USING CONTROLLABILITY OF SIMPLE MODELS TO GENERATE MAXIMALLY ROBUST WALKING-ROBOT CONTROLLERS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Petr Victorovich Zaytsev

January 2015

USING CONTROLLABILITY OF SIMPLE MODELS TO GENERATE MAXIMALLY ROBUST WALKING-ROBOT CONTROLLERS

Petr Victorovich Zaytsev, Ph.D.

Cornell University 2015

We study the ability of bipedal walking robots both to avoid falling down and to reach a specific goal, such as standing still or moving along a desired trajectory. The main questions are: When — in what states and with what controls — is it possible for the robot to avoid a fall or to reach a given goal? When reaching the goal is possible, how fast (in how many steps) can it be done? For various meanings of the word 'optimal', what are optimal ways of getting to the goal?

Our primary approach uses two simple 2D models of walking: the Inverted Pendulum (IP) and Linear Inverted Pendulum (LIP). Both models have a point mass at the hip and two massless legs: rigid legs in the IP, extensible in the LIP model. The hip of the LIP is constrained to move at a fixed height above the ground. Both models have two controls per step: step length and impulsive push-off force along the stance leg just before heel-strike in the IP, and step length and stepping time in the LIP model. For each model we numerically compute the *extended n-step viable regions*: all combinations of states and controls that let the robot take at least n steps and not fall. We also compute the *extended n-step controllable regions*: all states and controls that let the robot reach a given steady-state-walking goal by taking n steps or fewer.

Using these models we justify two claims: (i) In most cases and for most bipeds, when the biped can reach a given goal at all, it can do so within two steps. This result is consistent with some human walking-balance and visual-

guidance experiments. (ii) The ability to reach a specific steady-state target is almost equivalent to being able to avoid totally falling down. The controllable regions of the IP model also help us understand the trade-offs between different controllers. Such trade-offs are in robustness, convergence to the nominal motion, and energy efficiency. Based on things learned, we design a walking controller for a real robot Cornell Ranger, which is nearly maximally robust, near optimal in some regards, and is simple in structure.

BIOGRAPHICAL SKETCH

Petr Zaytsev was born on 24 September 1986 in Moscow, Russia. In 2003, Petr was accepted for the Specialist program of the Mechanics & Mathematics Departments at Moscow State University. He received a Specialist degree in Mechanics in 2008. The same year, Petr enrolled in the PhD program in Theoretical and Applied Mechanics at Cornell University. Since August 2008, Petr is a student at the Biorobotics and Locomotion Lab at Cornell University, researching legged robots.

To my parents

Nataliya M. Philippova and Victor A. Zaytsev

ACKNOWLEDGEMENTS

I would like to thank my advisor Andy Ruina for his advice and enthusiasm that helped me make progress in this thesis. Many of the ideas in the thesis originated in his head, while I worked out the details. I also wish to thank John Guckenheimer and Adrian Lewis for serving on my committee.

I would like to thank members of the Ruina Lab, both past and present, for their ideas, suggestions, and other help that contributed to the thesis: Pranav Bhounsule, Atif Chaudhry, Jason Cortell, Anoop Grewal, Javad Hasaneini, Boris Kogan, Matthew Kelly, Matthew Sheen, Feng Shuai, Gregg Stiesberg, Wouter Wolfslag. I am thankful to Cindy Twardokus and Marcia Sawyer for their help with administrative work.

I would like to thank my fellow lab-mates Pranav, Atif, Jason, Anoop, Javad, Boris, Matt, $\widehat{\text{Matt}}$, Ben, Gregg, Wouter, Tim, Ellen, Ruru, Lipeng, Zhentao, Kang, Dunwen, and all my friends in Ithaca for their company and support. I am highly grateful to Nithin Michael, Wacek Godycki, Irina Gaynanova, Evan Ballowe, Maxim Sheinin, and Anuttama Mohan for their help towards the end of my PhD. I enjoyed the time with the Mystery Machine members, with Olivia and her friends.

My research work was supported by National Science Foundation, McMullen Scholarship, and teaching assistantships from the Cornell Department of Mathematics.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Humans and most bipedal robots are naturally unstable in walking. They fall down, unless actively and appropriately controlled. A desirable feature for a robot is that it reliably avoids falling in as many practical situations as possible. Two interesting control approaches that address fall avoidance are Pratt's Capture Regions [64] and Aubin's Viability Theory [5]. Pratt observes that, if a biped can come to a stop, then it can avoid falling. Hence, a controller should be designed such as to ensure the ability of the robot to quickly come to a stop at any time. The Viability Theory is an area of mathematics that studies evolution of dynamical systems with given constraints to the system's state. The central question in Viability Theory is to find all initial states from where the system can evolve indefinitely without violating the constraints. Here we develop a controller design approach for walking robots which is closely related to viability theory and capture region ideas.

1.1 Past control approaches

Before we describe viability theory and capture region concepts — the true starting point of this thesis — in more detail, we describe some other approaches to control of walking.

1.1.1 Passive dynamics

A walking robot with no motors and no sensors can be mechanically designed so as to stably walk down a gentle slope, powered only by gravity [51, 20, 19]. Such *passive walkers* possess periodic trajectories, where the energy loss at each collision is exactly compensated by the work done on the robot by gravity over one step [27]. The first passive-dynamic robot to have a major impact on robotics was McGeer's 'four-legged biped' [51]. McGeer's walker has two pairs of rigidly connected legs with knees and is conceptually a 2D robot for the frontal-balance purposes. Collins et al. extended McGeer's ideas into 3D and designed a truly 3D two-legged passive-dynamic robot [20].

While fully passive walkers can only walk downhill, a small amount of actuation can be added to a passive-model design to realize walking on level ground [21]. Collins et al. equipped their 3D passive robot with ankle motors which were used to push-off just before each collision [22]. The robot successfully walked on flat ground at 0.44 m/s.

By making use of passive motions (e.g. passively swinging their leg between collisions), passive-dynamic-based robots tend to be energy effective. For example, Collins' powered biped used only 11 W to walk, making it one of the most energy-efficient walking robots to date (inferior only to Cornell Ranger [9], see Section 6.1). However, fully passive walkers have only a small basin of attraction around their periodic trajectories and require skill to be successfully started to steadily walk downhill [20]. Passive-dynamics-based robots share poor robustness with their passive parents. Collins' powered biped, designed based on its passive counterpart, could not walk reliably even on level ground. In controllers we develop in this thesis, we achieve a much stronger robustness (e.g.

see Fig. 3.7 on page 94).

1.1.2 Zero moment point

The *Zero Moment Point (ZMP)* is a point on the ground where the net moment of the ground reaction forces acting on the robot has a zero horizontal component[1] [87, 86]. In single stance, when only one foot is in contact with the ground, the ZMP is located either inside or on the edge of the stance foot. In double stance, the ZMP can be either below one of the feet or anywhere between the feet (anywhere in the convex hull of the contact regions of the two feet, called the support polygon). Relative locations of the ZMP and the projection of the Center of Mass (CoM) on the ground determine the overall motion of the robot. For example, the CoM must be directly above the ZMP for the robot to be completely stopped. When the CoM is not directly above the ZMP, the ground reaction forces have a moment about the CoM; this moment drives the CoM projection in the direction away from the ZMP.

Control over the ZMP location within the support polygon — e.g. with appropriate ankle torques — can be used to control the overall motion of the robot [85, 33]. For instance, in the standing balance task, the CoM has to be above the support polygon at all times. Therefore, to best maintain standing balance, the ZMP should be placed between the current CoM projection on the ground and the closest edge of the support polygon, thus pushing the CoM towards the center of the support polygon. This strategy can be extended to produce statically-stable walking: balancing on one foot is alternated with the

[1] More generally, a zero component along the surface of the ground, if the ground is not necessarily level.

double stance phase where the robot's weight is transferred from the hind to the front leg. A ZMP controller can also be used alternately with a footstep controller — such as one based on Pratt's capture regions (Section 1.2) or on the controllable regions described in this thesis — to produce dynamic walking or to recover from large disturbances [56].

Because ZMP-based controllers primarily rely on ankle torques, it is necessary for the robot to have large feet. If the feet are small, e.g. vanishingly small in point-feet models, or when the ZMP is close to the edge of the support polygon, the robot has small control authority over its ZMP and, hence, CoM location. In such cases, the robot may have to take a step to avoid falling. Also, the ZMP approach is originally based on static stability, which is not a requirement for, and does not occur naturally in, human locomotion. Robots primarily based on ZMP control, such as Honda's Asimo [73], tend to show walking and running gaits that do not resemble those of humans: they tend to have flat feet, bent knees, high energy use [21].

1.1.3 Hybrid zero dynamics

The central idea of the *Hybrid Zero Dynamics (HZD)* approach is dimensional reduction of the robot's dynamics [89, 30]. Feedback controls are used on all fully controlled degrees of freedom (internal joint angles) so as to effectively make them functions (*'slaves'*) of the uncontrolled degrees of freedom (ankle-joint angles in point-feet robots).

The HZD problem is to design the slave functions so as to produce a steady gait of the robot. The gait has to be stable and is sometimes picked to be in

some way optimal (e.g. maximally energy efficient). To find appropriate slave functions, numerical optimization can be used with both a measure of stability (such as eigenvalues) and energy cost included into the objective function [17]. Additionally, Grizzle [31] proposed using an event-based controller to change some parameters of the slave functions — hence, the nominal gait of the robot — in a discrete manner (say, once per step). This may allow the robot to change its gait characteristics, such as speed, on-the-fly and increase robustness. The HZD methods were successfully implemented to achieve steady walking of 2D robots [17, 76] and of a 3D robot in simulation [18].

A drawback of the HZD approach is that it uses high-gain controls of the internal (slaved) degrees of freedom. Such controls seem to tend to be energetically expensive and produce non-compliant behavior that is not natural. Also, HZD-based controllers realize precomputed trajectories which cannot be changed online — but for parameter adjustment with event-based controls as described above — to adapt to changes in the locomotion goals or environment, e.g. to avoid an obstacle.

1.1.4 Model predictive control

The *Model Predictive Control (MPC)* method uses on-the-fly optimization to compute the best control actions at any instant during locomotion [26, 47, 6]. For a given current state of the robot, an optimization problem is formulated to find optimal controls over a fixed *time horizon* into the future. An internal model of both the robot and the environment is used in the optimization routine. The optimality criteria may include stability, energy use, or any other desired ob-

jective; constraints can be incorporated to account for, say, actuation limits or obstacles on the ground. The calculated solution — the optimal set of controls — is implemented on the robot. The optimization problem is repeatedly solved online at a high frequency to update the optimal control actions.

The MPC is a universal approach which allows one to consider various locomotion problems: any optimization objective and constraints can be specified. The approach also naturally provides robustness: frequent update of the optimal control actions lets the robot promptly react to disturbances. However, running frequent optimizations on-the-fly requires significant computer power. This issue may be somewhat overcome by sacrificing accuracy of optimal solutions for speed of computation, as demonstrated in simulation by Mordatch et al. [54]. It seems likely that MPC methods will become more widespread with the advance of faster computers and optimization technics.

1.1.5 Optimal control policies

As opposed to the MPC methods (Section 1.1.4) where control actions are determined by running *on-the-fly* optimizations, one can compute optimal controls *offline* for each possible state of the robot. The offline optimization uses a discretization of the state space and typically employs the methods of dynamic programming [8]. The optimal solution — *optimal control policies* (control actions) at each discretization point — can be either used as a lookup table or fitted with parameterized functions, thus defining a controller of the robot. The optimal policies approach has been used for walking robots in simulation [48, 90, 45]. The approach is subject to the curse of dimensionality and, hence, its applica-

tion to many-degree-of-freedom models is problematic.

1.2 Capture regions

We now describe the capture region and viability theory concepts. They are closely related to our controller design approach and are the true starting point of this thesis.

The capture-regions concept was originally developed by Pratt, et al. [64, 66, 42]. The concept studies the ability of a biped to come to a full stop, in one, two, or more steps. A 'full stop' can be standing on one foot, on both feet, or any statically stable state of the robot.

For a given initial state, a point on the ground where the robot can step and come to a full stop without taking any further steps, is called a *one-step capture point*. The set of all such points is the *one-step capture region*. Similarly, *the n-step capture region* is defined as the set of all points on the ground where the robot can step and come to a stop in at most n steps in total. The limit of the n-step capture regions as $n \to \infty$ is the ∞-*step capture region*; it includes all possible stepping locations which allow the biped to eventually come to a stop.

The capture regions are useful in designing a footstep controller: at each instant of time, they show where the robot may step to to be able, using appropriate subsequent control actions, to come to a stop, if desired, and, hence, to avoid falling. Stepping inside the capture regions at each step is sufficient to be capable, using the given robot and actuators, to totally avoid falling. Such strategy is used by Pratt et al. [42, 63] to design a footstep controller for a 12-

degree-of-freedom robot in 3D. In the beginning of each step (just after toe-off), the next footstep location is planned based on the desired speed of the robot and the current stance foot position. Then, the planned stepping location is adjusted, if necessary, to be inside the one-step capture region corresponding to the current state of the robot. During the swing phase, the planned footstep is continuously recomputed to react to possible perturbations and, hence, changes of the one-step capture region. Pratt et al. use the Linear Inverted Pendulum model with finite-sized feet [42] as a proxy for the robot, to approximate the capture regions. This stepping controller is shown to generate walking motion (both in simulation and of the real robot) with successful recovery (in simulation) from sideways and forward pushes [63].

The size of the capture regions can be an indirect measure of robustness: for smaller capture regions, the robot is closer to being non-capturable (there being no possible place to step that avoids a fall) and a smaller random disturbance is required to cause the robot to fall [42, 63]. However, the capture regions do not indicate the maximum magnitude of disturbances that the robot can survive, nor what type of disturbances the robot is most sensitive to. Such knowledge about disturbances may be useful for robustness optimization of the controller. Also, as Pratt et al. [63] point out, one is often interested in maintaining a desired speed of the robot, rather than making it come to a stop. Therefore, the capture regions are a tool restricting permissible controls (stepping locations), rather than a means to achieve a desired trajectory of the robot.

The capture regions are defined for a given specific initial state of the robot. Koolen et al. [42] introduce the *viable-capture basins* to account for different initial states. The n-step viable-capture basin is a region in the state space that includes

all states for which the model is n-step capturable, i.e. is able to come to a stop within n steps. That is, it is the set of all initial states for which the n-step capture region is non-empty. Similar to the capture regions showing 'good' stepping locations, the viable-capture basins show 'good' states.

1.3 Viability theory

Developed by Aubin [4, 2, 3], viability theory studies behavior (trajectories) of a controlled dynamical system with passage through given regions of the state space not allowed. Such passage constitutes a failure. Of interest are the abilities of the system to avoid a failure and to reach a specific target state or a set of states.

Let K be the region in the state space that includes all allowed states, i.e. all states that satisfy a given set of constraints. For example, for walking robots, one may consider the non-falling constraint — hence K is all non-fallen states of the robot. An evolution of the system is called *viable*, if it always remains inside K, that is never fails. The set of all initial states for which there is at least one viable evolution is called the *Viability Kernel*. When inside the viability kernel, the system is able, with appropriate controls, to never fail (e.g. for walking robots, to never fall down). For any initial state outside the viability kernel it is impossible to avoid a failure.

Next, assume C to be a desired state or set of states of the system; C is called the *target region*. For walking robots, an example of such a target is all states for which the robot is standing still, as for Pratt's capture regions [64, 42] (Section 1.2 above). For a given target C, the *Capture Basin of C* is the set of all initial states

for which there is at least one viable evolution reaching C in a finite time. That is, the capture basin includes all states from which the target can be reached with appropriate controls.

Some of the basic problems of the viability theory are: find, or approximate, the viability kernel and capture basin for a given system [70, 71, 23]; construct a controller that attempts to keep the system inside the viability kernel at all times and to drive the system to a given target [7, 81, 72]. Updating the viability kernel and capture basin on-the-fly may help adapt the controller to possible changes in the system due to, say, external disturbances or changes in the environment. The viability theory was introduced into the field of legged locomotion by Wieber [91, 92, 93], who studied approximation of the viability kernel of a legged robot. However, a controller design accounting for the viability kernel was not proposed. Such a controller, for instance, would produce the basin-of-attraction of a given desired trajectory that is maximally close to the viability kernel, thus maximizing the range of perturbations the robot can recover from.

1.4 Control approach in this thesis

Here we design a maximally robust walking controller for a planar bipedal robot. The ideas in the following paragraphs are discussed in more detail later in the thesis. Our design approach uses the concepts of the viability theory (Section 1.3) and is a generalization of Pratt's capture-regions ideas (Section 1.2).

First, we generalize Pratt's viable-capture basins to consider any goal of locomotion (e.g. a specific desired trajectory), as opposed to just coming to a stop. We define the *n-step controllable region* C_n as all states from which the robot can,

with appropriate controls, reach a given target by taking at most n steps and not fall.[2] The ∞-step controllable region C_∞ is the limit of C_n as $n \to \infty$. When inside C_∞, the robot is always able to return to the target. Hence, one goal for our controller is to keep the robot inside C_∞ at all times.

Next, we study controls of a robot by considering the *extended state space* which includes both states and controls. A point in such extended space is a combination (q, u) of a state q and controls u to be used in that state. Similar to the regions C_n, we define the *extended n-step controllable regions* $\bar{C}_n$ — regions in the extended state space — as all combinations of states and controls which allow the robot to reach the target in at most n steps. The ∞-step region $\bar{C}_\infty$ is the limit of $\bar{C}_n$ as $n \to \infty$. For the robot to always be able to return to the target (to remain inside C_∞), it is sufficient to always choose controls inside $\bar{C}_\infty$ that correspond to the current state of the robot. Note, that our extended controllable regions are a generalization of Pratt's capture regions, when considering footstep locations as high-level controls of the robot — see more details in Section 2.6.5.

A curve (hyper-surface) inside the region $\bar{C}_\infty$ defines a controller for a given target of the robot. Our design problem is to find a controller curve that nearly maximizes robustness and, if possible, is approximately optimal in some other regards, such as energy efficiency, speed of convergence to the target, and simplicity. Robustness is maximized by choosing controller curves that are possibly farther inside the region $\bar{C}_\infty$, thus making it less likely that a random perturba-

[2] What we term here *controllability* (or *controllable*) is traditionally called *capturability* in the Viability Theory [3]. Both terms are defined with respect to an arbitrary target of the system. However, in the legged robotics community a conflicting definition is used: 'capturability' is associated with Pratt's capture-regions concept [64] and assumes the particular target of standing still. To avoid confusion, we use the term 'controllability' to mean something close to it definition in the classical Control Theory [39].

tion puts the robot outside C_∞. Optimization with respect to other objectives can be done by evaluating all points inside $\bar{C}_\infty$ with respect to each of the objectives. We make the robot energy efficient by choosing a target trajectory that approximately minimizes the energy cost of walking.

Similar to controllability of a biped, we also study *viability*: the ability of a robot to avoid falling down. We define the *n-step viable* and *extended n-step viable regions* which show all states and corresponding controls that let the robot take at least n steps and not fall. Both the viability and controllability concepts help us make two claims about bipedal locomotion (Section 2.3) and then justify the claims for two simple models of walking.

1.5 Contributions of the thesis

The main results and contributions of this thesis are as follows.

1. **Viability and controllability framework.** We generalized Pratt's capture-region concepts [42] to define the viable and controllable regions and their extended counterparts. These regions study the ability of a robot to avoid falling and to reach a given locomotion goal. The viability and controllability framework can be used as a tool in the analysis of stability, robustness, and other properties of a biped and its controllers. The framework provides a controller design method that nearly optimizes robustness and some other objectives.
2. We introduced the **extended state space** of a robot. A point in this space is a pair (q, u) of the state q and controls u to be used in that state. Curves

(surfaces) in the extended space are controllers of the robot. Analysis of the extended state space helps one study, compare, and design robot controllers.

3. **Design of a maximally robust walking controller.** We used the viability and controllability framework to design a high-level walking controller for the 2D robot Cornell Ranger. Once per step, the controller provides target values for the next step-length and the push-off magnitude at the next heel-strike. The controller provides stability around the nominal trajectory, is nearly maximally robust, has low energy cost, does not require significant online computations, and is simple in structure.

4. **Larger steps are good.** We discovered that increasing the step length is preferable for the robot Cornell Ranger in response to any disturbance. Whether walking too fast or too slow, taking bigger steps makes the push-offs more effective at controlling speed. Because we modeled Ranger with a simple point-mass model, we believe that this result applies to most bipedal robots.

5. **'Viable is controllable'.** We considered the following claim about bipedal locomotion (a slight generalization of Pratt's [38]): if a robot can avoid falling, then in most cases it also can reach any given target (e.g. a specific trajectory). We used the viability and controllability concepts to justify the claim for two simple point-mass models.

6. **'Two steps is almost everything'.** We considered the following claim about bipedal locomotion (a slight generalization of Koolen et al.'s [38]): if a biped can reach a given target at all, in most cases it can do so within two steps. Practically, in controller design (e.g. in model-predictive control), there is no need, at least for balance purposes, to plan the robot's

motion more than two steps ahead. We justified the claim for two simple point-mass models and provided supporting evidence from human-balance experiments.

CHAPTER 2

VIABLE AND CONTROLLABLE REGIONS

We would like to understand how a person does, and a robot might, both not fall down and also achieve desired goals of locomotion, such as speed, direction or path of motion, and foot placements. We try to understand the avoidance of falls with the concept of *viability*, and the achieving of different locomotion tasks through *controllability*.

The ability of a robot to avoid a fall or perform a specific task depends on the actuators available to it. A physically realistic model assumes limitations on the actuator capabilities, such as the maximum torque produced by a motor. By *feasible controls* of the robot we will mean controls (actuation, control actions) that can be implemented on the robot for given actuation limitations.

We will focus in our studies on how the state of a robot (positions, angles, and rates) changes after each step. We say that a step starts at *midstance*, which is defined as the point of local maximum in potential energy of the system, and ends at the next midstance. For simple point-foot planar bipeds (e.g. those presented in Sections 2.4.1 and 2.4.2), the midstance is the instant when the center of mass passes exactly above the stance foot. The midstance can be viewed as a Poincaré section of the system, while the map from a state of the robot at a midstance to the state at the next midstance a Poincaré map. Looking only at states at the Poincaré section of the robot makes the problem both discrete and lower-dimensional. In the rest of this thesis, by saying 'state' we will usually mean a state of the robot at this Poincaré section, unless otherwise stated.

2.1 Viable regions

One approach to understanding stability is through viable regions. These are the sets of states from which it is possible for the robot to avoid failure and also take one or more steps. A robot fails if it reaches a *failed state*. The definition of a failed state that we will use is a state of the robot, for which one, or both, of the following is true:

- a part of the robot other than the feet touches the ground (i.e. the robot has fallen);
- one of the constraints, if any, imposed on the robot is not satisfied.

Examples of possible constraints include actuator limitations, allowed foot placement locations, direction, or speed of motion (see Section 2.5.1 on page 33 for more details). Any non-failed state for which the robot is at a midstance, is called 0-*step viable*, and the collection of all such states is the 0-*step viable region* V_0.

Thus, the region V_0 is a subset of the state space of the model; V_0 includes all states at midstance allowed for the robot. We next look at the states, starting from which the robot is able to make one or more steps — these are not necessarily the entire V_0. A given state of a robot is called *1-step viable*, if it is 0-step viable and there is a feasible control which allows the robot to take (at least) one step without failing. The *1-step viable region* V_1 is the collection of all such states. Note, that making a step requires reaching the next midstance (next Poincaré section) — this condition excludes the situations when the robot is headed down, i.e. falling, the whole time. Similarly, the *n-step viable region* V_n

is all 0-step viable states, from which it is possible for the robot to make at least n steps and not fail. Any element of V_n is an *n-step viable state*. Viable regions are shown schematically on Fig. 2.1.

We can also see that the regions V_n form a sequence of nested regions: $V_n \subset V_{n-1}$ for any $n \geq 1$. Indeed, if the robot is able to take at least n steps without failing, then, of course, it can take at least $n-1$ steps as well. In this way, the n-step viable region V_n can be thought of as all (midstance) states of the robot, from which it is possible, for given constraints and feasible controls, to reach V_{n-1} by making exactly one step. This one-step relation between V_n and V_{n-1} is useful for computation of viable regions, as shown in Section 2.7.2 on page 55. On the other hand, states which are in V_{n-1}, but outside of V_n (i.e. in $V_{n-1} \setminus V_n$) are such that from them the robot can step $n-1$ times, but no feasible control will be able to prevent it from failure on the n-th step. For example, if the robot is in a state in $V_0 \setminus V_1$, it is not failed now, but will necessarily fail before making a single step.

Of large interest is the limit of the sequence $\{V_n\}$, which we call the ∞-step viable region V_∞. It includes all points in the Poincaré section from which the biped can make an arbitrarily large number of steps. In other words, if a robot is in V_∞, then there exist feasible controls that allow it to 'never' fail (here we mean never step-wise — stepping indefinitely does not necessarily mean avoiding a failure within any finite interval of time, unless duration of each step is constrained from below by a positive number). On the other hand, if a robot is in a state outside of V_∞, then no feasible control will be able to prevent it from failing eventually, i.e. within a finite number of steps. Hence, one of the basic objectives for a robot controller is to always keep the robot inside the ∞-step viable

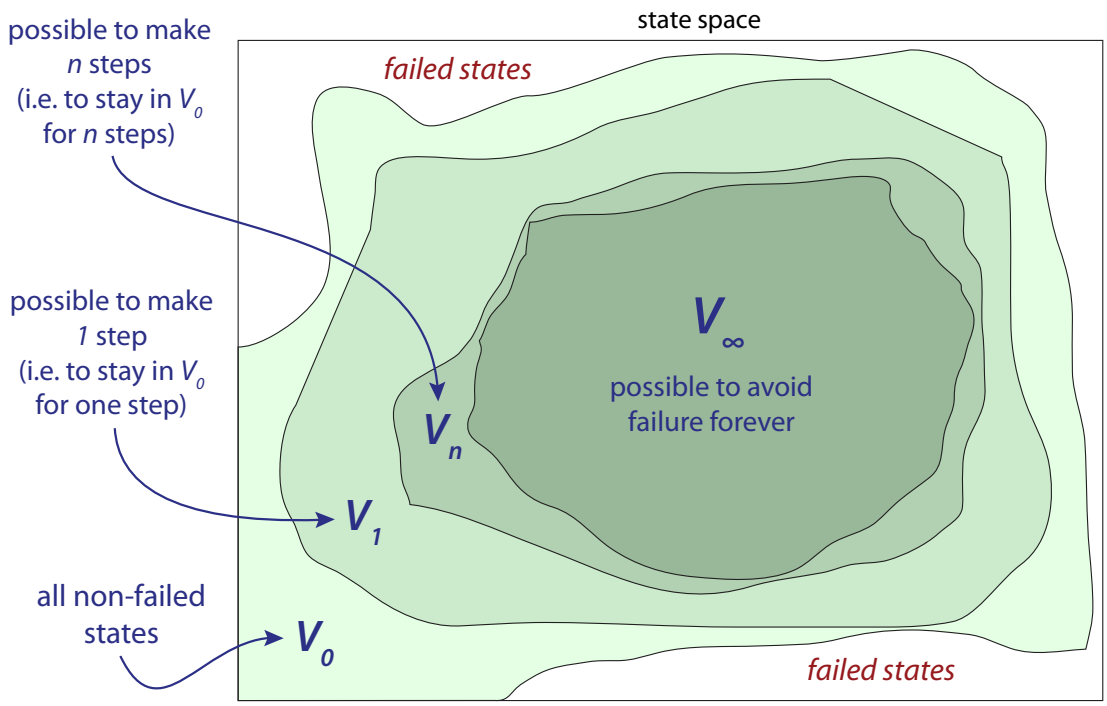


Figure 2.1: Schematics of viable regions. This figure schematically shows n-step viable regions V_n for a robot. The region V_0 is the set of all states in Poincaré section (midstance) for which the robot is not failed (is not fallen and meets all given constraints). V_n is a subset of V_{n-1} and includes all initial states at midstance from which it is possible, by means of feasible controls, to take at least n steps without failing. The ∞-step viable region V_∞ is all states from which it is possible to take infinitely large number of steps without failing — it is the limit of the sequence of sets $\{V_n\}$. Note, taking an infinite number of steps does not necessarily yield avoiding a failure forever time-wise, unless duration of each step is bounded from below by a positive number.

region V_∞. The region V_∞ is equivalent to the viability kernel — a concept from the viability theory discussed in Section 1.3 above.

2.2 Controllable regions

We define controllable regions with respect to more specific goals, instead of just the goal of not failing. We may have a specific overall configuration, location on the ground to stand at, or point in the phase space that we desire to achieve (see

Section 2.5.2 on page 35 for examples). For a given goal, let C_0 be the set of all states at midstance satisfying the goal. Then we say that C_0 is the *target region* (*0-step controllable region*), and any element of C_0 a *target state* (*0-step controllable state*). Note, that there can be more than one target state, i.e. more than one point in C_0 (it could be several points, a region in the state space, or a set of disjoint regions in the state space).

The n-step controllable regions are defined in a way similar to the regions V_n: C_n is the set of all 0-step viable states from which the biped can, with appropriate feasible controls, reach one of the target states in C_0 in n or fewer steps and without failing. Alternatively, C_n is all states in V_0, from where the robot can get to C_{n-1} within one step. As with the viable regions, this one-step relation can be conveniently used in numerical calculation of the controllable regions — see Section 2.7.1 on page 49 for details. The sequence $\{C_n\}$ is nested ($C_{n-1} \subset C_n$ for any $n \geq 1$), and its limit is called the ∞-*step controllable region* C_∞. It is the set of all points in the Poincaré section from which the biped can, with appropriate feasible controls, either reach the target region C_0 in a finite number of steps or approach it asymptotically. The controllable regions are shown schematically in Fig. 2.2.

Suppose the goal is all 'captured' states of the robot (all states where the robot can avoid falling down ever without making a single step, e.g. standing still — the concept introduced by Pratt [64, 42] and discussed in Section 1.2). In this case, regions C_n are equivalent to Pratt's n-step viable-capture basins [42]. So the n-step controllable regions for an arbitrary target C_0 are a generalization of the viable-capture basins concept.

The controllable regions C_n depend on the chosen target C_0. For each n, the

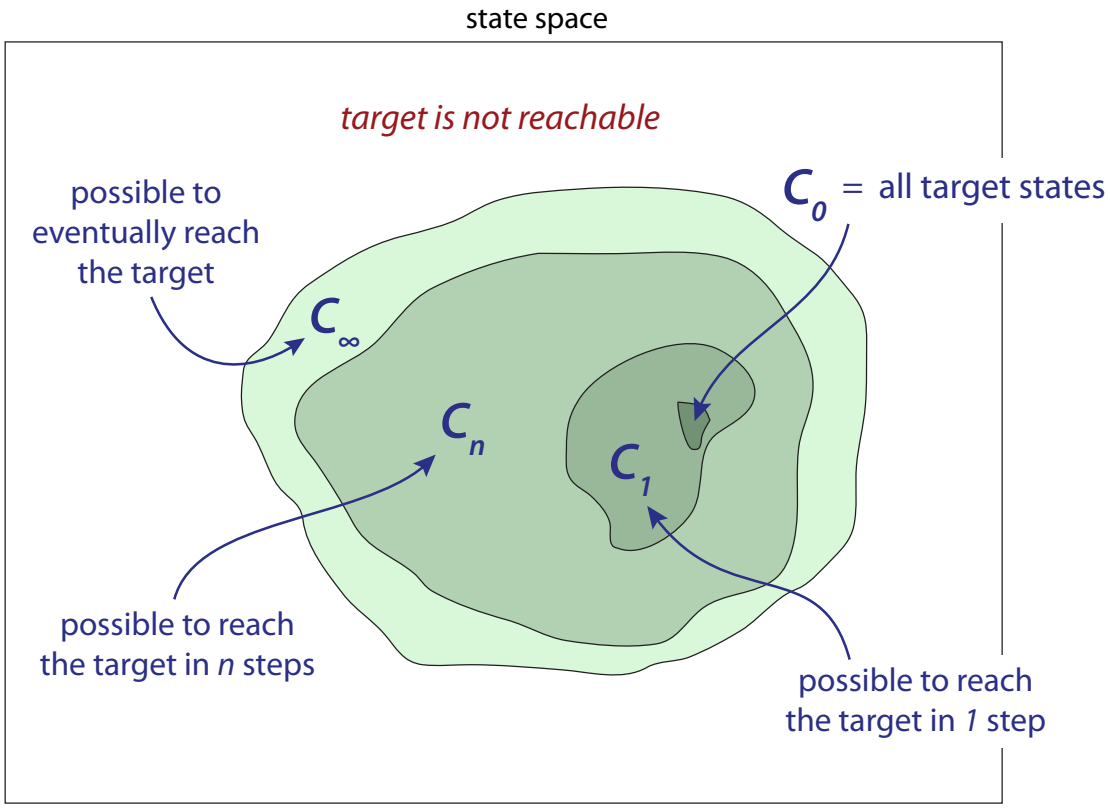


Figure 2.2: Schematics of controllable regions. Examples of n-step controllable regions C_n for a robot are schematically shown on this figure. All C_n are subsets of the Poincaré section of the robot. For a given goal (e.g. standing upright), the target region C_0 is the collection of all states which satisfy the goal. C_n consists of all initial states of the robot, from which it is possible, with given constraints and feasible controls, to reach C_0 by taking n or fewer steps without failing. The limit of the sequence $\{C_n\}$, the ∞-step controllable region C_∞, is all states from which it is possible to either reach the target in a finite number of steps or approach it asymptotically. If the biped is in a state outside of C_∞, no feasible control will let it reach the target.

region C_n can be represented as the result of a set-valued mapping, applied to C_0. We define such mapping for the 1-step controllable region. Let 2^{V_0} be the set of all subsets of V_0 (i.e. the power set of V_0). Define C as a function from 2^{V_0} to 2^{V_0}, such that

$$C(C_0) = C_1. \tag{2.1}$$

We call the map C the *controllability map* of the system. The controllability map assumes its argument, a region in V_0, to be the target region, and returns the 1-

step controllable region corresponding to this target. In the case of a single-point target $C_0 = \{q\}$, we will use a simplified notation $C(q)$, instead of $C(\{q\})$.

Next, consider the n-step controllable region C_n as all states, from which the region C_{n-1} can be reached within one step. Therefore, C_n is the 1-step controllable region corresponding to the target C_{n-1}:

$$C_n = C(C_{n-1}), \qquad \text{for } n \geq 1. \tag{2.2}$$

By similarly expressing C_{n-1} in terms of C_{n-2}, then C_{n-2} in terms of C_{n-3}, and so on, the above equation can be transformed as follows, to represent the region C_n as a function of C_0:

$$C_n = C(C_{n-1}) = C(C(C_{n-2})) = \ldots = C^n(C_0).$$

Therefore, the map C^n, the n-th power of the controllability mapping (2.1), is a set-valued function, which for any input target region C_0 returns the corresponding n-step controllable region C_n:

$$C_n = C^n(C_0), \qquad \text{for } n \geq 1. \tag{2.3}$$

For the ∞-step controllable region C_∞, the limit of the sequence of C_n, we use the following notation:

$$C_\infty = C^\infty(C_0), \tag{2.4}$$

where $C^\infty : 2^{V_0} \to 2^{V_0}$ is the point-wise limit of the sequence of the functions C^n as $n \to \infty$.

Unless otherwise stated, in this thesis by C_n we mean the n-step controllable region corresponding to a specified target region C_0. For controllable regions corresponding to a target X, different from C_0, we use the map C^n to explicitly point out the target: $C^n(X)$ (or $C^n(q)$ for a single-point target $X = \{q\}$).

2.2.1 Strict controllable regions

In addition to the controllable regions C_n, introduced in Section 2.2 above, we also define the *strict n-step controllable regions* $\hat{C}_n$. We do not study the regions $\hat{C}_n$ in detail, but only employ them in this thesis to define the extended controllable regions (see Sections 2.6.2 and 2.6.3) and to numerically compute the viable regions (Section 2.7.2), extended viable regions (Section 2.7.3), and extended controllable regions (Section 2.7.4).

For a given model, constraints, feasible controls, and a target region C_0, we say that a 0-step viable state q of the robot is *strict n-step controllable*, if there is a way for the robot to start the motion in the state q, take exactly n steps without a failure, and reach a target state in C_0 at the end of the n-th step. The *strict n-step controllable region* $\hat{C}_n$ is the set of all strict n-step controllable states.

Similar to the controllability map C for the controllable regions C_n (see equation (2.1) on page 20), we describe the strict controllable regions $\hat{C}_n$ in terms of a set-valued map $\hat{C}$ acting on the target C_0. We say that the map

$$\hat{C} : 2^{V_0} \to 2^{V_0}, \qquad \text{s.t. } \hat{C}(C_0) = \hat{C}_1, \tag{2.5}$$

is such that for any target region C_0 (the input) it returns the corresponding strict 1-step controllable region $\hat{C}_1$. Here 2^{V_0} is the power set of V_0, i.e. the set of all subsets of V_0. We call the map $\hat{C}$ the *strict controllability map* of the system.

The region $\hat{C}_n$ can be described as all states, starting from which the robot is able to reach a strict $(n-1)$-step controllable state (i.e. to reach the region $\hat{C}_{n-1}$) by making exactly one step and not failing. That is, $\hat{C}_n$ is the strict 1-step controllable region $\hat{C}(\hat{C}_{n-1})$ corresponding to the target $\hat{C}_{n-1}$, and we write

$$\hat{C}_n = \hat{C}(\hat{C}_{n-1}) = \hat{C}(\hat{C}(\hat{C}_{n-2})) = \ldots = \hat{C}^n(C_0).$$

Thus, each region $\hat{C}_n$ is represented by the corresponding power of the map $\hat{C}$:

$$\hat{C}_n = \hat{C}^n(C_0), \qquad \text{for } n \geq 1. \tag{2.6}$$

The *strict ∞-step controllable region* $\hat{C}_\infty$ is defined as all states at midstance, from which the robot can, with appropriate feasible controls, approach the target C_0 asymptotically. It is the limit of the regions $\hat{C}_n$ as n increases to infinity. We represent it by the set-valued map $\hat{C}^\infty$, the point-wise limit of the maps $\hat{C}^n$:

$$\hat{C}_\infty = \hat{C}^\infty(C_0). \tag{2.7}$$

Below in this thesis, by $\hat{C}_n$ we mean the strict n-step controllable region corresponding to the specified target region C_0. Strict controllable regions corresponding to a different target X are referred to as the value of $\hat{C}^n(X)$.

Each region $\hat{C}_n$ is a subset of the corresponding n-step controllable region C_n, and the following relation holds true:

$$C_n = \hat{C}_n \cup C_{n-1}. \tag{2.8}$$

That is, as illustrated on Fig. 2.3, the region C_n is all states, from which the target can be reached in exactly n steps (i.e. all points in $\hat{C}_n$), plus all states from which the robot can reach C_0 in $n-1$ or fewer steps (i.e. all points in C_{n-1}). Substitution of $n=1$ in equation (2.8) yields the relation between the controllability maps C and $\hat{C}$:

$$C(C_0) = \hat{C}(C_0) \cup C_0. \tag{2.9}$$

However, if each target state is on a (1-step) periodic trajectory, then the regions C_n and $\hat{C}_n$, as well as the maps C and $\hat{C}$, are the same. Note also, that as opposed to the regions C_n, the strict controllable regions $\hat{C}_n$ do not necessarily have a nested structure.

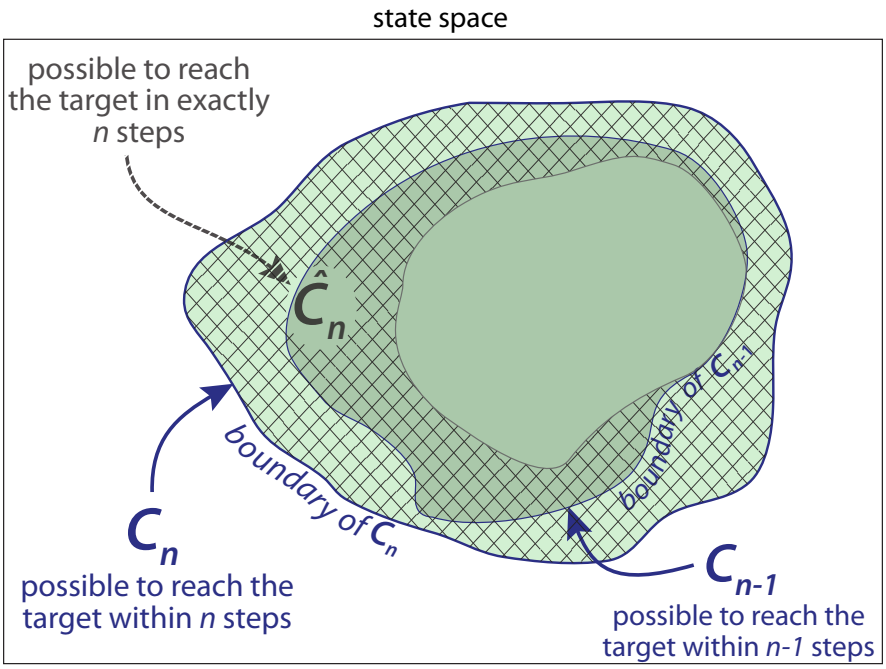


Figure 2.3: Controllable and strict controllable regions. The figure contrasts the n-step controllable region C_n with the strict n-step controllable region $\hat{C}_n$, and illustrates equation (2.8) on page 23. C_n is all states, from which the robot is able to reach a given target C_0 in n or fewer steps. $\hat{C}_n$ includes all states, from which the robot can reach C_0 in *exactly* n steps. The strict region $\hat{C}_n$, shown as the cross-hatched area, is a subset of C_n. The regions $\hat{C}_n$ and C_{n-1} together form the entire region C_n, and may also have points in common. If the target states in C_0 allow (1-step) periodic motion, the n-step regions C_n and $\hat{C}_n$ are the same.

2.2.2 Disturbances, model errors, and noise

For now, in construction of the viable and controllable regions concepts, we assume no disturbances occur during the motion and both the model knowledge and all sensor inputs, if any, are perfect. It is possible, though, to take them into consideration if characteristics of the noise and possible disturbances/errors are known. For example, one might introduce an additional requirement that all goals and constraints at each step are met for any disturbances and model errors within given bounds. We will use this approach in the design of a walking controller for a biped presented in Chapter 6.

2.3 Viability and controllability conjectures

At this point we have not specified the mechanical model(s) (number of links, actuator limits, etc) for which we are going to apply the concepts of viability and controllability. Nonetheless, we make statements, which we think are true for most bipedal robots. However, exceptions exist and their examples are mentioned in this section below.

2.3.1 Viable is controllable

Here we suggest a relation between ∞-step viability and ∞-step controllability. Suppose that the target region C_0 is entirely inside the ∞-step viable region: $C_0 \subset V_\infty$. Then, once the robot is in a target state, it is able to step forever (stepwise). Therefore, the ∞-step controllable region C_∞ (as well as all C_n) is a subset of V_∞ too:

$$C_0 \subset V_\infty \qquad \Longrightarrow \qquad C_\infty \subset V_\infty \tag{2.10}$$

From points in C_∞ the robot can either reach the target and then step indefinitely or asymptotically approach the target, thus, again, taking an infinite number of steps.

The statement (2.10) is a true, logically established result. We cannot make it stronger staying mathematically precise — however, we make a less precise statement about the difference $V_\infty \setminus C_\infty$ between the two sets. We claim that for most robots and most targets, the states from which it is possible to avoid failing, yet not possible to reach the target, are rare or nonexistent:

Claim A (*Viable is Controllable*): In most cases and for most bipeds, the

ability of a biped to take an arbitrary large number of steps is equivalent to its ability to reach any specific target. More formally, for most robots, the set of states which are inside V_∞, but outside of C_∞, is a zero-measure set for any target inside V_∞: $\mu(V_\infty \setminus C_\infty) = 0$ if $C_0 \subset V_\infty$ [1].

The claim is illustrated on Fig. 2.4a. Examples of the robots and targets for which the claim is not correct are provided in Appendix A.

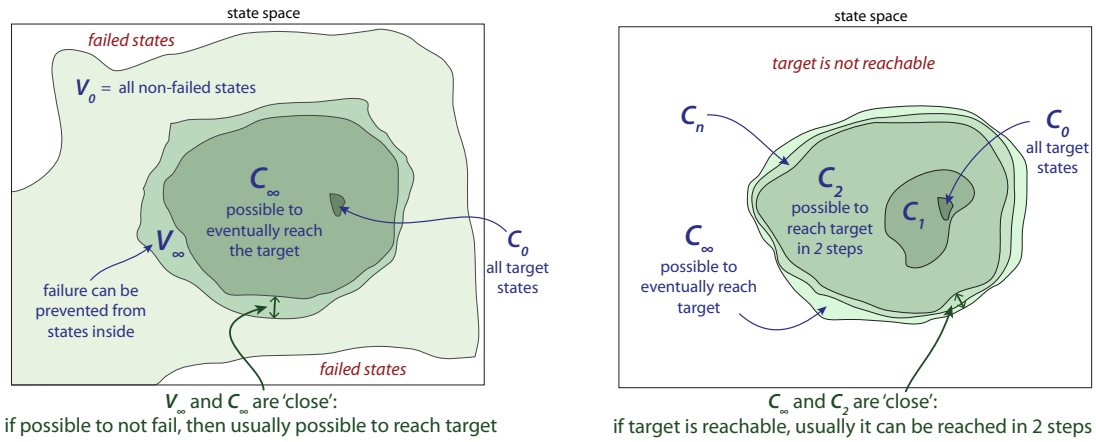
Pratt et al. make a similar statement in [38, 42], where they consider the target C_0 to be all captured states. They claim that the part of the state space, from which the robot is able to never fall down, but not able to ever reach a captured state (i.e. $V_\infty \setminus C_\infty$) is 'small'. One exception, suggested in [42], to both Pratt's and ours claim is purely passive walkers, which, once on a periodic gait, keep walking forever, but can never come to a stop (thus, never reaching the target, if the target is standing still) — see Appendix A for more details.

Although we cannot prove the $V_\infty \approx C_\infty$ claim above, we test it on simple walking models often used in the robotics community: the Inverted Pendulum (IP) model in 2D [43, 36] (introduced in Section 2.4.1 and discussed in detail in Chapter 3) and Linear Inverted Pendulum (LIP) model in 2D [40] (introduced in Section 2.4.2 and discussed in Chapter 4).

2.3.2 Two-step controllability

In addition to Claim A, we slightly generalize a hypothesis made by Koolen et al. [42] — we make the following statement about a general model of a biped

[1] By $\mu(\cdot)$ we denote a measure (metric of volume) in the Poincaré section of the model.



(a) 'Viability equals controllability' claim **(b)** 'Two-step controllability' claim

Figure 2.4: Illustration of controllability statements. The figure illustrates two statements we make in Section 2.3, which we think hold true for most bipeds. **(a)** Claim: for any target C_0 inside the viability kernel V_∞, the difference between V_∞ and the ∞-step controllable region C_∞ is 'small'. In other words, if the robot can avoid a failure, then in most cases it can reach a given target. Exceptions, such as passive walkers, exist. **(b)** Claim: for any target C_0 inside the viability kernel, the 2-step controllable region C_2 is most of C_∞. That is, if the robot can get to the target at all, in most cases it is capable of doing it in two steps. We test the claims on simple IP and LIP models in 2D.

and its controllable regions with an arbitrary target:

> **Claim B** (*Two-step controllability*): For a given model of a biped and an arbitrary target, if it is possible for the biped to reach the target at all, in most cases it can be done in two or fewer steps. More formally, assume the target C_0 is inside the viability kernel. Then the 2-step controllable region C_2 is most of the ∞-step controllable region C_∞: $\mu(C_\infty \setminus C_2) \ll \mu(C_\infty)$ for any $C_0 \subset V_\infty$.

An illustration is provided in Fig. 2.4b. The statement suggests that in designing a controller (e.g. for recovering from a disturbance or path planning) there is no need, except possibly for some extreme situations, to plan more than two steps ahead. As opposed to Koolen et al.'s conjecture, which considers only captured states (e.g. standing still) as possible target states, Claim A allows an arbitrary

target region in the viability kernel.

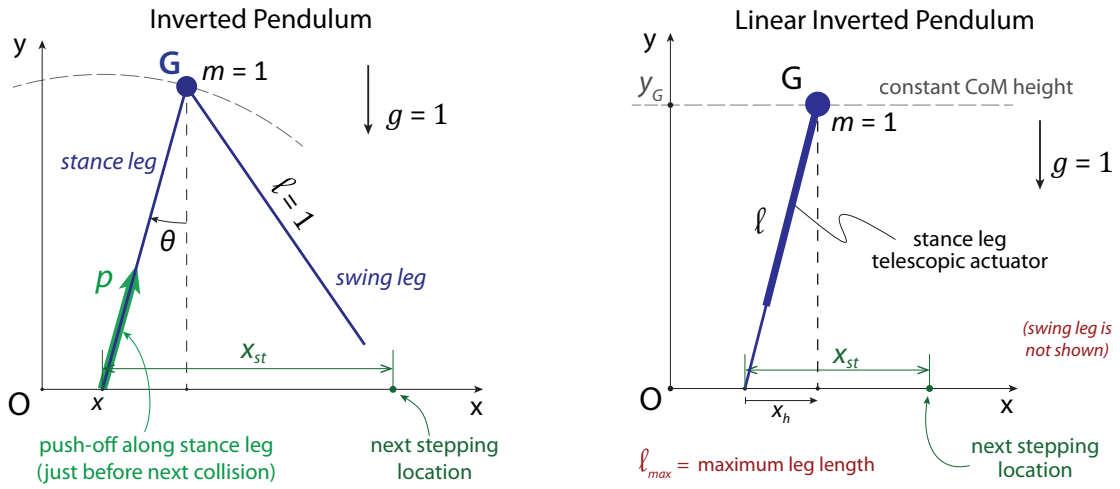
Note, that taking into account the Viable is Controllable Claim A (see Section 2.3.1), the Two-Step Controllability Claim B also suggests: in situations in which it is possible to avoid failing, it is also possible to reach any given target state (in the viability kernel) by taking two or fewer steps.

There is some evidence to our claim coming from treadmill experiments with humans [35, 74, 59, 61, 37]. In the robotics community several groups have used two-step policies to control a biped [46, 83, 15]. We test the two-step controllability ideas on the planar IP and LIP models of walking, and discuss the ideas in more detail in Chapter 5.

In the remainder of the Chapter 2 we describe the two simple 2D models of walking (IP and LIP). We then discuss several aspects of viable and controllable region concepts, such as possible goals and constraints. Several examples and special cases of these concepts are provided, in application to both the considered simple and more general models. Finally, we describe ways to compute viable and controllable regions for a given model of a walking robot.

2.4 Simple models of walking

For this thesis we consider two simple models of walking: the Inverted Pendulum (IP) [43, 36, 84] and Linear Inverted Pendulum (LIP), [41, 42], both in 2D. We give description of the IP and LIP models in this section; detailed discussion of both models, including their equations of motion, viable and controllable regions, can be found in Chapters 3 and 4 respectively.



(a) IP model in 2D

(b) LIP model in 2D

Figure 2.5: IP and LIP models of walking in 2D. Two simple planar models of a biped are shown. a) The Inverted Pendulum (IP) model has a point mass G and two rigid inextensible massless legs. Collisions are assumed instantaneous and completely inelastic. The system has two phase variables: angle θ and angle rate $\dot{\theta}$ of the stance leg. There are two control values per each step: push-off impulse p along the stance leg just before the collision instant, and step length x_{st} determining time and location of the collision. b) The Linear Inverted Pendulum has a point mass and two massless telescopic legs, with l_{max} being their maximum allowed length. Telescopic actuators in the legs behave such as to maintain constant height y_G of the hip above the ground. Two controls of the model in each step are the step size x_{st} and step time t_{st}. The only dynamic variable is the relative horizontal position of the hip x_h. For both models, we assume the massless swing leg can be moved instantaneously into any position, without influencing the dynamics of the stance leg.

2.4.1 Inverted Pendulum in 2D

The simple IP model in 2D consists of a point mass m at the hip and two rigid inextensible legs with point feet. The legs are assumed to be massless and of the same constant length l. We consider only walking motion of the robot on a flat ground — hence, the constraint that at least one foot is on the ground at any time. The model is shown on Fig. 2.5a.

When the swing leg hits the ground, a collision happens. We assume that collisions are plastic (the colliding foot has zero velocity after the impact), in-

stantaneous, that the hind (former stance) leg leaves the ground immediately after the instant of collision (no double stance), and that there is no impact from the ground on the hind leg. Just before the instant of each collision, a push-off impulse is applied along the stance leg. The amount p of the push-off is chosen by the controller. We use such preemptive push off because it seems to capture some essence of human gait [94] and may be an efficient way to put energy into the system to compensate for collisional loses [43, 69, 77]. Thus, each heel-strike consists of a push-off impulse along the stance leg, followed by a ground reaction impulse along the swing leg. These two impulses cause the hip velocity (hence, energy of the system as well) to be discontinuous at heel-strike, while there is no discontinuity in the configuration of the robot.

The full system has three degrees of freedom: position x of the stance foot on the ground and rotations of the two legs. However, because the legs are massless, motions of the swing leg do not influence dynamics of the stance leg — for the Newtonian dynamics, only the angle θ of the stance leg is a dynamic variable. We assume that the controller can instantaneously move the swing leg into any desired position, thus determining the time and location of the next heel-strike. We characterize the control of the swing leg by the length x_{st} of the next step (the distance between the two feet at heel-strike). Note, that there is no control of the dynamics of the system between collisions — the control only determines the time and location of the next heel-strike and the push-off amount at the next heel-strike. Thus, our IP model is a system with a single dynamic degree of freedom θ, two phase variables (angle θ and angular rate $\dot{\theta}$), and two control parameters per each step: the step size x_{st} and the push-off p.

We also describe the extension of the model to 3D, the 3D IP model, in Ap-

pendix E.

2.4.2 Linear Inverted Pendulum in 2D

The Linear Inverted Pendulum (LIP) model shown on Fig. 2.5b, like the IP model, has two massless legs and a point mass at the hip [41, 40]. But instead of legs being of constant length, they are telescopic with actuators acting so as to keep the center of mass G at a fixed height y_G above the ground. One of the big advantages of having this constraint is that it leads to a linear ODE governing the motion of the system, making it easier to work with. In contrast to discontinuous velocities during IP model collisions, the LIP model experiences no impulses at the instant of collision and velocities are continuous at all times.

As with the IP model, we only consider a single stance walking motion — exactly one foot is in contact with the ground at any instant of time and collisions are instantaneous. Because the swing leg is massless, we assume it can be instantaneously set by the controller to any angle and any length not exceeding the maximum allowed leg length l_{max}. The system has only one dynamic variable, which defines motions of the stance leg and which we choose to be the horizontal position x_h of the hip relative to the stance foot. Two available control parameters per step determine where and when to take a step: the step length x_{st} and step time t_{st} (by which we mean the time from the midstance, i.e. the beginning of the step, to the heel-strike).

Notice, that in the LIP model the compressive force in the stance leg is always proportional to its total length, making the stance leg, in effect, a zero-rest-length compressive spring. This is shown in more detail in Chapter 4.

2.4.3 Poincaré section

We chose the Poincaré section to be at midstance, the point along the robot's trajectory where the potential energy reaches its local maximum. For both the planar IP and LIP models this is also the instant when the stance leg is vertical (for the LIP model, the potential energy of the stance leg, acting as a compressive spring, is maximum when the leg's length is minimum).

Both the 2D IP and LIP models have only one essential dynamic variable, resulting in a 2-dimensional phase space. Therefore, the Poincaré section is one-dimensional and can be described only by the (horizontal) speed v of the CoM (the hip) at midstance. In our viability and controllability analysis of the IP and LIP models, we will be looking at how the speed of the robot changes from one midstance to the other.

2.5 Special cases, examples, and applications

We proceed to discuss in more detail different aspects of viable and controllable regions. These include constraints, goals, and walking controllers for a robot. We will provide different examples and suggest several possible applications of viable and controllable regions. One of the applications is design of robust walking controllers, which we describe in Chapter 6.

2.5.1 Constraints

We want the biped to never fail, i.e. never fall down and always satisfy all constraints imposed on it. For the IP model, we define falling down as when the hip passes through the ground. For the LIP model the hip never changes height, so another definition of failing is needed. Instead, we say the LIP model fails if the length l of the stance leg exceeds its prescribed maximum allowed value l_{max}. For both models we also require that the robot reaches midstance during each step.

Besides these basic requirements, we sometimes impose additional constraints on the model which have to be satisfied during each step of walking. Such constraints can be used to represent restrictions and features of the physical robot, its complicated model, or the environment. Examples include (both for our simple planar models and more general bipeds):

(i) The duration of a step must be at least $t_{\min}$ (a fixed positive number). This constraint aims to make sure that there is enough time for the actuators to move the swing leg to a desired position. It can serve as a proxy for the swing-leg actuator limitations, which are necessarily present in all physical robots.

(ii) The stance leg actuator effort (e.g. push-off impulse for the IP model and telescopic actuator force for the LIP model) must not exceed a fixed limit. This represents either actuator limitations or may be used to reduce energetic costs of the system.

(iii) The foot placements must always be along a given curve or line on the ground, or inside given 'stepping stones' (sequence of given points or

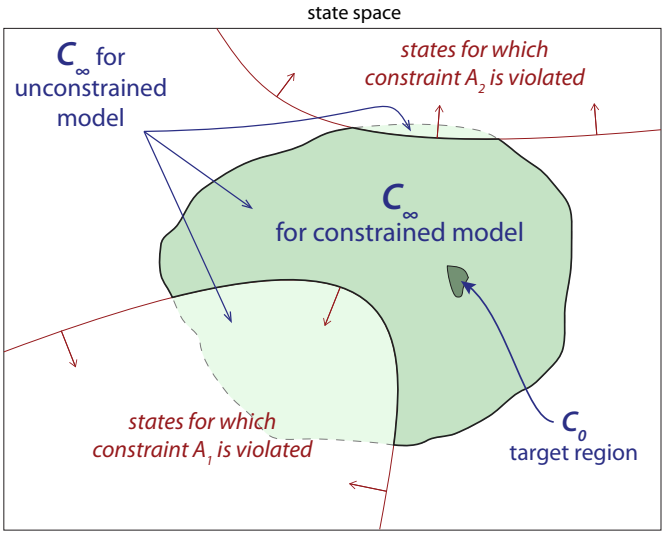


Figure 2.6: Influence of model constraints on viability and controllability. Additional constraints imposed on the model reduce the size and change the shape of viable and controllable regions. We illustrate this on the C_∞ region, but the reasoning is the same for all viable and controllable regions. The boundary of C_∞ for a given (unconstrained) model and a fixed target is shown by a dashed line (which coincides with the solid line on the sides). Each introduced constraint may cut off a part of C_∞, as the corresponding states cease to be ∞-step controllable. What remains is the C_∞ region for the constrained model, shown by the darkest shaded region. Changes to the viable and controllable regions induced by individual constraints may help understanding importance of these constraints for robustness of the model.

small regions on the ground). This constraint can be used for certain navigation tasks, such as following a given path or obstacle avoidance.

(iv) The speed and/or direction of motion of the CoM must be within fixed bounds. This requirement aims to prevent control actions from significantly changing the global motion of the robot (e.g. when it is desired to move at a certain speed).

The addition of one or more constraints to the model may reduce the size (and change the shape) of the viable and controllable regions. We illustrate this on Fig. 2.6. For example, consider the ∞-step controllable region C_∞ for a given model and a fixed target C_0. For any state inside C_∞, there is at least one trajectory that starts from this state and eventually reaches (or asymptotically ap-

proaches) the target. However, with new constraints introduced, some states in C_∞ may cease to be ∞-step controllable in that all trajectories going from them to the target violate one of the new constraints. Such states are not in the C_∞ region anymore. Thus, the ∞-step controllable region for the constrained model (C_∞^c) is a subset of that for the unconstrained model (C_∞): $C_\infty^c \subset C_\infty$. Similar logic can be applied to all viable and controllable regions: $C_n^c \subset C_n,\ V_n^c \subset V_n,\ V_\infty^c \subset V_\infty$. We conclude that each additional constraint introduced to the model may 'cut off' parts of the viable and controllable regions. More insights about the influence of constraints can be found in Section 2.7.5 on page 61.

Changes in viable and controllable regions due to individual constraints can help understanding the nature of these constraints and their importance for robustness. For example, if the constraint (i) from the list above causes a 'significant' reduction of C_∞ (e.g. as constraint A_1 does compared to constraint A_2 on Fig. 2.6), then we might want to 'weaken' this constraint if possible — e.g. by using stronger leg swinging actuators or improving control algorithms to allow more time to swing the leg.

2.5.2 Goals

As opposed to constraints which have to be met at each step, the goal (target) region C_0 represents a region at Poincaré section (at midstance) that we desire to reach in one or more steps. The region C_0 can be a single point (i.e. a unique target state), several points, a connected region, or several disjoint regions at the Poincaré section.

Pratt et al. assume the target to be all captured states (states in which the

biped can stand still — balanced, say, with ankle torques) [64, 42, 63]. Their n-step viable-capture basins (described in Section 1.2 here) are essentially the n-step controllable regions corresponding to their target. In this thesis we consider a more general case and allow the target C_0 to be any region we choose, which lies entirely in the viability kernel V_∞. Note, that if a target state is outside of V_∞, then reaching such state is always followed by a failure of the robot, either immediately or after a finite number of steps. This can, however, be acceptable for certain tasks, such as a goalkeeper in a soccer game saving the goal from a shot, which is often followed by the goalkeeper falling down.

Examples of targets, which aim to serve different tasks of locomotion and may be considered either in isolation or in combination, include:

(i) The hip (or the center of mass, CoM) is motionless and above the stance foot (e.g. the stance leg achieves a vertical position). This would be a special case of Pratt's captured states and can be used to bring the robot to a full stop.

(ii) The CoM is above the stance foot and has a specified speed, direction of motion, or both. This goal represents a 'stable' motion of the robot with a desired speed and/or a desired direction. Designing a controller which tries to meet this goal at every midstance may be a way to stabilize a desired motion of the robot.

(iii) A state along an optimal periodic trajectory — e.g. the state where the optimal trajectory crosses the Poincaré section of the model. The optimality can be with respect to any desired characteristics of the motion, such as energy use. As a special case of the example (ii) above, an optimal trajectory could be stabilized by taking, at each step, the target to be a point on

the optimal trajectory. We use such approach in the design of our walking controller in Chapter 6, where we set the target to be a point along a trajectory with low energy cost.

(iv) The projection of the CoM on the ground, or the stance foot, is in a specified point or region on the ground. This goal aims to move the robot to a desired location on the ground. A sequence of such goals ('stepping stones') can be used to navigate the robot along a desired path or through a rough terrain. A restricted area of stepping locations can also be used for obstacle avoidance.

2.5.3 Control strategies

Thus far, we assumed no specific controller of the robot: we looked at the states from which it is *physically* possible to avoid a failure or reach a desired target for a given model, given actuator limitations and other imposed constraints. Now we consider viability and controllability concepts for a particular control design.

Wieber notes in [92], that if a specific controller is used, then the biped avoids a failure only from a controller-specific subset of the viability kernel V_∞, and calls this subset the *invariant set* of the given controller. For the states in V_∞, but outside of the invariant set, the robot fails, although a failure could be avoided if a different controller was used. Similarly, for a specific controller and a given target, the set of all states from which the robot reaches (or asymptotically approaches) the target is the basin of attraction of the dynamical system. This basin of attraction is a subset of the ∞-step controllable region C_∞ of the model. For the states which are in C_∞, but outside of the basin of attraction, the robot

never reaches the target with the considered controller, but is able to do so if a different feasible controller is employed.

Generally, we would like controllers to have possibly larger basins of attraction and invariant sets. For example, if the invariant set of a controller u_1 is a strict subset of the invariant set of a controller u_2, then this suggests using the controller u_2 from the point of view of viability (with u_2, the robot avoids a failure in some situations, where it fails if u_1 is used). Note, however, that u_1 may still be preferable over u_2 by a different criteria, e.g. energy use of the robot. The most 'superior' controller (i.e. the controller with the biggest possible invariant set or basin of attraction), thus, is the one whose invariant set fully fills up the viability kernel (or the basin of attraction fully fills up the ∞-step controllable region C_∞). However, practical realization of such an ultimate controller may be impractical due to computational complexity.

One way to design a controller, whose performance is close to that of the ultimate controller, was suggested by Wieber [92, 93]. He proposed having several controllers u_i, such that their respective invariant sets, each a subset of V_∞, cover different parts of V_∞ — that is, each u_i is designed to work best in a different situation. Then imagine a high-level controller that switches between u_i, depending on the current state of the robot. The invariant set of this high-level controller is the union of the invariant sets of all u_i, and fills up most of V_∞.

A disadvantage of this approach is the complexity of the design process, the structure of the resulting controller, or both. Our goal is to design a controller, whose basin of attraction for a given target is as close as reasonably possible to the entire C_∞, but which nevertheless maintains a simple structure and is not computationally expensive. In Chapter 6 we design such controller for a multi-

degree-of-freedom robot.

2.5.4 Multi-DOF models

The controllability and viability concepts discussed in this thesis are defined for a general model of a biped, no matter how many degrees of freedom (DOF) or actuators the model has. However, for a complex model V_n and C_n are regions in a many-dimensional state space. So, the viability and controllability analysis may be too challenging, either computationally or analytically (for example, see Section 2.7 for methods to numerically compute the regions V_n and C_n, and related computational issues).

Whenever possible, we would like to avoid the computational burden of complex models. One way to do this is by using a simple model, such as the IP or LIP, to approximate the behavior of a given many-DOF robot. Encouraged by private discussions with Anoop Grewal and Andy Ruina[2] [29] we believe that, during normal walking, the motion of the center of mass (CoM) of any biped is 'close' to that of a point-mass model. Possible exceptions include extreme situations, such as recovery when no stepping is allowed, or special types of motion, e.g. tight rope walking. Therefore, one could design a high-level walking controller (i.e. the CoM motion controller), using a point-mass model as a proxy for the robot. Following this approach, in Chapter 6 we use the planar IP model (introduced in Section 2.4.1 and described in detail in Chapter 3) to design a walking controller for the four-link, six-DOF robot. A more detailed discussion on simple-model approximation of complex robots can be found in Grewal's work [29].

[2] Biorobotics and Locomotion Lab, Cornell University.

2.6 Extended viable and controllable regions

The viable (V_n) and controllable (C_n) regions show the states, starting from which the robot can avoid a failure or reach a specific goal, if appropriate feasible controls are used. However, these regions do not explicitly show which controls allow the robot to accomplish the set goals. Here we extend the concepts of viable and controllable regions to include controls available to the robot.

Suppose we are interested in feasible controls, which allow to not fail or to reach a given target. These can be either full controls of the robot (i.e. control laws for all actuators available on the robot) or only partial controls (i.e. for only some actuators). For example, for the planar IP model we may be interested in step sizes, which allow the robot to not fail with appropriate push-offs, but not interested in the corresponding push-offs. We call the space of all controls of interest the *control space* U. For simplicity, we will only consider controls during the next step, i.e. before the robot reaches the next midstance or fails. Thus, for the IP model, which has two control values per step (the step size x_{st} and push-off p), the control space U can be one- or two-dimensional. For any model we assume the control space to be finite-dimensional — that is, the set of all possible control actions in a single step can be described by a finite number of parameters.

Next, consider the Cartesian product of the Poincaré section S and the control space U. We call this product the *extended Poincaré section* $\bar{S}$ and any point in $\bar{S}$ an *extended state* of the robot. Each extended state is a pair (q, u) of the state $q \in S$ of the biped at midstance and the vector of controls $u \in U$.

Similar to the viable and controllable regions in the Poincaré section, we

define the *extended viable and controllable regions* as parts of the extended Poincaré section.

2.6.1 Extended viable regions

For a given model, constraints, and control space U, an extended state $\bar{q} = (q, u)$ is 0-*step viable*, if the state q is 0-step viable ($q \in V_0$) and controls u are feasible in the state q. The *extended* 0-*step viable region* $\bar{V}_0$ is the collection of all 0-step viable extended states — it is all possible combinations of states allowed for the robot at midstance (including those, for which the robot is about to fall) and all controls allowed for the next step.

Next, we call an extended state $\bar{q} = (q, u)$ *n-step viable*, if $q \in V_n$ and the robot can start from q, use controls $u \in U$ during the first step, and reach an $(n-1)$-step viable state at the end of the first step. The set of all n-step viable extended states is the *extended n-step viable region* $\bar{V}_n$. It is all combinations of states q at midstance and controls u, such that the biped can start from q, and take at least n steps without failing and using controls u during the first step. The regions $\bar{V}_n$ are shown schematically on Fig. 2.7.

Similar to the viable regions V_n, the sequence of extended viable regions $\{\bar{V}_n\}$ is nested: $\bar{V}_n \subset \bar{V}_{n-1}$. We call the limit of this sequence the *extended* ∞-*step viable region*, or the *extended viability kernel*, $\bar{V}_\infty$. For any point (q, u) in $\bar{V}_\infty$, the biped can take infinitely many steps without failing, if it starts from the corresponding state q at midstance and uses the corresponding control parameters u during the first step. The extended viability kernel, thus, shows which controls can be used at each step, so that the robot never fails.

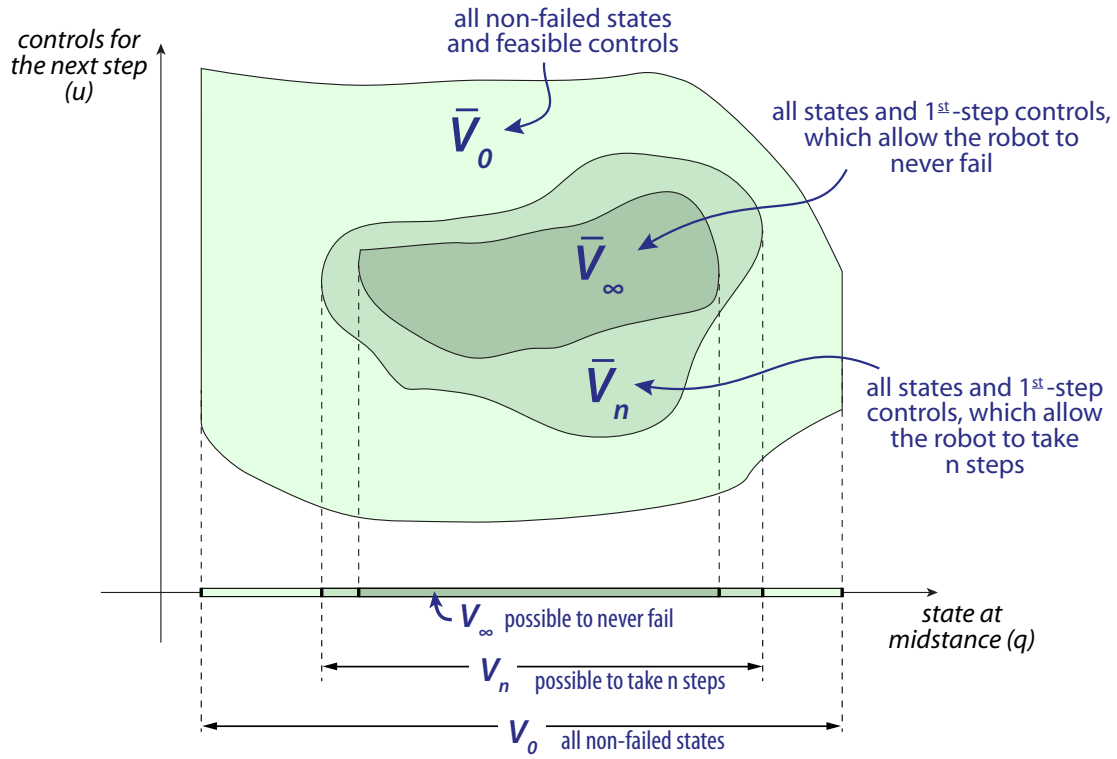


Figure 2.7: Schematics of extended viable regions. Extended n-step viable regions $\bar{V}_n$ are areas in the space formed by both the state and control variables of the model. Only states at midstance and controls of the next step are considered. $\bar{V}_0$ is all combinations of non-failed states and feasible controls. The region $\bar{V}_n$ includes all pairs of initial states and corresponding controls of the next step, which allow the robot to take (at least) n step without failing. For the combinations of states and corresponding controls in the extended viability kernel $\bar{V}_\infty$, the robot can step indefinitely. Note, the projections of the extended n-step viable regions $\bar{V}_n$ onto the state space are the n-step viable regions V_n (see Fig. 2.1 and Section 2.1). The regions $\bar{V}_n$ are extensions of the viable regions into the control space of the model.

One may use $\bar{V}_\infty$ to design a robust walking controller. For example, for each state of the robot one may choose controls which are possibly farther from the boundaries of $\bar{V}_\infty$. Therefore, it is less likely for a disturbance to move the system into a state outside of the viability kernel, where it would not be able to avoid failing.

We will also sometimes use the notion of viable controls. We say that for a given state q at midstance controls $u \in U$ are *n-step viable*, if the corresponding

extended state (q, u) is n-step viable (here n can be zero, a positive integer, or infinity).

2.6.2 Strict extended controllable regions

Before proceeding to define the extended controllable regions, we first introduce the strict extended controllable regions $\hat{\hat{C}}_n$. We only use the regions $\hat{\hat{C}}_n$ to define the extended controllable regions in Section 2.6.3 below and to numerically compute the extended viable and controllable regions in Sections 2.7.3 and 2.7.4.

For a given model, control space U, and a given target region C_0, we say that an extended state $\bar{q} = (q, u)$ is *strict n-step controllable* (for $n \geq 1$), if the state q is strict n-step controllable ($q \in \hat{C}_n$) and the robot can start from q, use controls $u \in U$ during the first step, and reach a strict $(n-1)$-step controllable state (i.e. reach the region $\hat{C}_{n-1}$) at the next midstance. The set of all strict n-step controllable extended states is the *strict extended n-step controllable region* $\hat{\hat{C}}_n$. For any point (q, u) in $\hat{\hat{C}}_n$, if the robot is initially in the state q and controls u are used during the first step, then the robot is able to reach the target C_0 taking exactly n steps.

A set-valued map $\hat{\hat{C}}$, which we call the *strict extended controllability map* (or simply the *extended controllability map*), describes the regions $\hat{\hat{C}}_n$:

$$\hat{\hat{C}} : 2^{V_0} \to 2^{\bar{V}_0}, \qquad \text{s.t. } \hat{\hat{C}}(C_0) = \hat{\hat{C}}_1.^3 \tag{2.11}$$

For any target region C_0, the map $\hat{\hat{C}}$ returns the corresponding strict extended 1-step controllable region $\hat{\hat{C}}_1$. Consequently, the region $\hat{\hat{C}}_n$ is expressed as all

[3] 2^X is the power set of a given region X, i.e. the set of all subsets of X.

initial states and first-step controls, for which the region $\hat{C}_{n-1}$ in the state space can be reached in one step:

$$\hat{\bar{C}}_n = \hat{\bar{C}}(\hat{C}_{n-1}), \qquad \text{for } n \geq 1, \tag{2.12}$$

where we assume $\hat{C}_0 = C_0$. The limit of the sequence $\{\hat{\bar{C}}_n\}$ is the *strict extended* ∞*-step controllable region* $\hat{\bar{C}}_\infty$. It is all extended states (q, u) which allow the robot to asymptotically reach the target region C_0.

2.6.3 Extended controllable regions

We now define the extended controllable regions $\bar{C}_n$ for a given model of the robot, target region C_0 and control space U. Similar to the extended viable regions, the regions $\bar{C}_n$ extend the concept of the controllable regions C_n into the control space of the robot.

First, we say that the *extended* 1*-step controllable region* $\bar{C}_1$ is equivalent to the strict extended 1-step controllable region $\hat{\bar{C}}_1$ defined in Section 2.6.2 above. It is all combinations of states at midstance and controls of the first step, for which the robot is able to reach a target state in C_0 at the end of the first step and not fail. Thus, from equation (2.11),

$$\bar{C}_1 = \hat{\bar{C}}_1 = \hat{\bar{C}}(C_0). \tag{2.13}$$

Any point (q, u) in the region $\bar{C}_1$ is a 1*-step controllable* extended state of the robot.

Next, for $n \geq 2$, we call an extended state (q, u) of the robot *n-step controllable*, if there is a way for the robot to reach the target C_0 in n or fewer steps, starting from the state q at midstance and using the controls $u \in U$ during the first step. The

extended n-step controllable region $\bar{C}_n$ is the set of all n-step controllable extended states of the robot. The extended regions $\bar{C}_n$ are shown schematically on Fig. 2.8.

Another way to describe $\bar{C}_n$ is all extended states, for which the $(n-1)$-step controllable region can be reached in one step — that is, $\bar{C}_n$ is the strict extended 1-step controllable region corresponding to the target C_{n-1}:

$$\bar{C}_n = \hat{\bar{C}}(C_{n-1}). \tag{2.14}$$

See Appendix B on page 177 for a formal proof of equation (2.14). Formula (2.14) is useful for iterative numerical computation of the regions $\bar{C}_n$, as explained in Section 2.7.4.

The sequence $\{\bar{C}_n\}$ of the extended controllable regions is nested: $\bar{C}_{n-1} \subset \bar{C}_n$. We call the limit of this sequence the *extended* ∞*-step controllable region* $\bar{C}_\infty$. It represents all extended states (q, u), such that if the robot starts in q and uses controls u during the first step, it is able to reach the target eventually or approach it asymptotically. Alternatively, $\bar{C}_\infty$ is also all extended states (q, u), such that the state q is in C_∞, and if the robot starts from q and uses controls u during the first step, then the state of the robot at the end of the first step is again in C_∞. Hence, the invariance property of the region $\bar{C}_\infty$.

Both the controllable regions C_n and the extended controllable regions $\bar{C}_n$ show whether it is possible for the robot to return to the target after a disturbance. The regions $\bar{C}_n$ also show all feasible controls which can be used to return to the target, when it is possible to do so. In Chapter 6 we employ the extended controllable regions to design a walking controller. For each state of the robot we consider all corresponding controls inside $\bar{C}_\infty$ and choose those, which are optimal in a certain way (e.g. with respect to stability, robustness, or efficiency).

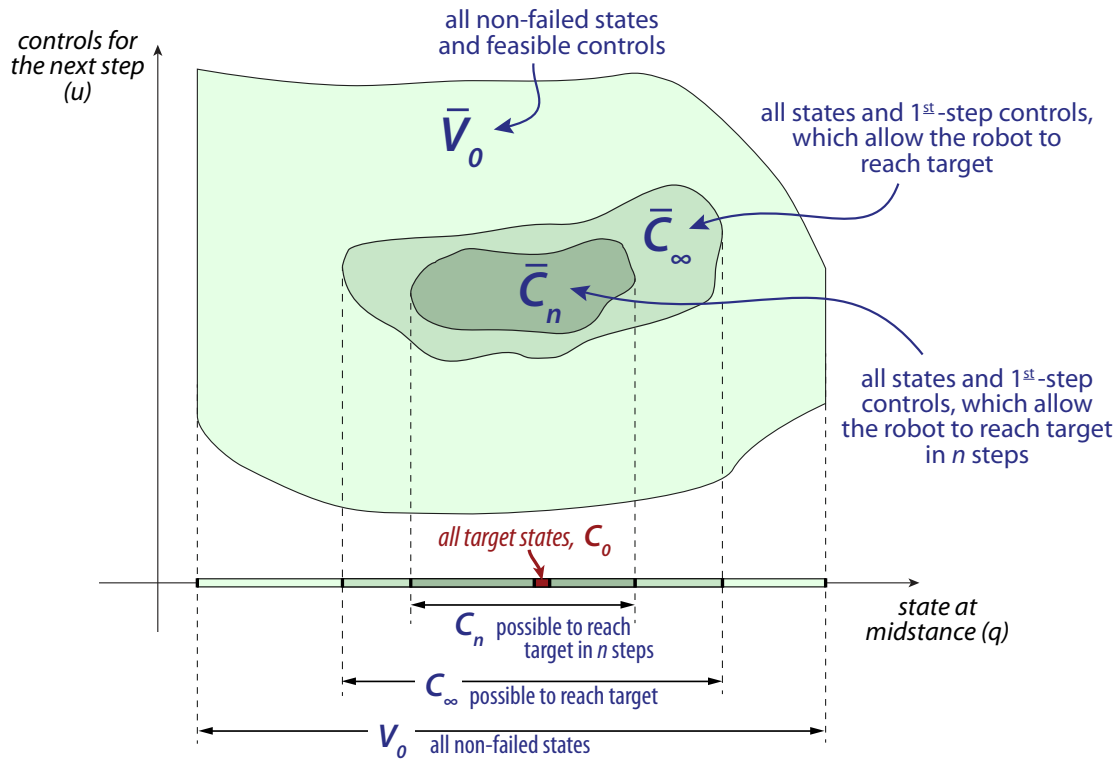


Figure 2.8: Schematics of extended controllable regions. Extended n-step controllable regions are areas in the space formed by both the state and control variables of the model. Only states at midstance and controls of the next step are considered. For a given set of target states C_0, the extended n-step controllable region $\bar{C}_n$ ($n \geq 1$) is all combinations of states and corresponding controls, such that the robot can reach a target state in n or fewer steps. The region $\bar{C}_\infty$, the limit of the sequence of nested regions $\{\bar{C}_n\}$, represents all combinations of states and next-step controls, which allow to eventually reach the target or approach it asymptotically. Projections of the extended controllable regions $\bar{C}_n$ onto the state space of the model are the controllable regions C_n (see Fig. 2.2 and Section 2.2).

We may sometimes use detailed notation of the extended viable and controllable regions, to explicitly show which control parameters of the model are considered in the control space U. For example, if for the planar IP model we are only interested in the step sizes x_{st}, then we denote the corresponding n-step viable and controllable regions by $\bar{V}_n^{x_{st}}$ and $\bar{C}_n^{x_{st}}$. However, if both the step size x_{st} and push-off p form the control space U, then we write $\bar{V}_n^{p,x_{st}}$ and $\bar{C}_n^{p,x_{st}}$ respectively.

2.6.4 Parameter extension

In the Sections 2.6.1 and 2.6.3 above, we have extended the state space of the robot by one or more control axes in order to find controls, which are 'good' for the model in a given state. One may also consider extending the state space by dimensions of other (non-control) parameters of the model or the environment, such as the ground slope, inertia of the legs, gravity, etc. Application of the viability and controllability concepts to this new space (in a way, analogous to the extended viable regions in Section 2.6.1 and extended controllable regions in Section 2.6.3) may help understanding the role of the corresponding parameters in the model. For example, consider an extended state of the model (q, γ), where q is the state of the robot at midstance and γ the ground slope. Construction of the corresponding extended viable $\bar{V}_n^\gamma$ and controllable $\bar{C}_n^\gamma$ regions can show which slope angles are fatal for the robot (i.e. outside of the extended viability kernel — no feasible control can prevent the robot from failure in one or more steps) and for which slope angles a given target can be reached (e.g. on which slopes the robot is able to maintain a desired velocity).

2.6.5 Relation to capture regions

The controllable and extended-controllable regions generalize Pratt's [64, 42] viable-capture basins and capture regions, respectively (see Section 1.2). The *n-step viable-capture basin* is the set of all states of the robot from which the robot can come to a stop by taking n or fewer steps. Thus, for a given model, Pratt's n-step viable-capture basin is equivalent to our n-step controllable region C_n, when our goal is standing still.

For a given initial state of the robot, the *n-step capture region* includes all points on the ground where, for the present step, the robot can step and come to a stop within at most $n-1$ subsequent steps (i.e. within n steps total). Coordinates of the next stepping location (two values for 3D robots, one value for 2D robots) are high-level control parameters of the model. For example, in our planar IP and LIP models, such a control parameter is defined as the length of the first step x_{st}. So, for a given initial state q_0, the n-step capture region represents all controls (first footsteps) which allow the robot to stop within n steps in total. On the other hand, consider the extended n-step controllable region $\bar{C}_n$, extended by inclusion of the footstep control and corresponding to the goal of standing still. $\bar{C}_n$ is all combinations of initial states and first-step controls which allow the robot to stop within n steps. Therefore, the n-step capture region is equivalent to the slice of the extended n-step controllable region along a specific initial state q_0.

Thus, our concept of extended controllable regions generalizes Pratt's capture regions in three different ways. First, controllable regions allow us to consider an arbitrary goal (target region in state space) of locomotion — as opposed to only the goal of standing still in capture regions. Second, extended controllable regions study all control parameters of interest of the model — as opposed to only footstep controls in capture regions. For example, in Chapter 3 we find both step-size and push-off controls of the planar IP model for various targets. Third, extended controllable regions show both initial states and corresponding controls that let the robot reach the target — as opposed to the assumption of a specific initial state in capture regions.

That is, capture regions consider controls from a particular subset of all con-

trol variables, to reach a specific target from a specific initial state. We generalize to consider all initial states, all possible targets, and all control variables. This difference reflects a slight difference in philosophy. While Pratt's goal is for practical on-the-fly model-predictive control, ours is understanding limits of control robustness.

2.7 Methods

We now describe methods we use to numerically compute the viable and controllable regions V_n and C_n, and the extended viable and controllable regions $\bar{V}_n$ and $\bar{C}_n$. In principal, the computational procedures presented below can be applied to any well-specified model of a biped. For all calculations below we use discretization of the state space. The computational requirements can be challenging even for simple models. Analytical and/or other insights are used, whenever possible, to reduce the computational work.

2.7.1 Controllable regions

We first describe how we compute the controllable regions C_n for a given model, constraints, and the target region C_0.

Assume that we have a way of computing the 1-step controllable region $C(q)$ corresponding to an arbitrary target point q in the state space. For some simple models, such as the planar IP and (2D or 3D) LIP models, this computation can be done analytically (see Sections 3.2.2 and 4.2.1). For more complex models, we develop a numerical procedure, which utilizes a massive brute-force simulation

(together with analytical and intuitive insights, whenever possible). This procedure approximately finds all initial states from which the robot can reach the state q within one step. See Section 3.2.5 for the description of such procedure for the planar IP model, constrained by a minimum time required to place the swing leg into a desired position.

The target region C_0 is a set of target points q. Obviously, then, the 1-step controllable region C_1, corresponding to the target C_0, is already found if there is only a single target state in C_0 (i.e. if $C_0 = \{q\}$). If there is more than one point q in C_0, then the region C_1 can be found by taking the union of all 1-step controllable regions corresponding to different individual target states in C_0:

$$C_1 = C(C_0) = \bigcup_{q \in C_0} C(q). \tag{2.15}$$

The union in the right-hand side of the above equation (as well as equations (2.17) and (2.19) below) can be calculated numerically, by discretizing the state space of the model. One example of such calculation can be found in Section 3.2.3 for the planar IP model.

Next, we proceed to find the n-step controllable regions C_n for $n > 1$. Recall from Section 2.2, that the region C_n can be expressed as the 1-step controllable region corresponding to the target C_{n-1}. That is, repeating equation (2.2),

$$C_n = C(C_{n-1}). \tag{2.16}$$

Similar to equation (2.15), the right-hand side of equation (2.16) above can be written as the union of all 1-step controllable regions, corresponding to individual states in C_{n-1}:

$$C_n = C(C_{n-1}) = \bigcup_{q \in C_{n-1}} C(q), \qquad \text{for } n \geq 1. \tag{2.17}$$

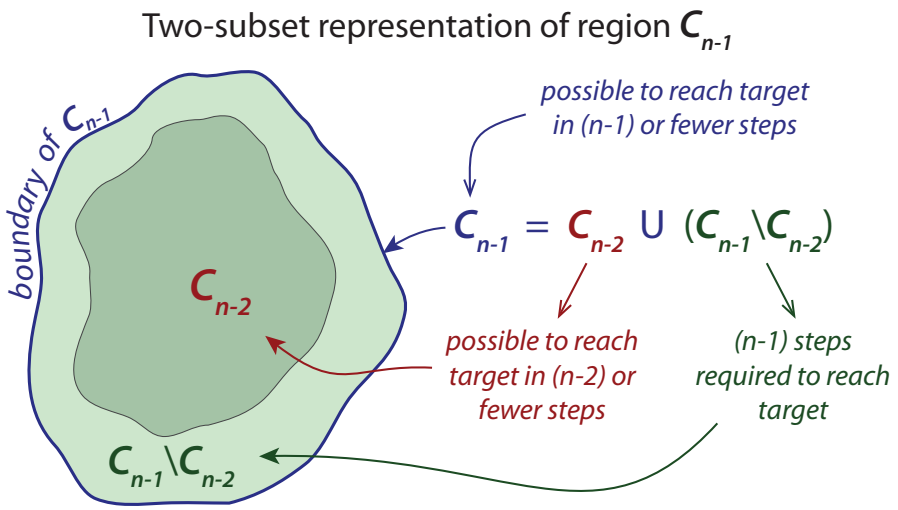


Figure 2.9: Representation of C_{n-1} as two subsets. This figure illustrates equation (2.18) in the text, an intermediate step in the procedure of computing the n-step controllable regions C_n for a given target C_0. The region C_{n-1} (all initial states which allow to reach the target in $n-1$ or fewer steps) is split into two subsets: the $(n-2)$-step controllable region C_{n-2} (possible to reach the target in $n-2$ or fewer steps) and all other points in C_{n-1} (i.e. the region $C_{n-1} \setminus C_{n-2}$ — $n-1$ steps are required to reach the target).

Thus, according to the above formula, the n-step controllable region C_n for a given target C_0 can be found, based on the $(n-1)$-step controllable region C_{n-1} corresponding to the same target and the knowledge of (or the ability to compute) the 1-step controllable region $C(q)$ for an arbitrary target state q. Note, that for $n = 1$, equation (2.17) above agrees with expression (2.15) for computing the 1-step controllable region C_1.

The amount of computation in a single step of (2.17) (that is, evaluation of $C(C_{n-1})$ once) can be reduced with the following observation. Recall from Section 2.2, that the controllable regions are nested, and $C_{n-2} \subset C_{n-1}$. Let us break the region C_{n-1} into two subsets, as shown on Fig. 2.9: the $(n-2)$-step controllable region C_{n-2} and the rest of C_{n-1}:

$$C_{n-1} = C_{n-2} \cup (C_{n-1} \setminus C_{n-2}). \tag{2.18}$$

We then can write

$$C(C_{n-1}) = C\left(C_{n-2} \cup (C_{n-1} \setminus C_{n-2})\right) =$$
$$= C(C_{n-2}) \cup C(C_{n-1} \setminus C_{n-2}).$$

The controllable region $C(C_{n-1} \setminus C_{n-2})$ in the right-hand side above can be expressed as the union of all 1-step controllable regions $C(q)$ corresponding to individual target states q in $C_{n-1} \setminus C_{n-2}$:

$$C(C_{n-1}) = C(C_{n-2}) \cup \left(\bigcup_{q \in C_{n-1} \setminus C_{n-2}} C(q) \right).$$

Also notice, that the term $C(C_{n-2})$ above is exactly the $(n-1)$-step controllable region C_{n-1}, according to (2.17). Therefore, equation (2.17) becomes

$$C_n = C_{n-1} \cup \left(\bigcup_{q \in C_{n-1} \setminus C_{n-2}} C(q) \right), \qquad \text{for } n \geq 2. \tag{2.19}$$

Formula (2.19) provides a way to iteratively compute the n-step controllable regions $C_n = C^n(C_0)$ for a given target C_0. At each iteration, a new set of points is added to the previously found region C_{n-1} in order to get C_n. This new set is the 1-step controllable region, corresponding to the difference $C_{n-1} \setminus C_{n-2}$ between the regions C_{n-1} and C_{n-2}. Compared to (2.17), equation (2.19) is more computationally efficient, because it only computes new points in C_n — as opposed to computation of the entire C_n at each iteration of (2.17). Iteration step (2.19) is schematically illustrated on Fig. 2.10.

We do not have a universal method to find the ∞-step controllable region C_∞ accurately. However, we suggest the following way of approximating C_∞. The n-step controllable regions C_n approach the region C_∞ when n goes to infinity. Assuming C_∞ is bounded, the difference $C_n \setminus C_{n-1}$ between two subsequent controllable regions decreases, as n increases. We then hypothesize that for most

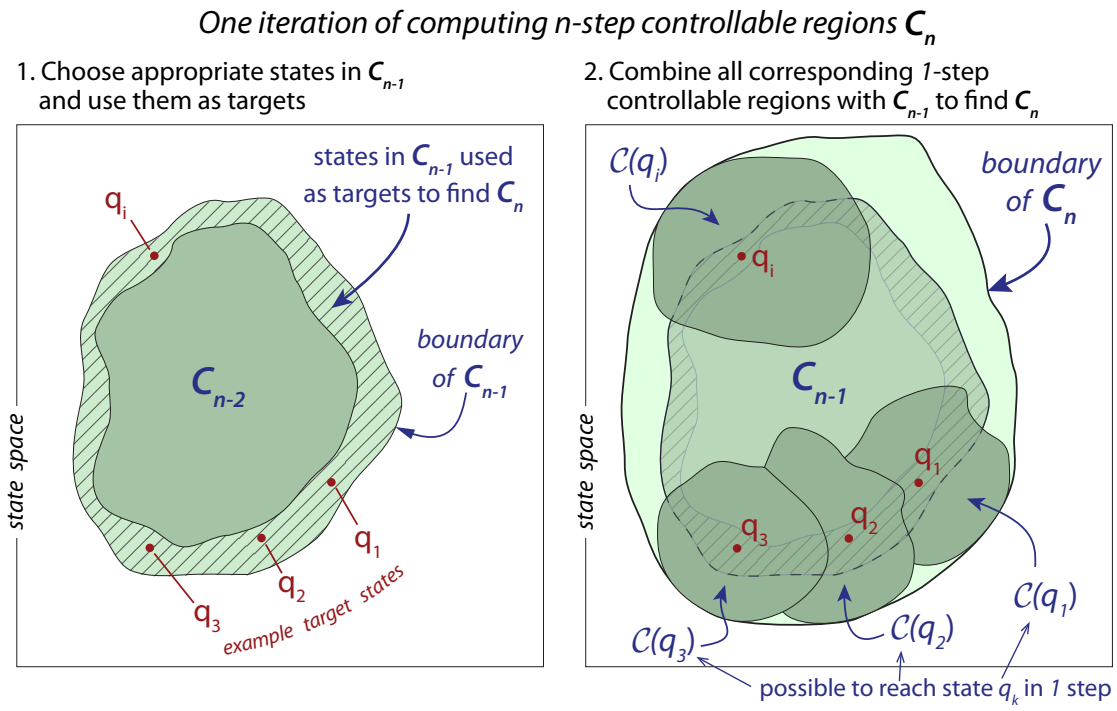


Figure 2.10: Illustration of the Cn computation procedure. The figure schematically illustrates one iteration step of the procedure we use to compute the n-step controllable regions C_n for a given target C_0. At each iteration, the region C_n is computed based on the previously found regions C_{n-1} and C_{n-2}. First, select all states in C_{n-1}, which are outside of C_{n-2}. For each selected state q_i find the corresponding 1-step controllable region $C(q_i)$, i.e. all states at midstance, from which the robot can reach the state q_i by making at most one step. Then, take the union of all such regions $C(q_i)$ and the region C_{n-1}. The result is the n-step controllable region C_n.

robots, it is sufficient to find only several regions C_n in order to approximate the ∞-step controllable region C_∞. For example, one could use the region C_5 as an approximation of C_∞, if there is 'small' difference between C_5 and C_4. Notice also, that if the difference $C_n \setminus C_{n-1}$ decreases with each n, then so does the amount of computational work required to find each C_n, according to (2.19).

On the other hand, for certain models (e.g. for which C_∞ is unbounded) the evolution of the n-step controllable regions may provide one with 'intuition' about the region C_∞, which can then be used to 'guess' the shape of C_∞. One example of such model is the simple LIP model, discussed in Chapter 4, which

can recover from any infinitely large velocity, assuming the actuators can swing the swing leg infinitely fast.

We now summarize the iterative procedure of computing the n-step controllable regions $C_n = C^n(C_0)$ for a given model, constraints, and the target region C_0:

(I). Develop a way to compute the controllability map $C(q)$ for an arbitrary target state q. This can be done either analytically (e.g. by defining analytical equations of the boundaries of $C(q)$), or numerically (say, by developing a numerical procedure, which approximates $C(q)$ for a given input q).

(II). Using results of the step (I), compute the 1-step controllable region C_1 corresponding to the given target C_0. If there is more than one target state in C_0, equation (2.15) should be employed.

(III). Iteratively compute as many of the n-step controllable regions C_n as desired, for $n = 2, 3, \ldots$. At each iteration, the region C_n is found according to formula (2.19), using the knowledge of the two previous regions, C_{n-1} and C_{n-2}, as well as the results of the step (I). This iteration step is illustrated on Fig. 2.10.

(IV). If, after several iterations in the step (III), each next region C_n does not significantly differ from the previous region C_{n-1}, use the last computed region C_n as an approximation of the ∞-step controllable region $C_\infty = C^\infty(C_0)$. If, however, the regions C_n do not seem to approach any specific area, use the knowledge about the evolution of the regions C_n to increase your intuition about the region C_∞.

The above algorithm allows us to find the controllable regions C_n for a given target C_0. However, if one also wishes to compute the extended controllable regions $\bar{C}_n$, the algorithm described in Section 2.7.4 should be used instead, which allows us to compute both the controllable and extended controllable regions C_n and $\bar{C}_n$.

2.7.2 Viable regions

We now describe an iterative procedure to compute the n-step viable regions V_n for a given robot. The procedure is similar to the one we used to find the n-step controllable regions C_n in Section 2.7.1. We assume that the 0-step viable region V_0 — all states allowed for the robot at midstance — is either given or can be found based on the given set of constraints.

Recall from Section 2.1, that the region V_n can be described as all states at midstance, starting from which the robot is able, with appropriate feasible controls, to take exactly one step and reach a state in the region V_{n-1}. It is convenient then to use the concept of the strict controllable regions, described in Section 2.2.1: V_n is the strict 1-step controllable region corresponding to the target V_{n-1}:[4]

$$V_n = \hat{C}(V_{n-1}) = \bigcup_{q \in V_{n-1}} \hat{C}(q), \qquad \text{for } n \geq 1. \tag{2.20}$$

Similar to equation (2.15) on page 50 for the 1-step controllable region $C(C_0)$, we represented here the strict 1-step controllable region $\hat{C}(V_{n-1})$ as the union of all

[4] In contrast to equation (2.16) for the controllable regions, the n-step viable region V_n is generally not equal to the ('non-strict') 1-step controllable region $C(V_{n-1})$ for the target V_{n-1}. This is because any controllable region includes all target states, i.e. $V_{n-1} \subset C(V_{n-1})$. On the other hand, the viable regions form a shrinking nested sequence, and $V_n \subset V_{n-1}$. That is, V_n generally does not include the entire region V_{n-1} and, thus, cannot be equal to $C(V_{n-1})$. Hence, the strict controllable regions are used here, instead of the controllable regions.

strict 1-step controllable regions $\hat{C}(q)$ corresponding to individual target states q in V_{n-1}.

Equation (2.20) provides a way to sequentially calculate the n-step viable regions V_n, starting from V_1. At each step, the region V_n is found based on the previously computed region V_{n-1}. Here we assume, that we have a way to find, either analytically or numerically, the strict 1-step controllable region $\hat{C}(q)$ for an arbitrary target state q. Similar to the step (IV) of finding the ∞-step controllable region C_∞ (page 54), the ∞-step viable region V_∞ can be approximated by one of the regions V_n for large n, when V_n converge to a fixed area in the state space. Other analytical and numerical insights can be used to find the viability kernel V_∞ — e.g. see Section 3.2.4 on page 84 where we find the viable regions of the planar IP model of a biped.

The iterative procedure (2.20) is computationally expensive. In order to find V_1 one has to compute the strict 1-step controllable region for every single state in V_0, i.e. every state allowed for the robot at midstance. Note, that these states also include all 1-step viable states, because $V_1 \subset V_0$. Therefore, all calculations which are necessary to find the region V_2, are already done at the previous iteration, during the computation of V_1 — hence, these calculations are repeated. In fact, the property of the viable regions to form a nested shrinking sequence ($V_n \subset V_{n-1}$) yields that all calculations performed at any single iteration (i.e. computation of V_n) repeat a part of the calculations performed at the previous iteration (i.e. computation of V_{n-1}).

This computational inefficiency may be partially overcome by storing, for every state $q \in V_0$, the corresponding region $\hat{C}(q)$, which is computed during the first iteration, when V_1 is found. The information about all regions $\hat{C}(q)$ can then

be reused to find every viable region V_n. The amount of computation in this case is replaced by significant memory demands.

Note, that formula (2.20) allows us to find only the viable regions V_n. However, if one also wishes to compute the extended viable regions $\bar{V}_n$, the algorithm described in Section 2.7.3 should be used instead, which allows to compute both the viable and extended viable regions V_n and $\bar{V}_n$.

2.7.3 Extended viable regions

The extended viable and controllable regions $\bar{V}_n$ and $\bar{C}_n$ for a specific model can be computed by iterative procedures, analogous to those we used to find the controllable regions in Section 2.7.1 and the viable regions in Section 2.7.2.

As discussed in Section 2.6.1, the extended n-step viable region $\bar{V}_n$ for $n \geq 1$ is all combinations (q, u) of initial states q and controls $u \in U$ of the first step, for which the robot can reach a state in V_{n-1} at the end of the first step. Therefore, similar to equation (2.20) for the viable regions, the extended region $\bar{V}_n$ is the strict extended 1-step controllable region corresponding to the target V_{n-1}:

$$\bar{V}_n = \hat{\bar{C}}(V_{n-1}) = \bigcup_{q \in V_{n-1}} \hat{\bar{C}}(q), \qquad \text{for } n \geq 1. \tag{2.21}$$

The (strict) extended controllability map $\hat{\bar{C}}(\cdot)$ is defined by (2.11) on page 43. Note, that formula (2.21) above requires the knowledge of the $(n-1)$-step viable region V_{n-1}. For each n, we find the region V_n as the projection of the corresponding extended region $\bar{V}_n$ onto the Poincaré section S:

$$V_n = \underset{S}{proj}\, \bar{V}_n. \tag{2.22}$$

That is, for each extended state (q, u) in $\bar{V}_n$, we include the corresponding state q into the viable region V_n.

Equations (2.21) and (2.22) together define a procedure to sequentially compute both the viable and extended viable regions V_n and $\bar{V}_n$. Each iteration of the procedure consists of two steps:

(I). Compute the extended viable region $\bar{V}_n$ according to (2.21), using the viable region V_{n-1} found at the previous iteration.

(II). Find the viable region V_n as the projection (2.22) of the just computed extended region $\bar{V}_n$ onto the state space of the robot.

We assume that the extended 0-step viable region $\bar{V}_0$ — all combinations of states of the robot allowed at midstance and corresponding controls allowed during the first step — is given or can be found either analytically or numerically, based on the given model, constraints, and feasible controls. The extended viability kernel $\bar{V}_\infty$, the limit of the regions $\bar{V}_n$, can be approximated by the extended n-step viable region $\bar{V}_n$ for a sufficiently large n.

Note, however, that formula (2.21) for finding the regions $\bar{V}_n$ is computationally expensive, similarly to equation (2.20) for computing only the viable regions V_n. See Section 2.7.2 on page 56 for more details on the issue.

2.7.4 Extended controllable regions

The extended n-step controllable regions $\bar{C}_n$ can be found similarly to the computation procedure for the controllable regions C_n described in Section 2.7.1.

Equation (2.14) equals the region $\bar{C}_n$ to the strict extended 1-step controllable region $\hat{\bar{C}}(C_{n-1})$. We represent $\hat{\bar{C}}(C_{n-1})$ as the union of all strict extended regions $\hat{\bar{C}}(q)$ corresponding to different target states q in C_{n-1}:

$$\bar{C}_n = \hat{\bar{C}}(C_{n-1}) = \bigcup_{q \in C_{n-1}} \hat{\bar{C}}(q), \qquad \text{for } n \geq 1. \tag{2.23}$$

The amount of computational work in the right-hand side of the above equation can be significantly reduced. For this purpose, notice that equation (2.23) is almost the same as equation (2.17) for the controllable regions C_n. The only difference is the extended map $\hat{\bar{C}}$ used in (2.23) compared to the map C in (2.17). We then follow the steps on page 50 that we used to transform equation (2.17) into the low-computer-work formula (2.19) for the controllable regions, to transform the formula (2.23) above into

$$\bar{C}_n = \bar{C}_{n-1} \cup \left(\bigcup_{q \in C_{n-1} \setminus C_{n-2}} \hat{\bar{C}}(q) \right), \qquad \text{for } n \geq 2. \tag{2.24}$$

Equation (2.24) above is more computationally efficient than (2.23), because it requires computation of the map $\hat{\bar{C}}(q)$ only for some states q in C_{n-1} (i.e. only those in $C_{n-1} \setminus C_{n-2}$), as opposed to the entire C_{n-1} in (2.23).

Note, that two controllable regions C_{n-1} and C_{n-2} are used in (2.24) to compute the extended region $\bar{C}_n$. In order to avoid calculation of the controllable regions by a separate procedure, we express the region C_n as the projection of the corresponding extended region $\bar{C}_n$ onto the Poincaré section S of the model:

$$C_n = \underset{S}{proj}\, \bar{C}_n. \tag{2.25}$$

That is, we simply 'drop' the controls u from each point (q, u) in $\bar{C}_n$ to get C_n.

Equations (2.24) and (2.25) together provide a method to sequentially find both the controllable and extended controllable regions C_n and $\bar{C}_n$, starting from

$n = 2$. At each step the regions C_n and $\bar{C}_n$ are found based on the previously calculated $\bar{C}_{n-1}$, C_{n-1} and C_{n-2}. Expression (2.23) can be used to find the 1-step region $\bar{C}_1$ and, hence, the region C_1 according to (2.25). Here we assume that we have a way, either analytical or numerical, to compute the extended controllability map $\hat{\bar{C}}(q)$ for an arbitrary state q at midstance. One example of such computation for the 2D IP model can be found in Section 3.2.2.

The extended ∞-step controllable region $\bar{C}_\infty$, the limit of the extended regions $\bar{C}_n$, can be approximated by the region $\bar{C}_n$ for a sufficiently large n, assuming $\bar{C}_\infty$ is bounded. When the region $\bar{C}_\infty$ is unbounded, it might be possible to guess its shape, based on the evolution of the regions $\bar{C}_n$ — for example, we make such guess for the planar LIP model in Section 4.2, supporting it with our intuition and analytical insights about the model.

We summarize the $\bar{C}_n$ computation procedure for a given target region C_0 as follows:

(I). Develop a way to compute the extended controllability map $\hat{\bar{C}}(q)$ for an arbitrary target state q. This can be done either analytically (e.g. by providing analytical formulas for the boundaries of $\hat{\bar{C}}(q)$), or numerically (e.g. by developing a numerical procedure, which approximates $\hat{\bar{C}}(q)$ for a given input q).

(II). Compute the extended 1-step controllable region $\bar{C}_1$ corresponding to the given target C_0, using equation (2.23). Use $\bar{C}_1$ to find the controllable region C_1 according to (2.25).

(III). Iteratively compute as many of the extended n-step controllable regions $\bar{C}_n$ as desired, for $n = 2, 3$, and so on. At each iteration, the region $\bar{C}_n$ is found according to (2.24), using the regions $\bar{C}_{n-1}$, C_{n-1} and C_{n-2} found at

the previous iterations. Then use $\bar{C}_n$ to find the controllable region C_n as in (2.25).

(IV). If, after several iterations of the step (III), each next region $\bar{C}_n$ is 'close' to the previous region $\bar{C}_{n-1}$, use the last computed $\bar{C}_n$ as an approximation of the extended ∞-step controllable region $\bar{C}_\infty$. Then find the controllable region C_∞ according to (2.25). If, however, the regions $\bar{C}_n$ do not seem to approach a fixed bounded area, use the knowledge about the evolution of the regions $\bar{C}_n$ to increase your intuition about $\bar{C}_\infty$.

2.7.5 Constraints

Suppose one or more additional constraints are imposed on a given model. In this case, the viable and controllable regions (hence, the extended viable and controllable regions too) may reduce in size and change in shape — see Section 2.5.1 for more details and examples of constraints. Hence, all regions have to be recomputed for the constrained model.

Consider the n-step controllable regions C_n, which we compute by the iterative procedure, described on page 54 and represented by equation (2.19). Notice, that any information about the model is only used to find the 1-step controllable region $C(q)$ for an arbitrary target state q (i.e. only in the step (I) of the procedure). Computation of all regions C_n relies on the knowledge of $C(q)$ for any given q. The map $C(\cdot)$, thus, is a crucial element of the controllable regions concept, as it contains all information about all regions C_n. Therefore, after the introduction of the new constraints to the model (or any other changes to the model of the robot or environment), only calculation of $C(q)$ has to be revisited.

This property of the method is useful for its implementation on a computer.

The same property is valid for the methods of finding the viable, extended viable, and extended controllable regions for a robot. After any changes in the model, only the calculation of the (strict) (extended) 1-step controllable region for an arbitrary target state has to be corrected in order to recompute all regions V_n, $\bar{V}_n$, and $\bar{C}_n$.

CHAPTER 3

INVERTED PENDULUM MODEL IN 2D

One of the simplest possible models of bipedal locomotion is the Inverted Pendulum (IP) model in 2D [43, 36, 84, 10], introduced in Section 2.4.1. The model is depicted on Fig. 2.5a on page 29. The system has one dynamic variable (the stance leg angle θ) and two phase variables (angle θ and angular velocity $\dot{\theta}$). There are two control parameters per each step: the step length x_{st}, which determines the next stepping location and time, and the push-off impulse p, applied along the stance leg just before the collision instant. It is assumed that the swing leg can be instantaneously moved to any desired position without influencing dynamics of the stance leg. The effects of ankle torques (except for at push-off) and hip torques (reacting against an upper body that our model does not have) are neglected. We consider only walking motions of the robot.

In this Chapter we derive the dynamics and walking constraints of the planar IP model (Section 3.1). We compute the viable and controllable regions V_n and C_n, and the extended viable and controllable regions $\bar{V}_n$ and $\bar{C}_n$ for the model (Section 3.2). Both the unconstrained model and the model with limited actuation are considered. The controllable regions for a range of different targets were found by Wolfslag [96] and also discussed in [10]. Based on our calculations, we justify for the 2D IP model the 'Viable is Controllable' and the 'Two-step controllability' claims made in Section 2.3 on page 25.

3.1 Equations of motion

There are formally two phases of the motion of the robot: the single stance phase, when exactly one leg is in contact with the ground, and instantaneous collision, which occurs when the swing leg hits the ground. During the single stance motion, we assume that the stance foot is a revolute joint and the robot acts as a simple inverted pendulum. This is described by the following equation in non-dimensional form:

$$\ddot{\theta} = \sin\theta. \tag{3.1}$$

The non-dimensionalization is done using the constants m, l and $\sqrt{l/g}$ for mass, length, and time correspondingly, where m is the mass of the robot (i.e. of the hip), l the leg length, and g the acceleration due to gravity. Note, that such non-dimensionalization is effectively equivalent to setting the values of m, l, and g equal to 1.

A collision occurs when the swing leg hits the ground, i.e. when the height of the swing foot above the ground becomes zero:

$$y_{sw} = \cos\theta - \cos\theta_{sw} = 0. \tag{3.2}$$

Here θ_{sw} is the angle of the swing leg relative to the vertical at the instant of collision. Note, that the swing leg angle is only important at heel-strike, because the motion of the massless swing leg does not influence the dynamics of the stance leg. We assume that at heel-strike, e.g. as on Fig. 3.1, the stance leg angle θ is positive, while the swing leg angle θ_{sw} is negative. Angle θ_{sw} is related to the step size x_{st} by

$$x_{st} = -2\sin\theta_{sw}. \tag{3.3}$$

We assume that the swing leg controller successfully achieves the step size x_{st}

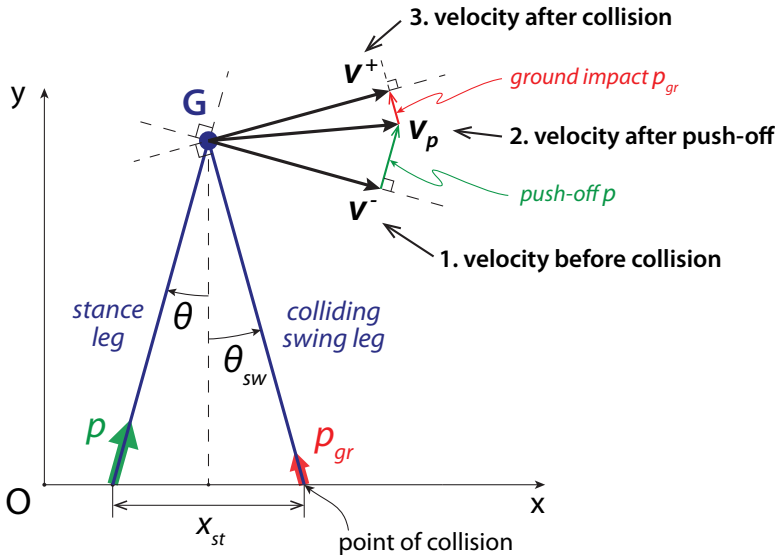


Figure 3.1: Collision in the planar IP model. Throughout the collision the configuration of the robot stays the same, but velocities are discontinuous. Two impulses produce jumps in the velocities: the push-off impulse p along the stance leg and the ground reaction impulse p_{gr} along the swing leg. First, the push-off is applied just before the instant of collision, instantaneously changing the hip velocity in the stance leg direction, from $\vec{v}^-$ to $\vec{v}_p$. The amount of push-off is chosen by the controller. Next, the ground reaction impact p_{gr} hits along the swing leg, so that after the impact the hip moves perpendicular to the swing leg. The hip velocity jumps from $\vec{v}_p$ to $\vec{v}^+$. After the collision the legs change their roles: the stance leg becomes the new swing leg and vice versa.

specified by the balance controller at each midstance, thus determining the angle θ_{sw} at the next collision.

When condition (3.2) is met, a collision happens. We use the superscript $^-$ for the values just before the instant of collision, and the superscript $^+$ for the values just after the collision. Throughout the collision, the configuration of the robot remains the same, but the legs swap their roles: the (former) stance leg becomes the (new) swing leg, and vice versa. Therefore, the leg angles swap too:

$$\theta^+ = \theta_{sw} = -\theta^-. \tag{3.4}$$

Velocities of the system are discontinuous across the collision. Two impulses affect velocity $\vec{v}$ of the hip, as illustrated on Fig. 3.1: the push-off impulse p and

the ground reaction impulse p_{gr}. First, the push-off impulse is applied along the stance leg, producing a jump in the hip velocity in the direction of the stance leg. The velocity $\vec{v}_p$ after the push-off becomes

$$\vec{v}_p = \vec{v}^- + p\vec{r}_{st}, \tag{3.5a}$$

where $\vec{r}_{st}$ is the unit vector along the stance leg, i.e. the vector from the stance foot to the hip. After the push-off, the swing leg collides with the ground, causing a ground reaction impulse p_{gr} along the swing leg — hence, the second jump in the hip velocity. We assume that after the collision the velocity of the swing foot is zero and the hip moves perpendicular to the swing leg. We use conservation of angular momentum about the point of collision to find the hip velocity $\vec{v}^+$ after the collision:

$$\vec{r}_{sw} \times \vec{v}_p = \vec{r}_{sw} \times \vec{v}^+. \tag{3.5b}$$

Here $\vec{r}_{sw}$ is the (unit) vector from the swing foot to the hip. Note, the impulse p_{gr} from the ground is such, that the new hip velocity $\vec{v}^+$ has no component in the direction of the swing leg, as shown on Fig. 3.1. Therefore, p_{gr} is equal in magnitude and opposite in sign to the component of the hip velocity $\vec{v}_p$ along the swing leg direction:

$$p_{gr} = -\vec{v}_p \cdot \vec{r}_{sw}. \tag{3.6}$$

Equations (3.5a) and (3.5b) together express the change in the CoM velocity during the heel-strike, from $\vec{v}^-$ to $\vec{v}^+$. In order to find the angle rate $\dot{\theta}^+$ after the heel-strike, we use the coordinate form of equations (3.5). The vectors $\vec{r}_{st}$ and $\vec{r}_{sw}$ along the stance and swing leg are

$$\vec{r}_{st} = \begin{bmatrix} \sin\theta^- \\ \cos\theta^- \end{bmatrix}, \qquad \vec{r}_{sw} = \begin{bmatrix} \sin\theta^+ \\ \cos\theta^+ \end{bmatrix}. \tag{3.7a}$$

Therefore, velocities $\vec{v}^-$ and $\vec{v}^+$ can be written as

$$\vec{v}^- = \frac{d}{dt}\vec{r}_{st} = \begin{bmatrix} \dot{\theta}^- \cos\theta^- \\ -\dot{\theta}^- \sin\theta^- \end{bmatrix}, \qquad \vec{v}^+ = \frac{d}{dt}\vec{r}_{sw} = \begin{bmatrix} \dot{\theta}^+ \cos\theta^+ \\ -\dot{\theta}^+ \sin\theta^+ \end{bmatrix}. \tag{3.7b}$$

Plugging expressions (3.7) above into two equations (3.5), we solve for $\dot{\theta}^+$ to get

$$\dot{\theta}^+ = \dot{\theta}^- \cos 2\theta_{sw} - p \sin 2\theta_{sw}. \tag{3.8}$$

Equations (3.4) and (3.8) represent the collision law for the 2D IP model, defining the state of the robot after the heel-strike. Between heel-strikes, the robot acts as an inverted pendulum, according to the differential equation (3.1). Kuo [43] and Ruina, et. al. [69] study in more detail the collisions and energetics of collisions of the planar IP model. These studies suggest that the preemptive push-off is an efficient way to compensate for the collisional losses.

Note, that the push-off impulse instantaneously changes the velocity of the hip and, hence, the velocity of the swing foot. As a result, the swing foot might not collide with the ground right after the push-off. In order to ensure the collision (and the desired stepping location), the swing leg controller may have to instantaneously change the angular rate of the swing leg. In part, this supports the strategy of the swing leg retraction prior to the instant of collision. More insights about the importance of the leg retraction in legged locomotion can be found in Hasaneini's work [32].

3.1.1 Limits of walking

We only study walking motions of the robot, in which exactly one foot is on the ground at each time. Any behavior that leads to the flight phase, or is otherwise

not physically realizable, has to be avoided and, if it occurs, is considered a failure. We consider possible ways for the system to fail and find the corresponding constraints (i.e. inequalities), which have to be satisfied to avoid the failure.

Negative leg tension. During the single stance motion, the ground reaction force along the stance leg counteracts gravity and keeps the hip moving along a circular trajectory. The reaction force can only be directed upwards, because suction of the leg from the ground is not physically realistic. When the tension is about to turn negative, the stance foot lifts off and the flight phase begins (assuming no problems of the type discussed in [52]). Therefore, we require that the tension in the stance leg is always non-negative:

$$F_{st} = \cos(\theta) - \dot{\theta}^2 \geq 0. \tag{3.9a}$$

Flight after push-off. Consider the ground reaction impulse p_{gr} which hits along the swing leg at the instant of heel-strike. Similar to the stance leg tension, the impulse p_{gr} is not allowed to be negative, i.e. the ground cannot pull on the swing leg. According to expression (3.6) for the ground reaction impulse, we write this constraint as

$$\vec{v}_p \cdot \vec{r}_{sw} \leq 0.$$

That is, the hip velocity $\vec{v}_p$ after the push-off has to have a negative (i.e. downward) component along the swing leg direction $\vec{r}_{sw}$ (e.g., such as on Fig. 3.1). On the other hand, if $\vec{v}_p$ has an upward component along the swing leg (say, after a too strong push-off), the ground impact does not occur and the robot transits into the flight phase. We use equations (3.5a) and (3.7) to write the above inequality as

$$p \cos 2\theta^- - \dot{\theta}^- \sin 2\theta^- \leq 0. \tag{3.9b}$$

Inequalities (3.9a) and (3.9b) above represent the no-flight constraints of the robot. Two other constraints include the non-negative push-off impulse p (similar to the non-negative ground reaction impulse p_{gr}):

$$p \geq 0, \tag{3.9c}$$

and the no-falling-down requirement, which we write as the positive height of the hip above the ground:

$$y = \cos\theta > 0. \tag{3.9d}$$

Four inequalities (3.9) represent the walking constraints of the simple IP model in 2D. One or more additional constraints may be imposed, based on a specific problem. Examples of such constraints can be found in Sec. 2.5.1. The walking constraints (3.9) above repeat the results in Wolsflag's work [96].

3.1.2 Poincaré map

For the viability and controllability analysis, we want to know how the state of the robot changes after each step. We have defined a step to start at midstance and end at the next midstance (see Chapter 2 on page 15). For the planar IP model, the midstance is the instant when the stance leg is vertical, i.e. when the stance leg angle is zero:

$$\textit{Midstance:} \qquad \theta = 0. \tag{3.10}$$

Recall from Section 2.4.3, that the midstance is one-dimensional. We describe it by a single variable, velocity $v = \dot{\theta}$ of the hip.

Given the initial midstance velocity $\dot{\theta}_0$ and two control parameters (the step size x_{st} and push-off p), we want to find the velocity $\dot{\theta}_1$ at the next midstance.

We use the total mechanical energy of the robot for our calculations, as it is conserved during the single stance motion between collisions. In non-dimensional form the energy is given by

$$E = \cos\theta + \frac{1}{2}\dot{\theta}^2. \tag{3.11}$$

We write the conservation of energy between the initial midstance and the instant just before the heel-strike:

$$1 + \frac{1}{2}\dot{\theta}_0^2 = \cos\theta^- + \frac{1}{2}(\dot{\theta}^-)^2.$$

Solving the above equation for the stance leg angle $\dot{\theta}^-$ just before the collision, we find

$$\dot{\theta}^- = \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2}. \tag{3.12}$$

Note, that we only consider non-negative $\dot{\theta}^-$ above, because we assume forward motion of the robot. Similarly, we use the conservation of energy between the instant just after the heel-strike and the next midstance:

$$1 + \frac{1}{2}\dot{\theta}_1^2 = \cos\theta^+ + \frac{1}{2}(\dot{\theta}^+)^2,$$

to find velocity $\dot{\theta}^+$ in terms of the final velocity $\dot{\theta}_1$:

$$\dot{\theta}^+ = \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_1^2}. \tag{3.13}$$

Velocities $\dot{\theta}^-$ and $\dot{\theta}^+$ just before and just after the collision are related to each other by the collision velocity jump (3.8). Plugging expressions (3.12) and (3.13) above into equation (3.8), we obtain

$$\sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_1^2} = -p\sin 2\theta_{sw} + \cos 2\theta_{sw}\sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2}. \tag{3.14}$$

Equation (3.14) above represents the Poincaré map of the system: it establishes the relationship between the initial and final velocities $\dot{\theta}_0$ and $\dot{\theta}_1$, and the

controls for the step (the push-off p and the swing leg collision angle θ_{sw}, which is a function of the step size x_{st} according to (3.3)).

We also write the four walking constraints (3.9) of the model in terms of $\dot{\theta}_0$, $\dot{\theta}_1$, p, and θ_{sw}. The flight-after-push-off constraint (3.9b) is transformed using formula (3.4) for θ^- and (3.12) for $\dot{\theta}^-$, to become

$$p\cos 2\theta_{sw} + \sin 2\theta_{sw}\sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2} \leq 0. \tag{3.15a}$$

Next, consider constraint (3.9a) representing the non-negative ground reaction force F_{st} in the stance leg. During the single stance motion, the force F_{st} is bigger when the angle rate $\dot{\theta}$ of the leg is smaller, and vice versa. That is, F_{st} increases as the robot moves from the initial midstance to heel-strike, and decreases from heel-strike to the next midstance. Therefore, it is sufficient to check that the force in the stance leg is non-negative only at the instants just before and just after the collision:

$$\cos\theta_{sw} - (\dot{\theta}^-)^2 \geq 0, \qquad \cos\theta_{sw} - (\dot{\theta}^+)^2 \geq 0.$$

Plugging expressions (3.12) for $\dot{\theta}^-$ and (3.13) for $\dot{\theta}^+$ into the inequalities above, gives us

$$\frac{2}{3}\left(1 + \frac{1}{2}\dot{\theta}_0^2\right) \leq \cos\theta_{sw}, \tag{3.15b}$$

$$\frac{2}{3}\left(1 + \frac{1}{2}\dot{\theta}_1^2\right) \leq \cos\theta_{sw}. \tag{3.15c}$$

The two constraints above restrict the maximum allowed collision angle θ_{sw} (hence, the step size x_{st} too) for each initial velocity $\dot{\theta}_0$ and final velocity $\dot{\theta}_1$ respectively.

Similar to the leg tension constraint, the no-falling constraint (3.9d) requires checking only at the instant of collision, when the height of the hip above the

ground is lowest:

$$\cos\theta_{sw} > 0.$$

However, the inequality above is guaranteed to hold true, whenever either of the conditions (3.15b) or (3.15c) is satisfied. Therefore, we do not include the above inequality into the constraints of the model. Finally, the non-negative push-off constraint (3.9c) also applies:

$$p \geq 0. \tag{3.15d}$$

The Poincaré map (3.14), the step size equation (3.3), and the four constraints (3.15) define all combinations of the four parameters — the initial and final hip velocities $\dot{\theta}_0$ and $\dot{\theta}_1$, and the controls p and x_{st} — which correspond to a non-failed step of the planar IP model. For example, if velocity $\dot{\theta}_1$ is fixed, then these equations define all possible initial velocities $\dot{\theta}_0$ and corresponding push-offs p and step sizes x_{st}, which let the robot reach velocity $\dot{\theta}_1$ at the next midstance. That is, they define the 1-step controllable region corresponding to the target velocity $\dot{\theta}_1$ (see Section 3.2.2 for numerical computation of the region C_1). The Poincaré map and the no-flight constraints of the planar IP model were derived by Wolfslag in [96].

3.2 Viability and controllability of the 2D IP model

In this section we compute the extended viable and controllable regions $\bar{V}_n$ and $\bar{C}_n$ for the 2D IP model. We also compute the (superficially simpler to describe) viable and controllable regions V_n and C_n by 'dropping' controls u from each extended viable state (q, u). The definitions of the regions V_n, C_n, $\bar{V}_n$, and $\bar{C}_n$ are in Sections 2.1, 2.2, 2.6.1, and 2.6.3 correspondingly.

Our model has one state variable (velocity $\dot{\theta}$ at midstance) and two control variables (push-off p and step size x_{st}). Therefore, an extended state may have up to three coordinates, $(\dot{\theta}, p, x_{st})$, leading to three-dimensional extended viable and controllable regions $\bar{V}_n^{p,x_{st}}$ and $\bar{C}_n^{p,x_{st}}$. However, for the purposes of simplicity (analytical, numerical, and graphical) we prefer to work with regions with two or fewer dimensions. We extend the state space by only one control variable at a time. We consider separately the regions $\bar{V}_n^{x_{st}}$, $\bar{C}_n^{x_{st}}$ extended by the step size x_{st} and the regions $\bar{V}_n^{p}$, $\bar{C}_n^{p}$ extended by the push-off p. Nevertheless, we later show the full three-dimensional extended viability kernel $\bar{V}_\infty^{p,x_{st}}$ and the three-dimensional extended ∞-step controllable region $\bar{C}_\infty^{p,x_{st}}$ in Appendix C.3 on page 182.

3.2.1 Allowed states and controls

We first describe all non-failed states at midstance and feasible controls of the model — that is, the 0-step viable region V_0 and the extended 0-step viable regions $\bar{V}_0^{x_{st}}$ and $\bar{V}_0^{p}$.

A state of the robot at midstance (velocity $\dot{\theta}$) is not failed, if the tension in the stance leg is non-negative. Therefore, from (3.9a) we get:

$$\cos\theta - \dot{\theta}^2 \underset{\text{at midstance}}{=} 1 - \dot{\theta}^2 \;\geq\; 0.$$

Considering only positive velocities (only motion forward), gives us all non-failed states, i.e. the 0-step viable region V_0:

$$V_0 : \qquad 0 \leq \dot{\theta} \leq 1. \tag{3.16a}$$

For now, we do not restrict the push-off amount p from above (we do not limit strengths) and allow any non-negative push-off, as stated by (3.9c). Hence, the

extended 0-step viable region $\bar{V}_0^p$ is

$$\bar{V}_0^p : \qquad 0 \le \dot{\theta} \le 1, \quad p \ge 0. \tag{3.16b}$$

The step size x_{st} cannot be negative (no backward motion) and is physically constrained from above by the total length of both legs (i.e. by 2). So, the extended region $\bar{V}_0^{x_{st}}$ is

$$\bar{V}_0^{x_{st}} : \qquad 0 \le \dot{\theta} \le 1, \quad 0 \le x_{st} \le 2. \tag{3.16c}$$

The 0-step regions V_0, $\bar{V}_0^{x_{st}}$, and $\bar{V}_0^p$ are shown on Fig. 3.5 on page 86. The figure also depicts the viable regions V_n and extended viable regions $\bar{V}_n^{x_{st}}$ and $\bar{V}_n^p$, which we find in Section 3.2.4.

3.2.2 One-step controllability

We use the iterative procedure described in Section 2.7.4 on page 60 to compute the controllable and extended controllable regions. First, we describe the extended controllability map $\hat{\bar{C}}$ — we find the extended 1-step controllable regions $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^p$ corresponding to an arbitrary target state.

For our planar IP model, an arbitrary target state is represented by a desired velocity $\dot{\theta}_t$ of the hip at midstance. The region $\bar{C}_1^{x_{st}}$ is all possible combinations $(\dot{\theta}_0, x_{st})$ of initial velocity $\dot{\theta}_0$ and step size x_{st}, for which there is a feasible push-off p, such that the robot reaches the target velocity $\dot{\theta}_t$ at the next midstance. Recall from Section 3.1.2, that a step of the robot is not failed, if the values of $\dot{\theta}_0$, x_{st}, p and the final velocity $\dot{\theta}_1$ satisfy the Poincaré map equations (3.14), (3.3) and the four walking constraints (3.15). Therefore, the region $\bar{C}_1^{x_{st}}$ is given by all $\dot{\theta}_0$ and x_{st}, which satisfy equalities (3.14), (3.3) and inequalities (3.15) for some value of p and the fixed final velocity $\dot{\theta}_1 = \dot{\theta}_t$. We are going to find the region $\bar{C}_1^{x_{st}}$ by

providing analytical equations for its boundary curves. We also illustrate $\bar{C}_1^{x_{st}}$ on Fig. 3.2b for the example case

$$\dot{\theta}_t = 0.3 \tag{3.17}$$

(corresponding to $\approx 0.94\ m/s$ for a robot with one-meter-long legs).

For each of the constraints (3.15) we find all $\dot{\theta}_0$ and x_{st}, where this constraint is active, i.e. turns into equality. The non-negative push-off constraint (3.15d) is active when $p = 0$. Plugging zero push-off into the Poincaré map (3.14) gives us

$$\sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_t^2} = \cos 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2},$$

which can be further transformed into

$$\dot{\theta}_0^2 = \frac{(2 - 2\cos\theta_{sw})\sin^2 2\theta_{sw} + \dot{\theta}_t^2}{\cos^2 2\theta_{sw}}. \tag{3.18a}$$

Note, the swing leg angle θ_{sw} (for $-\pi/2 \le \theta_{sw} \le 0$) is a one-to-one function (3.3) of the step size x_{st}. Hence, equation (3.18a) above defines a curve in the $(\dot{\theta}_0, x_{st})$-plane. This is the bottom-right boundary of $\bar{C}_1^{x_{st}}$ on Fig. 3.2b. For each given initial velocity, the curve (3.18a) shows where the robot has to step in order to reach the target in one step and without pushing off.

Similarly, the flight-after-push-off constraint (3.15a) is active, when

$$p\cos 2\theta_{sw} + \sin 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2} = 0.$$

Solving the above equation for p and plugging the result into the Poincaré map (3.14), we obtain

$$\dot{\theta}_0^2 = -(2 - 2\cos\theta_{sw})\sin^2 2\theta_{sw} + \dot{\theta}_t^2 \cos^2 2\theta_{sw}. \tag{3.18b}$$

Equation (3.18b) defines a curve in the $(\dot{\theta}_0, x_{st})$-plane — the bottom-left boundary of $\bar{C}_1^{x_{st}}$ on Fig. 3.2b. The stepping locations corresponding to this curve are such

that, with an appropriate push-off, the robot reaches the target velocity $\dot{\theta}_t$ in one step and has no collisional impact.

The flight-before-collision constraint (3.15b) gives us the top-right boundary of $\bar{C}_1^{x_{st}}$:

$$\cos\theta_{sw} = \frac{2 + \dot{\theta}_0^2}{3}. \tag{3.18c}$$

For each initial velocity, the above curve shows the largest step the robot is able to take without failing. For such steps, the tension in the stance leg becomes zero just before the instant of collision. Notice, that the curve (3.18c) is independent of the chosen target $\dot{\theta}_t$. Therefore, it restricts all extended viable regions $\bar{V}_n^{x_{st}}$ and all extended controllable regions $\bar{C}_n^{x_{st}}$ (see Figs. 3.5b and 3.4b below). If the robot attempts to take a step larger than that specified by (3.18c), the constraint (3.15b) will be violated at some point during the stance phase. A good analogy for the motion corresponding to (3.18c) was suggested by Wolfslag [96]. Imagine a massive point sliding off the top of a circular hill, with a positive initial velocity. The motion of the mass is equivalent to the motion of the hip in the IP model. Note the angle where the mass detaches from the surface of the hill. This angle corresponds to the maximum collision angle θ_{sw} (hence, the maximum step size for the given initial velocity) of the IP model and is given by equation (3.18c).

As opposed to the boundary (3.18c), the flight-after-collision constraint (3.15c) defines the largest allowed step size for a given *target* velocity $\dot{\theta}_t$:

$$\cos\theta_{sw} = \frac{2 + \dot{\theta}_t^2}{3}. \tag{3.18d}$$

This is the top horizontal boundary of $\bar{C}_1^{x_{st}}$ on Fig. 3.2. For the step sizes defined by (3.18d), the stance leg tension is zero just after the instant of collision, assuming the appropriate push-off p, which brings the robot to the target at the next midstance.

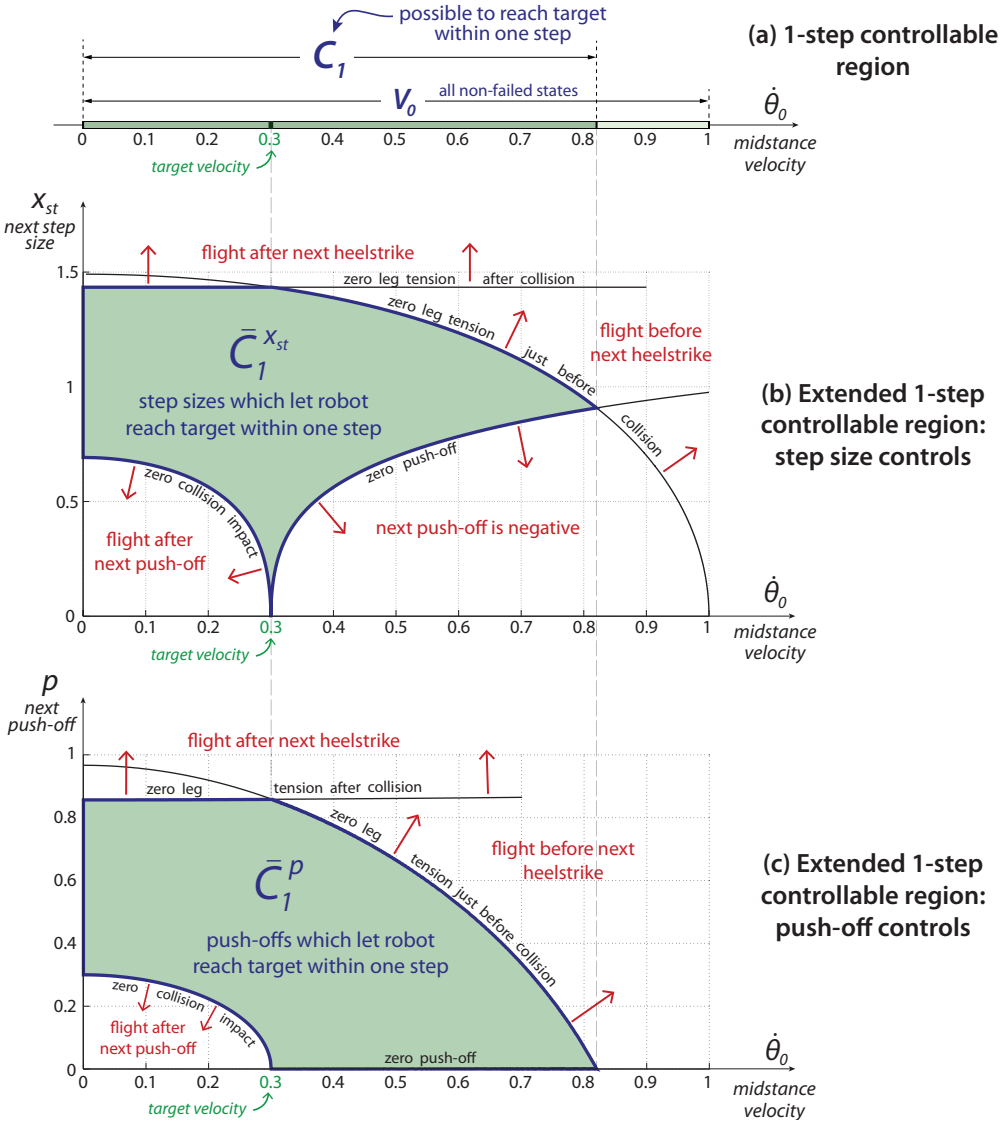


Figure 3.2: 1-step controllable and extended controllable regions for the 2D IP model. **(a)** The 1-step controllable region C_1 is all initial velocities $\dot{\theta}_0$, for which the robot can reach the target within one step. The target here is the velocity $\dot{\theta}_t = 0.3$ at midstance. **(b), (c)** For each velocity $\dot{\theta}_0$, the extended 1-step controllable regions $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{p}$ show, respectively, all step-sizes x_{st} and push-offs p, which let the robot reach the target within one step. The boundary curves of $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{p}$ correspond to the walking constraints of the model. For step sizes/push-offs beyond these curves, the robot either fails or does not reach the target. The region C_1 is the projection of both $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{p}$ onto the $\dot{\theta}_0$-axis.

Each of the four curves (3.18) cuts off a part of the extended space (initial velocities $\dot{\theta}_0$ and step sizes x_{st}), where the robot either fails or is not able to reach the target within one step. Thus, the curves (3.18) form the boundaries of the extended 1-step controllable region $\bar{C}_1^{x_{st}}$ for an arbitrary target velocity $\dot{\theta}_t$. Fig. 3.2b shows the region $\bar{C}_1^{x_{st}}$ for the target (3.17). For any given initial velocity $\dot{\theta}_0$, $\bar{C}_1^{x_{st}}$ shows the range of step sizes available to the step-size controller, such that the robot can, with an appropriate push-off p, reach the target velocity $\dot{\theta}_t$ within one step. For example, for $\dot{\theta}_0 = 0.4$ the allowed step sizes are approximately $0.56 \leq x_{st} \leq 1.39$. The appropriate push-offs are described by the extended 1-step controllable region $\bar{C}_1^p$, which we derive in this section below.

All initial velocities, for which there is at least one step size in $\bar{C}_1^{x_{st}}$ (i.e. the projection of $\bar{C}_1^{x_{st}}$ onto the $\dot{\theta}_0$-axis), constitute the 1-step controllable region C_1, shown on Fig. 3.2a. If the robot's velocity at midstance is outside of C_1, there is no way for the robot to reach the target velocity in one step. For the target (3.17), the maximum 1-step controllable velocity is $\dot{\theta}_0 \approx 0.82$.

We would also like to know all push-offs p, which let the robot reach the target within one step. Such push-offs are described by the extended 1-step controllable region $\bar{C}_1^p$. It consists of all combinations $(\dot{\theta}_0, p)$ of initial velocity $\dot{\theta}_0$ and push-off p, for which there is a feasible step size x_{st} such that the robot reaches the target $\dot{\theta}_t$ at the next midstance. The region $\bar{C}_1^p$ for the target (3.17) is shown on Fig. 3.2c. Similar to the derivation of $\bar{C}_1^{x_{st}}$ above, we obtain the boundaries of $\bar{C}_1^p$ by finding all p and $\dot{\theta}_0$, for which the walking constraints (3.15) turn into equalities. Thus, the stance-leg-tension constraints (3.15b) and (3.15c),

in combination with the Poincaré map (3.14), yield

$$p = \cot 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2} - \frac{1}{\sin 2\theta_{sw}} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_t^2}, \qquad \text{where} \quad \cos\theta_{sw} = \frac{2 + \dot{\theta}_0^2}{3} \tag{3.19a}$$

and

$$p = \cot 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2} - \frac{1}{\sin 2\theta_{sw}} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_t^2}, \qquad \text{where} \quad \cos\theta_{sw} = \frac{2 + \dot{\theta}_t^2}{3} \tag{3.19b}$$

respectively. The two above formulas differ only in the expressions for $\cos\theta_{sw}$. (3.19a) and (3.19b) each define a curve in the $(\dot{\theta}_0, p)$ plane — these are, correspondingly, the top-right and top-left boundaries of the region $\bar{C}_1^p$ on Fig. 3.2c.

Next, the flight-after-push-off constraint (3.15a) and the Poincaré map (3.14) give us

$$\begin{cases} p &= -\sin 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_t^2}, \\ \dot{\theta}_0^2 &= \dot{\theta}_t^2 \cos^2 2\theta_{sw} - (2 - 2\cos\theta_{sw}) \sin^2 2\theta_{sw}. \end{cases} \tag{3.19c}$$

We cannot eliminate θ_{sw} from the above equations analytically. Instead, for different initial velocities $\dot{\theta}_0$ in the range $0 \le \dot{\theta}_0 \le 1$ (all 0-step viable velocities) we numerically solve the system (3.19c) for p. All calculations are performed in Matlab using a non-linear equation solver *fsolve*. The resulting curve in the $(\dot{\theta}_0, p)$ coordinates is the bottom-left boundary curve of $\bar{C}_1^p$. Finally, the non-negative push-off constraint (3.15d) restricts the region $\bar{C}_1^p$ by the horizontal axis:

$$p = 0. \tag{3.19d}$$

The four curves (3.19) form the boundaries of the region $\bar{C}_1^p$, as shown on Fig. 3.2c. For each velocity $\dot{\theta}_0$, $\bar{C}_1^p$ shows the range of all push-offs p, which let

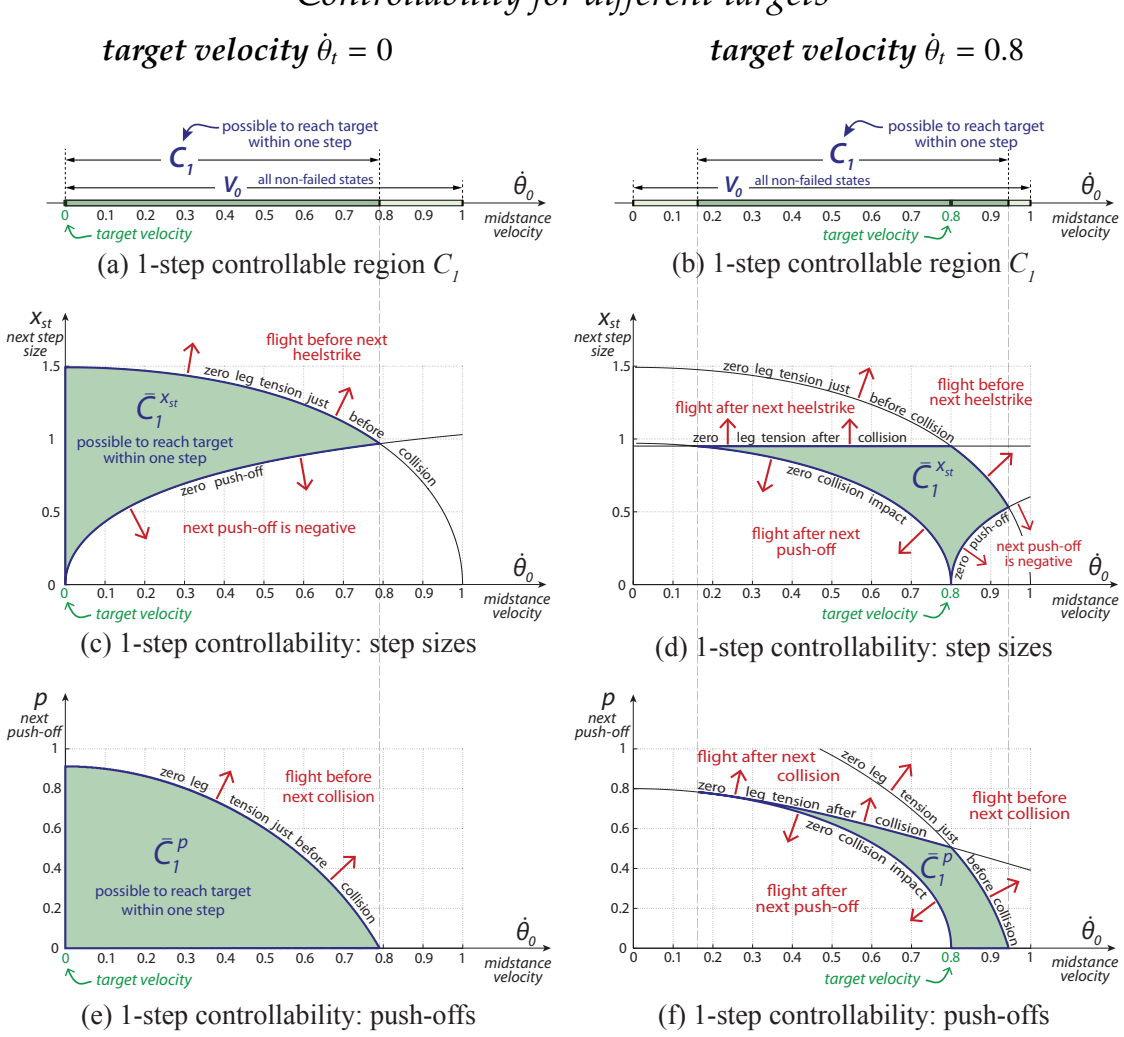


Figure 3.3: 1-step controllability of the 2D IP model for different targets. This figure is analogous to the Fig. 3.2 on page 77, but for different target velocities $\dot\theta_t$. The graphs (a), (c), (e) assume $\dot\theta_t = 0$, while the graphs (b), (d), (f) consider $\dot\theta_t = 0.8$. In the case $\dot\theta_t = 0$, the 1-step controllable region C_1 is equivalent to Koolen's 1-step viable-capture basin [42]. Fig. 3.2, where the target is $\dot\theta_t = 0.3$, would be a column of figures between the two columns on this Fig. 3.3.

the robot reach the target at the next midstance. Similar to $\bar C_1^{x_{st}}$, the projection of $\bar C_1^{p}$ onto the $\dot\theta_0$-axis is the 1-step controllable region C_1 shown on Fig. 3.2a.

We also show examples of the 1-step regions C_1, $\bar C_1^{x_{st}}$, and $\bar C_1^{p}$ for target velocities, different from (3.17). Fig. 3.3 shows such regions for the target velocities $\dot\theta_t = 0$ (plots (a), (c), (e) on the left) and $\dot\theta_t = 0.8$ (plots (b), (d), (f) on the right).

Controllable regions for a zero target velocity of $\dot{\theta}_0 = 0$ correspond to Koolen's viable-capture basins [42].

3.2.3 n-step and ∞-step controllability

We now compute the n-step controllable regions C_n and extended n-step controllable regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{p}$ (see Sections 2.2 and 2.6.3 for definitions) for our planar IP model for $n \geq 2$. The region C_n is all initial velocities $\dot{\theta}_0$, such that the robot can, with appropriate controls, reach the target in n or fewer steps. The corresponding appropriate controls are studied by the extended regions: $\bar{C}_n^{x_{st}}$ (and, respectively, $\bar{C}_n^{p}$) is all combinations of velocities $\dot{\theta}_0$ and step sizes x_{st} (push-offs p) for the next step, which let the robot reach the target within n steps.

We compute the regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{p}$ numerically, according to the iterative procedure described in Sec. 2.7.4 on page 60 (see step (III) of the procedure). We use the example target velocity (3.17). Technical details of the computation are in Appendix C.1 on page 179. Figs. 3.4b and 3.4c show, correspondingly, the regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{p}$ for several values of n. The projections of both $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{p}$ onto the $\dot{\theta}_0$-axis are the n-step controllable regions C_n, shown on Fig. 3.4a. The controllable regions C_n for the 2D IP model were also derived by Wolfslag [96]. The 1-step regions C_1, $\bar{C}_1^{x_{st}}$, and $\bar{C}_1^{p}$ on Fig. 3.4 are the same as those on Fig. 3.2. Note also the nested structure of the presented regions: $C_n \subset C_{n+1}$ for $n \geq 0$, and $\bar{C}_n^{x_{st}} \subset \bar{C}_{n+1}^{x_{st}}$, $\bar{C}_n^{p} \subset \bar{C}_{n+1}^{p}$ for $n \geq 1$.

Notice, that the top-right boundaries of all $\bar{C}_n^{x_{st}}$ are parts of the same curve. As discussed in Sec. 3.2.2 above, this is the curve (3.18c), which corresponds to the motion with zero tension in the stance leg just before the instant of collision.

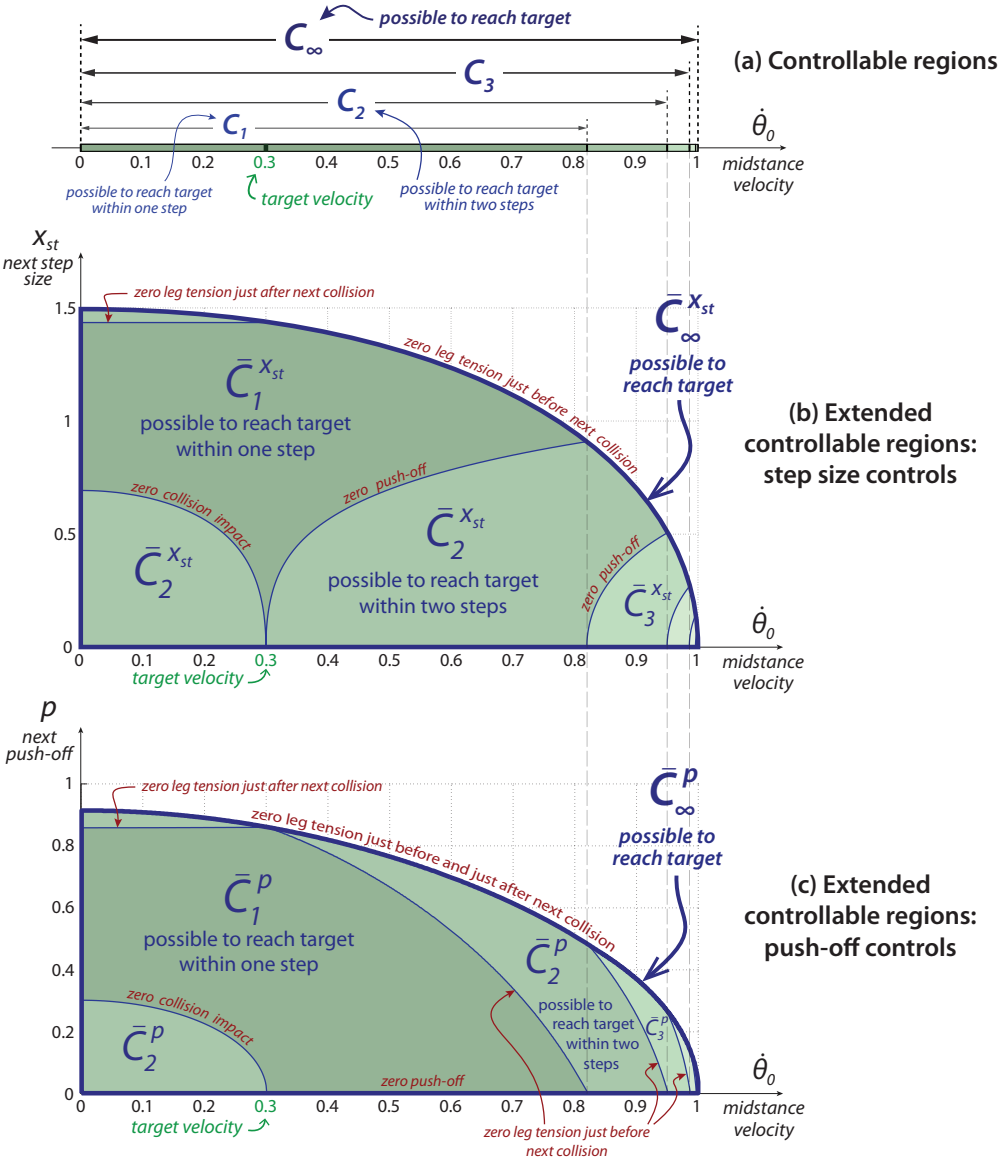


Figure 3.4: *n*-step controllability of the 2D IP model. **(a)** The n-step controllable region C_n is all initial velocities $\dot\theta_0$, for which the robot can reach the target velocity $\dot\theta_t$ in n or fewer steps. Here $\dot\theta_t = 0.3$. **(b), (c)** For each $\dot\theta_0$, the extended n-step controllable regions $\bar C_n^{x_{st}}$ and $\bar C_n^{p}$ show, respectively, all step sizes x_{st} and push-offs p of the next step, which let the robot reach the target in n or fewer steps. The regions C_n, $\bar C_n^{x_{st}}$, and $\bar C_n^{p}$ form sequences of nested regions, which approach the ∞-step regions C_∞, $\bar C_\infty^{x_{st}}$, and $\bar C_\infty^{p}$ (shown by their boundary) correspondingly. These are, respectively, all velocities $\dot\theta_0$ and combinations $(\dot\theta_0, x_{st})$ and $(\dot\theta_0, p)$ of velocities and controls for which the target can be reached or approached asymptotically. Each C_n is the projection of both $\bar C_n^{x_{st}}$ and $\bar C_n^{p}$ onto the $\dot\theta_0$-axis.

As n increases, the regions $\bar{C}_n^{x_{st}}$ appear to approach a fixed area in the plane, the area below the curve (3.18c). Therefore, following step (IV) of the iterative procedure on page 60, we say that the region below the curve (3.18c) is the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$:[1]

$$\bar{C}_\infty^{x_{st}}: \qquad 0 < \dot{\theta}_0 < 1, \quad 0 \le x_{st} \le 2\sin\left(\arccos\left(\frac{2+\dot{\theta}_0^2}{3}\right)\right). \tag{3.20}$$

We used equation (3.3) to write the curve (3.18c) in terms of the step size x_{st} in the formula above. The lines $\dot{\theta}_0 = 0$ and $\dot{\theta}_0 = 1$ are not in $\bar{C}_\infty^{x_{st}}$, because they correspond to 'standing' in place with infinitely small and infinitely fast steps. In these cases, according to the Poincaré map (3.14) and constraints (3.15), $x_{st} = p = 0$ and there is no change in the hip velocity: $\dot{\theta}_1 = \dot{\theta}_0$.

Similar to $\bar{C}_n^{x_{st}}$, the regions $\bar{C}_n^{p}$ on Fig. 3.4c approach a fixed region, the extended ∞-step controllable region $\bar{C}_\infty^{p}$. As shown in Appendix C.2 on page 181, the top boundary of $\bar{C}_\infty^{p}$ is the curve

$$p = \sqrt{\frac{3}{2+\dot{\theta}_0^2} - \frac{2+\dot{\theta}_0^2}{3}}. \tag{3.21}$$

This curve corresponds to the motion, such that the tension in the stance leg is zero both just before and just after the instant of collision. The extended ∞-step controllable region $\bar{C}_\infty^{p}$ is the area below the curve (3.21):

$$\bar{C}_\infty^{p}: \qquad 0 < \dot{\theta}_0 < 1, \quad 0 \le p \le \sqrt{\frac{3}{2+\dot{\theta}_0^2} - \frac{2+\dot{\theta}_0^2}{3}}. \tag{3.22}$$

The projection of both $\bar{C}_\infty^{x_{st}}$ and $\bar{C}_\infty^{p}$ onto the velocity axis is the ∞-step controllable region C_∞, shown on Fig. 3.4a. C_∞ is the limit of the n-step controllable regions C_n as n increases. We can immediately compute the region C_∞ by ignoring the controls x_{st} in the formula (3.20) for $\bar{C}_\infty^{x_{st}}$ (or, equivalently, the controls p

[1] Recall, that by definition (see Sec. 2.6.3), $\bar{C}_\infty^{x_{st}}$ is the limit of the regions $\bar{C}_n^{x_{st}}$ as $n \to \infty$.

in (3.22) for $\bar{C}^p_\infty$):

$$C_\infty : \qquad 0 < \dot{\theta}_0 < 1. \tag{3.23}$$

Computation of the extended controllable regions for target velocities $\dot{\theta}_t$, other than (3.17), reveals that the ∞-step regions C_∞, $\bar{C}^{x_{st}}_\infty$, and $\bar{C}^p_\infty$ stay the same for any target $0 \leq \dot{\theta}_t \leq 1$. As we show in Section 3.2.4 below, they are (almost) equal to the viable regions V_∞, $\bar{V}^{x_{st}}_\infty$, and $\bar{V}^p_\infty$ correspondingly.

As one can see in Fig. 3.4, the two-step controllable region C_2 is most ($\sim 95\%$) of the ∞-step controllable region C_∞. Similarly, the extended regions $\bar{C}^{x_{st}}_2$ and $\bar{C}^p_2$ are close to $\bar{C}^{x_{st}}_\infty$ and $\bar{C}^p_\infty$ respectively. This supports, for the case of the 2D IP model, the 'Two-step controllability' claim we made in Section 2.3.2 on page 26: if it is possible to reach a given target, in most cases it is possible to reach it in two steps. This two-step result remains valid in the case of limited actuation of the model, as we show in Section 3.2.5 below.

3.2.4 Viable and extended viable regions

We now compute the viable and extended viable regions V_n, $\bar{V}^{x_{st}}_n$, and $\bar{V}^p_n$ for our planar IP model. These regions represent all initial velocities $\dot{\theta}_0$ and next-step controls (step sizes x_{st} or push-offs p), such that the robot is able to take n steps and not fail. We do not use the (general) numerically expensive procedure described in Section 2.7.3; instead, we use some analytical insights about the model and the extended controllable regions (computed above in Section 3.2.3) in order to find the viable and extended viable regions. The 0-step regions V_0, $\bar{V}^{x_{st}}_0$, and $\bar{V}^p_0$ (which represent all allowed states and controls) are already described in Section 3.2.1 on page 73.

First, notice that for our simple model and any non-failed initial velocity $\dot{\theta}_0$, a step formally can be made infinitely small and infinitely fast (i.e. the robot puts the swing leg down at midstance). For such step, one can set $x_{st} = p = 0$ to see that all walking constraints (3.15) are satisfied and, according to the Poincaré map (3.14), the hip velocity does not change: $\dot{\theta}_1 = \dot{\theta}_0$. Therefore, the robot can always take an arbitrary large number of steps without failing. This means that all n-step viable regions are the same and equal to V_0:

$$V_\infty = V_n = V_0 : \qquad 0 \le \dot{\theta}_0 \le 1 \tag{3.24}$$

for any $n \ge 0$. The ability to take one step is equivalent to the ability to take any number of steps, and to just being in a non-failed state. Equality (3.24), however, is rather a result of the simplicity of our planar IP model — it may not hold true for more complex models and does not hold with the 2D IP model if additional constraints are considered. In Section 3.2.5 below we consider the 2D IP model with leg swinging and push-off limitations.

We also note that all viable regions (3.24) are the same as the ∞-step controllable region C_∞, given by (3.23) (except for velocities $\dot{\theta}_0 = 0$ and $\dot{\theta}_0 = 1$, which are ∞-step viable, but not controllable, as pointed out in Section 3.2.3 above):

$$V_0 = V_n = V_\infty \underset{\substack{\textit{except for}\\ \textit{boundary points}}}{\approx} C_\infty. \tag{3.25}$$

The above result confirms, for the case of the 2D IP model, the 'Viable is Controllable' conjecture we made in Section 2.3.1 on page 25: in most cases, when the robot is able to not fail, it is also able to reach any given target.[2] Below we find the extended viability kernels $\bar{V}_\infty^{x_{st}}$ and $\bar{V}_\infty^{p}$ for our model, and show that the 'Viable is Controllable' claim holds true for them too.

[2] The result (3.25) is even stronger: the robot is able, with appropriate controls, to reach any target from (almost) any non-failed initial state.

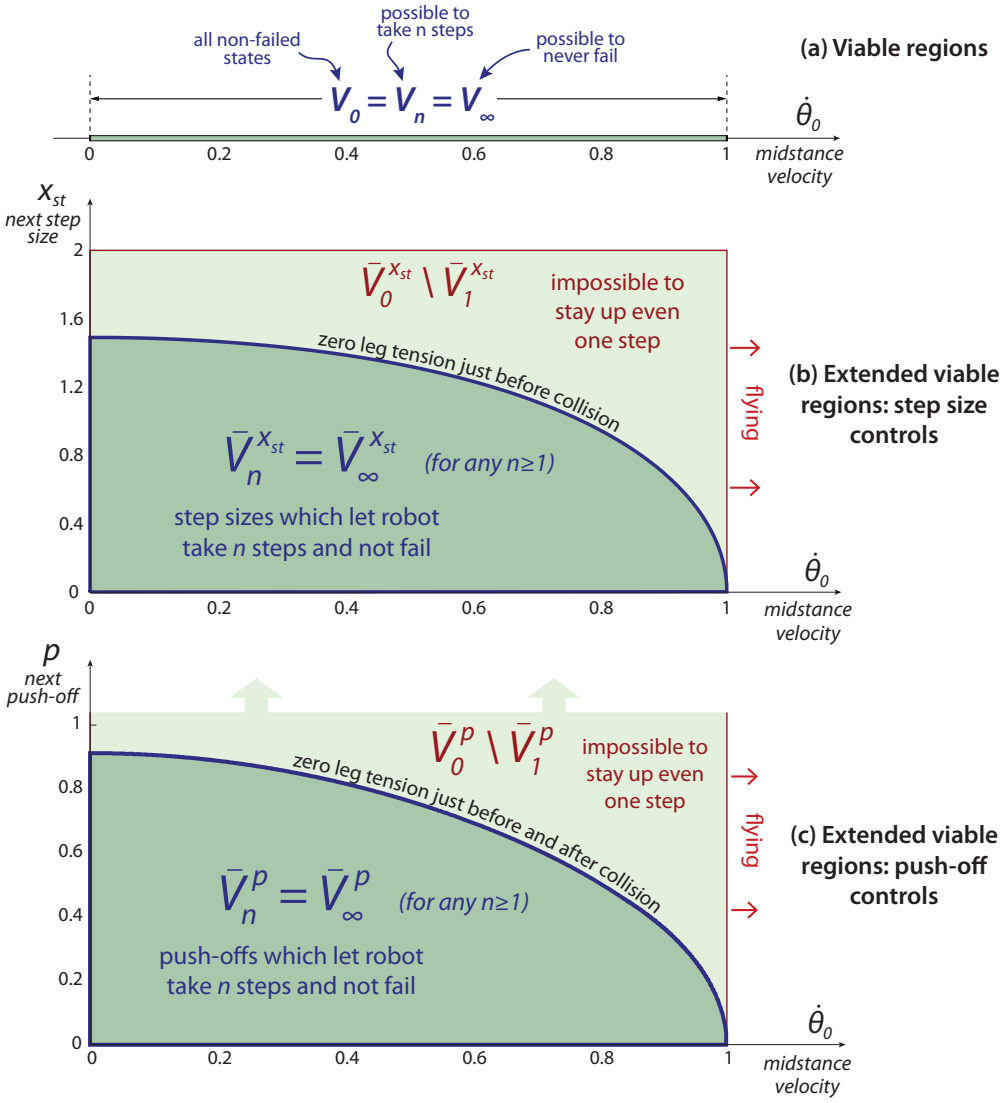


Figure 3.5: Viable and extended viable regions for the 2D IP model. **(a)** The n-step viable region V_n is all initial velocities $\dot{\theta}_0$, for which the robot can, with appropriate controls, take at least n steps and not fail. **(b), (c)** For each $\dot{\theta}_0$, the extended n-step viable regions $\bar{V}_n^{x_{st}}$ and $\bar{V}_n^{p}$ show, respectively, all step sizes x_{st} and push-offs p of the next step, which let the robot take n or more steps without failing. The regions V_n, $\bar{V}_n^{x_{st}}$, and $\bar{V}_n^{p}$ form sequences of nested regions, which approach the ∞-step regions V_∞, $\bar{V}_\infty^{x_{st}}$, and $\bar{V}_\infty^{p}$ (shown by their boundary) correspondingly. These are, respectively, all velocities $\dot{\theta}_0$ and combinations $(\dot{\theta}_0, x_{st})$ and $(\dot{\theta}_0, p)$ of velocities and controls for which the robot is able to step indefinitely. The 0-step regions V_0, $\bar{V}_0^{x_{st}}$, $\bar{V}_0^{p}$ represent all non-failed velocities and allowed controls.

The target (3.17), which we considered in Section 3.2.3 above to compute the controllable and extended controllable regions, is inside the viability kernel V_∞. Therefore, if the robot can reach the target, it can take infinitely many steps. Similar to (2.10), we thus say that the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$ is a subset of the extended viability kernel $\bar{V}_\infty^{x_{st}}$:

$$\bar{C}_\infty^{x_{st}} \subset \bar{V}_\infty^{x_{st}}. \tag{3.26}$$

Therefore, the region $\bar{V}_\infty^{x_{st}}$ includes all points below the curve (3.18c), the top boundary curve of $\bar{C}_\infty^{x_{st}}$ (see Fig. 3.4b). On the other hand, as we stated in Section 3.2.2, the curve (3.18c) does not depend on the target velocity and, thus, restricts (from the top) all extended viable regions $\bar{V}_n^{x_{st}}$ (for $n \geq 1$), including $\bar{V}_\infty^{x_{st}}$. So, the region $\bar{V}_\infty^{x_{st}}$ does not include any points above the curve (3.18c). We conclude that $\bar{V}_\infty^{x_{st}}$ is equal to the region $\bar{C}_\infty^{x_{st}}$ — except for the cases $\dot{\theta}_0 = 0$ and $\dot{\theta}_0 = 1$, which are ∞-step viable, but not controllable, as pointed out in Section 3.2.3 above:

$$\bar{V}_\infty^{x_{st}}: \qquad 0 \leq \dot{\theta}_0 \leq 1, \quad 0 \leq x_{st} \leq 2\sin\left(\arccos\left(\frac{2+\dot{\theta}_0^2}{3}\right)\right). \tag{3.27}$$

The n-step regions $\bar{V}_n^{x_{st}}$ are supersets of $\bar{V}_\infty^{x_{st}}$, but still have to remain below the curve (3.18c). Hence, they are all equal and we write

$$\bar{V}_1^{x_{st}} = \bar{V}_n^{x_{st}} = \bar{V}_\infty^{x_{st}} \underset{\substack{\textit{but for some} \\ \textit{boundary points}}}{\approx} \bar{C}_\infty^{x_{st}}. \tag{3.28}$$

The regions $\bar{V}_n^{x_{st}}$ are shown on Fig. 3.5b.

We show that the relation (3.28) is also valid for the extended n-step viable regions $\bar{V}_n^p$ that represent viable push-offs. Similar to the reasoning of (3.26) above, we first say that $\bar{V}_\infty^p$ includes all of the extended ∞-step controllable region $\bar{C}_\infty^p$ corresponding to the target (3.17): $\bar{C}_\infty^p \subset \bar{V}_\infty^p$ — hence, $\bar{C}_\infty^p \subset \bar{V}_\infty^p \subset \bar{V}_n^p$ for any $n \geq 0$. Next, assume that for a certain $n \geq 1$, there is a point in $\bar{V}_n^p$, which

is outside of $\bar{C}^p_\infty$ — let's call this point $(\dot{\theta}^*_0, p^*)$. With the initial velocity $\dot{\theta}^*_0$ and first push-off p^*, the robot is able to take (at least) one step and not fail (it can take more than one step for $n > 1$, but for now we are only interested in the first step). Therefore, the robot has a non-failed velocity $\dot{\theta}^*_1$ at the next midstance. According to (3.23), velocity $\dot{\theta}^*_1$ is ∞-step controllable (unless $\dot{\theta}^*_1$ is 0 or 1), i.e. the robot is able to reach the target starting with the velocity $\dot{\theta}^*_1$. Then the robot can reach the target with the initial velocity $\dot{\theta}^*_0$ too (unless $\dot{\theta}^*_0$ is 0 or 1) — this contradicts the assumption that the point $(\dot{\theta}^*_0, p^*)$ is outside of $\bar{C}^p_\infty$. We conclude that the assumption is wrong and there is no points in $\bar{V}^p_n$ which are outside of $\bar{C}^p_\infty$. Velocities $\dot{\theta}_0 = 0$ and $\dot{\theta}_0 = 1$, as pointed out in Section 3.2.3 above, are ∞-step viable, but not controllable. Thus, we get

$$\bar{V}^p_1 = \bar{V}^p_n = \bar{V}^p_\infty \underset{\substack{\textit{but for some}\\ \textit{boundary points}}}{\approx} \bar{C}^p_\infty, \tag{3.29}$$

and

$$\bar{V}^p_\infty : \qquad 0 \le \dot{\theta}_0 \le 1, \quad 0 \le p \le \sqrt{\frac{3}{2+\dot{\theta}^2_0} - \frac{2+\dot{\theta}^2_0}{3}}. \tag{3.30}$$

The extended regions $\bar{V}^p_n$ are shown on Fig. 3.5c. The relations (3.28) and (3.29) justify, for the case of the extended controllable regions of the 2D IP model, our 'Viable is Controllable' claim in Section 2.3.1.

3.2.5 Additional constraints

Section 2.5.1 on page 33 discusses possible constraints which one may impose on a simple model of the robot. Such constraints can serve as a proxy for limitations of the more complicated model or the environment. For our planar IP model, we consider two such constraints, which represent actuator limitations of the robot. We compute the controllable and extended controllable regions for

the constrained model and see how the controllability of the robot is affected by these limitations.

First, we bound from above the amount of push-off p, available to the controller at any collision, by a fixed positive value p_{max}:

$$p \leq p_{max} \tag{3.31a}$$

The above constraint represents limited actuation in the ankle motors (or telescopic leg actuators) on a physical robot. Next, we would like to account for limited swing-leg actuation in the robot. As opposed to our simple IP model, a controller on a physical robot cannot move the swing leg infinitely fast; a certain minimum time is required to place the leg into the desired position (defined by the step size x_{st}). We model this requirement with the *step time* t_{st}, the time from the midstance to the next collision. We say that the step time has to be not smaller than a preset positive constant $t_{st,min}$:

$$t_{st} \geq t_{st,min} > 0 \tag{3.31b}$$

Constraint (3.31b) is a (rough) proxy for the swing-leg actuator limitations of the robot (e.g. the torque limits of the hip motor). During 'normal walking', the step time t_{st} is approximately one half of the total duration of a step (midstance-to-midstance time).

We use the example bounds $p_{max} = 0.25$ and $t_{st,min} = 0.5$ (both are non-dimensional), which are chosen only to demonstrate the changes in the controllable regions due to the limited actuation. For comparison, the estimated maximum push-off for the Cornell Ranger robot [9, 11] is 0.3 (see Section 6.1.2). The estimated minimum step-time is 0.4 for Ranger and 0.26 ($\sim$ 0.09 sec.) for humans [79].

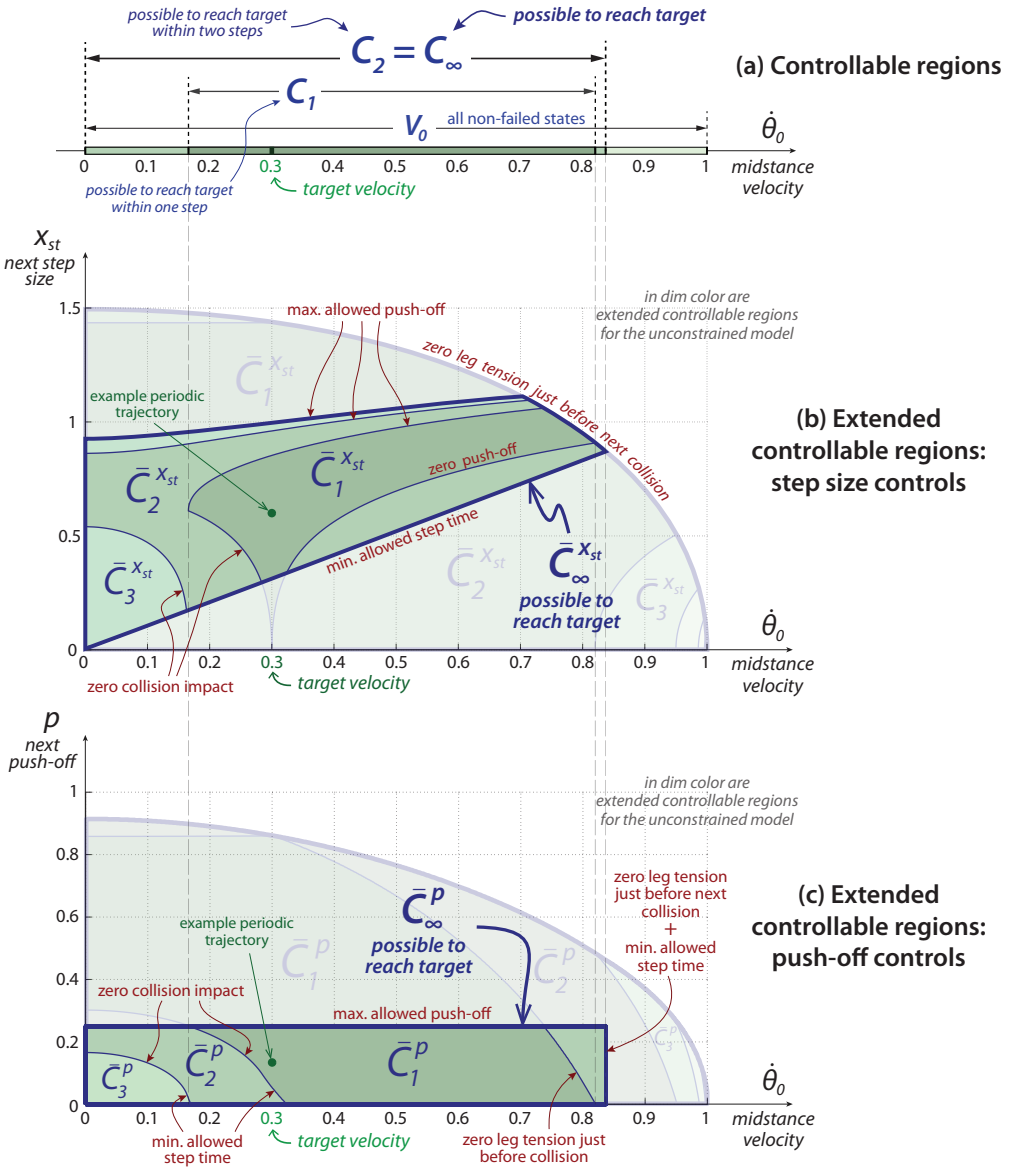


Figure 3.6: n-step controllability of the 2D IP model with limited actuation. This figure is analogous to Fig. 3.4 on page 82, but assumes limited actuation of the planar IP model. The limitations are: the maximum allowed push-off ($p \leq p_{max}$) and the minimum allowed step time ($t_{st} \geq t_{st,min}$, time from midstance to heel-strike). Constrained step time is a proxy for swing-leg actuator limitations. We use $p_{max} = 0.25$, $t_{st,min} = 0.5$. For our constrained model, $C_2 = C_\infty$, $\bar{C}_3^{x_{st}} = \bar{C}_\infty^{x_{st}}$, and $\bar{C}_3^{p} = \bar{C}_\infty^{p}$. The extended controllable regions for the unconstrained IP model (see Fig. 3.4) are shown in dim color for comparison.

We numerically compute the extended controllable regions $\bar{C}_n^{x_{st}}$ for the constrained model by (almost) the same numerical calculations we used for the *un*constrained IP model in Sections 3.2.2 and 3.2.3 above. The only required change in the calculations is in the computation of the extended controllability map $\hat{\bar{C}}(\dot{\theta}_t)$ (i.e. the extended 1-step controllable region) for an arbitrary target velocity $\dot{\theta}_t$ (see discussion in Section 2.7.5 on page 61). All regions $\bar{C}_n^{x_{st}}$ for $n \geq 2$ are then computed with the same calculations as in Section 3.2.3, based on the map $\hat{\bar{C}}(\dot{\theta}_t)$.

Consider the example target velocity (3.17). The region $\bar{C}_1^{x_{st}}$ for the constrained model includes all of the corresponding unconstrained region (see Fig. 3.2b), except for the points $(\dot{\theta}_0, x_{st})$ for which the bounds (3.31) cannot be satisfied (i.e. for which the robot cannot reach the target in one step and obey (3.31)). Notice, that for given $\dot{\theta}_0$ and x_{st} both the push-off p and step time t_{st}, such that the target is reached in one step, are unique: p is defined by the Poincaré map (3.14), t_{st} by numerical integration of the ODE (3.1) until the instant of collision.[3] Thus, for any point $(\dot{\theta}_0, x_{st})$ we can determine whether the actuation limits (3.31) are met. Discretizing the $(\dot{\theta}_0, x_{st})$-plane and removing from the unconstrained region all points violating (3.31), we approximately find the region $\bar{C}_1^{x_{st}}$ for the constrained model. The region is shown on Fig. 3.6b.

The same procedure can be used to compute the region $\bar{C}_1^{x_{st}}$, i.e. the map $\hat{\bar{C}}(\dot{\theta}_t)$, for any target, different from (3.17). Therefore, we follow the calculations in Section 3.2.3 to numerically calculate the n-step regions $\bar{C}_n^{x_{st}}$ for our constrained model. The regions $\bar{C}_n^{x_{st}}$ for the target velocity (3.17) are shown on Fig. 3.6b. Similarly, we compute the extended regions $\bar{C}_n^{p}$, which are shown on Fig. 3.6c. The n-step controllable regions C_n are projections of the extended re-

[3] We use the *ode45* solver in Matlab for numerical integration.

gions $\bar{C}_n^{x_{st}}$ (or $\bar{C}_n^p$) onto the velocity axis; C_n can be found on Fig. 3.6a.

According to our numerical results, $\bar{C}_n^{x_{st}} = \bar{C}_3^{x_{st}}$ for all $n \geq 3$. Therefore, $\bar{C}_3^{x_{st}}$ is equal to the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$ (similarly for $\bar{C}_3^p$ and $\bar{C}_\infty^p$):

$$\bar{C}_\infty^{x_{st}} = \bar{C}_3^{x_{st}}, \qquad \bar{C}_\infty^p = \bar{C}_3^p. \tag{3.32}$$

At the same time, the convergence is even 'faster' for the n-step controllable regions C_n. As one can see from Fig. 3.6a, the ∞-step controllable region C_∞ is the same as C_2:

$$C_\infty = C_2. \tag{3.33}$$

That is, the robot can always reach the target in two or fewer steps, when it is possible to reach the target at all. Result (3.33) supports our 'Two-step controllability' claim in Section 2.3.2 on page 26. The claim holds true for the extended controllability as well: the regions $\bar{C}_2^{x_{st}}$ and $\bar{C}_2^p$ occupy most of their respective ∞-step regions $\bar{C}_\infty^{x_{st}}$ and $\bar{C}_\infty^p$, as one can see from Figs. 3.6b and 3.6c.

Enforcement of the actuation limits (3.31) significantly reduces the size of the extended controllable regions (the regions for the unconstrained model are shown in dim color on Fig. 3.6). That is, there is a smaller choice of control strategies, which bring the robot to the target — this is expected, because constraints (3.31) directly restrict available actuation in the robot. However, these constraints have a smaller effect on the controllability of the robot, i.e. on the ∞-step controllable region C_∞. Only in extreme situations (large velocities $\dot{\theta}_0 > 0.84$) the robot cannot return to the target (or even avoid a failure) due to the insufficient actuation. We note though, that even without a large effect on controllability, limited actuation may worsen other important characteristics of a walking controller, such as robustness and stability. For example, in Section 6.2.3 we argue that smaller extended controllable regions may imply a less robust controller,

because control might have to be near one of its limits. More detailed discussion of different properties of controllers and their connection with controllable regions can be found in Section 6.2 on page 143, where we design a walking controller for Cornell Ranger robot.

3.3 On robustness of passive dynamics

Passive walkers [51, 20] (Section 1.1.1) have only a small basin of attraction around their periodic trajectories [20, 75]. Actuated robots based on passive-dynamics share poor robustness with their fully passive parents and do not walk reliably even on level ground [22]. Now we use the controllable regions of the IP model to compare robustness of passive walkers with what might be possible with actively controlled robots.

Schwab and Wisse [75] studied the *'Simplest Walking (SW) model'* — a 2D passive walker with two rigid legs and a point mass at the hip and at each foot — which was first described by Garcia et al. [27]. The foot mass is assumed significantly smaller than the mass at the hip. The SW (Schwab-Wisse or Simplest-Walking) model has stable periodic gaits on slopes below $\gamma \approx 0.019$ rad. Schwab and Wisse computed the basin-of-attraction of a specific periodic trajectory of the SW model, which we show here in Fig. 3.7.[4] The basin-of-attraction includes all initial velocities and step-lengths, such that the SW model asymptotically approaches the periodic trajectory (the 'target trajectory'). In Fig. 3.7 we also show

[4] We copied the basin of attraction from Fig. 4 in [75] and scaled it appropriately to match the axes of our Fig. 3.7. The axes of the original figure in [75] are the angle and the sum of the angle and angular-rate of the stance leg just after collision; the axes in our figure are the step-size (a function of the angle at collision) and the angular rate at the previous midstance. The copying and scaling was done approximately and only aims to show the qualitative result about passive-locomotion robustness, as presented in Fig. 3.7.

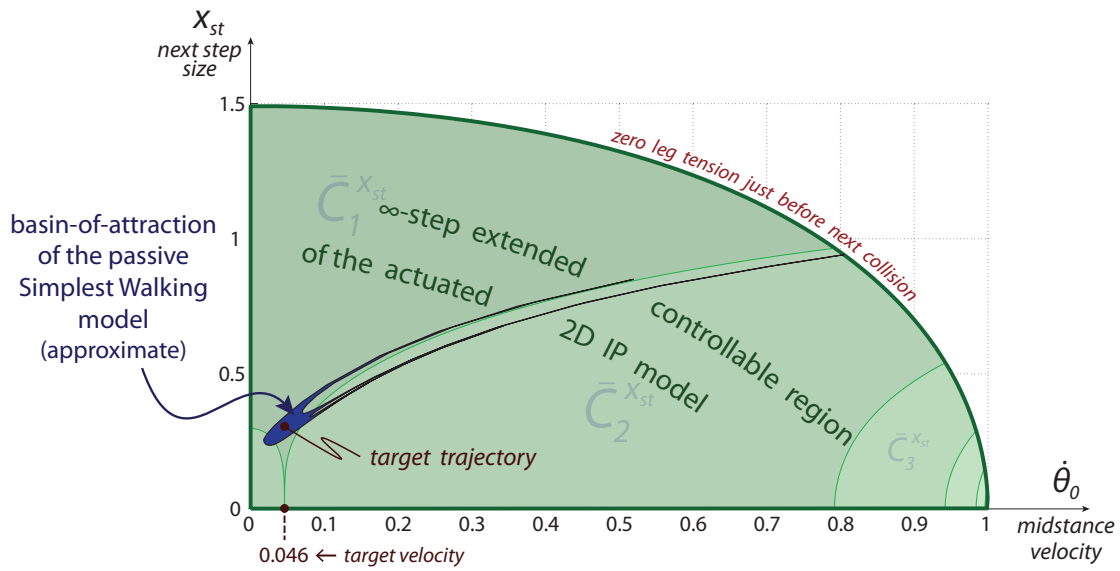


Figure 3.7: Comparative robustness of passive walkers and 2D IP model. For the (passive) Simplest Walking model [27], a stable periodic trajectory at the slope $\gamma = 0.004$ is considered. For the (actuated) 2D IP model with unbounded actuation, the midstance velocity matching that of the Simplest-Walking-model's periodic trajectory, is assumed as the target. Shown are the approximate basin-of-attraction[4] for the Simplest-Walking model (the narrow v-shaped region) and the ∞-step extended controllable region $\bar{C}_\infty^{x_{st}}$ of the IP model (similar to $\bar{C}_\infty^{x_{st}}$ in Fig. 3.4b). The Simplest-Walking-model's basin-of-attraction is only about 2% of $\bar{C}_\infty^{x_{st}}$ in area. This illustrates poor robustness of passive walkers, and passive-dynamics based controllers, relative to the full conceivably possible recovery ability of actively controlled robots.

the ∞-step controllable region $\bar{C}_\infty^{x_{st}}$ of the 2D IP model, with unbounded step-size and push-off actuation, corresponding to the same target trajectory (target velocity at midstance).

The basin-of-attraction of the passive SW model is significantly smaller than the region $\bar{C}_\infty^{x_{st}}$ (about 0.6% of $\bar{C}_\infty^{x_{st}}$ in area). Hence the poor robustness of passive walkers: even a small perturbation may push the robot outside of the basin-of-attraction and make it fall. On the other hand, a lot of such perturbations that make the passive walker fall, could otherwise be controlled with proper actuation — the push-off and step-size actuation in the IP model. This indicates the disadvantage, at least from the robustness perspective, of using passive-

dynamics based controllers, such as in [22], which aim to reproduce motions of passive walkers on level ground.

CHAPTER 4

LINEAR INVERTED PENDULUM MODEL IN 2D

This Chapter is analogous to the Chapter 3, but studies a different simple model of a biped, the Linear Inverted Pendulum (LIP) model in 2D [40, 41]. The model is introduced in Section 2.4.2 and illustrated in Fig. 2.5b on page 29. The model has one dynamic variable, the horizontal position x_h of the hip relative to the stance foot, and two phase coordinates, position x_h and velocity $\dot{x}_h$. There are two controls per each step, which define when and where to take a step: the step size x_{st} and step-time t_{st} (time from the vertical position at midstance to the next heelstrike). It is assumed that the (massless) swing leg can be instantaneously moved into any desired position without affecting the dynamics of the stance leg. Compared to the IP model, the LIP model does not have a preemptive push-off; the force in the stance leg is constrained to maintain the constant height y_G of the hip above the ground.

In this Chapter we derive equations of motion and walking constraints of the 2D LIP model (Section 4.1). We then compute viable, controllable, and extended viable and controllable regions, both for the unconstrained model and in the case of additional constraints imposed (Section 4.2). Based on the calculation results, we justify for the case of the 2D LIP model the 'Viable is Controllable' and 'Two-step controllability' claims stated in Section 2.3 on page 25. Section 4.3 compares the planar IP and LIP models, their dynamics and controls.

4.1 Equations of motion

The constant height constraint of the LIP model leads to linear ODEs governing the motion of the system, making them easier to work with than the nonlinear IP equations. The compressive force F_{st} in the stance leg, which maintains the constant height, is always proportional to the leg's total length. In effect, the stance leg is a zero-rest-length compressive spring always obeying the equation

$$F_{st} = -kl, \qquad \text{where } k = \frac{mg}{y_G}. \tag{4.1}$$

l is the total length of the leg, m the mass at the hip, g acceleration due to gravity, and k the spring constant. Hence, the linearity of the ODEs:

$$m\ddot{\vec{r}}_h = k\vec{r}_h + m\vec{g}, \tag{4.2}$$

where $\vec{r}_h = (x_h, y_h)^T$ is the position vector of the hip relative to the stance foot, and $\vec{g} = (0, -g)^T$. Assuming the initial conditions $y_h = y_G = mg/k$ and $\dot{y}_h = 0$, we get

$$\ddot{y}_h = 0,$$

i.e. the hip remains at the fixed height y_G at all times. Therefore, the x-component of equation (4.2) is the only ODE governing the dynamics of the system. We rewrite (4.2) in a non-dimensional form as

$$\ddot{x}_h = x_h, \qquad y_h = 1. \tag{4.3}$$

The non-dimensionalization uses the constants m, y_G, and $\sqrt{m/k}$ for the units of mass, length, and time respectively.

In contrast with the ODE (3.1) governing the motion of the planar IP model, equation (4.3) for the planar LIP model can be solved analytically:

$$x_h(t) = x_0 \cosh(t - t_0) + v_0 \sinh(t - t_0), \tag{4.4}$$

where x_0 and v_0 are the initial position and velocity of the hip, and t_0 the initial time. The above equation defines the trajectory of the robot between collisions.

The instant and location of the next collision are defined by the control parameters step-time t_{st} and step-size x_{st}. All 'collisions' are smooth in that the hip velocity is continuous at all times (in contrast to velocity jumps at heel-strikes in the IP model). The legs swap their roles at each collision: the (former) swing leg becomes the (new) stance leg, and vice versa. Thus, we write the change of the phase variables x_h and $\dot{x}_h$ at the instant of collision as

$$x_h^+ = x_h^- - x_{st}, \tag{4.5a}$$

$$\dot{x}_h^+ = \dot{x}_h^-. \tag{4.5b}$$

Here superscripts $^-$ and $^+$ refer to the values just before and just after the collision respectively.

4.1.1 Walking constraints

The constraints of the IP model in Chapter 3 prevent the robot transitioning to the flight phase. However, for the LIP model flight (i.e. running) is not possible because of the constant-CoM-height constraint; the robot is always in the walking mode. Instead, the motion of the LIP model is restricted by the maximum allowed length of each leg, l_{max}.

The length l of the stance leg is given by

$$l = \sqrt{x_h^2 + y_h^2} = \sqrt{x_h^2 + 1}. \tag{4.6}$$

Therefore, we write the maximum-leg-length constraint as

$$x_h^2 + 1 \leq l_{max}^2 \tag{4.7a}$$

The above inequality has to be satisfied at all times. The length of the swing leg is only relevant at the instant just before the heel-strike, when the robot has to take a step. The swing leg just before the heel-strike keeps its length as it becomes the stance leg just after the heel-strike. Therefore, the leg-length constraint for the swing leg is covered by (4.7a).

We note, that in the absence of the constraint (4.7a) (and any other constraints restricting the legs' extension, such as telescopic-actuator limitations or maximum allowed step duration), there are no restrictions on the duration of application of equation (4.4). Walking (staying up), or even a single step of walking, can be prolonged indefinitely by simply not taking a step. In this case, both the stance leg length and hip velocity increase exponentially.

As with the 2D IP model, we only consider the motion forward:

$$\dot{x}_h \geq 0. \tag{4.7b}$$

We also require that *the robot reaches the midstance at the end of each step.* The corresponding algebraic constraints are derived in Section 4.1.2 below. One or more additional constraints may be imposed on the model, based on a specific problem. Some examples of such constraints are listed in Section 2.5.1 on page 33.

4.1.2 Poincaré map

As for the IP model, we use a Poincaré section for the 2D LIP model at *midstance*, when the stance leg is vertical:

$$\textit{Midstance:} \qquad x_h = 0. \tag{4.8}$$

The dynamic state of the robot at midstance can be described by one variable, which we choose to be the hip velocity $\dot{x}_h$. We are going to find the Poincaré map of the model, i.e. how velocity $\dot{x}_h$ changes from one midstance to another for given control parameters (the step-size x_{st} and step-time t_{st}).

We use the following first integral, also called the orbital energy [41], of the system (4.3):

$$H = \frac{1}{2}\dot{x}_h^2 - \frac{1}{2}x_h^2.$$

H is the total energy of the robot, assuming $x_h = 0$ to be the reference for the potential energy of the spring-like leg. H is conserved between collisions:

$$\dot{H} = \dot{x}_h\ddot{x}_h - x_h\dot{x}_h = (\ddot{x}_h - x_h)\dot{x}_h = 0.$$

We write conservation of H between the initial midstance and the instant just before the collision:

$$v_0^2 = (\dot{x}_h^-)^2 - (x_h^-)^2, \tag{4.9a}$$

where v_0 is the initial midstance velocity. Similarly, for the interval between just after the collision and the next midstance:

$$v_1^2 = (\dot{x}_h^+)^2 - (x_h^+)^2, \tag{4.9b}$$

where v_1 is the velocity at the next midstance. The position x_h^- just before the collision is found by writing the analytical solution (4.4) at the time t_{st} and setting $t_0 = 0$:

$$x_h^- = v_0 \sinh t_{st} \tag{4.10}$$

Finally, elimination of x_h^-, $\dot{x}_h^-$, x_h^+, and $\dot{x}_h^+$ from the five equations (4.5), (4.9), and (4.10) allows us to find the Poincaré map of the model:

$$v_0^2 = v_1^2 + x_{st}^2 - 2v_0x_{st}\sinh t_{st}. \tag{4.11}$$

This equation establishes the relationship between the initial velocity v_0, the end velocity of the step v_1, and the controls x_{st} and t_{st} used during the step. The equations of motion and step-to-step transitions of the LIP model (both the planar model and its 3D extension) were derived by Kajita, et al. [41, 40].

Next, we write the model's constraints (4.7) in terms of the variables v_0, v_1, x_{st}, and t_{st}. The maximum-leg-length constraint (4.7a) has to be checked only just before and just after the heel-strike, when the leg is most stretched:

$$(x_h^-)^2 + 1 \leq l_{max}^2, \qquad (x_h^+)^2 + 1 \leq l_{max}^2.$$

Note, that the second of the above inequalities corresponds to the leg-length constraint for the swing leg (the stance leg just after the heel-strike is the swing leg just before the heel-strike). Employing equation (4.10) for x_h^- and the collision relation (4.5a), the inequalities become

$$v_0 \sinh t_{st} \leq x_{max}, \tag{4.12a}$$

$$v_0 \sinh t_{st} \geq x_{st} - x_{max}, \tag{4.12b}$$

where x_{max} is the horizontal deviation of the hip from the midstance position when the stance leg is maximally stretched ($l = l_{max}$):

$$x_{max} = \sqrt{l_{max}^2 - 1}.$$

We require that the robot reaches the midstance at the end of each step. This is guaranteed by two conditions. First, the robot has to step ahead of the hip: $x_{st} \geq x_h^-$. Using equation (4.10) we write this constraint as

$$v_0 \sinh t_{st} \leq x_{st}. \tag{4.12c}$$

Second, the robot's speed after the collision has to be sufficiently large. Equivalently, the Poincaré map (4.11) has to have a real solution with respect to the next midstance velocity v_1.

We consider only the motion 'forward' and allow only non-negative values of the step-time t_{st} and step-size x_{st}:

$$v_0 \geq 0, \tag{4.12d}$$

$$t_{st} \geq 0. \tag{4.12e}$$

$$x_{st} \geq 0,$$

Note, that the non-negativity of the step-size ($x_{st} \geq 0$) is followed from the stepping-ahead constraint (4.12c).

Thus, a step of our 2D LIP model is defined by three values: the initial mid-stance velocity v_0, step-size x_{st}, and step-time t_{st}. The velocity v_1 at the next midstance is determined by the Poincaré map (4.11). For a given v_0, the controls x_{st} and t_{st} have to be such that constraints (4.12) are satisfied. If at least one of the constraints is violated, the robot fails.

4.2 Viable and controllable regions

In this section we compute the n-step viable and controllable regions V_n and C_n, and their extended counterparts $\bar{V}_n$ and $\bar{C}_n$ for our 2D LIP model. The definition and detailed description of the (extended) viable and controllable regions was given in Chapter 2.

Pratt et al. [65, 42] found the capture regions of the LIP model, both in 2D and 3D: all ground locations where the robot can step and come to a stop in one or more steps. Our calculations in this section extend, for the 2D case of the LIP model, Pratt's results in two ways. First, we allow an arbitrary target velocity, instead of just standing up-right assumed by capturability. Second, as opposed

to all controls (stepping locations) that allow a return to the target for a given perturbed state of the robot, we also find all perturbed states for which such return is possible, i.e. the (extended) controllable regions of the robot.

The planar LIP model has one state variable at midstance (initial hip velocity v_0) and two controls (step-time t_{st} and step-size x_{st}). The full extended state of the model is three-dimensional, (v_0, t_{st}, x_{st}), yielding three-dimensional full extended viable and controllable regions $\bar{V}_n^{t_{st},x_{st}}$ and $\bar{C}_n^{t_{st},x_{st}}$. However, as for the 2D IP model in Section 3.2, we extend the state space by only one control at a time, to keep the calculations simple. We consider the two-dimensional extended regions $\bar{V}_n^{x_{st}}$, $\bar{C}_n^{x_{st}}$, $\bar{V}_n^{t_{st}}$, and $\bar{C}_n^{t_{st}}$.

We first describe all non-failed states and feasible controls of the model, i.e. the 0-step regions V_0, $\bar{V}_0^{x_{st}}$, and $\bar{V}_0^{t_{st}}$. In contrast to the IP model, the midstance velocity is allowed to be arbitrarily large, because flight is not possible for the LIP model:

$$V_0 : \qquad v_0 \geq 0. \tag{4.13a}$$

The step-size x_{st} is limited by the maximum allowed length of the legs l_{max}. The largest possible x_{st} corresponds to the configuration of the robot when both the stance and swing leg are maximally stretched:

$$\bar{V}_0^{x_{st}} : \qquad v_0 \geq 0, \quad 0 \leq x_{st} \leq 2x_{max}, \tag{4.13b}$$

where $x_{max} = \sqrt{l_{max}^2 - 1}$. For now, we do not put a lower bound on the step-time t_{st}:

$$\bar{V}_0^{t_{st}} : \qquad v_0 \geq 0, \quad t_{st} \geq 0. \tag{4.13c}$$

Controllability of the model with the minimum allowed step-time constraint, such as constraint (3.31b) for the planar IP model, is analyzed in Section 4.2.4.

The 0-step viable and extended viable regions V_0, $\bar{V}_0^{x_{st}}$, and $\bar{V}_0^{t_{st}}$ are shown in Fig. 4.4 on page 113. The example value of the maximum leg length l_{max} considered here is

$$l_{max} = 1.2 \tag{4.14}$$

4.2.1 one-step controllability

The extended controllable regions for our LIP model can be computed in a similar way as for the planar IP model in Section 3.2. First, we compute the extended 1-step controllable regions $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{t_{st}}$ for an arbitrary target state of the model, i.e. an arbitrary target velocity v_t at midstance.

For the LIP model the regions $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{t_{st}}$ can be found analytically. $\bar{C}_1^{x_{st}}$ is all initial velocities v_0 and step-sizes x_{st}, such that the robot can, with an appropriate step-time t_{st}, reach the target velocity v_t at the next midstance. Hence, $\bar{C}_1^{x_{st}}$ is all combinations (v_0, x_{st}) that satisfy the Poincaré map equation (4.11) and constraints (4.12) for some value of t_{st} and the fixed next-step velocity $v_1 = v_t$.

We find the region $\bar{C}_1^{x_{st}}$ by computing its boundaries. The boundary points correspond to the cases when one of the constraints (4.12) is 'active', i.e. turns into an equality. We use the Poincaré map (4.11) to eliminate the variable t_{st} from the constraint equations and write the equations of the corresponding bound-

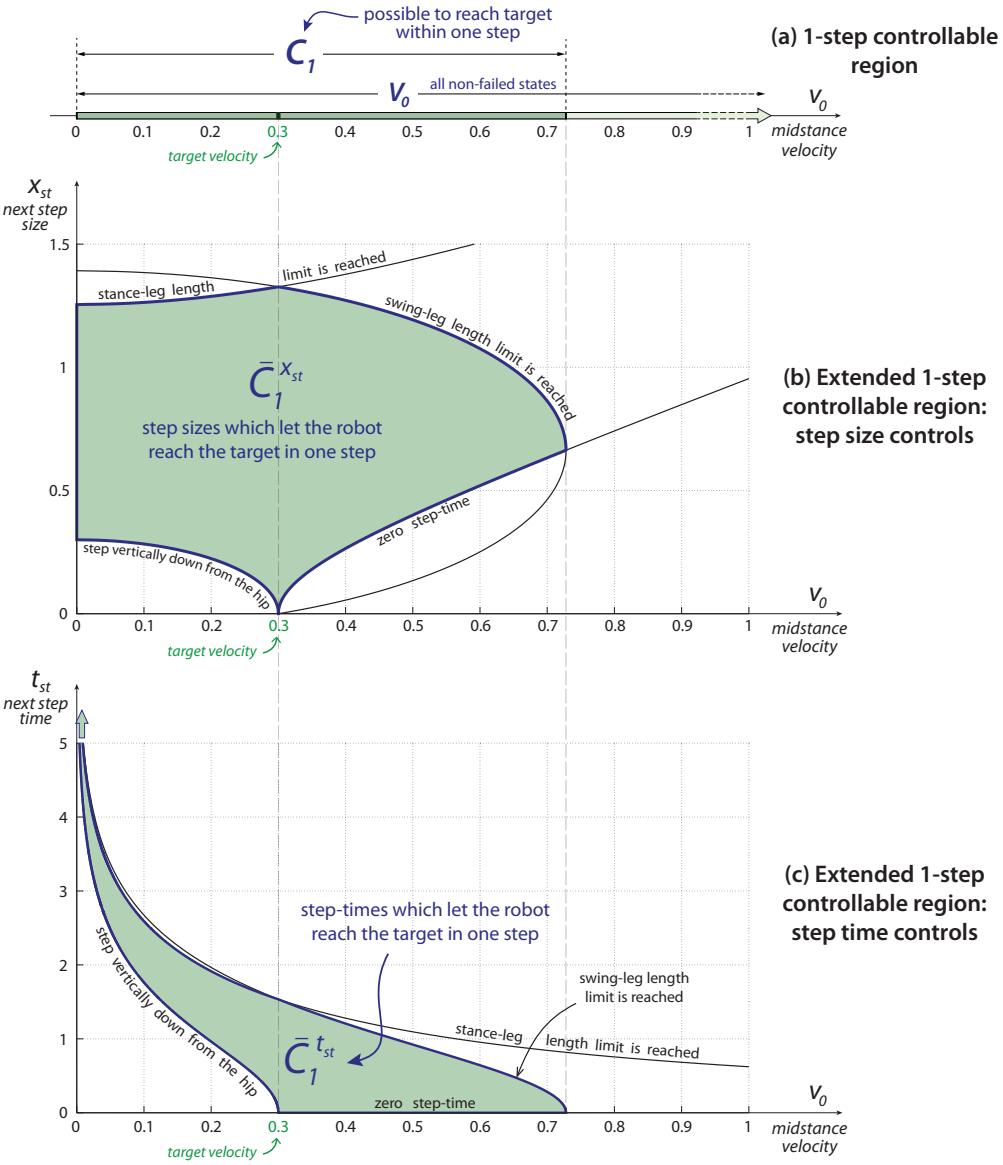


Figure 4.1: 1-step controllable and extended controllable regions for the 2D LIP model. **(a)** The 1-step controllable region C_1 is all initial velocities v_0, for which the robot can reach the target within one step. The target here is the velocity $v_t = 0.3$ at midstance. **(b), (c)** For each velocity v_0, the extended 1-step controllable regions $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{t_{st}}$ show, respectively, all step-sizes x_{st} and step-times t_{st}, that let the robot reach the target within one step. The boundary curves of $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{t_{st}}$ correspond to different constraints of the model. For step sizes/step-times beyond these curves, the robot either fails or does not reach the target. The region C_1 is the projection of both $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{t_{st}}$ onto the v_0-axis.

aries:

$$v_0^2 = v_t^2 + x_{st}^2 - 2x_{max}x_{st}, \tag{4.15a}$$

$$v_0^2 = v_t^2 - x_{st}^2 + 2x_{max}x_{st}, \tag{4.15b}$$

$$v_0^2 = v_t^2 - x_{st}^2, \tag{4.15c}$$

$$v_0 = 0, \tag{4.15d}$$

$$v_0^2 = v_t^2 + x_{st}^2. \tag{4.15e}$$

The region $\bar{C}_1^{x_{st}}$ is shown in Fig. 4.1b for the example target velocity

$$v_t = 0.3. \tag{4.16}$$

From the top, the region is bounded by the curves (4.15a) and (4.15b) representing the maximum leg-length constraints (4.12a) and (4.12b) respectively. For the step-sizes and initial velocities on these curves, the robot has to maximally extend its either stance or swing leg (and use an appropriate step-time t_{st}) to return to the target in one step. The bottom-left boundary of $\bar{C}_1^{x_{st}}$ is the curve (4.15c) representing the stepping-ahead constraint (4.12c). For the corresponding points, the robot steps exactly below the hip at an appropriate time t_{st}. Finally, the bottom-right boundary curve (4.15e) represents the non-negative step-time constraint (4.12e) and corresponds to infinitely fast steps. The robot puts the swing foot on the ground immediately after the initial midstance, at the location defined by the step-size x_{st}. For any point (v_0, x_{st}) outside the region $\bar{C}_1^{x_{st}}$, the robot either fails or is not able to reach the target in one step.

Similarly, for each initial velocity v_0 the extended 1-step controllable region $\bar{C}_1^{t_{st}}$ shows all step-times t_{st}, such that the robot can reach the target v_t in one step. As for the region $\bar{C}_1^{x_{st}}$ discussed above, we find the boundaries of $\bar{C}_1^{t_{st}}$ by using the Poincaré map (4.11) to eliminate x_{st} from the five constraint equations

(4.12). Then, we consider the cases when the constraints turn into equalities. The resulting equations define the boundary curves of the region $\bar{C}_1^{t_{st}}$:

$$v_0 \sinh t_{st} = x_{max}, \tag{4.17a}$$

$$v_0 \cosh t_{st} = \sqrt{x_{max}^2 + v_t^2}, \tag{4.17b}$$

$$v_0 \cosh t_{st} = v_t, \tag{4.17c}$$

$$v_0 = 0, \tag{4.17d}$$

$$t_{st} = 0. \tag{4.17e}$$

The region is shown in Fig. 4.1c for the target velocity (4.16). Similarly to $\bar{C}_1^{x_{st}}$, the region $\bar{C}_1^{t_{st}}$ is bounded from above by the maximum-leg-length boundaries (4.17a) and (4.17b). From below, $\bar{C}_1^{t_{st}}$ is constrained by the zero-step-time line (4.17e) (i.e. the v_0-axis) and the stepping-below-the-hip curve (4.17c). As the initial velocity v_0 approaches zero, the allowable step-times t_{st} grow infinitely large.

The projections of both $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{t_{st}}$ onto the v_0-axis form the 1-step controllable region C_1 shown in Fig. 4.1a. C_1 is all initial velocities, for which there is at least one pair of controls x_{st} and t_{st} such that the robot has the target velocity v_t at the next midstance. For the target velocity (4.16) the range of 1-step controllable initial velocities is $0 < v_0 \leq \sqrt{x_{max}^2 + v_t^2} \approx 0.73$.

We also show examples of the 1-step regions C_1, $\bar{C}_1^{x_{st}}$, and $\bar{C}_1^{t_{st}}$ for target velocities v_t, other than (4.16). Fig. 4.2 shows these regions for the cases $v_t = 0$ (plots (a), (c), (e) on the left) and $v_t = 0.8$ (plots (b), (d), (f) on the right). In the zero-target-velocity case, the n-step controllable regions C_n are equivalent to Koolen's n-step viable capture basins [42].

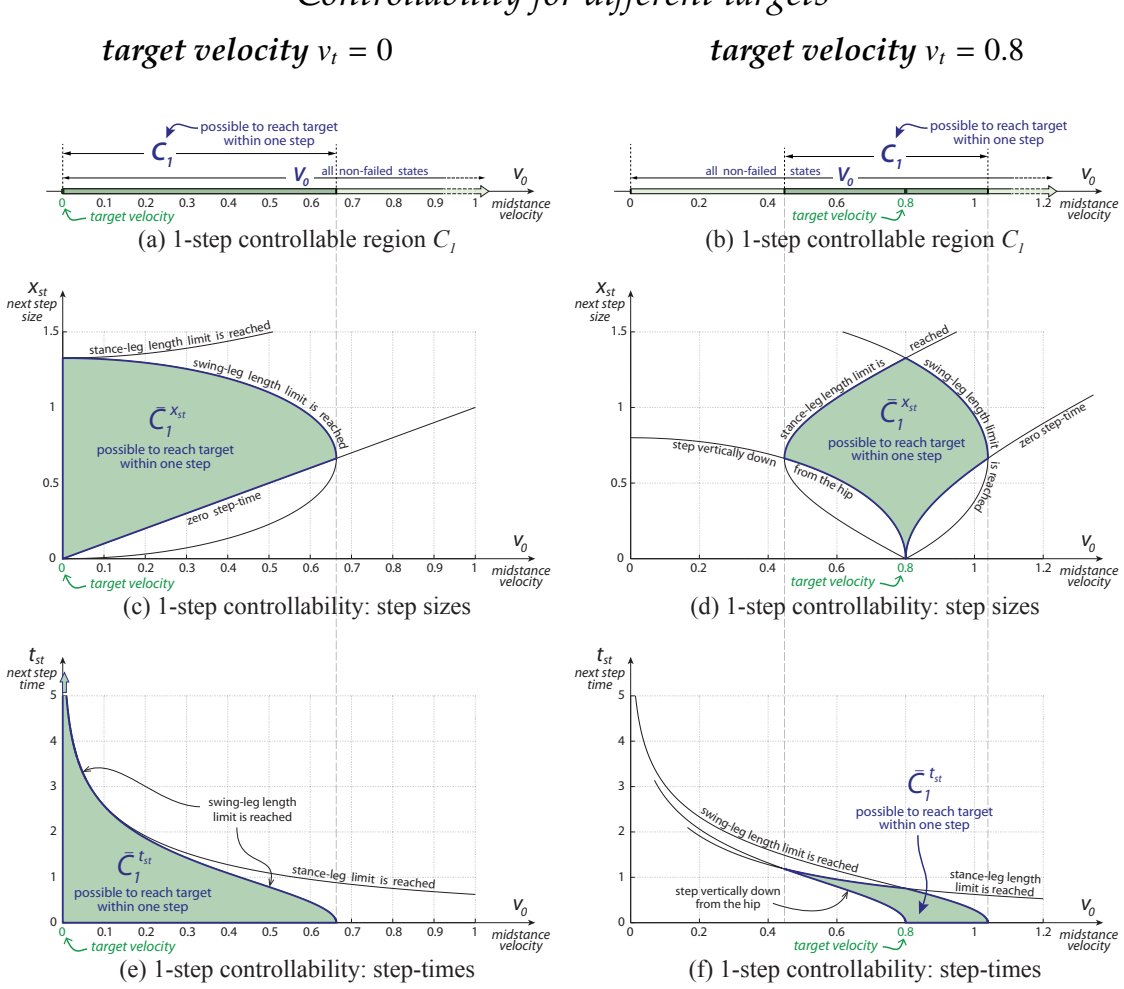


Figure 4.2: 1-step controllability of the 2D LIP model for different targets. This figure is analogous to the Fig. 4.1 on page 105, but for different target velocities v_t. The graphs (a), (c), (e) assume $v_t = 0$, while the graphs (b), (d), (f) consider $v_t = 0.8$. In the case $v_t = 0$, the 1-step controllable region C_1 is equivalent to Koolen's 1-step viable-capture basin [42]. Fig. 4.1, where the target is $v_t = 0.3$, would be a column of figures between the two columns on this Fig. 4.2.

4.2.2 *n*-step and ∞-step controllability

The extended n-step controllable regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ are all combinations of the initial velocity v_0 and the next-step control (step-size x_{st} and step-time t_{st} correspondingly), such that the robot can reach a given target velocity v_t in n or fewer steps.

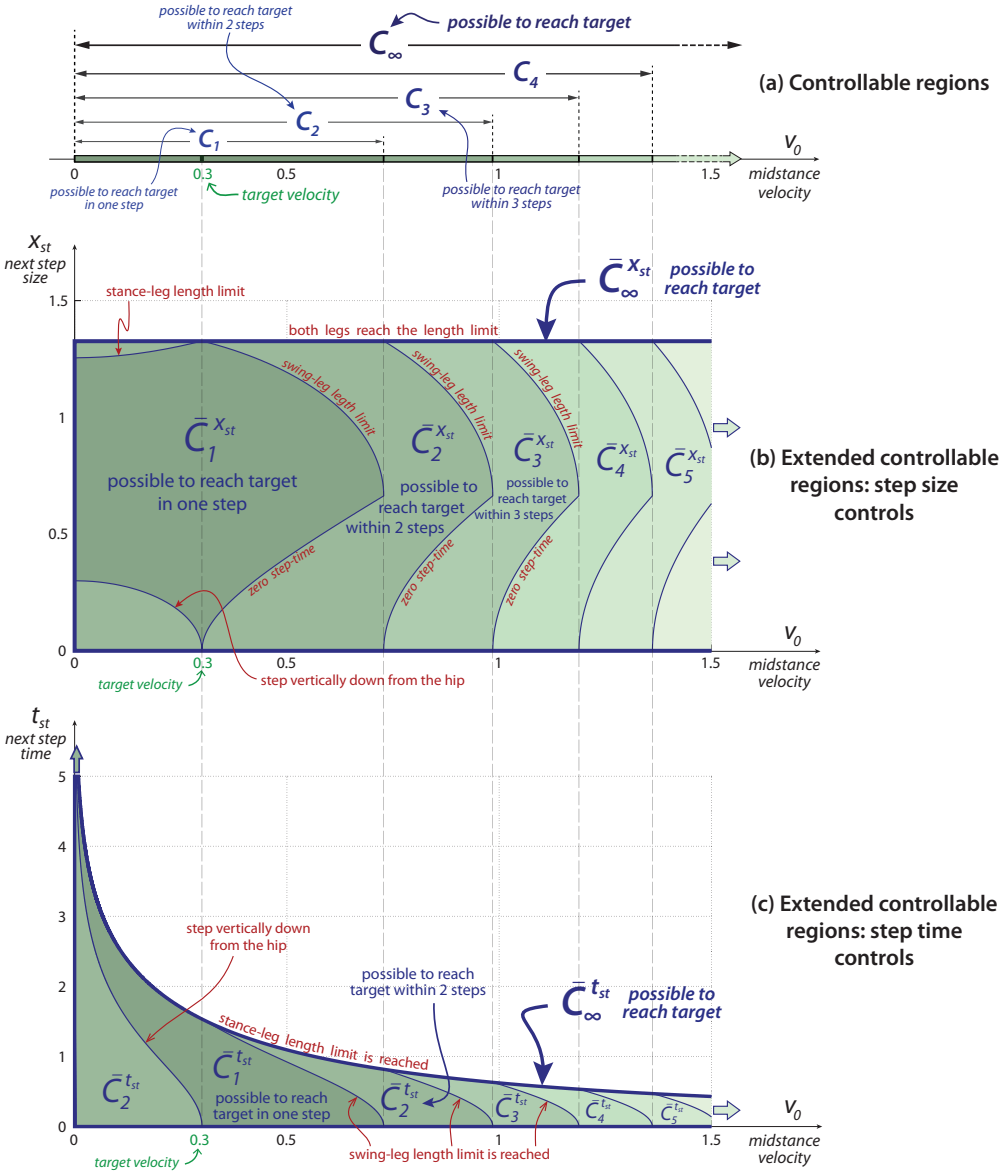


Figure 4.3: n-step controllability of the 2D LIP model. **(a)** The n-step controllable region C_n is all initial velocities v_0, for which the robot can reach the target velocity v_t in n or fewer steps. Here $v_t = 0.3$. **(b), (c)** For each v_0, the extended n-step controllable regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ show, respectively, where and when the robot can take the next step (i.e. all step sizes x_{st} and step-times t_{st}), such that the target is reached in n or fewer steps total. The regions C_n, $\bar{C}_n^{x_{st}}$, and $\bar{C}_n^{t_{st}}$ form sequences of nested regions, which approach the ∞-step regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{t_{st}}$ (shown by their boundary) correspondingly. These are, respectively, all velocities v_0 and combinations (v_0, x_{st}) and (v_0, t_{st}) of velocities and controls such that the target can be reached or approached asymptotically. Each C_n is the projection of both $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ onto the v_0-axis.

As for the IP model in Section 3.2.3, we compute the regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ numerically, following the procedure described in Section 2.7.4 on page 58. The regions are displayed for several values of n in Figs. 4.3b and 4.3c respectively, assuming the target velocity (4.16). The projections of both $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ onto the velocity axis are the n-step controllable regions C_n shown in Fig. 4.3a. For each n, C_n includes all initial velocities that allow the robot to reach the target v_t in at most n steps, given appropriate feasible controls. The 1-step regions C_1, $\bar{C}_1^{x_{st}}$, and $\bar{C}_1^{t_{st}}$ are the same as those presented on Fig. 4.1.

The controllable and extended controllable regions are nested: $C_n \subset C_{n+1}$, $\bar{C}_n^{x_{st}} \subset \bar{C}_{n+1}^{x_{st}}$, and $\bar{C}_n^{t_{st}} \subset \bar{C}_{n+1}^{t_{st}}$. As n increases, the regions grow indefinitely in the v_0 direction. In the limit $n \to \infty$, they approach the ∞-step controllable and extended controllable regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{t_{st}}$ respectively. C_∞ is all velocities v_0 from which a return to the target is possible; such are all positive velocities:

$$C_\infty : \qquad v_0 > 0. \tag{4.18a}$$

For each v_0, the extended region $\bar{C}_\infty^{x_{st}}$ shows all sizes x_{st} of the first step which allow a return to the target. $\bar{C}_\infty^{x_{st}}$ is only bounded from above by the largest step-size the robot can take for a given maximum leg-length l_{max}:

$$\bar{C}_\infty^{x_{st}} : \qquad v_0 > 0, \quad 0 \le x_{st} \le 2x_{max} \approx 1.33. \tag{4.18b}$$

The region $\bar{C}_\infty^{t_{st}}$ considers all ∞-step controllable step-times t_{st}. As shown in Fig. 4.3c, $\bar{C}_\infty^{t_{st}}$ is bounded by the two axes (constraints (4.12d) and (4.12e)) and the curve (4.17a), which corresponds to the stance-leg reaching its length limit l_{max}:

$$\bar{C}_\infty^{t_{st}} : \qquad v_0 > 0, \quad 0 \le t_{st} \le \sinh^{-1} \frac{x_{max}}{v_0}. \tag{4.18c}$$

The controllable regions presented in Fig. 4.3 were numerically computed assuming the target velocity (4.16). However, our calculations for other target

velocities reveal that the ∞-step regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{t_{st}}$ remain the same for any v_t. In Section 4.2.3 below we show that these regions are (almost) equal to the ∞-step viable regions V_∞, $\bar{V}_\infty^{x_{st}}$, and $\bar{V}_\infty^{t_{st}}$ correspondingly.

In contrast to the IP model (see Fig. 3.4 on page 82), the ∞-step controllable and extended controllable regions are unbounded for our LIP model. This is because flying is impossible in the LIP model and steps are allowed to be infinitely fast. The regions become bounded if the step-time t_{st} is constrained by a minimum allowed value — see Section 4.2.4. One consequence of the unbounded ∞-step regions is that they are not close to the corresponding (bounded) 2-step controllable and extended controllable regions (C_2 is not close to C_∞, $\bar{C}_2^{x_{st}}$ to $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_2^{t_{st}}$ to $\bar{C}_\infty^{t_{st}}$) — for most initial velocities, two steps are not enough to return to the target. Hence, the simple LIP model is one example where the 'Two-step controllability' claim (Section 2.3.2) is not valid — in contrast to the planar IP model for which the claim is true for most target velocities.

Notice that more than two steps are required to return to the target only for large initial velocities, i.e. those outside C_2, $v_0 \gtrsim 0.98$ (approximately 3 m/s for a biped with one-meter-long legs). Such velocities are beyond the normal range of walking of most bipeds; for example, for the IP model viable midstance velocities of walking are $\dot{\theta}_0 \leq 1$ (see Section 3.2.1). If we consider only 'not-too-large' velocities (e.g. $v_0 \leq 1$, matching the IP model range), the 'Two-step controllability' claim becomes valid for our LIP model. The claim is also justified for the LIP model if the minimum-step-time constraint is imposed on the model, as discussed in Section 4.2.4 below.

4.2.3 Viable and extended viable regions

The viable and extended viable regions V_n, $\bar{V}_n^{x_{st}}$, and $\bar{V}_n^{t_{st}}$ are the sets of all velocities and next-step controls, such that the robot can take n or more steps and not fail.

For any n, the extended n-step viable region $\bar{V}_n^{x_{st}}$ includes all points in the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$ for any target that allows periodic motion. The target (4.16) considered above does allow periodic motion: for example, $v_0 = v_1 = v_t$ and $x_{st} = x_{max}$ satisfy the Poincaré map (4.11) and constraints (4.12). Therefore, the region $\bar{V}_n^{x_{st}}$ includes all of $\bar{C}_\infty^{x_{st}}$: $\bar{C}_\infty^{x_{st}} \subset \bar{V}_n^{x_{st}}$. On the other hand, $\bar{V}_n^{x_{st}}$ is always a subset of the 0-step region $\bar{V}_0^{x_{st}}$: $\bar{V}_n^{x_{st}} \subset \bar{V}_0^{x_{st}}$. According to (4.18b) and (4.13b) the regions $\bar{C}_\infty^{x_{st}}$ and $\bar{V}_0^{x_{st}}$ are the same, except for the boundary points $v_0 = 0$. Hence, for the LIP model, all extended n-step viable regions $\bar{V}_n^{x_{st}}$ are equal for $n \geq 0$:

$$\bar{V}_0^{x_{st}} = \bar{V}_n^{x_{st}} = \bar{V}_\infty^{x_{st}} : \qquad v_0 \geq 0, \quad 0 \leq x_{st} \leq 2x_{max} \tag{4.19a}$$

We include the points $v_0 = 0$ in all $\bar{V}_n^{x_{st}}$, because the robot formally can 'step in place' ($v_0 = v_1 = 0, x_{st} = 0, t_{st} = 0$) while taking an arbitrarily large number of steps. The viable regions $\bar{V}_n^{x_{st}}$ are shown in Fig. 4.4b.

We use a similar reasoning to find the regions $\bar{V}_n^{t_{st}}$ of all n-step viable step-times t_{st}. For any n, $\bar{V}_n^{t_{st}}$ includes all points in the region $\bar{C}_\infty^{t_{st}}$ which is shown in Fig. 4.3b and is defined by (4.18c). That is, $\bar{V}_n^{t_{st}}$ contains all points below the curve (4.17a), the top boundary curve of $\bar{C}_\infty^{t_{st}}$. Notice, that the curve (4.17a) (i.e. the stance-leg length limit constraint) does not depend on the target velocity v_t and the step-size x_{st}. Therefore, for any point (v_0, t_{st}) above this curve the robot fails; all such points are outside of the region $\bar{V}_n^{t_{st}}$ for any $n \geq 1$. Thus, all $\bar{V}_n^{t_{st}}$ for

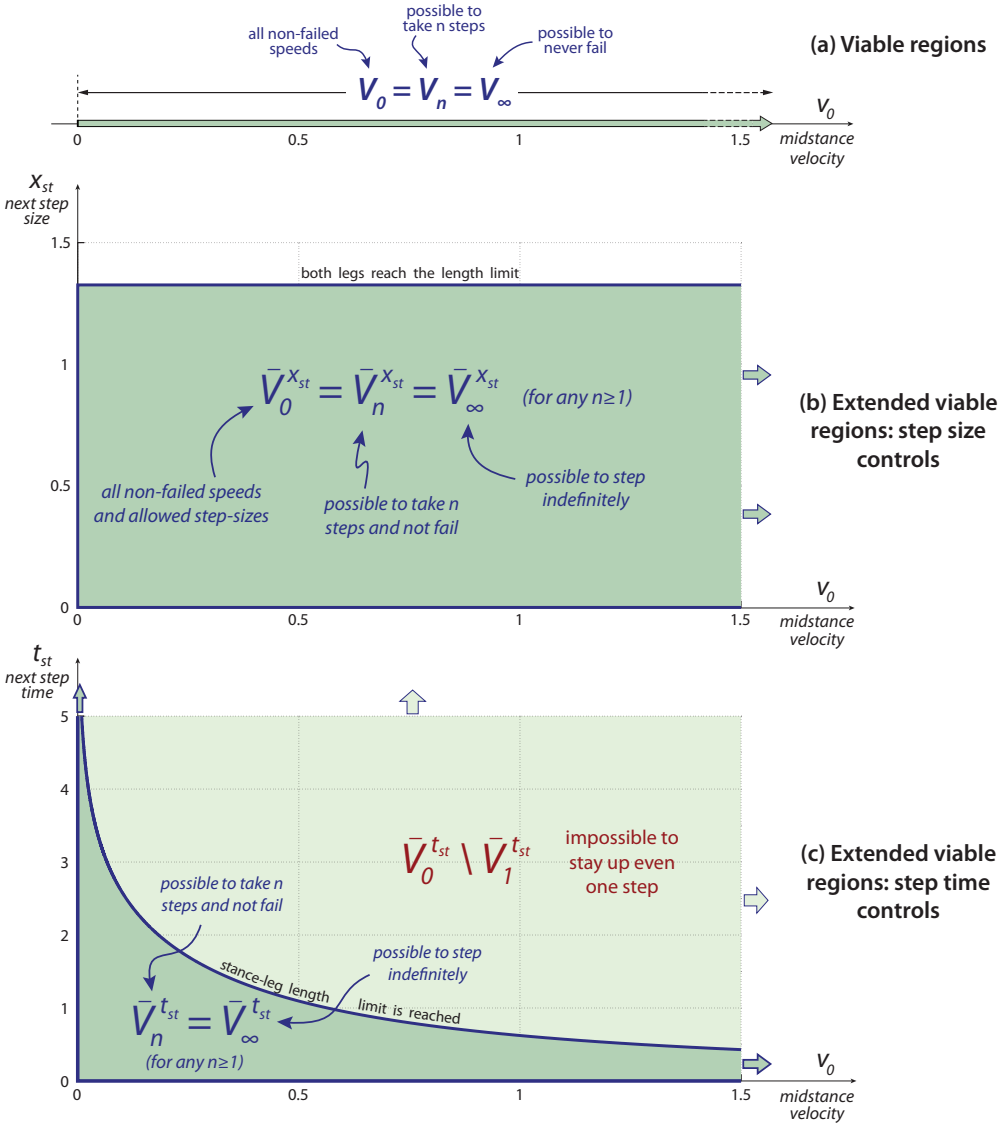


Figure 4.4: Viable and extended viable regions of the 2D LIP model. **(a)** The n-step viable region V_n is all initial velocities v_0, for which the robot can, with appropriate controls, take at least n steps and not fail. **(b), (c)** For each v_0, the extended n-step viable regions $\bar{V}_n^{x_{st}}$ and $\bar{V}_n^{t_{st}}$ show, respectively, all step-sizes x_{st} and step-times t_{st} of the next step, which let the robot take n or more steps and not fail. Each V_n is the projection of both $\bar{V}_n^{x_{st}}$ and $\bar{V}_n^{t_{st}}$ onto the v_0-axis. For the 2D LIP model, the regions V_n are identical for $n \geq 0$; so are, respectively, all $\bar{V}_n^{x_{st}}$ for $n \geq 0$ and all $\bar{V}_n^{t_{st}}$ for $n \geq 1$. The ∞-step regions V_∞, $\bar{V}_\infty^{x_{st}}$, and $\bar{V}_\infty^{t_{st}}$ are shown by their boundary. These are, correspondingly, all velocities v_0 and combinations (v_0, x_{st}) and (v_0, t_{st}) of velocities and controls for which the robot is able to step indefinitely. The 0-step regions V_0, $\bar{V}_0^{x_{st}}$, $\bar{V}_0^{t_{st}}$ represent all non-failed velocities and allowed controls.

$n \geq 1$ are the same as $\bar{C}_{\infty}^{t_{st}}$ (but for the points $v_0 = 0$):

$$\bar{V}_1^{t_{st}} = \bar{V}_n^{t_{st}} = \bar{V}_{\infty}^{t_{st}} : \qquad v_0 \geq 0, \quad 0 \leq t_{st} \leq \sinh^{-1} \frac{x_{max}}{v_0}. \tag{4.19b}$$

The extended n-step viable regions $\bar{V}_n^{t_{st}}$ are shown in Fig. 4.4c.

The n-step viable region V_n is all initial midstance velocities that allow the robot to take n or more steps and not fail. We find the regions V_n as the projections of both the $\bar{V}_n^{x_{st}}$ and $\bar{V}_n^{t_{st}}$ regions onto the velocity axis. As shown in Fig. 4.4a, all V_n are the same:

$$V_0 = V_n = V_{\infty} : \qquad v_0 \geq 0. \tag{4.19c}$$

As established above, the (extended) ∞-step viable and (extended) ∞-step controllable regions of the 2D LIP model are equal, but for some boundary points:

$$V_{\infty} \approx C_{\infty}, \qquad \bar{V}_{\infty}^{x_{st}} \approx \bar{C}_{\infty}^{x_{st}}, \qquad \bar{V}_{\infty}^{t_{st}} \approx \bar{C}_{\infty}^{t_{st}}. \tag{4.20}$$

This confirms, for the 2D LIP case, the 'Viable is Controllable' claim in Section 2.3.1. In most cases, when the robot is able to avoid failing, it is also able to reach any given target.

4.2.4 Swing time limitation

Now we consider the LIP model with restricted step-time. We require the control parameter step-time t_{st} (time from midstance to heel-strike) to be not smaller than a given fixed value:

$$t_{st} \geq t_{st,min} > 0. \tag{4.21}$$

The minimum step-time constraint is a proxy for leg-swinging actuator limitations of the real robot.

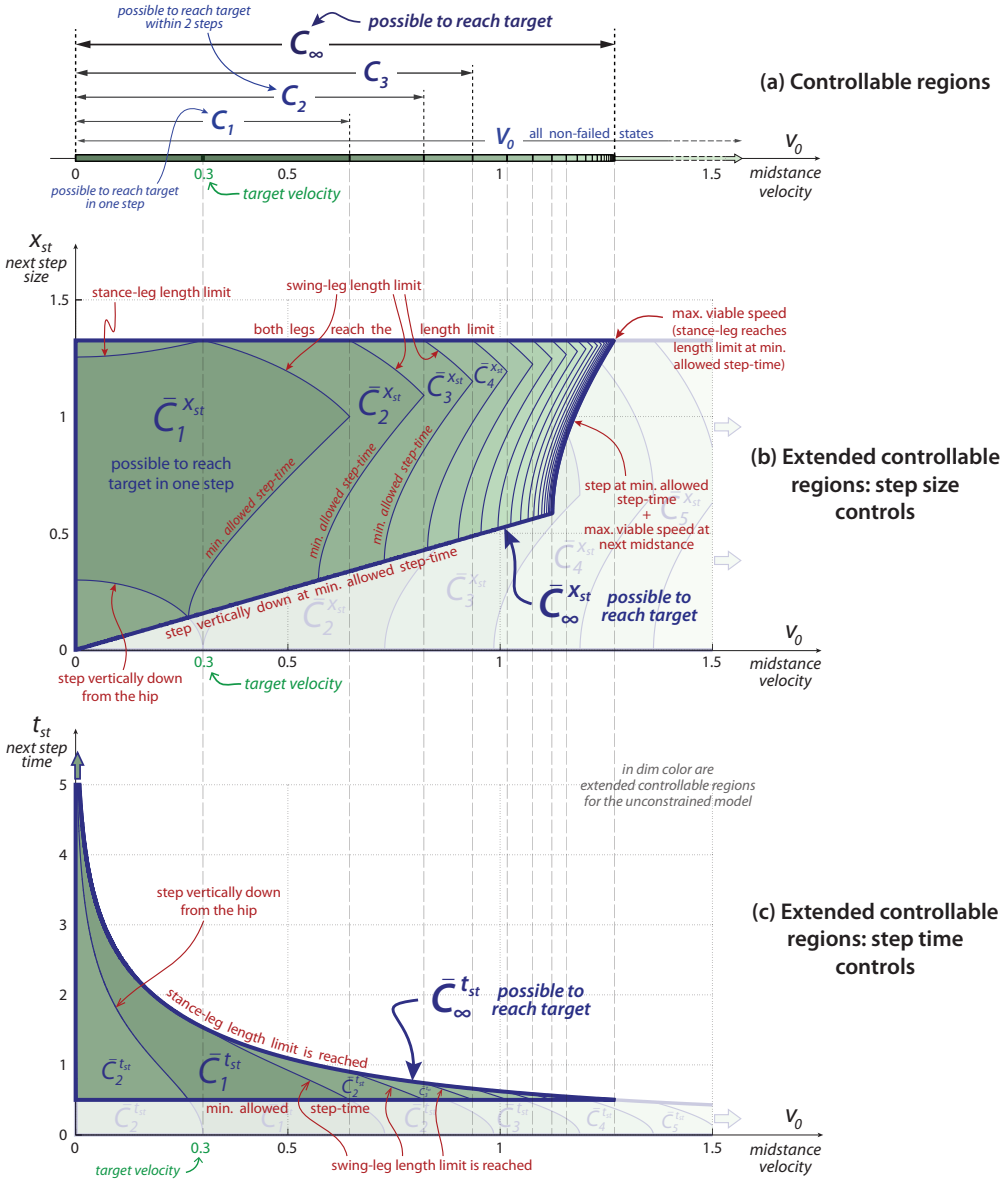


Figure 4.5: Controllability of the 2D LIP model with constrained swing time. This figure is analogous to Fig. 4.3 on page 109, but assumes a fixed lower bound on the allowed step-time: $t_{st} \geq t_{st,min}$. Constrained step-time t_{st} (time from midstance to heel-strike) is a proxy for leg-swinging actuator limitations of the robot. Here $t_{st,min} = 0.5$. The ∞-step controllable regions C_∞, $\bar{C}^{x_{st}}_\infty$, and $\bar{C}^{t_{st}}_\infty$ are the same (but for boundary points) as the ∞-step viable regions V_∞, $\bar{V}^{x_{st}}_\infty$, and $\bar{V}^{t_{st}}_\infty$ respectively (Fig. 4.6). The extended controllable regions for the unconstrained LIP model (Fig. 4.3) are shown here with light shading for comparison.

We numerically compute the extended n-step controllable regions $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ for the 2D LIP model constrained by (4.21). The computation methods are similar to those for the constrained 2D IP model (Section 3.2.5). The regions are displayed in Figs. 4.5b and 4.5c respectively. The smallest allowed step-time $t_{st,min}$ (dimensionless) used for numerical investigation is the same as for the 2D IP model in Section 3.2.5:

$$t_{st,min} = 0.5. \tag{4.22}$$

The extended controllable regions for the unconstrained LIP model (Fig. 4.3) are shown with light shading. Fig. 4.5a shows the n-step controllable regions C_n for the constrained LIP model: C_n are projections of both $\bar{C}_n^{x_{st}}$ and $\bar{C}_n^{t_{st}}$ onto the velocity axis.

The minimum step-time constraint (4.21) renders the ∞-step controllable regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{t_{st}}$ bounded. The upper bound of C_∞, the right-most point of all ∞-step regions, is

$$v_0 = \frac{x_{max}}{\sinh t_{st,min}} \approx 1.27. \tag{4.23}$$

This initial velocity is such that the stance leg reaches the length limit l_{max} at the time $t_{st,min}$ (hence, the above formula is equivalent to (4.17a) with $t_{st} = t_{st,min}$). For initial velocities greater than (4.23), the stance-leg length exceeds l_{max} before the robot is allowed to take a step, i.e. the robot fails. Hence, (4.23) is also the largest (one-step) viable velocity, as shown in Fig. 4.6. The points on the right boundary curve of $\bar{C}_\infty^{x_{st}}$ are such that the robot steps at the earliest allowed time $t_{st,min}$ and reaches the velocity (4.23) at the next midstance. For the points just to the right of this curve, it is possible to take one step, but impossible to avoid a failure during the second step. That is, all such points are in $\bar{V}_1^{x_{st}}$, but outside of $\bar{V}_2^{x_{st}}$, as shown in Fig. 4.6b. On the other hand, $\bar{V}_2^{x_{st}}$ is the same (but for some boundary points) as $\bar{C}_\infty^{x_{st}}$. Therefore, by the arguments similar to those in

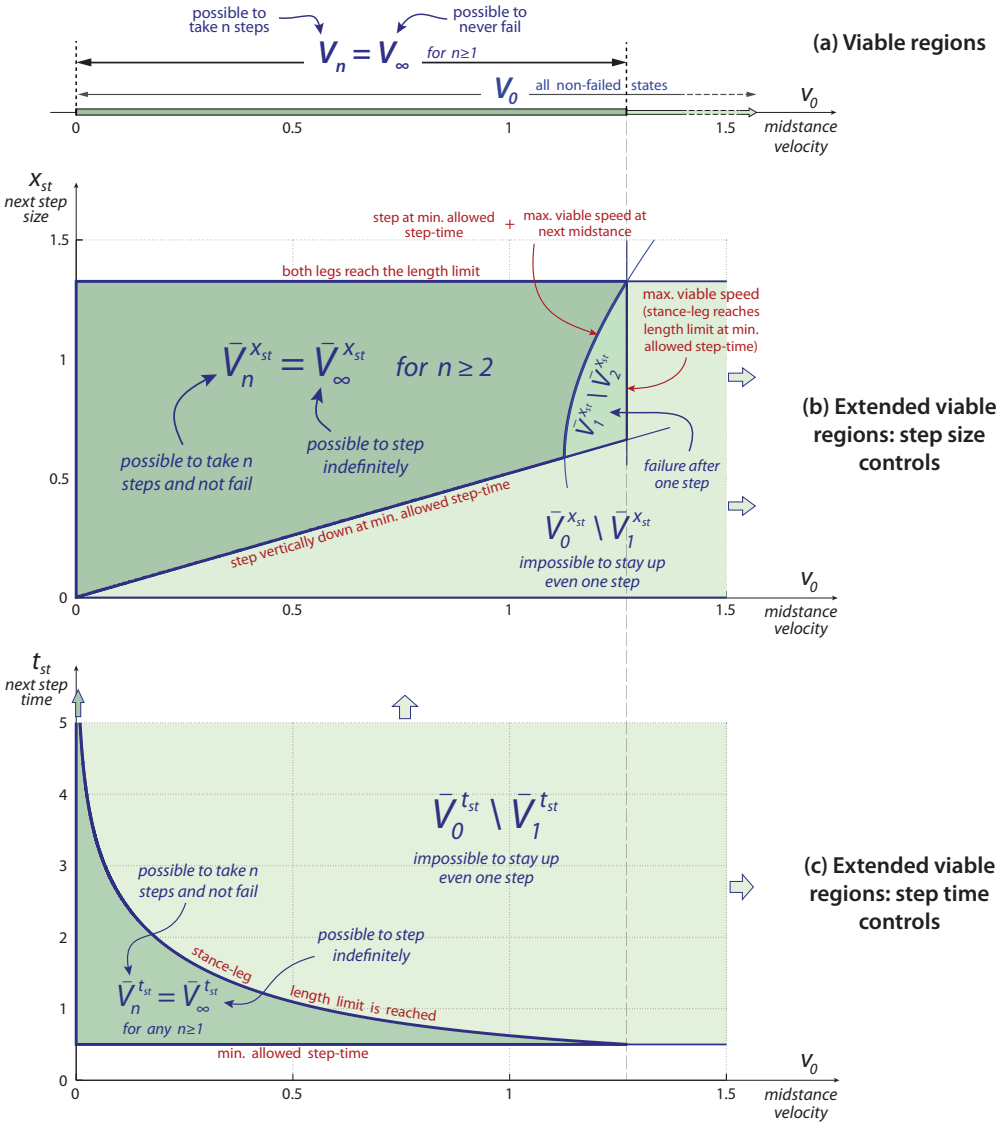


Figure 4.6: Viability of the 2D LIP model with constrained swing time. This figure is analogous to Fig. 4.4 on page 113, but assumes a fixed lower bound on the allowed step-time: $t_{st} \geq t_{st,min}$. Constrained step-time t_{st} (time from midstance to heel-strike) is a proxy for leg-swinging actuator limitations of the robot. Here $t_{st,min} = 0.5$. For the constrained LIP model $V_1 = V_\infty$: if it is possible to take one step, it is possible to step indefinitely. The ∞-step viable regions V_∞, $\bar{V}_\infty^{x_{st}}$, and $\bar{V}_\infty^{t_{st}}$ are the same (but for boundary points) as the ∞-step controllable regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{t_{st}}$ (Fig. 4.5) respectively.

Section 4.2.3, $\bar{V}_n^{x_{st}} = \bar{V}_\infty^{x_{st}} \approx \bar{C}_\infty^{x_{st}}$ for any $n \geq 2$. Similarly, we have $\bar{V}_n^{t_{st}} = \bar{V}_\infty^{t_{st}} \approx \bar{C}_\infty^{t_{st}}$ and $V_n = V_\infty \approx C_\infty$ for $n \geq 1$. Thus, we confirm, for the case of the 2D LIP model with constrained leg-swinging, the 'Viable is Controllable' claim in Section 2.3.1: in most cases, when it is possible to avoid a failure, it is also possible to reach any given target.[1]

The largest two-step controllable velocity is $v_0 \approx 0.82$ ($\sim 2.6\ m/s$ for a biped with one-meter-long legs), making the two-step controllable region C_2 about 65% of C_∞. Following discussion in Section 4.2.2, we say that more than two steps are required to return to the target only for 'large' velocities. For most (82%) velocities within the normal walking range (e.g. $v_0 \leq 1$, matching the walking range of the IP model, Section 3.2.1), a two-step recovery is possible, justifying the 'Two-step controllability' claim (Section 2.3.2) for the LIP model with constrained step-time.

4.3 IP vs. LIP

We considered the simple IP (Chapter 3) and LIP (Chapter 4) models of walking, both in 2D, and described their dynamics, viability, and controllability properties. Both models are often employed to study and control locomotion of walking robots [58, 40, 65, 43, 36, 44, 96, 82, 84]. In this section we compare the two models, based on their viability and controllability. We hypothesize that similar viability and controllability properties of the models — for example, if the respective extended controllable regions of the models are 'close' — may suggest that there is no strong advantage in using either model to control the CoM

[1] Our numerical calculations confirm that the (extended) ∞-step controllable regions in Fig. 4.5 remain the same for target velocities v_t other than (4.16).

motion of the robot. On the other hand, significant differences in the extended controllable regions may point out that one model is more suitable than the other for certain types of motion (e.g. fast walking) or robots.

Both the planar IP and LIP models have one-dimensional midstance and two control parameters per step. The midstance can be described by the same variable for both models: the velocity of the hip v_0. One of the controls, the step-size x_{st}, is also equivalent in the two models. However, the second control parameters do not match: the push-off p in the IP and the step-time t_{st} in the LIP model. Therefore, we only compare the step-size controllability of the models, that is we compare the extended regions $\bar{C}_n^{x_{st}}$. We also discuss the relationship between the push-off and step-time controls in Section 4.3.2 below. In particular, we find that the push-off is equivalent to the step-time in the small-step approximation of the models' dynamics.

4.3.1 Step-size controls

The extended n-step controllable regions $\bar{C}_n^{x_{st}}$ of the IP and LIP models with actuator limitations were described in Sections 3.2.5 and 4.2.4 correspondingly. To compare the regions $\bar{C}_n^{x_{st}}$ for the two models, we have to match the models' constraints (actuator limitations). We use the minimum-allowed-step-time constraint (4.21) (equivalent to (3.31b) for the IP), with $t_{st,min} = 0.5$ for both models. The maximum-leg-length constraint (4.7a) of the LIP model restricts the step-size x_{st} to be at most $x_{max} = 2\sqrt{l_{max}^2 - 1} \approx 1.33$. We impose on the IP model the maximum-step-size constraint $x_{st} \leq x_{max}$ with the largest allowed step x_{max} as above. We do not bound the maximum allowed push-off impulse p in the IP

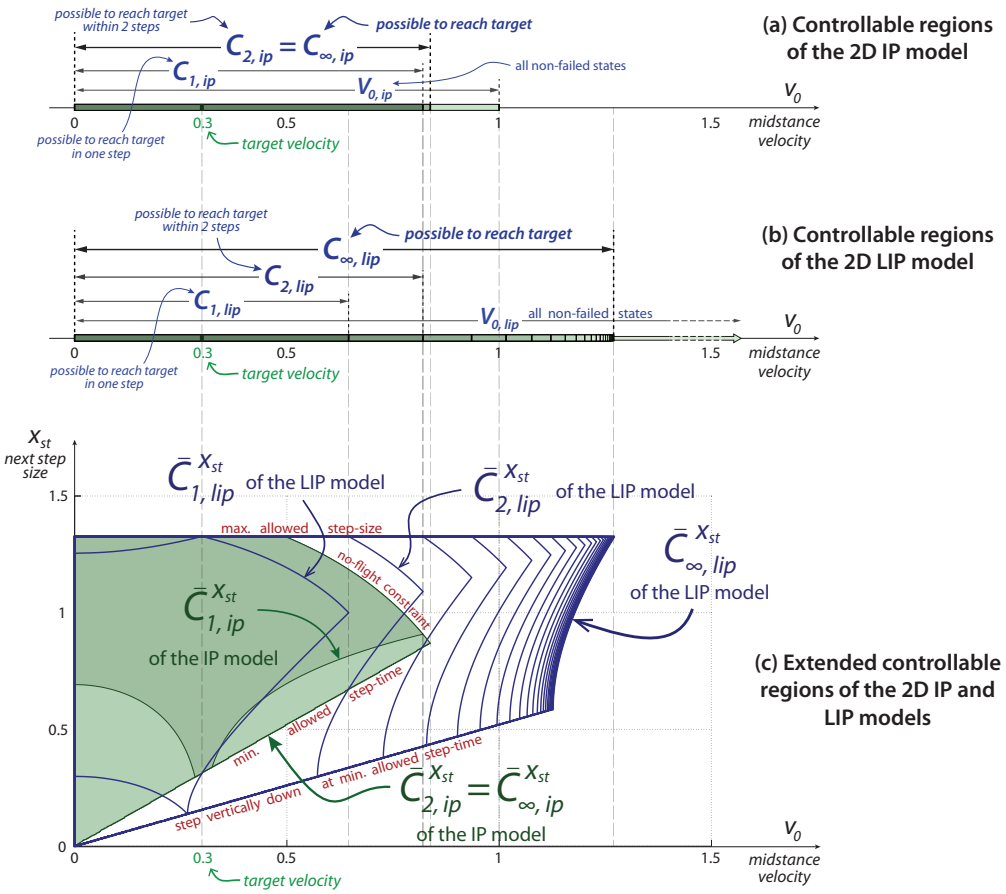


Figure 4.7: Comparison of the 2D IP and LIP model controllability. **(a), (b)** The n-step controllable regions $C_{n,ip}$ of the 2D IP and $C_{n,lip}$ of the 2D LIP model. For both models equivalent constraints are used on the minimum allowed swing-time and the maximum allowed step-size. **(c)** The extended n-step controllable regions $\bar{C}^{x_{st}}_{n,ip}$ of the IP (filled regions) and $\bar{C}^{x_{st}}_{n,lip}$ of the LIP model (shown by the boundaries) contrasted to each other. One-step controllability, $\bar{C}^{x_{st}}_{1,ip}$ vs. $\bar{C}^{x_{st}}_{1,lip}$: the LIP has more one-step step-size controls for smaller-than-the-nominal velocities, but the IP can recover from a larger range of velocities in one step. ∞-step controllability, $\bar{C}^{x_{st}}_{\infty,ip}$ vs. $\bar{C}^{x_{st}}_{\infty,lip}$: the LIP allows smaller steps and can recover (although in many steps) from $\sim 50\%$ larger velocities. The regions $C_{n,ip}$ and $\bar{C}^{x_{st}}_{n,ip}$ are similar to those in Fig. 3.6 (page 90); $C_{n,lip}$ and $\bar{C}^{x_{st}}_{n,lip}$ are equivalent to those in Fig. 4.5.

model, because there is no impulsive forces (hence, no matching constraint) in the LIP model.

The extended regions $\bar{C}^{x_{st}}_{n}$ of both the IP and LIP models with the discussed constraints, which we denote here as $\bar{C}^{x_{st}}_{n,ip}$ and $\bar{C}^{x_{st}}_{n,lip}$ respectively, are shown in

Fig. 4.7c. The target velocity is $v_t = 0.3$. The filled areas represent $\bar{C}^{x_{st}}_{1,ip}$ and $\bar{C}^{x_{st}}_{2,ip} = \bar{C}^{x_{st}}_{\infty,ip}$; the regions $\bar{C}^{x_{st}}_{n,lip}$ for $n \geq 1$ are shown by their boundaries and are copies of the regions $\bar{C}^{x_{st}}_{n}$ in Fig. 4.5b. The projections of the extended controllable regions onto the velocity axis are the controllable regions $C_{n,ip}$ and $C_{n,lip}$ shown in Figs. 4.7a and 4.7b respectively.

First, we look at the one-step controllability of the models, i.e. full recoveries from perturbations in one step. Comparing the regions $\bar{C}^{x_{st}}_{1,ip}$ and $\bar{C}^{x_{st}}_{1,lip}$ (Fig. 4.7c) we see two major differences. The LIP model has more recovery options (one-step dead-beat step-size controls) for velocities smaller than the nominal v_t. The no-flight restriction (the no-flight-after-push-off constraint (3.15a)) does not allow the IP model to speed up enough with small steps. On the other hand, $\bar{C}^{x_{st}}_{1,ip}$ extends farther to the right than $\bar{C}^{x_{st}}_{1,lip}$ — more larger-than-the-nominal velocities are one-step controllable for the IP than for the LIP model. The IP model is better at dissipating energy from the system, due to collision impacts with the ground.

Considering the ∞-step controllability, the LIP always outperforms the IP model: $\bar{C}^{x_{st}}_{\infty,ip}$ is a strict subset of $\bar{C}^{x_{st}}_{\infty,lip}$. As a result of the flight impossibility, the LIP model is able to recover from about 50% larger velocities, compared to the IP. However, for all velocities that are not ∞-step controllable by the IP model (velocities outside $C_{\infty,ip}$, $v_0 \gtrsim 0.84$): 1) the LIP requires several (more than two) steps to return to the target; 2) all such velocities are outside the normal walking range of bipeds. We also note, that the LIP model allows taking smaller steps, given a minimum allowed step-time. This is because the IP, in contrast to the LIP, has a fixed leg length: at the instant when the robot takes a step, the swing leg has to be placed on the ground at the horizontal distance $l \sin \theta_{sw}$ in front of

the hip (as defined by the fixed leg length l and the collision angle θ_{sw}). In the LIP model the swing foot is allowed to be placed directly below the hip.

Thus, the simple IP and LIP models of a biped have non-negligible differences in the step-size controllability and cannot be used as totally equivalent in a high-level controller design of a robot. One may choose one or the other model, based on the robot's specifics or the considered locomotion tasks of the robot. For example, the LIP model may be a better fit for fast walking and for walking with small steps. In Chapter 6 we use the IP model to design a high-level controller for the Cornell Ranger robot. The IP model better transmits the importance of collisions in Ranger's walk and the (almost) constant length of Ranger's legs.

The numerical experiments (not shown here) indicate that the comparative results presented in Fig. 4.7 remain qualitatively similar for target velocities other than $v_t = 0.3$.

4.3.2 Small steps: push-off in the IP is step-time in the LIP

The preemptive push-off in the simple IP model supplies energy to the system to compensate for collisional losses. In the LIP model the 'collisions' are smooth and there is no impulsive forces. Nevertheless, we draw the following analogy between the IP push-off impulse p and the LIP stepping time t_{st}.

In the LIP model, the robot speeds up from the initial midstance until the heel-strike and slows down from the heel-strike until the next midstance. The instant of the heel-strike, i.e. the step-time t_{st}, determines the ratio of the

speeding-up to slowing-down. Larger t_{st} means larger increase in the kinetic energy before the heel-strike and smaller kinetic-energy decrease after the heel-strike. Similarly, a larger push-off in the IP model means more energy input just before the collision and less energy dissipated due to the ground impact. This way, the step-time t_{st} and the push-off p play similar roles in the LIP and IP models, correspondingly.

The connection between t_{st} and p is more explicit in the small-step approximation of the models' dynamics. Consider the Poincaré map (3.14) of the 2D IP model (page 70). Using relation (3.3) between the step-size x_{st} and the collision angle θ_{sw} we write the Poincaré map as

$$\sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_1^2} = p\, x_{st} \cos\theta_{sw} + \cos 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2}.$$

Assuming small angles θ_{sw} (equivalently, small step-sizes), the above equation is approximated by

$$\dot{\theta}_1 \approx \dot{\theta}_0 + p\, x_{st}, \tag{4.24}$$

where $\dot{\theta}_0$ and $\dot{\theta}_1$ are velocities at the initial and next midstance respectively. For the 2D LIP model, we first solve the Poincaré map (4.11) of the model (page 100) with respect to the next-midstance velocity v_1:

$$v_1 = \sqrt{v_0^2 - x_{st}^2 + 2 v_0 x_{st} \sinh t_{st}}.$$

With the small-step assumption (small x_{st}), we write the approximate equation:

$$v_1 \approx v_0 + \sinh t_{st}\; x_{st}. \tag{4.25}$$

This equation can be further simplified, if we additionally assume small step-times t_{st} (thus excluding initial velocities v_0 close to zero):

$$v_1 \approx v_0 + t_{st}\, x_{st}. \tag{4.26}$$

Equations (4.24) for the IP model and (4.26) for the LIP model are the same, but for the swap of the push-off p with step-time t_{st}. Therefore, we say that *for small (and fast) steps, the push-off in the IP model is equivalent to the step-time in the LIP model.* Without the fast-step assumption, the push-off p is equivalent to $\sinh t_{st}$, as shown by equation (4.25).

In the dimensional form equations (4.24) and (4.26) become

$$l\,\tilde{\dot{\theta}}_1 \approx l\,\tilde{\dot{\theta}}_0 + \frac{1}{ml}\,\tilde{p}\,\tilde{x}_{st},$$
$$\tilde{v}_1 \approx \tilde{v}_0 + \frac{g}{y_G}\,\tilde{t}_{st}\,\tilde{x}_{st},$$

where $\tilde{}$ signifies a dimensional variable, l and m are the fixed leg length and the hip mass in the IP model, y_G the fixed height of the hip in the LIP model, and g the acceleration due to gravity. The equivalence between the IP push-off $\tilde{p}$ and the LIP step-time $\tilde{t}_{st}$ becomes scaled by the factor mgl/y_G:

$$\tilde{p} \sim \frac{mgl}{y_G}\,\tilde{t}_{st}. \tag{4.27}$$

CHAPTER 5

TWO STEPS IS ALMOST EVERYTHING

When a robot deviates from its preferred trajectory — due to an external disturbance, sensor noise, or perhaps to avoid an obstacle — it may have to take several steps to return back to the original trajectory, if a return is possible at all. The 'Two-step controllability' claim in Section 2.3.2 on page 26 suggests that in most cases, when full recovery is possible, it is possible to recover within two steps. That is, for most bipeds and most targets (nominal trajectories), two-step controllability is almost equivalent to ∞-step controllability. This claim is a slight generalization of Koolen et al.'s hypothesis [42] where, in contrast to an arbitrary target in our claim, the goal of coming to a full stop is assumed. Koolen hypothesizes that most locomotion of 3D bipedal robots, demonstrated to date, is such that at any instant the robot is able to come to a stop within two steps. Koolen also argues that in most cases humans are able to come to a stop by taking at most three steps.

In this chapter we support the 'Two-step controllability' claim in three ways. In Section 5.1 we justify the claim for two simple models of walking: the 2D Inverted Pendulum (IP, see Chapter 3) and 2D Linear Inverted Pendulum model (LIP, Chapter 4). In Section 5.2 we give examples of literature on control of legged robots, where the controls are either intentionally two-step horizon or show two-step recovery for considered disturbances. Section 5.3 overviews studies on human subjects (e.g. treadmill experiments) that support the 'Two-step controllability' claim. In Section 5.4 we discuss a counting argument which may help establish the possibility of two-step control strategies. We conclude the chapter in Section 5.5.

5.1 Simple models

Controllability of the simple planar IP and LIP models was studied in Chapters 3 and 4 respectively. For both models, the n-step controllable regions C_n show all perturbed velocities, such that a return of the robot to the nominal trajectory (the target) is possible within n steps. The corresponding controls u that allow an n-step recovery are shown by the *extended* n-step controllable regions $\bar{C}_n^u$. Here we compare the 2-step with ∞-step controllable and extended controllable regions for both the IP and LIP models.

Fig. 3.4 on page 82 displays the regions C_n, $\bar{C}_n^{x_{st}}$ (step-size controls), and $\bar{C}_n^p$ (push-off controls) of the 2D IP model with *unconstrained* actuation (arbitrarily large push-offs p and infinitely small step-times t_{st} are allowed), assuming the target velocity $\dot{\theta}_t = 0.3$. The two-step region C_2 is approximately $0 < \dot{\theta}_0 \lesssim 0.95$, while C_∞ includes all velocities $0 < \dot{\theta}_0 < 1$. That is, about 95% of ∞-step controllable states are also two-step controllable. A similar relation is valid for controls of the robot: the extended regions $\bar{C}_2^{x_{st}}$ and $\bar{C}_2^p$ cover about 95% and 97% in area of, respectively, $\bar{C}_\infty^{x_{st}}$ and $\bar{C}_\infty^p$. A large range of available one- or two-step step-size controls may be important, if allowed stepping locations are constrained.

Limited actuation of the IP model does not qualitatively change the relation between the two-step and ∞-step regions. Fig. 3.6 on page 90 shows controllability of the model assuming the maximum push-off and minimum swing-time constraints. All ∞-step controllable velocities are two-step controllable: $C_2 = C_\infty$. The extended regions $\bar{C}_2^{x_{st}}$ and $\bar{C}_2^p$ cover, respectively, 82% of $\bar{C}_\infty^{x_{st}}$ and 90% of $\bar{C}_\infty^p$.

The two-step controllability result remains valid for most target speeds of

the IP model. Exceptional cases include large velocities (say, $\dot{\theta}_t \gtrsim 0.8$) and over-restrictive actuation limits, e.g. 'small' maximum allowed push-off. In such cases the no-fly constraint and the push-off limitation do not allow the robot to gain speed fast — see examples in Appendix C.4. Notice, that velocities $\dot{\theta}_t > 0.8$ are beyond the normal walking range of bipeds — running is more energetically preferential for large velocities [77, 80, 53, 1].

Thus, we conclude that the 'Two-step controllability' claim holds true (for practical purposes) for the planar IP model: the robot can reach any given target in two steps in most cases, when it can reach it at all.

We can make almost the same conclusion about the planar LIP model. In the case of no actuation constraints (Fig. 4.3 on page 109), the ∞-step regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{t_{st}}$ are unbounded on one side, but the two-step regions C_2, $\bar{C}_2^{x_{st}}$, and $\bar{C}_2^{t_{st}}$ are finite. The robot can always recover from an arbitrarily large velocity, because flight is impossible for the LIP model and infinitely fast steps are allowed (in the absence of the minimum step-time constraint). However, if we consider only typical-real-walking-range velocities (say, $v_0 < 0.8$ as in the IP discussion above — the range of human walking speeds), the two-step controllability is close to the ∞-step controllability. Numerical calculations (not presented here) show that the 'Two-step controllability' claim is valid for any target speed less than, say, 0.8. Limiting the actuation of the LIP model (minimum step-time constraint) does not qualitatively change the relationship between the two-step and ∞-step controllability, as shown in Fig. 4.5 on page 115.

Koolen et al. [42] compute the n-step capture regions for the 3D LIP model and its two extensions: the LIP with finite-sized feet and LIP with both finite-sized feet and a flywheel. The capture regions are equivalent to slices, along a

given initial state, of the extended controllable regions for the zero target velocity (see Section 2.6.5). Koolen et al.'s considered initial state's velocity is about 0.76 (non-dimensional) — a rather large velocity for walking. For all three of their models, the two-step capture region is non-empty, i.e. the considered initial state is two-step capturable (two-step controllable for a standing-still target, in our language). For the simple LIP model, the two-step region is about 50% of the ∞-step capture region. For the two extensions (finite-size feet and flywheels) of the LIP model, the two-step regions cover, respectively, about 70% and 80% of the ∞-step regions.

5.2 Two-step controls in the robotics community

Two-step control strategies have been used to generate walking motions of robots [46, 83]. Lim et al. [46] use two-step receding horizon control with the capturability constraint, i.e. the robot has to come to a stop at the end of each planned two-step trajectory. In the animation world, van de Panne [83] generates the optimal (supposedly most physically realistic) motion of a character for a sequence of desired footstep locations provided a priori. Van de Panne finds two-step horizon control is found to be sufficient to produce a realistic animation.

Other research groups do not intentionally consider the two-step horizon, but instead discover that planning two or three steps ahead is sufficient for their models [14, 57, 15]. Nishiwaki et al. [57] study the problem of online generation of walking trajectories (of the robot's links) based on a given ZMP path. Each trajectory is generated as a solution of a Boundary-Value Problem (BVP) and

plans two, three, or more steps ahead, trying to satisfy several constraints (such as initial and final conditions, and following of the given ZMP path). Nishiwaki finds that planning more than three steps ahead negligibly improves the generated trajectories (their smoothness and errors in meeting the constraints).

Two-step control strategies were considered in the analysis of the Spring-Loaded Inverted Pendulum (SLIP) model, a simple model of hopping and running [68, 12, 28]. The SLIP model has a point-mass body and a massless spring (the leg) attached to it. The controls of the model for each step are the position of the spring-like leg at touch-down (transition from flight to stance) and two spring coefficients for the stance phase (one for the leg compression, the other for the leg decompression). Carver et al. [15] study controllability of the 3D SLIP model: for a fixed target state (horizontal position, height, and velocity of the point-mass at the flight apex) they find all position perturbations that can be corrected in at most two steps. Carver finds that all two-step controllable perturbations form a large (on the scale of the reference step-size) area surrounding the target location. For most perturbations, within the reference step-size of the target, two steps are enough to return to the target. A less restrictive goal for the SLIP model is *compass correction*: a target apex height, velocity magnitude, and velocity direction of the model, but no specific desired location in the plane. Compass-correction goal, Carver claims, requires only one step to correct for perturbations.

5.3 Evidence from humans

We consider two types of studies on human subjects that provide evidence in support of our 'Two-step controllability' claim. The first type investigates recovery strategies of humans after various disturbances during walking or standing. Studies of the second kind aim to understand how humans plan ahead during locomotion, e.g. where they look preparing for the next step.

Some disturbances commonly considered in treadmill experiments include side pushes [35], slipping [78, 49, 55], and tripping [74, 62, 25]. Hof et al. [35] study balance responses of young adults to sideway pushes (impulse $= 3$ to $12\ kg\,m\,s^{-1}$ in magnitude) at different phases of walking. The participants were always able to return to their preferred gait[1] in at most two steps after a push. When it was possible (based on the reaction time of the person, the instant of the push, etc.) to move the swing leg to a desired stepping location, balance was regained after the first step. Likewise, Tang et al. [78] report that humans can restore balance within two steps after slipping. The slipping was simulated in the experiment by jerking a plate under the colliding leg at the instant of collision; the plate perturbation was $10\ cm$ in either forward or backward direction at the speed $40\ cm/s$. Schillings et al. [74] research recovery strategies of humans after stumbling over an obstacle. Healthy subjects were walking at a comfortable speed, when a $4.5\ cm$ tall obstacle was released on the treadmill at different phases of the step cycle. The authors report that the normal gait was usually restored in two steps.

Other research that we use in support of our 'Two-step controllability' claim,

[1] That is, they were able to return to a reference speed in a fixed direction, while always staying within the width of the treadmill.

studies motion planning in humans. In a series of experiments by Patla et al. [60, 61] and Hollands et al. [37] human participants were asked to follow a specified path on the floor, while their gaze behavior was monitored. One of the objectives was to understand where and when humans focus their eyes when planning the motion. The results showed that in some tasks (change of the step length or width, obstacle avoidance [60], raised stepping stones [37]) the participants were looking at the location they were about to step next (i.e. looking one step ahead) most of the time. In other tasks (change of the walking direction [60], stepping stones on flat ground [61]) the subjects on average were looking two steps ahead, when they were focusing their eyes on specific areas on the ground.

An interesting two-step result comes from Matthis and Fajen [50], who study performance of humans in the obstacle avoidance task with limited visual information. Human subjects were asked to walk across a room filled with obstacles displayed on the floor by a projector. The obstacles were visible only within one, two, or more step-lengths around the participant. The participants' performance (frequency of collisions) significantly improved when the visibility radius was increase from one to two step lengths. On the other hand, visibility of more than two step lengths ahead weakly affected the performance.

5.4 Counting rules

It may be possible to determine the possibility of, say, one-step, two-step, or n-step controllability by comparing the number of control parameters with the number of target restrictions. First we define

m_{cont} = the number of controls available at each step. For example, for both the planar IP and LIP models we have $m_{\text{cont}} = 2$: push-off and step-size in the IP model, and step-size and step timing in the LIP model;

m_{goal} = the number of restrictions on the overall state space given by the goal. For both the planar IP and LIP models the goal, considered in this thesis, is a given target velocity at midstance — hence $m_{\text{goal}} = 1$.

By counting equations and unknowns, it would be non-degenerate for there to be a non-trivial (non-zero measure) n-step controllable region, if

$$n * m_{\text{cont}} \geq m_{\text{goal}}, \tag{5.1}$$

That is, there are at least as many control parameters as there are target values; or, the dimension of the control space is at least as big as the dimension of the target space. We must be aware, that in the same way that not all square matrices are non-singular, the counting argument gives a necessary, and maybe even likely, but not a sufficient condition.

Examples of the counting argument for simple models include:

- Consider a fixed target velocity of the 2D IP (or LIP) model. This problem has two controls per step and one target restriction: $m_{\text{cont}} = 2 \geq m_{\text{goal}} = 1$. Hence, the counting argument suggests there should be a region in the state space (the one-step controllable region), from where the target can be reached in one step. Further, because there is one more control ($m_{\text{cont}} = 2$) than there are goal constraints ($m_{\text{goal}} = 1$), one may expect a one-parameter family of one-step deadbeat controllers. These conclusions are confirmed by our calculations in Chapter 3 for the 2D IP and in Chapter 4 for the 2D LIP model.

- Carver et al. [15] use similar counting arguments to guess the controllability of the 3D SLIP model, described in Section 5.2. The model has four controls per step, $m_{\text{cont}} = 4$: next footstep location (2 numbers) and two spring coefficients for the stance phase. One of the goals Carver considers is reaching a specific full state of the model at flight apex: location of the point-mass in space (3 numbers) and horizontal velocity (2 numbers). That is, $m_{\text{goal}} = 5$ and we have $m_{\text{cont}} < m_{\text{goal}}$, but $2m_{\text{cont}} > m_{\text{goal}}$. Therefore, generally two steps are required to correct for perturbations from the target state [15].
- A less restrictive goal for the SLIP model, considered by Carver et al. [15], is *compass correction*, which, in contrast to a specific target state, disregards the location of the model in the horizontal plane. The compass correction target is defined by three values, $m_{\text{goal}} = 3$: the target apex height, velocity magnitude, and velocity direction of the model. Therefore, $m_{\text{cont}} > m_{\text{goal}}$ and, by the counting argument, perturbations are predicted to be correctable in one step [15].

We also consider examples of goals and counting arguments for these goals for the 3D Inverted Pendulum model — see Appendix E.

Based on our considerations of the IP and LIP models and Carver et al.'s analysis of the SLIP model [15], we argue that the counting rule (5.1), generally, works for simple models and practical goals.

5.5 Conclusion

We considered the following question of walking balance: how many steps does a biped generally need to recover from a perturbation? We approached this question by studying controllability of two simple models of walking: the planar IP and LIP models. Our numerical investigations of both models showed that, when the model is able to recover from a perturbation at all, in most cases it can do so in two steps or fewer. This result remains valid if the actuation of the models is limited within some practical bounds. We also provided some evidence, from both the robotic community and experiments with human subjects, showing the nearly maximal two-step recovery capabilities in more complicated robot models and humans.

The above considerations lead us to the following proposition: *two steps is almost everything*. That is, anything a biped is capable of doing at all, for most initial conditions it can do so within only two steps. Practically, this suggests that in a controller design for biped robots, there is no need, at least for balance purposes, to plan the motion of the robot more than two steps ahead.

CHAPTER 6

STABLE, ROBUST, EFFICIENT, AND SIMPLE WALKING CONTROLLER

In this chapter we describe a new controller design approach for bipedal robots. Some properties of controllers, which are considered important in legged locomotion, include:

- stability with respect to a desired motion (or range of motions);
- robustness to external disturbances, noise (in sensing, actuation, etc.), and model errors;
- energy efficiency;
- low computer power requirements.

The robotics community has produced walking robots, which excel in some of the above properties, but are less competent in others. For example, Boston Dynamics' robots [67, 88] seem to be highly robust (not falling after strong pushes, walking on rough terrain, ice, steep slopes, etc.), but they use two orders of magnitude more power then humans [9, p.9]. On the other hand, robots from the Cornell Locomotion Lab [9, 11, 21] consume about as much energy as humans (scaled), but can only walk on flat ground and in the absence of significant external disturbances.

We propose a walking controller which tries to compromise between robustness, efficiency, and stability. The proposed controller seems to have all of the mentioned properties while maintaining a simple structure and without requiring significant computation. That is, we use

- an informal approximate minimization of complexity and computation.

The design approach employs the concepts of (extended) viable and controllable regions, described in detail in Chapter 2. We use the four-legged 'bipedal' robot Cornell Ranger as an example case to explain the controller design method. However, all design steps are analogous for any bipedal walking robot. In Section 6.1, we model Ranger as the 2D IP model; constraints are employed to represent limited actuation in Ranger. We then find extended controllable regions for our proxy model of Ranger. Section 6.2 discusses desired properties of the controller and their relation to the controllable regions. A high-level walking controller is developed: given Ranger's velocity at midstance, it generates the next desired stepping location and amount of push-off to be used at the next collision. The controller is tested in simulation in Section 6.3, both in perfect conditions (no noises, model errors, etc.) and in the presence of random perturbations. The largest perturbations, such that the robot does not fail, are estimated. Section 6.4 summarizes the controller design procedure. The chapter is concluded in Section 6.5 with a discussion of the main results and future improvements of the controller design.

6.1 Test robot: Cornell Ranger

The testbed for the control ideas here is the Cornell Ranger robot. Ranger is perhaps the most energy efficient walking robot so far, having walked 65.2 kilometers on a single battery charge on May 1-2, 2011 [9, 11].

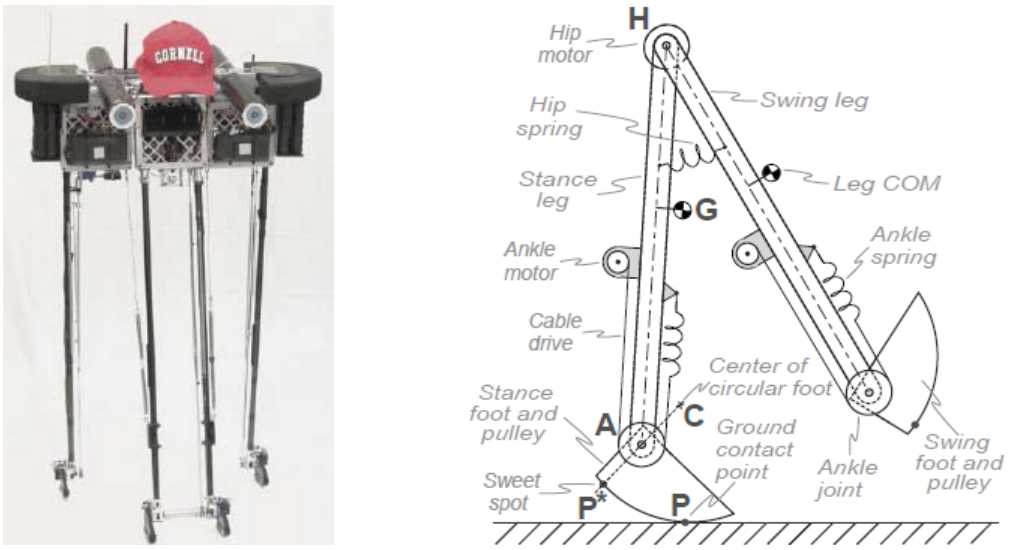
6.1.1 Model

In 2D conception[1] Cornell Ranger (Fig. 6.1) has two legs connected by a revolute hip joint. Each leg has a circular foot attached at the ankle joint. Most of the mass is concentrated close to the hip, the legs and feet are designed to be light. Ranger has three essential motors: one for each ankle and one for the hip joint. The ankle motors each have one internal degree of freedom. Therefore, the planar model of Ranger has the total of six degrees of freedom (DOF): three joint angles (two ankles and one hip), two DOF in the ankle motors, and the global orientation of the robot (e.g. the absolute angle of the stance foot). However, during the double stance motion, when both feet touch the ground, the absolute orientation of the robot can be found from geometry, making it the total of five DOF. Collision of the swing leg with the ground is assumed to be instantaneous with no change in configuration of the robot and no impulsive torques at the joints. No slipping is considered between the feet and the ground. Full details of the high-fidelity model of Ranger can be found in [9].

6.1.2 IP model proxy of Ranger

We use the planar Inverted Pendulum (IP) model, studied in detail in Chapter 3, as a proxy for Ranger. The IP model is simple, low-dimensional and, hence, easy to work with. At the same time, following discussion in Section 2.5.4 on page 39, we believe that the dynamics of the center of mass of most bipedal robots, even those with many degrees of freedom, are close to that of a point-mass model (such as the IP or LIP model) during the single stance motion.

[1] Ranger has two pairs of rigidly connected legs and can be considered a 2D biped with intrinsic sideways stability.



(a) Photo of Ranger **(b)** High-fidelity model of Ranger

Figure 6.1: Four-legged bipedal robot Cornell Ranger. a) The frontal view photo of the Cornell Ranger robot. The two inner legs, as well as the two outer legs, are interlocked, so that the robot can be considered a 2D biped with intrinsic side stability. b) Schematics of the Cornell Ranger high-fidelity model. The model has two rigid inertial legs and two massless circular feet. There are three actuators: an electric motor is attached to each of the two ankle joints through a linear spring, another electric motor creates torque at the hip joint. A torsional spring is attached between the legs at the hip. No slipping between the feet and the ground is considered. (Images from [9].)

There are two controls per step in our planar IP model: the next stepping location (the size x_{st} of the next step) and the amount of push-off p just before the next collision. We discuss how these idealized controls of the simple model are related to the actuation in the full model of Ranger, and how Ranger's actuator limitations put constraints on x_{st} and p.

The push-off impulse p represents all effort by the ankle motors in Ranger during and shortly before the double stance. Notice, that the double stance phase of the full model is replaced by an instantaneous collision in the IP model. For the nominal gait of the full Ranger model (the gait used during the 65.2 km walk), the double stance spans 5% of the step cycle (only 3% for the physical robot), and most of the work done by the ankle motors is done shortly before

and during the double stance (also true for the real robot) [9]. Because the ankle motors on Ranger have limits on the torque they can generate, we set the upper bound p_{max} on the allowed push-off p in the IP model (see equation (6.1a) below). We use a computer simulation of the full model of Ranger to make a rough estimate $p_{max} = 0.3$. Details of the simulation are given in Appendix D.2 on page 192.

The second actuation in the IP model, the step size x_{st}, replaces the hip torque in the full model, which drives the swing leg to a desired stepping location. In the IP model, the dynamics of the swing leg are not considered, assuming that the swing leg can be instantaneously placed into any desired position without affecting the dynamics of the stance leg. At the same time, in the real robot, as well as its detailed model, the capabilities of the hip motor are limited, resulting in bounds on how fast the swing leg can be moved around. We model this limitation by constraining from below the *step time* (also, *swing time*) t_{st}, defined for the IP model as the time from the midstance to the next collision, see equation (6.1c) below. We use both a computer simulation of the full model and physical experiments with Ranger to estimate the minimum step time $t_{st,min} = 0.4$ (~ 0.13 sec) — see Appendix D.1 on page 190 for details.

The minimum swing time implicitly constraints the step size x_{st} from below. We also set an upper bound $x_{st,max}$ on the control x_{st}, representing the maximum step size Ranger is physically capable of taking. The value of $x_{st,max}$ is estimated to be 0.9, constrained by the mechanical design of the robot.

To summarize, we use as a proxy for Ranger the 2D Inverted Pendulum

model augmented with the following constraints:

$$0 \leq p \leq p_{max} = 0.3, \tag{6.1a}$$

$$0 \leq x_{st} \leq x_{st,max} = 0.9, \tag{6.1b}$$

$$0.4 = t_{st,min} \leq t_{st}. \tag{6.1c}$$

6.1.3 Controllable regions

We proceed to find the (extended) controllable regions for the IP model of Ranger. The regions are used in Section 6.2 to design a high-level walking controller for Ranger.

We use the 'average step' of Ranger during its ultra-marathon walk [9, 11] as the (desired) nominal trajectory for our IP model. Thus, the nominal velocity $\dot{\theta}^*$ is Ranger's average speed during the walk:

$$\dot{\theta}^* = 0.19, \tag{6.2a}$$

The average step length of Ranger was approximately

$$x_{st}^* = 0.35. \tag{6.2b}$$

We set the nominal push-off amount such, that it produces a (one-step) periodic motion of our IP model, given the midstance velocity $\dot{\theta}^*$ and step size x_{st}^*. The nominal push-off p^* is found from the Poincaré map (3.14) and relation (3.3) by letting $\dot{\theta}_0 = \dot{\theta}_1 = \dot{\theta}^*$ and $x_{st} = x_{st}^*$:

$$p^* \approx 0.046. \tag{6.2c}$$

We will refer to the IP gait characterized by parameters (6.2) as the *nominal* (or *reference*) *trajectory of Ranger*.

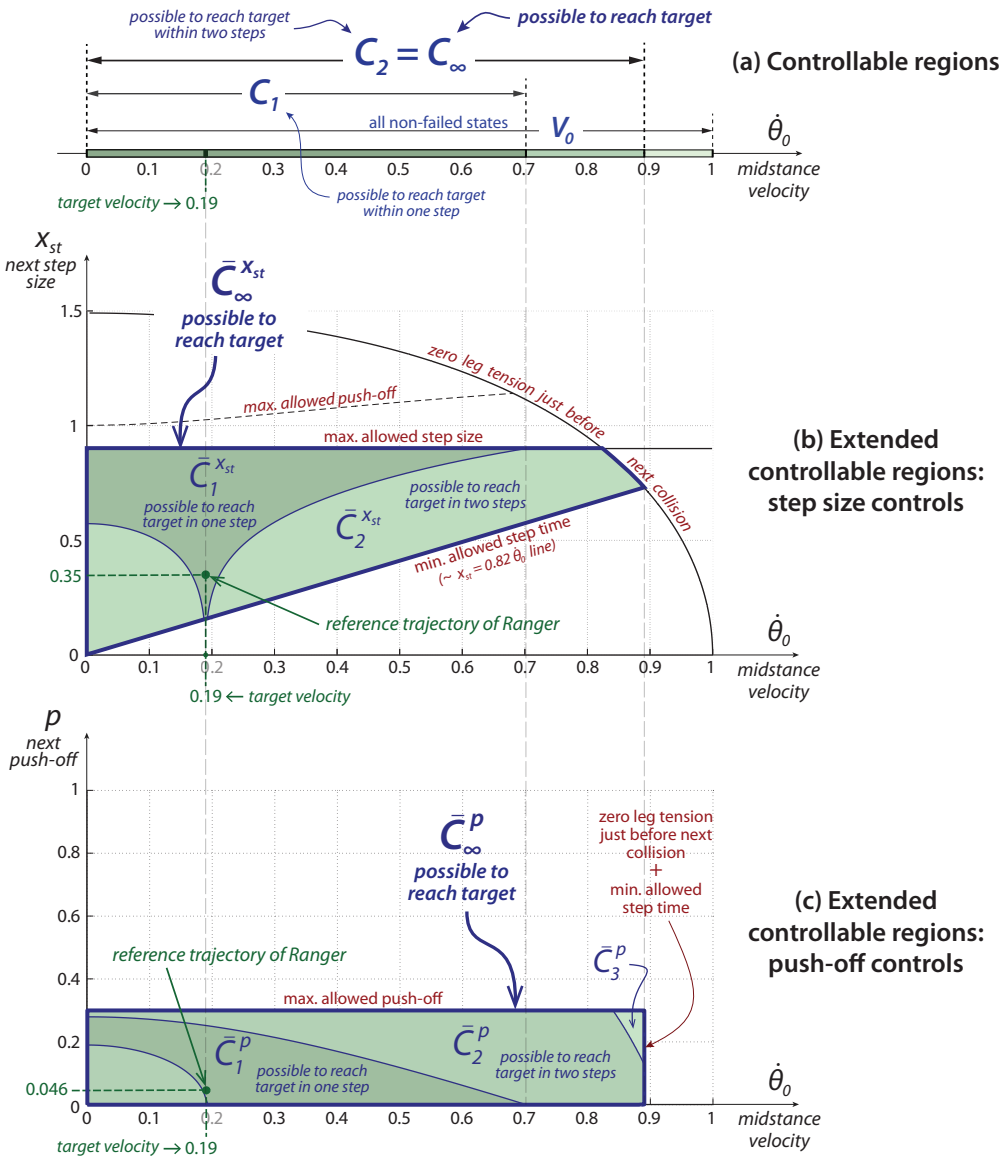


Figure 6.2: Controllability of the IP model of Ranger. Similar to Fig. 3.6, (a) controllable regions C_n and (b), (c) extended controllable regions $\bar C_n^{x_{st}}, \bar C_n^{p}$ for a constrained 2D IP model are shown. Here the constraints are proxies for limited actuation in the Ranger robot: lower bound on the step-time (midstance-to-heelstrike time), $t_{st} \geq t_{st,min} = 0.4$ (hip motor limitation); maximum allowed push-off, $p \leq p_{max} = 0.3$ (ankle motor limitation); and maximum step-size, $x_{st} \leq x_{st,max} = 0.9$ (mechanical restriction). The target speed $\dot\theta_t = 0.19$ corresponds to Ranger's energy-optimal gait. For this IP model of Ranger $C_2 = C_\infty$, $\bar C_2^{x_{st}} = \bar C_\infty^{x_{st}}$, and $\bar C_3^{p} = \bar C_\infty^{p}$.

We numerically compute the controllable and extended controllable regions C_n, $\bar{C}_n^{x_{st}}$, and $\bar{C}_n^p$ corresponding to the target velocity (6.2a). The computation is analogous to the computation of the corresponding regions for the constrained 2D IP model in Section 3.2.5 on page 88. The regions C_n, $\bar{C}_n^{x_{st}}$, and $\bar{C}_n^p$ for our IP model of Ranger are shown on Fig. 6.2.

As pointed out on the figure, two-step controllability is equivalent to ∞-step controllability:

$$C_2 = C_\infty. \tag{6.3}$$

That is, for our simple model of Ranger, if it is possible to return to the nominal trajectory at all (e.g. after a disturbance), it is always possible to do so within two steps. This result supports the 'Two-step controllability' claim made in Section 2.3.2. Note, that there are states (such as $\dot{\theta}_0 = 0.8$) for which a two-step policy is required to return to the nominal trajectory, i.e. a return in one step is impossible: $C_1 \neq C_2$. Relation (6.3) holds true for the step-size controls as well: $\bar{C}_2^{x_{st}} = \bar{C}_\infty^{x_{st}}$. All possible step-size controllers, which allow to reach the nominal speed, are either one- or two-step policies. Two-step controllability, however, is only approximate for the push-off controllers: $\bar{C}_2^p \approx \bar{C}_\infty^p$, while $\bar{C}_3^p = \bar{C}_\infty^p$.

The size of the region $\bar{C}_\infty^{x_{st}}$ is mostly restricted by the actuator limitations: the top boundary of $\bar{C}_\infty^{x_{st}}$ corresponds to the maximum step-size constraint (6.1b), while the bottom boundary to the minimum step-time (6.1c). On the other hand, the push-off limitation (6.1a) has no effect on $\bar{C}_\infty^{x_{st}}$ and, hence, on the controllability of the model.[2] Restrictions of the robot in taking a step appear to prevail over those in pushing off during a recovery of the robot.

[2] The dashed line on Fig. 6.2b shows the part of the $\bar{C}_\infty^{x_{st}}$ boundary corresponding to the maximum push-off, if the step-size was not restricted from above.

Notice, that all boundaries of $\bar{C}_{\infty}^{x_{st}}$ are target independent and are only related to the initial velocity and step-size. Therefore, for any combination $(\dot{\theta}_0, x_{st})$ outside of $\bar{C}_{\infty}^{x_{st}}$, the robot fails. On the other hand, for any point in $\bar{C}_{\infty}^{x_{st}}$, the robot can reach the nominal trajectory and, thus, step indefinitely. We conclude that the extended viability kernel $\bar{V}_{\infty}^{x_{st}}$ is the same as the ∞-step controllable region: $\bar{C}_{\infty}^{x_{st}} = \bar{V}_{\infty}^{x_{st}}$, supporting the 'Viable is Controllable' claim in Section 2.3.1 on page 25. Similarly, $\bar{C}_{\infty}^{p} = \bar{V}_{\infty}^{p}$. Because the viable and controllable regions V_n and C_n are projections of their extended counterparts onto the velocity axis, we also state: $C_{\infty} = V_{\infty}$.

6.2 Controller design for Ranger

We develop a high-level controller for Cornell Ranger, using the simple IP model of Ranger and its controllable regions described in Section 6.1 above. For a given midstance velocity $\dot{\theta}_0$ of the robot, the controller tells where to step (i.e. the next step size x_{st}) and how hard to push off (the next push-off p). Thus, we split the high-level controller into the step-size controller u_x:

$$x_{st} = u_x(\dot{\theta}_0), \tag{6.4a}$$

and the push-off controller u_p:

$$p = u_p(\dot{\theta}_0). \tag{6.4b}$$

For simplicity of the approach, we consider the two above controllers separately. First, we construct the step-size controller u_x in Sections 6.2.1 to 6.2.5, and then complement it with the push-off controller u_p in Section 6.2.6. This procedure can be done in the reverse order as well, starting with u_p and then augmenting it

with u_x. The resulting walking controller may depend on the design order quantitatively, but in either case it will have all desired qualities listed in Chapter 6 on page 135.

6.2.1 Step-size controller: objectives

The function $u_x(\cdot)$ in (6.4a) defines a curve in the $(\dot{\theta}_0, x_{st})$-plane, assigning a unique step size x_{st} to each velocity $\dot{\theta}_0$. Designing a step-size controller u_x is drawing a curve in the $(\dot{\theta}_0, x_{st})$-plane. We are going to state the criteria, which we use to draw the step-size controller curve u_x.

First, the controller has to avoid a failure of the robot (i.e. avoid falling down and violating constraints) at any cost.[3] To achieve this, we want the robot's velocity $\dot{\theta}_0$ to always be in the viability kernel V_∞ and the corresponding step-size x_{st} in the extended viability kernel $\bar{V}_\infty^{x_{st}}$, i.e. $(\dot{\theta}_0, x_{st}) \in \bar{V}_\infty^{x_{st}}$. However, recall from Section 6.1.3, that for our IP model of Ranger, $V_\infty = C_\infty$ and $\bar{V}_\infty^{x_{st}} = \bar{C}_\infty^{x_{st}}$. Thus, we require the step-size controller curve u_x to be inside the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$, which is shown on Fig. 6.2b. This condition guarantees not only the possibility of not failing, but also the possibility to reach the nominal velocity $\dot{\theta}^*$.

Next, we want the controller to produce a stable walking around the nominal trajectory. If the robot is exactly on the nominal trajectory, the controller should take no correcting actions, i.e. it has to use the nominal step-size x_{st}^* and nominal push-off p^*. Therefore, the curve u_x has to pass through the point $(\dot{\theta}^*, x_{st}^*)$.

[3] Koolen, et al [42] argue that in certain (possibly extreme) cases it might be preferential to let the robot fall down (safely) and then get up again, than trying to balance the robot at any cost. While accepting this argument, we do not consider such rather extreme strategies in this thesis.

Among all step-size controller curves (6.4a) which

- are inside the region $\bar{C}_{\infty}^{x_{st}}$,
- pass through the nominal trajectory point $(\dot{\theta}^*, x_{st}^*)$,

we want to find one which is (approximately) optimal with respect to several different objectives. For a bipedal robot, such objectives include, as listed on page 135, stability, robustness, and energy efficiency. At the same time, we want the controller to have a simple structure. We proceed to discuss and quantify each of the optimization objectives for our simple model of Ranger.

6.2.2 Stability

We call a controller stable if it returns the system back to the nominal trajectory after a perturbation. For our model, to reach the nominal trajectory is to reach the nominal velocity $\dot{\theta}^*$ at midstance.

We look at the error $\Delta\dot{\theta} = |\dot{\theta} - \dot{\theta}^*|$ between the perturbed velocity $\dot{\theta}$ and the nominal $\dot{\theta}^*$, and how this error changes after each step. We define the (next-step) *relative error* in velocity ν as the ratio of the error $\Delta\dot{\theta}_1$ after one step to the initial error $\Delta\dot{\theta}_0$:

$$\begin{aligned} \nu(\dot{\theta}_0, x_{st}, p) &= \frac{|\dot{\theta}_1(\dot{\theta}_0, x_{st}, p) - \dot{\theta}^*|}{|\dot{\theta}_0 - \dot{\theta}^*|} \\ &= \begin{array}{c} \textit{fractional reduction of disturbance} \\ \textit{after one step} \end{array} . \end{aligned} \tag{6.5}$$

The relative error ν is a function of the initial velocity $\dot{\theta}_0$ and controls x_{st}, p used during the step; the velocity $\dot{\theta}_1$ at the next midstance is found from the Poincaré

map (3.14). For the controller to be stable, the controls x_{st} and p have to be such, that the absolute error $\Delta\dot{\theta}$ decreases, i.e. the relative error $\nu < 1$. For example, $\nu = 0.7$ implies that the disturbance has decreased by 30% in one step.

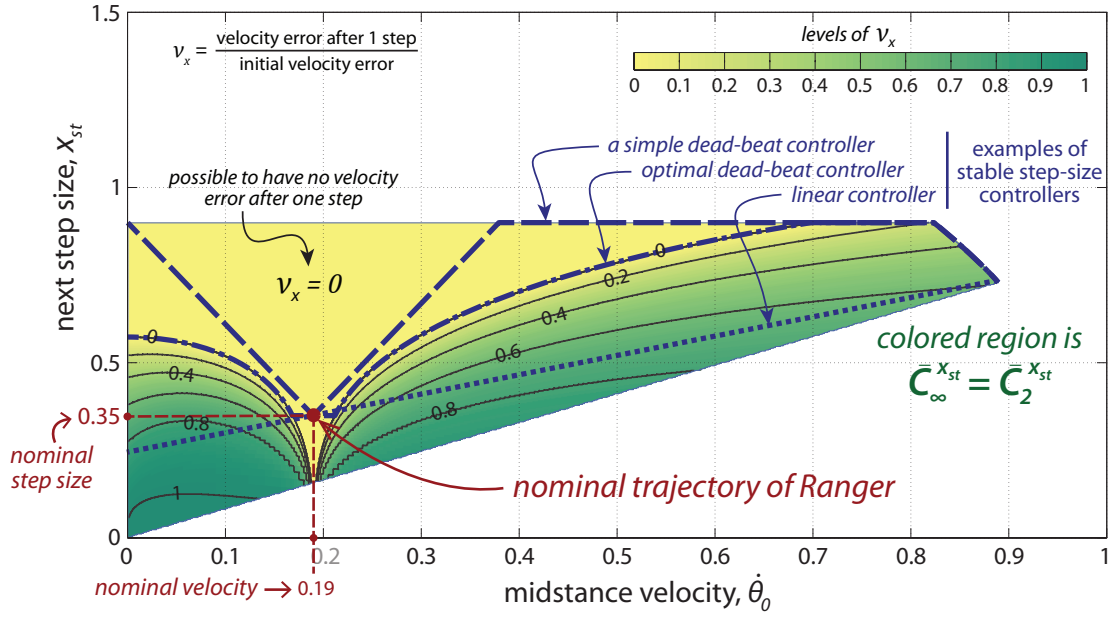
For the step-size controller design, we would like to know the effect of the step-size x_{st} on the relative error in velocity. We define the *best relative error* ν_x for a given x_{st} and initial velocity $\dot{\theta}_0$ as the smallest relative error ν over all viable push-offs:

$$\nu_x(\dot{\theta}_0, x_{st}) \;=\; \min_p \; \nu(\dot{\theta}_0, x_{st}, p). \tag{6.6}$$

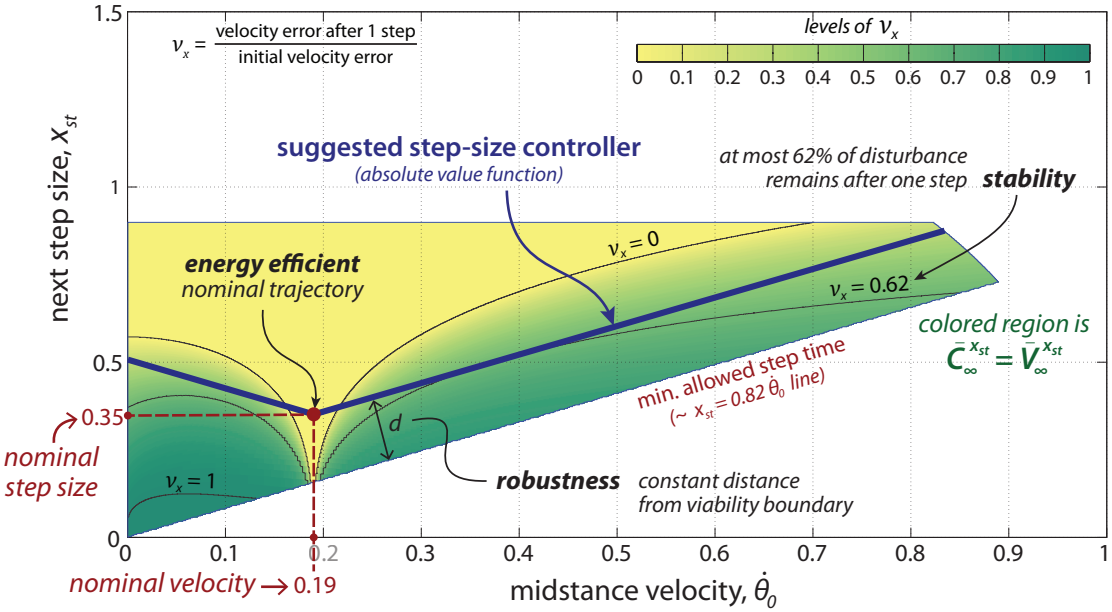
That is, ν_x is the best relative error that can be achieved (with an appropriate push-off) for a specific step-size controller. We want ν_x to be possibly smaller, thus indicating faster convergence to the nominal trajectory. If $\nu_x > 1$, the velocity error increases, i.e. the controller is not stable.

Fig. 6.3a shows the best error ν_x for all points $(\dot{\theta}_0, x_{st})$ inside the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$. The color scheme and contour lines on the figure indicate the level lines of ν_x, which varies from 0 to 1 inside $\bar{C}_\infty^{x_{st}}$. Notice, that ν_x is never greater than 1, i.e. it is always possible to avoid the disturbance growing; ν_x is exactly 1 (disturbance does not change) only in a small area, when both velocity $\dot{\theta}_0$ and step size x_{st} are small. Therefore, (almost) any curve u_x inside $\bar{C}_\infty^{x_{st}}$ produces a stable step-size controller.

The value of ν_x is smallest, $\nu_x = 0$, in the extended 1-step controllable region $\bar{C}_1^{x_{st}}$, the lightest area in the Fig. 6.3a. For the step sizes from $\bar{C}_1^{x_{st}}$ it is possible, with an appropriate push-off, to reach the nominal velocity $\dot{\theta}^*$ exactly (i.e. to fully eliminate disturbance) in one step. Hence, from the speed-of-convergence point of view, the best step-size controller curves are those inside the region $\bar{C}_1^{x_{st}}$. Two examples of such one-step dead-beat controllers are shown by the dashed



(a) Stability of the step-size controller; example controllers.



(b) Proposed step-size controller.

Figure 6.3: Step-size controller for Ranger. (continued on the next page)

and dash-dot lines in the figure. Each of them passes through the nominal trajectory of Ranger. The dash-dot controller always picks the smallest step size among those in $\bar{C}_1^{x_{st}}$ (except near the nominal trajectory) — as we discuss in Section 6.2.3 below, small step-sizes are preferable for Ranger for both robustness

Figure 6.3: (a) Consider stabilization of the IP model of Ranger around the nominal trajectory. For each initial velocity $\dot{\theta}_0$ and step size x_{st}, the quotient ν_x tells by how much the velocity error can be diminished in one step, given the best push-off (see (6.5), (6.6) for definition). Smaller ν_x indicates a bigger error decrease, faster convergence to the nominal trajectory, better step-size controller. The color scheme and contour lines show levels of ν_x for all points in the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$. Three example controllers are shown. The dashed line represents a simple dead-beat controller: $\nu_x = 0$, all disturbance is eliminated in one step. The dash-dot line is a dead-beat controller with the smallest step-sizes (good for robustness). The dotted controller is linear, but has small error reduction. Note, all controllers with fast convergence ask for larger than the nominal step-sizes for any disturbance. **(b)** The step-size controller proposed for Ranger is illustrated. It is represented by an absolute-value function (6.10). Such qualities as energy efficiency, robustness, stability, and speed of convergence are considered in the controller design. Energy efficiency is taken into account by the choice of a nominal trajectory with low energy use. Robustness by keeping the controller curve away from the boundaries of the region $\bar{C}_\infty^{x_{st}}$ — hence, a greater perturbation is required to fail the robot. Stability and speed of convergence by making sure the relative error quotient ν_x is 'sufficiently below' 1 for any velocity. For the proposed controller, $\nu_x < 0.62$: at most 62% of the initial disturbance remains after one step.

and safety reasons. On the other hand, the dash controller has a simple structure, being represented by an absolute value function.

Another example, shown by the dotted line, is a linear step-size controller. The convergence to the nominal velocity is slow (the error ν_x is close to 1) for this controller in the case of small perturbed velocities. Notice, that no linear controller, which passes through the nominal trajectory, allows fast convergence (small ν_x) for all perturbations. It appears preferential for Ranger to correct for any disturbance by taking larger steps than the nominal x_{st}^*. Section 6.5 provides more insights into the benefits of larger steps.

6.2.3 Robustness

We want the robot to be robust — it has to be able to avoid a failure (be ∞-step viable) and to return to the nominal trajectory (∞-step controllable) in the

presence of model errors, process and sensor noise, and external disturbances. These random processes may have a complex (e.g. non-Gaussian) nature and are generally hard to model accurately. A detailed discussion on noises and uncertainties in Ranger and their effects on Ranger's performance can be found in [16].

For the design of our step-size controller (6.4a), we assume that the effects of random disturbances and errors can be represented by random changes in the midstance velocity $\dot{\theta}_0$ and the step size x_{st} assigned by the controller, i.e. in the extended state $(\dot{\theta}_0, x_{st})$. Note, that this assumption takes into account possible process noise in the push-off as well, because velocity $\dot{\theta}_0$ is a function of the push-off at the previous collision. Hence, the goal for a robust controller is to minimize the probability of such random perturbations pushing the extended state outside of $\bar{C}_\infty^{x_{st}}$ $(= \bar{V}_\infty^{x_{st}})$. As a simple proxy for this probability, we use the distance to the boundary of $\bar{C}_\infty^{x_{st}}$: the closer the system is to the boundary, the more likely it is to be pushed outside of the region.

A step-size controller only determines the vertical coordinate of an extended state $(\dot{\theta}_0, x_{st})$, while the horizontal coordinate, the velocity $\dot{\theta}_0$, is always given. That is, the controller determines the distance from the extended state to only the top and bottom boundaries of $\bar{C}_\infty^{x_{st}}$ (see Fig. 6.2b). Therefore, from the point of view of robustness, we want the step-size controller curve to stay possibly far from these two boundaries, which correspond to the maximum allowed step-size (6.1b) and the minimum allowed swing-time (6.1c) respectively.[4]

The maximum-step-size boundary is a horizontal line. Maximizing the dis-

[4] We neglect the distance to the boundary portion in the top-right corner of $\bar{C}_\infty^{x_{st}}$, which corresponds to zero leg-tension just before heel-strike. First, this portion is short, compared to other boundaries. Second, it is in the range of large velocities ($\dot{\theta}_0 > 0.81$) that are far beyond Ranger's normal range of walking.

tance to it is equivalent to minimizing the step size x_{st}. We also note, that apart from the robustness reasons, large steps are not desirable for Ranger because of greater collision impacts and, as a result, a higher chance of a mechanical damage to the robot. The minimum-swing-time boundary is well approximated by a straight line going through the origin:

$$x_{st} = \alpha\dot{\theta}_0, \qquad \text{where } \alpha = 0.82. \tag{6.7}$$

The distance from an arbitrary point $(\dot{\theta}_0, x_{st})$ to this line is given by

$$d = \beta\,(x_{st} - \alpha\dot{\theta}_0), \tag{6.8}$$

where $\beta = 1/\sqrt{1+\alpha^2} \approx 0.77$. If $d < 0$, the extended state $(\dot{\theta}_0, x_{st})$ is outside of $\bar{C}_\infty^{x_{st}}$.

Thus, the robustness optimization of the step-size controller (6.4a) has two objectives:

- minimization of the step size x_{st};
- maximization of the distance d as in (6.8) — this approximately minimizes the probability of violating the minimum-swing-time restriction (6.1c).

These two objectives compete with each other (and with the stability objective discussed in Section 6.2.2 above) inside the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$. We resolve this competition and design a step-size controller for Ranger in Section 6.2.5.

6.2.4 Efficiency

By energy efficiency we mean little energy used by the robot for walking. The total energy cost includes both the work done by the motors and *electrical over-*

heads (energy required to run all electronics on the robot, such as microprocessors and sensors).

Our approach to energy efficient walking is as follows. We choose an energy-efficient nominal trajectory of the robot. We then design a controller which attempts to keep the motion close to the nominal trajectory at all times. Our assumption is that the energy use does not grow significantly, as long as the robot stays 'close' to the nominal trajectory (see Appendix D.3 on page 193 for supporting calculations). At the same time, in the case of large disturbances, we first want to make sure the robot does not fail and only worry about energy cost when it is safe to do so (i.e. when the robot is again close to the nominal trajectory).

The nominal trajectory we use for our IP model of Ranger is defined by (6.2). It corresponds to the reference trajectory used during Ranger's 65.2 km walk, which was designed to approximately minimize energy use of the robot [9]. The measure of efficiency used during the numerical minimization was the *total cost of transport*:

$$\text{TCOT} = \frac{\textit{total energy used by robot}}{\textit{weight} \times \textit{distance traveled}}. \tag{6.9}$$

TCOT is a non-dimensional number. For the ultra-marathon walk of Ranger, TCOT $\approx$ 0.28 [9] — compare it with the estimated 0.2 for humans [13], 3.2 for ASIMO [21, 22], 5 for Petman [9], and 0.2 for Cornell powered biped with knees [22].

Thus, taking energetic costs into account in the design of our walking controller for Ranger amounts to choosing trajectory (6.2) as the nominal trajectory, i.e. the speed $\dot{\theta}^*$ as the target speed at midstance.

6.2.5 Step-size controller

In Section 6.2 above we considered various optimization objectives for the design of our step-size controller (6.4a): stability, speed of convergence, robustness, and energy efficiency. Putting them together, the desired controller is represented by a curve in the $(\dot{\theta}_0, x_{st})$-plane which has to

- stay inside the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$ (viability & controllability, Sec. 6.2.1);
- pass through the nominal-trajectory point $(\dot{\theta}^*, x_{st}^*)$ (energy efficiency, Sec. 6.2.4);

and which tries to compromise between four competing goals:

(i) maximization of the distance d to the minimum-step-time boundary of $\bar{C}_\infty^{x_{st}}$; d is given by (6.8) (robustness, Sec. 6.2.3);

(ii) minimization of the step size x_{st} (robustness & robot damage prevention, Sec. 6.2.3);

(iii) minimization of the relative error ν_x after one step, defined by (6.6) (stability & speed of convergence, Sec. 6.2.2);

(iv) simplicity and little amount of computation.

Using these criteria as guidance, we propose the step-size controller presented on Fig. 6.3b (our choice is explained in the discussion below). The controller curve is a (scaled) absolute value function, centered at the nominal trajectory $(\dot{\theta}^*, x_{st}^*)$:

$$x_{st} = u_x(\dot{\theta}_0) \;=\; x_{st}^* + \alpha|\dot{\theta}_0 - \dot{\theta}^*|, \tag{6.10}$$

where $\dot{\theta}^*$ and x_{st}^* are given by (6.2a) and (6.2b) respectively. The coefficient α is such that the right arm of the controller curve (for which $\dot{\theta}_0 > \dot{\theta}^*$) is approximately parallel to the bottom boundary of $\bar{C}_\infty^{x_{st}}$, which corresponds to the minimum step-time constraint (6.1c). Thus, α is defined by equation (6.7).

We justify the choice of the controller (6.10) as follows. First, we would like the controller to be as simple as possible. As discussed in Section 6.2.2, step sizes bigger than the nominal x_{st}^* are preferred for Ranger for fast convergence (goal (iii) above). A (scaled) absolute value function centered at $(\dot{\theta}^*, x_{st}^*)$ is a simple curve satisfying $x_{st} \geq x_{st}^*$ and passing through the nominal trajectory. Two examples of such absolute-value controllers are the proposed controller (6.10) on Fig. 6.3b and the dashed-line controller on Fig. 6.3a. Note, that an absolute value function is a form of the controller that we guess is convenient for the particular nominal trajectory (6.2) that we use as an example. Extension of the controller for different nominal trajectories (e.g. higher target velocities) may require considering a more general form of the controller. For example, a slight generalization of the controller (6.10) is one with different slopes for velocities smaller than the nominal $\dot{\theta}^*$ than for velocities larger than $\dot{\theta}^*$.

Next, given the absolute-value form of the controller, the coefficient α (i.e. the slopes of the controller function) has to be decided. Our proposed controller (6.10) is such that the distance d to the step-time boundary of $\bar{C}_\infty^{x_{st}}$ is never smaller than it is for the nominal trajectory (goal (i) above). That is, the proposed controller is always as robust as when the robot is exactly on the nominal trajectory. A controller that is closer to the step-time boundary (i.e. smaller α) implies a smaller robustness: a higher risk of the robot's failure because there is not enough time to take a step. On the other hand, a larger distance to the

boundary (larger α) increases the robustness, but also requires taking bigger steps, which is not desirable due to a higher risk of a robot damage (goal (ii) above). Thus, the controller (6.10) is the compromise we choose between the competing objectives (i) and (ii): it is a controller with the smallest step-sizes that has the robustness of the nominal trajectory.

Similar reasoning can be used with regard to the speed of convergence (the relative error ν_x, goal (iii)). The coloring of Fig. 6.3b repeats that of Fig. 6.3a and shows the levels of the error ν_x inside the region $\bar{C}_\infty^{x_{st}}$. A controller curve closer to the step-time boundary (smaller α) means slower convergence (larger error ν_x), while a controller farther from the boundary (bigger α) requires larger step-sizes (higher risk of physical damage). For our proposed controller (6.10), $\nu_x < 0.62$ for any velocity. That is, at most 62% of the disturbance remains after one step, which we guess to be a sufficiently fast convergence.

Thus, based on the discussion above, we suggest (6.10) as a step-size controller for Ranger: for any disturbance, the controller is as robust as on the nominal trajectory and returns back to the nominal trajectory 'sufficiently fast'; at the same time the controller is energy efficient and simple. Note though, that the proposed controller is not unique, both its form (absolute value function) and parameter value (α) are hand-picked, based on our intuitive choices (e.g. of a *'safe'* distance d to the boundary, *'fast'* convergence, and *'simple'* form). Different priorities between the desired properties of the controller may lead to a quantitatively different result. For example, a smaller accent on avoiding large steps may suggest a bigger value of the coefficient α (e.g. the dead-beat controller on Fig. 6.3a represented by a dashed curve), which improves both the robustness and the speed of convergence. Alternatively, no requirement of a simple struc-

ture of the controller may suggest the minimum-step-size dead-beat controller shown by a dash-dot line on Fig. 6.3a.

6.2.6 Push-off controller

We now discuss the push-off controller (6.4b), which complements the step-size controller (6.10) that we designed in Section 6.2.5 above. For a given midstance velocity $\dot{\theta}_0$, the push-off controller u_p assumes (6.10) (Fig. 6.3b) as the next step-size x_{st}, and computes the push-off amount p to be applied just before the next collision.

We follow the same approach we used to the design the step-size controller. First, we find all push-offs that are viable (i.e. the robot does not fail in the next step) for each velocity $\dot{\theta}_0$ and the corresponding step-size $x_{st} = u_x(\dot{\theta}_0)$. Next, among all such push-offs we choose those that 'best' compromise between robustness, speed of convergence, energy use, and simplicity.

For a given velocity $\dot{\theta}_0$ the range of viable push-offs is determined by the Poincaré map (3.14), constraints (3.15), and actuator limitations (6.1). The set of all $\dot{\theta}_0$ and p that satisfy these equations (assuming the step-size controller (6.10)) is shown on Fig. 6.4 as the colored region $\bar{V}^p_{u_x}$. We found $\bar{V}^p_{u_x}$ by drawing its boundaries, which correspond to one of the constraints being active, i.e. turning into equality. From above, the region is bounded by the zero-ground-impact curve (constraint (3.15a)) and by the maximum allowed push-off restriction (6.1a). Note, that the non-smoothness of the zero-ground-impact curve at the nominal velocity $\dot{\theta}_0 = \dot{\theta}^*$ is due to the kink in the absolute-value function of the step-size controller (6.10) (see Fig. 6.3b). From below, the region is limited by

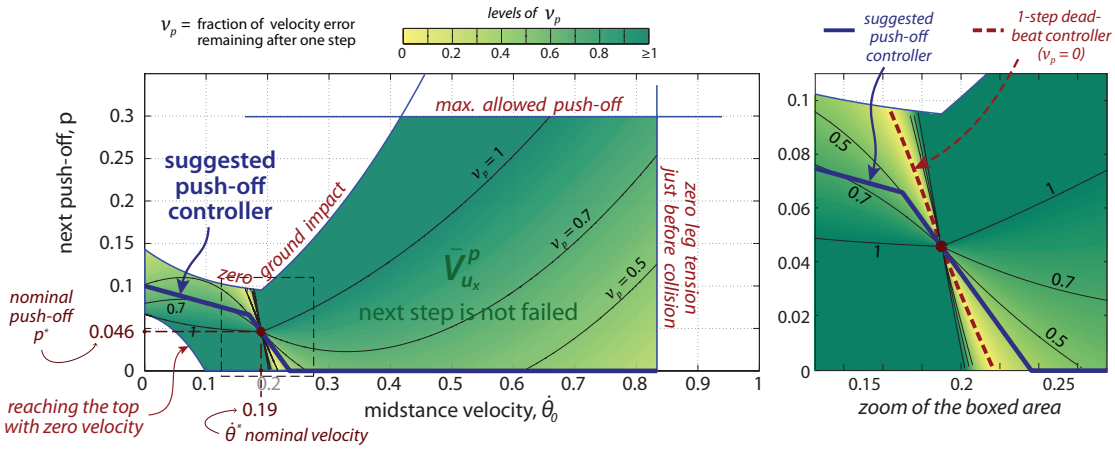


Figure 6.4: Push-off controller for Ranger. The proposed push-off controller for Ranger is a piece-wise linear function (6.12) of midstance velocity $\dot{\theta}_0$. It has two linear intervals, one for small velocities and one for velocities around the nominal $\dot{\theta}^*$, and is identically zero for large velocities. The controller is designed assuming the step-size controller (6.10) (Fig. 6.3b) and taking into account energy efficiency, robustness, and speed of convergence. Energy efficiency is accounted for by choosing a nominal trajectory with low energy use. Robustness by keeping the controller curve farther inside the region $\bar{V}^p_{u_x}$ (all velocities and push-offs for which the next step is not failed). Speed of convergence is defined by the quotient ν_p, the fraction of the initial disturbance that remains after one step. The colors and level lines show the values of ν_p for all points inside $\bar{V}^p_{u_x}$. The proposed controller is such as to always keep ν_p 'sufficiently small': $\nu_p < 0.72$. The picture on the right zooms into the area around the nominal trajectory.

the x-axis (zero push-off line, constraint (3.15d)) and by the requirement for the robot to reach the next midstance (the corresponding boundary curve is found by setting $\dot{\theta}_1 = 0$ in the Poincaré map (3.14)). Finally, the vertical boundary on the right corresponds to the non-negative leg-tension constraint (3.15b).

The push-off controller u_p is a curve in the $(\dot{\theta}_0, p)$-plane, which we want to be inside $\bar{V}^p_{u_x}$ and pass through the nominal trajectory point $(\dot{\theta}^*, p^*)$. The controller curve should be simple and correspond to a robust, energy efficient, and 'fast'-converging controller. We consider the mentioned qualities of the controller the same way as in the design of the step-size controller in Sections 6.2.2 – 6.2.4. Energy efficiency is taken into account only by the choice of the nominal trajectory with low energy use (Section 6.2.4). Robustness is represented by the distance

of the controller curve to the boundary of the region $\bar{V}^p_{u_x}$ (Section 6.2.3): if the curve is closer to the boundary, it is more likely that a random perturbation will cause the robot to fail. For the speed of convergence, we look at the relative error ν, defined by (6.5): ν is the fraction of the initial error in velocity that remains after one step, for a given initial velocity $\dot{\theta}_0$, step size x_{st}, and push-off p (Section 6.2.2). We call ν_p the relative error ν that assumes the step-size controller (6.10):

$$\nu_p(\dot{\theta}_0, p) = \nu(\dot{\theta}_0, u_x(\dot{\theta}_0), p). \tag{6.11}$$

Both the color scheme and level lines on Fig. 6.4 show the relative error ν_p for all points inside the region $\bar{V}^p_{u_x}$. For faster convergence to the nominal trajectory, the push-off controller curve has to be in possibly lighter areas of the region (smaller ν_p).

Based on the ν_p map, we want the controller to be approximately linear around the nominal velocity $\dot{\theta}^*$, use large push-offs for small velocities and small push-offs for big velocities. We use a simple curve that satisfies these requirements and is inside the region $\bar{V}^p_{u_x}$: a piece-wise linear controller that goes through the point $(\dot{\theta}^*, p^*)$ and is bounded from below by zero:

$$\begin{aligned} p = u_p(\dot{\theta}) &= \max\left\{\min\left\{p^* - \alpha_p(\dot{\theta}_0 - \dot{\theta}^*), \quad p_0 - \beta_p\dot{\theta}_0\right\}, \quad 0\right\} \\ &= \begin{cases} p_0 - \beta_p\dot{\theta}_0, & \text{if} \quad \dot{\theta}_0 < \frac{p^* + \alpha_p\dot{\theta}^* - p_0}{\alpha_p - \beta_p} \\ p^* - \alpha_p(\dot{\theta}_0 - \dot{\theta}^*), & \text{if} \quad \frac{p^* + \alpha_p\dot{\theta}^* - p_0}{\alpha_p - \beta_p} \le \dot{\theta}_0 \le \dot{\theta}^* + \frac{p^*}{\alpha_p} \\ 0 & \text{if} \quad \dot{\theta}_0 > \dot{\theta}^* + \frac{p^*}{\alpha_p} \end{cases} \end{aligned} \tag{6.12}$$

where α_p, β_p, and p_0 are positive controller parameters. The push-off controller (6.12), with α_p, β_p, and p_0 given by (6.13) as explained below, is shown on Fig. 6.4. The controller has two linear intervals (one for small velocities, the other for velocities close to the nominal $\dot{\theta}^*$) and is zero for large velocities.

The linear coefficient α_p determines the slope of the controller for small disturbances, i.e. for velocities around the nominal $\dot{\theta}^*$. As one can see from Fig. 6.4, the slope affects the speed of convergence to the nominal velocity (i.e. the relative error ν_p). Notice, that the one-step dead-beat controller (the $\nu_p = 0$ level line) is approximately a straight line. Hence, the convergence-optimal value of α_p is the one which approximates the dead-beat controller: $\alpha_p \approx 2$. However, we prefer to use a smaller slope in order to avoid high gains in the robot controller and also stay away from the dark areas ($\nu_p > 1$, divergence) both below and above the nominal trajectory. We hand-pick

$$\alpha_p = 1 \tag{6.13a}$$

such that the convergence remains 'sufficiently fast': $\nu_p < 0.55$ for small disturbances for the value of α_p above.

The parameters β_p and p_0 define the linear part of the push-off controller for small velocities: β_p is the slope of the controller, while p_0 is its vertical shift (the point of intersection of the controller with the y-axis). From the point of view of robustness, p_0 should be not too large (as to avoid the zero-ground-impact boundary on the top) and not too small (to avoid the reaching-the-top boundary on the bottom). On the other hand, larger p_0 is preferential due to faster convergence to the nominal trajectory (smaller ν_p, lighter areas in the figure). Similarly, for a given p_0, larger slope β_p corresponds to larger distance from the top boundary, but slower convergence. We hand-pick the values of β_p and p_0, such that the convergence is not 'too slow' (say, $\nu_p < 0.75$) and the controller is roughly 'parallel' to the top boundary:

$$p_0 = 0.1, \tag{6.13b}$$

$$\beta_p = 0.2. \tag{6.13c}$$

Given the values of the control parameters (6.13), ν_p is at most 0.72 for any disturbance. The maximum is reached at $\dot{\theta}_0 \approx 0.12$.

Note also, that saturation of the controller (6.12) from below by $p = 0$ (hence, the controller curve lying exactly along the $p = 0$ boundary of $\bar{V}^p_{u_x}$ for large velocities) is not a threat to robustness of the robot, because the robot is physically not capable of generating negative push-offs.

The piece-wise linear controller (6.12) with parameters (6.13) is our push-off controller for Ranger and is shown on Fig. 6.4. It provides all of robustness, energy-efficiency, and convergence and has a simple structure. This push-off controller is not unique: different priorities between the mentioned properties may lead to a quantitatively different controller.

6.2.7 Summary of the Ranger controller

We designed in Section 6.2 above a high-level walking controller for the Ranger robot and the nominal (desired) trajectory (6.2). At each midstance, depending on the current velocity $\dot{\theta}_0$, the high-level controller corrects the robot's trajectory by changing two parameters: the size of the next step x_{st} and the push-off impulse p just before the next collision. The new values of x_{st} and p are given by the step-size controller (6.10) and push-off controller (6.12) respectively:

$$x_{st} = u_x(\dot{\theta}_0) = x^*_{st} + \alpha|\dot{\theta}_0 - \dot{\theta}^*|, \tag{6.14a}$$

$$p = u_p(\dot{\theta}_0) = \max\left\{\min\left\{p^* - \alpha_p(\dot{\theta}_0 - \dot{\theta}^*), \quad p_0 - \beta_p\dot{\theta}_0\right\}, \quad 0\right\}. \tag{6.14b}$$

The high-level controller has three parameters defining the nominal trajectory:

$\dot{\theta}^*,\ x_{st}^*,\ p^*$	Nominal speed, step size, and push-off given by (6.2). p^* is a function of $\dot{\theta}^*$ and x_{st}^* as defined by (3.14).

and four tuning parameters stabilizing the motion around the nominal trajectory:

α	The slope of the step-size controller; given by (6.7). Increase in α improves robustness and convergence, but makes steps bigger (higher risk of robot damage).
α_p	The slope of the push-off controller around the nominal velocity; given by (6.13a). Larger α_p implies faster convergence, but higher control gains.
p_0	The constant term in the push-off controller for small velocities; given by (6.13b). Bigger p_0 decreases robustness, but increases the speed of convergence.
β_p	The slope of the push-off controller for small velocities; given by (6.13c). Larger β_p improves robustness, but decreases the speed of convergence.

Our controller (6.14) defines high-level behavior of Ranger: where to step and how hard to push off. Implementation of the controller on the physical robot requires designing low-level controllers which realize the commanded high-level behavior. Such low-level controllers may, for example, be the currents provided to each of the motors on the robot at each instant of time. In this thesis, we do not consider low-level controls. Hand-tuning of the parameters α, α_p, β_p, and p_0 may also be required for 'best' performance of the robot.

6.3 Simulation results

We test in simulation the walking controller (6.14) which we designed for Ranger in Section 6.2. The simulation of the 2D IP model of Ranger and all related calculations are done in Matlab.

To test the performance of the controller, we allow random disturbances to the model. These may be due to external disturbances (wind, uneven ground, etc.), process and sensor noise, and model errors. We assume that in our simple model of Ranger the effects of all such disturbances can be represented by random perturbations in the midstance velocity $\dot{\theta}_0$ and the two controls, the step size x_{st} and push-off p. We call these random perturbations μ_v, μ_x, and μ_p respectively:

$$\tilde{\dot{\theta}}_0 = \dot{\theta}_0 + \mu_v, \tag{6.15a}$$

$$\tilde{x}_{st} = x_{st} + \mu_x = u_x(\tilde{\dot{\theta}}_0) + \mu_x, \tag{6.15b}$$

$$\tilde{p} = p + \mu_p = u_p(\tilde{\dot{\theta}}_0) + \mu_p. \tag{6.15c}$$

Perturbation μ_v accumulates all disturbances that affected the robot's velocity during the previous step. Note, that we let the new (perturbed) velocity $\tilde{\dot{\theta}}_0$ be available to both the step-size controller u_x and the push-off controller u_p. We assume that possible state-estimation errors are taken into account by the controller perturbations μ_x and μ_p.

First, we simulate the controller (6.14) in the absence of noise, i.e. when $\mu_v = \mu_x = \mu_p = 0$. Fig. 6.5 displays two trajectories of the robot over 10 steps, showing the midstance velocity $\dot{\theta}_0$ as a function of the step number. One trajectory starts with the initial velocity $\dot{\theta}_0 = 0.05$, the other with $\dot{\theta}_0 = 0.5$. In both cases, the robot returns to within 10% of the nominal velocity $\dot{\theta}^*$ in five steps, and then

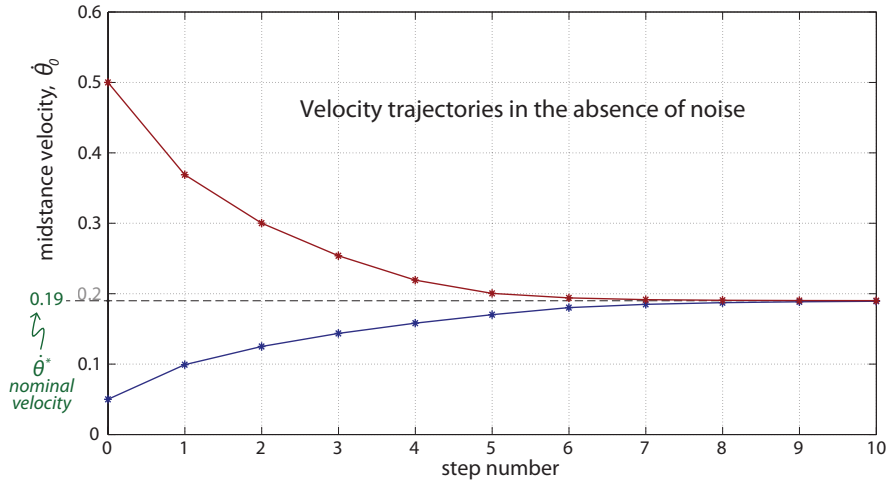
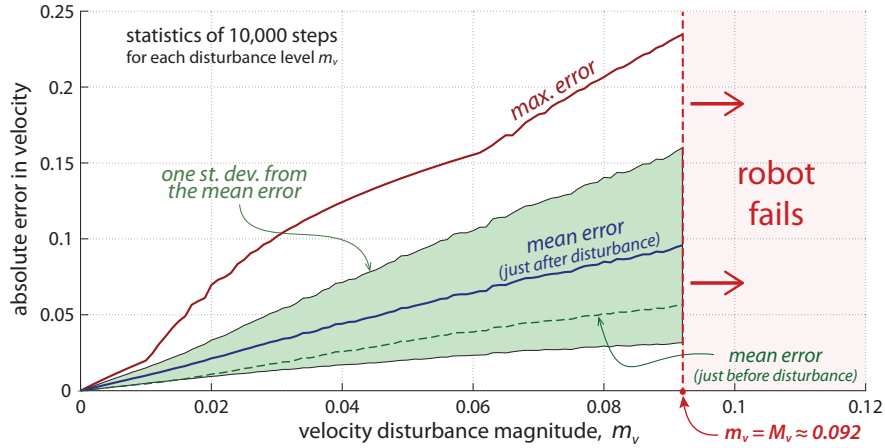


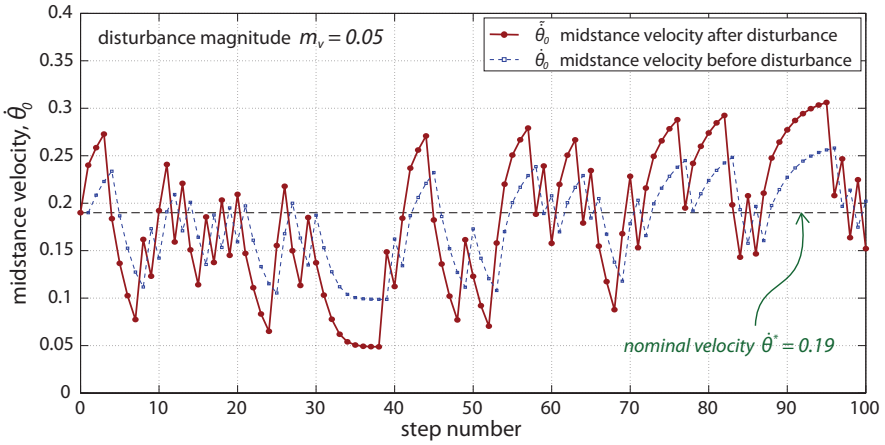
Figure 6.5: Simulation of the proposed controller with no noise. The simple model of Ranger with the proposed controller (6.14) is simulated. The evolution of the midstance velocity $\dot{\theta}_0$ over ten steps is shown for two different initial conditions. No noise or external disturbances are considered. For both initial velocities, the robot returns to within 10% of the nominal velocity $\dot{\theta}^*$ in five steps.

approaches $\dot{\theta}^*$ asymptotically. The total energy costs (TCOT, as defined in Appendix D.3) of the ten steps are 0.131 and 0.093 correspondingly, as compared to the 0.12 cost of the nominal trajectory (Fig. D.2).

Next, we consider disturbances (6.15) one at a time, starting with the velocity perturbations μ_v. We test the proposed controller by looking at its performance in the 'worst scenarios', when disturbances are largest. That is, we model the random disturbance μ_v as always taking on a value with the largest allowed magnitude: μ_v is randomly selected as either $+m_v$ or $-m_v$, where m_v is a given *noise level*. We consider different noise levels in the range $0 \le m_v \le 0.12$ and for each m_v we run a simulation over N steps, with $N = 10{,}000$ for simulations presented here. Each simulation starts at the nominal velocity $\dot{\theta}^*$ — for other initial velocities the motion converges to a neighborhood of $\dot{\theta}^*$ after the first few steps (see the discussion below). The statistics of each simulation are collected and displayed on Fig. 6.6a as a function of m_v. The average deviation of the per-



(a) Controller performance for different magnitudes of disturbance



(b) Example velocity trajectory

Figure 6.6: Simulations with random velocity perturbations. The simple model of Ranger with the proposed controller (6.14) is simulated. At each midstance a random perturbation $\mu_v = \pm m_v$ of a fixed magnitude m_v is added to the robot's velocity. **(a)** A series of 10000-step-long simulations is run for a range of different m_v. Displayed are the statistics of simulations as a function of m_v: the mean, maximum, and standard deviation of the error in perturbed velocity $\tilde{\dot{\theta}}_0$, and the mean error of the velocity $\dot{\theta}_0$ just before the perturbation. In all simulations with $m_v > M_v \approx 0.092$ (48% of the nominal velocity $\dot{\theta}^*$) the robot fails. **(b)** An example simulation with the disturbance magnitude $m_v = 0.05$: the trajectories of both $\tilde{\dot{\theta}}_0$ and $\dot{\theta}_0$ are shown.

turbed velocity $\tilde{\dot{\theta}}_0$ from the nominal $\dot{\theta}^*$, i.e. the mean error $\Delta\tilde{\dot{\theta}}_0 = |\tilde{\dot{\theta}}_0 - \dot{\theta}^*|$, is approximately equal to the disturbance magnitude m_v for all simulations.

In our simulation trials the robot consistently completes 10,000 steps without

a failure when

$$0 \le m_v \le M_v, \qquad \text{where } M_v \approx 0.092. \tag{6.16a}$$

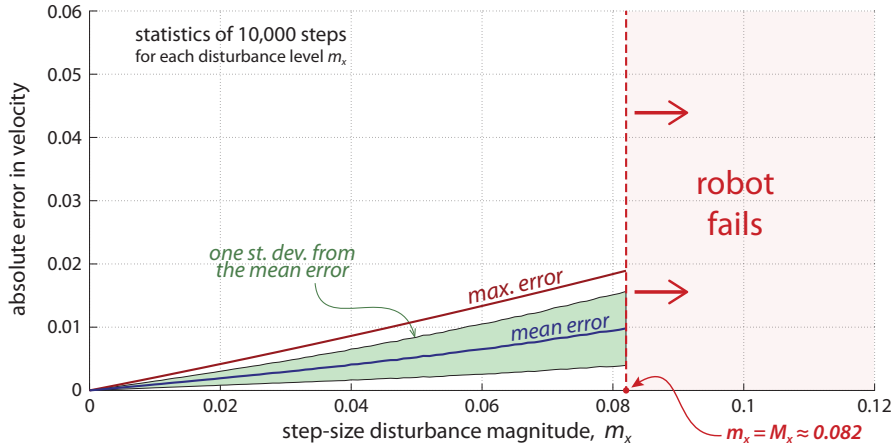
On the other hand, the robot consistently fails for any disturbance magnitude $m_v > M_v$. Each failure is due to a perturbation rendering the midstance velocity negative (hence, backwards motion which we do not allow). Therefore, we say that M_v is the largest magnitude of velocity perturbations, such that the proposed controller (6.14) is reliable (assuming no other disturbances in the model). The value of M_v above is about 48% of the nominal velocity $\dot{\theta}^*$. Fig. 6.6b shows an example trajectory of the robot for $m_v = 0.05$ over a 100 steps. The trajectories of both the perturbed velocity $\tilde{\dot{\theta}}_0$ and the midstance velocity $\dot{\theta}_0$ just before the perturbation are shown.

Similar to the velocity perturbations μ_v, we consider the noises μ_x and μ_p in the step-size and push-off controls respectively. Both noises are modeled as always taking on the maximum allowed magnitude: $\mu_x = \pm m_x$, $\mu_p = \pm m_p$. Fig. 6.7a shows the statistics of the 10,000-step-long simulations with only the step-size noise present in the model. The robot walks reliably (consistently completes 10,000 steps without a failure) for the noise levels m_x in the range

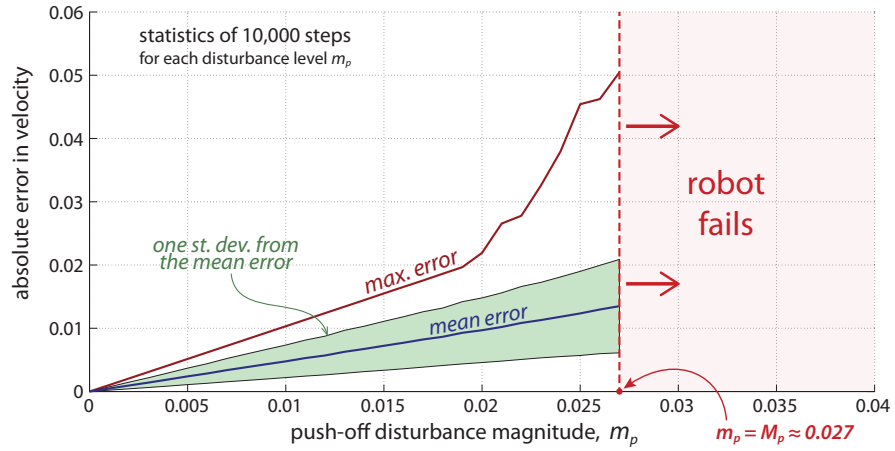
$$0 \le m_x \le M_x, \qquad \text{where } M_x \approx 0.082. \tag{6.16b}$$

The velocity $\dot{\theta}_0$ stays within 10% of the nominal velocity $\dot{\theta}^*$ (i.e. the maximum error is about 0.019) for the largest noise $m_x = M_x$ in the range above. For bigger noises, $m_x > M_x$, the robot fails because of flying after the push-off (violation of constraint (3.15a)). The maximum tolerable disturbance-magnitude M_x is about 23% of the nominal step-size x^*_{st} defined by (6.2b).

In the case of the push-off perturbations μ_p, as shown on Fig. 6.7b, the con-



(a) Step-size perturbations



(b) Push-off perturbations

Figure 6.7: Simulations with random disturbances in controls. Statistics of simulations analogous to those presented in Fig. 6.6a, but for different types of disturbances. Here the robot's controls are randomly perturbed, one at a time: **(a)** each step-size x_{st} is changed by $\mu_x = \pm m_x$, where μ_x is a given noise level; **(b)** each push-off is changed by $\mu_p = \pm m_p$. The robot consistently fails when $m_x > M_x \approx 0.082$ and $m_p > M_p \approx 0.27$ respectively. Each failure is due to the flight after the push-off. M_x and M_p are about 24% and 59% of the nominal values of the step-size and push-off correspondingly.

troller is reliable when

$$0 \le m_p \le M_p, \qquad \text{where} \quad M_p \approx 0.027. \tag{6.16c}$$

The largest tolerable noise M_p is approximately 59% of the nominal push-off p^* defined by (6.2c). For larger push-off noises, the robot fails to complete 10,000 steps due to flying after one of the push-offs.

We also test recovery of the robot from large initial disturbances. We consider two different initial velocities, $\dot{\theta}_0 = 0.01$ and $\dot{\theta}_0 = 0.7$. For each of them and for each different noise level we run $K = 1000$ simulations, each $N = 10$ steps long. In the case of the velocity perturbations μ_v,[5] for both initial velocities the robot consistently (in all 1000 runs) returns to a neighborhood of the nominal velocity $\dot{\theta}^*$ for any $m_v \leq M_v$. Similarly, when only the push-off perturbations are considered, all noise levels in the range $m_p \leq M_p$ are tolerable for both initial velocities. For the step-size perturbations μ_x, the tolerable disturbance levels are $m_x \leq 0.058$ ($\approx 0.7 M_x$) for the initial velocity 0.01 and $m_x \leq M_x$ for the initial velocity 0.7. In the case of the small initial velocity and $m_x > 0.058$, some of the 1000 simulations fail due to the flight phase after the push-off.

In simulations discussed above, we allowed only one type of random disturbances (μ_v, μ_x, or μ_p) at a time. The corresponding maximum tolerable noise-levels M_v, M_x, and M_p change when the disturbances are combined. We now run 10,000-step long simulations, each starting at the nominal velocity $\dot{\theta}^*$, for various combinations of the noise levels in the velocity (m_v), step-size (m_x), and push-off (m_p). Fig. 6.8 shows the surface, corresponding to the maximum tolerable combinations: for all triples (m_v, m_x, m_p) on and below this surface the robot in our trials completes 10,000 steps without a failure; for all (m_v, m_x, m_p) above the surface the robot fails. The points of intersection of the surface with the coordinate axes correspond to the largest disturbances M_v, M_x, and M_p given by (6.16).

[5] The first perturbation μ_v is applied at the end of the first step.

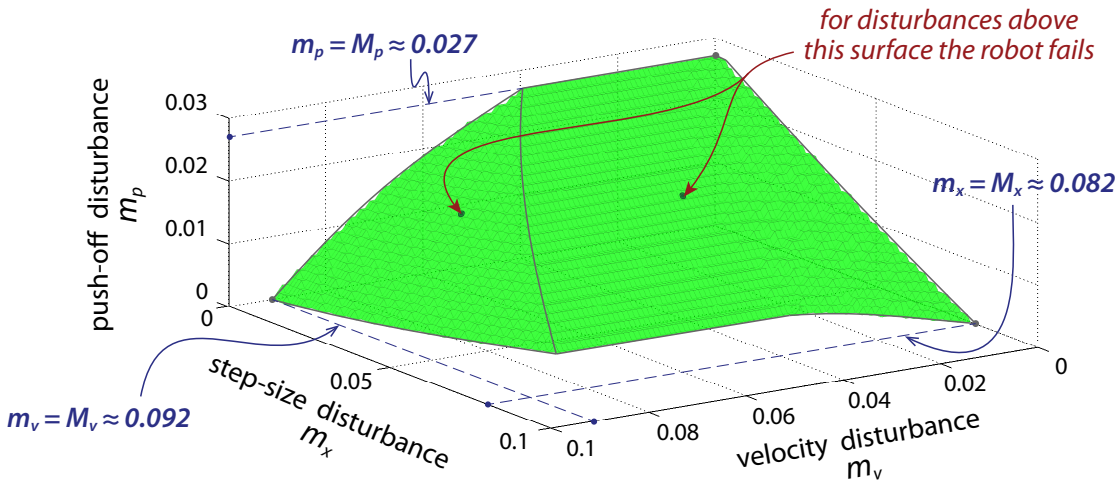


Figure 6.8: Largest tolerable disturbances. The simple model of Ranger with the proposed controller (6.14) is simulated. At each midstance random perturbations are added to the robot's velocity and controls for the next step (step-size and push-off). Perturbations' magnitudes m_v, m_x, and m_p are fixed in a single simulation. A series of 10,000-step long simulations is run for various combinations (m_v, m_x, m_p) of the magnitudes, starting at the nominal velocity $\dot{\theta}^*$. Displayed is the surface of maximum tolerable disturbances. For all (m_v, m_x, m_p) on and below the surface the robot completes 10,000 steps with no failures, for those above the surface the robot fails. The intersection of the surface with the m_v-axis is the largest velocity perturbation $M_v \approx 0.092$ that is tolerable in the absence of other perturbations (see Fig. 6.6a). Similarly, the intersections with the other axes are $M_x \approx 0.082$ (Fig. 6.7a) and $M_p \approx 0.027$ (Fig. 6.7b). M_v, M_x, and M_p are about 48%, 23%, and 59% of the nominal values of the velocity, step-size, and push-off respectively.

6.4 Controller design recipe

We designed in Section 6.2 a high-level walking controller for the Ranger robot. The controller is based on a simple 2D IP model and takes into account stability, robustness, energy efficiency, and simplicity. Here we summarize the controller design procedure, extending it to a more general model of a planar biped.

A Represent the robot by a point-mass model. The center-of-mass motion of any biped, we believe, is well approximated by point-mass models (see discussion in Section 2.5.4), such as the IP model (Chapter 3), LIP model (Chapter 4), and SLIP model (Spring-Loaded Inverted Pendulum, [12, 28, 68]).

Ranger is modeled here as the simple 2D IP model to develop a controller of its high-level behavior.

• *Impose constraints on the simple model to account for actuator restrictions of the real robot.* For Ranger, such restrictions include limited actuation in the ankles and limited leg-swinging abilities. These are represented by the step-size, step-time, and push-off constraints in the 2D IP model.

• *Define the Poincaré section of the simple model.* For the IP model of Ranger, the Poincaré section is at midstance, where the stance leg is vertical.

• *All further controller design bullets assume the point-mass model of the robot.*

B Determine the desired trajectory of the model. This may be a unique trajectory or a range of trajectories satisfying a given goal. Some examples of the goals are listed in Section 2.5.2. We suggest (and use for Ranger) the trajectory that is approximately optimal with respect to the energy use of the robot. The point where the desired trajectory crosses the Poincaré section of the model, is the *target region* for the controller. For Ranger, the target is a fixed nominal velocity at midstance.

C Compute the ∞-step controllable region extended by one of the controls. The control law is first designed for one control, u_1, and then for the other control(s) of the model for the next step. The choice of u_1 is not important. For the IP model of Ranger, we choose $u_1 = x_{st}$, the next step-size. The extended ∞-step controllable region $\bar{C}_\infty^{u_1}$ is all combinations of controls u_1 and initial states, such that the target can be reached in one or more steps. $\bar{C}_\infty^{u_1}$ can be computed numerically by the methods described in Section 2.7; for Ranger, $\bar{C}_\infty^{u_1}$ is shown in Fig. 6.2. A curve inside the region $\bar{C}_\infty^{u_1}$ defines a control law for u_1. The bullets D and E below aim to find the 'best' controller curve inside $\bar{C}_\infty^{u_1}$.

D Define the controller optimization objectives. These include any desired properties of the controller or the robot's motion. We suggest (and use for Ranger) the following. *Speed-of-convergence* ('stability'): how fast the model returns to the target after a disturbance (see Section 6.2.2). *Robusntess*: how big of a disturbance the model can tolerate without failing (Section 6.2.3). *Energy cost*: how much energy the robot requires for walking (Section 6.2.4).

• *Express all objectives as functions of the state and control* u_1. For Ranger these expressions are provided in the respective sections specified above.

• *Evaluate each objective for all points in the region* $\bar{C}_\infty^{u_1}$. This calculation shows which points in $\bar{C}_\infty^{u_1}$ are 'better' or 'worse' from the point of view of each objective. In the case of Ranger: the speed-of-convergence is addressed by Fig. 6.3a; the robustness is better for points farther from the boundaries of $\bar{C}_\infty^{u_1}$; the energy cost is 'relatively low' for all points and, thus, is neglected.

E Draw a simple controller curve through the 'best' parts of the extended ∞-step controllable region. Some points in $\bar{C}_\infty^{u_1}$ may be 'good' for one optimization objective, but 'bad' for the other. The parts of $\bar{C}_\infty^{u_1}$ that are 'best' for the controller depend on the priorities between different objectives. Both such priorities and the level of simplicity of the controller curve are manually determined by the robot designer for the specific robot. Hence, the controller curve is generally not unique. For Ranger, see the discussion on the choice of the step-size controller in Section 6.2.5.

F Assume the bullet E controller for the first control u_1, and repeat the bullets C, D, and E for the next control u_2. For Ranger, the second control parameter is the push-off just before the next collision, $u_2 = p$. The design steps C, D, E are similar to those for the step-size controller $u_1 = x_{st}$. The related discussion is provided in Section 6.2.6.

6.5 Discussion

There are two major steps in our controller design approach. First, we discover all controllers of the robot that do not lead to a failure and which also allow the motion to be stabilized around a given reference trajectory. That is, all controllers that the robot designer may want to consider. Second, given this set of controllers, various criteria, other than viability, can be used to pick the controller that is 'best' suited for the given robot and locomotion tasks. Some examples of such criteria include robustness, energy use, and convergence to the reference trajectory. Our approach reveals the trade-offs between the different criteria or desired properties of the controller. It shows that certain control values or types of a controller (e.g. linear, quadratic, etc.) may be beneficial with respect to one design objective, but less desirable with respect to another. For example, recall the trade-off between large steps and fast convergence in the case of Ranger (see Section 6.2.5). Thus, our design method suggests the 'best' controller for each given set of priorities between the controller objectives.

Bigger steps are good. As noted in Section 6.2.2, linear step-size controllers should, for most target velocities and step-sizes, be avoided for Ranger due to slow convergence to the nominal trajectory. Instead, in these cases, taking steps that are larger than the nominal is preferable for any disturbance. This partially agrees with Ranger's ultra-marathon walk controller [9], which was also discontinuous at the nominal velocity, using larger steps (with nominal push-offs) to slow down and nominal-size steps (with harder push-offs) to speed up.

The preference towards bigger steps can be explained as follows. In the case of too-large initial velocities, taking a larger step causes a bigger ground im-

pact and a bigger momentum loss at the collision. Hence, a decrease in the next-step velocity. For too-small initial velocities, consider the instant of collision (see Fig. 3.1 on page 65): the ground impact cancels all hip velocity along the colliding leg. Therefore, only the normal (to the colliding leg) component of the push-off impulse adds energy into the system. A larger step means a larger angle between the legs at the collision and a larger normal component of the push-off — thus, the push-off is more effective.[6] Also, recall that if the hip velocity along the colliding leg is too large after the push-off, the flight phase begins, i.e. the robot fails. A smaller step increases the tangential component of the push-off and, hence, of the hip velocity. Therefore, a larger step means a larger range of viable push-offs.

We conclude: *whether walking too slow or too fast, taking bigger steps increases the range of viable push-offs and makes the push-offs more effective at controlling speed.*

Some important future improvements of our controller design include: application of the method to 3D robots and extension of the method to control the target speed of the robot.

Extension to 3D models. The walking controller (6.14) was designed for the specific robot, Cornell Ranger, based on its simple point-mass model. On the other hand, discussion in Section 2.5.4 suggests that any biped's motion resembles that of a point-mass model. We believe, thus, that our controller design ideas can be used for most robots.

The design recipe in Section 6.4 extends the method to a general model of

[6] The push-off impulse still has to be larger than the nominal for the robot to speed-up.

a 2D biped. The same design method can be applied to 3D robots in a similar, however more computationally demanding, way: a 3D dynamical system implies both a higher-dimensional state space and a higher-dimensional control space. For example, the 3D version of the IP model [84, 24] has two dynamic variables at midstance and three control parameters each step (see Appendix E). The controllable regions extended by one of the controls (bullet C of the design procedure) are three-dimensional and, hence, computationally more expensive to calculate (compared to a planar model case).[7] Another consequence of a higher-dimensional system is a more difficult graphical representation and analysis of the calculation results. For example, steps D and E of the design procedure assume graphically determining which parts of the extended controllable region (i.e. which control values) produce the 'best' controller. In the case of Ranger, this results in drawing a controller curve inside the (two-dimensional) extended controllable region. In contrast, for the 3D IP model the controller is a *surface* inside the *three-dimensional* extended controllable region.

Control of the target speed. A weak point of our controller design is the requirement of a specific target trajectory. The target (nominal) speed, which the controller attempts to maintain at all times, is fixed. We would like to be able to control the nominal speed of the robot. We would like to extend our controller, such that the nominal speed is a parameter commanded to the robot at each midstance.

One possible approach involves (linear) interpolation between several fixed-target controllers. One can consider a number of different target speeds in a desired range of the robot's motion. Our controller design method can be used to

[7] Note, that all calculations are offline.

define a separate controller for each of the targets. At each midstance, given a desired nominal speed v_t of the robot, the control values are computed by (linear) interpolation between two (or more) predefined controllers. This approach, essentially a look-up table of controllers, requires significant human work.

A more promising approach to controlling the target is to consider the target speed as an extra dimension of the extended state space of the model. A point in the extended space, an extended state (q, v_t, u), now consists of the state q, target speed v_t, and control(s) u. The control law for each of the controls u can be design as a function of both q and v_t by our controller design method. This approach to a variable target speed is analogous to considering a higher-dimensional model, such as a 3D model discussed above. As for the extension to 3D models, this increase in dimensionality increases the control design calculations.

APPENDIX A

VIABILITY AND CONTROLLABILITY OF PASSIVE WALKERS

The *Simplest Walking (SW) model* is a 2D passive-dynamics model (Section 1.1.1), first studied by Garcia et al. [27]. We use the SW model to discuss viability and controllability of passive walkers.

The SW model has two rigid legs and all mass concentrated at the hip and at the point feet, see Fig. A.1. The foot mass is assumed to be much smaller than the mass at the hip; the effects of the swing-leg motions on the dynamics of the stance leg are neglected. Instantaneous collisions are assumed plastic, with no impulse acting on the hind leg from the ground. Scuffing during the swing phase is neglected. The SW model is fully passive: it has no actuators and nor sensors, and its motion is powered only by gravity.

Garcia et al. [27] showed that the SW model has two periodic gaits, one stable and one unstable, on any slope γ for $\gamma \lesssim 0.015$ rad.[1] The stable gait has retraction of the swing leg as it hits the ground, but such retraction is not observed in the unstable gait (instead, the swing leg moves forward as it collides). Such passive retraction may partially explain the difference in stability of the gaits [95, 34]. Standing still in the vertical position can be considered as another unstable 'periodic trajectory' of the model.

For a fixed slope $\gamma < 0.015$, possible behaviors of the model are as follows. When the initial state is exactly on one of the three periodic trajectories (one sta-

[1] In fact, Garcia et al. [27] point out that there are infinitely-many other periodic gaits, where the swing leg oscillates multiple times between heel-strikes. Such gaits are non-anthropomorphic and we do not consider them here. However, these gaits can be treated in our discussion here analogous, depending on their stability, to the stable gait q_{st} or unstable gait q_{un} that we consider.

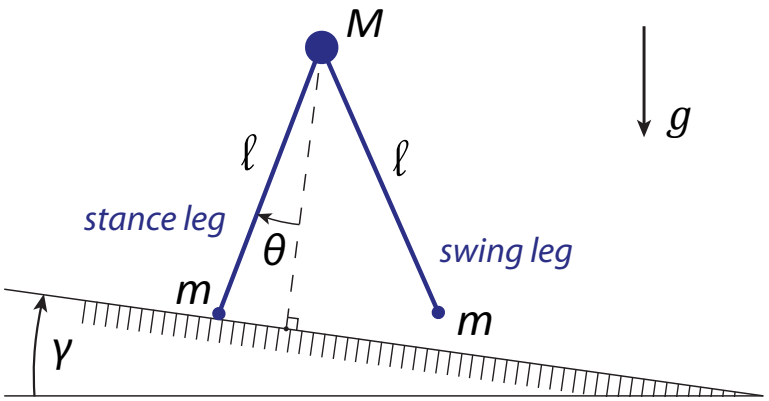


Figure A.1: Simplest walking model. A 2D bipedal model of a passive walker, first described by Garcia et al. [27]. The model has two rigid legs and a point mass at the hip and at each foot. The hip mass is assumed much larger than the foot mass. Effects of the swing-leg motions on the dynamics of the stance leg are neglected. Collisions are assumed plastic and instantaneous, with no impulse acting from the ground on the hind leg. Scuffing during the swing phase is ignored. The model has no motors and no sensors; it walks down a gentle slope powered by gravity alone.

ble and two unstable), the robot always remains on that trajectory. If the initial state is in the basin-of-attraction of the stable periodic gait, the robot asymptotically approaches that periodic gait. For all other initial states, the robot falls down after a finite number of steps.

Following Garcia et al. [27], we define the Poincaré section of the SW model to be at the instant just after collision. The dynamic state at that instant is described by two variables: the angle and angular rate of the stance leg (the swing-leg state is constrained by the impact conditions). All considerations below are in terms of states of the robot just after collision.

Let q_{st}, q_{un}, and q_{ss} be the states corresponding, respectively, to the stable periodic gait, unstable periodic gait, and standing still. Also, let the set B_{st} be the basin-of-attraction of the stable gait; B_{st} includes the state q_{st}. The viability kernel V_∞ is the set of all initial states (just after collision) such that the robot never falls down. Hence, V_∞ includes the basin-of-attraction B_{st} and two isolated

points q_{un} and q_{ss}:

$$V_\infty = B_{st} \cup \{q_{un}, q_{ss}\}. \tag{A.1}$$

The ∞-step controllable region C_∞ is all states, from where the robot reaches a given target or approaches it asymptotically. Assume that the target is the stable trajectory q_{st}. Then, the ∞-step controllable region is equal to the basin-of-attraction of the stable trajectory: $C_{\infty,st} = B_{st}$ (Fig. 3.7). The region $C_{\infty,st}$ and the viability kernel V_∞ differ only by two points q_{un} (the unstable periodic gait) and q_{ss} (standing still) — i.e. by a zero-measure set — which is consistent with our 'Viable is Controllable' claim in Section 2.3.1.

Assume that the target is standing still, as for Pratt's capture regions [64] (Section 1.2). The SW model does not reach the unstable standing-still point q_{ss} from any initial state, other than q_{ss} itself. Therefore, the ∞-step controllable region $C_{\infty,ss}$ corresponding to the target q_{ss}, is a single point: $C_{\infty,ss} = \{q_{ss}\}$. The regions $C_{\infty,ss}$ and V_∞ in this case differ by a non-trivial set, the basin-of-attraction B_{st} (and the unstable gait q_{un}). That is, the robot never comes to a stop unless already standing, but it has a substantial region in the state space, from where it keeps walking indefinitely. The standing-still target for the SW model is an example of an exceptional case, when our 'Viable is Controllable' claim does not work. This example was suggested by Koolen et al. [42].

APPENDIX B

SET-VALUED MAP REPRESENTATION OF EXTENDED CONTROLLABLE REGIONS

Here we prove equation (2.14) on page 45 in the text, which expresses the extended n-step controllable region $\bar{C}_n$ in terms of the extended controllability map $\hat{\bar{C}}$.

We can say that the region $\bar{C}_n$ includes all extended states for which the robot is able to reach the target either in exactly n steps (i.e. $\bar{C}_n$ includes all points in $\hat{\bar{C}}_n$) or in $n-1$ or fewer steps (i.e. all points in $\bar{C}_{n-1}$). Therefore, similar to equation (2.8) for the controllable regions, we write

$$\bar{C}_n = \hat{\bar{C}}_n \cup \bar{C}_{n-1}. \tag{B.1}$$

Sequentially applying the above equation to $\bar{C}_{n-1}$, $\bar{C}_{n-2}$, and so on, we get:

$$\begin{aligned}\bar{C}_n &= \hat{\bar{C}}_n \cup \bar{C}_{n-1} = \hat{\bar{C}}_n \cup \hat{\bar{C}}_{n-1} \cup \bar{C}_{n-2} = ... \\ &= \hat{\bar{C}}_n \cup \hat{\bar{C}}_{n-1} \cup ... \cup \hat{\bar{C}}_2 \cup \bar{C}_1 = \\ &= \hat{\bar{C}}_n \cup \hat{\bar{C}}_{n-1} \cup ... \cup \hat{\bar{C}}_2 \cup \hat{\bar{C}}_1 = \bigcup_{k=1}^{n} \hat{\bar{C}}_k.\end{aligned}$$

We used the equivalence (2.13): $\bar{C}_1 = \hat{\bar{C}}_1$. The above formula says that the extended n-step controllable region $\bar{C}_n$ consists of all extended states, from which the target is reachable in exactly one step (the region $\hat{\bar{C}}_1$), those which allow to reach it in exactly two steps ($\hat{\bar{C}}_2$), exactly three steps ($\hat{\bar{C}}_3$), and so on.

Next, we express the strict controllable regions $\hat{\bar{C}}_n$ in terms of the extended controllable map $\hat{\bar{C}}$, according to (2.12):

$$\bar{C}_n = \hat{\bar{C}}(\hat{C}_{n-1}) \cup \hat{\bar{C}}(\hat{C}_{n-2}) \cup ... \cup \hat{\bar{C}}(\hat{C}_1) \cup \hat{\bar{C}}(C_0) = \bigcup_{k=0}^{n-1} \hat{\bar{C}}(\hat{C}_k),$$

letting $\hat{C}_0 \equiv C_0$. We use the linear property of the transformation $\hat{\bar{C}}$ to get

$$\bar{C}_n = \hat{\bar{C}}(\hat{C}_{n-1} \cup \hat{C}_{n-2} \cup ... \cup \hat{C}_1 \cup C_0) = \hat{\bar{C}}\left(\bigcup_{k=0}^{n-1} \hat{C}_k\right).$$

Notice, that the argument of $\hat{\bar{C}}$ in the above equation, is a region in the state space of the model. This region includes all target states (C_0), all states from which the robot can reach the target in exactly one step ($\hat{C}_1$), in exactly two steps ($\hat{C}_2$), exactly three steps ($\hat{C}_3$), and so on — all the way up to exactly $n-1$ steps ($\hat{C}_{n-1}$). In other words, this region includes all initial states, from which the robot is able to reach the target in $(n-1)$ or fewer steps. This is exactly the description of the $(n-1)$-step controllable region C_{n-1}. Therefore,

$$\bar{C}_n = \hat{\bar{C}}(C_{n-1}). \tag{B.2}$$

APPENDIX C

SUPPLEMENTARY CALCULATIONS FOR THE 2D IP MODEL

C.1 Computation of the extended n-step controllable regions

We explain the numerical computation of the extended n-step controllable regions $\bar{C}_n^{x_{st}}$ for our 2D IP model for $n \geq 2$. The regions $\bar{C}_n^p$ are computed in a similar way. All calculations are performed in Matlab.

We use the iterative procedure described in Section 2.7.4 on page 60. For each n, the region $\bar{C}_n^{x_{st}}$ is calculated according to (2.24), using the previously found $\bar{C}_{n-1}^{x_{st}}$, C_{n-1}, and C_{n-2}:

$$\bar{C}_n^{x_{st}} = \bar{C}_{n-1}^{x_{st}} \cup \left(\bigcup_{\dot{\theta} \in C_{n-1} \setminus C_{n-2}} \hat{\bar{C}}(\dot{\theta}) \right), \qquad \textit{for } n \geq 2. \tag{C.1}$$

We already know the regions C_0, C_1, and $\bar{C}_1^{x_{st}}$, which are required for the first iteration (computation of $\bar{C}_2^{x_{st}}$): the target C_0 that we use is given by (3.17); the regions C_1 and $\bar{C}_1^{x_{st}}$ are found in Section 3.2.2 and shown on Figs. 3.2a and 3.2b on page 77 correspondingly. The four boundaries of $\bar{C}_1^{x_{st}}$ (not including the vertical axis) are given by equations (3.18).

We now describe how we numerically compute the union in the right-hand-side of (C.1). For our calculations, we discretize the velocity axis: we use $N = 1001$ grid points v_i, uniformly distributed in the interval $0 \leq \dot{\theta}_0 \leq 1$ (all non-failed velocities, V_0), such that $v_0 = 0$ and $v_{N-1} = 1$.[1] Next, consider a region A in the $(\dot{\theta}_0, x_{st})$-plane, which is convex in the x_{st} direction — i.e. any line segment, which connects two points in A and is parallel to the x_{st}-axis, lies entirely in A.

[1] N is chosen such that a single run of (C.1) takes (at most) on the order of seconds of computer time, and the regions $\bar{C}_n^{x_{st}}$ appear 'smooth' to the 'naked eye'.

We represent such region by two N-element vectors α^{min} and α^{max} corresponding to the upper and lower bounds of the region. That is, the i-th elements α_i^{min} and α_i^{max} of these vectors are, respectively, the smallest and largest step size x_{st}, such that the extended state (v_i, x_{st}) is in A. If no extended state is in A for a velocity v_i, we set the elements α_i^{min} and α_i^{max} to NaN ('Not-a-Number' value in Matlab).

The union C of two x_{st}-convex sets A and B can be found using the *min* and *max* functions:

$$\gamma^{min} = \min\{\alpha^{min}, \beta^{min}\}, \qquad \gamma^{max} = \max\{\alpha^{max}, \beta^{max}\},$$

where the vectors $\alpha^{min}, \alpha^{max}$ represent the region A, the vectors β^{min}, β^{max} the region B, and $\gamma^{min}, \gamma^{max}$ the union $C = A \cup B$. Both the minimum and maximum are taken element-wise. Note, that we assume here that the union C is also convex in the x_{st} direction. Similarly, we calculate the union $C = \bigcup_{k=1}^{K} A_k$ of several different sets A_k by

$$\gamma^{min} = \min\{\alpha^{1,min}, \ \dots, \alpha^{K,min}\}, \qquad \gamma^{max} = \max\{\alpha^{1,max}, \ \dots, \alpha^{K,max}\}. \tag{C.2}$$

We use the formula (C.2) above to numerically approximate the union in the right-hand-side of (C.1). The union is taken over all discretized target velocities v_i in the region $C_{n-1} \setminus C_{n-2}$. As before, we have to make the assumption that all extended 1-step controllable regions $\hat{\bar{C}}(v_i)$, as well as all regions $\bar{C}_n^{x_{st}}$, are convex in the x_{st} direction. This assumption is shown to hold true for the 1-step regions $\hat{\bar{C}}(v_i)$, as one can see from Figs. 3.2 and 3.3. We visually check the validity of the assumption for each of the regions $\bar{C}_n^{x_{st}}$ as we compute them.

After computation of $\bar{C}_n^{x_{st}}$ we also calculate the n-step controllable region C_n, which is necessary for the next iteration of (C.1) (i.e. computation of $\bar{C}_{n+1}^{x_{st}}$). Let vectors α^{min} and α^{max} be the discrete representation of the region $\bar{C}_n^{x_{st}}$. Then the

(discrete approximation of the) region C_n includes all velocities v_i, for which both elements α_i^{min} and α_i^{max} have numeric (not NaN) values.

C.2 Boundary of the extended ∞-step controllable region $\bar{C}_\infty^p$

Here we find an equation for and identify the motion corresponding to the upper boundary of the extended ∞-step controllable region $\bar{C}_\infty^p$ shown on Fig. 3.4c on page 82. According to representation (2.23) of the extended controllable regions, each $\bar{C}_n^p$ is a union of the extended 1-step controllable regions $\hat{\bar{C}}(q)$ for a range of different target states q. Therefore, the region $\bar{C}_\infty^p$ (the limit of $\bar{C}_n^p$ as $n \to \infty$) is such union too. Fig. C.1 shows the 1-step regions $\hat{\bar{C}}(\dot{\theta}_t)$ for the 2D IP model, corresponding to several different target velocities: $\dot{\theta}_t = 0,\ 0.3,\ 0.6,\ 0.8,$ and 0.9.[2] The figure also shows the boundary of the region $\bar{C}_\infty^p$.

As one can see from the figure, the top boundary of $\bar{C}_\infty^p$ is formed by the corner points (marked by the thick blue dots) of the regions $\hat{\bar{C}}(\dot{\theta}_t)$. These corner points correspond (e.g. see Fig. 3.2c) to the motion, such that the tension in the stance leg is zero both just before and just after the collision. Such motion is represented by equations (3.18c) and (3.18d) satisfied simultaneously. Thus, we get the system of equations:

$$\begin{cases} p = \cot 2\theta_{sw} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_0^2} \ - \frac{1}{\sin 2\theta_{sw}} \sqrt{2 - 2\cos\theta_{sw} + \dot{\theta}_t^2} \\ \cos\theta_{sw} = \frac{2+\dot{\theta}_0^2}{3} \\ \cos\theta_{sw} = \frac{2+\dot{\theta}_t^2}{3} \end{cases}$$

Excluding θ_{sw} and $\dot{\theta}_t$ from the above system, we obtain

$$p = \sqrt{\frac{3}{2 + \dot{\theta}_0^2} - \frac{2 + \dot{\theta}_0^2}{3}}. \tag{C.3}$$

[2] The cases $\dot{\theta}_t = 0,\ 0.3,$ and 0.8 correspond to the Figs. 3.3e, 3.2c, and 3.3f respectively.

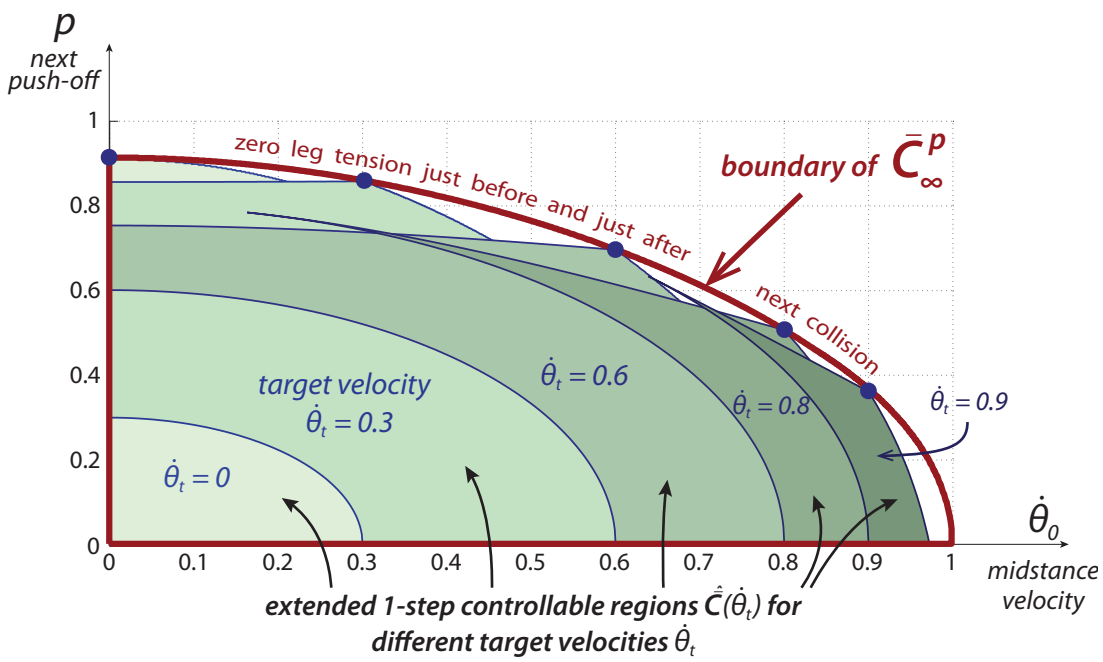


Figure C.1: Composition of the extended ∞-step controllable region $\bar{C}^p_\infty$. The region $\bar{C}^p_\infty$ is represented as a union of the extended 1-step controllable regions $\hat{\bar{C}}(\dot{\theta}_t)$ corresponding to different target velocities $\dot{\theta}_t$. Here the 1-step regions are shown for $\dot{\theta}_t = 0,\ 0.3,\ 0.6,\ 0.8$, and 0.9. The top boundary of $\bar{C}^p_\infty$ is formed by the corners of the regions $\hat{\bar{C}}(\dot{\theta}_t)$, which are marked by the thick blue dots. Each of these corners corresponds to the motion with zero tension in the stance leg both just before and just after the collision (see Fig. 3.2c).

Equation (C.3) above is the equation of the top boundary curve of the region $\bar{C}^p_\infty$.

C.3 Extended ∞-step controllable region with two control axes

We compute the extended ∞-step controllable region $\bar{C}^{p,x_{st}}_\infty$. The region $\bar{C}^{p,x_{st}}_\infty$ includes all combinations $(\dot{\theta}_0, x_{st}, p)$ of initial states $\dot{\theta}_0$ and two controls x_{st}, p for the next step, such that the robot can reach a given target velocity in a finite number of steps or approach the target asymptotically. We use the example target velocity (3.17).

According to the formula (2.23) on page 59, the extended n-step controllable region $\bar{C}_n^{p,x_{st}}$ is the union of 1-step regions $\hat{\bar{C}}(\dot{\theta})$ corresponding to all different targets $\dot{\theta}$ in the $(n-1)$-step controllable region C_{n-1}. At the same time, as n increases the regions $\bar{C}_n^{p,x_{st}}$ approach $\bar{C}_\infty^{p,x_{st}}$ and the regions C_n approach C_∞. Thus, $\bar{C}_\infty^{p,x_{st}}$ is the union of all 1-step regions $\hat{\bar{C}}(\dot{\theta})$ corresponding to different $\dot{\theta}$ in C_∞:

$$\bar{C}_\infty^{p,x_{st}} = \bigcup_{\dot{\theta}\in C_\infty} \hat{\bar{C}}(\dot{\theta}).$$

The ∞-step controllable region C_∞ is given by (3.23) on page 84. Hence, we get

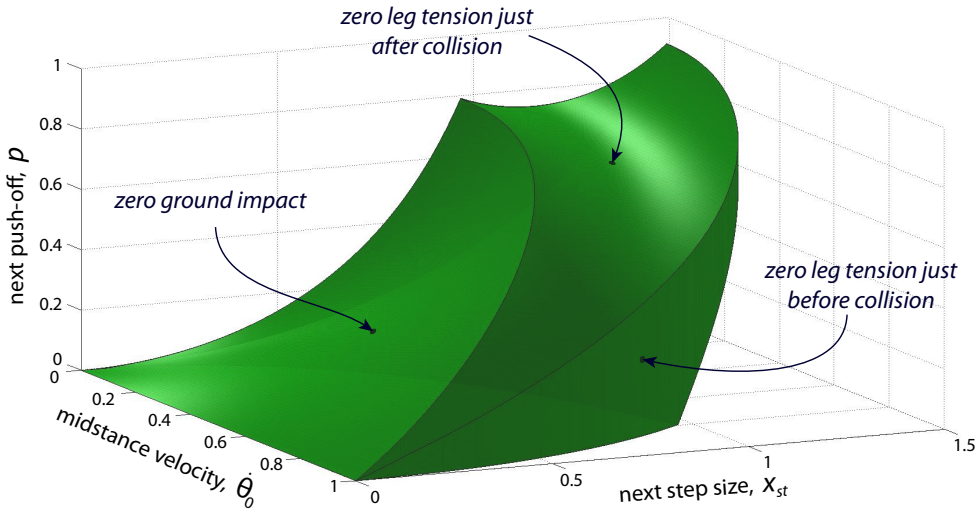
$$\bar{C}_\infty^{p,x_{st}} = \bigcup_{0<\dot{\theta}<1} \hat{\bar{C}}(\dot{\theta}). \tag{C.4}$$

We numerically compute the region $\bar{C}_\infty^{p,x_{st}}$, as in (C.4), in a way analogous to the calculation of the regions $\bar{C}_n^{x_{st}}$ in Appendix C.1. All calculations are performed in Matlab. First, we discretize the velocity axis in the interval $0 \le \dot{\theta}_0 \le 1$ ($N = 1001$ points v_i) and the step size axis in the interval $0 \le x_{st} \le 1.5$,[3] ($M = 1501$ points x_j). Next, for a fixed target velocity $\dot{\theta}_t$ and each discrete point (v_i, x_j) (i.e. initial velocity $\dot{\theta}_0 = v_i$ and step size $x_{st} = x_j$) we find the corresponding (unique) push-off p_{ij} from the Poincaré map (3.14). If $\dot{\theta}_t$, v_i, x_j, and p_{ij} do not satisfy at least one of the walking constraints (3.15), the robot fails and we assign $p_{ij} = NaN$.[4] Thus, for each push-off p_{ij} with a numeric value, the triple (v_i, x_j, p_{ij}) is a 1-step controllable extended state of the robot. Hence, the $N \times M$ matrix of push-offs $P = \{p_{ij}\}$ represents the (discretized) extended 1-step controllable region $\hat{\bar{C}}(\dot{\theta}_t)$ corresponding to the given target $\dot{\theta}_t$.

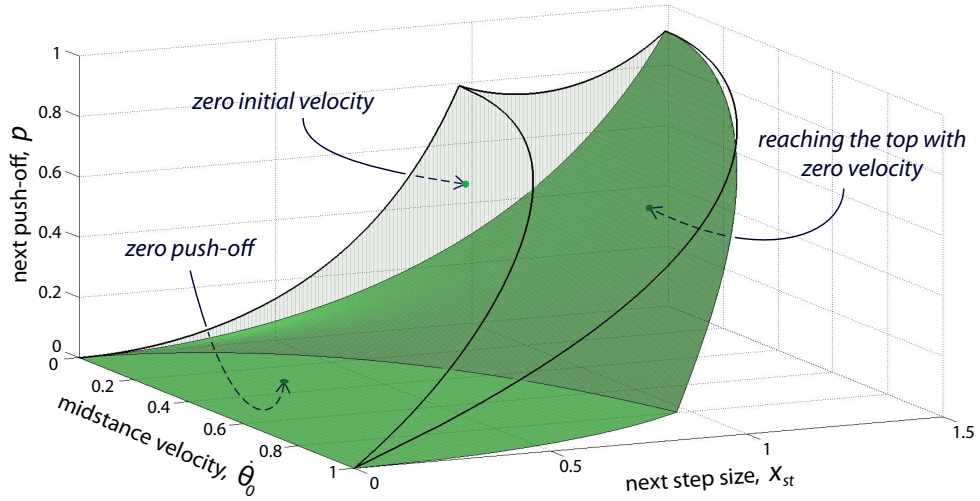
We calculate such push-off matrix P for each target velocity $\dot{\theta}_t = v_k$ in the interval $(0, 1)$ and call it P_k. These matrices P_k represent all different regions $\hat{\bar{C}}(\dot{\theta})$

[3] $x_{st} = 1.5$ is the rounded up largest ∞-step viable step size, based on the results in Section 3.2.4.

[4] 'Not-a-Number' value in Matlab.



(a) Front view of the region



(b) Back surfaces of the region

Figure C.2: Full extended ∞-step controllable region for the 2D IP model. A point in the axes on this figure is a triple $(\dot{\theta}_0, x_{st}, p)$, consisting of an initial state (velocity $\dot{\theta}_0$ at midstance) and two controls for the next step (step size x_{st} and push-off p). The extended ∞-step controllable region $\bar{C}_\infty^{p,x_{st}}$ is all $(\dot{\theta}_0, x_{st}, p)$ for which the robot is able to either reach a given target exactly in a finite number of steps or to approach it asymptotically. The region $\bar{C}_\infty^{p,x_{st}}$ is the same for any target velocity $\dot{\theta}_t \in (0, 1)$ and is equal to the extended viability kernel $\bar{V}_\infty^{p,x_{st}}$, all initial states and first-step controls for which the robot can stay up for an arbitrarily large number of steps. The boundary surfaces of $\bar{C}_\infty^{p,x_{st}}$ correspond to the walking constraints (3.15) of the planar IP model.

in the union (C.4). Similar to the formula (C.2) in Appendix C.1, we compute the union by taking the element-wise minimum and maximum of all matrices P_k:

$$P_{min} = \min\{P_1, \ \dots, P_{N-2}\}, \qquad P_{max} = \max\{P_1, \ \dots, P_{N-2}\}. \tag{C.5}$$

For each combination (v_i, x_j) of the initial velocity and next step size, the $N \times M$ matrices P_{min} and P_{max} give, correspondingly, the smallest and largest push-off in the region $\bar{C}_\infty^{p,x_{st}}$. Thus, P_{min} and P_{max} define the lower and upper boundary surfaces of $\bar{C}_\infty^{p,x_{st}}$ in the $(\dot{\theta}_0, x_{st}, p)$ axes — hence, defining the region $\bar{C}_\infty^{p,x_{st}}$ itself.

The extended ∞-step controllable region $\bar{C}_\infty^{p,x_{st}}$ for our 2D IP model is shown on Fig. C.2. The boundaries of $\bar{C}_\infty^{p,x_{st}}$ correspond to the motions, for which one of the walking constrains of the model is active (i.e. inequality turns into equality, the robot 'almost' fails). The three visible surfaces on Fig. C.2a represent the non-negative-ground-impact constraint (3.15a), the non-negative-leg-tension-after-collision constraint (3.15c), and the non-negative-leg-tension-before-collision constraint (3.15b). Notice, that the latter constraint (3.15b) is independent of the push-off p — therefore, the corresponding surface is a cylinder parallel to the p-axis. The three boundaries on the back side of the region (see Fig. C.2b), represent the non-negative-push-off constraint (3.15d), the assumption of a non-negative initial velocity (we only consider walking forward), and the requirement that the robot reaches the midstance at the end of each step. The latter 'reaching-the-top' surface is the extended 1-step controllable region $\hat{\bar{C}}(0)$ corresponding to the zero target velocity.

Any point inside $\bar{C}_\infty^{p,x_{st}}$ corresponds to a specific motion over the next step, which allows the robot to eventually reach the given target (3.17). However, recall from Section 3.2.3 that the ∞-step controllable region C_∞ is the same for any

target. Therefore, the formula (C.4), calculation (C.5), and, hence, the extended ∞-step controllable region $\bar{C}_{\infty}^{p,x_{st}}$ do not depend on the target velocity $\dot{\theta}_t$ (except for $\dot{\theta}_t = 0$ and $\dot{\theta}_t = 1$). In fact, $\bar{C}_{\infty}^{p,x_{st}}$ is also (almost) equal to the extended viability kernel $\bar{V}_{\infty}^{p,x_{st}}$. Indeed, according to (C.4), $\bar{C}_{\infty}^{p,x_{st}}$ includes all motions of the robot corresponding to all possible non-failed velocities at the next step — that is, it includes all non-failed steps of the robot (but for the cases $\dot{\theta}_0 = 0, 1$). Therefore, $\bar{C}_{\infty}^{p,x_{st}}$ is (almost) the same as the extended 1-step viable region $\bar{V}_{1}^{p,x_{st}}$:

$$\bar{C}_{\infty}^{p,x_{st}} \underset{\substack{\textit{except for some} \\ \textit{boundary points}}}{\approx} \bar{V}_{1}^{p,x_{st}}. \tag{C.6}$$

On the other hand, the region $\bar{C}_{\infty}^{p,x_{st}}$ is a subset of $\bar{V}_{\infty}^{p,x_{st}}$ (because the target allows periodic motion — see discussion in Section 3.2.4); and $\bar{V}_{\infty}^{p,x_{st}}$, in turn, is always a subset of $\bar{V}_{1}^{p,x_{st}}$:

$$\bar{C}_{\infty}^{p,x_{st}} \subset \bar{V}_{\infty}^{p,x_{st}} \subset \bar{V}_{1}^{p,x_{st}}. \tag{C.7}$$

From relations (C.6) and (C.7) we conclude that the regions $\bar{C}_{\infty}^{p,x_{st}}$ and $\bar{V}_{\infty}^{p,x_{st}}$ are the same (but for $\dot{\theta}_0 = 0, 1$):

$$\bar{C}_{\infty}^{p,x_{st}} \underset{\substack{\textit{except for some} \\ \textit{boundary points}}}{\approx} \bar{V}_{\infty}^{p,x_{st}}. \tag{C.8}$$

This results supports our 'Viable is Controllable' claim in Section 2.3.1 on page 25: in most cases, the ability to not fail is equivalent to the ability to reach any given target.

The region $\bar{C}_{\infty}^{p,x_{st}}$ is the 'full' extended ∞-step controllable region for our 2D IP model, in that it considers all control parameters of the model. The projections of $\bar{C}_{\infty}^{p,x_{st}}$ onto the $(\dot{\theta}_0, x_{st})$ and $(\dot{\theta}_0, p)$ coordinate planes are, correspondingly, the extended regions $\bar{C}_{\infty}^{x_{st}}$ and $\bar{C}_{\infty}^{p}$ shown on Figs. 3.4b and 3.4c on page 82. Similarly, the projections of the reaching-the-top boundary surface of $\bar{C}_{\infty}^{p,x_{st}}$ (the extended

1-step controllable region $\hat{\bar{C}}(0)$) are the regions $\bar{C}_1^{x_{st}}$ and $\bar{C}_1^{p}$ presented on Figs. 3.3c and 3.3e on page 80.

C.4 Controllability for large target speeds

For the 2D IP model, the 'Two-step controllability' claim on page 27 breaks in the cases of large target velocities (roughly $\dot{\theta}_t > 0.8$) and significant push-off limitation ('small' maximum allowed push-off). In such cases the robot may have to take three or more steps to return to the target.

Fig. C.3 shows the controllable and extended controllable regions for the target $\dot{\theta}_t = 0.95$. For small initial velocities, e.g. $\dot{\theta}_0 = 0.1$, the no-flying requirement prevents the robot from being able to reach the target in fewer than three steps.

The case of restrictive push-off limits is presented in graphs (a), (c), (e) of Fig. C.4. The figure assumes the target $\dot{\theta}_t = 0.5$, the minimum allowed step-time $t_{st,min} = 0.5$, and the maximum allowed push-off $p_{max} = 0.25$.[5] The push-off limit restricts the amount of energy that can be added into the system in one step, thus requiring the robot to take several steps to recover from small velocities. A larger bound p_{max} on the maximum push-off would make it possible to gain speed faster, hence rendering the two-step regions closer to the ∞-step regions. For example, graphs (b), (d), (f) of Fig. C.4 assume $p_{max} = 0.5$ — in this case $C_2 \approx C_\infty$, $\bar{C}_2^{x_{st}} = \bar{C}_\infty^{x_{st}}$, and $\bar{C}_2^{p} = \bar{C}_\infty^{p}$, i.e. the two-step controllability is (almost) equivalent to the ∞-step controllability.

[5] For comparison, the Cornell Ranger robot [9] has an estimated $p_{max} = 0.3$ and walks at an average speed $\dot{\theta} = 0.19$ (see Chapter 6.1).

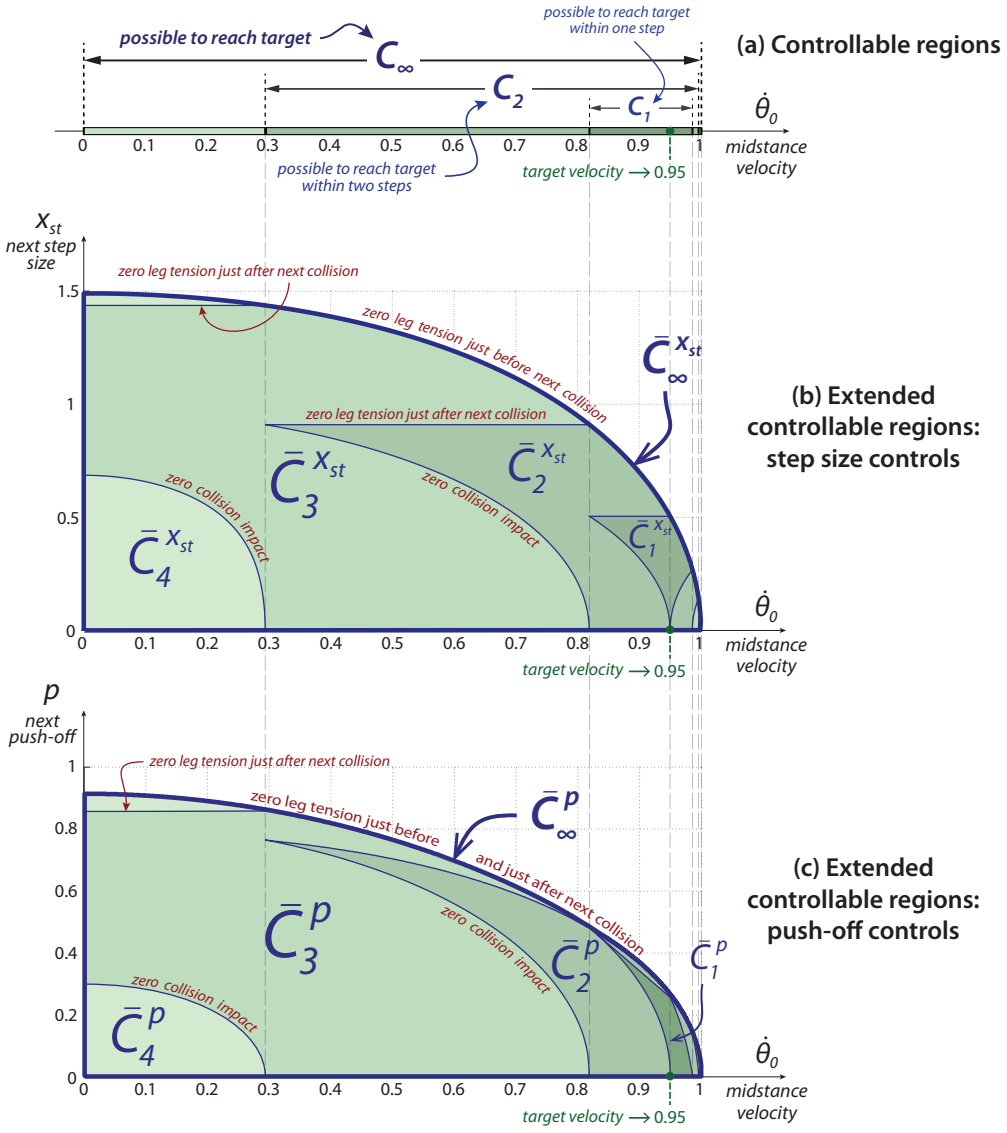
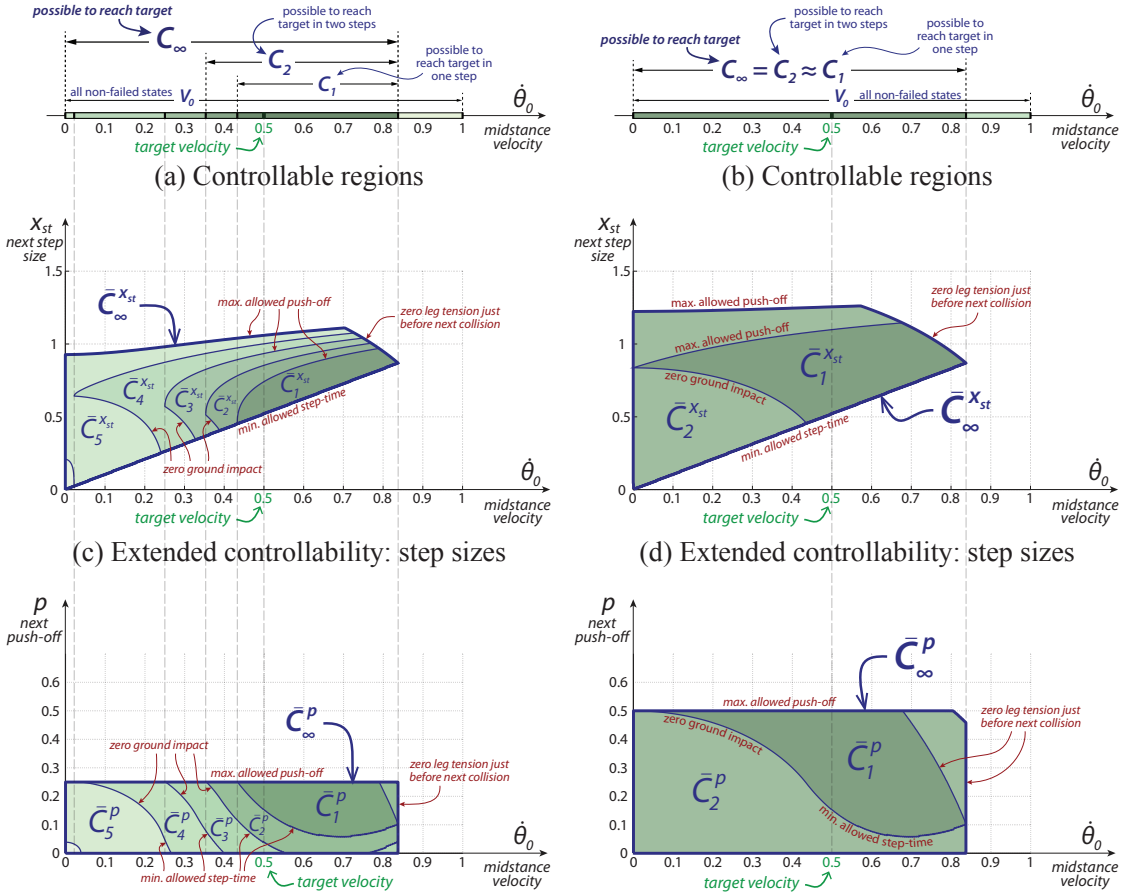


Figure C.3: Controllability of 2D IP model for large target speed. This figure is analogous to Fig. 3.4 on page 82, but for a different target velocity $\dot{\theta}_t$. Here $\dot{\theta}_t = 0.95$. For target velocities close to 1 (the largest viable speed), as opposed to smaller $\dot{\theta}_t$, the two-step controllable regions C_2, $\bar{C}_2^{x_{st}}$, and $\bar{C}_2^{p}$ do not cover most of the corresponding ∞-step regions C_∞, $\bar{C}_\infty^{x_{st}}$, and $\bar{C}_\infty^{p}$.

max. allowed push-off 0.25 *max. allowed push-off 0.5*



(a) Controllable regions (b) Controllable regions

(c) Extended controllability: step sizes (d) Extended controllability: step sizes

(e) Extended controllability: push-offs (f) Extended controllability: push-offs

Figure C.4: Controllability of the 2D IP model for different push-off limitations. Analogous to Fig. 3.6 on page 90, this figure shows the controllable and extended controllable regions of the 2D IP model with limited actuation. The actuator limitations are the maximum allowed push-off p_{max} and the minimum allowed step-time $t_{st,min}$. The graphs (a), (c), (e) on the left assume $p_{max} = 0.25$, while the graphs (b), (d), (f) on the right $p_{max} = 0.5$. For all graphs $t_{st,min} = 0.5$ and the target velocity is $\dot{\theta}_t = 0.5$. Larger push-offs improve controllability of the model and make the ∞-step controllability (regions C_∞, $\bar{C}_\infty^{x_{st}}$, $\bar{C}_\infty^{p}$) equivalent to the two-step controllability (regions C_2, $\bar{C}_2^{x_{st}}$, $\bar{C}_2^{p}$).

APPENDIX D

SUPPLEMENTARY CALCULATIONS FOR CORNELL RANGER

D.1 Leg swinging limitations

Here we find a rough estimate of the minimum step time $t_{st,min}$ for the Ranger robot. The minimum allowed step time (time t_{st} from midstance to heel-strike) is one of the constraints we use in the IP-model proxy for Ranger in Section 6.1.2 on page 137.

There are two limitations of Ranger which constrain how fast a step can be made. One is the limited torque the hip motor can generate to rotate the swing leg. We use the high-fidelity model of Ranger (including the model of the electric hip motor) [9] to estimate how the hip motor limitations affect $t_{st,min}$. We do a series of simulations as follows. We hold the stance leg fixed and allow the swing leg to rotate around the hip joint, without colliding with the ground. The swing leg is initially placed at an angle ϕ_0 with the vertical and has initial angular rate $\dot{\phi}_0$; both ϕ_0 and $\dot{\phi}_0$ vary between simulations. Each simulation is run[1] until the instant when the leg reaches the angle $-\phi_0$ — we use such simulation as a proxy for the swing leg's motion during the stance phase of the robot. The length of the corresponding step of the robot, according to (3.3), is $x_{st} = -2 \sin \phi_0$. In each simulation, we supply the maximum current allowed to the hip motor, changing its sign at 'midstance', when $\phi = 0$. The step time t_{st} is recorded as one half of the total duration of the simulation.

Thus, for each initial rate $\dot{\phi}_0$, we compute the step time t_{st} as a func-

[1] All calculations are performed in Matlab using the *ode45* solver.

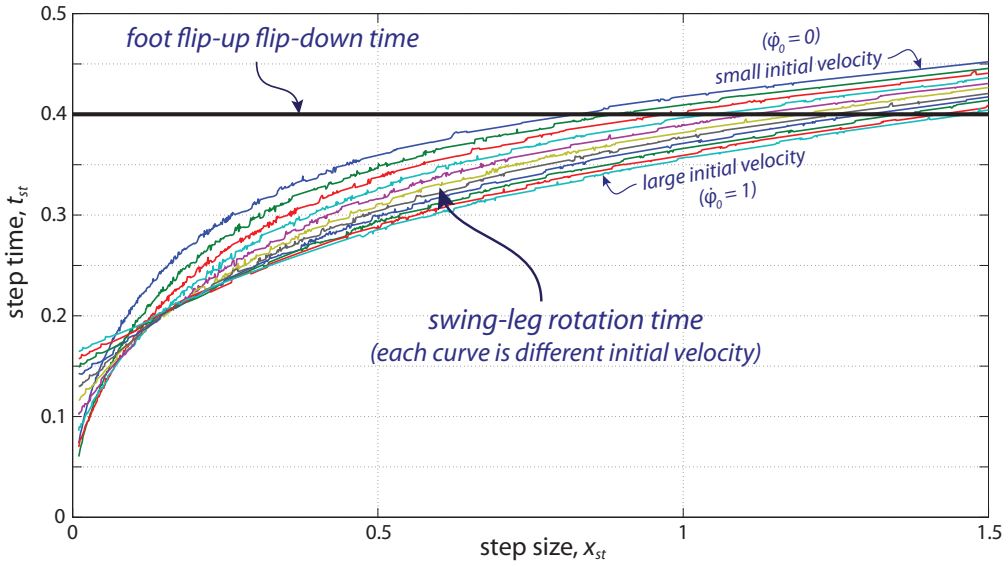


Figure D.1: Step-time restrictions of Ranger. The step time is defined as the time from midstance to the next heel-strike. The family of curves shows the minimum time required for the hip motor to move the swing leg to a desired position defined by the step-size x_{st}. Each curve corresponds to a different initial angular rate of the leg. The thick horizontal line shows the minimum time required for the swing foot to flip up and flip down (necessary to avoid scuffing). Based on this figure, the minimum allowed step-time $t_{st,min}$ is set to 0.4.

tion of the step size x_{st}. This function is shown by an individual curve on Fig. D.1, with different curves corresponding to different initial rates in the range $0 \leq \dot{\phi}_0 \leq 1$. The considered range of step-sizes is $0 < x_{st} \leq 1.5$. All values are non-dimensionalized. The figure shows that the biggest step time $t_{st} \approx 0.45$ occurs for a small initial velocity and large step-size. At the same time, there is no significant dependence of the step time on the initial velocity. That is, the effects of the power provided by the hip motor seem to be larger than those by the gravity.

The second step-time limitation of Ranger is related to scuffing avoidance. Each step, Ranger has to flip up and flip down the foot on the swing leg in order to avoid the scuffing the ground while in swing. By running physical experiments on the robot, we find that at least 0.25 sec is required to complete the

flip-up flip-down motion. This corresponds to about 0.8 in our non-dimensional units, and, hence, to the step time $t_{st} \approx 0.4$. This step time is shown by the thick horizontal line on Fig. D.1.

We can see from the figure that in most cases (except for large steps) the foot flipping time dominates the time required to swing the leg. On the other hand, the step-size x_{st} of the robot is bounded from above by (6.1b). Therefore, we choose the flip-up flip-down time as the fixed minimum step-time, for all initial velocities and all step sizes:

$$t_{st,min} = 0.4. \tag{D.1}$$

D.2 Push-off limitations

We find a rough estimate p_{max} of the maximum push-off allowed in the IP-model proxy of Ranger (see equation (6.1a) on page 140).

We use the high-fidelity model of Ranger and run its simulation along the nominal trajectory used during the 65.2 km walk.[2] For this trajectory, the pushing-off is done during a 0.1-second-long interval shortly before and during the double stance motion [9]. In our simulation, we replace the nominal current to the ankle motor during the push-off by the (constant) maximum current allowed to the motor. For the resulting motion, we record the increase in the total mechanical energy during the push-off:

$$\Delta E \approx 0.04 \tag{D.2}$$

in a non-dimensional form.

[2] The Matlab simulation was developed and provided by S. Javad Hasaneini.

We say that the maximum push-off p_{max} in the IP model of Ranger is one which produces the energy input ΔE. The hip velocities $\vec{v}^-$ and $\vec{v}_p$ just before and just after the push-off in the IP model are given by (3.7b) and (3.5a) correspondingly. Because there is no change in the potential energy, the total mechanical energy jump due to the push-off is

$$\frac{1}{2}v_p^2 - \frac{1}{2}(v^-)^2 = \frac{1}{2}p^2. \tag{D.3}$$

Therefore, for the maximum push-off p_{max} we have

$$\begin{aligned} \frac{1}{2}p_{max}^2 = \Delta E \qquad &\Rightarrow \\ p_{max} = \sqrt{2\Delta E} \approx 0.3. \end{aligned} \tag{D.4}$$

Note, that (D.2) is a rough estimate of Ranger's hardest push-off. The true maximum value of ΔE (hence, of p_{max} too) may be higher — for example, if a different (non-constant) function of the ankle-motor current or a different push-off timing is used. However, our results in Section 6.1.3 show that the push-off bound (D.4) above does not affect controllability of the IP model of Ranger (see Fig. 6.2). Furthermore, the controller we design for Ranger in Section 6.2 only requires push-offs $p < 0.1$ (see equation (6.4b)). Hence, a (possible) more liberal push-off constraint would not alter the results of our controller design.

D.3 Energy efficiency of the step-size controller

Here we estimate energy efficiency of our IP model of Ranger for different initial velocities and step sizes. The estimations justify our approach to efficiency of the step-size controller in Section 6.2.4 on page 150.

We define efficiency by the total cost of transport of the robot, TCOT, as in equation (6.9). We closely follow calculations for Ranger in [9] to approximate the TCOT for our model. For Ranger, the cost of transport is divided into three different terms:

$$\text{TCOT} = \frac{E_{\text{fixed}} + E_{\text{foot-flip}} + E_{\text{walk}}}{Mg \cdot d}, \tag{D.5}$$

where $M = 9.91$ kg is the total mass of the robot, g acceleration due to gravity, and d the total walking distance. E_{fixed} is the *electrical overheads* cost — this is the energy required to power all electronics in Ranger, such as microprocessors and sensors. The power running the electronics is assumed constant, and, hence, E_{fixed} is proportional to time: $E_{\text{fixed}} = P_{\text{fixed}}\, t$, with $P_{\text{fixed}} = 5.15$ W. $E_{\text{foot-flip}}$ is the work done to flip the swing foot up and down between collisions in order to avoid scuffing. The foot-flip work over a single step is independent of both the step length and step duration, and is estimated to be 1.06 J for Ranger. The last term, E_{walk}, is the work done by all motors on the robot to power the walk, not including the foot-flip motion. For our simple model of Ranger, we calculate E_{walk} for a single step as the mechanical energy added to the system by the push-off. This energy is given by expression (D.3) in the non-dimensional form. Hence, $E_{\text{walk}} = \frac{1}{2}p^2 \cdot Mgl$ in the energy units, where $l = 0.96$ m is the leg length. Plugging E_{fixed}, $E_{\text{foot-flip}}$, and E_{walk} back into equation (D.5), we write the TCOT of our model over one step as

$$\begin{aligned} c_1 &= \frac{1.06 + 5.15 \cdot \sqrt{\frac{l}{g}}T_{st} + 0.5Mglp^2}{Mglx_{st}} \\ &\approx \frac{0.011 + 0.017\, T_{st} + 0.5p^2}{x_{st}} \end{aligned} \tag{D.6}$$

where $\sqrt{\frac{l}{g}}T_{st}$ is the duration of the step (midstance-to-midstance time) in seconds, and lx_{st} is the step-size in meters. For a given initial velocity $\dot{\theta}_0$ and controls x_{st}, p, the non-dimensional step duration T_{st} is found by integrating the

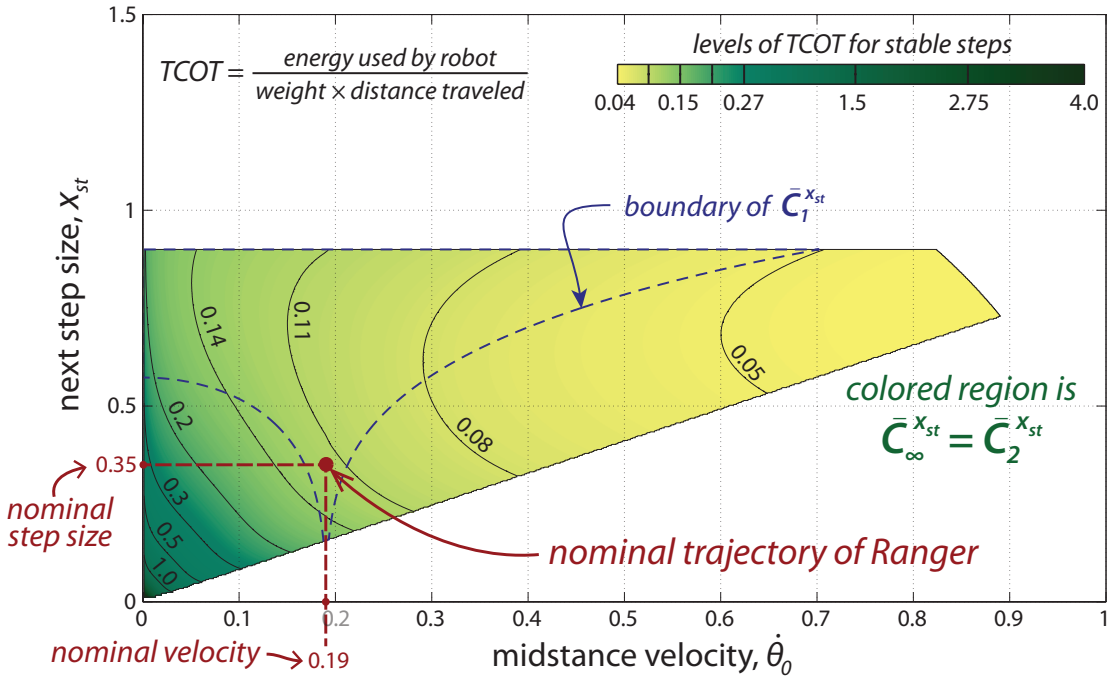


Figure D.2: Energy efficiency of step-size controllers for Ranger. For each perturbed velocity $\dot{\theta}_0$ and next step-size x_{st} the energy cost of the next step is considered. We only consider the steps that bring the robot closer to the nominal velocity $\dot{\theta}^*$. The Total Cost Of Transport (TCOT) is the total energy used by the robot divided by the robot's weight and traveled distance. For a given $\dot{\theta}_0$ and x_{st} the energy cost c_1^{min} is the smallest TCOT over all push-offs. The color code and contour lines show the values of c_1^{min} for all points in the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$. The energy cost grows 'rapidly' only for small steps with low velocities.

equations of motion (3.1) of the model.

We compute the cost c_1 for various perturbed steps of our model. We consider all velocities $\dot{\theta}_0$ and step sizes x_{st} inside the extended ∞-step controllable region $\bar{C}_\infty^{x_{st}}$ (see Fig. 6.2b). For each point $(\dot{\theta}_0, x_{st})$ there is a range of viable push-offs p, each bringing the robot to a different velocity $\dot{\theta}_1$ at the next midstance. However, because the step-size controller tries to stabilize the motion around the nominal velocity $\dot{\theta}^*$ (see Section 6.2.2), we only consider those push-offs, which bring the robot closer to the nominal velocity. That is, we only consider values of p, such that $|\dot{\theta}_1 - \dot{\theta}^*| \leq |\dot{\theta}_0 - \dot{\theta}^*|$. For each point $(\dot{\theta}_0, x_{st})$ we find the smallest cost c_1 over the considered range of p and call it c_1^{min}:

The values of c_1^{min} for all points in the region $\bar{C}_\infty^{x_{st}}$ are shown by the color scheme and level lines in Fig. D.2. For initial velocities smaller than the nominal $\dot{\theta}^*$, positive energy has to be supplied to the system — hence, the cost c_1^{min} is higher than $c_1^* \approx 0.12$ corresponding to the nominal trajectory. Similarly, the cost is lower for velocities larger than $\dot{\theta}^*$. From the point of view of energy efficiency, it suffices for the step-size controller to avoid taking short steps with small velocities. Such steps correspond to the bottom-left corner of $\bar{C}_\infty^{x_{st}}$ where the cost c_1^{min} is 'significantly' higher than the nominal c_1^*. However, recall from Section 6.2.2 that such steps are undesirable due to slow return to the nominal trajectory: for any perturbation we prefer to take possibly larger steps as they allow faster return. Thus, we do not take energy efficiency into account in the design of a step-size controller for Ranger, except for using the energy-efficient nominal trajectory (6.2).

APPENDIX E

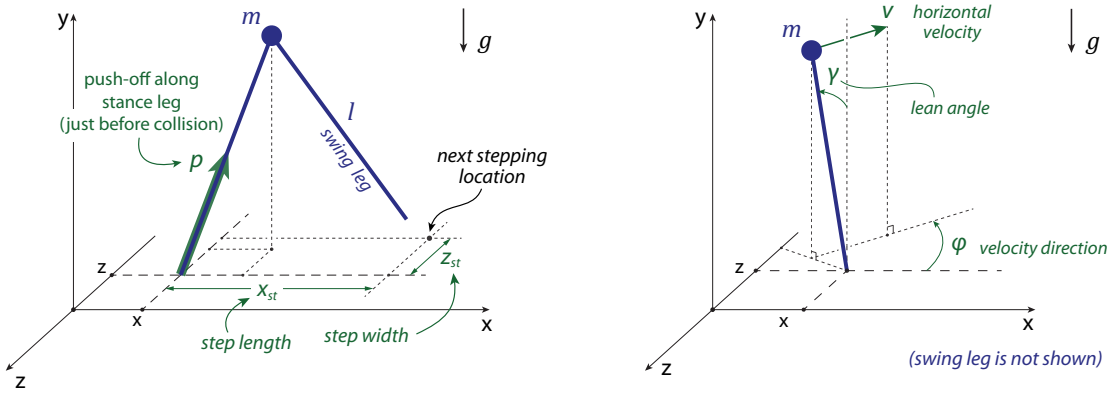
INVERTED PENDULUM IN 3D

We studied the planar Inverted Pendulum (IP) model and its controllability in Chapter 3. Here we begin to extend this model to 3D. We describe some possible locomotion goals for the 3D model, and discuss, by counting arguments, the model's possible controllability for these goals.

E.1 Model

The IP model has a point mass at the hip and two massless rigid legs (see Fig. E.1a). We assume that the swing leg can be instantaneously moved into any desired position without influencing the dynamics of the stance leg. As per the 2D model, restrictions on swing time can be added later. The collisions are assumed instantaneous, with the hind leg leaving the ground immediately after the collision (no double stance). At the instant just before each collision, a controlled push-off impulse is applied along the stance leg. There are three control parameters at each step: the push-off magnitude p and the stepping location defined by two numbers (e.g. positions x_{st} and z_{st} as in Fig. E.1a, or local coordinates). The dynamic state of the stance leg is four-dimensional: two angles defining orientation of the leg and two corresponding angle rates.

Midstance is the point along the trajectory of the robot where the potential energy has a local maximum or, equivalently, where the velocity of the hip is either horizontal or zero. At midstance there is one condition on the robot state, so the midstance state is three-dimensional. We define the three midstance vari-



(a) Model, controls **(b)** State at midstance

Figure E.1: 3D Inverted Pendulum (IP) model of walking. **(a)** The 3D IP model is an extension of the planar IP model shown in Fig. 2.5a on page 29. The model has a point mass at the hip and two massless rigid legs. Motions of the swing leg do not affect the dynamics of the stance leg. Collisions are assumed instantaneous; a push-off impulse is applied along the stance leg just before each collision. The controls are the push-off amount p and the stepping location defined by the positions x_{st} and z_{st} of the footstep. x, y, z is a fixed global coordinate system. **(b)** The midstance of the 3D IP model is at a point of local maximum of the hip height above the ground. The hip velocity is horizontal at midstance. The dynamic state of the model at midstance is defined by three values: the lean angle γ between the stance leg and the vertical, the velocity magnitude v, and the velocity direction (heading) φ. A local coordinate system, useful for considering balance without navigation, can be defined by the origin at the stance foot and the directions of y and v. In this local system, we can name the footstep positions x_{st} and z_{st} the step length and step width, respectively.

ables to be the lean angle γ between the stance leg and the vertical, the hip-velocity magnitude v, and the hip-velocity heading φ. An example of midstance is shown in Fig. E.1b.

Thus, the 3D IP model has three state variables at midstance and three control parameters at each step. Our central goal is that the robot never enters the flight phase and never falls down — such cases are failures. At this point, we do not yet consider actuation constraints of the model, i.e. we do not yet impose upper bounds on push-off nor minimum possible swing times.

E.2 Example goals

We consider a few possible example goals for locomotion for the 3D IP model. For each goal, we use the counting argument (number of equations equals the number of unknowns, see Section 5.4, page 131) to guess at the controllability of the model for that goal.

(i) **Standing upright with zero velocity**, as for Pratt's capture regions [64]. This goal is defined by two target values at midstance (number of restrictions on the goal state is $m_{\text{goal}} = 2$): lean angle $\gamma = 0$ and velocity $v = 0$. The number of control parameters is three ($m_{\text{cont}} = 3$). Hence, we have $m_{\text{cont}} > m_{\text{goal}}$. By the counting argument, there should be a non-trivial region in the state space (this is the one-step controllable region C_1) from where the robot can reach the target within one step. Because $m_{\text{cont}} = m_{\text{goal}} + 1$, there should be a one-parameter family of control actions that bring the robot to the goal.

(ii) **An arbitrary state at midstance.** This yields three target values (the lean γ, speed v, and heading φ) and we get $m_{\text{cont}} = m_{\text{goal}} = 3$. Therefore, the counting argument suggests that there is a non-trivial one-step controllable region C_1, from where the goal can be reached within one step with appropriate control actions. Generically, the control actions should be unique for each initial state in C_1 (no redundant controls).

(iii) **An arbitrary state at midstance *and* the stance foot is at a given point on the ground.** This gives us five target values: three values (γ, v, φ) describe the dynamic state at midstance and two (x, z) define the desired foot location. Hence, $m_{\text{cont}} = 3 < m_{\text{goal}} = 5$, but $2 * m_{\text{cont}} = 6 > m_{\text{goal}} = 5$. That is, at

least two steps are generally required to reach such a goal. We also expect that there is a non-trivial two-step controllable region, if the target foot location is 'sufficiently close' to the initial foot position. For each point in the controllable region, we expect a one-dimensional family of control actions that bring bring the robot to the target.

We perform both analytical and numerical calculations (not presented here) to find the one-step controllable region C_1 of the 3D IP model for various targets. The calculations are similar to those we use for the planar IP model in Chapter 3. With no loss in generality, we assume the initial heading to be zero, i.e. that the initial velocity is in the positive x-direction. For a given target, C_1 is the region in the (γ_0, v_0)-plane which includes all initial lean angles γ_0 and velocities v_0 such that the target can, with appropriate controls, be reached within one step. We consider different target states at midstance that have a zero lean $\gamma_t = 0$ (i.e. vertical position), velocity in the range $0 \leq v_t \leq 0.3$, and heading $0 \leq \varphi_t \leq 1.2$.[1] For each such target, our calculations show that the corresponding region C_1 is non-trivial (has a non-zero area). Thus, we confirm the conclusions in the examples (i) and (ii) above based on the counting argument.

Future important analysis of the 3D IP model follows the steps of our investigations of the planar IP model (Chapters 3 and 6): computation of the n-step controllable and extended controllable regions; analysis of the model with limited actuation; designing a maximally robust walking controller for a 3D robot based on the extended controllability of the 3D IP model.

[1] All numbers are non-dimensionalized using the constants m, l, and g for the hip mass, fixed leg length, and acceleration due to gravity, respectively.

BIBLIOGRAPHY

[1] RM Alexander. Optimization and gaits in the locomotion of vertebrates. *Physiol. Rev*, 69(1199-1227):29–64, 1989.

[2] Jean-Pierre Aubin. A survey of viability theory. *SIAM Journal on Control and Optimization*, 28(4):749–788, 1990.

[3] Jean Pierre Aubin, Alexandre M Bayen, and Patrick Saint-Pierre. *Viability theory: new directions*. Springer, 2011.

[4] Jean Pierre Aubin and Arrigo Cellina. *Differential inclusions: set-valued maps and viability theory*. Springer-Verlag New York, Inc., 1984.

[5] J.P. Aubin. Viability theory. systems & control: Foundations & applications, 1991.

[6] Christine Azevedo, Philippe Poignet, and Bernard Espiau. Moving horizon control for biped robots without reference trajectory. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2762–2767. IEEE, 2002.

[7] Alexandre M Bayen, Eva Crück, and Claire J Tomlin. Guaranteed overapproximations of unsafe sets for continuous and hybrid systems: solving the hamilton-jacobi equation using viability techniques. In *Hybrid Systems: Computation and Control*, pages 90–104. Springer, 2002.

[8] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[9] P.A. Bhounsule. *A controller design framework for bipedal robots*. PhD thesis, Cornell University, 2012.

[10] Pranav A Bhounsule. Control of a compass gait walker based on energy regulation using ankle push-off and foot placement. *Robotica*, 2014.

[11] Cornell Biorobotics and Locomotion Lab website. `http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/Ranger2011`.

[12] Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11):1217–1227, 1989.

[13] AC Bobbert. Energy expenditure in level and grade walking. *Journal of Applied Physiology*, 15(6):1015–1021, 1960.

[14] Thomas Buschmann, Sebastian Lohmeier, Mathias Bachmayer, Heinz Ulbrich, and Friedrich Pfeiffer. A collocation method for real-time walking pattern generation. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 1–6. IEEE, 2007.

[15] Sean G Carver, Noah J Cowan, and John M Guckenheimer. Lateral stability of the spring-mass hopper suggests a two-step control strategy for running. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(2):026106, 2009.

[16] A.I. Chaudhry. *Topics in robot locomotion: state estimation and feasible path generation for wheeled and legged robots*. PhD thesis, Cornell University, 2015.

[17] Christine Chevallereau, ABBA Gabriel, Yannick Aoustin, Franck Plestan, Eric Westervelt, Carlos Canudas De Wit, Jessy Grizzle, et al. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, 2003.

[18] Christine Chevallereau, Jessy W Grizzle, and Ching-Long Shih. Asymptotically stable walking of a five-link underactuated 3-d bipedal robot. *Robotics, IEEE Transactions on*, 25(1):37–50, 2009.

[19] Michael J Coleman and Andy Ruina. An uncontrolled walking toy that cannot stand still. *Physical Review Letters*, 80(16):3658, 1998.

[20] S.H. Collins, M. Wisse, and A. Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.

[21] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[22] Steven H Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1983–1988. IEEE, 2005.

[23] Guillaume Deffuant, Laetitia Chapel, and Sophie Martin. Approximating

viability kernels with support vector machines. *Automatic Control, IEEE Transactions on*, 52(5):933–937, 2007.

[24] J Maxwell Donelan, Rodger Kram, et al. Mechanical and metabolic determinants of the preferred step width in human walking. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1480):1985–1992, 2001.

[25] A. Forner Cordero, H. Koopman, and FCT Van der Helm. Multiple-step strategies to recover from stumbling perturbations. *Gait & posture*, 18(1):47–59, 2003.

[26] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practicea survey. *Automatica*, 25(3):335–348, 1989.

[27] Mariano Garcia, Anindya Chatterjee, Andy Ruina, and Michael Coleman. The simplest walking model: stability, complexity, and scaling. *Journal of biomechanical engineering*, 120(2):281–288, 1998.

[28] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Spring-mass running: simple approximate solution and application to gait stability. *Journal of theoretical biology*, 232(3):315–328, 2005.

[29] A.S. Grewal. *Model reduction and controller design simplification for bipedal robots*. PhD thesis, Cornell University, 2015.

[30] Jessy W Grizzle, Gabriel Abba, and Franck Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *Automatic Control, IEEE Transactions on*, 46(1):51–64, 2001.

[31] JW Grizzle. Remarks on event-based stabilization of periodic orbits in systems with impulse effects. In *Second International Symposium on Communications, Control and Signal Processing*. Citeseer, 2006.

[32] Seyed Javad Hasaneini. *Energy Efficient Bipedal Locomotion*. PhD thesis, 2014.

[33] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1321–1326. IEEE, 1998.

[34] Daan GE Hobbelen and Martijn Wisse. Swing-leg retraction for limit cycle walkers improves disturbance rejection. *Robotics, IEEE Transactions on*, 24(2):377–389, 2008.

[35] A. L. Hof, S. M. Vermerris, and W. A. Gjaltema. Balance responses to lateral perturbations in human treadmill walking. *The Journal of Experimental Biology*, 213(15):2655–2664, 2010.

[36] A.L. Hof. The 'extrapolated center of mass' concept suggests a simple control of balance in walking. *Human movement science*, 27(1):112–125, 2008.

[37] M.A. Hollands, D.E. Marple-Horvat, S. Henkes, and A.K. Rowan. Human eye movements during visually guided stepping. *Journal of motor behavior*, 27(2):155–163, 1995.

[38] Dynamic Walking Conference 2010 J.Pratt. `http://techtv.mit.edu/collections/locomotion:1216/videos/8036-dynamic-walking-2010-jerry-pratt-humanoid-disturbance-recovery-with-limited-available-footholds`.

[39] Thomas Kailath. *Linear systems*, volume 1. Prentice-Hall Englewood Cliffs, NJ, 1980.

[40] S. Kajita, O. Matsumoto, and M. Saigo. Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2299–2306. IEEE, 2001.

[41] Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1405–1411. IEEE, 1991.

[42] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113, 2012.

[43] A.D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of biomechanical engineering*, 124:113, 2002.

[44] Arthur D Kuo. The six determinants of gait and the inverted pendu-

lum analogy: A dynamic walking perspective. *Human movement science*, 26(4):617–656, 2007.

[45] Sergey Levine and Vladlen Koltun. Guided policy search. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1–9, 2013.

[46] Hun-ok Lim, Yoshiharu Kaneshima, and Atsuo Takanishi. Online walking pattern generation for biped humanoid robot with trunk. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 3111–3116. IEEE, 2002.

[47] Ian R Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, page 0278364910395339, 2011.

[48] Thijs Mandersloot, Martijn Wisse, and Christopher G Atkeson. Controlling velocity in bipedal walking: A dynamic programming approach. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 124–130. IEEE, 2006.

[49] D.S. Marigold and A.E. Patla. Strategies for dynamic stability during locomotion on a slippery surface: effects of prior experience and knowledge. *Journal of Neurophysiology*, 88(1):339–353, 2002.

[50] Jonathan S Matthis and Brett R Fajen. Visual control of foot placement when walking over complex terrain. 2013.

[51] Tad McGeer. Passive dynamic walking. *the international journal of robotics research*, 9(2):62–82, 1990.

[52] Tad McGeer and Leigh Hunt Palmer. Wobbling, toppling, and forces of contact. *American Journal of Physics*, 57(12):1089–1098, 1989.

[53] AE Minetti, LP Ardigo, and F Saibene. The transition between walking and running in humans: metabolic and mechanical aspects at different gradients. *Acta physiologica Scandinavica*, 150(3):315–323, 1994.

[54] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):43, 2012.

[55] L.M. Nashner. Balance adjustments of humans perturbed while walking. *Journal of Neurophysiology*, 44(4):650–664, 1980.

[56] Koichi Nishiwaki and Satoshi Kagami. Strategies for adjusting the zmp reference trajectory for maintaining balance in humanoid walking. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4230–4236. IEEE, 2010.

[57] Koichi Nishiwaki, Satoshi Kagami, Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2684–2689. IEEE, 2002.

[58] Federico Parietti and Hartmut Geyer. Reactive balance control in walking based on a bipedal linear inverted pendulum model. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5442–5447. IEEE, 2011.

[59] A.E. Patla and J.N. Vickers. Where and when do we look as we approach and step over an obstacle in the travel path? *Neuroreport*, 8(17):3661–3665, 1997.

[60] Aftab E Patla. Understanding the roles of vision in the control of human locomotion. *Gait & Posture*, 5(1):54–69, 1997.

[61] Aftab E Patla and Joan N Vickers. How far ahead do we look when required to step on specific locations in the travel path during locomotion? *Experimental brain research*, 148(1):133–138, 2003.

[62] M. Pijnappels, M.F. Bobbert, and J.H. van Dieën. How early reactions in the support limb contribute to balance recovery after tripping. *Journal of biomechanics*, 38(3):627–634, 2005.

[63] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *The International Journal of Robotics Research*, 31(10):1117–1133, 2012.

[64] J.E. Pratt and R. Tedrake. Velocity-based stability margins for fast bipedal walking. *Fast Motions in Biomechanics and Robotics*, pages 299–324, 2006.

[65] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207. IEEE, 2006.

[66] Jerry Pratt and Gill Pratt. Exploiting natural dynamics in the control of a 3d bipedal walking simulation. In *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99)*, pages 797–807, 1999.

[67] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, et al. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, pages 10823–10825, 2008.

[68] Marc H Raibert et al. *Legged robots that balance*, volume 3. MIT press Cambridge, MA, 1986.

[69] A. Ruina, J.E.A. Bertram, and M. Srinivasan. A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition. *Journal of Theoretical Biology*, 237(2):170–192, 2005.

[70] Patrick Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29(2):187–209, 1994.

[71] Patrick Saint-Pierre. Approximation of viability kernels and capture basins for hybrid systems. In *European Control Conference*, pages 2776–2783, 2001.

[72] Patrick Saint-Pierre. Hybrid kernels and capture basins for impulse constrained systems. In *Hybrid Systems: Computation and Control*, pages 378–392. Springer, 2002.

[73] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent asimo: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.

[74] AM Schillings, BMH Van Wezel, TH Mulder, and J. Duysens. Muscular responses and movement strategies during stumbling over obstacles. *Journal of Neurophysiology*, 83(4):2093–2102, 2000.

[75] AL Schwab and M Wisse. Basin of attraction of the simplest walking model. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 6, pages 531–539, 2001.

[76] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[77] M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75, 2005.

[78] P-F Tang, Marjorie H Woollacott, and Raymond KY Chong. Control of reactive balance adjustments in perturbed human walking: roles of proximal and distal postural muscle activity. *Experimental Brain Research*, 119(2):141–152, 1998.

[79] Darryl G Thelen, Laura A Wojcik, Albert B Schultz, James A Ashton-Miller, and Neil B Alexander. Age differences in using a rapid step to regain balance during a forward fall. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 52(1):M8–M13, 1997.

[80] Alf Thorstensson and H Roberthson. Adaptations to changing speed in human locomotion: speed of transition between walking and running. *Acta Physiologica Scandinavica*, 131(2):211–214, 1987.

[81] Claire J Tomlin, Ian Mitchell, Alexandre M Bayen, and Meeko Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

[82] Yao-Yang Tsai, Wen-Chieh Lin, Kuangyou B Cheng, Jehee Lee, and Tong-Yee Lee. Real-time physics-based 3d biped character animation using an inverted pendulum model. *Visualization and Computer Graphics, IEEE Transactions on*, 16(2):325–337, 2010.

[83] Michiel Van De Panne. From footprints to animation. In *Computer Graphics Forum*, volume 16, pages 211–223. Wiley Online Library, 1997.

[84] Gijs Van Oort and Stefano Stramigioli. Using time-reversal symmetry for stabilizing a simple 3d walker model. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4673–4678. IEEE, 2007.

[85] M. Vukobratović and J. Stepanenko. On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15(1):1–37, 1972.

[86] Miomir Vukobratović and Branislav Borovac. Zero-moment point – thirty

five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

[87] Miomir Vukobratovic and Davor Juricic. Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on*, (1):1–6, 1969.

[88] Boston Dynamics website. `http://www.bostondynamics.com`.

[89] Eric R Westervelt. *Toward a coherent framework for the control of planar biped locomotion*. PhD thesis, Citeseer, 2003.

[90] Eric C Whitman and Christopher G Atkeson. Control of a walking biped using a combination of simple policies. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 520–527. IEEE, 2009.

[91] P.B. Wieber. On the stability of walking systems. In *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002.

[92] Pierre-Brice Wieber. Constrained dynamics and parametrized control in biped walking. In *International Symposium on Mathematical Theory of networks and systems*, 2000.

[93] Pierre-Brice Wieber and Christine Chevallereau. Online adaptation of reference trajectories for the control of walking systems. *Robotics and Autonoous Systems*, 54(7):559–566, 2006.

[94] David A Winter. *Biomechanics and motor control of human gait: normal, elderly and pathological*. 1991.

[95] Martijn Wisse, CG Atkeson, and DK Kloimwieder. Swing leg retraction helps biped walking stability. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 295–300. IEEE, 2005.

[96] W. Wolfslag. Basin of attraction of controlled simplest walker. (unpublished), 2012.

GLOSSARY

ZMP	Zero Moment Point
CoM	Center of Mass
HZD	Hybrid Zero Dynamics
MPC	Model Predictive Control
IP	Inverted Pendulum
LIP	Linear Inverted Pendulum
$SLIP$	Spring-Loaded Inverted Pendulum
SW	Simplest Walking (model)
ODE	Ordinary Differential Equation
DOF	Degree(s) Of Freedom
$TCOT$	Total Cost Of Transport

Viable and controllable regions

V_0	0-step viable region: all non-failed states at midstance
V_n	n-step viable region ($n \geq 1$): all initial states, from which the robot can take at least n steps and not fail
V_∞	∞-step viable region (viability kernel): all initial states, from which the robot can step indefinitely
C_0	target region: the set of all states satisfying a given goal
C_n	n-step controllable region ($n \geq 1$): all initial states, from which the robot can reach the target C_0 in n or fewer steps
C_∞	∞-step controllable region: all initial states, from which the robot can reach the target C_0 or approach it asymptotically

$C(\cdot)$	controllability map: the 1-step controllable region as a function of the target region
$C^{\infty}(\cdot)$	∞-step controllability map: the ∞-step controllable region as a function of the target region
$\hat{C}_n$	strict n-step controllable region ($n \geq 1$): all initial states, from which the robot is able to reach the target C_0 in exactly n steps
$\hat{C}_{\infty}$	strict ∞-step controllable region: all initial states, from which the robot can approach the target C_0 asymptotically
$\hat{C}(\cdot)$	strict controllability map: the strict 1-step controllable region as a function of the target region
$\mu(\cdot)$	measure (volume metric) of a set
2^X	power set of the set X, i.e. the set of all subsets of X
U	control space: the space formed by considered control parameters of a model
S	Poincaré section
$\bar{S}$	extended Poincaré section: the Cartesian product $S \times U$
$\bar{q}$	extended state: a point in $\bar{S}$, i.e. a combination (q, u) of a state $q \in S$ and controls $u \in U$
$\bar{V}_0$	extended 0-step viable region: all combinations of non-failed states and allowed controls
$\bar{V}_n$	extended n-step viable region ($n \geq 1$): all combinations of initial states and controls, for which the robot can take n steps and not fail
$\bar{V}_{\infty}$	extended ∞-step viable region: all combinations of initial states and controls, for which the robot can step indefinitely

$\bar{C}_n$	extended n-step controllable region ($n \geq 1$): all combinations of initial states and controls, for which the robot can reach the target C_0 in n or fewer steps
$\bar{C}_\infty$	extended ∞-step controllable region: all combinations of initial steps and controls, for which the robot can reach the target C_0 or approach it asymptotically
$\hat{\bar{C}}_n$	extended strict n-step controllable region ($n \geq 1$): all combinations of initial states and controls, for which the robot can reach the target C_0 in exactly n steps
$\hat{\bar{C}}_\infty$	extended strict ∞-step controllable region: all combinations of initial states and controls, for which the robot can approach the target C_0 asymptotically
$\hat{\bar{C}}(\cdot)$	extended controllability map: the extended strict 1-step controllable region as a function of the target region
$\bar{V}_n^u$, $\bar{C}_n^u$	extended n-step viable/controllable region, corresponding to the state space extended by the control parameter(s) u

Inverted pendulum model in 2D

θ	angle between the stance leg and the vertical
$\dot{\theta}$	angle rate of the stance leg
x_{st}	control parameter step-length
p	control parameter magnitude of the push-off impulse
t_{st}	step time (also, swing time): time from the initial midstance to the next instant of collision
θ_{sw}	angle between the swing leg and the vertical just before the instant of collision

$\dot{\theta}_0$	initial velocity of the robot at midstance
$\dot{\theta}_1$	velocity of the robot at the next midstance
$\dot{\theta}_t$	target velocity of the robot at midstance
p_{max}	actuator limitation: the maximum allowed push-off impulse p
$t_{st,min}$	actuator limitation: the minimum allowed step-time t_{st}
$\bar{V}_n^{x_{st}}$	extended n-step viable region ($n \geq 0$), where the extension is by the control-parameter step-size x_{st}
$\bar{V}_n^{p}$	extended n-step viable region ($n \geq 0$), where the extension is by the control-parameter push-off p
$\bar{V}_n^{p,x_{st}}$	(full) extended n-step viable region ($n \geq 0$), where the extension is by both control-parameters the step size x_{st} and push-off p
$\bar{C}_n^{x_{st}}$	extended n-step controllable region ($n \geq 1$), where the extension is by the control-parameter step size x_{st}
$\bar{C}_n^{p}$	extended n-step controllable region ($n \geq 1$), where the extension is by the control-parameter push-off p
$\bar{C}_n^{p,x_{st}}$	extended n-step controllable region ($n \geq 1$), where the extension is by both control-parameters the step size x_{st} and push-off p
p_{gr}	magnitude of the collision impulse from the ground along the colliding leg
v_p	velocity of the hip just after the push-off impulse
$\vec{r}_{st}$	unit vector from the stance foot to the hip at the instant of collision
$\vec{r}_{sw}$	unit vector from the swing foot (collision point) to the hip at the instant of collision
$(\cdot)^-$	value of a given variable just before the instant of collision
$(\cdot)^+$	value of a given variable just after the instant of collision

y_{sw}	height of the swing foot above the ground
F_{st}	compressive force in the stance leg

Linear inverted pendulum model in 2D

x_h	horizontal position of the hip relative to the stance foot
y_h	height of the hip above the ground
$\dot{x}_h$	horizontal velocity of the hip
l	length of the stance leg
x_{st}	control parameter step-length
t_{st}	control parameter step-time (also, swing-time): time from the initial midstance to the next instant of collision
l_{max}	maximum allowed length of each leg
x_{max}	horizontal distance from the hip to the stance foot when the stance leg is at the maximum allowed length l_{max}
y_G	constant height of the hip above the ground (model constraint)
v_0	initial velocity of the robot at midstance
v_1	velocity of the robot at the next midstance
v_t	target velocity of the robot at midstance
$t_{st,min}$	actuator limitation: the minimum allowed step-time t_{st}
$\bar{V}_n^{x_{st}}$	extended n-step viable region ($n \geq 0$), where the extension is by the control-parameter step-size x_{st}
$\bar{V}_n^{t_{st}}$	extended n-step viable region ($n \geq 0$), where the extension is by the control-parameter step-time t_{st}
$\bar{V}_n^{t_{st},x_{st}}$	(full) extended n-step viable region ($n \geq 0$), where the extension is by both control-parameters the step size x_{st} and step-time t_{st}

$\bar{C}_n^{x_{st}}$	extended n-step controllable region ($n \geq 1$), where the extension is by the control-parameter step size x_{st}
$\bar{C}_n^{t_{st}}$	extended n-step controllable region ($n \geq 1$), where the extension is by the control-parameter step-time t_{st}
$\bar{C}_n^{t_{st},x_{st}}$	extended n-step controllable region ($n \geq 1$), where the extension is by both control-parameters the step size x_{st} and step-time t_{st}
$(\cdot)^-$	value of a given variable just before the instant of collision
$(\cdot)^+$	value of a given variable just after the instant of collision
$\vec{r}_h$	position vector of the hip relative to the stance foot
F_{st}	compressive force in the stance leg

Walking controller design

$x_{st,max}$	actuator limitation: the maximum allowed step-size x_{st}
$\dot{\theta}^*$	nominal (target) velocity of the IP model of Ranger at midstance
x_{st}^*	nominal step length of the IP model of Ranger
p^*	nominal push-off magnitude of the IP model of Ranger
$u_x(\cdot)$	step-size controller: a function of the midstance velocity $\dot{\theta}_0$ which outputs the desired step-size x_{st}
$u_p(\cdot)$	push-off controller: a function of the midstance velocity $\dot{\theta}_0$ which outputs the desired push-off p
$\Delta\dot{\theta}$	absolute value of the error in velocity, $\lvert\dot{\theta} - \dot{\theta}^*\rvert$
ν	the (next-step) relative error in velocity: the ratio of the error $\Delta\dot{\theta}$ in velocity at the next midstance to the initial error; ν is a function of the initial velocity $\dot{\theta}_0$, step-size x_{st}, and push-off p
ν_x	the best relative error in velocity: the smallest relative error ν over all push-offs p

ν_p	the relative error ν which assumes the (proposed) step-size controller u_x is used
d	distance from an extended state $(\dot{\theta}^0, x_{st})$ to the minimum-step-time boundary of the region $\bar{C}_\infty^{x_{st}}$
$\bar{V}_{u_x}^p$	the set of all combinations $(\dot{\theta}_0, p)$ of the initial velocity and push-off that are viable, if the (proposed) step-size controller u_x is used
α	a parameter of the step-size controller for Ranger: the slopes of the controller function; also, the approximate slope of the minimum-step-time boundary of the region $\bar{C}_\infty^{x_{st}}$ for Ranger
α_p	a parameter of the push-off controller for Ranger: the slope of the controller function for velocities around the nominal velocity
β_p	a parameter of the push-off controller for Ranger: the slope of the controller function for small velocities
p_0	a parameter of the push-off controller for Ranger: the commanded push-off at zero midstance velocity
μ_v	random perturbation in the midstance velocity $\dot{\theta}_0$
μ_x	random perturbation in the control-parameter step-size x_{st}
μ_p	random perturbation in the control-parameter push-off p
m_v	magnitude of the random velocity perturbation μ_v
m_x	magnitude of the random step-size perturbation μ_x
m_p	magnitude of the random push-off perturbation μ_p
M_v	largest magnitude m_v of velocity perturbations for which the robot does not fail, assuming no other perturbations in the model

M_x	largest magnitude m_x of step-size perturbations for which the robot does not fail, assuming no other perturbations in the model
M_p	largest magnitude m_p of push-off perturbations for which the robot does not fail, assuming no other perturbations in the model
$\tilde{\dot{\theta}}_0$	midstance velocity after the random perturbation μ_v
$\tilde{x_{st}}$	control-parameter step-size after the random perturbation μ_x
$\tilde{p}$	control-parameter push-off after the random perturbation μ_p

Miscellaneous

m_{cont}	number of control parameters per step
m_{goal}	number of restrictions on the state-space imposed by a given goal of locomotion
γ	slope angle in passive walking models, such as Simplest Walking model