if } 10\text{s} \leq \text{t} < 20\text{s} \\ 0 \text{ rad if } 20\text{s} \leq \text{t} < 30\text{s} \\ +0.001 \text{ rad if } 30\text{s} \leq \text{t} < 40\text{s} \\ 0 \text{ rad if } 40\text{s} \leq \text{t} < 50\text{s} \end{cases} \tag{96}$$

The robot's initial conditions were set on a approximated periodic solution from the pair ($\alpha_0 = 0.18$ rad and $\dot{x}_0 = 1.3$ m/s). The adaptation gain for the spring was set to $\gamma_\Omega = 0.5$ rad $\text{N}^{-1}\text{m}^{-1}\text{s}^{-1}$ together with $p = 1$. For the torque control during flight, the triangular profiles were used. Figures 91 - 95 show the results of the simulation. Observe in Figure 91 how the robot successfully accelerates and decelerates. As the robot reaches higher velocities, the apexes of $\dot{z}_c$ become lower and the leg length r compresses less during the stance phases.

The hip spring becomes more stiff as the robot accelerates, then the stiffness decreases again and settles to a fixed value. As expected, after the robot stops decelerating the control effort converges to zero as the robot settles on a passive gait thanks to the stiffness adaptation law.

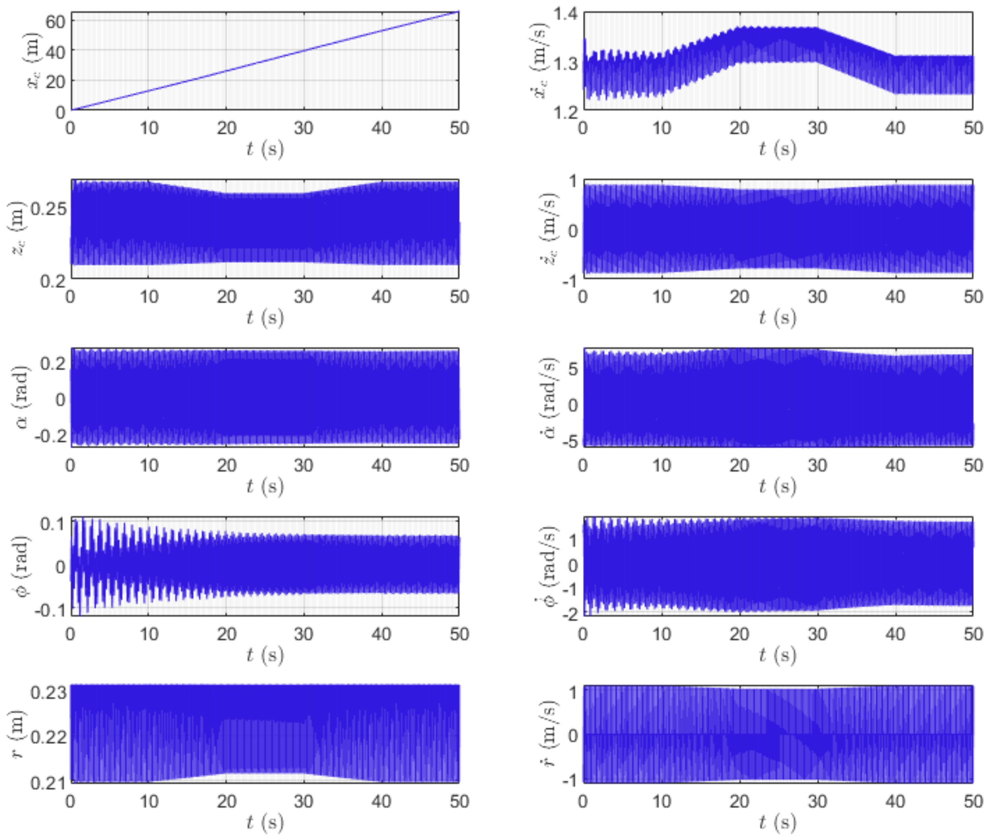


Figure 91: States time evolution of the robot as it accelerates and decelerates. Observe in the plot of $\dot{x}_c$ how the robot successfully accelerates, keeps the velocity constant, decelerates and finally keeps again a constant velocity.

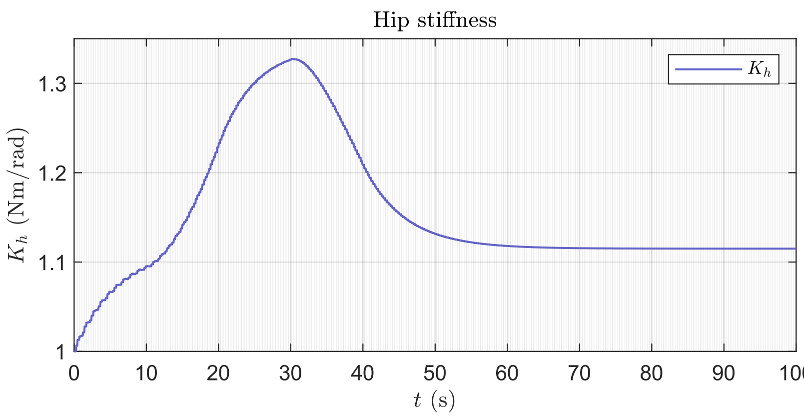


Figure 92: Hip stiffness adaptation. The spring becomes more stiff as the robot accelerates, then decreases again and settles to a fixed value.

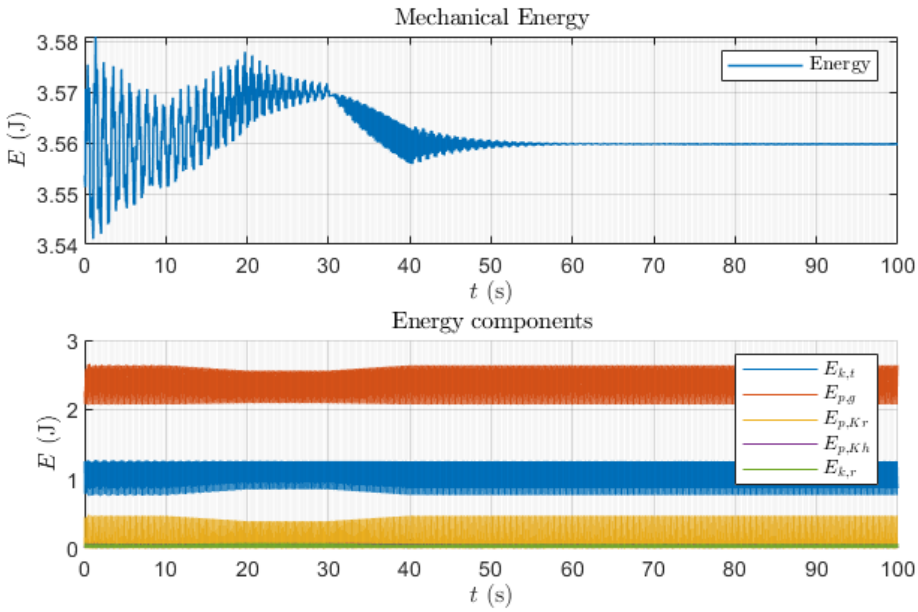


Figure 93: Energy time evolution of the robot as it accelerates and decelerates. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy.

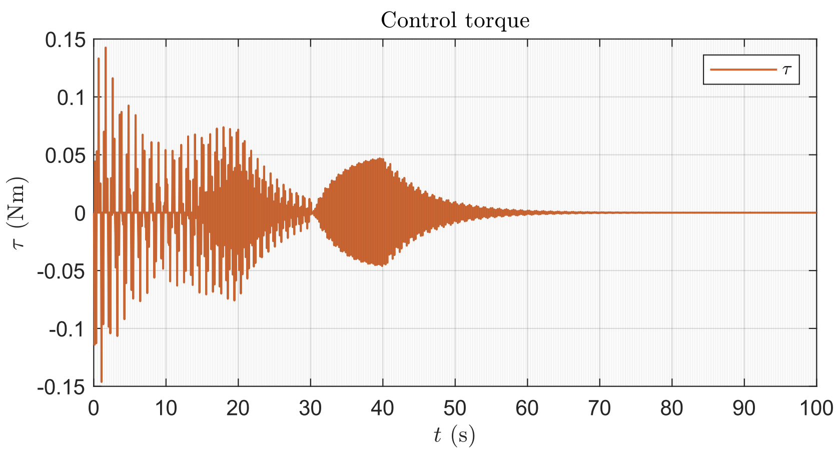


Figure 94: Control torque. After the robot stops decelerating the control effort converges to zero.

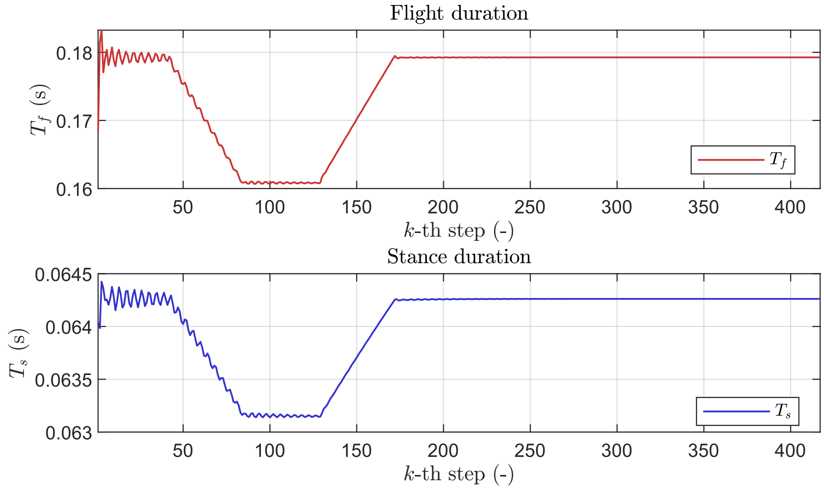


Figure 95: Stance and flight times. Both T_s and T_f decrease almost linearly during the acceleration.

To achieve tracking of a desired velocity, a constant value of α_{err} can be used, and its sign can be changed depending on whether the current velocity is lower or higher than the desired one. Alternatively, α_{err} can also be computed with a simple proportional-integral feedback law on the horizontal velocity error during the flight phase:

$$\alpha_{err} = \hat{K}_{p,\alpha} \cdot (\dot{x}_{c,flight} - \dot{x}_{c,des}) + \hat{K}_{i,\alpha} \cdot \int_0^t (\dot{x}_{c,flight} - \dot{x}_{c,des})dt \tag{97}$$

where $\hat{K}_{p,\alpha}$ and $\hat{K}_{i,\alpha}$ are positive gains. In this way, the angle deviation α_{err} is computed before each jump based on the velocity error of the previous jump and the integral of the error on all the previous jumps. The integral term was introduced to remove the steady state error. The control law (97) will be used in the next section in the case where damping is added to the joints of the robots.

5.3.6 Joint viscous damping

In this section, the case in which friction is present in the robot's joints is examined, so that energy loss occurs during any joint motion. For simplicity, only a viscous friction proportional to the joint velocities was modelled. The damping coefficients on the hip revolute joint and on the leg prismatic joint are called C_h and C_r respectively.

5.3.6.1 Non-dissipative touchdown controller with joint damping

Firstly, the case in which the unchanged non-dissipative touchdown controller with $p = 1$ and with disabled adaptive stiffness law is briefly shown in Figures 96-98. To highlight the effect of damping, the robot was initialized on a exact periodic gait of the undamped system, with $\alpha_0 = 0.171$ rad, $\dot{x}_0 = 1.263$ m/s and $\dot{z}_0 = -0.940$ m/s. The joint damping coefficients were set as $C_h = 0.001$ Nms$\cdot$rad^{-1} and $C_r = 0.01$ Nsm^{-1}.

As depicted in Figure 96, the robot accelerates forward while the flight apexes decrease in height. The horizontal forward acceleration is caused by the fact that the flight torque applied no longer matches the exact dynamics of the system and the damping causes the touchdown angle to be smaller than the desired one, because the leg is slowed down during the forward swinging. This causes a forward acceleration as the foot is placed before the neutral point of the successive stance phase and the gravity tends to tip the robot forwards.

The control effort gradually increases over time while the time duration of the stance and flight phases decrease, as shown in Figures 97 and 98 respectively. The robot eventually falls to the ground after about nine seconds.

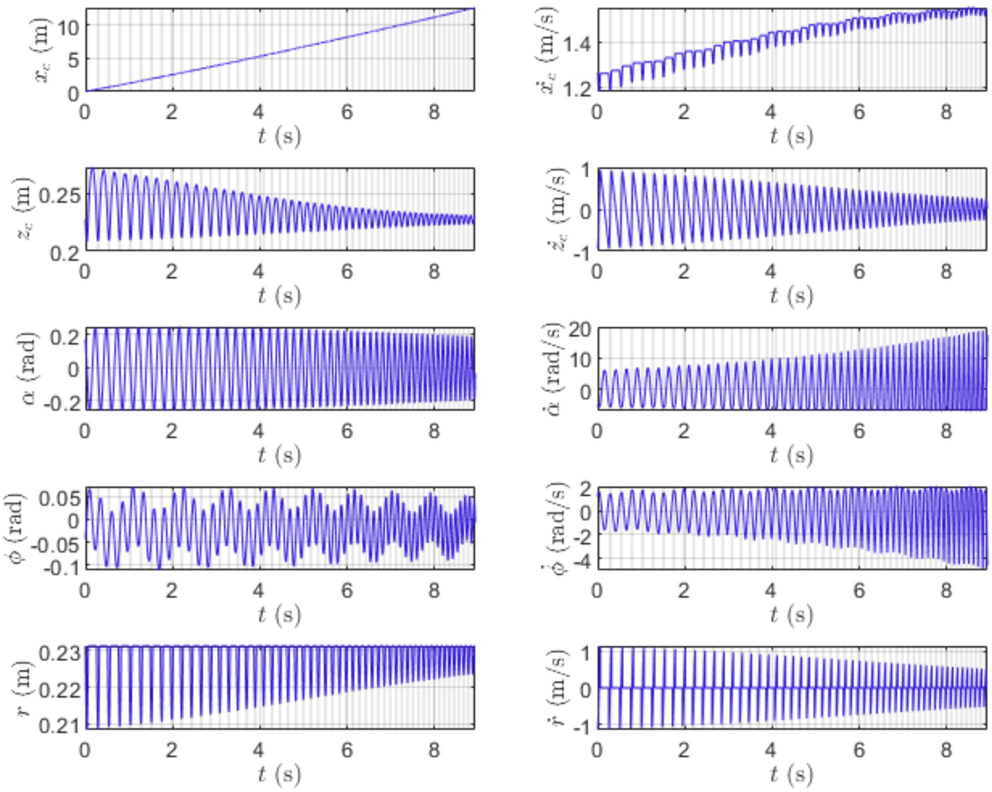


Figure 96: Evolution of the states of the robot with damping on the joints. The robot accelerates forward while the flight apexes decrease in height.

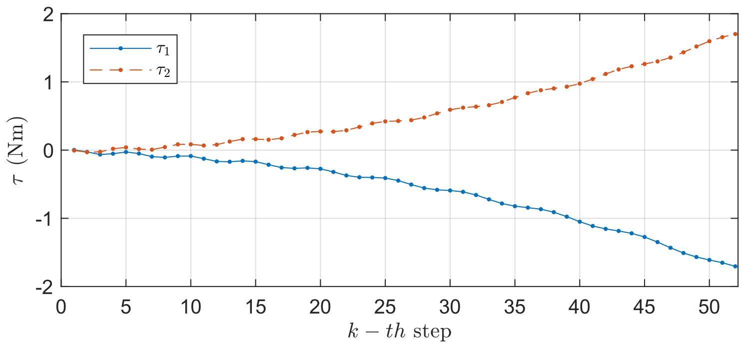


Figure 97: Control torque applied by the unchanged non-dissipative touchdown controller in the presence of damping on the joints.

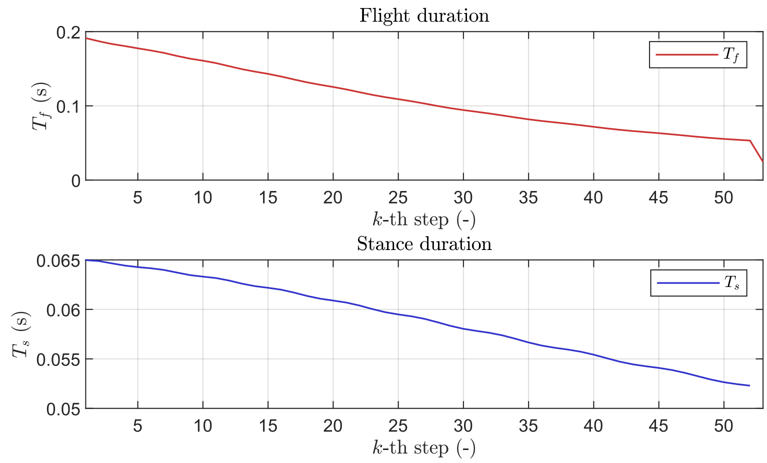


Figure 98: Flight and stance time duration in the presence of damping on the joints.

5.3.6.2 Extended controller with joint damping

Hereby, an approach to keep the robot running in the presence of damping in the joints is described. In section 5.3.5, a method to control the robot's horizontal velocity was proposed. The control law with proportional and integral terms on the velocity error during flight (97) can be used to counter the changes in the running velocity caused by the damping in the joints. Moreover, an energy regulator of the same form as (13) can be used to compensate the energy losses caused by the damping and the touchdown impacts and restore the energy of the robot to a desired value during stance. These two tasks can be performed in parallel with the unmodified non-dissipative touchdown controller.

To sum up, the proposed control strategy consists of three different tasks:

1. Non-dissipative touchdown controller (79) that computes the desired $\alpha_{td,des}$, as described in section 5.3.2. The controller can be set with $p = 1$ and the spring adaptation law (89) can be used to minimize the controller effort;

2. PI controller (97) that computes the deviation α_{err} to be added to $\alpha_{td,des}$ in order to track a desired horizontal velocity, as described in section 5.3.5;

3. Energy regulator (13) during stance to compensate the energy losses, as described in section 3.

The parameters used for the simulation are reported in Table 10. The robot was initialized on the same passive gait as in section 5.3.6.1. For the desired velocity and energy set-points, the nominal values of the passive periodic gait were used. During flight the piecewise constant torque profile was used.

In the simulation, the spring adaptation law was enabled at $t = 40$ s. In this way, the spring adaptation is not active during the initial transients, which could cause a relative change in the stiffness too high to be realistically achievable. Hereby, the switching on of the spring adaptation law is indicated in the graphs with vertical dashed black lines. Note that the controller here described stabilizes the robot with damping in the joints even without the use of the hip stiffness adaptation law. However, by adapting the stiffness the control inputs converge to lower values.

Parameter	Value
C_h	0.001 Nms·rad^{-1}
C_r	0.01 Nsm^{-1}
p	1
γ_Ω	0.5 rad N^{-1}m^{-1}s^{-1}
γ_E	40 NsJ^{-1}m^{-1}
$\hat{K}_{p,\alpha}$	0.03 rad·sm^{-1}
$\hat{K}_{i,\alpha}$	0.004 rad·m^{-1}
$\dot{x}_{c,des}$	1.263 m/s
E_{des}	3.574 J

Table 10: List of the numerical values assigned for the controller parameters in the simulation with damping.

Figure 99 shows the evolution of the states of the robot. The forward velocity oscillates in a initial transient after which the robot converges to the desired speed. A plot of the velocity during flight and its desired value is shown in Figure 100. In the phase plane, the states perform the trajectories as shown in Figure 101. Note how even after convergence to steady state there are discontinuities in the leg angular velocity $\dot{\alpha}$ at touchdown, as highlighted by the non-smooth red orbit in the top left graph.

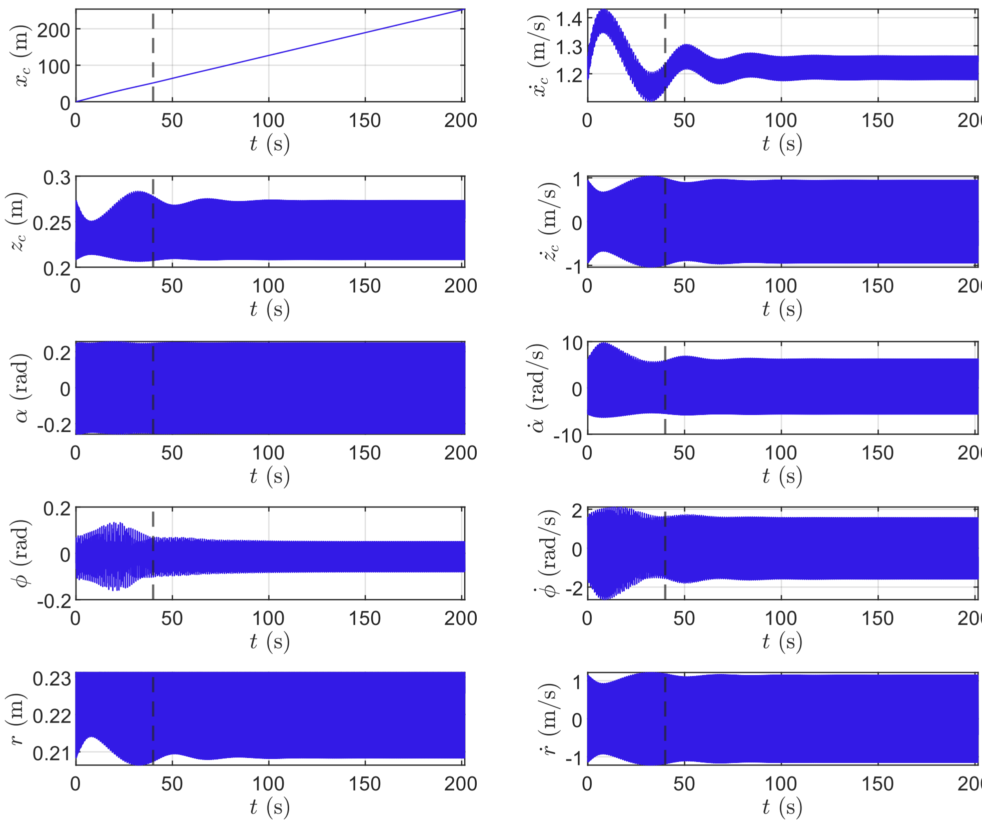


Figure 99: Evolution of the states of the robot with damping on the joints. After an initial transient, the robot converges to the desired speed.

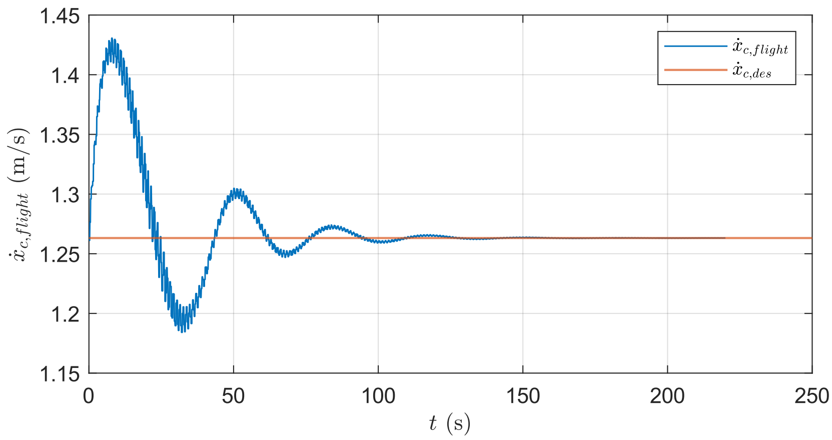


Figure 100: Horizontal velocity during flight compared to its desired value.

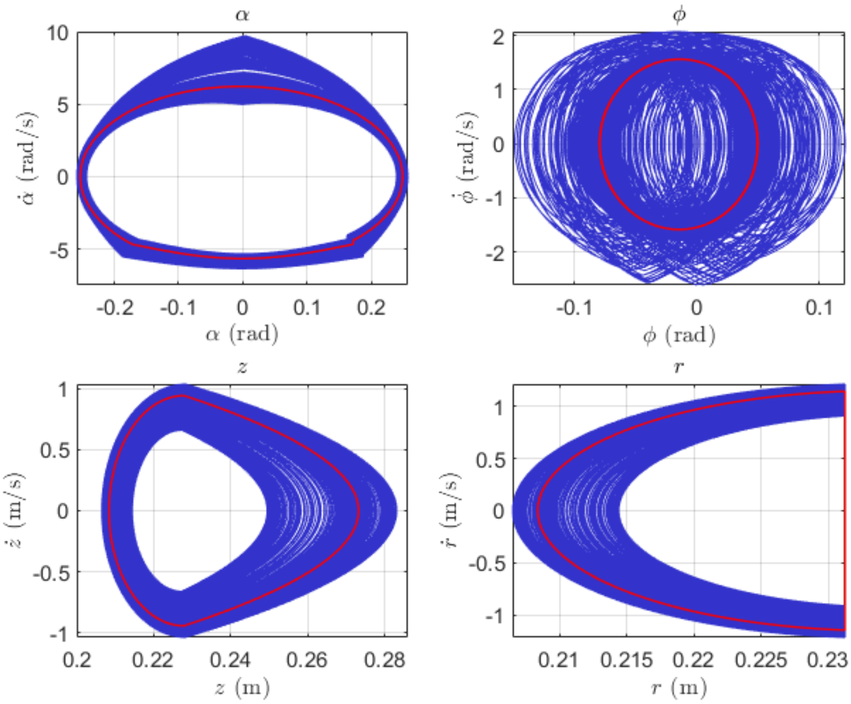


Figure 101: State trajectories in the phase plane. The red lines indicate the orbits after convergence to the steady state.

The flight hip control torque and the stance leg force are displayed in Figure 102 and 103 respectively. As expected, the control effort does not converge to zero as a passive gait with damping in the joints is not possible. Nonetheless, after the initial transient the control torques and forces settle on relatively small values thanks to the hip stiffness adaptation, whose time evolution is plotted in Figure 104.

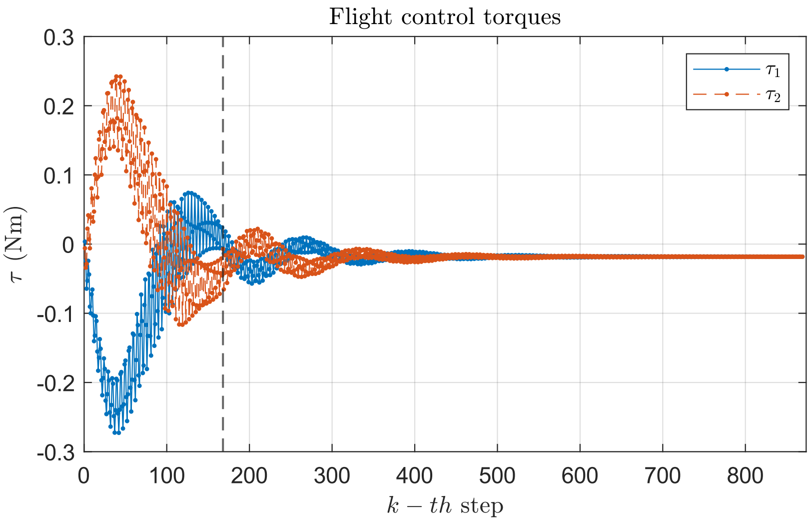


Figure 102: Control torque applied with the modified controller in order to compensate damping.

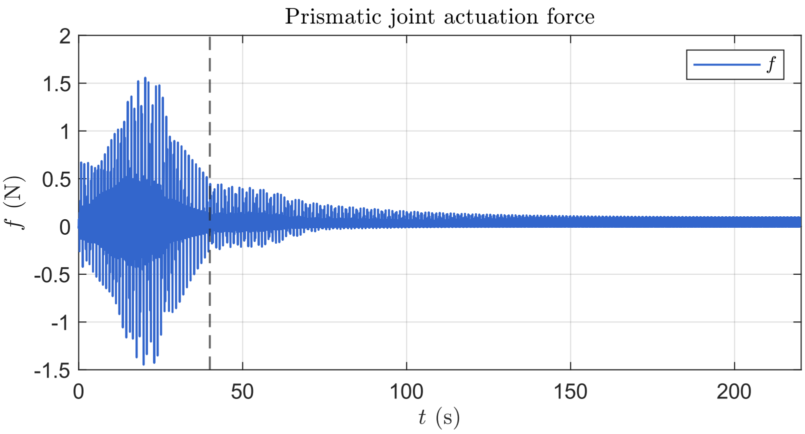


Figure 103: Prismatic joint force actuator. The leg actuator injects energy in the system during the stance phases.

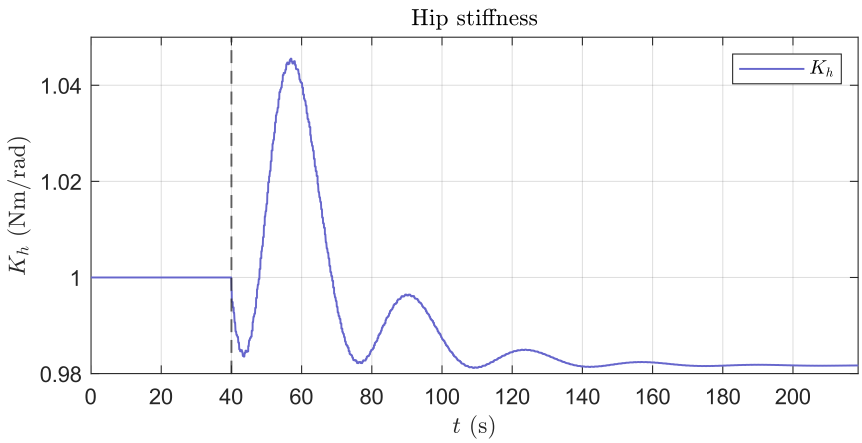


Figure 104: The spring adaptation law was enabled at $t = 40$ s.

The time duration of the stance and flight phases converge to fixed values, as shown in Figure 105. Finally, the energy of the robot is plotted in Figure 106. Unlike in the passive periodic gaits, the energy does not converge to a fixed value but oscillates in a limited interval instead.

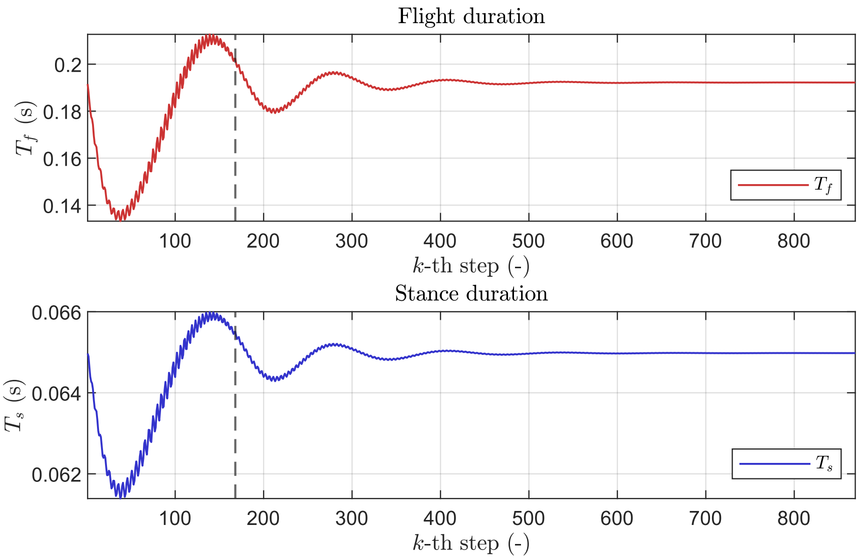


Figure 105: Flight and stance time duration in the presence of damping on the joints.

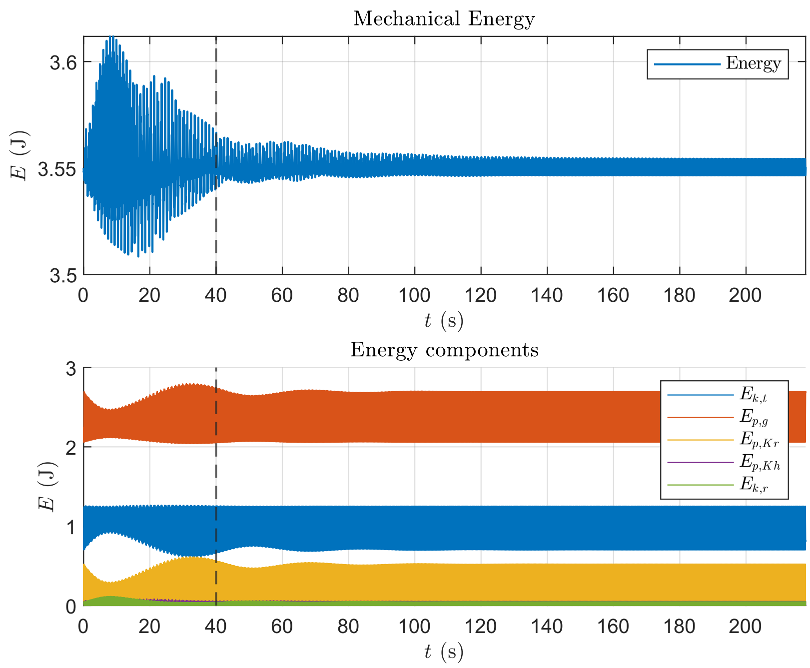


Figure 106: Energy of the robot. $E_{k,t}$ is the kinetic translational energy, $E_{p,g}$ the potential gravitational energy, $E_{p,Kr}$ the potential elastic energy of the linear spring, $E_{p,Kh}$ the potential elastic energy of the polar hip spring and $E_{k,r}$ kinetic rotational energy.

In this section, the non-dissipative control for orbital stabilization of passive gaits, the forward velocity control through foot placement and the energy regulation through actuation forces in the leg prismatic joint were successfully combined to achieve running with low effort in the presence of damping in the joints.

5.4 Comparison of the two control approaches

Sections 5.2 and 5.3 described two very different control approaches for the locomotion of the one-legged planar robot with body link: Raibert's controller based on the foot placement algorithm and Hyon's and Emura's non-dissipative touchdown control. The results presented so far offer the basis for a comparison between the two approaches. In this section, the main differences between the two approaches are summed up.

Table 11 reports some of the intrinsic differences of the two approaches in the case where no damping is present in the system.

Foot placement algorithm	Non-dissipative touchdown control
No hip compliance in the robot	Exploits hip compliance to passively swing the leg backward and forward
Tuning of six control gains required	Only one control parameter for the hip stiffness adaptation law
The body attitude can be controlled during stance	The body attitude evolves naturally according to the passive dynamics
The hopping height can be directly controlled with the energy regulator	The hopping height is determined by the passive gait constraints
Control torques do not converge to zero	Control torques converge to zero by adapting the hip stiffness
Can easily track velocity trajectories or travel to a desired set-point in space	Only one passive gait per energy level for a fixed hip spring stiffness. Changing velocity requires switching to a different passive gait with a modified controller

Table 11: Qualitative comparison between the two locomotion controllers.

In the presence of joint damping, Raibert's method can be used without any change needed to the control algorithm. This highlights the robustness of the approach, which is due to the fact that feedback loops are used in all of the controller components and modelling errors can be thus compensated. In Hyon's and Emura's method, the controller applies feed-forward torques during flight which are computed at lift-off based on an accurate knowledge of the model. If the model presents errors or does not take joint friction or damping into account, feedback loops have to be added in order to compensate, like discussed in section 5.3.6.

The two control approaches result in very different state trajectories. Figure 107 compares the snapshots of the robot along two trajectories obtained with the two controllers while running at a constant speed of 1.3 m/s.

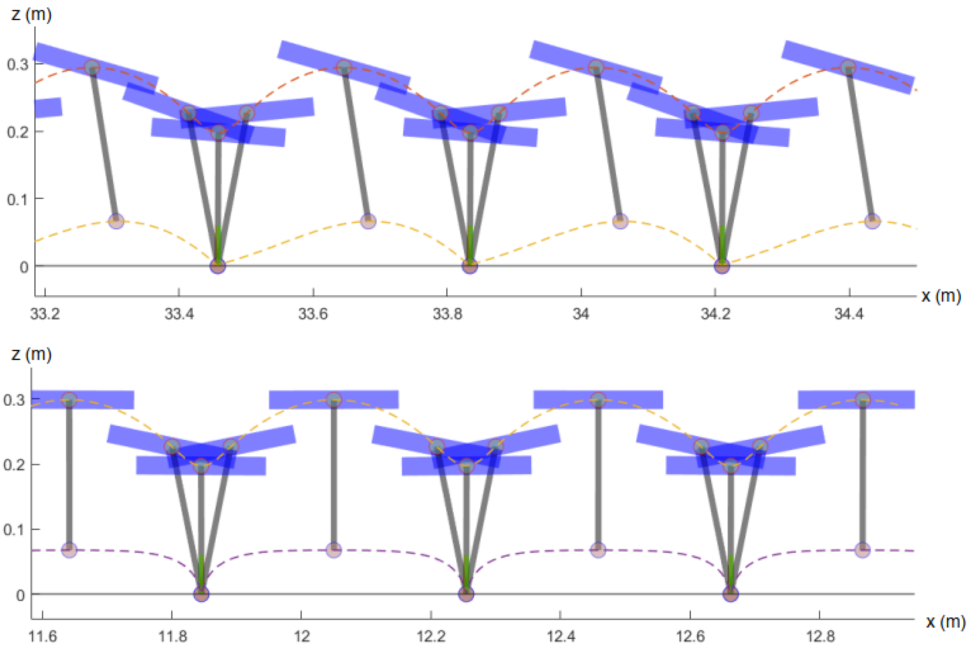


Figure 107: Snapshots of the robot along two trajectories obtained with the two controllers while running at a constant speed of 1.3 m/s. The upper picture is the foot placement algorithm, the lower is the non-dissipative touchdown control.

The main advantage of the the non-dissipative touchdown control is of course the low control input required to keep the robot running. For example, in the simulations described in section 5.3 the peak control torques were less than 0.3 Nm, compared to the high peaks of up to 30 Nm in the simulations with Raibert's approach of section 5.2. The control gains of the foot placement algorithm can be tuned in order to try to reduce the control effort, but there are lower thresholds after which the robot becomes unstable.

Another advantage may be in the fact that the convergence to passive gaits not only leads the control effort to zero, but also minimizes the foot impact forces at touchdown in the transversal direction with respect to the leg. This could lead to advantages in the implementation in a hardware setting in which transversal foot impacts are undesired.

6 Conclusions

A wide scope study of the locomotion of planar one-legged robots was performed, considering models of different complexity and different control approaches.

Firstly, the case of a one-dimensional hopper was considered. This was done in order to understand how to correctly simulate hybrid dynamical systems and to test a simple energy regulator with the aim of controlling the hopping height.

Secondly, the well-known SLIP model was studied with respect to its stability properties and periodic solutions. The SLIP was then extended to the pitching-SLIP, of which the passive gaits with somersaults were characterized. Two controllers based on energy and angular momentum regulation were then formulated and validated through numerical simulations. The results showed that it is possible to stabilize passive gaits with somersaults of the pitching-SLIP, making the control effort converge to zero. The controller was then successfully tested on the body-SLIP model, which extended the previous by adding an inertia to the leg. While both models present numerous simplifications with respect to more advanced robots, this study was performed with the intent of laying the foundations for a later extension to more complex cases, in which the somersaults are not necessarily present in the passive gaits.

Thirdly, the locomotion of a planar one-legged robot with a body link was studied. Raibert's control approach based on the foot placement algorithm was studied with the aim of understanding the principles behind its success and to later compare it with approaches based on passive dynamics. Then, the focus was shifted again towards the locomotion based on limit cycles and passive gaits. The non-dissipative touchdown controller of Hyon and Emura was implemented and studied. Furthermore, it was extended to be used with continuous torque profiles and to perform velocity tracking. Moreover, viscous friction was modelled in the joints in order to study its effect on the controller. The controller was successfully extended by combining it with a forward velocity control through foot placement and an energy regulator and was shown to be capable of stabilizing running in the presence of damping in the joints with low steady state control inputs. The purpose of this chapter was to achieve a deeper understanding of the principles behind passive locomotion by implementing and studying a controller from the literature, while also identifying its limits in more realistic simulation settings.

The outlook of this work is twofold: firstly, the controllers based on energy and angular momentum regulation should be further studied, to characterize their robustness and see whether they are applicable to more complex models; secondly, the non-dissipative touchdown controller could be extended to segmented legs in which a revolute joint is present instead of a prismatic joint, in order to be directly applicable to DLR's quadruped *Bert*. Then, the controller could be tested on hardware, which still has not be done to the author's knowledge. DLR's hardware test-bed of a single *Bert* leg could be a good starting point with this purpose in mind.

References

[1] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer handbook of robotics*, pp. 1203–1234, Springer, 2016.

[2] S. Seok, A. Wang, M. Y. Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, and S. Kim, "Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot," *Ieee/asme transactions on mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2014.

[3] R. Alexander *et al.*, "Three uses for springs in legged locomotion," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 53–61, 1990.

[4] T. McGeer *et al.*, "Passive dynamic walking," *Int. J. Robotics Res.*, vol. 9, no. 2, pp. 62–82, 1990.

[5] C. M. Thompson and M. H. Raibert, "Passive dynamic running," in *Experimental Robotics I: The First International Symposium Montreal, June 19–21, 1989*, pp. 74–83, Springer, 2006.

[6] P. A. Bhounsule, J. Cortell, and A. Ruina, "Design and control of ranger: an energy-efficient, dynamic walking robot," in *Adaptive Mobile Robotics*, pp. 441–448, World Scientific, 2012.

[7] A. Raffin, D. Seidel, J. Kober, A. Albu-Schäffer, J. Silvério, and F. Stulp, "Learning to exploit elastic actuators for quadruped locomotion," *arXiv preprint arXiv:2209.07171*, 2022.

[8] I. Poulakakis, E. Papadopoulos, and M. Buehler, "On the stability of the passive dynamics of quadrupedal running with a bounding gait," *The International Journal of Robotics Research*, vol. 25, no. 7, pp. 669–687, 2006.

[9] W. J. Schwind, *Spring loaded inverted pendulum running: A plant model.* University of Michigan, 1998.

[10] R. Blickhan and R. Full, "Similarity in multilegged locomotion: bouncing like a monopode," *Journal of Comparative Physiology A*, vol. 173, pp. 509–517, 1993.

[11] T. J. Roberts and E. Azizi, "Flexible mechanisms: the diverse roles of biological springs in vertebrate movement," *Journal of experimental biology*, vol. 214, no. 3, pp. 353–361, 2011.

[12] D. Lakatos, K. Ploeger, F. Loeffl, D. Seidel, F. Schmidt, T. Gumpert, F. John, T. Bertram, and A. Albu-Schäffer, "Dynamic locomotion gaits of a compliantly actuated quadruped with slip-like articulated legs embodied in the mechanical design," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3908–3915, 2018.

[13] M. H. Raibert, "Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 14, no. 3, pp. 451–463, 1984.

[14] S.-H. Hyon and T. Emura, “Energy-preserving control of a passive one-legged running robot,” *Advanced robotics*, vol. 18, no. 4, pp. 357–381, 2004.

[15] M. Ahmadi and M. Buehler, “Stable control of a simulated one-legged running robot with hip and leg compliance,” *IEEE transactions on robotics and automation*, vol. 13, no. 1, pp. 96–104, 1997.

[16] C. François and C. Samson, “A new approach to the control of the planar one-legged hopper,” *The International Journal of Robotics Research*, vol. 17, no. 11, pp. 1150–1166, 1998.

[17] D. Calzolari, C. Della Santina, A. M. Giordano, and A. Albu-Schäffer, “Single-leg forward hopping via nonlinear modes,” in *2022 American Control Conference (ACC)*, pp. 506–513, IEEE, 2022.

[18] C. Della Santina, D. Lakatos, A. Bicchi, and A. Albu-Schaeffer, “Using nonlinear normal modes for execution of efficient cyclic motions in articulated soft robots,” in *Experimental Robotics: The 17th International Symposium*, pp. 566–575, Springer, 2021.

[19] D. Lakatos, W. Friedl, and A. Albu-Schäffer, “Eigenmodes of nonlinear dynamics: Definition, existence, and embodiment into legged robots with elastic elements,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1062–1069, 2017.

[20] R. Goedel, R. G. Sanfelice, and A. R. Teel, “Hybrid dynamical systems: modeling stability, and robustness,” *Princeton, NJ, USA*, 2012.

[21] I. A. Hiskens, “Stability of limit cycles in hybrid systems,” in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pp. 6–pp, IEEE, 2001.

[22] S.-H. Hyon and T. Emura, “Quasi-periodic gaits of passive one-legged hopper,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2625–2630, IEEE, 2002.

A Reference tables for the periodic gaits

A.1 Pitching-SLIP

Tables 12-14 report the numerical values of the states at the apex (AP), touch-down (TD+), bottom (BOT) and lift-off (LO-) of three different periodic gaits of the pitching-SLIP.

	AP	TD+	BOT	LO-
x_c (m)	0.0000	0.7464	0.8857	1.0249
$\dot{x}_c$ (m/s)	13.1966	13.1966	13.0176	13.1966
z_c (m)	0.2003	0.1846	0.1807	0.1846
$\dot{z}_c$ (m/s)	0.0000	-0.5549	0.0000	0.5549
r (m)	0.2312	0.2312	0.1807	0.2312
$\dot{r}$ (m/s)	0.0000	-8.3884	0.0005	8.3884
α (rad)	3.1416	0.6461	0.0000	-0.6461
$\dot{\alpha}$ (rad/s)	-44.1199	-44.1206	-72.0475	-44.1206

Table 12: Unstable gait of the pitching-SLIP. $E = 89.6633$ J, $l_f = -2.3757\text{kg} \cdot \text{m}^2/s$.

	AP	TD+	BOT	LO-
x_c (m)	0	0.7371	0.8777	1.0183
$\dot{x}_c$ (m/s)	18.1294	13.1966	17.9989	18.1294
z_c (m)	0.2003	0.1836	0.1815	0.1836
$\dot{z}_c$ (m/s)	0.0000	-0.3989	0.0000	0.3989
r (m)	0.2312	0.2312	0.1815	0.2312
$\dot{r}$ (m/s)	0.0000	-11.3409	-0.0005	11.3409
α (rad)	3.1416	0.6536	0.0000	-0.6536
$\dot{\alpha}$ (rad/s)	-61.1913	-61.1907	-99.1710	-61.1907

Table 13: Stable gait of the pitching-SLIP. $E = 167.3822$ J, $l_f = -2.3757\text{kg} \cdot \text{m}^2/s$.

	AP	TD+	BOT	LO-
x_c (m)	0.0000	1.0948	1.2049	1.3149
$\dot{x}_c$ (m/s)	3.9925	3.9925	3.3346	3.9925
z_c (m)	0.5722	0.2034	0.1500	0.2034
$\dot{z}_c$ (m/s)	0.0000	-2.6901	0.0000	2.6900
r (m)	0.2312	0.2312	0.1500	0.2312
$\dot{r}$ (m/s)	0.0000	-4.2661	0.0000	4.2661
α (rad)	3.1416	0.4960	0.0000	-0.4961
$\dot{\alpha}$ (rad/s)	-9.6476	-9.6477	-22.2285	-9.6477

Table 14: "Very unstable" gait of the pitching-SLIP. $E = 13.6785$ J, $l_f = -0.5195\text{kg} \cdot \text{m}^2/s$.

A.2 Body-SLIP

Tables 15-16 report the numerical values of the states at the apex (AP), touchdown (TD+), bottom (BOT) and lift-off (LO-) of two different periodic gaits of the body-SLIP.

	AP	TD+	BOT	LO-
x_c (m)	0.0000	0.7607	0.8897	1.0179
$\dot{x}_c$ (m/s)	8.6373	8.6373	8.3000	8.6385
z_c (m)	0.2303	0.1923	0.1836	0.1921
$\dot{z}_c$ (m/s)	0.0000	-0.8640	0.0000	0.8541
r (m)	0.2312	0.2312	0.1836	0.2312
$\dot{r}$ (m/s)	0.0000	-5.5167	0.0235	5.5167
α (rad)	3.1416	0.5890	-0.0028	-0.5901
$\dot{\alpha}$ (rad/s)	-28.9829	-28.9824	-28.9824	-28.9830

Table 15: Unstable gait of the body-SLIP. $E = 40.2824$ J, $l_f = -1.5896\text{kg} \cdot \text{m}^2/s$.

	AP	TD+	BOT	LO-
x_c (m)	0.0000	0.7391	0.8737	1.0074
$\dot{x}_c$ (m/s)	12.8641	12.8641	12.5661	12.8644
z_c (m)	0.2046	0.1884	0.1844	0.1883
$\dot{z}_c$ (m/s)	0.0000	-0.5636	0.0000	0.5564
r (m)	0.2312	0.2312	0.1844	0.2312
$\dot{r}$ (m/s)	0.0000	-7.9174	0.0359	7.9174
α (rad)	3.1416	0.6184	-0.0029	-0.6190
$\dot{\alpha}$ (rad/s)	-43.9146	-43.9135	-68.1449	-43.9137

Table 16: Unstable gait of the body-SLIP. $E = 86.3312$ J, $l_f = -2.4086\text{kg} \cdot \text{m}^2/s$.

B Flight torques profiles

B.1 Piece-wise constant profile

The expressions here reported are taken from [14], with adapted signs to be consisted with the model used in this work. Defining the new variables:

$$\psi = \alpha - \phi \tag{98}$$

and

$$\sigma = J_b\phi + J_l\alpha, \tag{99}$$

the values for τ_1 and τ_2 can be computed as:

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = B(T_f)^{-1}P\left(\begin{bmatrix}\bar{\psi}\\ \bar{\dot{\psi}}\end{bmatrix} - A(T_f)\begin{bmatrix}\psi_{lo}\\ \dot{\psi}_{lo}\end{bmatrix}\right) \tag{100}$$

where

$$\begin{bmatrix}\bar{\psi}\\ \bar{\dot{\psi}}\end{bmatrix} = \frac{1}{J_b}\left((J_b + J_l)\begin{bmatrix}\alpha_{td,des}\\ \dot{\alpha}_{td,des}\end{bmatrix} - C(T_f)\begin{bmatrix}\sigma_{lo}\\ \dot{\sigma}_{lo}\end{bmatrix}\right) \tag{101}$$

and:

$$A(T_f) = \begin{bmatrix}\cos(\Omega_h T_f) & \frac{1}{\Omega_h}\sin(\Omega_h T_f)\\ -\Omega_h\sin(\Omega_h T_f) & \cos(\Omega_h T_f)\end{bmatrix}, \tag{102}$$

$$B(T_f) = \frac{1}{K_h}\begin{bmatrix}\cos(\Omega_h T_f) - \cos\left(\frac{\Omega_h T_f}{2}\right) & \cos\left(\frac{\Omega_h T_f}{2}\right) - 1\\ \Omega_h\sin(\Omega_h T_f) - \Omega_h\sin\left(\frac{\Omega_h T_f}{2}\right) & \Omega_h\sin\left(\frac{\Omega_h T_f}{2}\right)\end{bmatrix}, \tag{103}$$

$$C(T_f) = \begin{bmatrix}1 & T_f\\ 0 & 1\end{bmatrix}, \quad P = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \tag{104}$$

B.2 Triangular torque profile

To compute τ_A and τ_B so that the control input achieves the desired $\alpha_{TD,des}$ and $\dot{\alpha}_{TD,des}$ in time T_f, the flight dynamics of the angles (93) can be integrated from time 0 to T_f with the triangular profile (92) as input. Considering the initial conditions to be known (they are the angles and angular velocities at lift-off), one obtains two equations in two variables. The system can then be solved for τ_A and τ_B.

The expressions to compute τ_A and τ_B are:

$$
\begin{aligned}
\tau_A = \frac{\Omega_h}{\sqrt{J_b+J_l}} T_f (& J_b\sqrt{J_b+J_l}K_h \cos\left(\Omega_h \frac{T_f}{4}\right)^3 (\phi_{lo}-\alpha_{lo}) + \\
& + \sqrt{J_b^3 J_l K_h} \sin\left(\Omega_h \frac{T_f}{4}\right) \cos\left(\Omega_h \frac{T_f}{4}\right)^2 (\dot{\phi}_{lo}-\dot{\alpha}_{lo}) + \\
& + (-K_h((\phi_{lo} - \frac{\alpha_{td,des}}{4} - \frac{3}{4}\alpha_{lo})J_b - \frac{J_l}{4}(\alpha_{td,des}-\alpha_{lo}))\sqrt{J_b+J_l} + \\
& - \frac{1}{4}\Omega_h T_f \sqrt{Kh}(\sqrt{J_b^3 J_l}\dot{\phi}_{lo} + \sqrt{J_l^3 J_b}\dot{\alpha}_{lo})) \cos\left(\Omega_h \frac{T_f}{4}\right) + \\
& + \frac{1}{4}\sin\left(\Omega_h \frac{T_f}{4}\right) \sqrt{K_h}(\dot{\alpha}_{lo} - \dot{\alpha}_{td,des})(\sqrt{J_l^3 J_b} + \sqrt{J_b^3 J_l}))\cdot \\
& \cdot \left(4J_b \cos\left(\Omega_h \frac{T_f}{4}\right) \sin\left(\Omega_h \frac{T_f}{4}\right)\left(\cos\left(\Omega_h \frac{T_f}{4}\right) - 1\right)\right)^{-1}
\end{aligned} \tag{105}
$$

$$
\begin{aligned}
\tau_B = \frac{\Omega_h}{\sqrt{J_b+J_l}} T_f (& (K_h((\phi_{lo}-\alpha_{td,des})J_b - J_l(\alpha_{td,des}-\alpha_{lo}))\sqrt{J_b+J_l} + \\
& + \Omega_h T_f \sqrt{K_h}(\sqrt{J_b^3 J_l}\dot{\phi}_{lo} + \sqrt{J_l^3 J_b}\dot{\alpha}_{lo})) \cos\left(\Omega_h \frac{T_f}{4}\right)^3 + \\
- \sin\left(\Omega_h \frac{T_f}{4}\right) & \sqrt{K_h}(\sqrt{J_b^3 J_l}(\dot{\phi}_{lo} - \dot{\alpha}_{td,des}) + \sqrt{J_l^3 J_b}(\dot{\alpha}_{lo} - \dot{\alpha}_{td,des})) \cos\left(\Omega_h \frac{T_f}{4}\right)^2 + \\
& + (-K_h((\phi_{lo} - \frac{3}{4}\alpha_{td,des} - \frac{\alpha_{lo}}{4})J_b - \frac{3}{4}J_l(\alpha_{td,des}-\alpha_{lo}))\sqrt{J_b+J_l} + \\
& - \frac{3}{4}\Omega_h T_f \sqrt{K_h}(\sqrt{J_b^3 J_l}\dot{\phi}_{lo} + \sqrt{J_l^3 J_b}\dot{\alpha}_{lo})) \cos\left(\Omega_h \frac{T_f}{4}\right) + \\
& + \frac{1}{4}\sin\left(\Omega_h \frac{T_f}{4}\right) \sqrt{K_h}(\dot{\alpha}_{lo} - \dot{\alpha}_{td,des})(\sqrt{J_l^3 J_b} + \sqrt{J_b^3 J_l}))\cdot \\
& \cdot \left(4J_b \cos\left(\Omega_h \frac{T_f}{4}\right) \sin\left(\Omega_h \frac{T_f}{4}\right)(\cos\left(\Omega_h \frac{T_f}{4}\right) - 1)\right)^{-1}
\end{aligned} \tag{106}
$$