A template-based control architecture for dynamic legged locomotion

Shahbazi Aghbelagh, Mohammad

DOI
10.4233/uuid:baeca8ab-8534-4006-bb4a-d7c36921150a

Publication date
2016

Document Version
Final published version

Citation (APA)
Shahbazi Aghbelagh, M. (2016). *A template-based control architecture for dynamic legged locomotion*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:baeca8ab-8534-4006-bb4a-d7c36921150a

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

A Template-Based Control Architecture for Dynamic Legged Locomotion

Mohammad Shahbazi Aghbelagh

A Template-Based Control Architecture for Dynamic Legged Locomotion

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op

woensdag 7 december 2016 om 12:30 uur

door

Mohammad SHAHBAZI AGHBELAGH

Master of Science in Mechanical Engineering
K. N. Toosi University of Technology, Iran
geboren te Teheran, Iran

This dissertation has been approved by the
Promotor: Prof. dr. R. Babuška
Copromotor: Dr. G. A. Delgado Lopes

Composition of the doctoral committee:

Rector Magnificus,	chairman
Prof. dr. R. Babuška,	Technische Universiteit Delft, promotor
Dr. G. A. Delgado Lopes,	Technische Universiteit Delft, copromotor

Independent members:

Prof. dr. A. Seyfarth,	Technische Universität Darmstadt
Prof. dr. ir. M. Wisse,	Technische Universiteit Delft
Prof. dr. ir. H. van der Kooij,	Technische Universiteit Delft
Dr. R. Carloni,	Universiteit Twente

Other member:

Dr. U. Saranlı,	Middle East Technical University

This dissertation has been completed in fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate study.

The requirement of the TU Delft Graduate School for the Doctoral Education Program has been fulfilled.

ISBN 978-94-6186-750-6

Printed in the Netherlands.

Dedicated to *Morteza Aatlokhanlou* (1968–1987)
student martyred in Iran's Holy Defense

Preface

In the name of Allah, the compassionate, the merciful

The present thesis is the result of four and a half years research done in the Delft Center for Systems and Control (DCSC), Delft University of Technology, in partial fulfillment of the requirements for the PhD since March 2012. While only my name appears as the author of this book, it is most certainly not the result of individual work.

First of all, I would like to gratefully acknowledge the support of the Government of the Islamic Republic of Iran, Ministry of Science, Research and Technology.

Next, I would like to thank my promotor Prof. Robert Babuška for giving me the opportunity to do a PhD in DCSC. I highly appreciate his effort put forth in helping me to improve my research skills, and in backing me up whenever needed, even during weekends and holidays. I owe also a great debt of gratitude to my daily supervisor Dr. Gabriel Lopes. His friendship manner and kindness during our long technical (and non-technical) discussions are unforgettable. I would like to extend my gratitude to Dr. Uluç Saranlı, Associate Professor of computer engineering in METU, for his great support and feedback during a part of my PhD research, and also Prof. Edward R. Valstar for his fascinating and helpful mentoring.

During the years I worked in DCSC, I have found many good friends and enjoyed being with them in a couple of social events. The DCSC atmosphere was warm, friendly, and dynamics. As I am afraid if I forget a name, I prefer to mention no specific name here, but I am deeply grateful to all of my DCSC colleagues.

The summary of the thesis was translated to Dutch with the great help of Reinier Doelman and Tijmen van Oldenrijk. Hereby, I would like to appreciate their effort and time.

Next, I would like to kindly thank my closest friend Esmaeil Najafi for his motivation, support, and kindness. He has been a great support to me since our undergraduate study. Particularly, Esmaeil helped me a lot during my movement to Delft, and I am deeply grateful to him.

My special appreciation is reserved for my Iranian friends and their respected families in the Netherlands. Enjoyable moments my family and me had with them during weekly meetings and the special events of Hey'at Mohebban Al-Mahdi did help us not only in satisfying our spiritual need, but also in relieving the pain of missing our parents and relatives while living abroad. Here again I prefer to not

mention specific families because of the risk of forgetting certain names I should have mentioned.

I would like to express my greatest appreciation to my parents. I really thank God for having such a patient and self-giving parent, who did their best to support me throughout my life. I know that they had a tough time missing us during our stay abroad, but they did patiently cope with the situation. Words do not suffice to express the gratitude and love I feel toward, but I wish them a long and healthy life. I also deeply thank my kind sisters and their respected families for their support, kindness, and blessing.

Last but not least, I would like to express my deepest thanks to my wife, who has closely experienced the full effects of my PhD. There were too many discouraged times and late nights put to meet deadlines. Yet through it all, she has been incredibly supportive and patient. Every single achievement of my PhD studies owes also to her support, dedication, and understanding. I would like to kindly thank also her respected family for their understanding, and blessing. I am writing these last words for our honey daughter for the time she can read by herself: My dear daughter, thank you very much for all the great time you have provided for us. I bless God for giving you to me and your mother during our stay abroad.

Mohammad Shahbazi Aghbelagh
November 2016

Contents

Chapter

1

Introduction

This chapter introduces the subject and scope of the thesis. The existing control approaches to the problem of dynamic legged locomotion are categorized, and the particular methodology that we develop in this thesis is introduced. The outline of the thesis is then presented, and the chapter is concluded by highlighting the original contributions of the thesis.

1.1 Motivation

Legged animals and humans are able to traverse most of the landmasses on the earth [1]. This unique mobility feature has attracted a large body of research both in biomechanics (to gain a better understanding of legged locomotion), and in robotics (to design and control more efficient bio-inspired and bio-mimetic robots). The usefulness of such legged robots, hence, depends on the scientific and technological advances in those areas of research.

Despite great developments, disaster scenarios, such as the Fukushima nuclear accident, reveal that there is still much to be done to match the flexibility, agility, and speed of many land animals and humans, when it comes to traversing unstructured/rough terrain. An important reason is that man-made sensors,

actuators, and morphologies do not match the performance and specialization of their counterparts in nature (the importance of the *body*). Perhaps an even more important reason is constituted by the challenges associated with designing perception and control systems (the *brain*), particularly when dynamic, efficient, and autonomous platforms are desired.

Several task requirements such as gait pattern generation, footstep planning, and dynamically consistent low-level structures challenge the success of a locomotion control system at different levels. It is not straightforward to directly apply modern control theory to legged locomotion [2, 3]. The major difficulties arise from the intrinsic instability of an upright posture and from the hybrid nature of a legged mechanism. At the motion planning level, an effective implementation of planning algorithms for such robots demands substantial considerations at the technical development stage as well as computational efforts needed, especially in the presence of uncertainties and terrain irregularities. As such, the major part of planning algorithms is traditionally applied offline, possibly making the controller task-specific and susceptible to robustness issues. In the following section, we first give an overview of the literature on how these problems are approached from different points of view, and then elaborate on the particular approach followed in this thesis.

1.2 Spectrum of control methods in legged locomotion

As discussed above, legged locomotion in unstructured terrain is a difficult task to be implemented in a system with complex dynamics. To design and analyze such control systems, intuitive approaches inspired by biology alone have shown only limited success. Traditionally, the scientific community has followed two different approaches, one focusing on the task specialties (typically a topic of computer science), and the other on dealing effectively with the dynamics complexities (the realm of control).

In computer science, sophisticated planning algorithms for robots performing high-level tasks, even in unstructured environments have been extensively studied (see, for example, [4–7]); however, the considered robot dynamics are often too simplified. In this group of studies, the control laws are usually based on optimization problems in which the whole computation is carried out at once. Although impressive results are achieved in simulation, the direct implementation of such controllers on a real robot is not easy due to the overly simplified models considered, especially for the contact dynamics. Moreover, it is still challenging to

implement the developed planning algorithms online due to the extensive computations needed. Finally, no guarantee for feasibility or optimality of the solution can generally be given, which is typical for all numerical nonlinear optimization methods.

On the other side of the spectrum, when the focus is put on the handling of complexities of legged robots dynamics with the aim of experimental demonstrations, the tasks being accomplished are relatively simple. The resulting control systems are typically parameterized by variables that are governed by a high-level supervisory controller. Various existing locomotion control methods in the literature can be framed in this perspective, including ZMP [8], Virtual model control [9], Output function [10] and Capture point [11]. Note that the control laws in this setting are often task-specific in the supervisory level, possibly restricting the autonomy of the robot, although their effectiveness in accomplishing the particular task for which they are designed can be demonstrated experimentally.

For a legged platform intended to robustly negotiate unexpected situations possibly in unstructured environments (e.g., in a search and rescue scenario), a sufficient level of autonomy is a key requirement. This in turn demands sufficient flexibility at the planning level, i.e., the planning algorithm should be online implementable. Ideally, the process of motion planning and its refinement on the robot through low-level control structures should be systematic, in the sense that the human intervention in the process at runtime should be minimized.

1.3 Scope of the thesis

Towards satisfying the above-mentioned specifications, the second approach can be extended by partitioning the problem into various layers. The structure of such a methodology together with the two previously discussed approaches are schematically depicted in Figure 1.1[1]. The multi-layer scheme is inspired by the well-established notion of *template* and *anchor*, a bio-inspired abstraction/refinement approach to the problem of motion planning and control in legged robotics [12]. The study of legged locomotion using template (i.e., simplified) models provides insights to the dominant features of the dynamics while abstracting out less important details at the planning level [13]. When properly devised, the template layer enables the execution of the majority of planning computations online, whose results are then anchored in the real robot at the lower layers.

In our view, this hierarchical mechanism can contain multiple levels, depending

[1] Although the multi-layer scheme originally belongs to the second approach, we depicted it separately in this figure to highlight its characteristics.

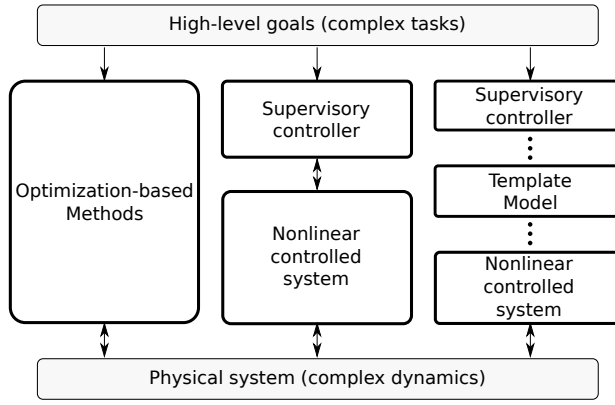


Figure 1.1: Categorization of the control approaches in legged locomotion.

on the structure of the robot and the type of tasks at hand. Ideally, when a layer is added to/removed from the structure, most of the other elements should remain the same, while only the neighboring layers prepare for the new situation. This will open a possibility for systematic generation of legged locomotion.

A number of studies addressing the control of dynamic legged locomotion can be interpreted according to the multi-layer scheme [14–19]. The general idea is to coordinate the actuated degrees of freedom of the real (high dimensional) robot in such a way that it behaves like the template (lower dimensional) model. This can enforce the real robot to mimic the template center of mass (CoM) motion [17], or may utilize the template as a target (i.e., the *hybrid zero dynamics* [20]) for asymptotic behavior of the real robot [14]. In both cases, there is a continuous coordination between the template model and the controlled system, the feature that might be unnecessary in light of real-time calculations. Moreover, the potential usefulness of the multi-layer architecture in addressing quadrupedal steady and transitional running was remained almost unexplored[2].

Motivated by these observations, the present thesis aims to answer the following main research question:

- **How to systematically develop a control architecture for robust and autonomous dynamic legged locomotion?**

As discussed above, we believe that a template-based multi-layer control architec-

[2] As will be discussed in Chapter 5, studies in this regard are mostly limited to the template level. Although it appears that quadrupedal robots developed by Boston Dynamics utilize the multi-layer scheme in their control system, the associated technical details have not been released publicly at the time of writing this thesis.

ture is the "right" approach to this problem. Its hierarchical and modular structure allows a deep understanding of the process of control in different levels, hence making it systematically extensible with respect to the tasks at hand. Moreover, since it is based on a template model, it allows the motion planning algorithms to be implemented online, the feature that is a key factor for autonomous reactions of a robot in unexpected situations.

The success of such a multi-layer scheme is however subject to addressing several questions at different levels:

1. How to improve online implementability of a controller designed for the template?

2. How to develop a unified controller at the template layer that performs different dynamic gaits and gait transitions?

3. What would be a good template for quadrupedal running? In other words, could a template developed for bipedal running be sufficiently descriptive also for quadrupedal running?

To answer the first question, we take an analytical approach to the dynamics of the considered template model: the standard spring-mass model, also referred to as the spring-loaded inverted pendulum (SLIP). While such a study for SLIP running has received much attention in the literature, the SLIP walking dynamics remained unexplored. We present novel approximate analytical solutions to the otherwise non-integrable double-stance (DS) dynamics of SLIP.

To address the second question, the analytic simplicity of the derived solutions is then used to develop a class of unified controllers that, in particular, automate the walk-run transitions in the SLIP model. Subsequently, at a lower layer, we embed the controlled SLIP walking into a high dimensional realistic bipedal model.

As for the third question, this thesis introduces a novel template for the study of quadrupedal steady and transitional running, called the dual-SLIP model. In doing so, we develop also a coordination controller for the synchronization of multiple template models. This high-level controller is particularly applied to the compositional treatment of two SLIPs in the dual-SLIP model towards achieving relatively complex locomotion patterns in quadrupedal running.

1.4 Organization of the thesis

Chapter 2 describes different aspects of the SLIP model. Particularly, we introduce a hybrid automaton that unifies modeling and control of walking and running.

Next, the analytical approaches to the dynamics of SLIP are reviewed, and our novel approximations, presented in the next chapter, are motivated. Additionally, we review the existing modeling frameworks for legged locomotion in the literature.

Chapter 3 presents novel approximations to the DS dynamics of lossy SLIP, with an aim to derive a computationally fast and analytically tractable solution. Since the proposed solutions are obtained through approximations, the influence of approximation errors is assessed through an extensive numerical analysis, and the role of the resulting DS solutions in the whole walking cycle is discussed.

Chapter 4 presents the development of a class of unified controllers that realize active walking, running, and biologically inspired walk-run transitions within the framework developed in Chapter 2 and by utilizing the DS approximations introduced in Chapter 3. The effect of approximations on the stability properties of the controlled system is carefully analyzed. A number of simulation tests are provided to assess the utility and performance of the proposed framework, and finally the controlled SLIP information is used in motion planning of a multi-body robot model.

Chapter 5 studies the functionality of the presented multi-layer architecture in the context of quadrupedal motion planning. A new (generic) coordination controller for synchronization of (multiple) template models is developed, using the max-plus algebra. Subsequently, a novel template for quadrupedal steady and transitional running is introduced, by composing two physically-unconnected SLIP models. Finally, a number of simulation experiments on the proposed template are provided: the realization of quadrupedal pronking, bounding, and the respective transition; and the robustness evaluation against ground height variations.

Finally, **Chapter 6** summarizes the merits of the thesis outcomes, outlines the main conclusions of the thesis, and recommends a number of possible extensions for future research.

1.5 Original contributions of the thesis

In what follows we highlight the individual contributions made in each chapter of the thesis along with the corresponding publications.

- **Chapter 2:**
  - A unified notation for the study of walking and running gaits [21];
  - The hybrid automaton representation for different gaits and gait transitions for the SLIP model [21, 22];
- **Chapter 3:**

  - The introduction of Axial-Torsional SLIP (AT-SLIP) model and the derivation of its parameters so as to make it approximately equivalent to the original SLIP model in DS phase [23, 24];
  - Analytical approximation of the SLIP dynamics in DS phase, in the presence of non-negligible damping [23];
  - An approximate closed-form solution for the lossless SLIP dynamics in DS phase using standard perturbation techniques [23];

- **Chapter 4:**
  - A unified class of controllers for realization of active walking and running on the SLIP model [21];
  - Automated transitions between walking and running on the SLIP model [21, 22];
  - Stability analysis of deadbeat-like controllers for SLIP walking when the derived dynamics approximations are used [21];
  - Embedding of SLIP-like motions in a higher dimensional (realistic) bipedal model [21];

- **Chapter 5:**
  - The introduction of the dual-SLIP model as a template for quadrupedal steady and transitional running [25];
  - The development of a max-plus based coordination controller for intermittent contact systems [25, 26];
  - The presentation of a “time-aware" deadbeat controller for SLIP running [25].

Chapter

2

Preliminaries and Framework Establishment

In the previous chapter, the existing control methods in the field of legged locomotion were categorized. We distinguished the particular method aimed to be developed in this thesis, in which the template models are of particular importance. The present chapter describes different aspects of the spring-loaded inverted pendulum (SLIP), the standard template model for dynamic legged locomotion (Section 2.2). Particularly, we introduce a hybrid automaton in Section 2.3 that well describes the unified framework developed for the modeling and control of walking and running. Next, the analytical approaches to the dynamics of SLIP are reviewed, and our novel approximations, presented in the next chapter, are motivated (Section 2.4). Additionally, we review the existing modeling frameworks for legged locomotion in the literature with an emphasis put on the recently developed methodology utilizing the max-plus algebra in Section 2.5. The chapter is concluded in Section 2.6 by summarizing the discussed aspects.

2.1 Introduction

The design, construction, and control of legged robots capable of performing agile and dynamic motions are challenging [2]. This is due to the complexities introduced by the intrinsic upright instability and the hybrid nature of legged locomotion. As discussed in Chapter 1, researchers typically consider simplified (abstract) representations, that capture only the essential characteristics of the system under study, thereby helping to gain a better understanding of the underlying principles of legged locomotion. A noteworthy formal definition of this idea is the notion of *template* and *anchor* introduced by Full and Koditschek [12].

Following this definition, the spring-loaded inverted pendulum (SLIP), which was first recognized in biomechanics as a good descriptive model of animals' hopping and running [27–30], can be interpreted as a template for the running gait [31]. Starting with Raiberts' hoppers [1], which were based on an intuitive exploitation of SLIP behaviors, a number of dynamic robots have been introduced (e.g., [16, 32–35]). Some of these robots were designed with no SLIP-like morphologies; however, they use SLIP as their target dynamics in the control calculations, extending the utility of SLIP to a broader scope.

Recently, Geyer et al. [36] studied the potential of describing the walking gait with compliant legs on the "bipedal" SLIP model[1] (see Figure 2.1). Contrary to the classical inverted pendulum, which is a widely-used template for the class of walkers that are referred to as limit cycle walking robots [37], the SLIP is able to model non-instantaneous DS phases. Moreover, the ground reaction forces produced in the SLIP are closer to the human data reported in [36]. Subsequently, the idea of realizing dynamic walking has been studied both on the SLIP model itself [38, 39], and on more complex robot models through the embedding of SLIP behaviors [17, 40, 41], showing the need for further investigations.

The primary goal of this chapter is to establish a framework in which the modeling and control of SLIP walking, running and their transitions can be studied in a unified scheme. One potential benefit such a framework can offer is the ability to perform online planning. This is a crucial requirement for robust and reliable reactions of autonomous robots negotiating unexpected situations. Here, an analytical approach to the system dynamics seems useful since numerical integration of the equations of motion for most legged robots is not preferable due to its time-consuming nature[2]. An analytical time-domain solution could also be

[1] Throughout this thesis, the SLIP model considered is bipedal unless otherwise specified.

[2] Note that offline planners can also be used online through lookup tables. However, the effectiveness of this approach is limited to the precomputed values for the space of system states that may not cover unforeseen situations.

useful in analytically deriving the associated Jacobian matrix.

Unfortunately, it is known that the exact analytical time-domain solution to the SLIP equations of motion in the stance phase is not available [42]. For the single-stance (SS) phase, however, a number of analytical approximations have been proposed [43–48]. In the present chapter we review this body of literature with the aim to motivate the first approximate solutions to the DS dynamics of SLIP we introduce in Chapter 3.

We also present a review of frameworks for modeling of legged locomotion. In particular, we describe the methodology that approaches the problem from a fresh point of view by utilizing the max-plus algebra. This material is meant as background information for the readers unfamiliar with the modeling frameworks and methodologies for multi-legged robots (with many legs) and the theory of max-plus algebra.

2.2 Spring-Loaded Inverted Pendulum (SLIP)

2.2.1 Model description and dynamics

The SLIP model in the walking and running gaits is illustrated in Figures 2.1a and 2.1b, respectively (see Table 2.1 for the general notation used in this thesis). The body is represented as a point mass m at the CoM that is connected to two mass-less springy legs, moving in the sagittal plane with gravitational acceleration g. The legs are represented by linear lossless springs with the nominal constant k_0 and rest length l_{rest}. The system motion is measured with respect to the fixed coordinate frame $\mathcal{W}$. As depicted in Figure 2.1, it is convenient to further define polar coordinates (r, θ) at the toe position.

As common in the literature, the considered SLIP model is assumed to have a point foot, thereby no actuation at the toe is permitted. We also ignore the effect of the touchdown impact, and assume the toe position remains fixed during stance (no slipping). Moreover, we assume no physical meaning for the swing leg, thereby ignoring foot scuffing.

In general, the continuous and discrete state vector of the system in the Cartesian coordinates (x, y) can be expressed as $s = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T$, $s \in \mathcal{S} \subset \mathbb{R}^4$, and $q = \begin{bmatrix} x_{\text{h}} & x_{\text{f}} & \alpha^i & \mathcal{M} \end{bmatrix}^T$, respectively. x_{h} and x_{f} denote the hind (h) and fore (f) foot positions, α^i, $i \in \{\text{r}, \text{w}\}$, are the touchdown angles of running and walking gaits, and $\mathcal{M} \in \{\text{F}, \text{SS}, \text{DS}\}$ denote the primary phases of motion defined as follows. Depending on how the springs act on the body, three primary phases can be distinguished: swing or flight (F), single-stance (SS) and double-stance (DS).

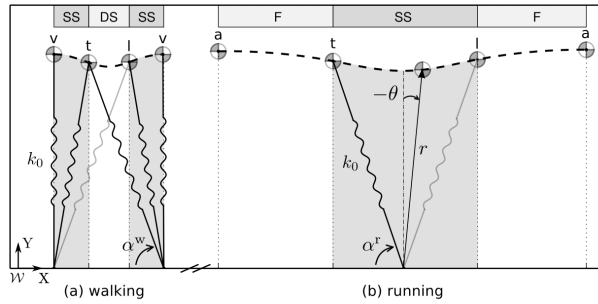


Figure 2.1: SLIP model in walking and running gaits. See Table 2.1 for the notation. The corresponding single-stance phases are shaded in both gaits.

As no spring acts on the body in the flight phase, the system is driven by gravity, following a ballistic trajectory. The system undergoes oscillatory motions in the SS and DS phases due to the influence of a single and double spring, respectively, interacting with gravity. The equations describing the CoM motion in Cartesian coordinates (x, y) are written as

- flight (F):

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \end{bmatrix}, \tag{2.1}$$

- single-stance (SS):

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \end{bmatrix} + T_{\mathrm{SS}_i}(s,q) F_{\mathrm{l}_i}(s,q),\ i \in \{\mathrm{h},\mathrm{f}\}, \tag{2.2}$$

- double-stance (DS):

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \end{bmatrix} + T_{\mathrm{DS}}(s,q) \begin{bmatrix} F_{\mathrm{l_h}}(s,q) \\ F_{\mathrm{l_f}}(s,q) \end{bmatrix}, \tag{2.3}$$

where T_{SS_i} and T_{DS} are transformation matrices, and F_{l_i} are the legs spring force:

$$\begin{aligned} T_{\mathrm{SS}_i}(s,q) &= \begin{bmatrix} -\sin\theta_i(s,q) \\ \cos\theta_i(s,q) \end{bmatrix}, \quad i \in \{\mathrm{f},\mathrm{h}\}, \\ T_{\mathrm{DS}}(x,y) &= \begin{bmatrix} -\sin\theta_{\mathrm{h}}(s,q) & -\sin\theta_{\mathrm{f}}(s,q) \\ \cos\theta_{\mathrm{h}}(s,q) & \cos\theta_{\mathrm{f}}(s,q) \end{bmatrix}, \\ F_{\mathrm{l}_i}(s,q) &= k_0(l_{\mathrm{rest}} - l_i(s,q)), \quad \text{for } l_i \le l_{\mathrm{rest}}, \quad i \in \{\mathrm{f},\mathrm{h}\}, \end{aligned}$$

with l_i being the legs length. Notice that θ_i are measured from the vertical in the counter-clockwise direction (see Figure 2.1).

In the running gait the system alternates between the F and SS phases. Two important events define the phase transitions: *touchdown* and *liftoff*. The touchdown event occurs when the fully stretched leg touches the ground (assuming flat terrain) with touchdown angle α^{r}, and the leg lifts off the ground when it is fully stretched (i.e., reaches its rest length) in the spring decompression. A single running step starts at the highest vertical position of the CoM in the F phase, called the *apex* and ends at the subsequent apex. We further divide the SS phase into the spring compression and decompression subphases. The *bottom* event, where the spring is at its minimum length, triggers the transition from the compression to decompression subphases.

In the walking gait the system alternates between the SS and DS phases. Here the fore leg touchdown and the hind leg liftoff define the phase transitions. The system switches from SS to DS when the fully stretched leg touches the ground making α^{w} as the touchdown angle, and switches back to the next SS when the hind leg is fully stretched. A single walking step starts at the vertical leg orientation (VLO) in the SS phase, and ends at the subsequent VLO. Similarly to the running gait, we further divide the DS phase into the "virtual" compression and decompression subphases. The corresponding transitions between these two are defined by the so-called *virtual bottom* event. The virtual bottom is the point where the CoM has the minimum distance from the mid-stance of the legs' toe[3]. We explain in Chapter 4 why the virtual bottom is of interest.

2.2.2 Return maps

The Poincaré map [49] is a commonly used tool for the analysis of periodic systems, such as the SLIP model in walking and running, which reduces the dimension of the system state via a discrete task-space abstraction in the Poincaré section. For the running gait we take the Poincaré section at apex [50] by defining the transversal Σ^{r} to the orbit Γ^{r} (see Figure 2.2). The reduced system state at this section z_{a} contains only two variables

$$z_{\mathrm{a}} = \begin{bmatrix} \dot{x}_{\mathrm{a}} & y_{\mathrm{a}} \end{bmatrix}^T, \; z_{\mathrm{a}} \in \Sigma^{\mathrm{r}} \subset \mathbb{R}^2, \tag{2.4}$$

We do not include the CoM horizontal position x_{a} in z_{a} because it has no influence on the system dynamics at the Poincaré section from one return to another, when traversing a flat surface; however, we do keep track of it within each locomotion step because of the toes' position and the transition guards. Moreover, the definition

[3] The SLIP walker can experience multiple virtual bottom events. In this thesis, we focus on the human-like walking patterns (i.e., with the vertical ground reaction forces having an M shape profile), and therefore a single virtual bottom is experienced.

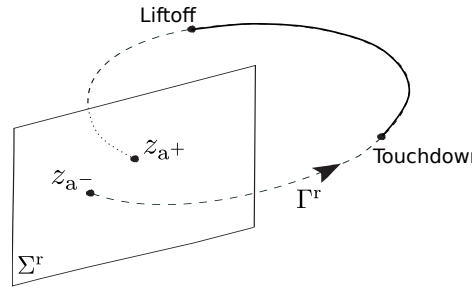


Figure 2.2: Abstract view of the Poincaré section taken at apex for a running gait.

of the apex point requires $\dot{y}_a = 0$. Note that the dimension of Σ^r can further be reduced to one, if the total energy is conserved. Though, we keep it in the general form (two-dimensional) to avoid confusion in non-energy conservative cases such as active gaits.

Accordingly, the apex return map (ARM) $\mathcal{P}^r : \Sigma^r \to \Sigma^r$ gives a mapping between two subsequent apex states in Σ^r:

$$z_{a+} = \mathcal{P}^r(z_{a-}). \tag{2.5}$$

Following the definition of the phases in Figure 2.1b, the ARM can be constructed by composing four subsequent maps: the apex to touchdown (${}^{t}_{a-}\mathcal{P}^r$), the touchdown to bottom (${}^{b}_{t}\mathcal{P}^r$), the bottom to liftoff (${}^{l}_{b}\mathcal{P}^r$), and the liftoff to apex (${}^{a+}_{l}\mathcal{P}^r$). The ARM (2.5) therefore takes the following form:

$$z_{a+} = ({}^{a+}_{l}\mathcal{P}^r \circ {}^{l}_{b}\mathcal{P}^r \circ {}^{b}_{t}\mathcal{P}^r \circ {}^{t}_{a-}\mathcal{P}^r)(z_{a-}). \tag{2.6}$$

For the walking gait, we take the Poincaré section at VLO [51]. Following a similar process as for running, the transversal $\Sigma^w \subset \mathbb{R}^3$ to the orbit Γ^w is defined, and the VLO return map (VRM) $\mathcal{P}^w : \Sigma^w \to \Sigma^w$ defines a mapping between two subsequent VLO states in Σ^w:

$$z_{v+} = \mathcal{P}^w(z_{v-}), \tag{2.7}$$

where the VLO state in Σ^w takes the form

$$z_v = \begin{bmatrix} \dot{x}_v & y_v & \dot{y}_v \end{bmatrix}^T, \; z_v \in \Sigma^w. \tag{2.8}$$

Here, the definition of the VLO state already implies the horizontal position x_v. The VRM is the composition of four maps: the VLO to touchdown (${}^{t}_{v-}\mathcal{P}^w$), the touchdown to virtual bottom (${}^{\tilde{b}}_{t}\mathcal{P}^w$), the virtual bottom to liftoff (${}^{l}_{\tilde{b}}\mathcal{P}^w$), and the

liftoff to VLO (${}^{\mathrm{v}^+}_{\mathrm{l}}\mathcal{P}^{\mathrm{w}}$), as illustrated in Figure 2.1a:

$$z_{\mathrm{v}^+} = ({}^{\mathrm{v}^+}_{\mathrm{l}}\mathcal{P}^{\mathrm{w}} \circ {}^{\mathrm{l}}_{\tilde{\mathrm{b}}}\mathcal{P}^{\mathrm{w}} \circ {}^{\tilde{\mathrm{b}}}_{\mathrm{t}}\mathcal{P}^{\mathrm{w}} \circ {}^{\mathrm{t}}_{\mathrm{v}^-}\mathcal{P}^{\mathrm{w}})(z_{\mathrm{v}^-}). \tag{2.9}$$

The ARM and VRM, formulated in (2.6) and (2.9), are effective representations for the SLIP dynamics in running and walking that can be utilized for the stability and control purposes.

2.3 Hybrid automaton representation for the SLIP model

As described in the previous section, each walking and running cycle comprises a sequence of different phases with different dynamics, and the switching between the phases is governed by discrete events (transition guards). This means that the SLIP model is hybrid. A number of representations have been proposed for the modeling and analysis of hybrid systems, among which is the hybrid automaton framework [52]. Figure 2.3 depicts the hybrid automaton that represents the controlled SLIP system in walking, running, and walk-run transitions. We only describe here the notation and the definition of symbols, while the internal structure of the automaton is later treated in detail in Chapter 4. Also note that the notation introduced here is best useful when both running and walking gaits and their transitions are to be analyzed. As such, we use relaxed notations in chapters in which only a single gait is studied.

Every single node, depicted by a circle, represents the system in a particular phase, during which the system parameters, including the legs stiffness, are kept constant. The notation used to represent a particular node is

$$\mathrm{phase}^{\mathrm{gait}}_{\mathrm{subphase}}.$$

For the transition guards we use the following notation:

$$\mathrm{G}^{\mathrm{gait}}_{\mathrm{event}}.$$

Associated with each stance mode, the legs' stiffness are represented as

$$k^{\mathrm{gait}}_{\mathrm{subphase}}.$$

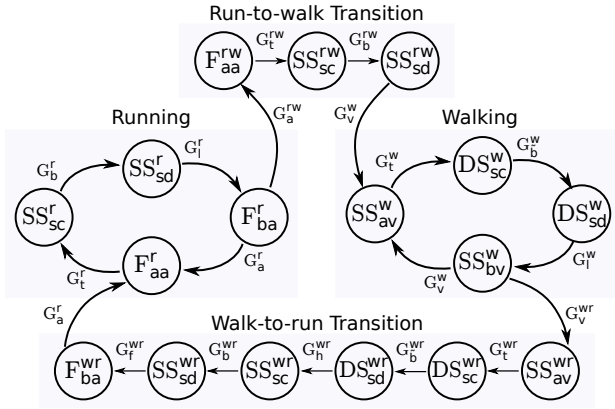


Figure 2.3: Hybrid automaton representing the pattern of walking, running and walk-run transitions in the SLIP model. See Table 2.1 for the notation.

Table 2.1: Notation used to define the SLIP hybrid automaton

gait		**phase**	
r	running	F	flight
w	walking	SS	single-stance
rw	run-to-walk transition	DS	double-stance
wr	walk-to-run transition	**event**	
		a	apex
subphase		b	bottom
sc	spring compression	v	VLO
sd	spring decompression	$\tilde{\text{b}}$	virtual bottom
bv	before VLO	t	touchdown
av	after VLO	l	liftoff
ba	before apex	f	fore leg liftoff
aa	after apex	h	hind leg liftoff

Finally, the system states at any particular event take the following form:

$$s_{\text{event}}.$$

Table 2.1 details all the notation that can be used in the possible combinations of these symbols. For instance, $\text{SS}^{\text{w}}_{\text{bv}}$ denotes the SS phase before VLO in the walking gait, and $\text{G}^{\text{r}}_{\text{l}}$ denotes the transition guard due to the liftoff event in the running gait.

Following the definition of the hybrid automaton, a number of reset maps are defined as follows. Whenever the system transitions to SS or DS mode, the leg(s) stiffness is set to the value commanded by a controller; At the touchdown instant

the toe position is reset such that the fully stretched leg makes an angle with the horizontal corresponding to the touchdown angle. The control system also updates the touchdown angle in each cycle once needed.

2.4 Analytical approaches to the SLIP dynamics

Towards the design of controllers that are computationally efficient for online implementation, in this section we seek analytical time solutions to the SLIP dynamics allowing the construction of the ARM and VRM. As will be shown for the stance phases, this can only be constituted in approximate forms.

2.4.1 Flight dynamics

Integrating the flight dynamics expressed by (2.1) results in the following well-known expressions for the movement of the CoM:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ -gt + \dot{y}_0 \end{bmatrix}, \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \dot{x}_0 t + x_0 \\ -gt^2/2 + \dot{y}_0 t + y_0 \end{bmatrix}, \tag{2.10}$$

where $(.)_0$ denotes the initial value at time $t = 0$. The submaps ${}^{\mathrm{t}}_{\mathrm{a}-}\mathcal{P}^{\mathrm{r}}$ and ${}^{\mathrm{a}+}_{\mathrm{l}}\mathcal{P}^{\mathrm{r}}$ can be determined using (2.10).

2.4.2 Single-stance dynamics

The simplicity of the physical structure of the SLIP model seemingly suggests that deriving a closed-form solution to its dynamics in the stance phases is simple too. However, as discussed earlier, such an analytical solution has remained an open problem to date. It has been shown that the mentioned dynamics under the influence of gravity are non-integrable [42, 53]. Lacking a closed-form solution for the SLIP dynamics in stance has limited the use of functional analysis tools, which could be instrumental in the design of dexterous and dynamic legged robots. If an arbitrarily close approximation to the exact solution of the SLIP dynamics is to be predicted, then the common approach is through relatively extensive forward-in-time numerical integrations (e.g., [54–56]), where the tradeoff between precision and computational complexity plays an important role. As such, precise numerical approximations might not be suitable for online implementation. Moreover, the numerical nature of such solutions precludes the possibility of deriving a closed-form expression for the associated Jacobian matrix.

Nevertheless, several alternatives aiming at finding an accurate analytical approximation to the SLIP dynamics in the SS phase have been proposed [43–47],

which rely on simplified and linearized dynamic models. The approximation presented in [44] ignores the effect of gravity in stance, something that can hardly be fulfilled in reality.

Geyer et al. [45] proposed a simple solution, in terms of elementary functions, that approximates the ARM without the need for an iterative process. The simplicity of the solution is such that it can further be used to investigate the stability of ARMs in some special situations. Although the method takes into account a linearized effect of gravity in the force balance equations, it presumes the conservation of angular momentum, an assumption that can significantly be violated in non-symmetric gaits[4] due to gravity.

Using an iterative algorithm based on the mean-value theorem, Schwind and Koditschek [43] proposed another noteworthy approximation. The main characteristic of their method is its iterative form, in the sense that at least two iterations are required in order to meet a minimal accuracy threshold. Although, the quality of the results increasingly improves with each iteration, the mathematical complexity of the resulting solution negatively affects the utility of the method for further usage such as stability analysis.

Inspired by this work, Arslan et al. [57] expanded the Geyer et al. method by adding a gravity correction-based iteration, forming a two-step iterative solution. The method has been further expanded in [48] to handle energy-dissipative elements and the results have been experimentally validated in [58] recently. Subsequent controller designs [48, 59, 60] carried out using this approximation have shown promising results.

Recently, based on the perturbation theory, another solution has been proposed in [47]. The method assumes the angular momentum as a conserved quantity in the radial motion as in the Geyer et al. method. However, for the angular motion the effect of gravitational torque is included leading to a varying angular momentum. The resulting equations are then solved using standard perturbation techniques. Based on the error analysis reported, the prediction performance is comparable to the Arslan et al. extension.

Among the methods addressed above, the Geyer et al. method with the Arslan et al. extension has received much attention, and also inspired some of the findings we present in the next chapter. As such, we briefly present them in the rest of this section.

[4] In a symmetric gait, the trajectory is symmetric with respect to the vertical leg orientation.

Approximate single-stance map by Geyer et al.

In [45] Geyer et al. derived a simple solution to the stance phase of the SLIP hopper. The approximation relies on two assumptions: (i) the angle swept during the stance ($\Delta\theta$) is sufficiently small, and the stance phase is predominantly vertical (i.e., $\cos\theta \approx 1$); (ii) the compression of the leg is much smaller than its rest length. Combined with further simplifications detailed in [45], the following expressions for the radial and angular motions of the CoM in polar coordinates (r, θ) are derived[5]:

$$\begin{aligned} r(t) =& \frac{f}{\hat{\omega}_0^2} + \lambda_1 \sin\hat{\omega}_0 t + \lambda_2 \cos\hat{\omega}_0 t, \\ \theta(t) =& \frac{2g/l_0 + \omega_0^2 + \omega^2}{\hat{\omega}_0^2}\omega t \\ &+ \frac{2\omega}{l_0\hat{\omega}_0}(\lambda_1 \cos\hat{\omega}_0 t - \lambda_2 \sin\hat{\omega}_0 t) + \lambda_3, \end{aligned} \tag{2.11}$$

where $\hat{\omega}_0^2 = \omega_0^2 + 3\omega^2$, $f = -g + l_0\omega_0^2 + 4l_0\omega^2$, $\omega = p/(m{l_0}^2)$ and $\omega_0^2 = k/m$, with $p = mr^2\dot{\theta}$ being the angular momentum of mass m around the toe conserved during motion, which can be substituted by the known angular momentum at the initial condition, $p_0 = mr_0^2\dot{\theta}_0$. Finally, λ_1, λ_2 and λ_3 are constant values determined by the initial conditions. This approximate solution is valid for any pair of initial and final conditions within the SS phase including the touchdown and liftoff points.

Gravity correction scheme by Arslan et al.

The above-presented method fails shortly when the stance trajectory is non-symmetric. This is because in this case the average angular momentum due to gravity is nonzero, thereby the total angular momentum can no longer be assumed conserved. Motivated by this, Arslan et al. [57] have incorporated an average effect of gravity on the angular momentum, which can be approximately modeled as

$$\bar{p}_\mathrm{g} = \frac{t_\mathrm{e}}{2} mg\bar{r}(\sin\theta_0 + \sin\theta_\mathrm{e}), \tag{2.12}$$

where subscript "e" stands for the end state, and $\bar{r}$ is the approximated average leg length that can be obtained using (2.11):

$$\begin{aligned} \bar{r} =& \frac{1}{t_\mathrm{e}}\int_0^{t_\mathrm{e}} r(t)\,\mathrm{d}t, \\ =& \frac{f}{\hat{\omega}_0^2} + \frac{1}{\hat{\omega}_0 t_\mathrm{e}}(\lambda_1 - \lambda_1\cos\hat{\omega}_0 t_\mathrm{e} + \lambda_2 \sin\hat{\omega}_0 t_\mathrm{e}). \end{aligned} \tag{2.13}$$

[5] The instant of the initial condition is defined as $t = 0$.

Now, the gravity correction term $\bar{p}_{\mathrm{g}}$ is added to the original angular momentum p_0 to yield the updated angular momentum

$$\hat{p} = p_0 + \bar{p}_{\mathrm{g}}. \tag{2.14}$$

Finally, $\hat{p}$ replaces p_0 in all corresponding derivations, which gives a two-iteration form to the solution.

The presented method effectively establishes an analytical solution to the SS dynamics of the SLIP from any initial to any final state. The maps ${}^{\mathrm{b}}_{\mathrm{t}}\mathcal{P}^{\mathrm{r}}$, ${}^{\mathrm{l}}_{\mathrm{b}}\mathcal{P}^{\mathrm{r}}$, ${}^{\mathrm{t}}_{\mathrm{v}^-}\mathcal{P}^{\mathrm{w}}$ and ${}^{\mathrm{v}^+}_{\mathrm{l}}\mathcal{P}^{\mathrm{w}}$ can be constructed accordingly.

2.4.3 Double-stance dynamics

To the best of the author's knowledge, an analytical solution to the DS dynamics of SLIP, even in an approximate form, was remained unexplored. In Chapter 3, we introduce the first approximate solutions to this problem. Therefore, we leave the complete treatment of this topic to that chapter.

2.5 Modeling frameworks for legged locomotion

In this section we review existing modeling frameworks for legged locomotion, with an emphasis put on the recently developed method in the max-plus algebra. We utilize the max-plus linear systems later in Chapter 5, where the coordination of SLIP models towards quadrupedal running is addressed. Most of the material discussed in the present section are taken from [61].

2.5.1 Central pattern generators

In neuroscience, the neural networks that generate animals' limb coordination patterns are referred to as the central pattern generators (CPGs) [12, 62–64]. The focus in the above-mentioned studies (and the references therein) is put on understanding the control mechanism of limb coordination by analyzing the interaction between populations of neurons. Currently, CPGs are standard tools also in robotics for generation of references whose realization leads to coordination of legs.

A common mathematical implementation for CPGs is accomplished by abstracting the periodic motion of a leg in phase $\theta_i \in \mathbb{S}^1$, with $\mathbb{S}^1$ representing the circle. The coordination between the legs can then be induced using the following

Table 2.2: Buehler clock design parameters

Symbol	Definition
ϕ_s	stance phase (in which the legs are assumed to be in stance)
τ_c	cycle time
τ_s	stance time
τ_d	double-stance time

nonlinear coupled differential equations:

$$\dot{\theta}(\tau) = V + h(\theta(\tau)), \tag{2.15}$$

where, $\theta = \begin{bmatrix}\theta_1 & \dots & \theta_n\end{bmatrix}^T \in \mathbb{T}^n$ is the full phase vector with $\mathbb{T}^n$ being the n-torus, τ represents time, $V \in \mathbb{R}^n$ denotes the desired phase velocity vector, and the desired coupling behavior is included in h. Weighted sums of sinusoidal functions are common realization for h. The abstract phase $\theta(\tau)$ can then be mapped into reference trajectories for the legs of the robot via a parameterized map g:

$$q_{\text{ref}}(\tau) = g(p, \theta(\tau)), \tag{2.16}$$

with p being a set of parameters that shape resulting phase curves in space.

Current gait reference generation frameworks mostly implement the CPGs-based representations to induce synchronization. Despite the straightforward implementation of CPGs, there are some disadvantages to this approach, mainly due to the nature of (2.15) as a set of nonlinear coupled differential equations that need to be solved in real-time. Moreover, typical of differential questions, the evolution of the phase variables in the transient phases is less understood. This is more so when the parameters of (2.15) are varying with respect to time. Examples of such cases include varying speed gaits and gait transitions, which are of particular importance in this thesis.

2.5.2 Buehler clock

The "Buehler clock" [33], illustrated in Figure 2.4, is another approach for synchronization of cyclic systems. As can be seen, piecewise constant velocity references represent the relation between the phase and time. The real-time computations needed are, hence, very simple, as opposed to solving differential equations in the case of CPGs. The design parameters of the Buehler clock are defined in Table 2.2. Interested readers are referred to [33, 61] for the mathematical formulations and more details.

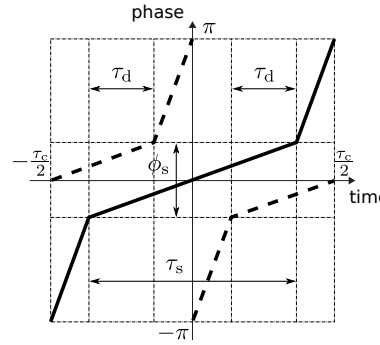


Figure 2.4: "Buehler clock" model for a tripod gait of a hexapod robot (reproduced from [33]). The solid and dashed lines represent the trajectories corresponding to the reference phase of a group of legs in time.

2.5.3 Timed event graphs

By considering only two physical states for the legs of a robot, stance and swing, distinguished from each other by the touchdown and liftoff events, Lopes et al. [61] proposed a different approach to model legged locomotion. The model finds its origin in the notion of Petri nets [65]. When additionally considering that there exists a time structure associated to the Petri net, then according to the following definition, it is convenient to use the notion of *timed event graphs*.

Definition 2.1 ([66])**.** *A timed Petri net $\mathcal{G}$ is characterized by a set of places $\mathcal{P}$, a set of transitions $\mathcal{G}$, a set of arcs $\mathcal{D}$ from transitions to places and vice versa, an initial marking $\mathcal{M}_0$, and a holding time vector $\mathcal{T}$. If each place has exactly one upstream and one downstream transition, then the timed Petri net is called a* timed event graph.

As in [61], in what follows we illustrate the method using a two-legged robot example. For each leg a circuit composed of two places (f_i for swing and g_i for stance) and two respective transitions ($t_{t,i}$ for touchdown and $t_{l,i}$ for liftoff) is considered (see Figure 2.5a1). Since it is assumed that both legs are in stance at the start time, each circuit is initialized with a token in the stance places. A minimum time (holding time, see [67], Definition 2.43) is also considered for each place. Specifically, each leg must stay at least τ_{sw} time units in swing and τ_{st} time units in stance. Note that up to this point no individual mechanism for the synchronization of the legs is considered.

Figure 2.5a2 shows a sample simulation of the described system, in which the events are fired randomly according to a bounded uniform distribution. The gray rectangles represent stance and white space represents flight. As can be seen, the evolution of event timings of the legs are independent, hence no synchronized

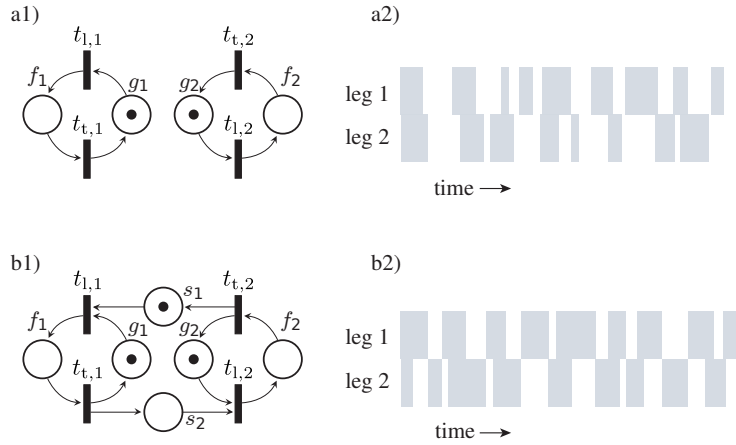


Figure 2.5: An illustrative example for the timed event graphs (adopted from [61]). The top figure depicts the modeling of two unsynchronized legs, in which the moment of touchdown and liftoff events for each leg is independent. The bottom figure represents the same modeling simulation, but in the presence of synchronization. As can be seen, a liftoff event can only occur when the other leg is in stance, which follows Definition 2.2.

behavior is observed. Lopes et al. defined their notion of the synchronization as follows.

Definition 2.2 ([61])**.** *We say that the legs of a robot are synchronized if each leg's liftoff event is a function of the touchdown events of other legs.*

It is important to realize that this liftoff constraint is not unique or strictly needed to synchronize legs, as the opposite condition would also be valid. We now repeat the previous simulation, but this time in the presence of the synchronization mechanism that follows Definition 2.2. As shown in Figure 2.5b1, here an additional token needs to be considered in either of s_1 or s_2 such that the whole net is alive. The resulting synchronized behavior is depicted in Figure 2.5b2.

Having determined the evolution of reference event timings, one can generate continuous-time reference phase trajectories for the legs using a map, a sample derivation of which is detailed in [61], Section V.

If one considers the timed event graph example in Figure 2.5b1 with its events firing as soon as they are enabled, the associated equations describing the evolution of event timings can be derived using the operator max. To do so, associate the holding time τ_{st} to the stance places g_i, the holding time τ_{sw} to the swing places f_i, and the double-stance time τ_{Δ} to the synchronization places s_i. Moreover, for leg i denote the touchdown and liftoff time instants in the κth cycle by $t_{\mathrm{t},i}(\kappa)$ and $t_{\mathrm{l},i}(\kappa)$, respectively. Then, the evolution equations can easily be derived as

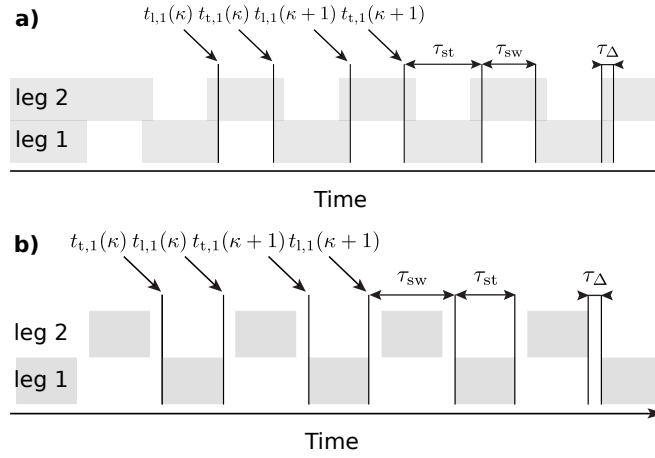


Figure 2.6: Event schedule for a walking gait (a) and a running gait (b) of a two-legged system. Gray rectangles represent stance and white space represents swing.

$$t_{t,1}(\kappa) = t_{l,1}(\kappa) + \tau_{sw}, \tag{2.17}$$

$$t_{t,2}(\kappa) = t_{l,2}(\kappa) + \tau_{sw}, \tag{2.18}$$

$$t_{l,1}(\kappa) = \max(t_{t,1}(\kappa-1) + \tau_{st}, t_{t,2}(\kappa-1) + \tau_\Delta), \tag{2.19}$$

$$t_{l,2}(\kappa) = \max(t_{t,2}(\kappa-1) + \tau_{st}, t_{t,1}(\kappa) + \tau_\Delta). \tag{2.20}$$

Equation (2.17) states that leg 1 touches down τ_{sw} time units after it has lifted off. Equation (2.19) states that leg 1 will lift off the ground after τ_{st} time units of it being in stance and τ_Δ time units of it being in double-stance (i.e., leg 2 has touched down). Equations (2.18) and (2.20) are interpreted analogously. The resulting event schedule are depicted in Figure 2.6a.

Notice that to induce synchronization, the liftoff event of leg 2 in the first cycle $t_{l,2}(1)$ is elongated, which complies with Definition 2.2. In general, every event schedule produced by equations (2.17)–(2.20) ensures constant swing durations and adjusts stance durations once needed. This lets the double-stance phase to emerge automatically, which is a characteristic for walking gaits. By swapping the roles of touchdown and liftoff events in equations (2.17)–(2.20), a running gait will emerge, in which the double-swing phase is present, as shown in Figure 2.6b. Every event schedule produced accordingly ensures constant stance durations and adjusts the swing durations for synchronization.

Since equations (2.17)–(2.20) contains only the max and + operators, Lopes et al. explored the advantages of describing these relations in the max-plus algebra,

whose formal definition is given in the next section. The significance of this approach is in translating the nonlinear equations (2.17)–(2.20) to a set of state-space form in the max-plus sense. The reset of this section is devoted to a detailed presentation of the max-plus linear systems for scheduling, the methodology we utilize in the design of our coordination controller in Chapter 5.

2.5.4 Max-plus based gait scheduler

Max-plus algebra

The max-plus algebra was introduced in the sixties independently by Giffler [68] and Cuninghame-Green [69]. The structure of the max-plus algebra [67] is as follows. Let $\varepsilon := -\infty, e := 0$, and $\mathbb{R}_{\max} = \mathbb{R} \cup \{\varepsilon\}$. Define the operations $\oplus, \otimes : \mathbb{R}_{\max} \times \mathbb{R}_{\max} \to \mathbb{R}_{\max}$ by

$$x \oplus y := \max(x, y)$$
$$x \otimes y := x + y.$$

The set $\mathbb{R}_{\max}$ with the operations $\oplus$ and $\otimes$ is called the max-plus algebra, denoted by $\mathcal{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \varepsilon, e)$.

Please consult [67, 70] for a complete treatment of the max-plus algebra, and [61] for a presentation of most relevant elements of the max-plus algebra in the context of legged locomotion. Particularly, see [61], Theorem 10, which introduces the notion of coupling time whose existence implies a finite number of steps after which a modeled gait pattern surely reaches the steady-state behavior. This is similar to having stable limit cycles in CPGs. Moreover, the following theorem can be used to translate the equations of event timings evolution in the max-plus algebra from an implicit representation to an explicit form that is easy to solve.

Theorem 2.1 (see [67], Th 3.17)**.** *Consider the following system of linear equations in the max-plus algebra:*

$$x = A \otimes x \oplus b \tag{2.21}$$

with $A \in \mathbb{R}_{\max}^{n \times n}$ *and* $b, x \in \mathbb{R}_{\max}^{n \times 1}$*. Now let*

$$A^* := \bigoplus_{p=0}^{\infty} A^{\otimes p}.$$

If A^* *exists then* $x = A^* \otimes b$ *solves the system of max-plus linear equations* (2.21)*.*

Max-plus linear systems

As shown in [61], the set of nonlinear equations (2.17)–(2.20) can be translated to the following linear state-space form in the max-plus algebra:

$$\begin{bmatrix} t_{\mathrm{t},1}(\kappa) \\ t_{\mathrm{t},2}(\kappa) \\ t_{\mathrm{l},1}(\kappa) \\ t_{\mathrm{l},2}(\kappa) \end{bmatrix} = \left[\begin{array}{cc|cc} \varepsilon & \varepsilon & \tau_{\mathrm{sw}} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \tau_{\mathrm{sw}} \\ \hline \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_{\Delta} & \varepsilon & \varepsilon & \varepsilon \end{array}\right] \otimes \begin{bmatrix} t_{\mathrm{t},1}(\kappa) \\ t_{\mathrm{t},2}(\kappa) \\ t_{\mathrm{l},1}(\kappa) \\ t_{\mathrm{l},2}(\kappa) \end{bmatrix} \oplus \left[\begin{array}{cc|cc} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \hline \tau_{\mathrm{st}} & \tau_{\Delta} & \varepsilon & \varepsilon \\ \varepsilon & \tau_{\mathrm{st}} & \varepsilon & \varepsilon \end{array}\right] \otimes \begin{bmatrix} t_{\mathrm{t},1}(\kappa-1) \\ t_{\mathrm{t},2}(\kappa-1) \\ t_{\mathrm{l},1}(\kappa-1) \\ t_{\mathrm{l},2}(\kappa-1) \end{bmatrix}. \tag{2.22}$$

To generalize the exemplified system equations (2.22), define the *temporal state scheduling vector* (hereon, scheduling vector or schedule) for $n \in \mathbb{N}\backslash\{0,1\}$ legs as

$$S(\kappa) = [\underbrace{t_{\mathrm{t},1}(\kappa) \dots t_{\mathrm{t},n}(\kappa)}_{t_{\mathrm{t}}(\kappa)} \; \underbrace{t_{\mathrm{l},1}(\kappa) \dots t_{\mathrm{l},n}(\kappa)}_{t_{\mathrm{l}}(\kappa)}]^T. \tag{2.23}$$

The evolution equations for n legs can then be written as

$$S(\kappa) = A_0 \otimes S(\kappa) \oplus A_1 \otimes S(\kappa - 1), \tag{2.24}$$

with A_0 and $A_1 \in \mathbb{R}_{\max}^{2n \times 2n}$ being the linear system matrices. It is clear that both matrices are max-plus zero in the block diagonals, with all the parameters lying in the off-diagonal blocks. With further manipulations in the equations the following expression for these matrices can be derived [61]:

$$A_0 = \left[\begin{array}{c|c} \mathcal{E} & \tau_{\mathrm{sw}} \otimes E \\ \hline P & \mathcal{E} \end{array}\right] \text{ and } A_1 = \left[\begin{array}{c|c} E & \mathcal{E} \\ \hline \tau_{\mathrm{st}} \otimes E \oplus Q & E \end{array}\right], \tag{2.25}$$

where the max-plus zero $\mathcal{E} \in \mathbb{R}_{\max}^{n \times n}$, and (square) identity $E \in \mathbb{R}_{\max}^{n \times n}$ matrices are defined by

$$[\mathcal{E}]_{ij} = \varepsilon \text{ and } [E]_{ij} = \begin{cases} e, & \textbf{if } i = j \\ \varepsilon, & \textbf{otherwise} \end{cases}$$

To develop a systematic method for constructing the additional matrices P and $Q \in \mathbb{R}_{\max}^{n \times n}$, we parameterize the leg synchronization rhythm (i.e., the gait G) by

$$\mathrm{G} = h_1 \prec h_2 \prec \cdots \prec h_m, \tag{2.26}$$

where h_i, $i \in \{1, 2, \ldots, m\}$, is a set of integers corresponding to a group of legs that are simultaneously in swing (hence, $m \leq n$ is the number of mentioned groups). Any set h_i should have the following properties:

1. h_i is not empty, and takes elements of $\{1, 2, \ldots, n\}$ with no overlap between sets;

2. The union of all h_i equals $\{1, 2, \ldots, n\}$.

According to this ordering relation, each leg in the set h_{i+1} swings τ_Δ time units after all the legs in the set h_i have reached stance. For example, for the set of equations (2.22) one can represent the gait as $\mathrm{G} = \{1\} \prec \{2\}$. Given this notation, the matrices P and Q can be generated by

$$
\begin{aligned}
[P]_{pq} &= \begin{cases} \tau_\Delta, & \forall j \in \{1, \ldots, m-1\}; \forall p \in h_{j+1}; \forall q \in h_j \\ \varepsilon, & \textbf{otherwise} \end{cases} \\
[Q]_{pq} &= \begin{cases} \tau_\Delta, & \forall p \in h_1; \forall q \in h_m \\ \varepsilon, & \textbf{otherwise} \end{cases}
\end{aligned}
\tag{2.27}
$$

This completes the derivation of the max-plus linear system (2.24), which is of an implicit form. As mentioned earlier, an explicit alternative form can also be obtained. Let a new system matrix A be defined as

$$
A := A_0^* \otimes A_1, \tag{2.28}
$$

in which A_0^* can be generated using Theorem 2.1. It was shown in [71] that for the given synchronization specifications formulated in Definition 2.2 and equation (2.26) the matrix A_0^* and, hence, the matrix A exist. Consequently, the representation (2.24) can be rewritten in the following explicit form (here the cycle index κ is incremented by 1):

$$
S(\kappa + 1) = A \otimes S(\kappa). \tag{2.29}
$$

The system matrix A constitutes a number of mathematical properties in the max-plus sense that are instrumental in characterizing the resulting gait behavior. The max-plus eigenvalue and eigenvector of A are the cycle time and the steady state behavior, respectively. Also, the coupling time of A describes the transient behavior [67]. Please refer to [61], Section IV-C, for a detailed description of the properties.

For those gaits that have varying temporal parameters, such as the switching between different gaits, an extension to max-plus linear systems can be considered,

Table 2.3: Comparison between standard CPGs and switching max-plus methods [61]

Property	CPGs	Switching max-plus
Dynamics	continuous	discrete
System representation	differential equation (2.15)	max-plus linear system (2.30)
Control parameterization	set of phase offset parameters and gains	ordered set of numbers (gait) and temporal parameters τ_{sw}, τ_{st} and τ_{Δ}
Steady state	limit cycle	max-plus eigenvector
Cycle time	depends on the gain	max-plus eigenvalue
Convergence	depends on the gain	maximum 2 cycles
Transitions with constraint guarantees	obstacles encoded in vector fields	switch state matrices
Implementation	numerical differential equation solver	additions, maximizations, linear interpolation
Output smoothness	C^{∞}	C^{n} with n finite

namely switching max-plus linear systems. Let $\mu(\kappa)$ be a "switching" integer function whose value designates a certain gait. Slightly revising (2.24) and (2.29) to

$$\begin{aligned} S(\kappa) &= A_0(\mu(\kappa)) \otimes S(\kappa) \oplus A_1(\mu(\kappa)) \otimes S(\kappa-1), \\ S(\kappa+1) &= A(\mu(\kappa+1)) \otimes S(\kappa), \end{aligned} \tag{2.30}$$

enables different gait parameters during the evolution of schedule vector S. We will utilize the switching max-plus system equations (2.30) in Chapter 5 in automating the transitions between different running gaits of a quadruped.

Finally, Lopes et al. [61], proposed a map that transforms the produced event schedule into a continuous-time reference trajectory, which can directly be sent to the low-level tracking controllers of the robot's legs.

The switching max-plus method, which is founded based on the timed event graphs, is a generalization of the Buehler Clock. A comparison between the standard CPGs-based methods versus the switching max-plus methodology is established in Table 2.3.

2.6 Summary

In this chapter a framework for modeling and control of legged locomotion on the SLIP template was established. A hybrid automaton was introduced that formally describes the sequence of corresponding phases of motion for the SLIP running,

walking, and transitions between them. Next, the analytical approaches to the stance dynamics of SLIP were reviewed, among which the Geyer et al. method and the Arslan et al. extension were detailed. This background information motivates the DS map developed in Chapter 3. Finally, a review of modeling frameworks for locomotion in multi-legged robots was presented. The max-plus based gait scheduling methodology was discussed, and its relations with other available methods were explained.

Chapter

3

Analytical approximations for the double-stance dynamics of the lossy SLIP

During the review of the literature discussing the SLIP modeling in the previous chapter, it was revealed that an analytical approach to the SLIP double-stance dynamics was remained unexplored. With an aim to derive a computationally fast and analytically tractable solution, this chapter presents two novel approximations to those dynamics.

The energy dissipation, which is inevitably present in real applications, is also included in the considered model (Section 3.2). Two different approaches aiming at different specifications are followed, and the resulting time-domain solutions are addressed (Section 3.3). The role of the resulting double-stance solutions in the whole walking cycle is discussed (Section 3.4). Since the proposed solutions are obtained through approximations, the influence of approximation errors is assessed through an extensive numerical analysis (Section 3.5), and the chapter is concluded in Section 3.6.

3.1 Introduction

In previous chapters, we explained why an analytical approach to the dynamics of SLIP is an important requirement for the success of the control framework developed in this thesis. In short, such a solution can be utilized for online planning, an important feature enhancing the level of autonomy in legged robotics. Moreover, it can be used to obtain the associated Jacobian matrix analytically, which provides useful information for the subsequent numerical calculations such as optimization. It was also discussed that deriving the exact analytical solutions for the stance dynamics of SLIP is an open problem to date. Alternative approximate solutions for the SS phase were reviewed and the need for a similar solution to the DS dynamics was highlighted.

This chapter fills the gap by exploring two different approaches to the aforementioned problem. We derive a complete approximate solution to the DS dynamics of the lossy SLIP model, in which the effect of non-negligible damping is explicitly accounted for. At the core of the solution is the idea of approximating the SLIP trajectories in the DS phase (depicted in Figure 3.1a) by those of an auxiliary system, which we refer to as the lossy AT-SLIP (see Figure 3.1b). We determine conditions and parameter combinations under which the two systems are approximately equivalent. The significance of the AT-SLIP model is in its relatively simple structure that facilitates the process of approximation. When the two systems are approximately equivalent, any solution we derive to the AT-SLIP dynamics also qualifies as a solution to the SLIP dynamics in DS. In deriving the approximate solution to the lossy AT-SLIP dynamics, we are primarily motivated by the approach proposed in [45] and extended in [48], namely the Updated-momentum (UM) approach. However, substantial extensions have been made due to additional complexities introduced by the torsional spring and damper elements.

Although the proposed solution following the UM method is computationally much more efficient than numerical integration,[1] it embodies a two-step iterative form. This might limit its effectiveness in special cases where parametric analysis is needed. To overcome this potential limitation we take a different approach, namely the Perturbation-based (PB) approach, wherein we compute the approximate solutions to the *lossless* DS in a single step. Although the new approximations seem more complicated, they are suitable for functional analysis as there is no need for an extra iteration in the calculations. Nevertheless, both the previous two-step and the new single-step approximation methods yield sufficiently accurate predictions of system trajectories.

[1] An approximate analytic solution of the SLIP dynamics is evaluated at least 250 times faster than the corresponding numerical solution, as discussed in [72].

For both solutions, a careful characterization of the approximation errors is presented. The predictive power is assessed via an extensive numerical analysis, which simulates highly non-symmetric trajectories in the presence of non-negligible damping, with initial conditions that cover reasonable domains of the SLIP motion. The comparison between the accuracies of the two solutions provides useful information based on which the validity of the presented approaches is justified.

3.2 The lossy SLIP model and its derivatives

3.2.1 The lossy SLIP model

As pointed out in the Introduction, in the present chapter we aim to study a lossy version of the SLIP. Such a system in DS configuration is depicted in Figure 3.1a. Except for the addition of a viscus damping b_0, the system constitutes the same physical and mathematical specifications as the lossless bipedal SLIP defined in Section 2.2[2]. The CoM equations of motion in DS previously derived in (2.3) are extended to the following due to the presence of damping:

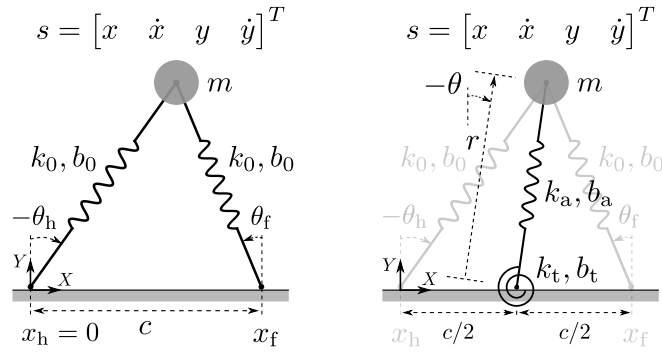
$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = T_{\mathrm{DS}}(s,q) \begin{bmatrix} k_0\left(l_{\mathrm{rest}} - l_{\mathrm{f}}(s,q)\right) - b_0\dot{l}_{\mathrm{f}}(s,q) \\ k_0\left(l_{\mathrm{rest}} - l_{\mathrm{h}}(s,q)\right) - b_0\dot{l}_{\mathrm{h}}(s,q) \end{bmatrix} + \begin{bmatrix} 0 \\ -mg \end{bmatrix}. \tag{3.1}$$

3.2.2 The lossy AT-SLIP model

As depicted in Figure 3.1b, the AT-SLIP consists of the same mass m on top of a *single* leg with an axial stiffness k_{a} and damping b_{a}. The toe is affixed to the ground at the midpoint of the fore and hind toes. A torsional spring and damper is added to the toe, with spring constant k_{t} and viscous damping b_{t}. Denote by r_{rest} and θ_{rest} the rest length of the axial spring and the zero torque angle of the torsional spring, respectively.

The AT-SLIP CoM equations of motion in Cartesian coordinates (x, y) are as

[2] Here $\mathcal{M} \in \{\mathrm{SS}, \mathrm{DS}\}$, since throughout this chapter we focus only on the walking gait.



(a) SLIP (b) AT-SLIP

Figure 3.1: (a) The "bipedal" SLIP model depicted in the double-stance (DS) phase, and (b) the Axial-torsional SLIP (AT-SLIP) model.

follows:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = T_{\mathrm{AT}}(s,q) \begin{bmatrix} k_{\mathrm{a}}\left(r_{\mathrm{rest}} - r(s,q)\right) - b_{\mathrm{a}}\dot{r}(s,q) \\ \dfrac{k_{\mathrm{t}}\left(\theta_{\mathrm{rest}} - \theta(s,q)\right)}{r(s,q)} - \dfrac{b_{\mathrm{t}}\dot{\theta}(s,q)}{r(s,q)} \end{bmatrix} + \begin{bmatrix} 0 \\ -mg \end{bmatrix}, \tag{3.2}$$

where (r, θ) is a polar coordinates system, as defined in Figure 3.1b, and

$$T_{\mathrm{AT}}(s,q) := \begin{bmatrix} -\sin\theta(s,q) & \cos\theta(s,q) \\ \cos\theta(s,q) & \sin\theta(s,q) \end{bmatrix}.$$

3.2.3 Equivalence of models

The aim of this section is to find a set of parameters $(\theta_{\mathrm{rest}}, r_{\mathrm{rest}}, k_{\mathrm{a}}, k_{\mathrm{t}}, b_{\mathrm{a}}, b_{\mathrm{t}})$, as functions of the SLIP parameters, with which the AT-SLIP best approximates the SLIP in DS. To do so, we first assume that the motion of the CoM is such that it imposes the following conditions to the AT-SLIP states:

$$\frac{r_{\mathrm{rest}} - r}{r_{\mathrm{rest}}} \ll 1, \tag{3.3}$$

$$\cos\theta \approx 1. \tag{3.4}$$

Assumption (3.3) requires that the maximum compression of the axial spring is much smaller than its rest length. Assumption (3.4) implies that the angle spanned

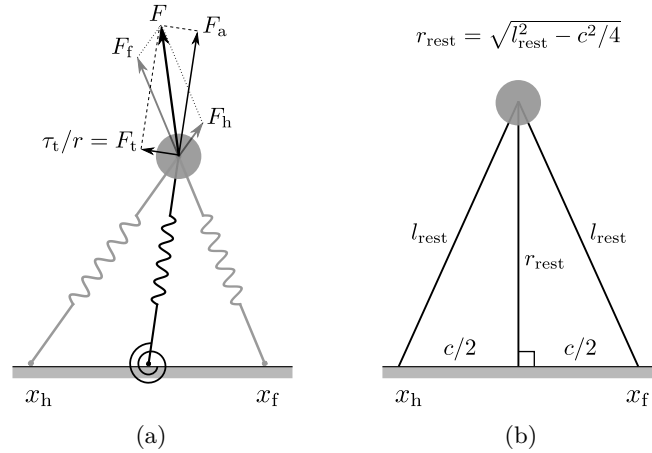


Figure 3.2: An illustration of the equivalency conditions between the SLIP and AT-SLIP models: (a) the springs force balance in an arbitrary configuration of the systems that is in accordance with assumptions (3.3) and (3.4), and (b) the zero spring force configuration of the two systems.

by the torsional spring is small with the axial spring predominantly vertical. These assumptions largely remain valid for the normal range of motions of the SLIP system, and have been commonly made in the literature (see, for example, [45]).

When equivalent, the two systems exert the same resultant forces on the mass m for any arbitrary state (see Figure 3.2a). Moreover, the spring constants leading to the equivalency in the presence and absence of damping should be the same. This follows from the fact that the spring force is conservative whereas the damping force is dissipative. Taking this into account, in the rest of this section we propose a step by step procedure for relating the parameters of AT-SLIP to those of the SLIP.

According to Figure 3.2b, the zero spring force position of the SLIP is a symmetric configuration. For the chosen toe position illustrated in Figure 3.1b, this requires that

$$\theta_{rest} = 0. \tag{3.5}$$

Moreover, for the given configuration one can derive:

$$r_{rest} = \sqrt{l_{rest}^2 - c^2/4}. \tag{3.6}$$

Next, we equate (3.1) to (3.2) while setting all damping coefficients to zero

and solve the resulting relation for k_a and k_t to yield:

$$\begin{bmatrix} k_\mathrm{a}(s,q) \\ k_\mathrm{t}(s,q) \end{bmatrix} = A^{-1}(s,q) T_\mathrm{DS}(s,q) \begin{bmatrix} k_0 \left(l_\mathrm{rest} - l_\mathrm{f}(s,q)\right) \\ k_0 \left(l_\mathrm{rest} - l_\mathrm{h}(s,q)\right) \end{bmatrix}, \tag{3.7}$$

with:

$$A(s,q) = T_\mathrm{AT}(s,q) \begin{bmatrix} r_\mathrm{rest} - r(s,q) & 0 \\ 0 & \dfrac{-\theta(s,q)}{r(s,q)} \end{bmatrix}.$$

It is clear that the resulting stiffnesses are functions of (x, y) and, hence, vary across the range of motion. However, numerical parameter estimation reveals that for the range of motion for which assumptions (3.3) and (3.4) remain largely valid, they can be approximated by specific limit values. To find a closed-form solution for the above observation, we compute the limit of $k_\mathrm{a}(s,q)$ and $k_\mathrm{t}(s,q)$ when x and y approach $c/2$ and r_rest, respectively:

$$k_\mathrm{a} = \lim_{(x,y)\to(c/2,r_\mathrm{rest})} k_\mathrm{a}(s,q) = 2\left(r_\mathrm{rest}/l_\mathrm{rest}\right)^2 k_0, \tag{3.8}$$

$$k_\mathrm{t} = \lim_{(x,y)\to(c/2,r_\mathrm{rest})} k_\mathrm{t}(s,q) = c^2/2\left(r_\mathrm{rest}/l_\mathrm{rest}\right)^2 k_0. \tag{3.9}$$

The resulting simple expressions also make intuitive sense. For instance, if $c \to 0$ then $r_\mathrm{rest} \to l_\mathrm{rest}$, thereby $k_\mathrm{a} \to 2k_0$ and $k_\mathrm{t} \to 0$.

Now that we have derived the expressions for k_a and k_t, we can equate (3.1) to (3.2), but this time in the presence of damping forces. By following a similar procedure as for the spring constants and in the sense of approximation, we eventually obtain:

$$b_\mathrm{a} = 2\left(r_\mathrm{rest}/l_\mathrm{rest}\right)^2 b_0, \tag{3.10}$$

$$b_\mathrm{t} = c^2/2\left(r_\mathrm{rest}/l_\mathrm{rest}\right)^2 b_0. \tag{3.11}$$

In summary, for the set $(\theta_\mathrm{rest}, r_\mathrm{rest}, k_\mathrm{a}, k_\mathrm{t}, b_\mathrm{a}, b_\mathrm{t})$ we derived expressions (3.5), (3.6), (3.8), (3.9), (3.10) and (3.11), with which the AT-SLIP approximates the dynamics of the SLIP in DS, provided that assumptions (3.3) and (3.4) hold.

Note that to find the equivalency relations in the case of lossless SLIP, we have also presented a different approach in [24]. Although the obtained expressions for $k_\mathrm{a}, k_\mathrm{t}$ and r_rest are different from those derived here, they result in very similar numerical values. The reader is referred to [24] for the derivation of the alternative method.

3.3 Analytical double-stance maps

In this section we present novel approximate solutions to the DS dynamics of both the lossy and lossless SLIP models. We address this open problem by solving, instead, the dynamics of the AT-SLIP model. As shown in the previous section, the resultant force exerted on the CoM of AT-SLIP and of SLIP in DS are approximately the same. Thereby, solving the equations of motion of the two systems results in approximately the same trajectories, when starting from the same initial conditions. We begin by deriving a complete solution for the lossy AT-SLIP and then focus on the special case of lossless system by taking two different approaches.

3.3.1 The Updated-momentum approach for the lossy AT-SLIP

The equations of motion of AT-SLIP in Cartesian coordinates (x, y) were derived in Section 3.2.2. Here, we transform them into polar coordinates (r, θ) for the sake of convenience, yielding:

$$m(\ddot{r} - r\dot{\theta}^2) = k_{\mathrm{a}}(r_{\mathrm{rest}} - r) - b_{\mathrm{a}}\dot{r} - mg\cos\theta, \tag{3.12}$$

$$\dot{p} = -k_{\mathrm{t}}\theta - b_{\mathrm{t}}\dot{\theta} + mgr\sin\theta, \tag{3.13}$$

where $p := mr^2\dot{\theta}$ is the angular momentum. The aim is to find a time domain solution for $r(t)$ and $\theta(t)$. The set of equations (3.12) and (3.13) represents a coupled nonlinear system of ODEs whose exact analytical solution is an open problem to date. To develop a workaround solution, we make assumptions (3.3) and (3.4), similar to the studies for the SS phase (see for example [45]), and seek a way to decouple the radial motion from the angular motion as follows.

Radial motion

Let us first apply assumption (3.4) in (3.12) and (3.13), leading to

$$m(\ddot{r} - r\dot{\theta}^2) = k_{\mathrm{a}}(r_{\mathrm{rest}} - r) - b_{\mathrm{a}}\dot{r} - mg, \tag{3.14}$$

$$\dot{p} = -b_{\mathrm{t}}\dot{\theta}. \tag{3.15}$$

Even this simplified version does not admit available analytical techniques. As such, we temporarily ignore the influence of torsional damping[3], resulting in the

[3] We remedy the situation in Section 3.3.1 by reinforcing a correction on the angular momentum.

conservation of the angular momentum:

$$\dot{p} \approx 0, \tag{3.16}$$

thereby the angular momentum can be represented by its value at the initial condition (i.e., $p \approx p_0 := mr_0^2\dot{\theta}_0$). The corresponding angular velocity, $\dot{\theta} \approx p/(mr^2)$, is substituted into (3.14) to yield:

$$\ddot{r} + b_\mathrm{a}/m\dot{r} + k_\mathrm{a}/mr - p^2/(m^2r^3) = -g + k_\mathrm{a}/mr_\mathrm{rest}. \tag{3.17}$$

Still in this form, the equation contains the term $1/r^3$ which precludes the application of standard analytical tools. Following assumption (3.3) and similar to [45], we approximate this term using a Taylor series expansion:

$$1/r^3|_{r=r_\mathrm{rest}} = 1/r_\mathrm{rest}^3 - 3/r_\mathrm{rest}^4(r - r_\mathrm{rest}) + \dots. \tag{3.18}$$

Substituting the first two terms into (3.17) yields the following equation:

$$\ddot{r} + 2\xi\hat{\omega}_0\dot{r} + \hat{\omega}_0^2 r = F. \tag{3.19}$$

To keep the presentation concise, hereon all new parameters used in the derivations are defined in Table 3.1, unless otherwise specified. The table also provides the reference to the equation in which the defined parameter is used for the first time.

The above equation is a linear inhomogeneous ODE whose solution, by setting the initial time to zero, can be determined as

$$r(t) = F/\hat{\omega}_0^2 + Me^{-\xi\hat{\omega}_0 t}\cos(\omega_\mathrm{d}t + \phi). \tag{3.20}$$

Differentiation with respect to time yields the following relation for the radial velocity:

$$\dot{r}(t) = -M\hat{\omega}_0 e^{-\xi\hat{\omega}_0 t}\cos(\omega_\mathrm{d}t + \phi + \phi_2). \tag{3.21}$$

Angular motion

For the angular motion, substitute $\dot{p} = mr^2\ddot{\theta} + 2mr\dot{r}\dot{\theta}$ into (3.15) and rearrange it to the following form:

$$\ddot{\theta}/\dot{\theta} = -2\dot{r}/r - b_\mathrm{t}/(mr^2). \tag{3.22}$$

Table 3.1: UM method parameters

Symbol	Definition	Eq.
ω	$:= p/(mr_{\text{rest}}^2)$	(3.19)
$\hat{\omega}_0$	$:= \sqrt{k_{\text{a}}/m + 3\omega^2}$	(3.19)
ξ	$:= b_{\text{a}}/(2m\hat{\omega}_0)$	(3.19)
F	$:= -g + r_{\text{rest}}k_{\text{a}}/m + 4r_{\text{rest}}\omega^2$	(3.19)
ω_{d}	$:= \hat{\omega}_0\sqrt{1-\xi^2}$	(3.20)
A	$:= r_0 - F/\hat{\omega}_0^2$	(3.20)
B	$:= (\dot{r}_0 + \xi\hat{\omega}_0 A)/\omega_{\text{d}}$	(3.20)
M	$:= \sqrt{A^2+B^2}$	(3.20)
ϕ	$:= \arctan(-B/A)$	(3.20)
ϕ_2	$:= \arctan(-\sqrt{1-\xi^2}/\xi)$	(3.21)
z_0	$:= r_{\text{rest}}^2\omega e^{(z_2\cos(\phi-\phi_2))}$	(3.25)
z_1	$:= \left(3r_{\text{rest}} - 2F/\hat{\omega}_0^2\right) b_{\text{t}}/(mr_{\text{rest}}^3)$	(3.25)
z_2	$:= -2Mb_t/(mr_{\text{rest}}^3\hat{\omega}_0)$	(3.25)
z_3	$:= -\omega_{\text{d}}\sin(\phi-\phi_2) - \xi\hat{\omega}_0\cos(\phi-\phi_2)$	(3.26)
z_4	$:= \cos(\phi-\phi_2)$	(3.26)
z_5	$:= z_1 + z_2 z_3$	(3.27)
z_6	$:= z_2 z_4$	(3.27)
ϕ_3	$:= \arctan\left(\omega_{\text{d}}/(z_5+\xi\hat{\omega}_0)\right)$	(3.28)
z_7	$:= z_0\left(3r_{\text{rest}} - 2F/\hat{\omega}_0^2\right)/r_{\text{rest}}^3$	(3.28)
z_8	$:= -2z_0M/\left(r_{\text{rest}}^3\hat{\omega}_0\right)$	(3.28)
z_9	$:= \theta_0 - z_8\sin\phi$	(3.28)
z_{10}	$:= z_8\hat{\omega}_0 e^{-z_6}/\sqrt{z_5^2 + 2z_5\xi\hat{\omega}_0 + \hat{\omega}_0^2}$	(3.28)
z_{11}	$:= \theta_0 - z_{10}\cos(\phi+\phi_3)$	(3.28)
z_{12}	$:= -z_{10}/\sqrt{z_5^2 + 2z_5\xi\hat{\omega}_0 + \hat{\omega}_0^2}$	(3.32)
z_{13}	$:= M^2\omega_{\text{d}}^2 b_{\text{a}}/(4\xi\hat{\omega}_0)$	(3.34)
z_{14}	$:= M^2\xi\hat{\omega}_0 b_{\text{a}}/2$	(3.34)
z_{15}	$:= M^2\omega_{\text{d}} b_{\text{a}}/4$	(3.34)
z_{16}	$:= z_{13} + z_{14}\cos^2(\phi+\phi_2) - z_{15}\sin(2\phi+2\phi_2)$	(3.34)
ϕ_4	$:= \arctan\left(\omega_{\text{d}}/(2z_5+\xi\hat{\omega}_0)\right)$	(3.35)
z_{17}	$:= 4Mz_0^2 b_{\text{t}} e^{-2z_6}/\left(r_{\text{rest}}^5\sqrt{\omega_{\text{d}}^2 + (2z_5+\xi\hat{\omega}_0)^2}\right)$	(3.35)
z_{18}	$:= z_0^2(3r_{\text{rest}} - 4F/\hat{\omega}_0^2) b_{\text{t}} e^{-2z_6}/(2r_{\text{rest}}^5 z_5)$	(3.35)

One can immediately take the integral over time from both sides:

$$\int \ddot{\theta}/\dot{\theta}\,\mathrm{d}t = -2\int \dot{r}/r\,\mathrm{d}t - b_\mathrm{t}/m\int 1/r^2\,\mathrm{d}t + const. \tag{3.23}$$

The term $1/r^2$, with r as defined in (3.20), is approximated by a Taylor series expansion:

$$1/r^2|_{r=r_\mathrm{rest}} = 1/r_\mathrm{rest}^2 - 2/r_\mathrm{rest}^3(r - r_\mathrm{rest}) + \ldots. \tag{3.24}$$

Substituting the first two terms into (3.23) and computing the integrals yields the following expression for the angular velocity:

$$\dot{\theta}(t) = z_0/r^2 e^{-\left(z_1 t + z_2 e^{(-\xi\hat{\omega}_0 t)}\cos(\omega_\mathrm{d} t + \phi - \phi_2)\right)}. \tag{3.25}$$

In order to derive a closed-form expression for $\theta(t)$, we need to compute the integral of $\dot{\theta}$ over time, which is not as simple. Here, we consider another approximation:

$$e^{(-\xi\hat{\omega}_0 t)}\cos(\omega_\mathrm{d} t + \phi - \phi_2) \approx z_3 t + z_4. \tag{3.26}$$

Applying this into (3.25) translates to:

$$\dot{\theta}(t) \approx z_0/r^2 e^{-(z_5 t + z_6)}. \tag{3.27}$$

Once again, we use an approximation of the term $1/r^2$ according to (3.24). Finally, integrating the resultant $\dot{\theta}$ over time yields[4]:

$$\theta(t) = \begin{cases} z_7 t + z_8\sin(\omega_\mathrm{d} t + \phi) + z_9, & \text{if } b_\mathrm{t} = 0, \\ -z_7/z_5 e^{-(z_5 t + z_6)} + & \\ z_{10} e^{-(z_5 + \xi\hat{\omega}_0)t}\cos(\omega_\mathrm{d} t + \phi + \phi_3) + & \\ z_7/z_5 e^{-z_6} + z_{11}, & \text{otherwise.} \end{cases} \tag{3.28}$$

Necessary corrections

The reduced representations (3.14) and (3.15), used as a basis for the above approximations, are only valid when the DS trajectories are symmetric. This is

[4] For b_t sufficiently close to zero, equation (3.28) results in invalid values, due to the presence of b_t in the denominators. According to our numerical analysis, this holds for $\xi_0 < 10^{-6}$, with $\xi_0 = b_0/(2\sqrt{mk_0}) = b_\mathrm{t}/\left(c^2\sqrt{mk_0}(r_\mathrm{rest}/l_\mathrm{rest})^2\right)$. However, for those values of ξ_0 one can safely ignore the effect of torsional damping by using the expression corresponding to $b_\mathrm{t} = 0$ condition.

only the case for steady-state walking. Even for this class of trajectories, the accuracy of the derived approximations can still be improved by a correction with respect to the total energy of the system.

To expand the domain of validity of our approximations to highly non-symmetric trajectories, we introduce two correction terms that compensate for the inaccuracies in the angular momentum: (i) the gravity-correction (denoted by p_{g}), inspired by [57]; (ii) a correction associated with the effect of torsional spring (denoted by p_{t}). The corrected angular momentum is computed as

$$\hat{p} = p + p_{\mathrm{g}} + p_{\mathrm{t}}, \tag{3.29}$$

which replaces p in the corresponding derivations in our approximations, giving a two-iteration form to the UM method, similar to the Arslan et al. extension [57] to the Geyer et al. method [45] reviewed in Section 2.4.2. The reset of this section presents the derivation of the above-mentioned correction terms.

The effect of gravitational force on the total angular momentum p_{g} can be approximately modeled by (2.12), which is repeated here for convenience:

$$p_{\mathrm{g}} = t_{\mathrm{e}} m g \bar{r} (\sin\theta_0 + \sin\theta_{\mathrm{e}})/2, \tag{3.30}$$

where $\bar{r}$ is the estimated average radial movement during the time interval from $t_0 = 0$ to t_{e}. This can be readily computed using (3.20):

$$\begin{aligned} \bar{r} &= 1/t_{\mathrm{e}} \int_0^{t_{\mathrm{e}}} r(t)\,\mathrm{d}t \\ &= F/\hat{\omega}_0^2 - M/(\hat{\omega}_0 t_{\mathrm{e}}) \left(z_4 - e^{-\xi\hat{\omega}_0 t_{\mathrm{e}}} \cos(\omega_{\mathrm{d}} t_{\mathrm{e}} + \phi - \phi_2)\right). \end{aligned} \tag{3.31}$$

An average effect of the torsional spring on the angular momentum can be computed as

$$p_{\mathrm{t}} = \int_0^{t_{\mathrm{e}}} -k_{\mathrm{t}} \theta(t)\,\mathrm{d}t.$$

Substituting θ from (3.28) results in

$$p_{\mathrm{t}} = \begin{cases} -k_{\mathrm{t}}(z_9 t_{\mathrm{e}} + z_7/2t_{\mathrm{e}}^2 - \\ z_8/\hat{\omega}_0(\cos(\hat{\omega}_0 t_{\mathrm{e}} + \phi) - \cos\phi)), & \text{if } b_{\mathrm{t}} = 0, \\ -k_{\mathrm{t}}\big(\big(z_7/z_5 e^{-z_6} + z_{11}\big)t_{\mathrm{e}} - \\ z_7/z_5^2 e^{-z_6}\big(1 - e^{-z_5 t_{\mathrm{e}}}\big) + \\ z_{12}\big(e^{-(z_5+\xi\hat{\omega}_0)t_{\mathrm{e}}} \cos(\omega_{\mathrm{d}} t_{\mathrm{e}} + \phi + 2\phi_3) \\ -\cos(\phi + 2\phi_3)\big)\big) & \text{otherwise.} \end{cases} \tag{3.32}$$

The derived p_g and p_t are substituted in (3.29) to give the corrected angular momentum $\hat{p}$ with which the corresponding derivations should be repeated.

Once the improved approximations are determined, there is still the possibility to increase the accuracy by introducing a correction with respect to the system energy. To do so, we need to derive a closed-form solution for the energy losses due to damping.

The amount of energy losses due to the axial damping of the AT-SLIP can be expressed as

$$E_\mathrm{br} = \int_0^{t_\mathrm{e}} b_\mathrm{a} \dot{r}^2 \,\mathrm{d}t. \tag{3.33}$$

Using the derived expression (3.21) for $\dot{r}$ and calculating the integral leads to the following simple expression:

$$\begin{aligned} E_\mathrm{br} =& e^{-2\xi\hat{\omega}_0 t_\mathrm{e}}\big(- z_{13} - z_{14}\cos^2(\omega_\mathrm{d} t + \phi + \phi_2) \\ &+ z_{15}\sin(2\omega_\mathrm{d} t + 2\phi + 2\phi_2)\big) + z_{16}. \end{aligned} \tag{3.34}$$

Similarly, one can obtain the following for the torsional damping of the AT-SLIP, by using (3.27) for $\dot{\theta}$:

$$\begin{aligned} E_\mathrm{bt} =& \int_0^{t_\mathrm{e}} b_\mathrm{t} \dot{\theta}^2 \,\mathrm{d}t \\ =& z_{17}\big(\cos(\omega_\mathrm{d} t_\mathrm{e} + \phi + \phi_4) e^{-(2z_5 + \xi\hat{\omega}_0)t_\mathrm{e}} - \\ & \cos(\phi + \phi_4)\big) + z_{18}\big(e^{-2z_5 t_\mathrm{e}} - 1\big). \end{aligned} \tag{3.35}$$

Now, $E_\mathrm{b} = E_\mathrm{br} + E_\mathrm{bt}$ accounts for the total energy losses of the AT-SLIP, which also gives an approximation of the energy lost in the DS phase of SLIP motion. Subsequently, this can be used to compute a corrected angular velocity at the end of DS phase (i.e., the liftoff state):

$$\hat{\dot{\theta}}_\mathrm{e} = \mathrm{sgn}(\dot{\theta}_\mathrm{e})\sqrt{|2/m(E_0 - V_\mathrm{e} - E_\mathrm{b}) - \dot{r}_\mathrm{e}^2|}/r_\mathrm{e}, \tag{3.36}$$

where E_0 and V_e are best obtained from the original SLIP model:

$$\begin{aligned} E_0 =& 1/2m\big(\dot{x}_0^2 + \dot{y}_0^2\big) + 1/2k_0(l_\mathrm{rest} - l_\mathrm{f0})^2 \\ &+ 1/2k_0(l_\mathrm{rest} - l_\mathrm{h0})^2 + mgy_0, \end{aligned} \tag{3.37}$$

$$V_\mathrm{e} = 1/2k_0(l_\mathrm{rest} - l_\mathrm{fe})^2 + 1/2k_0(l_\mathrm{rest} - l_\mathrm{he})^2 + mgy_\mathrm{e}, \tag{3.38}$$

with l_{i0} and l_{ie}, $i \in \{\mathrm{f}, \mathrm{h}\}$, are the legs' length at the initial (i.e., fore leg touchdown) and the end (i.e., hind leg liftoff) states, respectively.

This completes the approximate solution proposed in this study to the dynamics

of AT-SLIP system, which also qualifies as a solution to the DS dynamics of the lossy SLIP from any initial to any end states, including the touchdown to liftoff map. The accuracy of the presented approximations will be assessed in Section 3.5.

It has to be noted that although the two-iteration scheme presented above enables the inclusion of the effects of gravity and torsional spring, it causes a loss of continuity of the angular momentum at the start point. More specifically, the angular momentum at the start point is different than what is assigned by the initial condition. However, the difference is so small in the sense of approximation, and it can be safely ignored in practice.

Overview of the approximate DS map by the UM method

From the implementation point of view, it is worth summarizing the required calculations for the UM method in an algorithm. Given the system states at the initial condition s_0 and q_0, the following explains the derivation of the map step by step:

1. transform s_0 from Cartesian to polar coordinates;
2. compute the AT-SLIP parameters using (3.5), (3.6), (3.8), (3.9), (3.10) and (3.11);
3. assign to the angular momentum its value at the initial condition: $p \leftarrow p_0 = mr_0^2\dot{\theta}_0$;
4. compute the parameters listed in Table 3.1;
5. solve (3.20), (3.21), (3.28) and (3.25) for the intended end state;
6. compute necessary corrections using (3.30) and (3.32);
7. compute the updated angular momentum using (3.29);
8. redo steps 4 and 5 with the updated angular momentum;
9. compute the dissipated energies due to the axial (3.34) and torsional (3.35) damping of AT-SLIP;
10. apply energy correction (3.36);
11. transform $\begin{bmatrix} r_\mathrm{e} & \dot{r}_\mathrm{e} & \theta_\mathrm{e} & \dot{\theta}_\mathrm{e} \end{bmatrix}^T$ from polar to Cartesian coordinates.

We have implemented this algorithm in a Python script that can be accessed online [73].

3.3.2 The Perturbation-based approach for the lossless AT-SLIP

The UM method, presented in the previous section, effectively approximates the dynamics of both lossy and lossless AT-SLIP models. In this section, we propose a different approach to approximate the *lossless* AT-SLIP which, thanks to the available tools in perturbation theory, features a straightforward form. From the perspective of functional analysis, this is generally more effective than the UM method, in which the derivations need to be repeated once using the updated angular momentum.

Inspired by the study on the approximation of the SS map [47] and different from the previous method, we assume that the angular momentum is only conserved for the radial motion and not for the angular motion. Consequently, the radial motion equation is solved similarly to the previous method to yield the following relation:

$$r(t) = F/\hat{\omega}_0^2 + M\cos(\hat{\omega}_0 t + \phi), \tag{3.39}$$

For the angular motion, we start by substituting

$$\theta(t) = \nu(t)u(t) \tag{3.40}$$

into (3.13), with $\sin\theta \approx \theta$ and $b_\mathrm{t} = 0$ (because of the lossless case). This will help us to translate the equation into the form suitable for perturbation techniques. The resulting relation becomes:

$$\begin{aligned}\ddot{u} + (2\dot{\nu}/\nu + 2\dot{r}/r)\dot{u} + (\ddot{\nu}/\nu + 2\dot{r}\dot{\nu}/(r\nu) - g/r + \\ k_\mathrm{t}/(mr^2))u = 0.\end{aligned} \tag{3.41}$$

To be able to use the perturbation based solution, we choose ν such that the first derivative of u disappears:

$$2\dot{\nu}/\nu + 2\dot{r}/r = 0. \tag{3.42}$$

It is clear that $\nu = 1/r$ solves this equation. Substituting this into (3.41) yields the following relation between u and its second derivative:

$$\ddot{u} - (-\hat{\omega}_0^2 + \frac{F+g}{r} - \frac{k_\mathrm{t}/m}{r^2})u = 0. \tag{3.43}$$

To solve this equation analytically, we need to approximate the terms $1/r$ and $1/r^2$

with r as defined in (3.39). By defining $\varepsilon := \frac{M}{F/\hat{\omega}_0^2}$ and taking into consideration that ε remains close to zero (according to assumption (3.3)), one can obtain the following approximations:

$$\frac{1}{r} \approx \frac{1}{F/\hat{\omega}_0^2} - \frac{1}{F/\hat{\omega}_0^2}\varepsilon\cos(\hat{\omega}_0 t + \phi), \tag{3.44}$$

$$\frac{1}{r^2} \approx (\frac{1}{F/\hat{\omega}_0^2})^2 - 2(\frac{1}{F/\hat{\omega}_0^2})^2\varepsilon\cos(\hat{\omega}_0 t + \phi). \tag{3.45}$$

Applying these approximations in (3.43) translates to the following form:

$$\ddot{u} - (\mu - \varepsilon\delta\cos(\hat{\omega}_0 t + \phi))\, u = 0, \tag{3.46}$$

where:

$$\mu := \frac{1}{F/\hat{\omega}_0^2}\left(g - \frac{k_t/m}{F/\hat{\omega}_0^2}\right), \tag{3.47}$$

$$\delta := \mu + \frac{1}{F/\hat{\omega}_0^2}\left(F - \frac{k_t/m}{F/\hat{\omega}_0^2}\right). \tag{3.48}$$

Since it is assumed that ε remains close to zero, one can derive an analytical approximation for this equation using standard perturbation techniques. The solution is a power series in terms of ε:

$$u(t) = u_0(t) + \varepsilon u_1(t) + \varepsilon^2 u_2(t) + \cdots \tag{3.49}$$

Substituting this into (3.46) and balancing the terms with the same power of ε yields a series of linear time-invariant ODEs. We approximate the solution by only the first-order expansion:

$$\varepsilon^0: \quad \ddot{u}_0 - \mu u_0 = 0, \quad u_0(0) = u(0), \dot{u}_0(0) = \dot{u}(0), \tag{3.50}$$

$$\varepsilon^1: \quad \ddot{u}_1 - \mu u_1 = -\delta u_0\cos(\hat{\omega}_0 t + \phi), \tag{3.51}$$

$$u_1(0) = 0, \dot{u}_1(0) = 0.$$

Depending on the value of μ different conditions are possible:

- If $\mu > 0$:

Define $\lambda^2 = \mu$, and solve (3.50) to yield:

$$u_0(t) = c_1 e^{\lambda t} + c_2 e^{-\lambda t}. \tag{3.52}$$

The parameters c_1 and c_2 are determined by the initial conditions given in (3.50).

Having the solution of u_0, one can solve (3.51):

$$\begin{aligned} u_1(t) = & c_3 e^{\lambda t} + c_4 e^{-\lambda t} \\ & + \frac{c_1 \delta}{4\lambda^2 + \hat{\omega}_0^2} e^{\lambda t} \left(\cos(\hat{\omega}_0 t + \phi) - \frac{2\lambda}{\hat{\omega}_0} \sin(\hat{\omega}_0 t + \phi) \right) \\ & + \frac{c_2 \delta}{4\lambda^2 + \hat{\omega}_0^2} e^{-\lambda t} \left(\cos(\hat{\omega}_0 t + \phi) + \frac{2\lambda}{\hat{\omega}_0} \sin(\hat{\omega}_0 t + \phi) \right). \end{aligned} \tag{3.53}$$

Similarly, c_3 and c_4 are determined by the initial conditions specified in (3.51).

- If $\mu < 0$ and $\mu \neq -\hat{\omega}_0^2/4$:

By defining $\lambda^2 = -\mu$, and following the similar procedure as before:

$$\begin{aligned} u_0(t) &= c_1 \sin(\lambda t) + c_2 \cos(\lambda t), \\ u_1(t) &= c_3 \sin(\lambda t) + c_4 \cos(\lambda t) \\ &\quad + \frac{N\delta}{4\lambda^2 - \hat{\omega}_0^2} \left(-\frac{2\lambda}{\hat{\omega}_0} \sin(\hat{\omega}_0 t + \phi) \sin(\lambda t + \psi) \right. \\ &\quad \left. - \cos(\hat{\omega}_0 t + \phi) \cos(\lambda t + \psi) \right), \end{aligned} \tag{3.54}$$

where $N := \sqrt{c_1^2 + c_2^2}$, and $\psi := \arctan(-c_1/c_2)$.

The solution of u_0 and u_1 in especial cases $\mu = -\hat{\omega}_0^2/4$ and $\mu = 0$ are as follows:

- if $\mu = -\hat{\omega}_0^2/4$:

$$\begin{aligned} u_0(t) =& c_1 \sin(\lambda t) + c_2 \cos(\lambda t), \\ u_1(t) =& c_3 \sin\left(\frac{\hat{\omega}_0 t}{2}\right) + c_4 \cos\left(\frac{\hat{\omega}_0 t}{2} +\right) \\ & \frac{N\delta}{8\hat{\omega}_0^2 \sin\left(\frac{\hat{\omega}_0 t}{2}\right)} \left(-4 \sin(\hat{\omega}_0 t - \phi + \psi) + \right. \\ & 2\hat{\omega}_0 t (\cos(\hat{\omega}_0 t + \phi - \psi) - \cos(\phi - \psi)) + \\ & \sin(2\hat{\omega}_0 t + \phi + \psi) - \sin(\hat{\omega}_0 t + \phi + \psi) - \\ & \left. 3 \sin(\hat{\omega}_0 t + \phi - \psi) - \sin(\phi - \psi) \right); \end{aligned}$$

- if $\mu = 0$:

$$\begin{aligned} u_0(t) &= c_1 t + c_2 \\ u_1(t) &= c_3 t + c_4 \\ &\quad + \frac{\delta}{\hat{\omega}_0^2} \left(-\frac{2c_1}{\hat{\omega}_0} \sin(\hat{\omega}_0 t + \phi) + (c_1 t + c_2) \cos(\hat{\omega}_0 t + \phi) \right). \end{aligned}$$

Once u_0 and u_1 are computed, then the time domain solution of θ can be written as:

$$\theta(t) = u(t)\nu(t) = \frac{u_0(t) + \varepsilon u_1(t)}{r(t)}. \tag{3.55}$$

Equations (3.39) and (3.55) are closed-form relations that approximate the dynamics of the AT-SLIP, and hence the dynamics of the lossless SLIP in DS, in a single-iteration form. A final energy-based correction on the angular velocity at the end of motion will additionally increase the prediction accuracy, similarly to what presented in Section 3.3.1 for the UM method.

Concerning the lossy AT-SLIP, while the solution of ν would not be as simple, the subsequent derivations require more algebraic computations and additional simplifications. For this class of problems, we hence conclude that the proposed UM approach suits better.

3.4 The whole walking cycle

As discussed in Section 2.2, the SLIP walking gait is the composition of alternating SS and DS phases. In the present chapter, we have developed analytical approximate time domain solutions to the DS dynamics of the system in the presence of non-negligible damping. The relevant literature of such a solution to the SS dynamics was reviewed in Section 2.4.2.

A complete walking step, previously illustrated in Figure 2.1a, is depicted here in Figure 3.3 with more details. According to the definition of the touchdown event, it is easy to show that the touchdown instant solves the following equation:

$$l_{\mathrm{h}}(t_{\mathrm{t}}) \cos \phi_{\mathrm{h}}(t_{\mathrm{t}}) - l_{\mathrm{rest}} \sin \alpha = 0. \tag{3.56}$$

The system transitions back to the SS phase by the liftoff event, when the ground reaction force of the hind leg reduces to zero during decompression, that is when

$$k_0 \left(l_{\mathrm{rest}} - l_{\mathrm{h}}(t_{\mathrm{l}})\right) - b_0 \dot{l}_{\mathrm{h}}(t_{\mathrm{l}}) = 0. \tag{3.57}$$

Notice that in case of lossless SLIP, this relation will reduce to $l_{\mathrm{h}}(t_{\mathrm{l}}) - l_{\mathrm{rest}} = 0$ (i.e., the system lifts off when the hind leg is fully stretched, as discussed in Section 2.2.1).

Solving (3.56) and (3.57) for the touchdown and liftoff instances analytically is challenging, even by applying the derived approximations. However, one can solve these transition equations numerically, and the resulting solutions are feasible

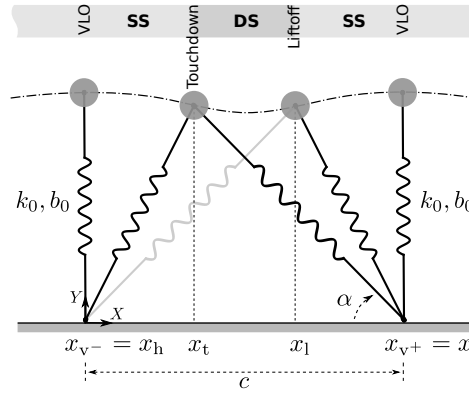


Figure 3.3: A complete step of SLIP walking.

owing to the one-dimensional and monotonic nature of the equations.

For the system depicted in Figure 3.3, every trajectory which is mirrored about the vertical passing $x = c/2$ is symmetric. It can be shown that for a symmetric DS of lossless SLIP, the correction terms (3.30) and (3.32) are naturally reduced to zero. Therefore, for this class of trajectories, the equivalent AT-SLIP model does not need the torsional spring, if the prediction of liftoff state (namely, the DS map) is desired. This means that the symmetric DS trajectories of the lossless SLIP can be approximated by the trajectories of a *monopedal* SLIP model endowed with certain parameters, as shown in Figure 3.4. This highlights the merits of the equivalency conditions achieved in this study.

3.5 Characterization of approximation errors

In this section, we investigate the performance of the proposed approximations by quantifying the prediction errors. First, the predicted liftoff states by our approximations, both for the lossless and lossy SLIP models in DS, are compared to the "ground truth" data, obtained via numerical integration of the original SLIP model. Then, we consider a whole walking step and assess the associated errors, which are contributed by both SS and DS approximations.

3.5.1 Simulation setting and performance criteria

The "ground truth" data are obtained by solving the system equations of motion using a hybrid solver that we have developed in Python 2.7 using the standard available modules. The solver is equipped with a variable step size integrator that captures the switching between phases, governed by (3.56) and (3.57), precisely.

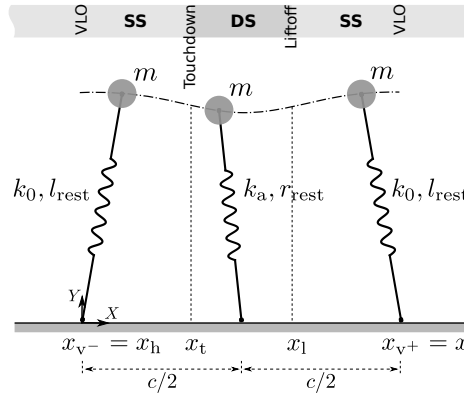


Figure 3.4: A symmetric lossless walking step can be approximated by the composition of "three SS phases". Note that $k_\mathrm{a} = 2\left(r_{\mathrm{rest}}/l_{\mathrm{rest}}\right)^2 k_0$, and $r_{\mathrm{rest}} = \sqrt{l_{\mathrm{rest}}^2 - c^2/4}$.

Interested readers can access the code online [73].

The simulation covers a set of most feasible locomotion properties, by spanning five dimensions: the horizontal velocity at VLO $\dot{x}_\mathrm{v}$, the height at VLO y_v, the legs stiffness k_0, the "relative" touchdown angle α_{rel}[5], and the damping ratio defined as $\xi_0 := b_0/(2\sqrt{mk_0})$. Table 3.2 gives the variation domains of these initial conditions and system parameters. Without loss of generality, we set $m = 80\,\mathrm{kg}$ and $l_{\mathrm{rest}} = 1\,\mathrm{m}$ similarly to average human data, and we assume the vertical velocity at starting VLO $\dot{y}_\mathrm{v}$ is zero.

A simulation is considered valid if (i) the system accomplishes one complete step of walking, and (ii) it is possible for the system to further touch down for the next step. To satisfy the first condition, trials in which the CoM hits the ground or moves backward, or when the system completely leaves the ground (airborne phase), are not stored. Concerning the second condition, denote by E_{min} the minimum amount of the mechanical energy at VLO required to make the next touchdown:

$$E_{\mathrm{min}} = mgl_{\mathrm{rest}} \sin\alpha + 1/2k_0 l_{\mathrm{rest}}^2 (1 - \sin\alpha)^2.$$

Then, the total energy at the starting VLO, E_v, should satisfy the following property:

$$E_\mathrm{v} \geq E_{\mathrm{min}}.$$

[5] Similarly to [48], the relative touchdown angle is defined as: $\alpha_{\mathrm{rel}} := \alpha - \alpha_\mathrm{n}$, with α_n being the *neutral* touchdown angle [50] resulting in a symmetric trajectory, for given initial state and system parameters.

Table 3.2: Initial conditions and system parameters spanned during the simulation

$y_{\mathrm{v}}(\mathrm{m})$	$\dot{x}_{\mathrm{v}}(\mathrm{m/s})$	$k_0(\mathrm{kN/m})$	α_{rel}	ξ_0
$[l_{\mathrm{rest}} \sin\alpha, l_{\mathrm{rest}}]$	$[0.5, 2]$	$[10, 40]$	$[-0.15, 0.15]$	$[0, 0.4]$

Parameter combinations resulting in E_{v} that violate this property are excluded. The total number of 70,117 runs satisfied the first condition, out of which 55,593 runs also satisfied the second condition.

To investigate the predictive performance, we use the following percentage error measures in different system states:

$$e_{\mathrm{p}} := 100 \frac{||(x,y)_{\mathrm{true}} - (x,y)_{\mathrm{approx}}||_2}{||(x,y)_{\mathrm{true}}||_2}, \tag{3.58}$$

$$e_{\dot{\mathrm{p}}} := 100 \frac{||(\dot{x},\dot{y})_{\mathrm{true}} - (\dot{x},\dot{y})_{\mathrm{approx}}||_2}{(||(\dot{x},\dot{y})_{\mathrm{true}}||_2)_{\max} - (||(\dot{x},\dot{y})_{\mathrm{true}}||_2)_{\min}}, \tag{3.59}$$

$$e_{\mathrm{E}} := 100 \frac{|E_{\mathrm{true}} - E_{\mathrm{approx}}|}{E_{\mathrm{true}}}, \tag{3.60}$$

which evaluate percentage normalized errors associated with position, velocity and total energy, respectively. When the predictive performance of the DS or the whole walking step is under the investigation, these measures are evaluated at the liftoff or VLO states, respectively. Notice that $e_{\dot{\mathrm{p}}}$, by definition, remains practical even when the norm of true velocity ($||(\dot{x},\dot{y})_{\mathrm{true}}||_2$) approaches zero, as the "min" and "max" norms are obtained from the entire range of simulation.

3.5.2 Performance of the lossless double-stance map

For the lossless SLIP model in DS, we have presented two different solutions following the UM and PB approaches in Section 3.3.1 and 3.3.2, respectively. To analyze the liftoff prediction errors, we provide both methods with the touchdown states of the ground truth data as the initial conditions.

The mean, standard deviation and maximum values of the resulting normalized errors are listed in the left part of Table 3.3. As can be seen, the liftoff states predicted by the PB approach are slightly more accurate than those of the UM approach. This can also be seen in Figure 3.5, in which the sum of the mean position and velocity errors ($e_{\mathrm{p}} + e_{\dot{\mathrm{p}}}$) are plotted with respect to the relative touchdown angle α_{rel}. While the UM solution errors increase for larger negative α_{rel}, the PB errors are almost uniform. Overall, both methods succeed in accurately approximating the dynamics of the original system for the purpose of control design. This will be demonstrated in the next chapter where controllers designed

Table 3.3: Summary of the percentage prediction errors analysis, covering both the UM and PB methods for the lossless and lossy SLIP models.

	Double-stance phase									Whole walking cycle		
	Lossless						Lossy					
	PB method			UM method								
	mean	sd	max	mean	sd	max	mean	sd	max	mean	sd	max
e_{p} (%)	0.17	0.78	22.36	0.27	0.37	13.21	0.47	0.58	13.21	0.71	0.67	17.59
$e_{\dot{\mathrm{p}}}$ (%)	0.61	1.09	35.15	0.83	1.39	31.17	1.56	1.79	28.45	1.05	1.62	22.86
e_{E} (%)	0	0	0	0	0	0	2.92	1.50	11.97	0.06	0.10	8.85

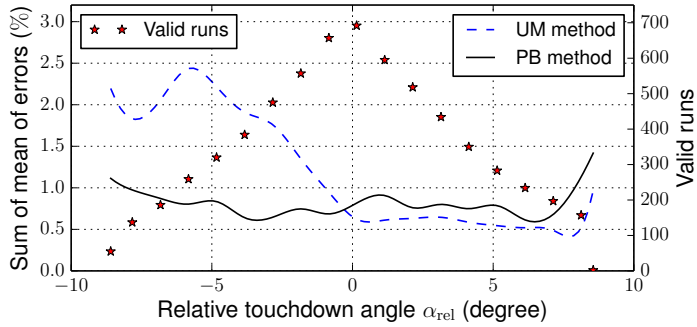


Figure 3.5: Lossless double-stance prediction errors are plotted versus the relative touchdown angle. For each bin, the number of valid runs, satisfying the requirements described in Section 3.5.1, are given in the right axis.

based on these approximations are presented.

3.5.3 Performance of the lossy double-stance map

In the presence of damping, the UM approach introduced in Section 3.3.1 is applicable. The corresponding numerical errors are reported in Table 3.3 and plotted with respect to the damping ratio in Figure 3.6. The error measures slightly increase for high damping ratios. This can mainly be due to the additional simplification (3.26) that negatively affects the derivation of both angular position (3.28) and the energy losses due to torsional damping (3.35).

3.5.4 Performance of the whole walking step

We now consider the composition of SS and DS phases which form a complete walking step, and investigate the predictive performance using the UM method. The corresponding normalized errors at VLO state are listed in the right part of Table 3.3. Note that from a control design point of view, the magnitude of these errors are important since they reflect the accuracy of the VLO return map (VRM)

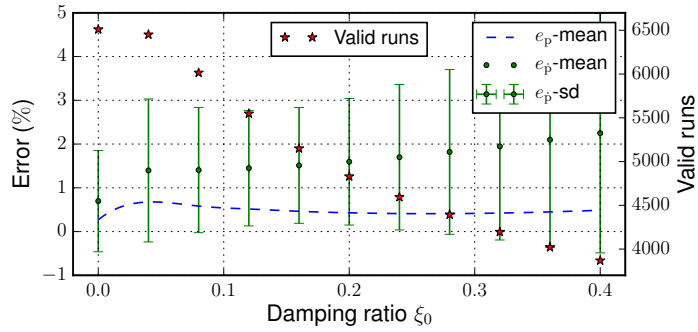


Figure 3.6: Lossy double-stance prediction errors are plotted versus the damping ratio. For each bin, the number of valid runs, satisfying the requirements described in Section 3.5.1, are given in the right axis.

defined in Section 2.2.2, which serves as the basis for the walking control policies presented in the next chapter.

The normalized position and velocity errors are depicted in Figure 3.7 with respect to the damping ratio. There is no considerable change in the prediction performance when damping is introduced, and the method effectively handles the presence of a rather large damping. As can be seen in Figure 3.7a, the predicted VLO positions are even slightly more accurate for higher damping ratios, owing to the resulting smaller leg compressions satisfying assumption (3.3). The dependence of the prediction errors on the maximum relative leg compression, defined as $(r_{\text{rest}} - r_{\min})/r_{\text{rest}}$, is depicted in Figure 3.8.

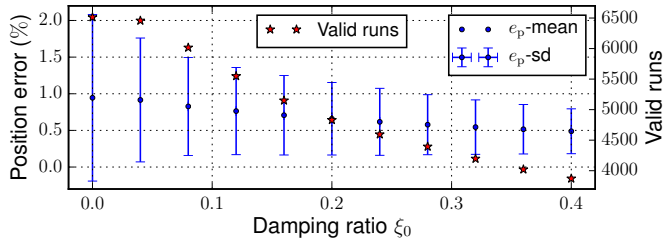
Furthermore, a meaningful correlation between the fixed points of the Poincaré section and the prediction accuracy of our approximations for the lossless SLIP can be observed[6]. Let d_{v} be a measure of the normalized distance between two subsequent returns at the Poincaré section[7]:

$$d_{\text{v}} = \left|\frac{y_{\text{v}^+} - y_{\text{v}^-}}{y_{\text{v}^-}}\right| + \left|\frac{\dot{x}_{\text{v}^+} - \dot{x}_{\text{v}^-}}{\dot{x}_{\text{v}^-}}\right|. \tag{3.61}$$

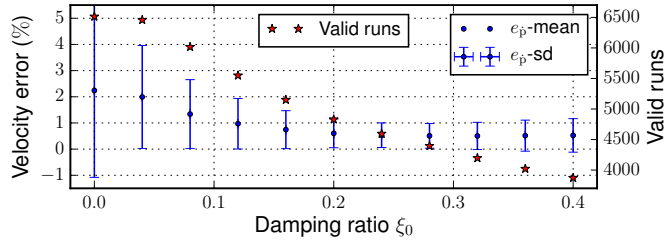
Figure 3.9 depicts the prediction errors e_{p} and $e_{\dot{\text{p}}}$ as functions of d_{v}. A small value of d_{v} indicates that the system state at the Poincaré section is close to a fixed point. It can be seen in the figure that, in this situation, the approximation error is small too, meaning that the prediction is more accurate when the system state converges to the fixed points, which are of particular importance in most cases. This also verifies the results presented in Section 3.5.2.

6 This simulation is a piece of the results presented in [21], in which the simulation setting is slightly different than what is detailed in Section 3.5.1. To keep the presentation concise, the readers are referred to [21] for more details.

7 Note that the conservation of the total energy dictates $(\dot{y}_{\text{v}^+} - \dot{y}_{\text{v}^-}) \to 0$, if $d_{\text{v}} \to 0$.



(a) Position errors



(b) Velocity errors

Figure 3.7: Approximate VLO return map errors are plotted versus the damping ratio. For each bin, the number of valid runs, satisfying the requirements described in Section 3.5.1, are given in the right axis.

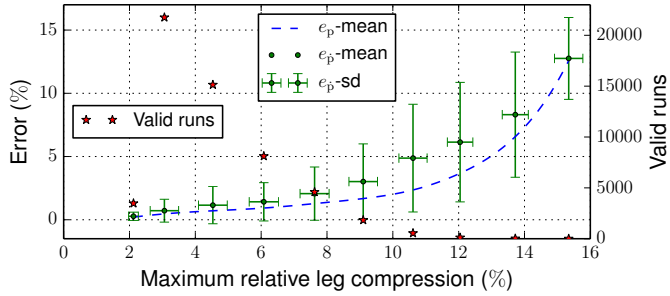


Figure 3.8: The dependence of the prediction errors on the maximum relative leg compression, defined as $(r_{\text{rest}} - r_{\text{min}})/r_{\text{rest}}$, is shown. For each bin, the number of valid runs, satisfying the requirements described in Section 3.5.1, are given in the right axis.

3.6 Conclusion

Analytic approximations of the otherwise non-integrable double-stance dynamics of the dissipative SLIP model were addressed in this chapter. We have followed two different approaches, one focusing on the inclusion of an explicit effect of damping (the UM method), while the other is intended to yield a closed-from solution (the PB method). At the core of both methods is the approximation of the

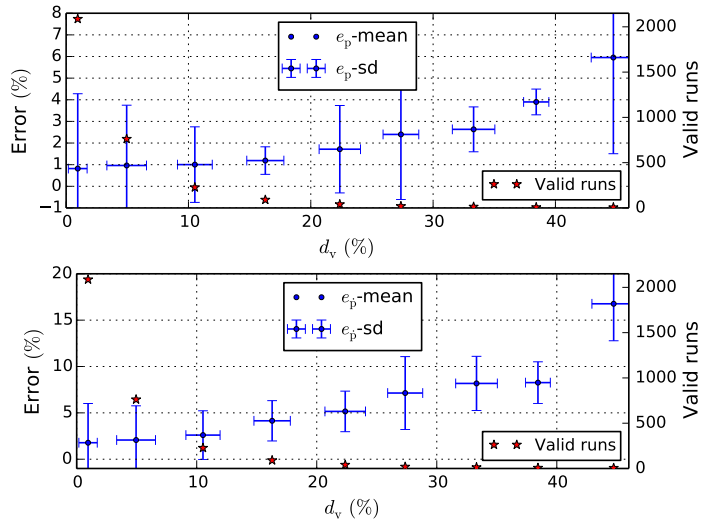


Figure 3.9: Correlation between the system states at the Poincaré section and the accuracy of the approximate VRM. For each bin, the number of valid runs, satisfying the requirements described in Section 3.5.1, are given in the right axis.

original bipedal SLIP trajectories by those of an auxiliary system referred to as the AT-SLIP. We derived a simple approximate mapping between the parameters of the two systems, which under certain conditions enables the AT-SLIP to accurately approximate the original SLIP in double-stance.

Assessed through a comprehensive numerical investigation covering highly non-symmetric trajectories with considerable amounts of damping, both methods provided accurate approximations of the original system dynamics. Due to the strong coupling in the SLIP equations of motion in double-stance, the study of these dynamics is significantly more complicated than the single-stance dynamics. However, we have achieved almost the same accuracy compared to the existing solutions to the single-stance dynamics. Furthermore, for the especial case of lossless SLIP model (with no damping), we presented two distinct methods which yield almost the same predictive performance, verifying the feasibility of the approaches taken.

The analytical perspective taken in this study opens avenues for tailoring the existing models in understanding humans and animals gaits in biomechanics, and for enhancing online planning and control methods for legged locomotion with non-instantaneous double-stance phases in robotics. The presented method can intuitively be extended to model the three-dimensional SLIP system, which has recently attracted significant interest in the field of dynamic walking, by incorporating another torsional spring and damper. In the next chapter, we present

control applications exploiting the predictive power of the proposed methods which, thanks to the simplicity resulted from the analytical nature of the solutions, are suitable for online implementations.

Chapter

4

Unified Control of Walking and Running on the SLIP Model

In the previous chapter the first approximations to the DS dynamics of SLIP were introduced. Based partly on these findings and within the framework established in Chapter 2, in the present chapter the control of walking, running and their transitions on the SLIP model is addressed.

Following a deadbeat-like scheme and beyond the inherent limitations of the passive SLIP model, control formulations for active running (Section 4.2) and walking (Section 4.3) are proposed. Subsequently, a biologically inspired control synthesis realizing the transitions between walking and running is presented in Section 4.4. The effect of approximations on the stability properties of the controlled system is carefully analyzed in Section 4.5. A number of simulation tests are provided to assess the utility and performance of the proposed framework (Section 4.6). The controlled SLIP information is also used in motion planning of a multi-body robot model (Section 4.7), and the chapter is concluded in Section 4.8.

4.1 Introduction

It is known that the passive SLIP model is able to perform *self-stable* walking and running. This surprising feature has inspired the development of a class of robots that exploit the SLIP-like morphologies or control (see [12] and the references therein). It is expected that such robots would have a superior performance at least in terms of power consumption, since no external input is involved in the passive SLIP motions. However, such a passive setup offers its own disadvantages as follows.

First, the process of extending this scheme to physical setups is not straightforward, as the real world effects, such as the touchdown impact, impose significant variations on the system energy. Second, it is known that the basins of attraction associated with the self-stable equilibriums of SLIP are quite limited, covering a comparatively small spectrum of the possible motions [36, 74, 75]. Finally, in addition to steady-state walking and running, a legged robot also needs to be capable of showing transient behaviors, such as gait transitions and accommodating unstructured terrains. Except in some rare situations [75], controlled transient motions in the context of passive SLIP have not been explored yet. Among others, the above-mentioned reasons clearly show the need for an active control over the SLIP walking and running gaits.

This chapter presents the development of a class of controllers, relying on the previously derived analytical maps, that realize active gaits and gait transitions. In all the considered applications, the control system adjusts the legs' stiffness in order to inject the required energy necessary for achieving desired locomotion properties. To use the analytical approximations presented in Chapter 3, we limit the variation of the leg compliance to discrete jumps, making the leg stiffness piecewise constant. According to the hybrid automaton representation proposed in Section 2.3, every node in Figure 2.3 corresponding to stance phases has a constant stiffness that is reset when switching between the nodes. Once needed, the control system also modulates the touchdown angle to govern the interchange between kinetic and potential energy. Note that in the literature, this type of controllers is sometimes referred to as deadbeat control[1] [77].

The presented control methodology highly relies on the prediction of the cyclic dynamics of the SLIP. Aiming at performing the control action online, approximate solutions are used for the predictions to reduce the computational time. However, the effect of consequent errors due to simplifications and approximations on the control performance needs to be analyzed. In this regard, an extensive numerical

[1] For a feasibility analysis of deadbeat control in the context of the SLIP hopper please refer to [76].

analysis was performed in the previous chapter to identify the characteristics of the errors statistically. In the present chapter, however, different control applications are provided to explore the effect of mentioned errors on the stability properties in action.

The class of controllers developed in this chapter are useful in gaining a better understanding of human motions in biomechanics and designing more sophisticated control methods for robots with SLIP-like morphologies. Of more significance, perhaps, is the opportunity to use the controlled SLIP information in motion planning for more complex human-like morphologies, embodying multi-body structures. Although for the running gaits this topic has received much attention (see for example [78, 79]), for the walking gaits it has remained fairly unexplored, mainly due to the complexities associated with the presence of DS phase. In this chapter, we also discuss the problem of SLIP-based planning for the realization of dynamic walking on a multi-body model, and extend the previous results by developing an embedding controller that enables a complete coordination between the dynamics of SLIP and real robot.

4.2 Active running

Unlike passive running examples, the goal of an active running on the SLIP model is to take the given system state at a Poincaré section z_{a^-} to *a desired* state at the subsequent return $z^*_{\mathrm{a}^+}$. To do so, according to Figure 2.3, the associated controller (i) assigns different stiffnesses for $\mathrm{SS}^{\mathrm{r}}_{\mathrm{sc}}$ and $\mathrm{SS}^{\mathrm{r}}_{\mathrm{sd}}$, denoted by $k^{\mathrm{r}}_{\mathrm{sc}}$ and $k^{\mathrm{r}}_{\mathrm{sd}}$, respectively; and (ii) adjusts the touchdown angle α^{r}, which is used in the reset map upon the execution of the guard $\mathrm{G}^{\mathrm{r}}_{\mathrm{t}}$. The resulting controller is used once per step at apex where $\mathrm{G}^{\mathrm{r}}_{\mathrm{a}}$ fires. To complete the design of this controller, we further impose a constraint motivated by biological observations. From the experimental results reported in [80], it can be seen that during running the ratio of the maximum leg length change to the rest length is almost the same (≈ 0.1) across all participants and all running speeds, suggesting the following constraint on the spring length at the bottom point:

$$r_{\mathrm{b}} = \sigma \circ s_{\mathrm{b}} = 0.9 l_{\mathrm{rest}}, \tag{4.1}$$

with σ being a projection operator on the corresponding continuous state s_{b}. Now we define the error equations as

$$\begin{aligned} z^{*}_{\mathrm{a}^{+}} - \mathcal{P}^{\mathrm{r}}(z_{\mathrm{a}^{-}}) &= \begin{bmatrix} e_1 & e_2 \end{bmatrix}^{T}, \\ 0.9 l_{\mathrm{rest}} - r_{\mathrm{b}} &= e_3, \end{aligned} \tag{4.2}$$

where

$$\mathcal{P}^{\mathrm{r}}(.) = ({}^{\mathrm{a}^{+}}_{\mathrm{l}}\mathcal{P}^{\mathrm{r}} \circ {}^{\mathrm{l}}_{\mathrm{b}}\mathcal{P}^{\mathrm{r}} \circ {}^{\mathrm{b}}_{\mathrm{t}}\mathcal{P}^{\mathrm{r}} \circ {}^{\mathrm{t}}_{\mathrm{a}^{-}}\mathcal{P}^{\mathrm{r}})(.),$$

and $e_i \in \mathbb{R}$, $i \in \{1, 2, 3\}$ being the corresponding errors. The control inputs $u = \begin{bmatrix} k^{\mathrm{r}}_{\mathrm{sc}} & k^{\mathrm{r}}_{\mathrm{sd}} & \alpha^{\mathrm{r}} \end{bmatrix}^{T}$ are then the solution of the following optimization problem:

$$u = \arg\min \sum_{i} \rho_i e_i^2, \; i \in \{1, 2, 3\}, \tag{4.3}$$

where $\rho_i \in \mathbb{R}$, $i \in \{1, 2, 3\}$, are weights of the corresponding criteria. They additionally normalize different terms of the cost function. In the simulation study reported later, we have set all the weights equal to one.

It is clear that the control inputs are bounded (i.e., the stiffness cannot take negative values and the touchdown angle is between 0° and 90°). These requirements can either be formulated as constraints imposed to the optimization problem (4.3) or their undesired values can be penalized by some additional terms in the cost function. We use the latter option as it gives more flexibility in the choice of the available optimization tools. Moreover, typical of nonlinear optimization problems, no guarantee for the existence or optimality of the solution can be given. However, our numerical investigations reveal a large domain of validity for the resulting solutions.

It has to be noted that a simplified approach for realizing active running can be found in the literature [47, 81], where $k^{\mathrm{r}}_{\mathrm{sc}}$ is manually chosen before executing the controller. Then $k^{\mathrm{r}}_{\mathrm{sd}}$ is derived from the difference between the total energy of the system at current apex and the desired total energy at the subsequent apex. Next, the ARM is used to derive α^{r} numerically. Although this method is computationally simpler, preassigning $k^{\mathrm{r}}_{\mathrm{sc}}$ may lead to unfeasible values for $k^{\mathrm{r}}_{\mathrm{sd}}$ causing high discrete jumps in the stiffness profile. In contrast, our method gives the controller the flexibility to choose all the control inputs simultaneously. Moreover, we also impose the constraint (4.1) to the maximum leg retraction which reinforces the similarity with biological evidence [80].

4.3 Active walking

Producing active walking in the bipedal SLIP model is more challenging than active running due to the additional complexity imposed by the DS dynamics, and other constraints [82] on the system state. Studies investigating varying speed walking are mostly limited to intuitive approaches where the ankle push-off and torso pitch are exploited to regulate the walking speed [82–86]. The presence of feet (and thus ankles torque) is a crucial element in most of the existing controlled limit cycle walking models.

Since our bipedal SLIP is an abstract model without feet, in the following we seek a simple method that enables the system to produce, to some degree, an active walking gait, based on the DS map introduced in the previous chapter. Similarly to active running, we focus on an adjustable stiffness scheme. While several events can be chosen as the moment either of the legs' stiffness is adjusted, we instantaneously adjust them at the virtual bottom defined in Section 2.2.1. In fact, we have devised the walking hybrid automaton in Figure 2.3 as the combination of four distinct modes such that it is suitable for the purpose of control design. In this sense, the virtual bottom event divides the DS phase of the bipedal SLIP into the virtual compression and decompression subphases, denoted by $\mathrm{DS}^{\mathrm{w}}_{\mathrm{sc}}$ and $\mathrm{DS}^{\mathrm{w}}_{\mathrm{sd}}$, respectively.

The control problem here is defined as follows: given the VLO state at Poincaré section z_{v^-}, the legs' stiffness and the touchdown angle are adjusted in such a way that the system converges to the desired state at the subsequent return $z^*_{\mathrm{v}^+}$. Note that several constraints influence the reachability set of $z^*_{\mathrm{v}^+}$. In each walking cycle, irrespective of the phase being SS or DS, the legs' stiffness before the virtual bottom k' is distinguished from that of after the virtual bottom k''. In other words,

$$k^{\mathrm{w}}_{\mathrm{av}} = k^{\mathrm{w}}_{\mathrm{sc}} = k', \quad \text{and,} \quad k^{\mathrm{w}}_{\mathrm{sd}} = k^{\mathrm{w}}_{\mathrm{bv}} = k''.$$

Following a similar process as for active running, we first define the error equations:

$$z^*_{\mathrm{v}^+} - \mathcal{P}^{\mathrm{w}}(z_{\mathrm{v}^-}) = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T, \tag{4.4}$$

where

$$\mathcal{P}^{\mathrm{w}}(.) = ({}^{\mathrm{v}^+}_{\mathrm{l}}\mathcal{P}^{\mathrm{w}} \circ {}^{\mathrm{l}}_{\mathrm{b}}\mathcal{P}^{\mathrm{w}} \circ {}^{\tilde{\mathrm{b}}}_{\mathrm{t}}\mathcal{P}^{\mathrm{w}} \circ {}^{\mathrm{t}}_{\mathrm{v}^-}\mathcal{P}^{\mathrm{w}})(.).$$

Then we solve (4.3) for the walking control inputs $u = \begin{bmatrix} k' & k'' & \alpha^{\mathrm{w}} \end{bmatrix}^T$ with e_i, $i \in \{1, 2, 3\}$, as defined in (4.4).

The feasibility of the numerical solution of the resulting optimization may become restricted in some situations. Therefore, we also consider a modified control method that reduces the computational complexity motivated by [81], however, by constituting a relatively smaller domain of attraction. Given the system energy at VLO and the desired system energy at the next VLO, the difference between these two has to be compensated by the stiffness change at virtual bottom. This yields the following relation between k' and k'':

$$k'' = \frac{\mu_4 + (\mu_3 - \mu_2)k'}{\mu_3 - \mu_1}, \tag{4.5}$$

with

$$\begin{aligned}
&\mu_1 = (l_{\text{rest}} - y^*_{\text{v}+})^2, \quad \mu_2 = (l_{\text{rest}} - y_{\text{v}-})^2,\\
&\mu_3 = (l_{\text{rest}} - l')^2 + (l_{\text{rest}} - l'')^2,\\
&\mu_4 = m\left((\dot{x}^*_{\text{v}+})^2 + (\dot{y}^*_{\text{v}+})^2 - (\dot{x}^2_{\text{v}-} + \dot{y}^2_{\text{v}-}) + 2g(y^*_{\text{v}+} - y_{\text{v}-})\right),
\end{aligned}$$

where l' and l'' are the fore and hind leg's length at virtual bottom, respectively. Now by presuming a reasonable value for k' (e.g., $k' \approx k_0$), the value of k'' can be immediately obtained using (4.5). Subsequently, the only remaining control parameter, $u = \alpha^{\text{w}}$, is obtained by solving the optimization problem (4.3) with e_i, $i \in \{1, 2, 3\}$, as defined in (4.4). A quantitative demonstration of the method will be given in the simulations presented in the next sections.

4.4 Walk-run transitions

Humans prefer to transition from walking to running at certain speeds. However, there is no unequivocal explanation as to why they change the gait [87]. It is also evidenced that the transition stride does not resemble either of the gaits [88], implying that this stride includes a sequence of transient dynamics that take the system state from a VRM to an ARM.

In the control literature, there are a few studies investigating walk-run transitions in the SLIP models. Hodgins [89] presented a simple leg lengthening approach, similar to Raibert's legs control (i.e., decoupled control of horizontal and vertical oscillations), that enables the walk-run transitions for a particular pair of walking and running speeds. Rummel et al. [51] identified an almost continuous morphing of gait patterns between walking and running in *passive* limit cycles at low and medium speeds. Martinez and Carbajal [75] detected a new intermediate gait pattern, hopping, that can connect a walking to a running limit cycle by

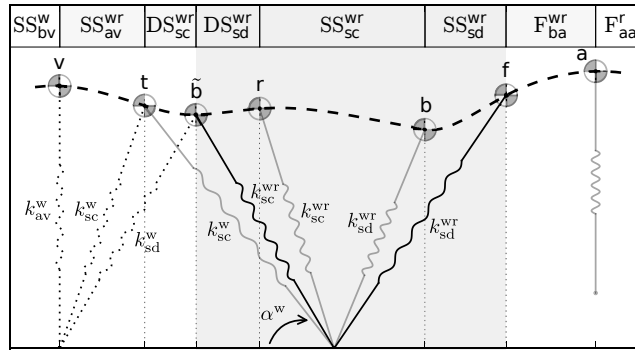


Figure 4.1: Schematic of walk-to-run transition (WRT). The controller is active in the shaded phases.

exploiting two different touchdown angles in each stride. Neither of these papers deals with the varying compliance, something that is evidenced in the human motion [90]. Moreover, for the realization of walk-run transitions, switching to an intermediate gait has not been identified in humans. Finally, humans and animals transition between walking and running during acceleration, the observation that we explore here on the SLIP model for the first time.

4.4.1 Walk-to-run transition

In this section, we synthesize a controller that takes the system state from walking to running. We take two biological observations into account: (i) the recorded trajectories of the humans' CoM during the walk-to-run transition (WRT) reported in [91]; and (ii) the experiment carried out in [88] showing that the transition is not an abrupt event, and it is triggered in the transition stride and completed after an adjustment period. Motivated by these, we consider a WRT stride as illustrated in Figure 4.1, and we hypothesize that the WRT process is triggered at virtual bottom. This means that we have no control on α^{w}. An important assumption we make only for the control calculation is that the current DS phase is suddenly terminated at virtual bottom and the system switches to a virtual SS phase. This phase ends with the liftoff event of the hind leg. In other words, in the design of the WRT controller we ignore the influence of the hind leg after virtual bottom by considering a simplified model.

The process of transitioning can be described as follows. The system starts the transition stride at a VLO for which a particular condition (namely, G^{wr}_{v}) instructs the transition to running. The system walks in the same pace as the previous walking stride until the virtual bottom event $G^{wr}_{\tilde{b}}$. At this moment, the stiffness of the fore leg is updated to a new value denoted by k^{wr}_{sc}, but the hind

leg still uses the stiffness assigned by the walking controller $k_{\mathrm{sd}}^{\mathrm{w}}$. The next update in the stiffness occurs when the fore leg is fully compressed corresponding to the bottom event, represented by guard $\mathrm{G}_{\mathrm{b}}^{\mathrm{wr}}$ in the hybrid automaton Figure 2.3. The new stiffness, denoted by $k_{\mathrm{sd}}^{\mathrm{wr}}$, is then used until the liftoff event of the fore leg $\mathrm{G}_{\mathrm{f}}^{\mathrm{wr}}$, leading the system to the $\mathrm{F}_{\mathrm{ba}}^{\mathrm{wr}}$ phase which ends at apex.

In the implementation side, we construct the WRT map mapping z_{v} onto z_{a}:

$$z_{\mathrm{a}} = \mathcal{P}^{\mathrm{wr}}(z_{\mathrm{v}}). \tag{4.6}$$

According to the hybrid automaton of the WRT in Figure 2.3, $\mathcal{P}^{\mathrm{wr}}$ is a composition of the following submaps: the VLO to touchdown (${}_{\mathrm{v}}^{\mathrm{t}}\mathcal{P}^{\mathrm{wr}}$), the touchdown to virtual bottom (${}_{\mathrm{t}}^{\tilde{\mathrm{b}}}\mathcal{P}^{\mathrm{wr}}$), the virtual bottom to hind leg liftoff (${}_{\tilde{\mathrm{b}}}^{\mathrm{h}}\mathcal{P}^{\mathrm{wr}}$), the hind leg liftoff to bottom (${}_{\mathrm{h}}^{\mathrm{b}}\mathcal{P}^{\mathrm{wr}}$), the bottom to fore leg liftoff (${}_{\mathrm{b}}^{\mathrm{f}}\mathcal{P}^{\mathrm{wr}}$), and the fore leg liftoff to apex (${}_{\mathrm{f}}^{\mathrm{a}}\mathcal{P}^{\mathrm{wr}}$):

$$\begin{aligned} z_{\mathrm{a}} = ({}_{\mathrm{f}}^{\mathrm{a}}\mathcal{P}^{\mathrm{wr}} \circ {}_{\mathrm{b}}^{\mathrm{f}}\mathcal{P}^{\mathrm{wr}} \circ {}_{\mathrm{h}}^{\mathrm{b}}\mathcal{P}^{\mathrm{wr}} \circ {}_{\tilde{\mathrm{b}}}^{\mathrm{h}}\mathcal{P}^{\mathrm{wr}} \circ {}_{\mathrm{t}}^{\tilde{\mathrm{b}}}\mathcal{P}^{\mathrm{wr}} \circ \\ {}_{\mathrm{v}}^{\mathrm{t}}\mathcal{P}^{\mathrm{wr}})(z_{\mathrm{v}}). \end{aligned} \tag{4.7}$$

The analytical maps studied in Chapter 3 are used in the derivation of the above relation. The control problem is to find the control inputs $u = \begin{bmatrix} k_{\mathrm{sc}}^{\mathrm{wr}} & k_{\mathrm{sd}}^{\mathrm{wr}} \end{bmatrix}^T$ that let the system converges to the desired system state at the subsequent apex z_{a}^*. As such, we first define the error equations as:

$$z_{\mathrm{a}}^* - \mathcal{P}^{\mathrm{wr}}(z_{\mathrm{v}}) = \begin{bmatrix} e_1 & e_2 \end{bmatrix}^T. \tag{4.8}$$

Then, the control inputs u are the solution of an optimization problem that takes the same form as (4.3), where e_i, $i \in \{1, 2\}$, are determined by (4.8).

In the control calculation, the assumption that the DS phase suddenly ends at virtual bottom is only made to reinforce the following biological evidence observed for humans in [92]: "in the transition stride the hind leg is walking and the fore leg is running". Motivated by this, we keep the walking controller for the hind leg and introduce the WRT controller for the fore leg. This means that the WRT controller uses a simplified model of the system dynamics (i.e., assuming no hind leg), while in the real simulation, the hind leg does push the system forward. As such, one must adapt the control system to compensate for this mismatch. Using physical insight and tuning in preliminary simulations, we have concluded that it is sufficient to adjust reference point z_{a}^*. Accordingly, in the simulation study presented in Section 4.6, we use a relatively smaller value for $\dot{x}_{\mathrm{a}}^*$ and a larger value for y_{a}^* in comparison to the expected values.

Another note here is that $\mathrm{SS}_{\mathrm{sc}}^{\mathrm{wr}}$ may also contain a decompression phase prior

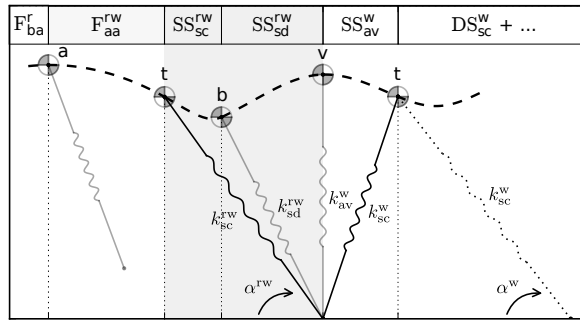


Figure 4.2: Schematic of run-to-walk transition (RWT). The controller is active in the shaded phases.

to the compression phase. But since it has no consequence in the synthesis of the controller, to keep it simple, we do not further partition $\mathrm{SS}^{\mathrm{wr}}_{\mathrm{sc}}$ into two phases.

4.4.2 Run-to-walk transition

In the run-to-walk transition (RWT), the running SLIP whose speed is decreasing transitions to the walking SLIP. We assume that the transition starts at the first apex whose horizontal velocity is less than the preferred RWT speed, and ends at the subsequent VLO. The RWT controller takes the running system state at apex z_{a} to the desired walking system state at VLO (z^*_{v}). Figure 4.2 details the procedure we devised for the transition process schematically. Without loss of generality, we assume that the system trajectory, which has an oscillatory behavior, encompasses only one bottom event in the RWT process. Consequently, the sequence of the subphases that the system takes during the transition forms the RWT automaton in Figure 2.3. The controller assigns (i) a particular constant stiffness $k^{\mathrm{rw}}_{\mathrm{sc}}$ for $\mathrm{SS}^{\mathrm{rw}}_{\mathrm{sc}}$ and $k^{\mathrm{rw}}_{\mathrm{sd}}$ for $\mathrm{SS}^{\mathrm{rw}}_{\mathrm{sd}}$, and (ii) the RWT touchdown angle α^{rw} when the guard $\mathrm{G}^{\mathrm{rw}}_{\mathrm{t}}$ is fired. To formalize the problem, we first construct the RWT map that maps z_{a} to z_{v}:

$$z_{\mathrm{v}} = \mathcal{P}^{\mathrm{rw}}(z_{\mathrm{a}}). \tag{4.9}$$

The map $\mathcal{P}^{\mathrm{rw}}$ is a composition of the following submaps: the apex to touchdown (${}^{\mathrm{t}}_{\mathrm{a}}\mathcal{P}^{\mathrm{rw}}$), the touchdown to bottom (${}^{\mathrm{b}}_{\mathrm{t}}\mathcal{P}^{\mathrm{rw}}$), and the bottom to VLO (${}^{\mathrm{v}}_{\mathrm{b}}\mathcal{P}^{\mathrm{rw}}$):

$$z_{\mathrm{v}} = ({}^{\mathrm{v}}_{\mathrm{b}}\mathcal{P}^{\mathrm{rw}} \circ {}^{\mathrm{b}}_{\mathrm{t}}\mathcal{P}^{\mathrm{rw}} \circ {}^{\mathrm{t}}_{\mathrm{a}}\mathcal{P}^{\mathrm{rw}})(z_{\mathrm{a}}). \tag{4.10}$$

For each of the associated submaps that belong to the three phases F, SS and DS, we derived a mathematical expression in Chapters 2 and 3. Similarly to the

previous applications, we write the error equations:

$$z_{\mathrm{v}}^{*} - \mathcal{P}^{\mathrm{rw}}(z_{\mathrm{a}}) = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T, \tag{4.11}$$

and then solve the optimization (4.3), with $e_i = i \in \{1, 2, 3\}$ as defined in (4.11). This yields the control inputs $u = \begin{bmatrix} k_{\mathrm{sc}}^{\mathrm{rw}} & k_{\mathrm{sd}}^{\mathrm{rw}} & \alpha^{\mathrm{rw}} \end{bmatrix}^T$ with which the system transitions from running to walking. A quantitative demonstration is presented in the simulation example reported in Section 4.6.1.

4.5 Stability analysis

For the SLIP model in running, any controller that modulates the touchdown angle and either the leg stiffness or the rest length grants the controllability of the system [50], with a reasonable basin of attraction [48]. Since the controller presented in Section 4.2 meets this requirement, we therefore omit the stability analysis of the running gait. On the other hand, the stability and robustness of the controlled SLIP walking introduced in Section 4.3 have not been studied. In particular, since the controller uses the novel approximate return maps, the effect of approximation errors on the robustness of the resulting controlled system needs to be carefully analyzed.

The stability of a fixed point of the VRM ($\mathcal{P}^{\mathrm{w}}$) is determined by studying the effect of a small perturbation in its neighborhood. This is typically approximated by using the corresponding Taylor expansion. If the eigenvalues of the associated Jacobian are inside the unit circle, then the intended fixed point is stable, thereby an associated basin of attraction exists. The relative location of the fixed point inside its basin of attraction and the size of the basin indicate how robust the controlled system is.

Obtaining the exact basin of attraction for the SLIP model is not always straightforward. An intuitive *steps-to-fall* method for obtaining an approximation of the domain has been used in [36, 74]. Through a numerical search in the neighborhood of the equilibrium point, all the initial conditions leading to a predefined minimum number of successive steps with the Poincaré states converging to the equilibrium are collected to form an approximation of the basin of attraction. Following this approach for the passive SLIP model, self-stable fixed points have already been identified [36, 74, 75]. Vejdani et al. [93] obtained a larger domain of attraction with provably the fastest convergence rate, by modulating the touchdown angle. This study is based on the fact that, in the context of deadbeat control, more than one walking step is required to grant control authority on the SLIP

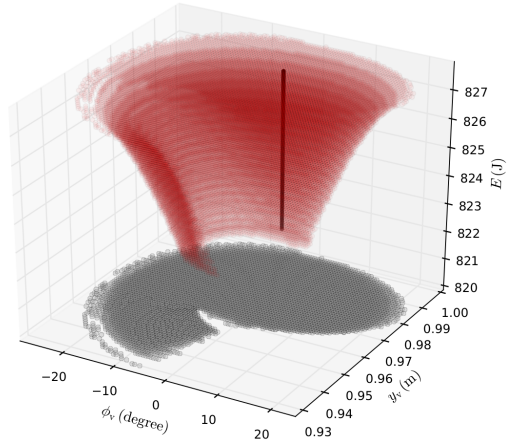


Figure 4.3: Self-stable basin of attraction of the intended set of equilibria depicted by the black line.

model [76]. The authors derived the domain of attraction of the two-step deadbeat stabilized gaits via backward-in-time integration of the system dynamics.

The exact pinpointing of the desired equilibrium is a strict and mostly unnecessary requirement imposed to the control system, which can readily be violated in many situations. In practice, it is sufficient if the controller takes the Poincaré state to the self-stable region of the equilibrium. Then by definition, the state will remain in this small neighborhood of the equilibrium and will asymptotically converge to it in a finite number of steps. Therefore, even for the one-step walking control, a meaningful domain of attraction can be obtained, if the exact pinpointing constraint is relaxed. This also enables the analysis of the effect of approximation on the control performance, while the deadbeat control of the system is generally infeasible when the dynamics are represented by approximate solutions.

A walking trajectory of the passive SLIP model can be fully determined by the Poincaré state (z_v), the dimensionless stiffness ($\tilde{k}_0 = k_0 l_\mathrm{rest}/(mg)$), the dimensionless energy ($\tilde{E} = E/(l_\mathrm{rest} mg)$), and the touchdown angle (α^w) [36][2]. To express the Poincaré state, one can use the triple $\begin{bmatrix} y_\mathrm{v} & \phi_\mathrm{v} & E \end{bmatrix}^T$ instead of the already defined $\begin{bmatrix} \dot{x}_\mathrm{v} & y_\mathrm{v} & \dot{y}_\mathrm{v} \end{bmatrix}^T$, as it is more convenient for the stability analysis, where ϕ_v denotes the velocity angle at the Poincaré section. The velocity magnitude v_v can be calculated from the system energy E, since the system is passive. Given an appropriate k_0, there exists a one-to-one relationship between E and α^w for a certain domain of E following which an equilibrium point is stable. We identified

[2] Without loss of generality, we shall use the dimensional counterparts by assuming the following numerical values: $m = 80\,\mathrm{kg}, l_\mathrm{rest} = 1\,\mathrm{m}, g = 9.81\,\mathrm{m/s^2}, k_0 = 16.5\,\mathrm{kN/m}$.

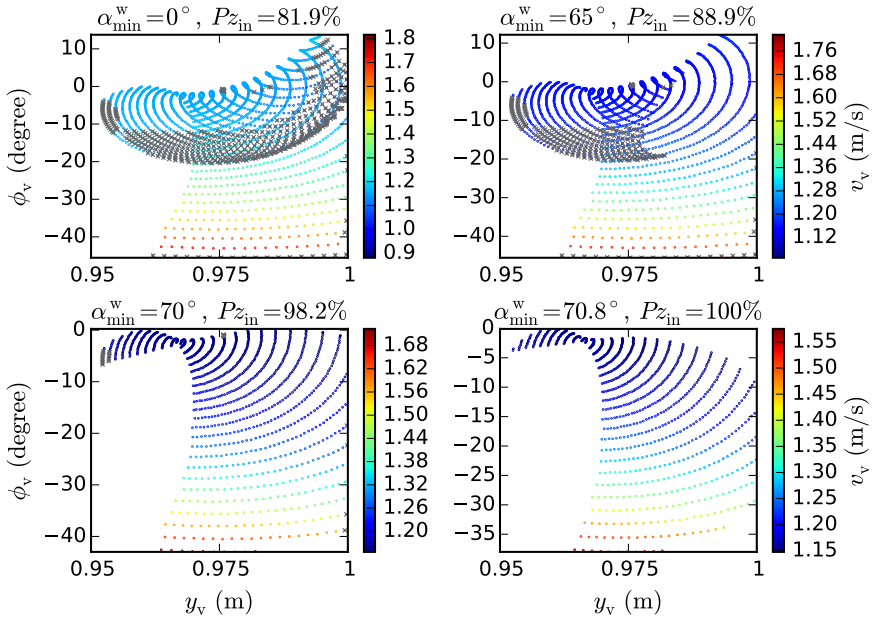


Figure 4.4: The effect of approximation on the basin of attraction for the proposed walking controller. The gray crosses indicate where the controller fails to stabilize the equilibrium.

the self-stable basin of attraction of the equilibrium $\bar{z}_{\mathrm{v}} = \begin{bmatrix} 0.9818\,\mathrm{m} & 0 & E \end{bmatrix}^T$ with $E \in \begin{bmatrix} 822.06\,\mathrm{J} & 827.62\,\mathrm{J} \end{bmatrix}$, by following the steps-to-fall approach[3], as shown in Figure 4.3.

We shall now identify the basin of attraction for the one-step deadbeat stabilization of the same equilibrium in the active SLIP walking, with the control policy following the modified method presented in Section 4.3. In doing so, we integrate the "exact" return map backward-in-time starting from the initial condition that coincides with the equilibrium ($\bar{z}_{\mathrm{v}} = \begin{bmatrix} 0.9818\,\mathrm{m} & 0 & 826\,\mathrm{J} \end{bmatrix}^T$), while spanning the control parameters $k'' \in \begin{bmatrix} 0.5k_0 & 1.5k_0 \end{bmatrix}$ and $\alpha^{\mathrm{w}} \in \begin{bmatrix} 0^\circ & 90^\circ \end{bmatrix}$. We repeat the same computation by using the derived approximate map following the UM method and construct a lookup table for the control inputs $u = \begin{bmatrix} k'' & \alpha^{\mathrm{w}} \end{bmatrix}^T$ accordingly. Now for the initial states belonging to the domain of attraction, we shall check to what extent the resulting state is inside the self-stable domain, if the control parameters of the approximate solution are used. Let Pz_{in} be the corresponding relative percentage, calculated accordingly.

[3] The minimum number of steps is set to 50. Also, note that the given specific value of vertical position at VLO, i.e., $y_{\mathrm{v}} = 0.9818\,\mathrm{m}$, was selected based on legacy.

Figure 4.4 illustrates the obtained results. As can be seen, by increasing the lower bound of the touchdown angle $\alpha^{\mathrm{w}}_{\mathrm{min}}$, which increases the validity domain of our approximations according to assumption (3.4), one can obtain larger Pz_{in}, so that for a certain lower bound ($\alpha^{\mathrm{w}}_{\mathrm{min}} = 70.8^{\circ}$) Pz_{in} reaches 100%. This means that the walking controller based on our approximate solution succeeds in stabilizing all the Poincaré states belonging to the domain of attraction, provided that the touchdown angle is bounded from below to that particular value. Although the approximate nature of the controller precludes the possibility of exact pinpointing, the state is brought to the self-stable domain such that y_{v} and ϕ_{v} converges to the equilibrium and v_{v} converges to a constant value in the domain $\left[1.11\,\mathrm{m/s} \quad 1.17\,\mathrm{m/s}\right]$. A similar numerical analysis can be performed for other equilibrium points representing human-like walking.

4.6 Simulation and results

In this section we demonstrate the utility of the proposed framework in simulation. In a first example, we simulate a locomotion scenario that humans usually follow in order to speed up and slow down using the lossless SLIP model. In the next two examples, explicitly provided for the SLIP walking, the focus is put on assessing the utility of the proposed approximations in Chapter 3 in practice. As such, small modifications on the presented unified control scheme may be made.

The simulations were carried out in Python 2.7, and the results were plotted in Matplotlib 1.3. All the optimizations and implicit equations involved in the control calculations, including those which govern the transition guards, were solved using the `scipy.optimize.root` function.

4.6.1 Example I: human-like motions on the controlled SLIP

According to Figure 4.5, the SLIP model starts walking with an initial condition close to (but not necessarily belonging to) a passive limit cycle with the average (per-stride) locomotion speed of $\dot{\bar{x}} = 1.17\,\mathrm{m/s}$ up to the time $t_{\mathrm{wr}} = 0.8\,\mathrm{s}$. The modified walking controller proposed in Section 4.3 guides the system in this phase. For the DS dynamics, the controller uses the derived approximate solution following the UM method.

At the first step after t_{wr} the WRT controller is activated. The control parameters $k^{\mathrm{wr}}_{\mathrm{sc}}$ and $k^{\mathrm{wr}}_{\mathrm{sd}}$, that lead to the WRT, as proposed in Section 4.4.1, are computed, and the transition takes place according to the hybrid automaton introduced in Figure 2.3.

The robot is then commanded to run with the average acceleration of $a = 1.05\,\mathrm{m/s^2}$, while keeping the same height as that of the walking gait ($y_{\mathrm{a}}^* = y_{\mathrm{v}}^* = 0.98\,\mathrm{m}$). The robot stops accelerating at a certain instant and keeps a constant locomotion speed of $\dot{\bar{x}} = 5\,\mathrm{m/s}$ for 1.5 s. Next, a decelerated run with the same rate is considered. All the running phases are controlled following the method described in Section 4.2.

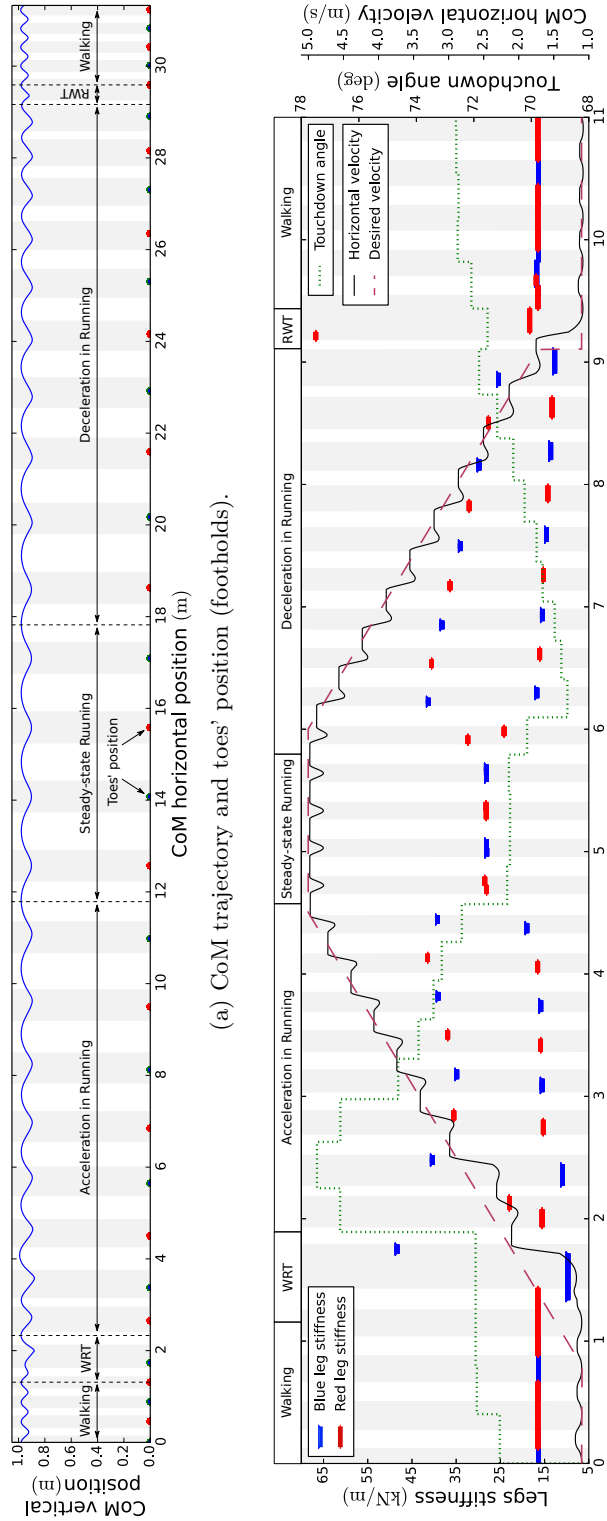
At the first apex for which the horizontal velocity is less than the preferred RWT speed (2 m/s) the RWT controller, proposed in Section 4.4.2, is invoked and the control parameters $k_{\mathrm{sc}}^{\mathrm{rw}}, k_{\mathrm{sd}}^{\mathrm{rw}}$ and α^{rw}, are calculated accordingly. Consequently, the robot transitions back to the initial walking limit cycle.

Results and discussion

The CoM trajectory together with the toes' positions are plotted in Figure 4.5a. Figure 4.5b shows the control parameters, namely the legs' stiffness and the touchdown angle. In addition, the horizontal CoM velocity and the set from which the desired horizontal velocities at apex/VLO are chosen, are depicted. Finally, Figures 4.6a and 4.6b illustrate the energetics of the system around the transition gaits, where the gravitational potential energy, the kinetic energy due to horizontal momentum, and the total system energy are plotted. In each plot, the corresponding SS phases are shaded and the gait markers are used. The detailed view for the WRT and RWT were already given in Figures 4.1 and 4.2. Also, Table 4.1 reports summary information on the variables used in the simulation along with the corresponding results.

As can be seen in the results, active walking and running gaits together with their transitions are realized effectively in simulation. In the following, we explore some of the important features that can be identified from the results.

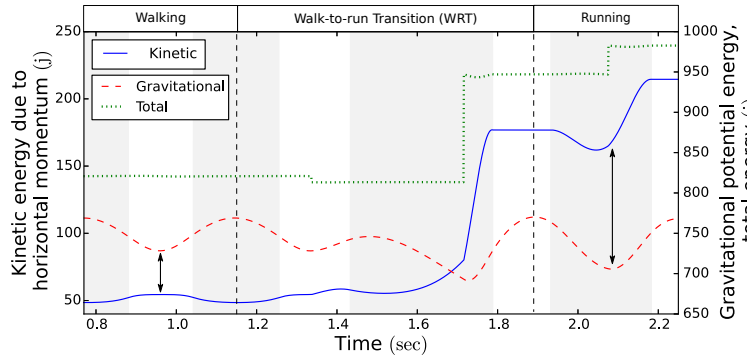
Relevance and feasibility analysis: In order to evaluate the feasibility of the results, we compare the leg stiffness obtained in our simulation to experimental observations. Even though the calculated stiffness in our work is entirely derived using (4.3) without a priori knowledge, there is a good match to the experimental values reported in Fig. 1 of [80] for the human. As an example, the mean value of the measured leg stiffness of several participants for the speeds of 4.5 and 5.5 m/s roughly equal to 30.8 and 31.4 kN/m, respectively. The computed leg stiffness in our method for the speed of 5.0 m/s is 28.2 kN/m (see Figure 4.5b), which seems relevant when compared to the experimental results. This match holds also for all other speeds reported, partly owing to the constraint (4.1) we have used in Section 4.2. Moreover, the trajectory of the CoM during the WRT, depicted in Figure 4.5a, resembles that observed for a human in [91] qualitatively.



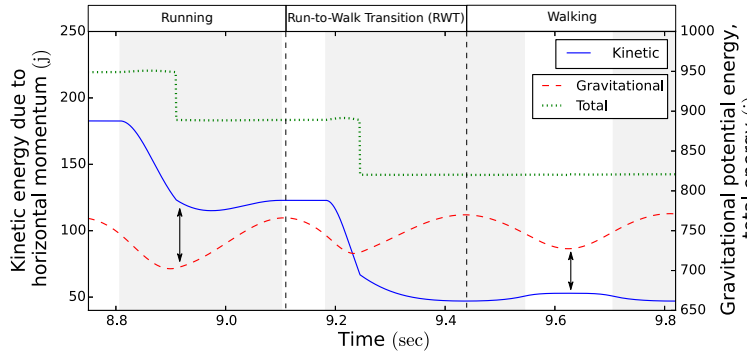
(a) CoM trajectory and toes' position (footholds).

(b) Control inputs together with the horizontal velocity of CoM. The dashed line is the set from which the desired velocity at apex/VLO is chosen.

Figure 4.5: Simulation of a bipedal SLIP model. The system starts in walking, switches to running with first increasing and then decreasing speed and finally transitions back to walking. The corresponding SS phases are shaded in each panel.



(a) Energetics of the system during WRT. The gravitational potential energy and the kinetic energy due to horizontal momentum are out-of-phase in walking (before the transition) and in-phase in running (after the transition).



(b) Energetics of the system during RWT. The gravitational potential energy and the kinetic energy due to horizontal momentum are in-phase in running (before the transition) and out-of-phase in walking (after the transition).

Figure 4.6: Analysis of the transition periods of a simulated SLIP model from an energy perspective.

As depicted in Figure 4.5b, in the accelerated run, the controller commands a smaller stiffness for the compression compared to the decompression phase, which increases the system energy level. Combined with the influence of the touchdown angle, this shortens the decelerating part and prolongs the accelerating part in each stance phase, which can be seen in the same figure. Consequently, the locomotion speed and the step length increase. In contrast, for the decelerated run, the stiffness in the compression phase takes a larger value, resulting in smaller steps with reduced speeds.

Energy-based analysis: There is a common way of defining the walking and running gaits in biomechanics which is related to dynamical characteristics: the involved fluctuations in kinetic energy due to forward momentum and gravitational

Table 4.1: System and control parameters

System parameters:			
Body mass	m	80	kg
Gravity acceleration	g	9.81	m/s^2
Leg rest length	l_{rest}	1	m
Initial condition:			
$\dot{x}$ at starting VLO	$\dot{x}_{\text{v}}$	1.1	m/s
y at starting VLO	y_{v}	0.98	m
$\dot{y}$ at starting VLO	$\dot{y}_{\text{v}}$	0	m/s
Initial control parameters:			
Leg stiffness	k_0	16.5	kN/m
Touchdown angle	α^{w}	70	degree
WRT (See Figure 4.1):			
$z_{\text{a}}^* = \begin{bmatrix}\dot{x}_{\text{a}}^* & y_{\text{a}}^*\end{bmatrix}^T = \begin{bmatrix}0.6 & 1.6\end{bmatrix}^T$			
$z_{\text{v}} = \begin{bmatrix}\dot{x}_{\text{v}} & y_{\text{v}} & \dot{y}_{\text{v}}\end{bmatrix}^T = \begin{bmatrix}1.101 & 0.9797 & -0.0442\end{bmatrix}^T$			
$[\alpha^{\text{w}} \quad k_{\text{sc}}^{\text{w}} \quad k_{\text{sd}}^{\text{w}}]^T\|_{\text{WRT-stride}} = \begin{bmatrix}71.94 & 16.50 & 16.50\end{bmatrix}^T$			
$[k_{\text{sc}}^{\text{wr}} \quad k_{\text{sd}}^{\text{wr}}]^T = \begin{bmatrix}9.860 & 48.51\end{bmatrix}^T$			
RWT (See Figure 4.2):			
$z_{\text{v}}^* = \begin{bmatrix}\dot{x}_{\text{v}}^* & y_{\text{v}}^* & \dot{y}_{\text{v}}^*\end{bmatrix}^T = \begin{bmatrix}1.1 & 0.98 & 0\end{bmatrix}^T$			
$z_{\text{a}} = \begin{bmatrix}\dot{x}_{\text{a}} & y_{\text{a}}\end{bmatrix}^T = \begin{bmatrix}1.754 & 0.9766\end{bmatrix}^T$			
$[\alpha^{\text{rw}} \quad k_{\text{sc}}^{\text{rw}} \quad k_{\text{sd}}^{\text{rw}}]^T = \begin{bmatrix}71.53 & 66.97 & 18.35\end{bmatrix}^T$			

potential energy are out-of-phase in walking and in-phase in running [91]. This can be seen in Figures 4.6a and 4.6b for the controlled system we developed in this study. Of particular interest are the transition periods where the shape of the mentioned fluctuations before and after the transition are different in result of the changes in the legs' compliance and touchdown angle.

Effect of approximation: The magnitude of approximation errors and its effect on the stability of the proposed controlled system were studied in Sections 3.5 and 4.5, respectively. As an example here, while the RWT controller is supposed to take the system states to $z_{\text{v}}^* = \begin{bmatrix}\dot{x}_{\text{v}}^* & y_{\text{v}}^* & \dot{y}_{\text{v}}^*\end{bmatrix}^T = \begin{bmatrix}1.1 & 0.98 & 0\end{bmatrix}^T$, due to the mentioned errors, the system converges to $z_{\text{v}} = \begin{bmatrix}1.085 & 0.9807 & -0.018\end{bmatrix}^T$ instead. If the controller was turned off in the subsequent limit cycle walking, the system would lose stability, although it could walk for a number of steps. The active walking system, however, compensates for the mentioned error so that the system can walk for an infinite number of steps.

4.6.2 Example II: controlled passive walking on the SLIP

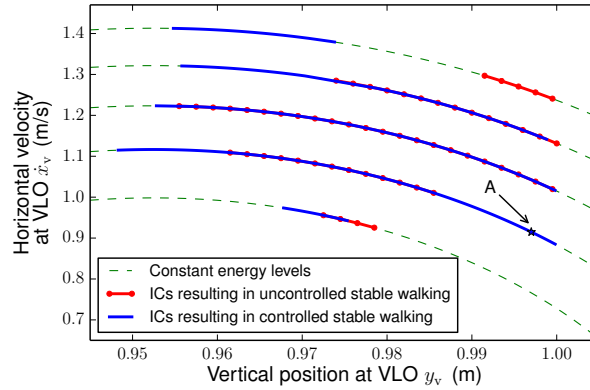
In order to assess exclusively the predictive power of the approximate solution based on the PB method, proposed in Section 3.3.2, we now consider a special case of the active walking controller presented in the current chapter. We start by the observation that the total energy of the system is invariant to the touchdown angle, since we assume no physical meaning for the swing leg. As such, modulating the touchdown angle to induce desired behaviors still results in a passive system. This way, the control inputs vector will only contain the touchdown angle: $u = \left[\alpha^{\mathrm{w}}\right]$, while $k' = k'' = 16.5\,\mathrm{kN/m}$. Finally, considering the control policy formulated in (4.4), we assume that the goal is to converge to the same state as the initial one, i.e., $z^*_{\mathrm{v}+} = z_{\mathrm{v}^-}$.

Figure 4.7a summarizes the simulation results (similarly to the previous example: $m = 80\,\mathrm{kg}$ and $l_{\mathrm{rest}} = 1\,\mathrm{m}$). The dashed lines depict the domain of initial conditions, with no vertical velocity, that belong to particular energy levels. The dotted lines represent subsets of this domain for which there is at least one (fixed) touchdown angle leading to a stable walking gait in the uncontrolled SLIP[4]. Similarly, the initial conditions belonging in the solid lines form the domain of stable walking in the controlled SLIP system. Note that running a sufficiently refined set of simulations would yield the aforementioned domains in the form of areas rather than the sets of points. However, such an extensive numerical study is beyond the scope of this example.

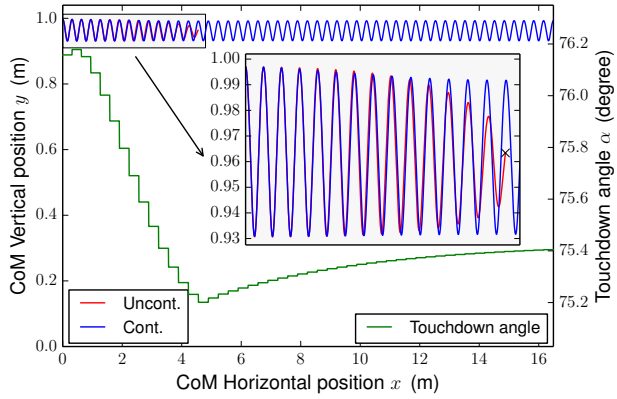
For the investigated energy levels, the controlled system achieves a larger domain of stability compared to the uncontrolled SLIP. This is worthwhile since the controlled system does not require extensive numerical search for the proper touchdown angle a priori. However, for relatively high energy levels, especially when the initial height is close to the rest length, the presence of the considered controller does not yield noticeable improvements. Here it seems that the approximation error propagated in the control calculation is so that the system experiences undesired transitions to the flight phase, or prevents foot protraction. Nevertheless, selection of a different desired state for the mentioned domains can still grant stable walking for the controlled system, as revealed in our investigations.

Finally, Figure 4.7b compares the CoM trajectories of the uncontrolled and controlled SLIP, starting with the initial condition of example point A given in Figure 4.7a. While the maximum number of steps that can be produced in the uncontrolled system occurs when the constant touchdown angle $\alpha = 76.1^\circ$ is used (14 steps), the controlled system produces stable walking. The control parameter,

[4] The angle is spanned from 65° to 90° with the delta increment 0.1°, and the gait is assumed stable if it can perform at least 50 steps in a row.



(a) Comparison of the domain of initial conditions resulting in stable walking gaits in the uncontrolled and controlled systems. The arbitrarily chosen amounts of energy levels from bottom to top are as follows: $\{806, 816, 826, 836, 846\}$, all are in Joules.



(b) CoM trajectories for the uncontrolled and controlled systems, starting with the initial condition of example point A. The control parameter (touchdown angle α) is plotted as well.

Figure 4.7: Simulation of a controlled SLIP model demonstrating the practical utility of the proposed DS map.

i.e., the varying touchdown angle, is depicted as well.

4.6.3 Example III: energy-regulated walking on the lossy SLIP

The control applications presented in the previous examples were provided for the lossless SLIP. The current example is explicitly designed to demonstrate the

practicality of the UM approach in the presence of damping forces. In such a case, it is clear that neither the uncontrolled SLIP nor the controlled passive SLIP (through modulation of the touchdown angle) can exhibit stable walking, due to the energy dissipation. Here, we devise a simple updating mechanism for the leg stiffness that, in the same vein as the controller proposed in Section 4.3, regulates the system energy in a step-to-step manner.

Consider a walking step starting at VLO with the initial continuous state vector $s_0 = \left[0, \dot{x}_{\mathrm{v}0}, y_{\mathrm{v}0}, 0\right]^T$, and the associated total energy $E_{\mathrm{v}0} = 1/2m\dot{x}_{\mathrm{v}0}^2 + 1/2k'(l_{\mathrm{rest}} - y_{\mathrm{v}0})^2 + mgy_{\mathrm{v}0}$. To compensate for the damping losses, we update the magnitude of stiffness at virtual bottom. The amount of energy dissipated from the starting VLO till the virtual bottom can readily be computed. For the rest of motion till the next VLO, however, we should predict the energy that will be dissipated. For that, we construct the map $\mathcal{H}(.)$ that maps the states at virtual bottom s_{b} onto the states at the next VLO s_{v}:

$$s_{\mathrm{v}} = \mathcal{H}(s_{\mathrm{b}}). \tag{4.12}$$

The UM approximations are used in the computation of the above derivation. The total energy at the next VLO is then calculated using the predicted states s_{v} as $E_{\mathrm{v}} = 1/2m\left(\dot{x}_{\mathrm{v}}^2 + \dot{y}_{\mathrm{v}}^2\right) + 1/2k'(l_{\mathrm{rest}} - y_{\mathrm{v}})^2 + mgy_{\mathrm{v}}$. Finally, the legs' stiffness are updated at the virtual bottom according to the following relation:

$$k'' = k' + 2(E_{\mathrm{v}0} - E_{\mathrm{v}})/\left((l_{\mathrm{rest}} - l')^2 + (l_{\mathrm{rest}} - l'')^2\right), \tag{4.13}$$

where l' and l'' are the legs' length at virtual bottom[5].

We have applied this energy regulation mechanism to the SLIP model used in the previous section with the same settings but in the presence of damping (with the damping ratio taking various values: $\xi_0 := b_0/(2\sqrt{mk_0}) \in \{0.05, 0.1, 0.2, 0.4\}$). Figure 4.8 shows the resulting domain of initial conditions leading to stable walking, even in the presence of non-negligible damping. Different colors indicate the largest damping value the system can handle (see the figure caption for a detailed description).

As can be seen, using the simple updating mechanism proposed, it is possible to perform stable walking gait still in the presence of damping, with comparatively large domain of attraction. Notice that, here again, the controller does not yield good results for very high energy levels for the same reason discussed in the previous example. However, for relatively low energy levels, the proposed control

[5] Notice that the updating mechanism formulated in (4.13) is slightly different from the one previously derived in (4.5). Here, the total energy at the next VLO is calculated using the same stiffness as the starting VLO (namely, k').

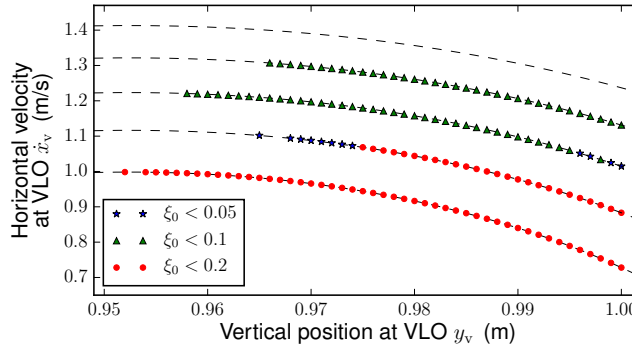


Figure 4.8: Domain of initial conditions resulting in stable walking gaits in the stiffness controlled lossy SLIP model. The arbitrarily chosen amount of energy levels from bottom to top are as follows: $\{806, 816, 826, 836, 846\}$, all are in Joule. Different colors (and markers) indicate the largest damping ratio the system can handle.

policy significantly enlarges the domain of initial conditions leading to stable walking, compared to the lossless passive system whose results was depicted in Figure 4.7a.

4.7 SLIP-like motions in multi-body robot walking

We now present an example of application to bipedal walking where the controlled SLIP model is used as a planner for the control of a multi-body robot model. The problem of controlling walking robots based on the bipedal SLIP model has been studied by taking a feedback linearization-based approach [17], and optimization-based approaches [6, 41]. The method we follow in this section is inspired by [17]; however, aimed at performing a maximal mapping between the real robot and the SLIP model, we extend the previous results. Therefore, we shall focus here on the specific contribution made, and refer readers to consult [17] for the description of the model and the detailed derivations of the equations of motion, contact model, and control policy.

The considered robot model is the standard planar five degrees of freedom (DoF) biped robot with point feet, as depicted in Figure 4.10. In order to have an appropriate coordinate variable that can be shared between the robot and its SLIP planner, we similarly use the relative angle β defined between the horizontal and the line connecting the hind stance foot to the CoM (see Figure 4.10). The trajectories planned by the SLIP model (namely the CoM position, velocity and acceleration) are parameterized as functions of β. Further, the generated swing

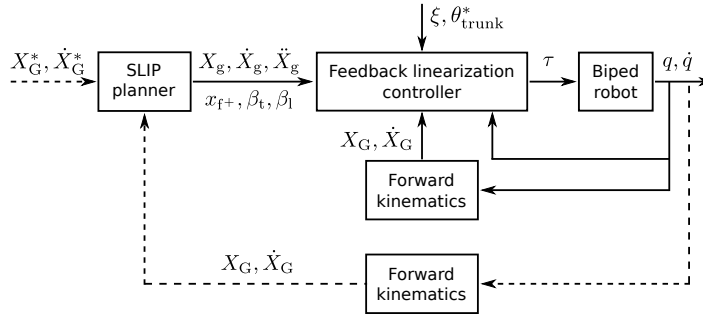


Figure 4.9: Block diagram of the overall control system developed to reproduce SLIP dynamics in a multi-body robot. The superscript '$*$' denotes the reference value.

foot trajectories[6] using the virtual constraint ξ, as defined in equation (13) of [17], are also expressed as functions of β.

4.7.1 Embedding controller

The structure of the overall control system is depicted in Figure 4.9. Continuous and event-driven signals are depicted in the solid and dashed lines, respectively. The SLIP planner, which uses the modified walking controller proposed in Section 4.3, is invoked once per step at VLO. It provides the controller with: (i) the CoM trajectories $(X_{\text{g}}, \dot{X}_{\text{g}}, \ddot{X}_{\text{g}})$, (ii) the next touchdown foot position (x_{f^+}), and (iii) the relative angle in the next touchdown and liftoff events $(\beta_{\text{t}}, \beta_{\text{l}})$. The latter two are used in the generation of the swing foot trajectories. The controller unit performs the following tasks:

- the SLIP trajectory tracking by the robot's CoM;
- the swing foot reference trajectory tracking by satisfying the virtual constraint ξ;
- the trunk stabilization at the desired posture.

Different from the approach taken in [17], we instruct the robot's CoM to track the SLIP trajectories not only in the acceleration level, but also in the position and velocity levels, leading to a complete coordination between the two systems. This is feasible owing to the analytical simplicity of the derived approximations, based on which the proposed walking controller is formulated. As such, the reference terms in the control policy (see equation (20) of [17]) need to be adapted. This is

[6] For the swing foot reference trajectory, we use circular arcs for the sake of demonstration. More sophisticated trajectories are available in the literature, whose implementation is, however, beyond the scope of this study.

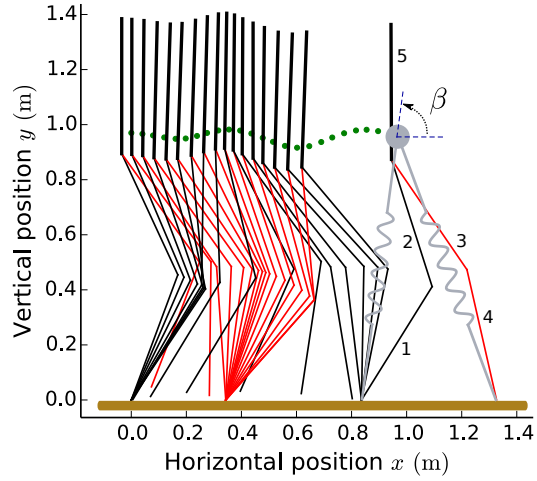


Figure 4.10: Stick diagram of the walking gait produced by the controller that uses the SLIP template information.

done by revising equation (21) of [17] to the following:

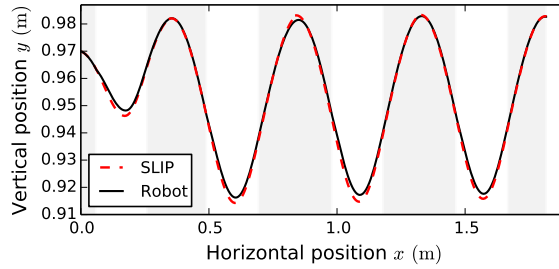
$$\ddot{X}_{\text{task}} = \begin{bmatrix} \ddot{X}_{\text{g}} + k_{P_{\text{g}}}(X_{\text{g}} - X_{\text{G}}) + k_{D_{\text{g}}}(\dot{X}_{\text{g}} - \dot{X}_{\text{G}}) \\ -K_{P_\xi}\xi - K_{D_\xi}\dot{\xi} \\ \ddot{\theta}^*_{\text{trunk}} + k_{P_{\text{trunk}}}\tilde{\theta}_{\text{trunk}} + k_{D_{\text{trunk}}}\dot{\tilde{\theta}}_{\text{trunk}} \end{bmatrix}, \tag{4.14}$$

where $X_{\text{g}} = \begin{bmatrix} x & y \end{bmatrix}^T$ and $X_{\text{G}} = \begin{bmatrix} x_{\text{G}} & y_{\text{G}} \end{bmatrix}^T$ are the vector of CoM positions for the SLIP and robot model, respectively, $k_{P_{\text{g}}}$ and $k_{D_{\text{g}}}$ are proportional and derivative gains associated with the position and velocity errors, and the rest as defined in [17].

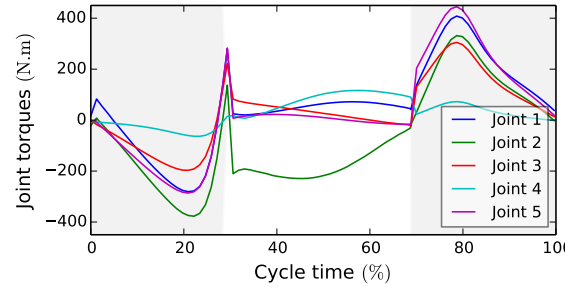
4.7.2 Results and discussion

We implement the presented control strategy on the multi-body robot model in simulation. For the sake of comparison, we choose the same numerical values for the simulation parameters as in [17]. The initial condition ($z_0 = \begin{bmatrix} 0.9695\,\text{m} & 1.9^\circ & 821.55\,\text{J} \end{bmatrix}^T$) is chosen from the obtained basin of attraction in Figure 4.4, thereby the stabilization of the equilibrium point in the SLIP planner is guaranteed. Based on the simulation results, we show that the real robot also stabilizes *the same* equilibrium point.

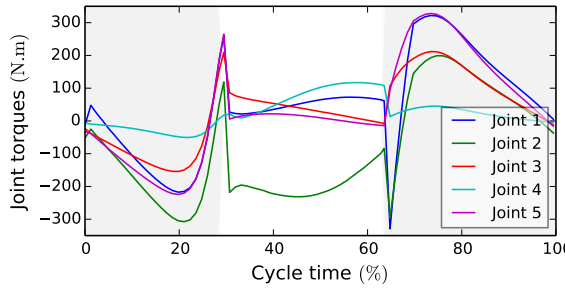
Figure 4.10 shows the stick diagram of the robot walking produced by the presented control strategy. The green dots are the positions of the robot's CoM.



(a) CoM trajectories.



(b) Joint inputs with the same liftoff states as those of SLIP.



(c) Joint inputs with predefined liftoff states.

Figure 4.11: Simulation results of embedding SLIP dynamics into the multi-body robot. The corresponding SS phases are shaded in each panel.

One can see that they perfectly coincide with the SLIP positions as compared in Figure 4.11a. Note that the method proposed in [17] fails to stabilize the same equilibrium point. This full control authority is useful especially for the transient behaviors such as gait transitions, for which the computational burden of planning can partly be passed to the SLIP control, as investigated in this chapter.

As shown in Figure 4.9, the hind leg of the robot lifts off at the same configuration (namely, the same β_l) as that of SLIP. Aimed at reducing the required

control inputs, we also explore an independent condition in which the leg lifts off at a predefined configuration. The corresponding control inputs (namely, joint torques) for a single step of walking for both cases are plotted in Figures 4.11b and 4.11c. It is clear that although the torques' amplitudes in the latter case are reduced, the associated profiles at the liftoff transition are smoother when the SLIP information is used.

It seems that even the reduced torques are still rather large. Notice that the simulation model is instructed to walk at around normal pedestrian speeds, which is a relatively challenging task for an 80 kg humanoid. Nevertheless, we suspect that optimizing the distribution of physical parameters and the swing leg dynamics can additionally reduce the control inputs, which is an interesting topic for future research.

4.7.3 Remark on stability

Since the biped robot is instructed to exactly reproduce the controlled SLIP dynamics, its stability follows the same properties. However, due to external disturbances such as the effect of touchdown impact and uncertainties, the robot can encounter robustness issues. Nevertheless, the robustness properties of the feedback linearization-based controllers have extensively been studied, and we therefore invite interested readers to consult the relevant literature.

4.8 Conclusion

This chapter presented a unified framework for the study of walking, running and walk-run transitions on the bipedal SLIP. By assuming a piecewise constant profile for the stiffness of the bipedal SLIP, together with the modulation of the touchdown angle, one can synthesize controllers that create varying speed gaits and their transitions. Choosing different stiffness for the spring compression and decompression phases grants the required changes needed in the system's total energy. In addition, the leg touchdown angle can modulate the interchange between kinetic and potential energy, being necessary for locomotion speed regulation.

The stability/robustness of the resulting controlled system was analyzed numerically. The negative effect of approximations on the robustness of the controller was compensated by subjecting the controllable touchdown angle to a certain lower bound. In other words, the proposed approximations in this thesis are more effective for the human-like walking gaits (i.e, for larger touchdown angles). For the deadbeat-form controllers, we also expanded the existing results on the stability analysis by leveraging on the self-stability property of the SLIP model.

Simulation tests verified the practicality of the synthesized controllers in producing human-like walking and running. Additionally, control designs meant for evaluation of the predictive power of the approximations, introduced in the previous chapter, showed significant promise even in the presence of non-negligible damping forces. Also simulated was an embedding of the controlled SLIP on a more realistic walking model, following a feedback linearization-based approach.

Concerning the control calculations, the presented method utilizing the approximate analytical maps is substantially faster than the methods based on numerical integrations, offering a feasible planning strategy for online purposes. Moreover, as shown in the simulation of multi-body robot walking, when the controlled SLIP information is used in the motion planning of the real robot, the resulting control inputs are smoother. Finally, it is suspected that a co-design of the robot's morphology and control based on the SLIP counterpart can additionally improve the results in terms of actuation and speed limits.

Chapter

5

Coordination of Monopedal SLIP Models Towards Quadrupedal Running

In the previous chapter, a class of unified controllers for the bipedal walking and running and respective transitions of the SLIP model was developed, and the SLIP-like motions were embedded into a realistic multi-body walking model. This chapter extends these results to the context of quadrupedal running.

We start by observing that a single SLIP model could not be a sufficiently descriptive template for quadrupedal running, and a novel template, called the dual-SLIP model, is introduced as such (Section 5.2). The relevance of the new template for different quadrupedal gaits is discussed, and the intended gaits to be realized in this chapter are described within the max-plus methodology reviewed in Section 2.5.4. Next, the development of a coordination controller that enables the dual-SLIP model to perform quadrupedal running is presented in detail (Section 5.3). In particular, a *time-aware* deadbeat controller is developed that extends the functionality of the controller presented in the previous chapter. Subsequently, a number of simulation experiments on the proposed template are provided (Section 5.4): the realization of quadrupedal pronking, bounding, and the respective transition; and the robustness evaluation against ground height variations. Finally, the concluding remarks are given in Section 5.5.

5.1 Introduction

The study of legged locomotion using template models provides insights to the dominant features of the dynamics while abstracting out less important details [13]. The resulting platform-independent reductive models typically admit analytical representations that are more tractable than the original higher-order robot models. On the other hand, oversimplified representations are not preferred as their anchoring [12] on the real robot would be overcomplicated. Hence, the choice of a sufficiently descriptive template for a given robot is of particular importance.

The SLIP model, thoroughly studied in this thesis as a template for bipedal walking and running, represents a body with a point mass, thereby cannot be solely useful in capturing the pitch dynamics [94] which play an important role in quadrupedal running motions. Having realized this, a number of template models have been proposed exclusively representing quadrupedal running systems, most of which assume rigid torsos [95–100], while only a few recent ones feature flexible torsos [101, 102].

Cao and Poulakakis [101] focused on investigating the consequences of torso flexibility on quadrupedal template running, with an emphasis on passively generated bounding gaits. Similarly, Pouya et al. [102] analyzed different types of actuation in the spine of a segmented torso in simulation to evaluate the bounding gaits characteristics such as periodicity and stability. In both studies, a successful gait generation depends on the identification of proper limit cycles for which extensive numerical search is often needed.

In this chapter we approach the problem of generating quadrupedal running in a template setting from a fresh point of view. The template model we propose, referred to as the dual-SLIP model, is composed of a pair of physically-unconnected SLIPs each representing a part of the body of a quadruped (see Figure 5.1). We show that if the fore and hind SLIPs are coordinated properly, the dual-SLIP template can represent different steady and transitional quadrupedal running behaviors. To the best of author's knowledge, such a "spatio-temporal" coordination framework for the SLIP model remained unexplored in the related literature. Compared to the aforementioned studies, the significance of the proposed approach is that it does not need an attractor limit cycle of the whole system to be identified a priori, and that it can benefit from the mature literature of SLIP for the low-level control structures.

For the coordination of individual SLIPs in the dual-SLIP model, we use the max-plus based methodology presented in Chapter 2. This particular choice was motivated after reviewing the alternative methods in the same chapter. In short,

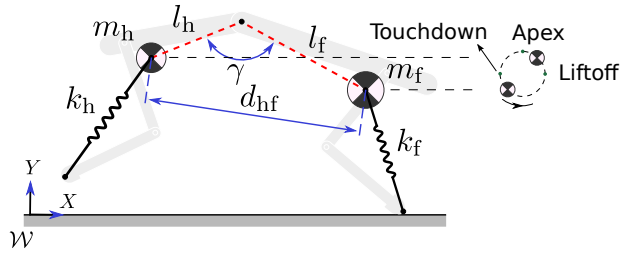


Figure 5.1: dual-SLIP template model for quadrupedal running.

the max-plus based methodology describes the evolutions of the coordination parameters using simple linear difference equations, making the interpretation of the resulting behaviors easier, particularly for transient phases. The success of the coordination framework proposed relies also on the presence of a low-level controller that is capable of realizing spatio-temporal references. In this regard, we also contribute to the existing control methods of SLIP by proposing a *time-aware* deadbeat control scheme, explicitly accounting for the control of certain event timings.

5.2 The Dual-SLIP template model

5.2.1 Model description

As shown in Figure 5.1, the proposed dual-SLIP model consists of a fore and hind SLIP models representing the fore and hind bodies of a quadruped. We assume the mass of fore (hind) legs and the respective part of torso are lumped in m_f (m_h). As depicted in the dashed red lines, a virtual articulated massless mechanism is also considered to loosely simulate a torso. This *virtual* torso has no effect on the system dynamics and is meant only for evaluating the feasibility of the model. Nevertheless, the effect of actual torso is implicitly captured in the states of m_f and m_h.

The considered individual SLIP models are of the standard form described in Section 2.2.1 and depicted in Figure 2.1, reproduced here with more details in Figure 5.2. Since throughout this chapter the focus is on the running gaits, we shall skip the gait marker in the notation. Also note that in order to capture the coordination parameters, the discrete state vector is adapted to $q := \begin{bmatrix} x_f & y_f & \mathcal{M} & \mathcal{Q}_j & \mathcal{Q}_e \end{bmatrix}^T$, where $\mathcal{M} \in \{descent, compr., decompr., ascent\}$, and $\mathcal{Q}_j$ and $\mathcal{Q}_e$ are ordered lists containing events information described in detail in Section 5.3.2. Similarly, the Poincaré state vector, previously defined in (2.4), is extended by the horizontal

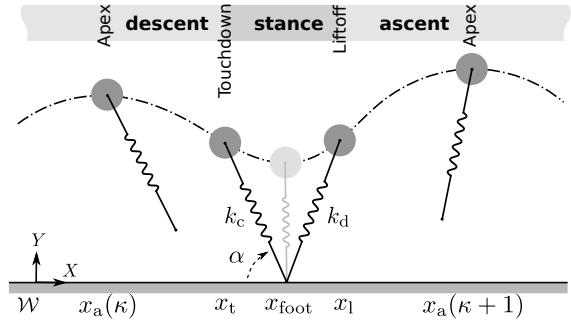


Figure 5.2: Monopedal SLIP model and its phases of motion.

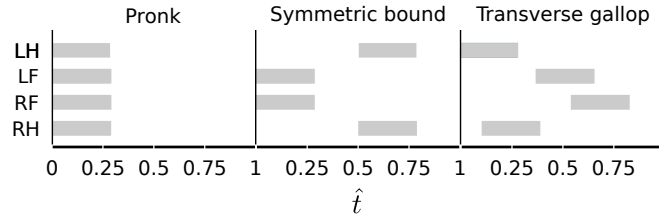


Figure 5.3: Patterns of different quadrupedal running gaits with respect to normalized time. Gray rectangles represent stance and white space represents swing. LH stands for left-hind leg, and so on.

position of the mass at apex, i.e., $z := \begin{bmatrix} x_a & \dot{x}_a & y_a \end{bmatrix}^T$. As will be shown in Section 5.3.1, this allows the capturing of in-stride event timings, which is critically needed for synchronization purposes.

5.2.2 Relevance and feasibility

The proposed dual-SLIP template, depicted in Figure 5.1, is capable of representing a number of running gaits of quadrupeds including: *pronking*, *bounding* and *galloping*, whose temporal patterns are illustrated in Figure 5.3. The relevance of the model can readily be seen for pronking and bounding gaits in which bilateral legs behave seamlessly. For the galloping gait, one can observe that the CoM of the fore and hind bodies move in the same vein as the SLIP; however, feet positioning needs further considerations. Nevertheless, we suspect that our template would be relevant for a galloping gait as well.

The consideration of physically-unconnected SLIPs as a quadrupedal template seems to warrant a flexible torso, a distinguished feature in quadrupedal animals. Several biomechanical studies seek to explain how torso flexibility enhances the locomotory skills of animals [96, 103–105]. When it comes to implementing on a robot, however, only a few quite recent platforms feature a flexible torso [106–109].

Function 5.1 Bounding gait temporal state scheduling

1: **function** Bounding($S(\kappa-1)$)
2: $t_{\mathrm{t,f}}(\kappa) \leftarrow \max(t_{\mathrm{l,f}}(\kappa-1)+\tau_{\mathrm{sw}}, t_{\mathrm{l,h}}(\kappa-1)+\tau_{\Delta})$
3: $t_{\mathrm{l,f}}(\kappa) \leftarrow t_{\mathrm{t,f}}(\kappa)+\tau_{\mathrm{st}}$
4: $t_{\mathrm{t,h}}(\kappa) \leftarrow \max(t_{\mathrm{l,h}}(\kappa-1)+\tau_{\mathrm{sw}}, t_{\mathrm{l,f}}(\kappa)+\tau_{\Delta})$
5: $t_{\mathrm{l,h}}(\kappa) \leftarrow t_{\mathrm{t,h}}(\kappa)+\tau_{\mathrm{st}}$
6: **return** $S(\kappa)$
7: **end function**

Even for a quadruped with rigid torso, the relative position of the legs' CoM varies with respect to different configurations of the whole body, suggesting that the proposed template could still be capable in capturing the dynamics of those robots.

5.2.3 Gait definition

As outlined earlier, we are interested in developing a leg coordination framework that is event-driven, i.e., governed by difference equations rather than differential equations as in Central Pattern Generators (CPGs)-based methods. We carefully reviewed such a framework for modeling of legged locomotion using the max-plus algebra in Chapter 2. The reader is highly recommended to read Section 2.5.4 before continuing here. Following this methodology, here we derive such equations for the bounding ($\mathrm{G_b} = \{1\} \prec \{2\}$) and pronking ($\mathrm{G_P} = \{1,2\}$) gaits of the dual-SLIP model.

Let the scheduling vector S be defined as:

$$S(\kappa) = \begin{bmatrix} t_{\mathrm{t,f}}(\kappa) & t_{\mathrm{t,h}}(\kappa) & t_{\mathrm{l,f}}(\kappa) & t_{\mathrm{l,h}}(\kappa) \end{bmatrix}^T. \tag{5.1}$$

The evolution of these temporal states for a symmetric bounding and pronking gaits of the dual-SLIP model in the traditional algebra are represented in Functions 5.1 and 5.2, respectively. A graphical representation of the bounding gait is similar to what was already given in Figure 2.6b. The associated max-plus representation of these gaits, which is of the form (2.24), can systematically be derived following the procedure explained in detail in Section 2.5.4 (For example, the scheduling equations for the bounding gait will be represented by (2.22), if the role of swing and stance times are swapped). Thus, we assume that the linear system matrices A_0 and A_1 for both gaits are given.

The event schedules produced by Functions 5.1 and 5.2 ensures constant swing durations and adjusts the stance durations for synchronization. Once again, we emphasize that this touchdown constraint is not unique or strictly needed to

Function 5.2 Pronking gait temporal state scheduling

1: **function** Pronking($S(\kappa-1)$)
2: $t_{\mathrm{t,f}}(\kappa) \leftarrow \max(t_{\mathrm{l,f}}(\kappa-1), t_{\mathrm{l,h}}(\kappa-1)) + \tau_{\mathrm{sw}}$
3: $t_{\mathrm{t,h}}(\kappa) \leftarrow \max(t_{\mathrm{l,h}}(\kappa-1), t_{\mathrm{l,f}}(\kappa)) + \tau_{\mathrm{sw}}$
4: $t_{\mathrm{l,f}}(\kappa) \leftarrow t_{\mathrm{t,f}}(\kappa) + \tau_{\mathrm{st}}$
5: $t_{\mathrm{l,h}}(\kappa) \leftarrow t_{\mathrm{t,h}}(\kappa) + \tau_{\mathrm{st}}$
6: **return** $S(\kappa)$
7: **end function**

synchronize legs, as the opposite condition would also be valid.

5.3 Coordination controller

This section presents the coordination controller proposed to induce spatio-temporal coordination between the SLIPs in the dual-SLIP model. It should be emphasized that the method can be applied to compositional treatments of intermittent contact systems of any types, in which the spatial and temporal states can (partly) be controlled at least once per cycle. With respect to the scope of the present chapter, we illustrate the method exclusively for the dual-SLIP model; however, in Appendix A additional examples are provided to demonstrate the effectiveness of the proposed coordination framework in coordinating a variety of intermittent contact systems, such as the spring mass hoppers and juggling balls.

The problem of synchronization in physically-unconnected intermittent contact systems was studied by Klavins and Koditschek in [110]. The authors employed the phase regulation method [111] to construct, in canonical coordinates, reference vector fields that serve as a model for controlling the compositional behaviors. Compared to this research, the method proposed in this chapter differs, most notably, in the following elements: (i) the proposed method manipulates difference equations of max-plus linear systems instead of differential equations of reference vector fields; (ii) the separation of reference generation from the low-level control calculations considered in our method gives more freedom to a control designer to simply expand the network, and adjust the synchronization rhythm without needing to redesign the low-level controllers; and (iii) defining the synchronization rhythm in the max-plus based framework is straightforward and systematic, and there is no need for moving from the physical coordinates to the canonical coordinates of phase and phase velocity.

An overview of the coordination controller is depicted in Figure 5.4. The success of the framework relies on the presence of an effective individual controller, denoted by "C", for the SLIP models. As can be seen, the controllers of the

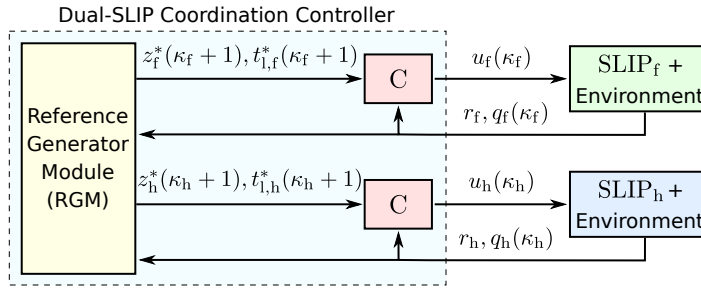


Figure 5.4: Overall structure of the proposed coordination controller.

fore and hind SLIPs are not directly connected to each other. The compositional behavior is coordinated in the Reference Generator Module (RGM), whose basic function is to provide the individual controllers with desirable spatio-temporal references.

The RGM generates two event-driven sets of references for each SLIP separately: the spatial reference, z^*, and the temporal reference, S^*. However, the evolution of spatial and temporal states in the real system are generally not independent, i.e., there could be situations in which the produced references are conflicting. This issue needs to be carefully addressed in the design of the controller so that the overall behavior remains feasible. In what follows, we first describe the structure of the low-level controller and then the underlying mechanism that produces and updates the appropriate references in the RGM.

5.3.1 Individual controller

The majority of the existing control methods for the SLIP model in the literature, including the unified controller we proposed in Chapter 4, focus on the stabilization of Poincaré states in the deadbeat scheme. For the temporal coordination, however, one needs to grant control authority over the event timings. Though, any adjustment on the timings could also affect the Poincaré states and vice versa. In the following, we extend the controller presented in Section 4.2 so as to achieve a *time-aware* deadbeat controller for the SLIP model.

To simplify the problem, let us assume the spatial control specifications are to keep constant the horizontal speed and vertical position, from the κth to the $(\kappa+1)$th apex at the Poincaré section. Following this, one can observe that there is a monotonic relationship between the normalized stance time, $\hat{\tau}_{\mathrm{st}} = \tau_{\mathrm{st}}/(\tau_{\mathrm{st}}+\tau_{\mathrm{sw}})$, and the normalized step length, $\hat{x}_{\mathrm{a}}(\kappa) = (x_{\mathrm{a}}(\kappa) - x_{\mathrm{a}}(\kappa-1))/l_{\mathrm{rest}}$, for different control parameters, as shown in Figure 5.5. Moreover, the resulting in-stride trajectories are symmetric as a consequence of the particular spatial control

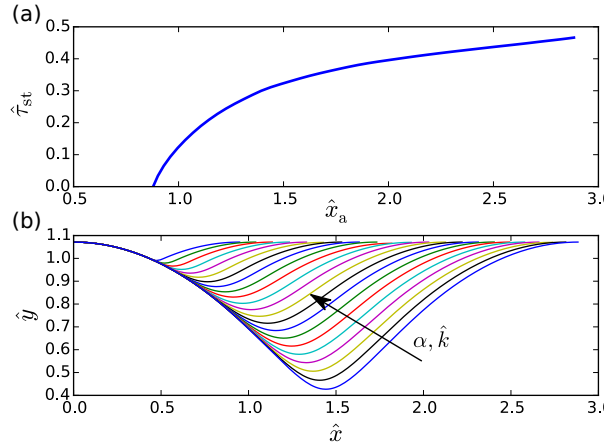


Figure 5.5: (a) Relationship between the normalized stance time, $\hat{\tau}_{\mathrm{st}} = \tau_{\mathrm{st}}/(\tau_{\mathrm{st}} + \tau_{\mathrm{sw}})$, and the normalized step length, $\hat{x}_{\mathrm{a}}(\kappa) = (x_{\mathrm{a}}(\kappa) - x_{\mathrm{a}}(\kappa - 1))/l_{\mathrm{rest}}$, for different control parameters; (b) Corresponding in-stride trajectories with $\hat{y} = y/l_{\mathrm{rest}}$.

specifications posed. As such, the problem of controlling the horizontal position of the system at Poincaré section can easily be translated to the control of liftoff time instant since $\hat{t}_{\mathrm{l}}^{*}(\kappa + 1) = \hat{\tau}_{\mathrm{sw}}/2 + \hat{\tau}_{\mathrm{st}}$. Taking this into consideration, we formulate the control calculations in the following optimization problem:

$$u(\kappa) = \begin{bmatrix} \alpha & k_{\mathrm{c}} & k_{\mathrm{d}} \end{bmatrix}^{T} = \operatorname{argmin} J, \tag{5.2}$$

subject to:

$$\begin{aligned} & 50^{\circ} < \alpha < 90^{\circ}, \\ & k_{\mathrm{c}}, k_{\mathrm{d}} > 0, \\ & 0.9 l_{\mathrm{rest}} \leq y_{\mathrm{a}}(\kappa + 1) \leq 1.25 l_{\mathrm{rest}}, \end{aligned}$$

where J is the cost function:

$$\begin{aligned} J = & w_1 (t_{\mathrm{l}}^{*}(\kappa + 1) - t_{\mathrm{l}}(\kappa + 1))^2 + w_2 (\dot{x}_{\mathrm{a}}^{*}(\kappa + 1) - \\ & \dot{x}_{\mathrm{a}}(\kappa + 1))^2 + w_3 (y_{\mathrm{a}}^{*}(\kappa + 1) - y_{\mathrm{a}}(\kappa + 1))^2. \end{aligned} \tag{5.3}$$

w_i, $i \in \{1, 2, 3\}$, are used to weigh different terms of the cost function. New control parameters are calculated when a new set of references is generated in the RGM, and only if the corresponding SLIP is in the *descent* phase.

To solve the optimization problem (5.2), we have tested different solvers including the `COBYLA` method of NLopt[1], and a modification of Powell hybrid

[1] http://ab-initio.mit.edu/nlopt

method implemented in the `root` function of Scipy.optimize[2]. We eventually found the latter more effective in our particular setting as it allows defining multiple cost functions. To implement constraints, we used common parameter transformation methods and also applied necessary penalties in the cost functions.

The above-presented time-aware controller enables the explicit control of liftoff instant, while still regulating the desired Poincaré states. Compared to the active running control presented in Section 4.2, this additional feature, namely the control of timing, is achieved at the cost of relaxing the constraint (4.1), which reinforced the biological inspiration for bipedal running.

5.3.2 Reference Generator Module (RGM)

The RGM is designed to perform mainly the following functions:

i) updating the schedule for a given cycle index κ;

ii) producing a schedule for the next cycle for each SLIP upon its touchdown event.

In doing so, we implement the max-plus gait scheduling methodology presented in Section 2.5.4. In the rest of this section, we explain why, when and how the RGM performs the above-mentioned functions.

Why does a previously produced schedule for the κth cycle need to be updated? This is due to: (i) the limitations on the realization of the generated references, as discussed in the previous section; and (ii) deviations possibly imposed by external disturbances and system uncertainties. Moreover, according to the definition of gaits in the max-plus algebra (2.24 and 2.29), updating the current schedule substantially influences the $(\kappa+1)$th schedule vector as well, which contains the temporal references needed in the control calculations (see (5.3)).

When does an update occur in the schedule? (i) Upon the occurrence of a touchdown or liftoff event of either of SLIPs in the κth cycle, the corresponding schedule vector, $S(\kappa)$, must be updated. For that, the actual event time instant is stored in the list of *just-occurred* events, $\mathcal{Q}_{\mathrm{j}}(\kappa) \in q(\kappa)$ (see Section 5.2.1); (ii) Moreover, whenever an event (touchdown or liftoff) is expected to occur but has not occurred yet, the schedule needs to be updated as well. In this case, an internal predictor estimates a new time instant for the delayed event, and the new information is stored in the list of *expected-to-occur* events, $\mathcal{Q}_{\mathrm{e}}(\kappa) \in q(\kappa)$. Note that a delayed touchdown event influences $S(\kappa+1)$ instead of $S(\kappa)$, even though the cycle index is not actually incremented in this case.

[2] http://www.scipy.org

Algorithm 5.1 Reference Generator Module (RGM) functioning

1: $\kappa_{\min}$: minimum of the cycle indexes corresponding to just-occurred/expected-to-occur events
2: $\kappa_{\max} = \max(\kappa_{\mathrm{f}}, \kappa_{\mathrm{h}})$
3: $Q(\kappa)$: list of just-occurred/expected-to-occur, and already occurred events and their time instants in the κth cycle
4: **for** $\kappa = \{\kappa_{\min}, \kappa_{\min}+1, \ldots, \kappa_{\max}\}$ **do**
5: **update** $S(\kappa)$ with respect to $Q(\kappa)$ ▷ Algorithm (5.2)
6: **produce** $S(\kappa+1)$ ▷ Equation (2.29)
7: **end for**

Algorithm 5.2 Updating an existing schedule

1: n: number of systems to be synchronized (for the dual-SLIP model $n = 2$)
2: **for** $i = \{1, 2, \ldots, 2n\}$ **do**
3: $S(\kappa) \leftarrow A_0 \otimes S(\kappa) \oplus A_1 \otimes S(\kappa-1)$ ▷ Equation (2.24)
4: **replace** elements of $S(\kappa)$ corresponding to $Q(\kappa)$
5: **if** $S(\kappa) = A_0 \otimes S(\kappa) \oplus A_1 \otimes S(\kappa-1)$ **then**
6: **break**
7: **end if**
8: **end for**
9: **return** $S(\kappa)$

How does the RGM update the schedule? The update procedure is described in Algorithm 5.1. We shall need to elaborate on the difference between updating $S(\kappa)$ (line 5 of the algorithm) and producing $S(\kappa+1)$ (line 6). While for the latter task, it is straightforward to use the explicit representation (2.29), for the former task we utilize the implicit formulation (2.24) in an iterative scheme that is described in Algorithm 5.2. The difference between the tasks lies on the fact that we *update* a schedule (say, $S(\kappa)$) once a just-occurred/expected-to-occur event in the κth cycle is detected, while we *produce* a new schedule for subsequent uses. As such, during the execution of the update Algorithm 5.2, the "known" temporal states are enforced by their corresponding values in $Q(\kappa)$.

Note that the RGM function presented here in Algorithm 5.1 features an improved formulation that is significantly simpler compared with what we initially presented in [26]. The main improvement has been achieved by allowing different systems to live in different cycles, i.e., here κ_{f} and κ_{h} can take different cycle indexes that are not necessarily consecutive. The simulation experiments of multiple spring-mass hoppers presented in Appendix A use the previously developed method. Although this method also performs satisfactory, the advantageous simplifications made here are in light of the thesis objective: the systematic design of control architecture within the multi-layer scheme.

Table 5.1: Simulated system parameters

	symbol	value
Fore SLIP mass	m_{f}	30 kg
Hind SLIP mass	m_{h}	20 kg
Leg rest length	l_{rest}	0.7 m
Fore torso length	l_{f}	0.7 m
Hind torso length	l_{h}	0.4 m

5.4 Simulated gaits and their transitions

In this section various simulation tests are performed to evaluate the capabilities of the proposed template in generating different gaits and feasible transitions and its robustness to external disturbances (in the form of ground elevation)[3]. The numerical values of the parameters associated with the dual-SLIP model used for the simulation are listed in Table 5.1. The simulation was carried out using a hybrid solver that we have developed in Python 2.7. The solver is equipped with a variable step size integrator that captures the phase switchings precisely. Due to variable step size integrations, the solver must also ensure that the time evolution remains the same between the fore and hind SLIPs. The reader is referred to the attached video for a full demonstration of the simulated gaits and their transitions[4].

5.4.1 Robust quadrupedal bounding

Several metrics are proposed in the literature to evaluate the disturbance rejection properties of a controlled legged robot (see [112] and references therein). In our setting, the dual-SLIP template is assigned to converge to a bounding gait, while accommodating two variations in the ground height equal to %25 of the leg rest length. Figure 5.6 (top) shows the trajectories of the SLIPs together with the snapshots corresponding to the touchdown moments of either of legs. The respective horizontal and vertical ground reaction forces, normalized by the corresponding weights, are plotted in the middle and bottom panels of the figure. Moreover, the vertical positions of the SLIPs and the torso joint angle, γ, with respect to time are depicted in Figure 5.7. The readers may find a more complete illustration of the results in the video attachment.

As can be seen, the spatio-temporal coordination controller proposed in this chapter enforces the nominal gait quickly after the disturbance (and after the

[3] See Appendix A for additional examples provided for different intermittent contact systems.
[4] The video is available online also in the following link: https://youtu.be/9GMy6xKXCJs

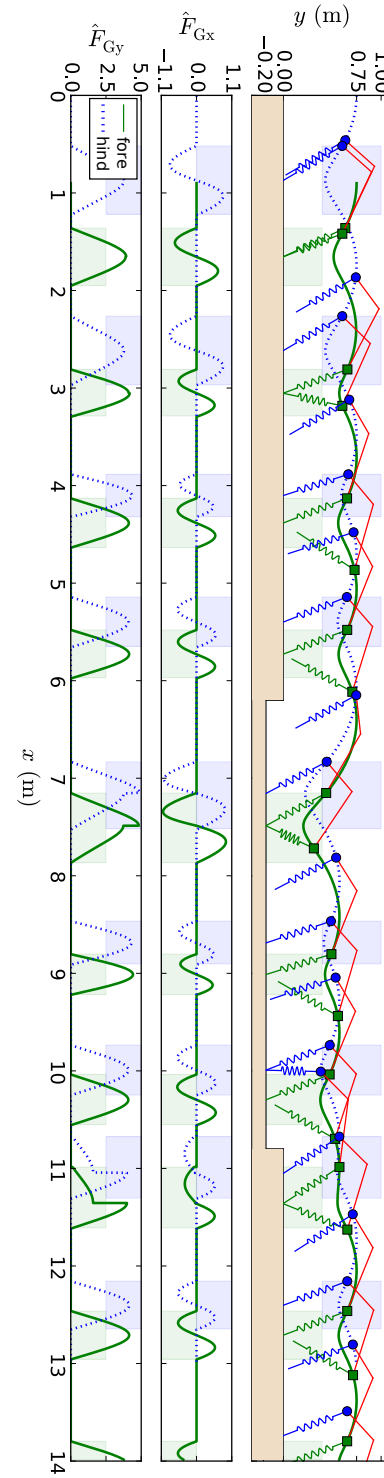


Figure 5.6: Robustness evaluation of a quadrupedal bounding realized in the dual-SLIP model. Colored rectangles represent the corresponding stance phases and white space represents swing.

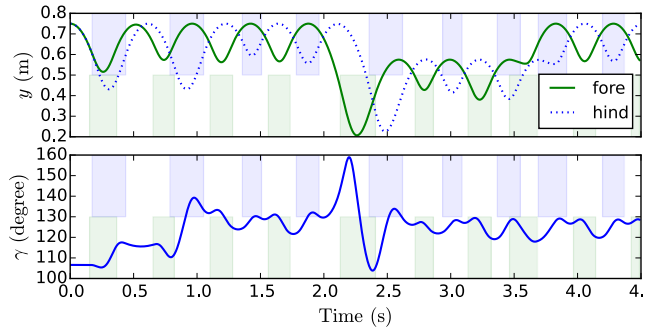


Figure 5.7: Vertical positions of the SLIPs (top) and the torso angle (bottom) during the ground height variation test. Colored rectangles represent the corresponding stance phases and white space represents swing.

sufficiently off-phase initial condition). Our simulation tests show that remarkably larger step-down disturbances can also be handled by the controller. As shown in Figure 5.7 (bottom), γ remains far from possibly critical value $\gamma = 180^{\circ}$, ensuring that the distance between the SLIPs, d_{hf}, will not exceed $(l_{\mathrm{f}} + l_{\mathrm{h}})$.

To accommodate the ground height variations, we predict an estimated touchdown time instant and adjust the reference liftoff time instant accordingly. Thanks to this adjustment, the ground reaction forces do not experience large variations, as can be seen in Figure 5.6 (middle) and (bottom). However, an overly large adjustment requires larger leg flexions that can be infeasible in terms of realizing simulation results in real robotic hardware.

5.4.2 Transition from pronking to bounding

As discussed in Section 2.5.4, the problem of choosing gait transitioning parameters in the max-plus algebra is as straightforward as *switching* between different gait matrices. The switching max-plus linear system representation (2.30), which is repeated here for convenience, is a useful tool for systematic generation of transition gaits:

$$\begin{aligned} S(\kappa) &= A_0(\mu(\kappa)) \otimes S(\kappa) \oplus A_1(\mu(\kappa)) \otimes S(\kappa - 1), \\ S(\kappa + 1) &= A(\mu(\kappa + 1)) \otimes S(\kappa), \end{aligned} \tag{5.4}$$

in which $\mu(\kappa)$ is a "switching" integer function whose value designates a certain gait.

In a simulation test, we now study the transition between the pronking and bounding gaits (with the same τ_{st} and τ_{sw}) in our dual-SLIP template. By replacing the linear matrices of the pronking gait $\mathrm{G_p} = \{1, 2\}$ with those of bounding gait

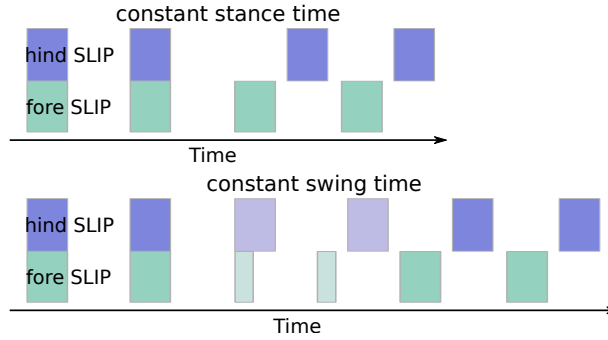


Figure 5.8: Temporal references generated for the transitioning from pronking to bounding with constant stance time (top) and constant swing time (bottom). Colored rectangles represent the corresponding stance phases and white space represents swing.

$G_p = \{1\} \prec \{2\}$ directly, the simplest transition gait can be considered, whose generated temporal references can be schematically depicted in Figure 5.8 (top). As can be seen, the stance time during the transition remains unchanged, which inevitably requires longer swing time. On the other hand, similar to what is presented in Section VI-B of [61], by introducing some auxiliary gaits (specifically, one or two gaits), the stance time of either of SLIPs during the transition can independently be shrunk so to achieve a unique swing time for all legs. The generated references following this method are depicted in Figure 5.8 (bottom).

Additionally, depending on the order of legs in the bounding gait ($\{1\} \prec \{2\}$ or $\{2\} \prec \{1\}$), either of the above-mentioned transitions can be realized in two different ways. Accordingly, different transient behaviors will emerge during transition, although the steady-state behavior would be identical. Overall, four different cases for the transitioning from pronking to bounding can be considered:

i) pronking $\{1,2\} \rightarrow$ bounding $\{1\} \prec \{2\}$ & τ_{st} constant,

ii) pronking $\{1,2\} \rightarrow$ bounding $\{1\} \prec \{2\}$ & τ_{sw} constant,

iii) pronking $\{1,2\} \rightarrow$ bounding $\{2\} \prec \{1\}$ & τ_{st} constant,

iv) pronking $\{1,2\} \rightarrow$ bounding $\{2\} \prec \{1\}$ & τ_{sw} constant.

Figure 5.9 shows the simulated transitions for cases (i) (top) and (ii) (bottom)[5]. As can be seen, the duration of transitioning in case (i) is longer than case (ii). Also, notice that in case (i) the hind leg compresses significantly more than case (ii); however, the maximum magnitude of the ground reaction force in case (ii) is slightly larger than case (i).

[5] Since the figure shows only the vertical positions, the results corresponding to cases (iii) and (iv) would be similar to cases (i) and (ii), respectively, provided that the role of fore and hind legs are swapped.

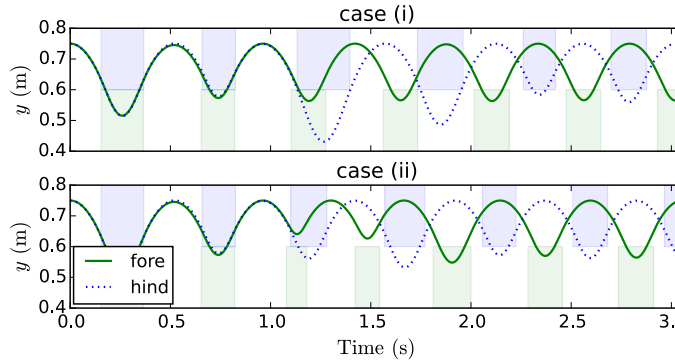


Figure 5.9: Simulated transitions from pronking to bounding in the dual-SLIP model. Colored rectangles represent the corresponding stance phases and white space represents swing.

Of particular interest is the way the coordination framework treats "idealistic" plans. Comparing the preplanned reference schedules (Figure 5.8) with those realized during simulation (Figure 5.9), one can observe that the schedules are automatically updated in accordance with the feasible behaviors designated by the real system. In other words, in case the low-level controllers cannot fulfill the assigned control specifications, this information is fed back to the RGM and a new schedule is produced subsequently.

A remarkable difference among the four possible transitions can be seen in terms of normalized relative distance, $\hat{d}_{\mathrm{hf}} = d_{\mathrm{hf}}/(l_{\mathrm{f}} + l_{\mathrm{h}})$, as shown in Figure 5.10. Notice that for the transition case (i) d_{hf} exceeds $(l_{\mathrm{f}} + l_{\mathrm{h}})$ for a small duration of time. However, this may not necessarily mean that the transition is infeasible, as arguments like this should be justified when anchoring the dual-SLIP model in the real robot. Nevertheless, it would be worth exploring whether it is possible to control $\hat{d}_{\mathrm{hf}}$ in the current implementation.

Our investigations show that by considering a torsional spring at the *virtual* torso joint (see Figure 5.1), and translating the resulting torque effect to both SLIP models, a desired $\hat{d}_{\mathrm{hf}}$ profile can be enforced, As an example, Figure 5.11 shows the result of applying a torsional spring with different stiffnesses to keep the desired d_{hf} in bounding the same as that of pronking. While this simple solution sufficiently addresses our particular problem, more sophisticated approaches can also be followed. For instance, similar to [101], an LQR controller can be used to actively enforce a varying desired torso angle being sufficiently compatible with the natural dynamics of the system. Determining such desired torso angle profile is straightforward as the steady-state behavior of the dual-SLIP system is known a priori.

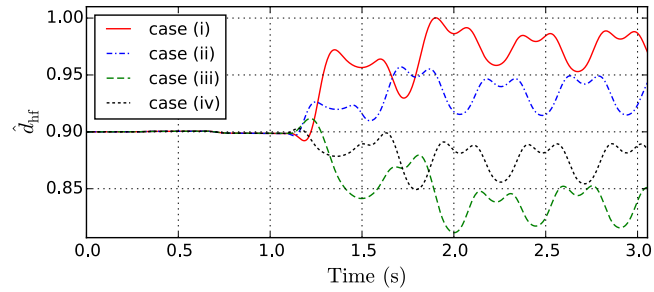


Figure 5.10: Normalized relative distance, $\hat{d}_{\mathrm{hf}} = d_{\mathrm{hf}}/(l_{\mathrm{f}} + l_{\mathrm{h}})$, with respect to time for different cases of transition.

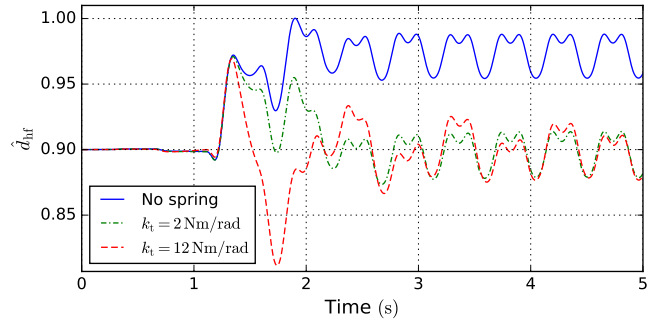


Figure 5.11: The effect of applying a torsional spring with constant k_{t} at the virtual torso joint on the normalized distance $\hat{d}_{\mathrm{hf}}$ in transition case (i).

5.4.3 Remark on multi-body quadrupedal running

As shown in this chapter, the study of quadrupedal running using the dual-SLIP model provides insights to the dominant features of the dynamics, using which we were able to generate different quadrupedal gaits and gait transitions systematically. The resultant motions can easily be used in the control of multi-body quadruped models with telescopic legs. However, the application of the results in the control of quadrupeds with articulated legs would not be as simple. One approach to this problem is to follow the feedback linearization-based method, similar to what we presented for bipedal walking in Section 4.7. Although it is possible, this approach presents its own disadvantages due to the many design parameters that need to be tuned in order to make sure that the relevant template is anchored in the real robot.

One alternative is to use inverse kinematics to map the dual-SLIP states onto the multi-body joint states that can later be sent as references to the standard

inverse dynamics controllers, i.e.,

$$\zeta^*, \dot{\zeta}^* = \text{Inverse Kinematics}(s_i, q_i),\ i \in \{\text{f}, \text{h}\}, \tag{5.5}$$

where, ζ and $\dot{\zeta}$ are the vector of generalized position and velocity of the multi-body model in joint space, and the marker "$*$" denotes the reference value. Here, the same idea of using physically-unconnected SLIPs in the proposed template can be exploited to solve the inverse kinematics problem analytically, hence offering a computationally fast and reliable mapping between the states of the two systems. To do so, considering the real robot model and its relation to the dual-SLIP template as depicted in Figure 5.1, the fore and hind parts of the robot form simple planar three limb mechanisms, for which the closed-form solutions of the inverse kinematics are available. Once the map is constructed, one can define the reference joint accelerations as

$$\ddot{\zeta}^* = k_\text{p}(\zeta^* - \zeta) + k_\text{d}(\dot{\zeta}^* - \dot{\zeta}), \tag{5.6}$$

with k_p and k_d being some positive definite matrices. The generated joint space references $\zeta^*, \dot{\zeta}^*$ and $\ddot{\zeta}^*$ are constraint-consistent desired motions that can readily be used in standard inverse dynamics control methods, such as the recently developed one particularly suitable for floating base (legged) morphologies [113].

5.5 Conclusion

In this chapter a novel template for quadrupedal steady and transitional running has been proposed. The model represents a quadruped by means of two physically-unconnected SLIP models, coordinated in a way that the overall behavior simulates different quadrupedal gaits and their transitions. Since the proper coordination between the individual SLIP models of the proposed template plays the major role in the success of the control system, a new (generic) coordination controller for synchronization of (multiple) template models was developed. At the core of the controller is a reference generator module that is formulated in the max-plus algebra. The significance of this formulation is in its linear form which facilitates the generation of the reference events that are used in the individual controllers.

Theoretically, the method can be applied to compositional treatments of intermittent contact systems of any types, in which the spatial and temporal states can (partly) be controlled at least once per cycle. In particular, we implemented the method on the dual-SLIP model to realize the transition from pronking to bounding gaits and to achieve robust reactions against ground height variations. Different

transition gaits were developed systematically. It was shown that although the corresponding steady-state behaviors before and after transitions are the same, the transient effects significantly vary from one transition gait to another. Since developed systematically, the template model admitted the inclusion of a torsional spring at the virtual torso to give more control on the transient behaviors.

The results of this chapter constitute a first step towards achieving relatively complex locomotion patterns in quadrupedal running, by properly composing individual SLIP templates in a synchronized fashion. Thanks to the underlying coordination framework, the composed template is simple to analyze, flexible with respect to control specifications, and potentially suitable for online implementation.

CHAPTER

6

Conclusions

In this thesis we have developed a template-based hierarchical control architecture for generation of dynamic locomotion in legged robots. This chapter summarizes the main contributions of the thesis and proposes some recommendations for possible future research.

6.1 Conclusions

We started by establishing a framework for modeling and control of legged locomotion on the SLIP template. A hybrid automaton was introduced that formally describes the sequential composition of the primary phases of motion for the SLIP running, walking, and for the transitions between these modes.

Towards planning and control of such dynamic gaits online, analytic approximations of the otherwise non-integrable double-stance dynamics of the dissipative SLIP model were addressed. We have followed two different approaches, one focusing on the inclusion of an explicit effect of damping, while the other is intended to yield a closed-form solution. At the core of both methods is the approximation of the original bipedal SLIP trajectories by those of an auxiliary system referred to as the AT-SLIP. The required relationships between the parameters of the two

systems were derived by applying simple approximations.

A comprehensive numerical investigation covering highly non-symmetric trajectories with considerable amounts of damping was conducted for both methods to assess their predictive power. We have achieved almost the same accuracy for the proposed double-stance maps compared to the existing solutions to the single-stance dynamics, although the coupling effects are severer in our case. Furthermore, for the especial case of lossless SLIP model (with no damping), we presented two distinct methods which yield almost the same predictive performance, verifying the feasibility of the approaches taken.

Subsequently, we presented control applications exploiting the predictive power of the proposed methods which, thanks to the simplicity resulted from the analytical nature of the solutions, are suitable for online implementations. By assuming a piecewise constant profile for the stiffness of the bipedal SLIP, together with the modulation of the touchdown angle, one can synthesize unified controllers that create varying speed gaits and their transitions. Choosing different stiffness for the spring compression and decompression phases grants the required changes needed in the system's total energy. In addition, the leg touchdown angle can modulate the interchange between kinetic and potential energy, which is necessary for locomotion speed regulation.

The stability/robustness of the resulting controlled system was analyzed numerically. The negative effect of approximations on the robustness of the controller was compensated by subjecting the controllable touchdown angle to a certain lower bound. In other words, the proposed approximations in this thesis are more effective for the human-like walking gaits (i.e, for relatively large touchdown angles). For the deadbeat-form controllers, we also expanded the existing results on the stability analysis so as to ensure that a controller utilizing the proposed approximations can stably achieve a desired behavior.

We also conducted a number of simulation tests verifying the practicality of the synthesized controllers in producing human-like walking and running. Control designs explicitly meant for evaluation of the predictive power of the approximations showed significant promise even in the presence of non-negligible damping forces. Also studied was an embedding of the controlled SLIP on a more realistic walking model, following a feedback linearization-based approach, which presented a complete coordination between the states of the two systems. This result offers a feasible dynamic motion planning strategy for online purposes.

To extend the functionality of the method to the context of quadrupedal running, we also proposed a new coordination controller for synchronization of multiple template models. This was based on the observation that a single SLIP

represents a body with a point mass, thereby it cannot be solely useful in capturing the pitch dynamics of the body. At the core of the controller is a reference generator module that is formulated in the max-plus algebra framework. The significance of this formulation is in its linear form which facilitates the generation of the reference events that are used in the individual controllers. In this regard, we also expanded previous results by accounting for the natural dynamics of the system in the generation of the reference events as well as the control calculations. Theoretically, the method can be applied to compositional treatments of intermittent contact systems of any types, in which the spatial and temporal states can (partly) be controlled at least once per cycle. We tested this in simulation for the coordination of a number of well-known template models, including the SLIP model.

Accordingly, a novel template for quadrupedal steady and transitional running, called the dual-SLIP model, was proposed. The model represents a quadruped by means of two physically-unconnected SLIP models, composed in a way that the overall behavior simulates different quadrupedal gaits and their transitions, using the coordination controller developed. The relevance and feasibility of the template in representing different running gaits of a quadruped were discussed. Finally, the transition from pronking to bounding gaits and robustness features of the system against ground height variations were explored.

The following summarizes the main conclusions of this thesis:

- The proposed analytical approximations for the DS dynamics of the SLIP model are useful tools in motion planning and control of 2D bipedal robot walking, due to the following reasons:
  - They are evaluated very fast compared to numerical integration;
  - They enable the possibility of computing closed-form representations for the associated Jacobian matrix;
  - The approximation errors remain small for a reasonable operational space of the SLIP walking, as was empirically shown in this thesis.
- In the context of deadbeat control for SLIP walking, the exact pinpointing of a desired equilibrium is a strict and mostly unnecessary requirement imposed to the control system. In this thesis it was shown that, by leveraging on the self-stability property of the SLIP model, a wide class of controllers can stabilize the desired equilibrium.
- It was shown that when the information of the controlled SLIP walking is used in the motion planning of the studied five-DoF planar robot model, the resulting torque profiles are comparatively smooth. However, the distribution

of physical parameters of the two systems should carefully be matched to avoid the demand of possibly large torques.

- This thesis provided a first step towards achieving relatively complex locomotion patterns, e.g., quadrupedal running, by properly composing individual SLIP templates in a synchronized fashion. Thanks to the underlying coordination framework, the composed template is simple to analyze, flexible with respect to control specifications, and potentially suitable for online implementation.

6.2 Recommendations for future research

The multi-layer and hierarchical nature of the presented control architecture provides a straightforward procedure for possible extensions of the method's functionalities. For example, we suspect that the introduced dual-SLIP model can further be useful in generation of different quadrupedal running gaits that are not discussed in this thesis such as galloping. For such an extension, most elements of the control structure should remain the same, and only the process of anchoring will need further considerations.

The analytical perspective taken in this study opens avenues for tailoring the existing models in understanding humans and animals gaits in biomechanics, and for enhancing online planning and control methods for legged locomotion with non-instantaneous double-stance phases in robotics. We suspect it might be possible to follow a similar approach in different disciplines to tackle issues that are due to a lack of analytical representations.

In what follows we present some unexplored and open problems of the present thesis along with possible directions for future research.

- The proposed DS map was developed for the SLIP walking in 2D. The method can intuitively be extended to model the 3D SLIP walking, which has recently attracted significant interest in the field of dynamic walking. This might be accomplished by incorporating an additional torsional spring in the plane perpendicular to sagittal plane.

- Despite its importance, the SLIP walking on inclined/rough terrain, with the two feet being at different heights, has not received adequate attention in the relevant literature. The extension of the introduced DS map to handle those situations is an interesting subject for future research. In doing so, the process of producing the equivalent AT-SLIP model needs to be revisited.

Specially, the determination of the zero-torque angle of the torsional spring and the virtual toe position may need particular attention.

- The transition equations to/from our DS approximations were solved numerically in this thesis. Additional effort can be put to find a sufficiently accurate closed-form solution for these discrete equations. However, the influence of any further simplifications on the overall performance should be carefully analyzed.

- The introduced approximations following the PB approach are only applicable for the lossless SLIP. Since energy dissipation is inevitable in real platforms, this makes the direct application of the method difficult. Extending the method to incorporate energy dissipation via a damping element will offer a straightforward representation for the DS dynamics.

- In this thesis, the proposed active SLIP walking gait was embedded in the multi-body (realistic) robot model, while the transition strategies between walking and running gaits were only studied on the SLIP model. Also, the produced quadrupedal running gaits on the dual-SLIP model can be applied on a relevant quadrupedal robot model. To this end, recently developed inverse dynamics control of floating base systems [113] can be used to realize the dual-SLIP-like motions in the real robot model. This will let different gaits of a quadruped to evolve systematically.

- Stability analysis of the compositional behaviors of multiple intermittent contact systems in the max-plus based coordination controller was not performed. Although we provided empirical evidence showing that for the reasonable operational space of the studied systems the composition controller is stable, a formal study of such statement is generally an open problem in the relevant literature.

- Concerning the embedding of the SLIP-like motions in real robots, we suspect that a co-design of the robot's morphology and control based on the SLIP counterpart can substantially improve the control performance in terms of the actuation and speed limits. This is a very interesting topic for future research.

Bibliography

Bibliography

[1] M. H. Raibert *et al.*, *Legged robots that balance*. MIT press Cambridge, MA, 1986, vol. 3.

[2] H. Miura and I. Shimoyama, "Dynamic walk of a biped," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 60–74, 1984.

[3] J. Furusho and M. Masubuchi, "A theoretically motivated reduced order model for the control of dynamic biped locomotion," *Journal of dynamic systems, measurement, and control*, vol. 109, no. 2, pp. 155–163, 1987.

[4] S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. Van De Panne, "Locomotion skills for simulated quadrupeds," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 59.

[5] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3D-SLIP model," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5134–5140.

[6] I. Mordatch, M. De Lasa, and A. Hertzmann, "Robust physics-based locomotion using low-dimensional planning," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 71, 2010.

[7] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.

[8] M. Vukobratović and B. Borovac, "Zero-moment point–thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[9] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[10] A. D. Ames, E. A. Cousineau, and M. J. Powell, "Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics," in *Proceedings of the 15th ACM International conference on Hybrid Systems: Computation and Control*. ACM, 2012, pp. 135–144.

[11] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Proceedings of 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 200–207.

[12] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[13] M. H. Dickinson, C. T. Farley, R. J. Full, M. Koehl, R. Kram, and S. Lehman, "How animals move: an integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, 2000.

[14] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.

[15] D. Koepl and J. Hurst, "Force control for planar spring-mass running," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 3758–3763.

[16] U. Saranli and D. E. Koditschek, "Template based control of hexapedal running," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2003, pp. 1374–1379.

[17] G. Garofalo, C. Ott, and A. Albu-Schaffer, "Walking control of fully actuated robots based on the bipedal SLIP model," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1456–1463.

[18] M. M. Ankaralı and U. Saranlı, "Control of underactuated planar pronking through an embedded spring-mass Hopper template," *Autonomous Robots*, vol. 30, no. 2, pp. 217–231, 2011.

[19] M. Millard, E. Kubica, and J. McPhee, "Forward dynamic human gait simulation using a SLIP target model," *Procedia IUTAM*, vol. 2, pp. 142–157, 2011.

[20] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[21] M. Shahbazi, R. Babuška, and G. A. D. Lopes, "Unified Modeling and Control of Walking and Running on the Spring-Loaded Inverted Pendulum," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1178–1195, October 2016.

[22] M. Shahbazi, G. Lopes, and R. Babuška, "Automated Transitions Between Walking and Running in Legged Robots." in *Proceedings of the 19th IFAC World Congress*, vol. 19, no. 1, August 2014, pp. 2171–2176.

[23] M. Shahbazi, U. Saranlı, R. Babuška, and G. A. Lopes, "Approximate analytical solutions to the double-stance dynamics of the lossy Spring-Loaded Inverted Pendulum," *Bioinspiration & Biomimetics*, vol. 12, no. 1, p. 016003, 2016.

[24] M. Shahbazi, R. Babuška, and G. Lopes, "Analytical Approximation for the Double-stance Phase of a Walking Robot," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5754–5760.

[25] M. Shahbazi and G. Lopes, "Coordination of Monopedal SLIP Models Towards Quadrupedal Running," in *Proceedings of IEEE Conference on Intelligent Robots and Systems (IROS)*, October 2016, pp. 5440–5445.

[26] M. Shahbazi and G. A. D. Lopes, "A Max-Plus Based Synchronization Controller for Multiple Spring-Mass Hoppers," in *Proceedings of 2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, December 2015, pp. 1669–1674.

[27] G. Cavagna, F. Saibene, and R. Margaria, "Mechanical work in running," *Journal of Applied Physiology*, vol. 19, no. 2, pp. 249–256, 1964.

[28] R. Alexander and A. Jayes, "Vertical movements in walking and running," *Journal of Zoology*, vol. 185, no. 1, pp. 27–40, 1978.

[29] R. Blickhan, "The spring-mass model for running and hopping," *Journal of biomechanics*, vol. 22, no. 11, pp. 1217–1227, 1989.

[30] T. A. McMahon and G. C. Cheng, "The mechanics of running: how does stiffness couple with speed?" *Journal of Biomechanics*, vol. 23, pp. 65–78, 1990.

[31] R. Altendorfer, U. Saranli, H. Komsuoglu, D. Koditschek, H. B. Brown Jr, M. Buehler, N. Moore, D. McMordie, and R. Full, *Evidence for spring loaded inverted pendulum running in a hexapod robot.* Springer, 2001.

[32] M. Ahmadi and M. Buehler, "Controlled passive dynamic running experiments with the ARL-monopod II," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 974–986, 2006.

[33] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[34] R. Playter, M. Buehler, and M. Raibert, "BigDog," in *Pro. of SPIE 6230, Unmanned Systems Technology VIII*, 2006, pp. 62 302O–62 302O.

[35] J. A. Grimes and J. W. Hurst, "The design of ATRIAS 1.0 a unique monoped, hopping robot," in *Proceedings of the 2012 International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2012, pp. 548–554.

[36] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[37] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[38] J. Rummel, Y. Blum, and A. Seyfarth, "Robust and efficient walking with spring-like legs," *Bioinspiration & biomimetics*, vol. 5, no. 4, p. 046004, 2010.

[39] L. Visser, S. Stramigioli, and R. Carloni, "Robust bipedal walking with variable leg stiffness," in *Proceedings of 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob).* IEEE, 2012, pp. 1626–1631.

[40] J. Ketelaar, L. C. Visser, S. Stramigioli, and R. Carloni, "Controller design for a bipedal walking robot using variable stiffness actuators," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 5650–5655.

[41] Y. Liu, P. Wensing, D. Orin, and Y. Zheng, "Dynamic Walking in a Humanoid Robot Based on a 3D Actuated Dual-SLIP Model," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5710–5717.

[42] P. Holmes, "Poincaré, celestial mechanics, dynamical-systems theory and chaos," *Physics Reports*, vol. 193, no. 3, pp. 137–163, 1990.

[43] W. J. Schwind and D. E. Koditschek, "Approximating the stance map of a 2-DOF monoped runner," *Journal of Nonlinear Science*, vol. 10, no. 5, pp. 533–568, 2000.

[44] R. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, "A simply stabilized running model," *SIAM Journal on Applied Dynamical Systems*, vol. 2, no. 2, pp. 187–218, 2003.

[45] H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: simple approximate solution and application to gait stability," *Journal of theoretical biology*, vol. 232, no. 3, pp. 315–328, 2005.

[46] J. J. Robilliard and A. M. Wilson, "Prediction of kinetics and kinematics of running animals using an analytical approximation to the planar spring-mass system," *Journal of experimental biology*, vol. 208, no. 23, pp. 4377–4389, 2005.

[47] H. Yu, M. Li, P. Wang, and H. Cai, "Approximate Perturbation Stance Map of the SLIP Runner and Application to Locomotion Control," *Journal of Bionic Engineering*, vol. 9, no. 4, pp. 411–422, 2012.

[48] U. Saranlı, Ö. Arslan, M. M. Ankaralı, and Ö. Morgül, "Approximate analytic solutions to non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping," *Nonlinear Dynamics*, vol. 62, no. 4, pp. 729–742, 2010.

[49] S. N. Rasband, *Chaotic dynamics of nonlinear systems*. Courier Dover Publications, 2015.

[50] W. J. Schwind, "Spring loaded inverted pendulum running: a plant model," Ph.D. dissertation, Electrical Enginnering: Systems, University of Michigan, 1998.

[51] J. Rummel, Y. Blum, and A. Seyfarth, "From walking to running," in *Autonome Mobile Systeme 2009*. Springer, 2009, pp. 89–96.

[52] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical computer science*, vol. 138, no. 1, pp. 3–34, 1995.

[53] E. T. Whittaker, *A treatise on the analytical dynamics of particles and rigid bodies: with an introduction to the problem of three bodies*. Cambridge University Press Archive, 1970.

[54] A. Seyfarth and H. Geyer, "Natural control of spring-like running–optimized self-stabilization," in *Proceedings of the Fifth International Conference on Climbing and Walking Robots. Professional Engineering Publishing Limited*, 2002, pp. 81–85.

[55] Y. Blum, J. Rummel, and A. Seyfarth, "Advanced swing leg control for stable locomotion," in *Autonome Mobile Systeme 2007*. Springer, 2007, pp. 301–307.

[56] A. Seyfarth, H. Geyer, and H. Herr, "Swing-leg retraction: a simple control model for stable running," *Journal of Experimental Biology*, vol. 206, no. 15, pp. 2547–2555, 2003.

[57] O. Arslan, U. Saranli, and O. Morgul, "An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2388–2393.

[58] I. Uyanik, O. Morgul, and U. Saranli, "Experimental Validation of a Feed-Forward Predictor for the Spring-Loaded Inverted Pendulum Template," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 208–216, Feb 2015.

[59] M. M. Ankarali and U. Saranli, "Stride-to-stride energy regulation for robust self-stability of a torque-actuated dissipative spring-mass hopper," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, pp. 033 121–033 121, 2010.

[60] O. Arslan, U. Saranli, and O. Morgul, "Reactive footstep planning for a planar spring mass hopper," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2009, pp. 160–166.

[61] G. Lopes, B. Kersbergen, T. van den Boom, B. De Schutter, and R. Babuska, "Modeling and Control of Legged Locomotion via Switching Max-Plus Models," *IEEE Trans. on Robotics*, vol. 30, no. 3, pp. 652–665, 2014.

[62] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[63] P. Holmes, R. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: models, analyses, and challenges," vol. 48, no. 2, pp. 207–304, 2006.

[64] S. Grillner, "Control of locomotion in bipeds, tetrapods, and fish," vol. Supplement 2: Handbook of Physiology, The Nervous System, Motor Control, pp. 1179–1236, 2011.

[65] J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1981.

[66] B. Heidergott, G. Olsder, and J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems.* Kluwer, 2006.

[67] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, *Synchronization and Linearity: an Algebra for Discrete Event Systems.* Wiley, 1992.

[68] B. Giffler, "Mathematical Solution of Production Planning and Scheduling Problems," IBM ASDD, Tech. Rep., Oct. 1960.

[69] R. Cuninghame-Green, "Describing Industrial Processes with Interference and Approximating Their Steady-State Behaviour," *J. of the Operational Research Society*, vol. 13, no. 1, pp. 95–100, 1962.

[70] R. A. Cuninghame-Green, *Minimax Algebra.* Springer-Verlag, 1979, vol. 166.

[71] G. Lopes, T. Van den Boom, B. De Schutter, and R. Babuška, "Modeling and control of legged locomotion via switching max-plus systems," in *Proc. of the Int. Workshop on Discrete Event Systems*, 2010, pp. 392–397.

[72] G. Piovan and K. Byl, "Approximation and Control of the SLIP Model Dynamics via Partial Feedback Linearization and Two-Element Leg Actuation Strategy," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 399–412, 2016.

[73] M. Shahbazi, "Bipedal Spring-Loaded Inverted Pendulum (SLIP) model and an associated approximate return map in Python," 2015.

[74] J. Rummel, Y. Blum, H. M. Maus, C. Rode, and A. Seyfarth, "Stable and robust walking with compliant legs," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 5250–5255.

[75] H. R. Martinez and J. P. Carbajal, "From walking to running a natural transition in the SLIP model using the hopping gait," in *Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2011, pp. 2163–2168.

[76] S. G. Carver, N. J. Cowan, and J. M. Guckenheimer, "Lateral stability of the spring-mass hopper suggests a two-step control strategy for running," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 2, p. 026106, 2009.

[77] G. Perdikaris, *Computer Controlled Systems: Theory and Applications*, ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer, 1991.

[78] I. Poulakakis and J. Grizzle, "Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper," in *Proceedings of the European Control Conference*, 2007, pp. 1–8.

[79] S. H. Tamaddoni, F. Jafari, A. Meghdari, and S. Sohrabpour, "Biped hopping control bazsed on spring loaded inverted pendulum model," *International Journal of Humanoid Robotics*, vol. 7, no. 02, pp. 263–280, 2010.

[80] A. Arampatzis, G.-P. Brüggemann, and V. Metzler, "The effect of speed on leg stiffness and joint kinetics in human running," *Journal of biomechanics*, vol. 32, no. 12, pp. 1349–1353, 1999.

[81] K. T. Kalveram, D. F. Haeufle, A. Seyfarth, and S. Grimmer, "Energy management that generates terrain following versus apex-preserving hopping in man and machine," *Biological cybernetics*, vol. 106, no. 1, pp. 1–13, 2012.

[82] J. E. Pratt, "Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots," DTIC Document, Tech. Rep., 2000.

[83] D. G. Hobbelen and M. Wisse, "Controlling the walking speed in limit cycle walking," *The International Journal of Robotics Research*, vol. 27, no. 9, pp. 989–1005, 2008.

[84] J. K. Holm and M. W. Spong, "Kinetic energy shaping for gait regulation of underactuated bipeds," in *Proceedings of IEEE International Conference on Control Applications (CCA)*. IEEE, 2008, pp. 1232–1238.

[85] Y. Huang, B. Vanderborght, R. Van Ham, Q. Wang, M. Van Damme, G. Xie, and D. Lefeber, "Step length and velocity control of a dynamic bipedal walking robot with adaptable compliant joints," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 598–611, 2013.

[86] T. Haarnoja, J.-L. Peralta-Cabezas, and A. Halme, "Model-based velocity control for Limit Cycle Walking," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 2255–2260.

[87] B. I. Prilutsky and R. J. Gregor, "Swing-and support-related muscle actions differentially trigger human walk–run and run–walk transitions," *Journal of Experimental Biology*, vol. 204, no. 13, pp. 2277–2287, 2001.

[88] A. Hreljac, R. T. Imamura, R. F. Escamilla, and W. B. Edwards, "When does a gait transition occur during human locomotion," *Journal of Sports Science and Medicine*, vol. 6, no. 1, pp. 36–43, 2007.

[89] J. K. Hodgins, "Biped gait transitions," in *Proc. of IEEE Int. Conference on Robotics and Automation (ICRA)*. IEEE, 1991, pp. 2092–2097.

[90] C. T. Farley and O. Gonzalez, "Leg stiffness and stride frequency in human running," *J. of biomechanics*, vol. 29, no. 2, pp. 181–186, 1996.

[91] V. Segers, M. Lenoir, P. Aerts, and D. De Clercq, "Kinematics of the transition between walking and running when gradually changing speed," *Gait & posture*, vol. 26, no. 3, pp. 349–361, 2007.

[92] V. Segers, P. Aerts, M. Lenoir, and D. De Clercq, "Spatiotemporal characteristics of the walk-to-run and run-to-walk transition when gradually changing speed," *Gait & posture*, vol. 24, no. 2, pp. 247–254, 2006.

[93] H. R. Vejdani, A. Wu, H. Geyer, and J. W. Hurst, "Touch-down angle control for spring-mass walking," in *Proc. of IEEE Int. Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5101–5106.

[94] R. Blickhan and R. Full, "Similarity in multilegged locomotion: bouncing like a monopode," *Journal of Comparative Physiology A*, vol. 173, no. 5, pp. 509–517, 1993.

[95] P. Nanua and K. Waldron, "Energy comparison between trot, bound, and gallop using a simple model," *Journal of biomechanical engineering*, vol. 117, no. 4, pp. 466–473, 1995.

[96] K. Leeser, "Locomotion Experiments on a Planar Quadruped Robot with Articulated Spine, Master's thesis," 1996.

[97] I. Poulakakis, E. Papadopoulos, and M. Buehler, "On the stable passive dynamics of quadrupedal running," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2003, pp. 1368–1373.

[98] Z. G. Zhang, Y. Fukuoka, and H. Kimura, "Stable quadrupedal running based spring-loaded two-segment legged on a model," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2004, pp. 2601–2606.

[99] U. Culha and U. Saranli, "Quadrupedal bounding with an actuated spinal joint," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 1392–1397.

[100] A. K. Valenzuela and S. Kim, "Optimally scaled hip-force planning: A control approach for quadrupedal running," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1901–1907.

[101] Q. Cao and I. Poulakakis, "Quadrupedal bounding with a segmented flexible torso: passive stability and feedback control," *Bioinspiration & biomimetics*, vol. 8, no. 4, p. 046007, 2013.

[102] S. Pouya, M. Khodabakhsh, A. Spröwitz, and A. Ijspeert, "Spinal joint compliance and actuation in a simulated bounding quadruped robot," *Autonomous Robots*, pp. 1–16, 2015.

[103] M. Hildebrand, "Further studies on locomotion of the cheetah," *Journal of mammalogy*, vol. 42, no. 1, pp. 84–91, 1961.

[104] R. M. Alexander, *Elastic mechanisms in animal movement*. Cambridge University Press, 1988.

[105] R. Hackert, N. Schilling, and M. S. Fischer, "Mechanical self-stabilization, a working hypothesis for the study of the evolution of body proportions in terrestrial mammals?" *Comptes Rendus Palevol*, vol. 5, no. 3, pp. 541–549, 2006.

[106] J. L. Pusey, J. M. Duperret, G. C. Haynes, R. Knopf, and D. E. Koditschek, "Free-standing leaping experiments with a power-autonomous elastic-spined quadruped," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 87 410W–87 410W.

[107] S. Kim, "Cheetah-inspired quadruped: high-speed locomotion platform, 2013."

[108] M. Khoramshahi, A. Sprowitz, A. Tuleu, M. N. Ahmadabadi, and A. Ijspeert, "Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3329–3334.

[109] Q. Zhao, K. Nakajima, H. Sumioka, X. Yu, and R. Pfeifer, "Embodiment enables the spinal engine in quadruped robot locomotion," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 2449–2456.

[110] E. Klavins and D. Koditschek, "Phase regulation of decentralized cyclic robotic systems," *The Int. J. of Robotics Research*, vol. 21, no. 3, pp. 257–275, 2002.

[111] E. Klavins, D. Koditschek, and R. Christ, "Toward the regulation and composition of cyclic behaviors," in *Algorithmic and computational robotics: new directions*. A K Peters, LTD, 2001, pp. 205–219.

[112] J. D. Karssen and M. Wisse, "Running with improved disturbance rejection by using non-linear leg springs," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1585–1595, 2011.

[113] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 3406–3412.

Appendix

Additional Examples for the Coordination Controller Presented in Chapter 5

This appendix provides additional examples for the coordination controller presented in Chapter 5. The synchronization of a number of well-known intermittent contact systems in robotics is studied: spring-mass hoppers, monopedal SLIPs, and juggling balls.

A.1 Spring-mass hoppers

Consider a point mass m affected by gravity (with the gravitation acceleration denoted by g) with vertical position coordinate y, on top of a linear spring with spring constant k and rest length y_{rest} (see Figure A.1). Two main phases of motion are possible: flight (when $y > y_{\text{rest}}$, i.e., the entire system is airborne), and stance (when $y \leq y_{\text{rest}}$, i.e., the system is in contact with the ground). We further divide the flight phase into the ascending and descending subphases, and the stance phase into the spring compression and decompression. The transition between phases and the corresponding events are illustrated in Figure A.1. By defining the system state vector as $x = \begin{bmatrix} y & \dot{y} \end{bmatrix}^T$, then $x_{\text{a}}, x_{\text{t}}, x_{\text{b}}$ and x_{l} denote the

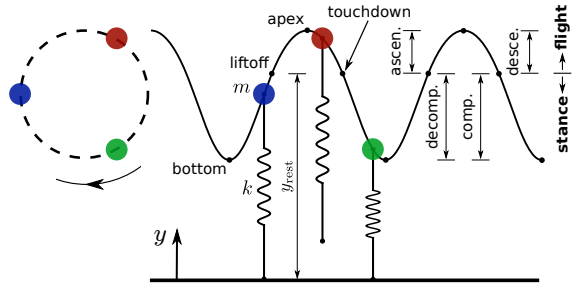


Figure A.1: Multiple spring-mass hoppers are synced in an alternating rhythm.

hopping state at apex, touchdown, bottom and liftoff points, respectively.

The equation of motion in the flight phase is:

$$m\ddot{y} = -mg, \tag{A.1}$$

and its time-domain solution is:

$$y(t) = -1/2gt^2 + \dot{y}_0 t + y_0, \tag{A.2}$$

where y_0 and $\dot{y}_0$ denotes the hopping height and velocity at the time instant $t = 0$, respectively. The equation of motion in the stance phase is:

$$m\ddot{y} = -mg + k(y_{\text{rest}} - y), \tag{A.3}$$

with the corresponding time-domain solution:

$$y(t) = M\cos(\omega t + \phi) + F, \tag{A.4}$$

where $\omega^2 = k/m, F = y_{\text{rest}} - g/\omega^2$. Similarly, by denoting the hopping height and velocity at the time instant $t = 0$ by y_0 and $\dot{y}_0$, we derive $M = \sqrt{(y_0 - F)^2 + (\dot{y}_0/\omega)^2}$ and $\phi = \arctan(\frac{-\dot{y}_0/\omega}{y_0 - F})$.

Simulation Results

In this section, we present a simulation experiment to induce synchronized behaviors in multiple physically-unconnected hoppers. As noted in Chapter 5, for this particular simulation example, we used a preliminary version of the developed coordination controller. Please refer to [26] for more details about the coordination method and the individual low-level controller.

Figure A.2 illustrates the simulation of five hoppers with different masses, started from arbitrarily chosen initial hopping heights at apex. The synchronization

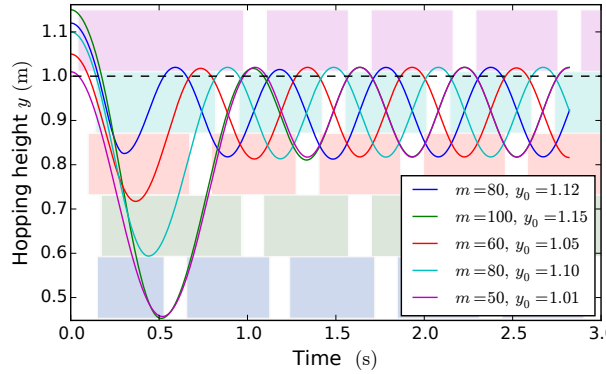


Figure A.2: A simulation depicting the synchronization of five hoppers with different initial conditions and masses ($y_{\text{rest}} = 1\,\text{m}$ and $x_{\text{ref}} = 1.02\,\text{m}$). The masses are in kilogram.

rhythm is characterized by $\text{G} = \{1\} \prec \{3\} \prec \{4\} \prec \{2, 5\}$, which requires that the hoppers bounce in an alternating fashion where the hoppers 2 and 5 are to be synced in-phase. The solid lines depict the hopping heights, colored rectangles represent corresponding stance, and white space represents flight.

Figure A.3 shows the simulation results of two hoppers that are to be synced out-of-phase, under external disturbances. The considered disturbance is a constant upward force with the magnitude of $0.3mg$, applied in stance phases that are indicated by the yellow rectangles (from $t = 1\,\text{s}$ to $t = 1.5\,\text{s}$ on hopper 1 and from $t = 2.5\,\text{s}$ to $t = 3\,\text{s}$ on hopper 2). In addition to the colored rectangles and white space, here we plot the schedule for the corresponding undisturbed hoppers in dotted rectangles. The control inputs, namely the stiffness of the hoppers are depicted as well.

Discussion

Characteristics of the proposed method: As can be seen in Figure A.2, following the method proposed, it is feasible to synchronize multiple cyclic hybrid systems (five spring-mass hoppers in this example) according to any arbitrarily chosen synchronization rhythm, given in the form of (2.26). For any given gait G, the developed software systematically computes the system matrices A_0, A_1 and A, that are then used in the presented max-plus based scheduler to produce the temporal reference states for each hopper. Then, the independent individual controllers computes the control inputs that stabilize the hoppers at the desired hopping height and phase relationship.

It should be emphasized that, in addition to the flexibility feature discussed above, the developed method is also significantly efficient in terms of computational

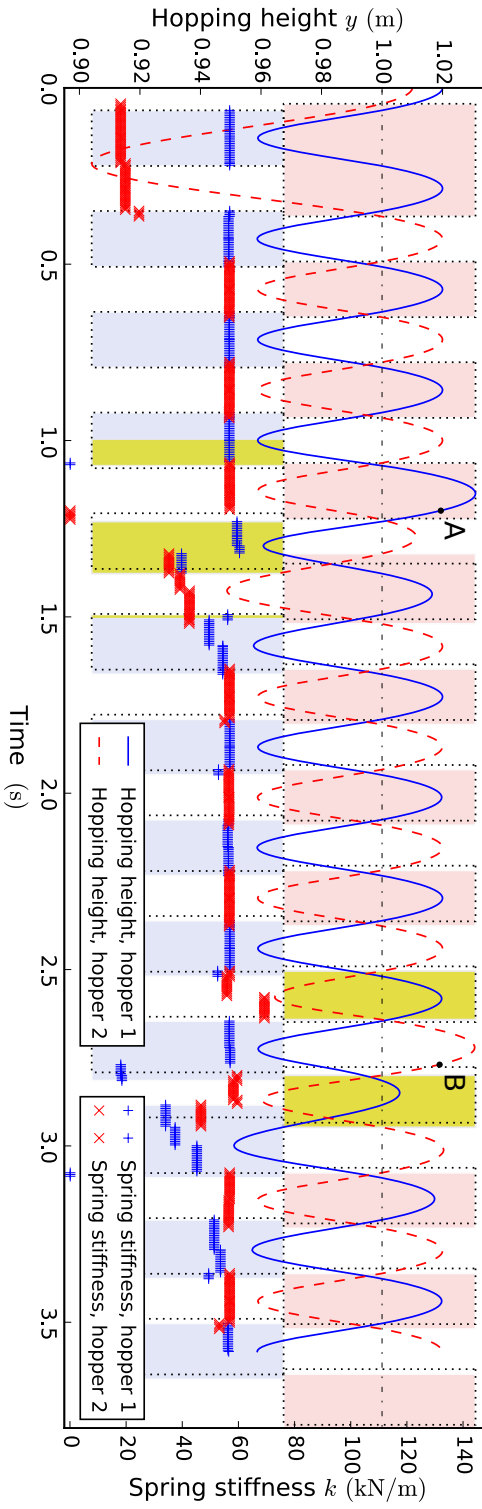


Figure A.3: Two hoppers starting from different initial conditions synchronized while external disturbances are imposed to either of hoppers in stance phases depicted in yellow rectangles ($m = 80\,\mathrm{kg}$, $y_{\mathrm{rest}} = 1\,\mathrm{m}$ and $x_{\mathrm{ref}} = 1.02\,\mathrm{m}$).

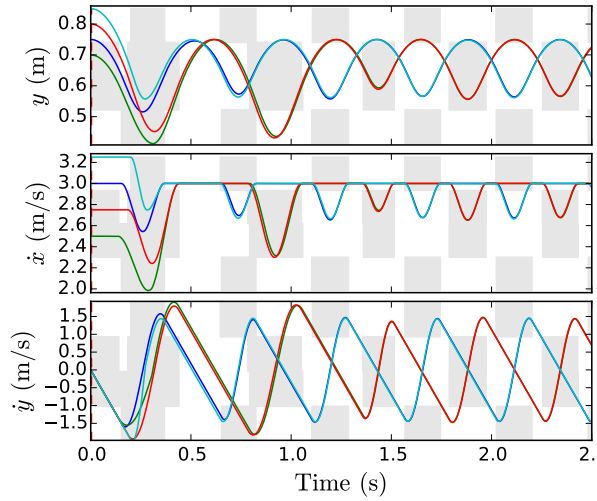


Figure A.4: Coordination of four SLIP models starting from arbitrarily chosen initial conditions and with respect to the gait $G = \{1, 4\} \prec \{2, 3\}$

effort. The schedule generation and control input calculations are event-driven, and the hoppers are provided by the stiffnesses that follow a piecewise constant profile. This means that in most of the execution time the hoppers behave according to their natural dynamics. Moreover, thanks to the linearity of the equations describing the scheduler in the max-plus algebra, increasing the number of hoppers imposes minimal additional computations needed.

Disturbance rejection: As shown in Figure A.3, the system of hoppers quickly stabilize after the disturbances have ended. Although there is no connecting element between the two controllers, the undisturbed hopper also reacts to the disturbance imposed to the other hopper. This reaction is dictated through the new reference liftoff instants provided by the schedule generator. In particular, when the touchdown event of the disturbed hopper has been delayed (see point A and B in Figure A.3), the schedule generator assigns a longer reference liftoff instant for the undisturbed hopper. Consequently, its controller computes a smaller stiffness that leads to a longer stance duration, as can be seen in the figure.

A.2 Monopedal SLIPs

The spatio-temporal coordination of two SLIPs forming the dual-SLIP template was studied in Chapter 5. To show that the controller is capable of coordinating more than two SLIP models, in Figure A.4 we present the coordination of four SLIP models starting from arbitrarily chosen initial conditions. The coordination

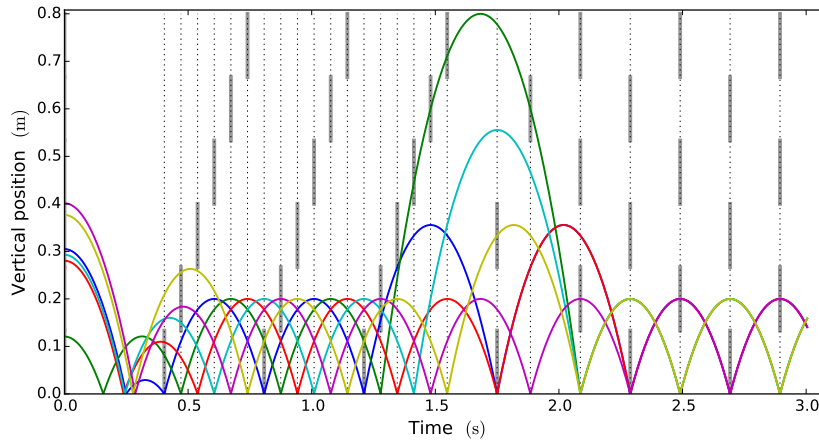


Figure A.5: Six juggling balls starting from arbitrarily chosen initial conditions are synced and transitioned according to: $\mathrm{G}_1 = \{1\} \prec \{2\} \prec \{3\} \prec \{4\} \prec \{5\} \prec \{6\} \rightarrow \mathrm{G}_2 = \{1, 3, 5\} \prec \{2, 4, 6\}$. Ground hits are depicted by gray bars.

rhythm $G = \{1, 4\} \prec \{2, 3\}$ simulates the quadrupedal running trot, the gait that requires the presence of double-swing phase. The time-aware deadbeat controller of the SLIPs is executed when: (i) the corresponding SLIP is in the descent phase; and (ii) an event (apex, touchdown or liftoff) for any of the SLIP models occurs/delays. Notice that the execution rate of the individual controllers is different from the schedule update rate, which is upon a touchdown or liftoff event.

A.3 Juggling balls

Likewise Klavins and Koditschek [110], we also applied our method to a system of juggling balls, as depicted in Figure A.5. Here we also introduced the gait switching by using the switching max-plus linear system (5.4). It is clear that for a juggling ball we only have a swing (flight) phase. As such, the duration of stance phase τ_{st} must be set to zero in the corresponding derivations. For the individual law-level controllers, similar to [110], we used the well-known mirror law with small modifications. Finally, it has to be noted that in this example we did not apply the updating mechanism of the RGM, i.e., the reference schedule was produced a priori.

Samenvatting

De grote drijvende kracht achter de ontwikkeling van looprobots is de mogelijke superioriteit tijdens het opereren in ongestructureerde/rommelige omgevingen, gemotiveerd door de prestaties van lopende dieren. Alhoewel een flink aantal looprobots zijn ontwikkeld en er daarom significante verbeteringen zijn behaald, is er nog veel te doen om vergelijkbare flexibiliteit, wendbaarheid en snelheid te halen als dieren op het land en mensen. Dit geeft overduidelijk de noodzaak aan voor verder onderzoek naar zowel zaken gerelateerd aan het lichaam, zoals sensoren, actuatoren en morfologiën, als zaken geraleerd aan het brein, te weten waarneming en regeling.

Meerdere opdrachteseisen zoals het genereren van een looppatroon, het plannen van stappen, en dynamisch consistente diepe structuren, dagen het succes uit van een loop-regelsysteem op verschillende niveaus. De belangrijkste problemen komen naar voren vanuit de intrinsieke instabiliteit van het rechtop staan, en van de hybride aard van een loopmechanisme. Een effectieve implementatie van bewegingsplanning van zulke robots vereist substantiële aandacht tijdens de technische ontwerpfase, alsook de vereiste rekenkracht, in het bijzonder in de aanwezigheid van onzekerheid en een oneffen terrein. Om deze reden wordt een groot deel van het planningsalgoritme traditioneel buiten operatie om uitgevoerd, waardoor de regelaar taak-specifiek wordt en gevoelig voor problemen met de robuustheid.

In deze thesis wordt een sjabloon-gebaseerde, hiërarchische regeltechnische architectuur ontwikkeld voor het genereren van dynamische loopbewegingen met verbeterde autonomie en robuustheid. We presenteren een gelaagd raamwerk waarbinnen verschillende dynamische manieren van lopen en de transities ertussen systematisch behandeld worden, geïnspireerd door het gevestigde idee van sjabloon en anker, een op biologische systemen geïnspireerde abstractie/specificering van

het probleem van bewegingsplanning en-regeling. Met deze aanpak wordt een essentieel onderdeel van het bewegingsplanningsalgoritme ingegeven door het (vereenvoudigde) sjabloonmodel, terwijl de resulterende bewegingen later vastgelegd (verankerd) worden in de echte robot. Wanneer dit op de juiste manier ontworpen wordt, maakt de sjabloonlaag het mogelijk dat de uitvoering van het grootste deel van de planningsberekeningen tijdens de uitvoering worden gedaan, waardoor de autonomie en robuustheid van de robot worden verbeterd.

In de sjabloonlaag gebruiken we een standaard massa-veermodel, ook wel een veerbekrachtigd geïnverteerd dat al langer gebruikt wordt als beschrijvend model van dynamische loopbewegingen. Om de mogelijkheid van online implementatie te verbeteren van de planningsalgoritmes in ons gelaagd raamwerk, introduceren we de eerste analytische benadering van de normaal gesproken niet-integreerbare dubbele-stand-dynamica van het SLIP-model. Wij stelden twee verschillende manieren van benadering voor: De eén heeft de focus op het meenemen van het expliciete effect van energieverlies; de ander is bedoeld om een expliciete oplossing te verkrijgen in het geval van afwezigheid van energieverlies. De kern van beide methoden bestaat uit de benadering van de originele tweebenige SLIP-modeltrajecen via de trajecten van een hulpsysteem dat we axiaal-torsie SLIP (AT-SLIP) noemen. Om de voorspellende kracht van de voorgestelde oplossingen vast te stellen zijn uitgebreide numerieke onderzoeken gedaan van situaties met sterk asymmetrische trajecten met aanzienlijke hoeveelheden dempking. De nauwkeurigheid van de voorgestelde dubbele-stand afbeeldingen is bijna dezelfde als die van bestaande oplossingen voor de enkele-stand dynamica in de literatuur, alhoewel de niet-lineaire koppelingseffecten erger waren in ons geval.

Vervolgens wordt het eenvoudige analytische karakter van de afgeleide oplossingen gebruikt in het ontwerpen en analyseren van dynamische manieren van lopen die daadwerkelijk geïmplementeerd kunnen worden. Een klasse van samengevoegde regelaars wordt gesynthetiseerd die SLIP bewegingen voor lopen en rennen met variabele snelheid creëren, waarvan de onderliggende berekeningen voortkomen uit de afgeleide analytische afbeeldingen. Een aantal simulatietesten zijn uitgevoerd om de uitvoerbaarheid te verifiëren van de gesynthetiseerde regelaars bij het produceren van mensachtige bewegingen en het gebruik van zulke resultaten bij het plannen van beweging vaan een meerdelig robotmodel met twee benen. Het is aangetoond dat wanneer de informatie van het SLIP-lopen wordt gebruikt in het regelen van een robotmodel met meerdere vrijheidsgraden, dat de resulterende torsieprofielen in vergelijking geleidelijk zijn, en een haalbare dynamische bewegingsplanstrategie zijn voor in-operatie doelen.

De stabiliteit/robuustheid van het SLIP-dynamisch lopen is numeriek geanaly-

seerd. Het is aangetoond dat als de regelbare aanrakingshoek wordt onderworpen aan een bepaalde ondergrens, die mensachtig lopen simuleert, dat dan het negatieve effect van benaderingen op de stabiliteit van het geregelde systeem geheel kan worden gecompenseerd. We hebben ook laten zien dat de strikte vereiste van het exact vastleggen van een gegeven gewenst evenwicht in de context van zo-snel-mogelijk-regelende regelaars, kan worden versoepeld door het gebruik van de zelf-evenwicht-eigenschap van het SLIP-model.

Een duaal SLIP-model is voorgesteld om de functionaliteit van het gepresenteerde regeltechnische raamwerk uit te breiden naar de context van vierbenig-bewegingsplanning, een nieuw sjabloon voor vierbenig gestadig en transitie-rennen. Het model representeert een vierbenige robot door het combineren van twee SLIP modellen, fysiek niet gekoppeld, op zo een manier dat het gehele gedrag verschillende vierbenige loopbewegingen en hun transities simuleert. Omdat de juiste coördinatie tussen individuele SLIP-modellen van het voorgestelde sjabloon een belangrijke rol speelt in het succes van een regelsysteem, is een nieuwe (generieke) regelaar ontwikkeld voor het synchroniseren van (meerdere) sjabloonmodellen. Aan de basis van de regelaar ligt een referentiegeneratormodule die geformuleerd is in max-plus algebra. De significantie van deze formulering ligt in zijn lineaire vorm, die het genereren van de referentiegebeurtenissen voor de individuele regelaars faciliteert.

Theoretisch gezien kan de methode toegepast worden op compositorische behandelingen van vrijwel-periodieke contactsystemen van elk type, waarin de spatiële en temporele toestand tenminste één keer per cyclus (deels) kan worden gecontroleerd. Om precies te zijn, we hebben de methode geïmplementeerd op het duaal-SLIP-model om de transitie te realiseren van stuiterende naar springende loopmanieren en om robuuste reacties te behalen op variaties in de grondhoogte. De resultaten vormen een eerste stap in de richting van het behalen van vrij complexe looppatronen, bijvoorbeeld het rennen van vierbenige robots, door op de juiste manier individuele SLIP-sjablonen samen te voegen op een gesynchroniseerde manier. Door toedoen van het onderliggen coördinatieraamwerk is het samengestelde sjabloon eenvoudig te analyseren, flexibel met betrekking tot de regelspecificaties, en mogelijk geschikt voor een in-gebruik implementatie.

Summary

The main driving force behind the development of legged robots is their potential superiority in operating in unstructured/cluttered environments, motivated by legged animals performance. Although quite a number of legged robots have been developed and hence significant improvements have been achieved, there is still much to be done to match the flexibility, agility, and speed of many land animals and humans. This clearly shows the need for further research on both *body* related aspects such as sensors, actuators, and morphologies and *brain* related problems, i.e., perception and control.

Several task requirements such as gait pattern generation, footstep planning, and dynamically consistent low-level structures challenge the success of a locomotion control system at different levels. The major difficulties arise from the intrinsic instability of an upright posture and from the hybrid nature of a legged mechanism. An effective implementation of motion planning of such robots demands substantial considerations at the technical development stage as well as computational efforts needed, specially in the presence of uncertainties and terrain irregularities. As such, the major part of planning algorithms is traditionally applied offline, possibly making the controller task-specific and susceptible to robustness issues.

This thesis develops a template-based hierarchical control architecture for generation of dynamic legged locomotion with improved autonomy and robustness. We present a multi-layer framework in which different dynamic gaits and their transitions are systematically addressed inspired by the well-established notion of *template* and *anchor*, a bio-inspired abstraction/refinement approach to the problem of motion planning and control. Following this approach, a critical part of motion planning algorithms is applied in the template (i.e., simplified) model, while the resulting motions are later anchored in the real robot. When properly devised, the template layer enables the execution of the majority of planning computations

online, hence improving the autonomy and robustness of the controlled robot.

In the template layer, we used the standard spring-mass model, also referred to as the spring-loaded inverted pendulum (SLIP), which has long been a descriptive model of dynamic legged locomotion. In order to improve the online implementability of the planning algorithms in our multi-layer framework, we introduced the first analytical approximations to the otherwise non-integrable double-stance dynamics of the SLIP model. We proposed two different methods of approximation, one focusing on the inclusion of an explicit effect of energy dissipation, while the other is intended to yield a closed-form solution in the lossless case. At the core of both methods is the approximation of the original bipedal SLIP trajectories by those of an auxiliary system referred to as the Axial-Torsional SLIP (AT-SLIP). To assess the predictive power of the proposed solutions, a comprehensive numerical investigation covering highly non-symmetric trajectories with considerable amounts of damping was conducted. The accuracy of the proposed double-stance maps is almost the same as that of the existing solutions to the single-stance dynamics in the literature, although the nonlinear coupling effects were severer in our case.

Next, the analytical simplicity of the derived solutions is utilized in the design and analysis of dynamic gaits suitable for online implementation. A class of unified controllers that create varying speed SLIP walking and running and the transitions between them are synthesized, whose underlying calculations are adopted from the derived analytical maps. A number of simulation tests were conducted to verify the practicality of the synthesized controllers in producing human-like motions and using such results in motion planning of a multi-body bipedal robot model. It was shown that when the information of the SLIP walking is used in the control of the higher dimensional robot model, the resulting torque profiles are comparatively smooth, offering a feasible dynamic motion planning strategy for online purposes.

The stability/robustness of the SLIP dynamic walking was analyzed numerically. It was shown that by subjecting the controllable touchdown angle to a certain lower bound, which simulates human-like walking, the negative effect of approximations on the stability of the controlled system can be fully compensated. We also demonstrated that the strict requirement of exact pinpointing of a given desired equilibrium in the context of deadbeat control can be relaxed by leveraging on the self-stability property of the SLIP model.

Towards extending the functionality of the presented control architecture to the context of quadrupedal motion planning, a novel template for quadrupedal steady and transitional running, called the dual-SLIP model, was proposed. The model represents a quadruped by means of two physically-unconnected SLIP models, composed in a way that the overall behavior simulates different quadrupedal

gaits and their transitions. Since the proper coordination between the individual SLIP models of the proposed template plays the major role in the success of the control system, a new (generic) coordination controller for synchronization of (multiple) template models was developed. At the core of the controller is a reference generator module that is formulated in the max-plus algebra. The significance of this formulation is in its linear form which facilitates the generation of the reference events that are used in the individual controllers.

Theoretically, the method can be applied to compositional treatments of intermittent contact systems of any types, in which the spatial and temporal states can (partly) be controlled at least once per cycle. In particular, we implemented the method on the dual-SLIP model to realize the transition from pronking to bounding gaits and to achieve robust reactions against ground height variations. The results constitute a first step towards achieving relatively complex locomotion patterns, e.g., quadrupedal running, by properly composing individual SLIP templates in a synchronized fashion. Thanks to the underlying coordination framework, the composed template is simple to analyze, flexible with respect to control specifications, and potentially suitable for online implementation.

Curriculum vitae

Mohammad Shahbazi Aghbelagh was born on 16 October 1984 in Tehran, Iran. He received the BSc and MSc degrees in mechanical engineering from K. N. Toosi University of Technology, Tehran, Iran, in 2007 and 2010, respectively. He did his MSc thesis titled Design, Control, and Implementation of an Upper-Limb Rehabilitation System Using a Haptic Interface under the supervision of Dr. Ali Nahvi and Dr. Farid Najafi. From 2010 to 2011 he was active in lecturing mechanical engineering and control theory courses to undergraduate students in Islamic Azad University, Iran.

In March 2012, Mohammad joined the Delft Center for Systems and Control (DCSC) at Delft University of Technology as a PhD candidate. His research objective was to develop a systematic approach to the problem of dynamic locomotion in legged robots. Mohammad obtained the DISC certificate for fulfilling the graduate course program of the Dutch Institute of Systems and Control. He also received the certificate of the TU Delft Graduate School for fulfilling the Doctoral Education Program. During the period of PhD study, he co-supervised the graduation assignment of two Msc students in the field of balance control and variable stiffness actuation.

Mohammad's research interests include nonlinear control problems for hybrid systems, robotics, and event-driven control systems.

Email: mohammad.shahbazi63@gmail.com