Capturability-Based Analysis and Control of Legged Locomotion

by

Twan Koolen

THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE

Thesis committee
Prof. Dr. Frans C.T. van der Helm
Prof. Dr. Robert Babuška (external committee member)
Dr. Martijn Wisse (mentor)
Ir. Tomas de Boer (mentor)

Faculty of Mechanical, Maritime and Materials Engineering
Delft University of Technology

March 16, 2011

Capturability-Based Analysis and Control of Legged Locomotion, Part 1: Concepts and Definitions

John Rebula, Twan Koolen, Tomas de Boer, Jerry Pratt

Abstract—

This three-part paper discusses the analysis and control of legged locomotion in terms of N-step capturability: the ability of a legged system to come to a stop without falling by taking N or fewer steps. We consider this ability to be crucial to legged locomotion and a useful, yet not overly restrictive criterion for stability.

Part 1 lays the theoretical foundation for this approach. Formal definitions of N-step capturability and related terms are given, and general disturbance robustness metrics based on capturability are proposed. In Part 2, we will use the theoretical framework developed in the current part to analyze N-step capturability for three simple gait models. Exact results from simplified models can be applied as approximations to analyze more complex models and real robots. Part 3 describes how the results for the simple models were used to control a complex lower body humanoid robot with two six degree of freedom legs.

Index Terms—

Capture point, Robot performance measure, Gait stability measure, Viability, Legged robots, Capturability.

I. Introduction

PREVENTING falls is essential in legged locomotion. A fall can be energetically costly and dangerous for a legged system and other agents. Healthy humans are able to avoid falling in almost all conditions experienced in everyday life. While many legged robots can currently walk, run, and dance without falling, these tasks are usually performed in a controlled environment. Unexpected perturbations will easily topple most current bipedal robots. The ability of legged robots to avoid falling must be significantly improved before they can find utility in complex environments.

Measuring how close a legged system is to falling can provide useful insight into controller design. However, effectively quantifying closeness to falling is challenging. For traditional control systems, stability can be analyzed using measures such as eigenvalues, phase margins or loop gain margins. However, legged locomotion is generally characterized by non-linear dynamics, under-actuation, possibly non-periodic motions, and a combination of continuous and discrete dynamics. These properties limit the relevance of traditional analysis and control techniques to legged locomotion.

Existing stability measures for legged locomotion such as the Zero Moment Point stability criterion may be readily computed but only apply to specific classes of trajectories [1], [2]. On the other hand, more general techniques, such as the Viability Margin [3], have been proposed but are difficult to compute, limiting their usefulness. We propose the analysis of legged locomotion based on *capturability*, which offers measures applicable to mostly general motions, may be readily approximated, and are useful in designing controllers that prevent falls.

Instead of identifying states and actions that will not lead to a fall, we will consider states and actions that allow the system to eventually come to a stop. We argue that such states and actions are conceptually easier to identify and serve as a useful conservative subset of the states and actions that will not lead to a fall. Furthermore, we propose that the number of steps required to come to a stop may be used to measure how close the legged system is to falling.

Central in our approach is the concept of capturability, or more formally *N-step capturability*, which we define as the ability of a legged system to come to a stop without falling by taking N steps or fewer, given its dynamics and actuation limits. We consider this ability to be crucial to legged locomotion because it specifies a useful criterion for fall avoidance, applicable to a wide range of legged locomotion scenarios. Divided over three parts, we will elaborate on definitions related to N-step capturability, how to approximate it, and its usefulness in the analysis and control of legged locomotion.

While other stability analysis and control techniques implicitly provide indicators of the ability of a system to avoid a fall, they often require significant assumptions and limitations. The advantage of capturability analysis is that it does not assume a periodic limit cycle, nor does it require a specific control system design. Instead, it takes the ability to eventually come to a stop as a basis for analysis and control. This basis may not be appropriate for certain classes of legged systems such as purely passive dynamic walkers, but it is applicable in nearly any situation that a practical legged system may find itself in. The concept of capturability is applicable to locomotion over rough terrain with impassable regions, grasping end effectors, crawling, brachiating, and break dancing.

Manuscript received December 2010. This work was funded through the Army Tank and Automotive Research and Development Command, the Defense Advanced Research Projects Agency, the Office of Naval Research, NASA, and the Honda Research Institute.

J. Rebula is with the University of Michigan, Ann Arbor

T. Koolen and T. de Boer are with the Delft University of Technology, the Netherlands

J. Pratt and T. Koolen are with the Institute for Human and Machine Cognition, Florida

Both preventing a fall and coming to a stop require adequate foot placement. Foot placement plays a critical role in the evolution of a legged system as a result of the ground reaction force constraints that are typical to legged locomotion. The surface of a foot can only exert unilateral pressure on the environment, and the ratio of shear stress to normal stress is limited by friction. Therefore, the overall interaction forces and torques between the ground and the legged system are constrained by the location of the feet. With non-grasping feet on flat ground, these constraints can be summarized by stating that the center of pressure on the ground must lie within the convex hull of the support feet and the ground reaction force vector must lie inside the friction cone. With more complex ground interaction, like grasping, the constraints are more complex. Since foot placement is critical to preventing a fall, we will focus on it in this paper.

Outline

In Part 1, we introduce the theoretical framework. We define concepts required to analyze the capturability of legged systems and present an example scenario to illustrate the definitions. In addition, we present a metric for stability, based on the available step locations that bring a system to a stop. In this part, we keep the presented framework general so that it is applicable to a wide range of legged locomotion scenarios. In parts 2 and 3 we will demonstrate the usefulness of the capturability framework by applying it to periodic and non-periodic bipedal walking on relatively easy terrain.

We conjecture that determining the ability of a legged robot or animal to come to a stop is a very challenging task in general. Simplified models of walking can be used to approximate capturability of more complex models. As we will show in Part 2, N-step capturability can be calculated exactly for simple models, such as a Linear Inverted Pendulum [4], [5], with added finite-sized feet and upper body momentum.

These solutions can then be used as approximations in more complex models or in real legged robots. In Part 3, we will show that these approximate solutions are very useful for controlling humanoid balance, push recovery, and walking.

The remainder of this part is structured as follows. Section II provides a survey of relevant literature. Section III contains definitions of the various concepts that constitute the proposed capturability framework. In Section IV, we provide a number of quantitative robustness metrics based on these definitions. Finally, a discussion is provided in Section V.

II. Background

The question "how stable is a given legged system?" has been the subject of much research and debate, in both robotics and biomechanics. We will now present previous work attempting to answer this question, including previous work on capturability.

The Zero Moment Point (ZMP) is often used as an aid in control development, with the constraint that it must remain in the interior of the base of support of a legged robot. A common ZMP control method is to maintain the ZMP along a precomputed reference trajectory [6]. During walking, the error between the actual and desired ZMP can be used as a measure of the error of the current state of the robot [7]. The repeatability of the gait can also be used as an error measure [6]. One drawback to following a precomputed trajectory is the inability of the robot to recover from a large unexpected push. Further work has expanded the ZMP method to include step placement adjustment in reaction to disturbances [8], [9], but there is no measure of the ability of the robot to reactively avoid a fall when following a given preplanned ZMP trajectory. In addition, the ZMP requires significant modification to apply to non-flat terrain [3] or dynamic gait with a foot that rotates on the ground.

Poincaré maps have been used to measure the local stability of periodic gaits, and to induce periodic gaits of real robots based on reference trajectories [10]. Based on Poincaré Map analysis, the Gait Sensitivity Norm [2] provides a measure of robustness for Limit Cycle Walkers [11] and has been shown to correlate well with the disturbance rejection capabilities of simulated planar walkers. The Gait Sensitivity Norm is calculated as the sensitivity of a given gait measure, such as step time, to a given disturbance type, such as a step down in terrain, using a simulated model or experimental data. Another Poincaré map method based on Floquet multipliers has been used to analyze the stability of human walking gaits [12]. However, Poincaré map analysis assumes cyclic gait to yield a measure of stability. In addition, it requires a linearization of walking at a given point in the gait cycle, which limits the applicability of the method to large disturbances between steps where the linearization fails to capture essential dynamics of the motion [12].

Poincaré map analysis has also been applied to the case of passive Limit Cycle Walkers under stochastic environmental perturbations [13], without linearizing the system around the fixed point, yielding a probabilistic basin of attraction. The stability of a walker is described with a mean first passage time, which is the expected number of steps before failure, given a set of statistics for the stochastic environmental disturbance. However, this method assumes an approximately periodic gait, and does not apply to large general disturbances such as a significant push. Poincaré map analysis has been extended to control a walker in acyclic desired gaits [14], by applying linear control based on a continuous family of Poincaré maps along the entire trajectory. This control method can provide a measure of robustness about the desired trajectory, but it does not consider the robustness of the desired trajectory itself.

The concepts of Virtual Constraints and Hybrid Zero Dynamics have been used to obtain and prove asymptotic stability of periodic motions for walking robots [15]. Introducing Virtual Constraints reduces the dimensionality of the walking system under consideration by choosing a single desired gait, allowing a tractable stability analysis. However, if actuator limitations render the robot incapable of maintaining the Virtual Constraints after a large perturbation, it is possible a fall could be avoided only by changing the desired trajectory to alter foot placement and use of angular momentum (see Part 2).

The Foot Placement Estimator, like the present work, considers the footstep location to be of primary importance and

can be used both to control and to analyze bipedal systems [16]. For a simple planar biped that maintains a rigid A-frame configuration, the Foot Placement Estimator demarcates the range of foot placement locations that will result in a statically standing system. This approach is quite similar to ours, though it is unclear how to extend this method to more general systems.

Wieber uses the concept of Viability theory [17] to reason about the subset of state space in which the legged system must be maintained to avoid falling in the general case. He shows a method of analyzing the Lyapunov stability of standing on non-flat terrain given a balance control law. However, the standing assumption precludes the use of this method in walking, and it provides no information on choosing step locations to avoid falling. We also adopt Viability theory in capturability analysis, and explicitly compute acceptable regions to step. We will not, however, provide a Lyapunov stability analysis, as we do not assume a controller or associated Lyapunov function.

In previous work, we have implicitly used the concept of capturability to develop the notion of capture points, the places on the ground to step that will allow a legged robot to come to a stop. We have used capture points to control various simulated and real legged robots. Some simulation models closely match the simplified models which we used to estimate capture points, [18]. In these models, nearly exact control was achieved (modulo numeric round-off and slight variations between the simulation and simplified model), validating the simplified models and capture point calculations. We have used capture points based on simple models to control more complex models and a simulated 12 degree of freedom (DoF) humanoid robot to design controllers that balance, recover from pushes, and walk across randomly placed stepping stones [19], [20]. Some of these capture point-based control methods were also implemented on a real 12 DoF robot [21]. We will extend the concept of capture points, applying the theory to general legged systems, considering multiple steps and providing a more complete analysis of the ability of a legged system to come to a stop.

III. Definitions

This section presents definitions for the concepts that constitute the capturability framework. We illustrate these concepts with an example scenario.

A. Example Scenario

Fig. 1 depicts a situation which will be used to illustrate the theoretical framework to be developed in this section.

In the example, a person is about to cross a pond that has some stepping stones in it. Before starting to walk, the person can avoid falling without taking a step. In addition, there are a few stones that the person can step to without falling. The person starts stepping to one of the stones. At the beginning of the first swing phase, it will be possible for the person to change his mind and step to one of the other stones. However, part way through the swing phase, the person will be committed to stepping to the chosen stone to avoid a fall. For moderate and slow speed walking, the person will have

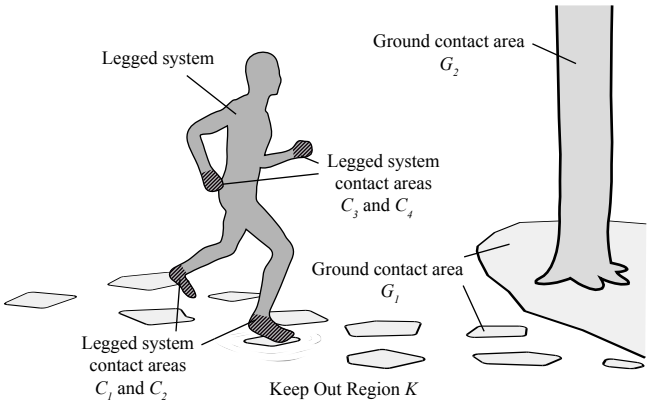

Fig. 1. An example scenario involving legged locomotion. A person is walking across stepping stones in a pond. The person wishes to avoid falling into the pond or onto the ground. It is acceptable, however, to maintain balance by placing a hand on the nearby tree. To define this notion of falling, we have specified allowable contact in terms of robot and ground contact areas (labeled).

the option of either coming to a stop after the first step, or taking a step to another stone while maintaining his walking speed. This option of stopping without taking another step may cease to be available at some point in the execution of the next swing phase.

Suppose that the person slips and momentarily loses his balance. Whereas the person could previously choose from several stones to step to next, he must now step to a specific stone in order to prevent a fall. The person may need to lunge his upper body or windmill his arms to regain balance. In addition, the person may choose to steady himself against a tree at the edge of a pond using a hand. If no combination of stones, trees, and body lunging options is available to prevent a fall, then the person will find himself in the pond.

We aim to introduce a framework to more precisely describe this scenario, as well as a capturability margin that will describe how close the person is to falling.

B. Hybrid Dynamic Systems and Viability

Before defining capturability for a legged system we will consider the more general concept of *viability* of a hybrid dynamic system, as developed in [17] and introduced in the field of legged locomotion in [3], [22].

Definition **(Hybrid Dynamic System)** Let X (the state space) and $\mathcal{U}$ (the control space) be finite-dimensional spaces.

A *hybrid dynamic system* is a system which has continuous flow dynamics described by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{1a}$$

$$\mathbf{u}(t) \in U(\mathbf{x}(t)) \tag{1b}$$

and discrete jump dynamics described by

$$h_i(\mathbf{x}) = 0 \Rightarrow \mathbf{x}_+ = \mathbf{g}_i(\mathbf{x}_-) \quad i \in I \subset \mathbb{N} \tag{2}$$

Here, $\mathbf{x}(t) \in X$ is the state of the system and $\mathbf{u}(t)$ is the system's control input, which is confined to the state-dependent set of allowable control inputs $U(\mathbf{x}(t)) \subset \mathcal{U}$. The vector field

$\mathbf{f}$ desribes the flow dynamics. When the system state $\mathbf{x}_-$ lies on a switching surface defined by $h_i(\mathbf{x}_-) = 0$, discrete jump dynamics $\mathbf{g}_i$ map the state to $\mathbf{x}_+$ instantaneously.

A subset of the switching surfaces will be selected as the *stepping surfaces* of the hybrid dynamic system, defined by the index set $J \subset I$.

An *evolution* associated with the system is a solution to flow dynamics (1a) and jump dynamics (2) for some input satisfying (1b).

For the example of the person crossing a pond, the person can be modeled as a hybrid dynamic system. The state $\mathbf{x}(t)$ contains all the state variables that are needed to determine the current and future evolution of the system, together with the control inputs. The control inputs $\mathbf{u}(t)$ are the low level muscle signals. We will model the dynamics of the person, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$, as the dynamics of the person's body, including the mechanical linkages and muscle dynamics. The discrete switching surfaces model the change in the flow dynamics when a step is taken or when the knee locks.

Examples of evolutions include standing still, taking a step, taking several steps, and falling into the pond. Clearly some evolutions are more desirable than others. We will use the term viable to describe those that avoid certain failed states.

Definition **(Viable)** Let X be the state space of a hybrid dynamic system and let $X_{\text{failed}} \subset X$ be the set of *failed states*. An evolution $\mathbf{x}(\cdot)$ is *viable* on $[0, t_f]$ if for all $t \in [0, t_f]$, $\mathbf{x}(t) \in X \setminus X_{\text{failed}}$. An evolution is *globally viable* if it is viable on $[0, \infty]$.

A subset A of state space X is said to be *locally viable* (or to enjoy the *local viability* property) if for any initial state $\mathbf{x}_0 \in A$, there exists at least one evolution that is viable on $[0, t_f]$, for some $t_f > 0$. It is said to be *(globally) viable* if we can take $t_f = \infty$.

A dynamic system in state $\mathbf{x}_0$ is *locally viable* if $\{\mathbf{x}_0\}$ is locally viable, and *(globally) viable* if $\{\mathbf{x}_0\}$ is globally viable.

For the person crossing a pond, viable evolutions include quiet standing and running across the stepping stones. See Fig. 2 for a graphical representation.

Definition **(Viability kernel)** The viability kernel is the set of all initial states $\mathbf{x}_0 \in X \setminus X_{\text{failed}}$ for which at least one evolution starting at $\mathbf{x}_0$ is globally viable.

The viability kernel [17] defines the safe subset of state space in which the dynamic system must operate to maintain the ability to prevent failure (see Fig. 3). Using a measure of distance from the current state to the edge of the viability kernel has been proposed as a robustness metric, though one which is very difficult to compute in general [3]. Also, it is unclear how to define a dynamically relevant distance in state space. Therefore, we propose a capturability analysis, which considers the ability to eventually come to a stop, leading to a capturability metric that is simple to calculate for simple models (Part 2). Results from these simple models are effective when analyzing complex robots (Part 3).

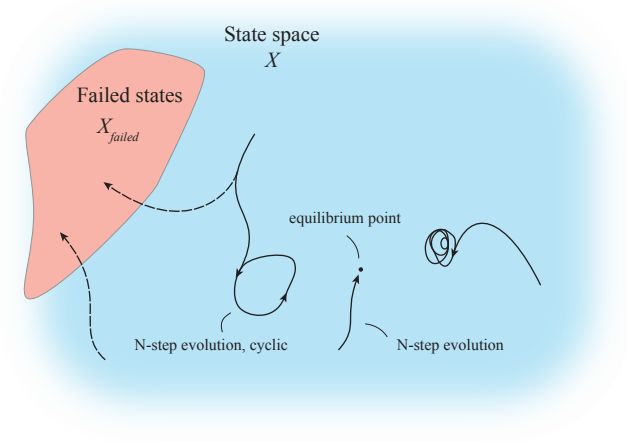

Fig. 2. A representation of the state space for a hybrid dynamic system. Several evolutions are shown, including trajectories that lead to failed states. Note that two different evolutions may share the same initial state if the controls $\mathbf{u}(t)$ are chosen differently. Adapted from [3].

C. Capturability

The viability kernel fully defines all states from which failed states can be avoided. Here we consider a more restricted but important subset of states: those which require a finite number of crossings of hybrid switching surfaces to avoid failing. We will refer to those states as capturable states.

Definition **(N-step evolution)** An *N-step evolution* is an evolution that contains at most N crossings of the stepping surfaces $h_j(\mathbf{x}) = 0$ for some $j \in J$.

Definition **(N-step capturability)** A state $\mathbf{x}_0$ is *N-step capturable*, for $N \in \mathbb{N}$, if there exists at least one globally viable N-step evolution starting at $\mathbf{x}_0$. The *N-step viable-capture basin* is the set of all N-step capturable states. The 0-step viable-capture basin will also be referred to as the set of *captured states* and a hybrid dynamic system in the 0-step viable-capture basin basin will be referred to as *captured*.

By induction, if follows that for $M \geq 0$ and $N > M$, there exists a viable $(N - M)$-step evolution that starts in the N-step viable-capture basin and reaches the M-step viable-capture basin. A system that is N-step capturable can reach a captured state in N steps or fewer.

We see that capturability is a stricter constraint than viability, since the system has to be able to avoid failure with no more than N crossings of hybrid switching surfaces. We believe capturability analysis is typically computationally simpler than viability analysis, since one only has to consider the flow dynamics, and not the switching surfaces, to determine 0-step capturable states. Then to determine the 1-step viable-capture basin, one can determine the states that can reach the 0-step viable-capture basin through a single step. This process can be repeated recursively. The N-step viable-capture basin will be the set of states that can reach the $(N-1)$-step viable-capture basin in a single step.

For small N, computing N-step capturability may be much easier than computing viability in general. In addition, the N-step viable-capture basin can be approximated by finding

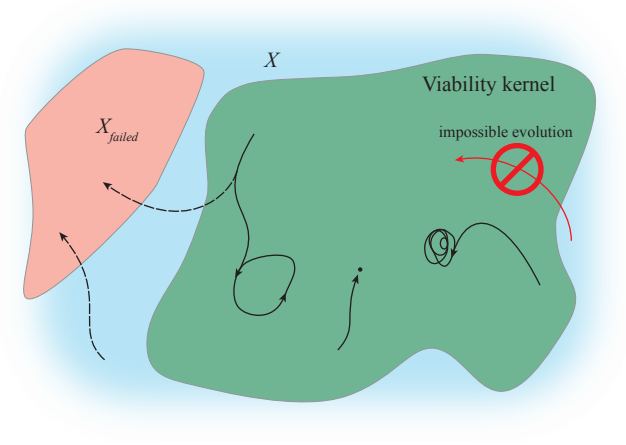

Fig. 3. A representation of the viability kernel for a hybrid dynamic system, extended from Fig. 2. The viability kernel contains all states in which it is possible to avoid a failed state forever, so that all states outside the viability kernel inevitably lead to a failed state. It is impossible for an evolution to cross into the viability kernel (impossible evolution shown).

states that are easily determined to be 0-step capturable (such as fixed points or 0-step limit cycles) and recursing out to find states that can reach that set of states in a single crossing of a hybrid switching surface.

D. Legged Systems

The above definitions were general to hybrid dynamic systems. We now make definitions specific to legged systems that will fit them into the above framework. In order to properly define relevant concepts such as falling and stepping, we need to define when contact between sets of particles occurs.

Definition **(Contact)** Let W denote the set of all particles in a three-dimensional Euclidean space $\mathbb{R}^3$. Let $p : (W, X) \to \mathbb{R}^3$ be a mapping from particles in W and states in X to particle positions in $\mathbb{R}^3$. Let $A \subset W$ and $B \subset W$ be two sets of particles, and let $\mathbf{x}$ be the state of the robot. We define *contact* between A and B in state $\mathbf{x}$ as

$$\text{contact}(A, B, \mathbf{x}) \Leftrightarrow \inf_{\substack{a\in A\\ b\in B}} \|p(a, \mathbf{x}) - p(b, \mathbf{x})\| = 0$$

Examples of contact for the person in state $\mathbf{x}(t)$ crossing a pond include a foot, C_1, resting on a stepping stone, G_1, denoted as $\text{contact}(C_1, G_1, \mathbf{x}(t))$ (Fig. 1).

Definition **(Legged System)** A *legged system* is a hybrid dynamic system augmented by:

- a set of *legged system particles* $R \subset W$, whose positions in $\mathbb{R}^3$ at time t depend only on $\mathbf{x}(t)$;
- an indexed family of sets of legged system particles called the *legged system contact areas*, $\{C_m\}_{m\in M}$, where M is the set of index values, and $\forall m \in M, C_m \subset R$;
- an indexed family of sets of non-legged system particles called the *ground contact areas*, $\{G_m\}_{m\in M}$, where $\forall m \in M, G_m \subset W \setminus R$;
- a set of particles K that are neither part of the legged system nor of a ground contact area, $K \cap R = K \cap (\cup_{m\in M} G_m) = \emptyset$, called the *keep out region*;
- a single particle in each legged system contact area, selected as the *contact reference point*.
- The stepping surfaces are defined to be reached when a legged system contact area C_m comes into contact with its corresponding ground contact area G_m. For a legged system undergoing a valid evolution $\mathbf{x}(t)$ over the interval $[t_0, t_f]$, with a t^*, $t_0 < t_* < t_f$, the stepping surfaces are

$$\forall m \in M, h_m(\mathbf{x}(t^*)) = 0 \Leftrightarrow$$
$$\neg\,\text{contact}(C_m, G_m, \mathbf{x}(t^*_-)) \wedge \text{contact}(C_m, G_m, \mathbf{x}(t^*_+))$$

We define a *step* as the discrete event occuring at t_*, when the state lies on a stepping surface.

The set of particles R of the example person consists of the body and clothing of the person. In our example, depicted in Fig. 1 we define four disjoint legged system contact areas for the person. C_1 and C_2 correspond to the soles of each foot, and C_3 and C_4 correspond to the hands. Two ground contact areas are defined, G_1, which consists of the top surfaces of the stones and the area around the pond, and G_2, which consists of the surface of a tree beside the pond. We will specify the keep out region for the person crossing the pond to be the set of particles comprising the water of the pond.

Any point can be selected as the contact reference point in each legged system contact area. Contact reference points provide a convenient way of referring to the position of a contact area. Example choices might be the center of the foot, or a likely initial point of contact. For the example person, we will choose the contact reference points to be points on the bottom surface of each heel.

For the example person, a step occurs at the instant a foot contacts a stepping stone after swing. However, a rocking motion that alternately lifts the heel and toe of the same foot off of a stepping stone would not be considered stepping, since at least one point of the foot would maintain contact with the stepping stone at all times. However, lifting one of the feet off of a stepping stone and placing it back down onto a stepping stone would result in a step at the instant of foot contact. Likewise, placing a hand onto a tree would constitute a step.

E. Failed states

The set of failed states for a legged system can be specified as follows. Using the definitions above, we begin by defining a subset of state space in which the legged system is considered to have fallen.

Definition **(Fallen states):** Consider a legged system that has a state space X, and a set of particles R. Let $\{C_1, C_2, \dots\}$ denote the legged system contact areas corresponding to the legged system and let C denote their union. Similarly, let $\{G_1, G_2, \dots\}$ denote the associated ground contact areas and let G denote their union. Furthermore, let K be the legged system's keep out region.

The set of *fallen states* for this legged system, X_{fallen}, is the set of states for which there is contact between either:

- the legged system and the keep out region;
- any part of the legged system that is not part of a legged system contact area and the ground;

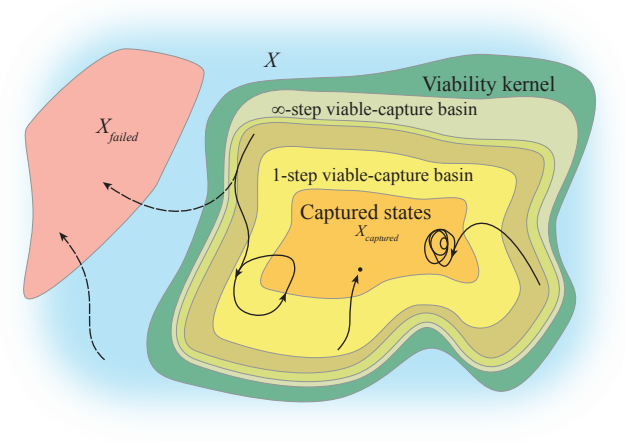

Fig. 4. A representation of the viable-capture basins for a legged system, extended from Fig. 3. Several N-step viable-capture basins are shown, including $N = 1$ and ∞. The boundary between two N-step viable-capture basins is a stepping surface. The ∞-step viable-capture basin approximates the viability kernel. States in the 0-step viable-capture basin, referred to as captured states, can be fixed points and infinite length 0-step evolutions (possibly a limit cycle).

- a legged system contact area and any ground contact area that is not associated with that legged system contact area.

More precisely,

$$X_{\text{fallen}} = \{\mathbf{x} \in X \mid \text{contact}(R, K, \mathbf{x}) \vee \text{contact}(R \setminus C, G, \mathbf{x}) \vee \exists i \mid \text{contact}(C_i, G \setminus G_i, \mathbf{x})\}$$

The set of fallen states includes states in which a stepping stone is contacted with a hand and states in which any part of the body touches the water. In addition to fallen states, there may be other undesirable states that would be damaging to a legged system. The fallen states, plus the undesirable states make up the failed states X_{failed}. Undesirable states for a person might include those describing hyperextension of a joint or dangerously high contact velocities.

F. Viability and Capturability for a Legged System

Suppose the person in our example is standing and balancing and does not need to take another step to avoid a fall. Then that person would be captured. If the person is walking slowly and can stop in a single step, then he would be 1-step capturable. If he was running quickly then he might require 3 steps in order to come to a stop and would therefore be 3-step capturable.

For $N > 0$, the N-step viable-capture basin is equivalent to the set of initial states $\mathbf{x}_0$ for which at least one 1-step evolution starting at $\mathbf{x}_0$ is viable and it reaches the $(N-1)$-step viable-capture basin in finite time. This allows us to answer questions of capturability iteratively, as shown in Part 2 for several simple models of gait.

Being able to eventually come to a stop is not an absolute requirement of avoiding a fall since the ∞-step viable-capture basin is a subset of the viability kernel. However, we argue that for typical legged systems, the additional requirement of being able to eventually come to a captured state has only minor implications, so that the difference between the ∞-step viable-capture basin and the viability kernel is 'small'. When considering normal human locomotion, it is hard to imagine a state in which a human can avoid falling down, but cannot eventually come to a captured state. One class of notable exceptions is purely passive dynamic walking robots, where walking persists in an infinite limit cycle with no possibility of reaching a captured state.

Fig. 4 shows a representation of a state space segmented into failed states, the viability kernel, the ∞-step viable-capture basin, and captured states, along with several evolutions that are viable and several that are not viable.

G. Capture Regions

While the viability kernel and the N-step viable-capture basin describe the *state space* in which a controller should maintain the robot, we will now define a subdivision of *Euclidean space* into capture regions, which demarcate desirable places to step. These capture regions will prove useful in finding control laws for a robot.

Definition (**N-step capture point**) Let $\mathbf{x}_0$ be the current state of a legged system. A point $\mathbf{r}$ is an *N-step capture point* for this legged system if and only if there exists at least one viable N-step evolution starting at $\mathbf{x}_0$ and ending in a captured state, in which a contact reference point is placed at $\mathbf{r}$ at the time of the first step in the evolution.

Definition (**N-step capture region**) The N-step capture region is the set of all N-step capture points.

The N-step capture region describes positions in Euclidean space that are suitable points for a legged system to place the contact reference point to achieve a captured state in N steps or fewer. Therefore, the N-step capture region describes all of the desired positions that a control system should target as swing locations. If the example person is constrained to contact the ground heel first, with the contact reference point on the heel, the N-step capture regions will be two dimensional areas on the ground (see Fig. 5).

IV. Robustness Metrics Based on Capture Regions

We now introduce the capturability margin, which attempts to answer the question "how robust is a given legged system?"

Definition (**N-step capturability margin**) An *N-step capturability margin* of a legged system in state $\mathbf{x}(t)$ is a measure of the size of the N-step capture region, such as surface area. The ∞-step capturability margin is referred to as the *capturability margin* for short.

For a simple legged system which always lands with the contact reference point on the ground, such as the models considered in Part 2, the ∞-step capture regions will be two dimensional surfaces on the ground. Therefore, the area of the two dimensional ∞-step capture region provides a useful capturability margin. This same measure is used in Part 3 to approximate the capturability margin of a real robot. The

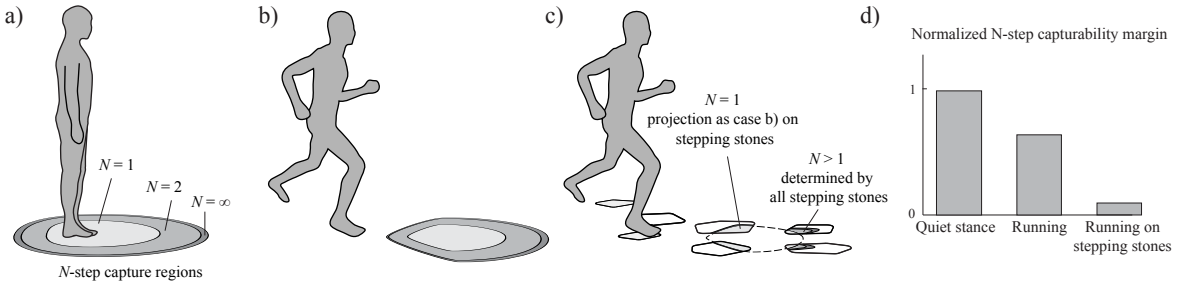

Fig. 5. a) A graphical representation of N-step capture regions for a human in a captured state, quiet stance. b) Smaller size N-step capture regions for the case of running compared to quiet stance. This corresponds to a smaller capturability margin. c) An even smaller capturability margin for the case of running on stepping stones. d) For all three cases, a representation of the magnitude of the normalized capturability margin: the capturability margin divided by the legged system capturability. The latter is equal in all three cases since we consider the same system.

capturability margin accounts for limited footholds availability naturally (see Fig. 5).

In the case of three dimensional N-step capture regions, such as when considering brachiation, the surface area of the N-step capture region may serve as a useful capturability margin as it can grow with the range of acceptable hand configurations at contact.

The capturability margin gives an idea of the foot placement accuracy and precision required to avoid a fall. In addition, a non-dimensional representation of the capturability margin can be calculated by normalizing it to a reference configuration, such as standing on flat ground.

Definition **(Legged system capturability)** The legged system capturability is the capturability margin of a legged system in a user defined base state configuration

Definition **(Normalized capturability margin)** The capturability margin of the legged system in the current state divided by the legged system capturability.

V. Discussion

We argue that maintaining viability is a fundamental task of legged control, ensuring that the system can avoid falling. We considered a slightly smaller subset of viable states, the capturable states, from which the system can avoid falling using only a finite number of steps. Finally, we presented N-step capture regions, subsets of Euclidean space, which provide a low dimensional indicator of capturability. This dimensionality reduction from m system states to 3 Euclidean dimensions is a result of using only a single particle on the legged system to generate a capture region, and can be thought of as a slice of the state space at the step, projected into Euclidean space. The dimensionality reduction in calculating N-step capture regions results in a loss of information, so capture regions do not completely describe capturability of a state as they do not specify the evolutions that lead to captured states. However, capture regions specify where a legged system should step to avoid falling, and provide a convenient way to quantify and visualize capturability.

A. Advantages of Capturability

The capturability margin and capture regions include several features making them useful for analysis and control. For example, the capturability margin tracks the system throughout a step, so a poor controller could result in the shrinking or loss of the N-step capture regions for the smallest existing N. The affects of a distrubance could likewise be described in changes to the capture regions before and after the disturbance. In addition, the capturability margin gives a measure of the largest allowable disturbance without falling; the largest allowable distrubance is simply the smallest disturbance resulting in a capturability margin of 0.

B. Computability of Capturability

Computing capturability can be conceptually easy in practice. One simple algorithm would be to start with a small set of states that are known to be captured, such as default standing positions. Then for points in the neighborhood of this set search for short evolutions that reach this set. For any that are found, grow the set. For any state that can reach a set of N-step capturable points in a single step, add it to the set of $(N-1)$-step capturable points.

While this algorithm is conceptually simple, it is likely computationally prohibitive for a complex legged system. In addition, including the full state of the system requires knowledge of all relevant environment information, such as the ground profile and contact characteristics. Encoding the entire environment for all time is prohibitive in general. Also note that for a system with regions of chaotic dynamics, the capturability may be uncomputable as determining whether some states are in the viable-capture basin may be undecidable [23].

To determine if a given state is capturable requires finding an evolution starting from that state that reaches a captured state. But to know if a state is captured requires verifying that a 0-step evolution exists for $t \to \infty$. Two evolutions that have this property are fixed points and infinite length evolutions (possibly 0-step limit cycles). We can find fixed points by finding some actuation $\mathbf{u}(t) \in U(\mathbf{x}(t))$ such that $\dot{\mathbf{x}}(t) = 0$. We can find 0-step limit cycles by finding a state and actuation trajectory such that no steps are taken, no failed states are reached, and $x(t + \Delta t) = x(t)$. We could also use

a Lyapunov-like analysis to show that a given volume of state space contains captured states without requiring them to be fixed points or part of a limit cycle. For very complex systems it may be impossible to consider infinite length evolutions. However, in such cases we could make approximations and find 0-step evolutions that are viable for a sufficiently long time such that for all practical purposes, we can consider it captured.

Although capturability may be computationally intractable to compute for a general robot, we have found that computing capturability for a simple model of a complex machine can provide useful insight for analysis and control. As we show in Part 2, capture regions are readily calculated for simple models of walking based on the Linear Inverted Pendulum model. The results from these models can be applied to more complicated machines to determine appropriate step locations to avoid falls, as we show in Part 3.

C. Capturability for a Given Controller

In this part we discussed capturability for all possible control inputs. Another approach would be to assume an existing controller and determine capturability given that controller. In order to simplify capturability analysis and control we can also assume a partial controller, such as a controller that provides balance and swing leg control, and takes a target step location as an input. Such a controller might have internal state, which must be incorporated into the robot state $\mathbf{x}(t)$, but the range of actuator inputs to consider might be more limited, easing capturability analysis of the partially controlled system. We have used this approach to reduce the actuation dimensionality of calculating 1-step capture regions of a complex lower body humanoid from 12 to 3, admitting a machine learning solution in simulation [20]. We also use such a parameterized controller for the complex robot in Part 3.

D. Marginal Capturability

Passive dynamic walking models [24] can walk forever and lack the ability to come to a stop. In fact, an infinitely repeatable gait in the viability kernel has been found for a simulated 3D passive walking model that has no captured states [25]. The capturability margin for a passive dynamic walker in a limit cycle is zero; since no actuation is possible, no actuation exists to bring the robot to a captured state. The concept of capturability therefore is not informative when considering the local robustness of a purely passive dynamic walker. However, with sufficient actuation, capturability based analysis and control applies to a limit cycle walker based on a passive dynamic walker.

E. Efficiency, Smoothness, and Other Considerations

We believe that capturability is one important factor in achieving safe legged locomotion, but it provides little information about other potentially important considerations such as energetic efficiency, actuation smoothness, speed, or accurate limb placement. The capturability margin may be best used as one indicator of an effective evolution, in conjunction with other measures to account for efficiency or other task specific requirements. For example, an energy sensitive task may benefit from a controller designed to maximize a weighted sum of energy efficiency and capturability in the spirit of linear quadratic regulation.

F. Application of Viability Theory

Viability theory in general [26] offers interesting control results, assuming $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ is convex for all $\mathbf{u}(t) \in U(\mathbf{x}(t))$. In particular, if a viability kernel is known, it is possible to compute viable controls based on a contingent cone that points inward at each point on the margins of the viability kernel. Based on this analysis, acceptable control can be calculated to maintain viability, using a simple strategy such as *slow control*, which employs minimal input magnitudes, or *heavy control*, which changes the control inputs as slowly as possible. If a viability kernel is further known to be compact and convex, viability theory states that there is an equilibrium point in the viability kernel, where $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = 0$.

The application of these results of viability theory require knowledge of a viability kernel for a particular robot, as well as certain conditions of convexity and compactness. While we have not explicitly investigated these properties for any particular robot dynamics in this paper, we believe that this may be an effective approach to designing the control of legged systems. In particular, calculation of viability kernels and associated controllers for the simple walking systems such as those presented in Part 2 might provide useful insight into controller design for more complex robots, as we show with capture regions in Part 3.

G. Scenario-specific Definitions

The generality of capturability analysis requires several scenario-specific user definitions as described in section III-D, and the choice of these definitions will impact the capture regions and associated capturability margins. As a result, the capturability margin of a legged system may not be directly comparable to the capturability margin of another legged system, or even the same system with different allowable contact regions. However, with careful definitions, capturability margins can quantify the benefits of adding control capabilities to a model, as we show in Part 2.

H. N-Step Capturability is a Relevant Approximation to Viability

If the primary objective of a control system is to assure failed states are avoided, then the primary objective of control system design should be to achieve viability over a large and relevant set of theoretically viable states. For reasonable N, we believe N-step capturability analysis focuses on a large number of relevant states, without disregarding many of the theoretically viable states.

It may be beneficial to consider small N since analysis and control will be computationally less complex than for large N, and an analysis for large N focuses on uncommon and likely irrelevant states.To illustrate this point, consider a human

walking normally. It is hard to imagine a state in which the human can avoid falling, but will require more than 5 steps to stop. In fact, we hypothesize that nearly all human legged locomotion is in a 3-step viable capture basin and that all 3D bipedal robot locomotion demonstrated to date likely falls in a 2-step viable capture basin. Therefore, considering N-step capturability instead of full viability focuses on the least difficult states from which to avoid falling. For large N, it may be best to just take the fall and switch to an emergency falling controller to protect the legged system and surrounding environment. On the other hand, the ∞-step capture region also includes useful information as it specifies where not to step.

VI. Conclusion

Capturability analysis provides a unified and practical approach to analysis and control of legged systems. The concepts introduced in this paper apply to a wide range of legged systems, with any number of legs, including those with grasping end effectors. There are no assumptions of flat terrain, so unlike ZMP approaches, capturability analysis applies to rough terrain. The main strength of capturability analysis lies in the explicit focus on avoiding a fall in a global sense, instead of relying on indicators of local stability or small disturbance robustness. Furthermore, in addition to providing information on how close the robot is to falling, capturability analysis generates capture regions (Section III), which define appropriate foot placement for a given robot state, explicitly providing practical control information.

Acknowledgment

The authors would like to thank A. Goswami and E. Westervelt for their helpful comments on this work.

References

[1] K. Masani, A. H. Vette, and M. R. Popovic, "Controlling balance during quiet standing: Proportional and derivative controller generates preceding motor command to body sway position observed in experiments," *Gait & Posture*, vol. 23, no. 2, pp. 164–172, Feb. 2006.

[2] D. G. E. Hobbelen and M. Wisse, "A Disturbance Rejection Measure for Limit Cycle Walkers: The Gait Sensitivity Norm," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1213–1224, Dec. 2007.

[3] P.-B. Wieber, "On the stability of walking systems," in *Proc. Int. Workshop Humanoid Hum. Friendly Robot.*, Tsukuba, Japan, 2002, pp. 53–59.

[4] S. Kajita and K. Tanie, "Study of dynamic biped locomotion on rugged terrain—derivation and application of the linear inverted pendulum mode—," in *Proc. 1991 IEEE Int. Conf. Robot. Automat.*, vol. 2. IEEE Comput. Soc. Press, Apr. 1991, pp. 1405–1411.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *Proc. 2001 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, vol. 1, Oct./Nov. 2001, pp. 239–246.

[6] M. Vukobratovic and J. Stepanenko, "On the stability of anthropomorphic systems," *Math. Biosci.*, vol. 15, pp. 1–37, 1972.

[7] Y. Okumura, T. Tawara, K. Endo, T. Furuta, and M. Shimizu, "Realtime ZMP compensation for biped walking robot using adaptive inertia force control," in *Proc. 2003 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, vol. 1, Oct. 2003, pp. 335–339.

[8] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, J. Sola, E. Yoshida, N. Mansard, K. Yokoi, and J.-P. Laumond, "Reactive stepping to prevent falling for humanoids," in *2009 IEEE-RAS Int. Conf. Humanoid Rob.* IEEE, Dec. 2009, pp. 528–534.

[9] K. Nishiwaki and S. Kagami, "Strategies for Adjusting the ZMP Reference Trajectory for Maintaining Balance in Humanoid Walking," in *Proc. 2010 IEEE Int. Conf. Robot. Automat.* IEEE, May 2010, pp. 4230–4236.

[10] J. Morimoto, J. Nakanishi, G. Endo, G. Cheng, C. G. Atkeson, and G. Zeglin, "Poincaré-Map-Based Reinforcement Learning For Biped Walking," in *Proc. 2005 IEEE Int. Conf. Robot. Automat.* IEEE, Apr. 2005, pp. 2381–2386.

[11] D. G. E. Hobbelen and M. Wisse, "Limit Cycle Walking," in *Humanoid Robots, Human-like Machines*, M. Hackel, Ed. Vienna, Austria: ITech Education and Publishing, Jun. 2007, ch. 14, pp. 277–294.

[12] J. B. Dingwell, J. P. Cusumano, P. Cavanagh, and D. Sternad, "Local Dynamic Stability Versus Kinematic Variability of Continuous Overground and Treadmill Walking," *J. Biomech. Eng.*, vol. 123, no. 1, pp. 27–32, Feb. 2001.

[13] K. Byl and R. Tedrake, "Metastable walking on stochastically rough terrain," in *Proc. Robot.: Sci. Syst. IV*, Zurich, Switzerland, Jun. 2008.

[14] I. R. Manchester, U. Mettin, F. Iida, and R. Tedrake, "Stable dynamic walking over rough terrain: Theory and experiment," in *Proc. Int. Symp. Robot. Res.*, Aug./Sep. 2009.

[15] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-de-Wit, and J. W. Grizzle, "RABBIT: a testbed for advanced control theory," *IEEE Control Syst. Mag.*, vol. 23, no. 5, pp. 57–79, Oct. 2003.

[16] D. L. Wight, E. G. Kubica, and D. W. Wang, "Introduction of the Foot Placement Estimator: A Dynamic Measure of Balance for Bipedal Robotics," *J. Comput. Nonlinear Dynam.*, vol. 3, no. 1, p. 011009, 2008.

[17] J.-P. Aubin, *Viability theory*. Boston, MA: Birkhäuser, 1991.

[18] J. E. Pratt, J. Carff, S. V. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Proc. 2006 IEEE-RAS Int. Conf. Humanoid Rob.* IEEE, Dec. 2006, pp. 200–207.

[19] J. E. Pratt and R. Tedrake, "Velocity-based stability margins for fast bipedal walking," in *Fast Motions in Biomechanics and Robotics*, ser. Lecture Notes in Control and Information Sciences, M. Diehl and K. Mombaur, Eds. Springer Berlin Heidelberg, 2006, vol. 340, ch. 14, pp. 299–324.

[20] J. R. Rebula, F. Cañas, J. E. Pratt, and A. Goswami, "Learning capture points for humanoid push recovery," in *Proc. 2007 IEEE-RAS Int. Conf. Humanoid Rob.*, 2007.

[21] J. E. Pratt and B. T. Krupp, "Design of a bipedal walking robot," *Proc. 2008 SPIE*, vol. 69621, 2008.

[22] P.-B. Wieber, "Constrained dynamics and parametrized control in biped walking," in *Int. Symp. Math. Th. Netw. Syst.* Perpignan France: SIAM, 2000.

[23] M. Sipser, *Introduction to the Theory of Computation*. Course Technology, 2005.

[24] T. McGeer, "Passive walking with knees," in *Proc. 1990 Int. Conf. Robot. Automat.* Cincinnati, OH, USA: IEEE, May 1990, pp. 1640–1645.

[25] M. J. Coleman, M. Garcia, K. Mombaur, and A. Ruina, "Prediction of stable walking for a toy that cannot stand," *Phys. Rev. E*, vol. 64, no. 2, p. 022901, Jul. 2001.

[26] J.-P. Aubin, "A Survey of Viability Theory," *SIAM J. Control Optim.*, vol. 28, no. 4, pp. 749–788, 1990.

John Rebula received his S.B. degree in Mechanical Engineering from MIT in 2006, and is currently in the Ph.D. program in the Mechanical Engineering Department at the University of Michigan. He is interested in analysis, planning, and control of human and robotic locomotion.

Twan Koolen received his B.Sc. degree from the faculty of mechanical engineering at the Delft University of Technology in the Netherlands in 2009. He is currently pursuing an M.Sc. degree in mechanical engineering at the Delft Biorobotics Laboratory and working as a Research Associate at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include robustness in legged locomotion, multibody dynamics and robust control.

Tomas de Boer received the M.Sc. degree in mechanical engineering from Eindhoven University of Technology of, Eindhoven, The Netherlands, in 2006. He is currently working toward the Ph.D. degree with Delft University of Technology, Delft, The Netherlands. His current research interests include bioinspired robotics, legged locomotion, mechatronics, (multibody) dynamics and (compliant) control.

Jerry Pratt received his Ph.D. degree in Computer Science from MIT in 2000. He is currently a research scientist at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include the analysis and control of bipedal and quadrupedal walking and running, humanoid robots, and exoskeletons. Some of the robotic projects he has been involved in include Spring Turkey, Spring Flamingo, M2, the RoboKnee, tBot, LittleDog algorithms, M2V2, and the IHMC ROAM Exoskeleton.

Capturability-Based Analysis and Control of Legged Locomotion, Part 2: Application to Three Simple Gait Models

Twan Koolen, Tomas de Boer, John Rebula, Ambarish Goswami, Jerry Pratt

***Abstract*—This three-part paper discusses the analysis and control of legged locomotion in terms of N-step capturability: the ability of a legged system to come to a stop without falling by taking N or fewer steps. We consider this ability to be crucial to legged locomotion and a useful, yet not overly restrictive criterion for stability.**

Part 1 introduced our theoretical framework for assessing N-step capturability. In the current part, we use this framework to analyze three simple models of legged locomotion. All models are based on the 3D Linear Inverted Pendulum Model. The first model relies solely on point foot step location to maintain balance, the second model adds a finite-sized foot, and the third model additionally incorporates a reaction mass, enabling the use of centroidal angular momentum. We analyze how these three mechanisms influence N-step capturability, for N up to and including infinity. Part 3 will show that these results can be used to control a complex humanoid robot.

***Index Terms*—**

Capture point, Simplified gait models, Linear inverted pendulum model, Bipedal robots, Gait stability measure, Capturability.

I. Introduction

HUMANS are very adept at preventing falls during locomotion. We aim to integrate this remarkable ability in legged robots to enable their practical use in everyday environments. We propose to analyze the problem of disturbance rejection in terms of *N-step capturability*: the ability to come to a stop without falling by taking N steps or fewer.

In Part 1 of this three-part paper, N-step capturability was characterized using two concepts[1]:

1) *N-step viable-capture basin*: the set of states in which a hybrid dynamic system is N-step capturable; and
2) *N-step capture region*: the set of points to which a legged system in a given state can step to become $(N-1)$-step capturable, i.e., the set of all *N-step capture points*.

Manuscript received December 2010. This work was funded through the Army Tank and Automotive Research and Development Command, the Defense Advanced Research Projects Agency, the Office of Naval Research, NASA, and the Honda Research Institute.

T. Koolen and T. de Boer are with the Delft University of Technology, the Netherlands

J. Rebula is with the University of Michigan, Ann Arbor

A. Goswami is with the Honda Research Institute, California

J. Pratt and T. Koolen are with the Institute for Human and Machine Cognition, Florida

[1]See Part 1 for formal definitions.

Note that the N-step viable-capture basin specifies N-step capturability in terms of *state space*, whereas the N-step capture region does so in terms of *step locations*. In Part 1, we argued that viable-capture basins are more fundamental than capture regions, but that capture regions are more practical for designing control laws, since they directly provide useful information about where to step to prevent a fall.

Computing viable-capture basins and capture regions for a complex legged robot can be a challenging task. We are not aware of any better way to complete this task for a general robot than using computationally intensive model based methods, such as dynamic programming over the entire state and action space.

However, some linear models of legged locomotion permit simple and insightful ways of analyzing capturability. In this part of our three-part paper, we present three models for which it is possible to find viable-capture basins and capture regions in closed form. The results can be used as approximations for more complex legged systems and prove useful in their control, as we will demonstrate in Part 3 for a bipedal robot that has two 6-degree-of-freedom legs.

All three models presented are based on the 3D Linear Inverted Pendulum Model (3D-LIPM) [1], [2], which comprises a single point mass maintained on a plane by a variable-length leg link. The complexity of the presented models increases incrementally. To each subsequent model, another stabilizing mechanism is added. These mechanisms are generally considered fundamental in dealing with disturbances, both in the biomechanics and robotics literature [3]–[8].

The first model has a point foot. It relies solely on foot placement to come to a stop. The second model is obtained by adding a a finite-sized foot and ankle actuation to the first model, enabling modulation of the Center of Pressure (CoP). The third model extends the second by the addition of a reaction mass and hip actuation, enabling the human-like use of rapid trunk [3], [9] or arm motions [10], [11].

For each model, we assess N-step capturability for a subset of state space and compute N-step capture regions for N up to and including infinity. We compare results for these three models using two metrics. The first metric is the maximum distance between the foot's contact reference point and the *instantaneous capture point* (to be defined in Section II-D) for which the legged system is capturable. The second metric is the area of the ∞-step capture region for a given state. We will show that both metrics increase in magnitude as the number

of stabilizing mechanisms added to the 3D-LIPM increases. Furthermore, based on the incremental construction of the models, we will show the contribution of foot placement, CoP modulation, and angular momentum to capturability.

To illustrate the results obtained in this research, a Matlab graphical user interface (GUI) was created that allows the user to manipulate the control inputs for all models described in this paper, while the N-step capture regions are dynamically updated. The GUI is available online with the paper.

The remainder of this part is structured as follows. Section II presents the first model, the 3D-LIPM with point foot. The second model, called the 3D-LIPM with finite-sized foot, is presented in Section III, followed by the 3D-LIPM with finite-sized foot and reaction mass in Section IV. The models are compared in terms of capturability in Section V. Finally, the results are discussed in Section VI.

II. 3D-LIPM WITH POINT FOOT

The 3D Linear Inverted Pendulum Model, described by Kajita *et al.* [1], [2] and depicted in Fig. 1, comprises a point mass at the end of a telescoping massless mechanism (representing the leg), which is in contact with the ground. The point mass is kept on a horizontal plane by suitable generalized forces in the mechanism. Torques may be exerted at the base of the pendulum. For this first model, however, we set all torques at the base to zero. Hence, the base of the pendulum can be seen as a point foot. Foot position changes, which occur when a step is taken, are assumed instantaneous, and have no instantaneous effect on the position and velocity of the point mass.

Following the capture point framework introduced in Part 1, we consider the 3D-LIPM with point foot to be a *legged system*. The system's dynamics are derived in Section II-A. Its control input is the point foot position, which we will denote $\mathbf{r}_{\text{ankle}}$. We define a time-varying set of allowable values for this control input, described in Section II-B. The point foot is the only *legged system contact area*, and its corresponding *ground contact area* consists of all particles on the plane at $z = 0$. There is no *keep out region*. Changing the location of the point foot is considered taking a *step*. Since the point mass is unable to touch the ground and there is no keep out region, there are no *fallen states*. Moreover, the set of *failed states* is also defined to be empty. The point $\mathbf{r}_{\text{ankle}}$ will be the *contact reference point* for all models in this paper.

A. Equations of Motion

The equations of motion for the body mass are

$$m\ddot{\mathbf{r}} = \mathbf{f} + m\mathbf{g} \tag{1}$$

where m is the mass, $\mathbf{r} = \begin{pmatrix} x & y & z \end{pmatrix}^T$ is the position of the center of mass (CoM), expressed in an inertial frame, $\mathbf{f} = \begin{pmatrix} f_x & f_y & f_z \end{pmatrix}^T$ is the actuator force acting on the point mass and $\mathbf{g} = \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T$ is the gravitational acceleration vector.

A moment balance for the massless link shows that

$$-(\mathbf{r} - \mathbf{r}_{\text{ankle}}) \times \mathbf{f} = \mathbf{0} \tag{2}$$

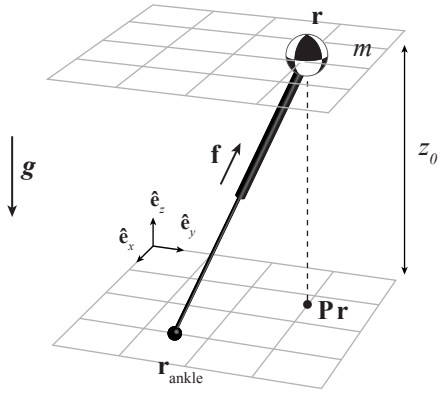

Fig. 1. Schematic representation of the 3D-LIPM with point foot. The model comprises a point foot at position $\mathbf{r}_{\text{ankle}}$, a point mass at position $\mathbf{r}$ with mass m and a massless telescoping leg link with an actuator that exerts a force $\mathbf{f}$ on the point mass that keeps it at constant height z_0. The projection matrix $\mathbf{P}$ projects the point mass location onto the xy-plane. The gravitational acceleration vector is $\mathbf{g}$.

where $\mathbf{r}_{\text{ankle}} = \begin{pmatrix} x_{\text{ankle}} & y_{\text{ankle}} & 0 \end{pmatrix}^T$ is the location of the ankle.

If $\dot{z} = 0$ initially, the point mass will stay at $z = z_0$ if $\ddot{z} = 0$. Using (1), we find $f_z = mg$. This can be substituted into (2) to find the forces f_x and f_y,

$$f_x = m\omega_0^2(x - x_{\text{ankle}})$$
$$f_y = m\omega_0^2(y - y_{\text{ankle}})$$

where $\omega_0 = \sqrt{\frac{g}{z_0}}$ is the reciprocal of the time constant for the 3D-LIPM.

The equations of motion, (1), can now be rewritten as

$$\ddot{\mathbf{r}} = \omega_0^2(\mathbf{P}\mathbf{r} - \mathbf{r}_{\text{ankle}}) \tag{3}$$

where $\mathbf{P} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{smallmatrix}\right)$ projects $\mathbf{r}$ onto the xy-plane.

Note that the equations of motion are linear. This linearity is what makes the model valuable as an analysis and design tool, as it allows us to make closed form predictions. In addition, the equations are decoupled and represent the same dynamics in the x- and y-directions. Each of the first two rows of (3) describes a separate 2D-LIPM with point foot. Therefore, results obtained for the 2D model can readily be extended to the 3D model.

B. Allowable Control Inputs

We introduce two constraints on the stepping capabilities of the model. First, we introduce an upper limit on step length, i.e., the distance between subsequent ankle locations. This maximum step length is denoted l_{max} and is assumed to be constant; it does not depend on the CoM location $\mathbf{r}$. Second, we introduce a lower limit to the time between ankle location changes, $\Delta t_{\text{s,min}}$, which models swing leg dynamics.

We define the following times and time intervals:

- $t_{\text{s,prev}}$: time at which the previous step was taken;
- t: current time;
- $\Delta t_{\text{s}} = t - t_{\text{s,prev}}$: time since the previous step;

- $\Delta t_{\text{s,min}}$: minimum time between steps.

Furthermore, we define G as the set of points that make up the ground. The set of allowable control inputs is then formally written as

$$U_{\text{ankle}} = \begin{cases} \{\mathbf{r}_{\text{ankle}}\}, & \Delta t_{\text{s}} < \Delta t_{\text{s,min}} \\ \{\mathbf{p} \in G \mid \|\mathbf{p} - \mathbf{r}_{\text{ankle}}\| \leq l_{\text{max}}\}, & \text{otherwise.} \end{cases} \tag{4}$$

That is, the new foot position must be the same as the current foot position if the minimum step time has not yet passed. Otherwise the new foot position may be chosen anywhere within the disk of radius l_{max} and center $\mathbf{r}_{\text{ankle}}$.

C. Dimensional Analysis

We perform a dimensional analysis to reduce the number of variables involved and to simplify subsequent derivations. Let us define dimensionless point mass position $\mathbf{r}'$, ankle (point foot) position $\mathbf{r}'_{\text{ankle}}$ and time t' as[2]

$$\mathbf{r}' = \frac{\mathbf{r}}{z_0} \qquad \mathbf{r}'_{\text{ankle}} = \frac{\mathbf{r}_{\text{ankle}}}{z_0} \qquad t' = \omega_0 t.$$

Throughout this paper, the dimensionless counterparts of all positions and lengths will be obtained by dividing by z_0, and times and time intervals will be nondimensionalized by multiplying by ω_0.

The dimensionless point mass position can be differentiated with respect to dimensionless time to obtain dimensionless velocity $\dot{\mathbf{r}}'$ and acceleration $\ddot{\mathbf{r}}'$:

$$\dot{\mathbf{r}}' = \frac{d}{dt'}\mathbf{r}' = \frac{\dot{\mathbf{r}}}{\omega_0 z_0}$$
$$\ddot{\mathbf{r}}' = \frac{d}{dt'}\dot{\mathbf{r}}' = \frac{\ddot{\mathbf{r}}}{\omega_0^2 z_0} = \frac{\ddot{\mathbf{r}}}{g}.$$

Using these dimensionless quantities, the equations of motion, (3), become

$$\ddot{\mathbf{r}}' = \mathbf{P}\mathbf{r}' - \mathbf{r}'_{\text{ankle}}. \tag{5}$$

Further derivations will be simplified by the absence of ω_0 in this equation, as compared to (3).

D. Instantaneous Capture Point

As a first step toward examining N-step capturability, we now introduce the *instantaneous capture point*. For the 3D-LIPM with point foot, it is the point on the ground that enables the system to come to a stop if it were to instantaneously place and maintain its point foot there. Although its definition is motivated by the current model, it will also be useful in the analysis of the other models presented in this part, and we consider it an important quantity to monitor even for more complex, real, legged systems.

Note that the instantaneous capture point is not necessarily a capture point. According to the definitions given in Part 1, capture points must be reachable, considering the dynamics and actuation limits, while the instantaneous capture point does not take into account the step time or step length constraints as defined in Section II-B.

[2]All dimensionless quantities will be marked with a prime.

The instantaneous capture point can be computed from energy considerations. For a given constant foot position, we can interpret the first two rows of (5) as the descriptions of two decoupled mass-spring systems, each with mass and negative stiffness of unit magnitude. Dimensionless *orbital energies* [1], [2], $E'_{\text{LIP},x}$ and $E'_{\text{LIP},y}$, are then defined as the Hamiltonians of these systems:

$$E'_{\text{LIP},x} = \frac{1}{2}\dot{x}'^2 - \frac{1}{2}(x' - x'_{\text{ankle}})^2 \tag{6a}$$
$$E'_{\text{LIP},y} = \frac{1}{2}\dot{y}'^2 - \frac{1}{2}(y' - y'_{\text{ankle}})^2. \tag{6b}$$

Since Hamiltonians are conserved quantities, so are the orbital energies.

The orbital energy for a direction determines the behavior of the 3D-LIPM in that direction when the CoM is moving toward the foot. Considering the x'-direction for example, three cases of interest arise:

1) $E'_{\text{LIP},x} > 0$. The orbital energy is sufficient to let x' reach x'_{ankle}, after which x' continues to accelerate away from x'_{ankle}.
2) $E'_{\text{LIP},x} < 0$. x' reverses direction before x' reaches x'_{ankle}.
3) $E'_{\text{LIP},x} = 0$. x' comes to a rest exactly at x'_{ankle}.

We can solve for a foot location that results in either desired orbital energies or, equivalently, a desired velocity vector at a given value of $\mathbf{r}'$ [1], [2]. To determine the instantaneous capture point, we are interested in the foot placement required to obtain zero orbital energy in each direction. Solving (6) for $\mathbf{r}'_{\text{ankle}}$ and choosing the solution for which the point mass moves toward the point foot shows that the dimensionless version of the instantaneous capture point [12] is

$$\mathbf{r}'_{\text{ic}} = \mathbf{P}\mathbf{r}' + \dot{\mathbf{r}}' \tag{7}$$

or, in terms of the original physical quantities:

$$\mathbf{r}_{\text{ic}} = \mathbf{P}\mathbf{r} + \frac{\dot{\mathbf{r}}}{\omega_0}. \tag{8}$$

This quantity was independently described by Hof et al. [13]–[15] and named the Extrapolated Center of Mass. It was shown to have significant ties to balancing and walking in human test subjects.

E. Instantaneous Capture Point Dynamics

If the point foot is not instantaneously placed at the instantaneous capture point, the instantaneous capture point will move. We will now analyze this motion. The results of this analysis are depicted graphically in Fig. 2. The dynamics that describe the motion of the instantaneous capture point on the ground can be derived by reformulating the dimensionless equations of motion in state space form. The state space model is based on the x'-dynamics only (i.e., the first row of (5), a 2D-LIPM), but the derivations can readily be extended to both directions, as noted in Section II-A. The first row of (5) is rewritten in state space form as

$$\begin{pmatrix} \dot{x}' \\ \ddot{x}' \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} x' \\ \dot{x}' \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ -1 \end{pmatrix}}_{\mathbf{B}} x'_{\text{ankle}}. \tag{9}$$

The state matrix $\mathbf{A}$ has eigenvalues

$$\lambda_{1,2} = \pm 1$$

and corresponding eigenvectors

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The eigendata show that there is a saddle point with one stable and one unstable eigenvector. The state matrix can be diagonalized using the similarity transformation $\mathbf{T} = \mathbf{V}^{-1}$, which results in the new state vector

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}}_{\mathbf{T}} \begin{pmatrix} x' \\ \dot{x}' \end{pmatrix}. \tag{10}$$

The new state x'_1 is identical to the instantaneous capture point x'_{ic}, and x'_2 is the point reflection of the instantaneous capture point across the projection of the point mass onto the ground. The diagonalized state space model is

$$\begin{pmatrix} \dot{x}'_1 \\ \dot{x}'_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}}_{\mathbf{TAT}^{-1}} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} + \underbrace{\begin{pmatrix} -1 \\ 1 \end{pmatrix}}_{\mathbf{TB}} x'_{\text{ankle}}. \tag{11}$$

The diagonal state matrix TAT^{-1} shows that the model's instantaneous capture point dynamics are first order. State $x'_1 = x'_{\text{ic}}$ corresponds to the unstable eigenvalue $+1$ and is thus of primary interest in stabilizing the system.

These derivations can be repeated for the y'-direction, so that the first row of (11) can be extended to

$$\dot{\mathbf{r}}'_{\text{ic}} = \mathbf{r}'_{\text{ic}} - \mathbf{r}'_{\text{ankle}}. \tag{12}$$

This derivation proves the following Theorem:

Theorem 1: The instantaneous capture point moves on the line through the point foot and itself, away from the point foot, at a velocity proportional to its distance from the point foot.

As the instantaneous capture point moves away from the foot, its velocity increases exponentially. Fig. 2 shows the motion of both the instantaneous capture point and the point mass when the point foot is kept fixed. Note that the projection of the point mass onto the xy-plane describes a hyperbolic curve, as shown in [2].

An explicit formulation of the instantaneous capture point trajectory for a fixed foot position is found by solving (12):

$$\mathbf{r}'_{\text{ic}}(\Delta t') = [\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}]e^{\Delta t'} + \mathbf{r}'_{\text{ankle}}. \tag{13}$$

This equation will prove useful, both in determining whether a state is N-step capturable and in computing N-step capture regions.

F. Capturability

The instantaneous capture point is now used to determine N-step capturability for the 3D-LIPM with point foot in state space. N-step capturability can be described using N-step viable-capture basins. Computing these viable-capture basins requires us to examine all of state space and determine which states are N-step capturable. Although this process is possible, in this paper we will only examine a part of state space that we consider interesting. The reasons for this choice are brevity and clarity of presentation and because only these parts of the state space need to be considered to compute the N-step capture regions and the capturability measures proposed in the introduction. For the current model in particular, we will only consider those states for which the legged system has just taken a step. The step time will be marked as time zero: $t' = t'_{\text{s,prev}} = 0$.

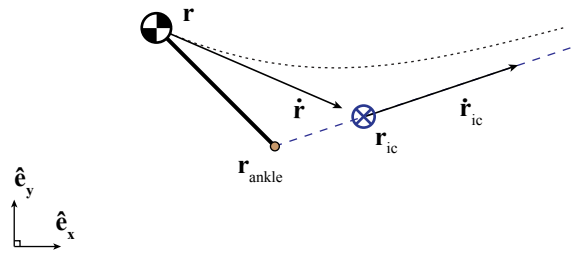

Fig. 2. Top view of the 3D-LIPM with point foot for a given initial state at time t. By adding the CoM velocity vector $\dot{\mathbf{r}}$ (divided by ω_0, see (8)) to the projected CoM position $\mathbf{Pr}$, we find the instantaneous capture point location $\mathbf{r}_{\text{ic}}$. The future trajectories of the point mass and the instantaneous capture point are along the dotted lines for a constant foot location $\mathbf{r}_{\text{ankle}}$. For this figure, $\mathbf{Pr} = [-0.4, 0.4, 0]$, $\dot{\mathbf{r}} = [0.7, -0.3, 0]$, $\mathbf{r}_{\text{ankle}} = [0, 0, 0]$, and model parameters z_0, m and g are all set to unit magnitude.

For the 3D-LIPM with point foot, N-step capturability at $t' = t'_{\text{s,prev}}$ can be fully described in terms of the initial distance between the contact reference point (the point foot) and the instantaneous capture point, $\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\|$. The maximum distance for which the state is still N-step capturable will be denoted d'_N. Fig. 3 shows an evolution that captures the legged system in the minimum number of steps and the values of d'_N for five values of N. The procedure for computing d'_N will now be given.

The requirement for 0-step capturability may be readily determined: the definition of the instantaneous capture point shows that the legged system is 0-step capturable if and only if the instantaneous capture point coincides with the point foot location. The requirement for 0-step capturability is thus $\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\| \leq d'_0$, with $d'_0 = 0$ for this model.

For higher N, the goal is to reach an $(N-1)$-step capturable state using a 1-step evolution. This is only possible if the distance between the foot and the instantaneous capture point, evaluated at the earliest possible step time ($\Delta t_{\text{s,min}}$) is such that there exists a step of allowable length that makes the legged system $(N-1)$-step capturable:

$$\left\|\mathbf{r}'_{\text{ic}}(\Delta t'_{\text{s,min}}) - \mathbf{r}'_{\text{ankle}}\right\| - l'_{\text{max}} \leq d'_{N-1}. \tag{14}$$

Using (13), this can be rewritten as

$$\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\| \leq (d'_{N-1} + l'_{\text{max}})e^{-\Delta t'_{\text{s,min}}} = d'_N \tag{15}$$

which leads to a recursive expression for d'_N:

$$d'_N = (d'_{N-1} + l'_{\text{max}})e^{-\Delta t'_{\text{s,min}}}, \quad d'_0 = 0. \tag{16}$$

The maximum distance for N-step capturability, d'_N, follows a converging geometric series:

$$d'_{N+1} - d'_N = (d'_N - d'_{N-1})e^{-\Delta t'_{\text{s,min}}}, \quad \forall N \geq 1.$$

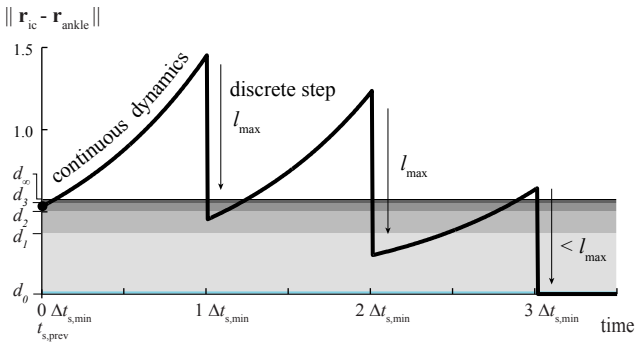

Fig. 3. N-step capturability for the 3D-LIPM with point foot, characterized using the values of d'_N (shown for $N \in \{0 \dots 3, \infty\}$). d'_N is the maximum distance between the instantaneous capture point and the contact reference point, evaluated at step time, for which the legged system is N-step capturable. The contact reference point is $\mathbf{r}_{\text{ankle}}$, the ankle (point foot) position. An example initial state at $t_{\text{s,prev}}$ is shown, which lies within the 3-step viable-capture basin. Note that this state is different from the state depicted in Fig. 2. Because the initial state corresponding to $t_{\text{s,prev}}$ is not 1-step capturable, the distance between the ankle and instantaneous capture point at $t_{\text{s,prev}} + \Delta t_{\text{s,min}}$ is larger than l_{max}. A first step of length l_{max} towards the instantaneous capture point results in a discrete jump in the distance between the instantaneous capture point and the ankle. A second step is required to make the state 1-step capturable, and a third step is required to reach a captured state. For this figure, $\Delta t_{\text{s,min}}$ and l_{max} are set to unit magnitude.

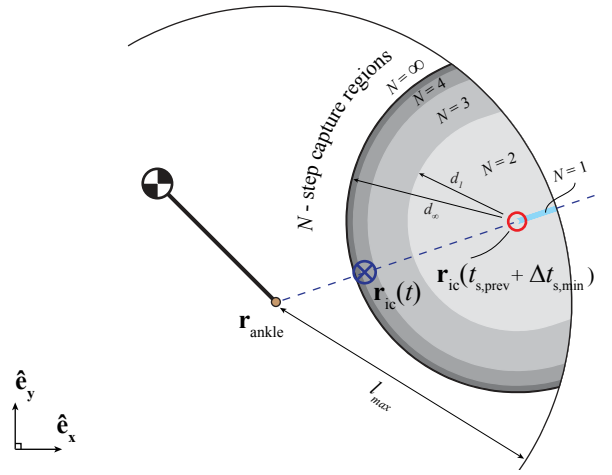

Fig. 4. Top view of the 3D-LIPM with point foot and N-step capture regions, for the same state as shown in Fig. 2. Additional to the information in Fig. 2, this figure gives a schematic representation of the N-step capture regions for $N \in \{1 \dots 4, \infty\}$. Before the first step, the instantaneous capture point $\mathbf{r}_{\text{ic}}$ will move away from the point foot, $\mathbf{r}_{\text{ankle}}$, on the dashed line. The instantaneous capture point location at the earliest time at which a step may be taken, $\mathbf{r}_{\text{ic}}(t_{\text{s,prev}} + \Delta t_{\text{s,min}})$, is depicted by a small circle. The set of possible future instantaneous capture point locations for which the minimum step time has passed is the ray starting at the small circle and pointing along the dashed line, away from the point foot. An N-step capture region is then found using this ray and the value of d_{N-1}: all points within a distance of d_{N-1} to the ray are N-step capture points, as long as they lie inside the maximum step length circle, which has radius l_{max}. For this figure, model parameters $\Delta t_{\text{s,min}}$ and l_{max} are set to unit magnitude.

The ratio of the geometric series, $e^{-\Delta t'_{\text{s,min}}}$, can be interpreted as a measure of the mobility of the legged system. The ratio is a dimensionless quantity that takes a value in the interval $[0, 1)$ if the minimum step time is strictly positive. Hence, the series d'_N converges. Moreover, notice that being allowed to take more steps to come to a stop suffers from diminishing returns. The nature of the series allows the requirement for ∞-step capturability to be computed in closed form:

$$d'_\infty = d'_0 + \sum_{N=0}^{\infty} \left[d'_{N+1} - d'_N \right] \tag{17a}$$

$$= l'_{\text{max}} \frac{e^{-\Delta t'_{\text{s,min}}}}{1 - e^{-\Delta t'_{\text{s,min}}}} \tag{17b}$$

since $d'_0 = 0$ for the 3D-LIPM with point foot. The obtained values of d'_N will be used to determine N-step capture regions.

G. Capture Regions

The N-step capture regions for the 3D-LIPM with point foot are shown in Fig. 4 for an example state. How these regions are computed is described next.

The legged system will come to a stop if it puts its CoP onto the instantaneous capture point by stepping there. However, stepping is only possible after the minimum step time has passed. Hence, we first determine the set of possible future instantaneous capture point locations that satisfy the minimum step time constraint. The point in this set that is closest to the foot is the (future) instantaneous capture point location evaluated at the earliest possible step time, $\mathbf{r}_{\text{ic}}(t_{\text{s,prev}} + \Delta t_{\text{s,min}})$. This point is found using (13). If no step is taken, then the instantaneous capture point will just keep moving farther away from the point foot, as shown by Theorem 1. Therefore, the set of possible future instantaneous capture point locations is a ray starting at $\mathbf{r}_{\text{ic}}(t_{\text{s,prev}} + \Delta t_{\text{s,min}})$ which points away from $\mathbf{r}_{\text{ankle}}$ (see Section II-E).

N-step capture regions for $N \in [1, \infty]$ can now be found using this ray and the expression for d'_N in (16). After taking a single step to an N-step capture point, the legged system's state should be $(N-1)$-step capturable. Step locations that put the legged system in such a state are readily found using (16): all points within a distance of d'_{N-1} to a possible future instantaneous capture point are N-step capture points, provided that the legged system can reach those points given the maximum step length constraint.[3] This results in the nested regions depicted in Fig. 4.

Note that finding the 1-step capture region is especially simple. Since $d'_0 = 0$, the step of finding points with distance d'_{N-1} to the ray simply results in the ray itself. The 1-step capture region is then the part of the ray that is inside the maximum step length circle.

III. 3D-LIPM WITH FINITE-SIZED FOOT

In this section, we extend the 3D-LIPM with point foot by making the foot size finite. The finite-sized foot articulates with the leg at a 2-DoF ankle joint, and is assumed massless. At the ankle, torques may be applied in the pitch and roll directions. However, the torques are limited in such a way that the foot does not start to rotate with respect to the ground. The foot orientation (which is the yaw direction, rotation about the z-axis) may be chosen arbitrarily when a step is taken. The model is shown in Fig. 5.

[3] A point that cannot be reached can never be an N-step capture point.

A. Equations of Motion

Only slight modifications to the derivation of the equations of motion for the 3D-LIPM are necessary. Equation (1) also applies to this model. Adding controllable ankle torques $\tau_{\text{ankle},x}$ and $\tau_{\text{ankle},y}$ and a reaction torque $\tau_{\text{ankle},z}$ changes the moment balance of the massless leg link, (2), to

$$-(\mathbf{r}-\mathbf{r}_{\text{ankle}})\times\mathbf{f}+\boldsymbol{\tau}_{\text{ankle}}=\mathbf{0} \tag{18}$$

where $\boldsymbol{\tau}_{\text{ankle}} = \begin{pmatrix}\tau_{\text{ankle},x} & \tau_{\text{ankle},y} & \tau_{\text{ankle},z}\end{pmatrix}^T$ is the ankle torque and $\mathbf{r}_{\text{ankle}}$ is now the projection of the ankle joint onto the ground.

As before, $f_z = mg$ due to the model constraint $\ddot{z}=0$, and we find the actuator forces f_x, f_y and the reaction torque $\tau_{\text{ankle},z}$ from (18):

$$\begin{aligned} f_x &= m\omega_0^2(x-x_{\text{ankle}})+\frac{\tau_{\text{ankle},y}}{z_0}\\ f_y &= m\omega_0^2(y-y_{\text{ankle}})-\frac{\tau_{\text{ankle},x}}{z_0}\\ \tau_{\text{ankle},z} &= -\frac{\tau_{\text{ankle},x}}{z_0}(x-x_{\text{ankle}})-\frac{\tau_{\text{ankle},y}}{z_0}(y-y_{\text{ankle}}). \end{aligned}$$

The equations of motion can then be derived by substituting this into (1), resulting in

$$\ddot{\mathbf{r}}=\omega_0^2(\mathbf{Pr}-\mathbf{r}_{\text{CoP}}) \tag{19}$$

where

$$\mathbf{r}_{\text{CoP}}=\mathbf{r}_{\text{ankle}}-\frac{1}{mg}\begin{pmatrix}\tau_{\text{ankle},y}\\-\tau_{\text{ankle},x}\\0\end{pmatrix}=\mathbf{r}_{\text{ankle}}-\frac{\boldsymbol{\tau}_{\text{ankle}}\times\hat{\mathbf{e}}_z}{mg}$$

is the location of the CoP. This follows readily from a moment balance for the foot, considering that the ankle torques are such that the foot does not rotate with respect to the ground, by model definition.

Comparing (19) to (3) clearly shows that the dynamics are essentially unchanged. The only difference is that it is now possible to displace the CoP without taking a step. Hence, the results of Section II-E are still valid if $\mathbf{r}_{\text{ankle}}$ is replaced by $\mathbf{r}_{\text{CoP}}$.

The dynamics of our 3D-LIPM with finite-sized foot are the same as those of the original 3D-LIPM by Kajita *et al.* [2], where the 3D-LIPM's virtual inputs are interpreted as ankle torques, expressed in a ground-fixed frame.

B. Allowable Control Inputs

The allowable control inputs as defined for the 3D-LIPM with point foot in Section II-B also apply to the 3D-LIPM with finite-sized foot[4]. We augment these allowable control inputs by specifying limits on the ankle torques. The allowable ankle torques are easiest to describe in terms of their resulting CoP location, i.e., as $\mathbf{r}_{\text{CoP}} \in U_{\text{CoP}}$.

We define the base of support $B(t)$ as the (time-varying) set of points on the ground where the CoP may be set instantaneously by applying suitable ankle torques, without taking a step. The base of support coincides with the convex hull of the points on the ground that are in contact with the foot. To fulfill the requirement that the foot must not rotate, the CoP location, which keeps the foot in static equilibrium (i.e., the Foot Rotation Indicator [16]), must not leave the base of support. Mathematically, this requirement gives:

$$U_{\text{CoP}}=B(t). \tag{20}$$

The foot orientation may be chosen without restriction when a step is taken.

[4]Note that the ankle location is still used as the reference point for determining step length.

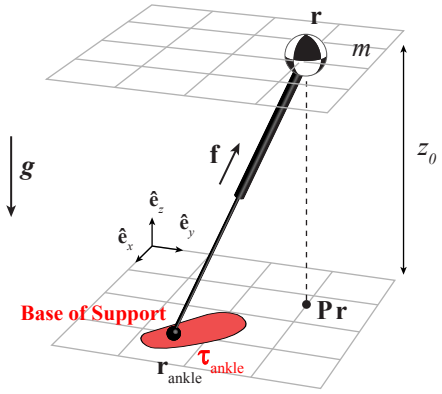

Fig. 5. The 3D-LIPM with finite-sized foot. The 3D-LIPM with point foot (Fig. 1) is extended with a a finite-sized foot and the ability to apply ankle torques $\boldsymbol{\tau}_{\text{ankle}}$ to obtain the 3D-LIPM with finite-sized foot.

C. Dimensional Analysis

In addition to the dimensionless quantities defined for the 3D-LIPM with point foot in Section II-C, we define dimensionless ankle torque $\boldsymbol{\tau}'_{\text{ankle}}$ as

$$\boldsymbol{\tau}'_{\text{ankle}}=\frac{\boldsymbol{\tau}_{\text{ankle}}}{m\omega_0^2 z_0^2}.$$

The dimensionless counterpart of the CoP is then

$$\mathbf{r}'_{\text{CoP}}=\frac{\mathbf{r}_{\text{CoP}}}{z_0}=\mathbf{r}'_{\text{ankle}}-\boldsymbol{\tau}'_{\text{ankle}}\times\hat{\mathbf{e}}_z$$

and the equations of motion reduce to

$$\ddot{\mathbf{r}}'=\mathbf{Pr}'-\mathbf{r}'_{\text{CoP}}. \tag{21}$$

D. Equivalent Constant CoP

To find the capture region for this model, the effect of a time-varying CoP must be investigated.

Suppose a time-varying CoP causes the instantaneous capture point to move from an initial position to a final position in a certain time interval. The *equivalent constant CoP* is the point where the CoP could have been held constant, while that CoP would still move the instantaneous capture point from the initial position to the final position in the same time interval.[5]

We can use (13) to compute the equivalent constant CoP as

$$\mathbf{r}'_{\text{CoP,eq}}=\frac{\mathbf{r}'_{\text{ic}}(\Delta t')-\mathbf{r}'_{\text{ic}}(0)e^{\Delta t'}}{1-e^{\Delta t'}} \tag{22}$$

[5]The equivalent constant CoP is only equivalent in terms of instantaneous capture point position and not necessarily in terms of other parts of the state.

Let us now examine the equivalent constant CoP for a discrete CoP change. Suppose the CoP is initially located at $\mathbf{r}'_{\text{CoP},0}$, and is kept there for $\Delta t'_0$. Subsequently, it is changed to $\mathbf{r}'_{\text{CoP},1}$ and kept there for $\Delta t'_1$. The final instantaneous capture point position is found by applying (13) twice:

$$\begin{aligned}\mathbf{r}'_{\text{ic}}(\Delta t'_0) &= [\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{CoP},0}]e^{\Delta t'_0} + \mathbf{r}'_{\text{CoP},0} \\ \mathbf{r}'_{\text{ic}}(\Delta t'_0 + \Delta t'_1) &= [\mathbf{r}'_{\text{ic}}(\Delta t'_0) - \mathbf{r}'_{\text{CoP},1}]e^{\Delta t'_1} + \mathbf{r}'_{\text{CoP},1}\end{aligned} \tag{23}$$

Solving (22) and (23) for $\mathbf{r}'_{\text{CoP,eq}}$ (with $\Delta t' = \Delta t'_0 + \Delta t'_1$), we find

$$\mathbf{r}'_{\text{CoP,eq}} = (1-w)\mathbf{r}'_{\text{CoP},0} + w\mathbf{r}'_{\text{CoP},1} \tag{24}$$

where

$$w = \frac{e^{\Delta t'_1} - 1}{e^{\Delta t'_0 + \Delta t'_1} - 1}$$

The scalar w lies in the interval $[0, 1]$ because both $\Delta t'_0$ and $\Delta t'_1$ are nonnegative. The equivalent constant CoP is thus a weighted average of the two individual CoPs, where the weighting factors depend only on the time intervals. This statement can be generalized to any number of CoP changes and, in the limit, even to continuously varying CoPs, thus proving the following Theorem:

Theorem 2: For the 3D-LIPM with finite-sized foot, the equivalent constant CoP is a weighted average of the CoP as a function of time.

The time-varying CoP must always be inside the base of support, which is a convex set. By definition, a weighted average of elements of a convex set must also be in the convex set. Therefore:

Corollary 1: If the base of support of the 3D-LIPM with finite-sized foot is constant, then the equivalent constant CoP for any realizable instantaneous capture point trajectory lies within the base of support.

Theorem 2 and Corollary 1 greatly simplify the analysis of capturability and capture regions, since only constant CoP positions within the base of support have to be considered in our subsequent derivations.

Equation (24) reveals some interesting properties of computing the equivalent constant CoP for a piecewise constant CoP trajectory:

- distributivity over addition: adding a constant offset to the individual CoP locations results in an equivalent constant CoP that is offset by the same amount;
- associativity: when computing the equivalent constant CoP for a sequence of three individual CoP locations, the order of evaluation of the composition does not matter;
- non-commutativity: when computing an equivalent constant CoP for a sequence of individual CoP locations, the order of the sequence being composed does matter.

E. Capturability

The concepts of instantaneous capture point and equivalent constant CoP are now used to determine capturability for the 3D-LIPM with finite-sized foot. We first analyze 0-step capturability.

We are allowed to replace point foot position by CoP in Theorem 1 because the dynamics are essentially the same. Hence, the instantaneous capture point diverges away from the CoP. Since the base of support is a convex set and is assumed to not change without taking a step, a Corollary of that Theorem is:

Corollary 2: Once the instantaneous capture point of the 3D-LIPM with finite-sized foot is outside the base of support, it is impossible to move it back inside without taking a step.

Since a captured state can only be reached when the CoP can be made to coincide with the instantaneous capture point, Corollary 2 provides us with a test of whether a state is 0-step capturable:

Corollary 3: The 3D-LIPM with finite-sized foot is 0-step capturable if and only if the instantaneous capture point is inside the base of support.

For higher N, capturability is analyzed in much the same way as for the 3D-LIPM with point foot. Like before, we will not compute complete N-step viable-capture basins. For this model we restrict the analysis to states at which a step has just been taken, for the same reasons as mentioned in Section II-F, and for which the foot is optimally oriented, which will be defined below. For these states, capturability can again be expressed in terms of the distance $\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\|$, just like in Section II-F.

The strategy that brings the legged system to a halt in as few steps as possible comprises stepping as soon as possible in the direction of the instantaneous capture point, with the foot optimally oriented and the CoP always maintained as close to the instantaneous capture point as possible. With "optimally oriented" we mean that the distance between the border of the base of support and the instantaneous capture point is minimized, given a fixed ankle location. This means that the CoP is located at a point on the edge of the base of support that has the greatest distance to the ankle and will be closest to the instantaneous capture point. The greatest distance between the ankle and the edge of the base of support will be denoted r_{max} and is normalized as $r'_{\text{max}} = r_{\text{max}}/z_0$.

Due to the assumption of optimal foot orientation, the requirement for 0-step capturability becomes $\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\| \leq d'_0$, with $d'_0 = r'_{\text{max}}$. Similar to Section II-F, we start at (14) and arrive at formulas for d'_N and d'_∞:

$$d'_N = (l'_{\text{max}} - r'_{\text{max}} + d'_{N-1})e^{-\Delta t'_{\text{s,min}}} + r'_{\text{max}}, \quad \forall N \geq 1 \tag{25}$$

$$d'_\infty = l'_{\text{max}} \frac{e^{-\Delta t'_{\text{s,min}}}}{1 - e^{-\Delta t'_{\text{s,min}}}} + r'_{\text{max}}. \tag{26}$$

It is seen that the difference between d'_∞ for the model with point foot and d'_∞ for the model with finite-sized foot is simply the normalized maximum distance between the contact reference point and the edge of the foot, r'_{max}.

F. Capture Regions

The N-step capture regions for the 3D-LIPM with finite-sized foot are shown in Fig. 6.

The first step in computing N-step capture regions is again to find the set of all points where the instantaneous capture point can be after the minimum step time has passed. Equation (12) shows that the instantaneous capture point can only be pushed in certain directions: the set of possible instantaneous

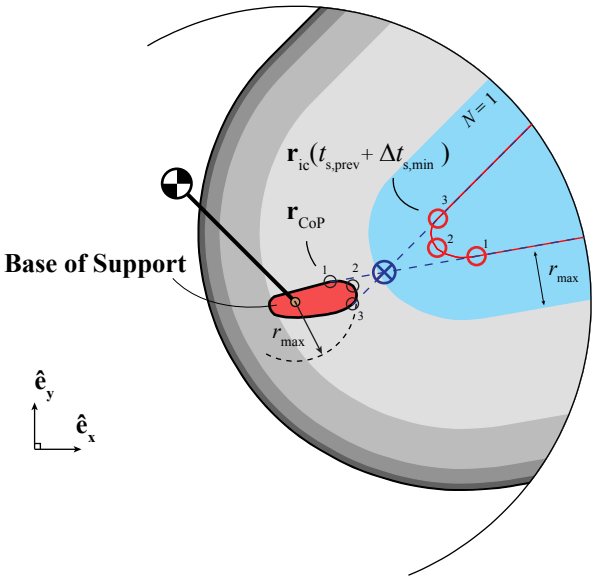

Fig. 6. Top view of the 3D-LIPM with finite-sized foot, with a schematic representation of the N-step capture regions. The figure is an extension of Fig. 4: state parameters $\mathbf{r}_{\text{ankle}}$, $\mathbf{r}$ and $\dot{\mathbf{r}}$ are identical. We have omitted labels that were already shown in Fig. 4 to avoid cluttering. CoP locations 1 and 3 are just in line of sight of $\mathbf{r}_{\text{ic}}(t)$ and determine to which locations the instantaneous capture point may be directed by CoP modulation (dashed lines). CoP location 2 is closest to $\mathbf{r}_{\text{ic}}(t)$ and results in the closest possible instantaneous capture point at minimum step time, $\mathbf{r}_{\text{ic}}(t_{\text{s,prev}}+\Delta t_{\text{s,min}})$. The near boundary of the set of all possible instantaneous capture point locations at minimum step time is a scaled point reflection of the base of support across the instantaneous capture point, as demonstrated by example CoP locations 1 to 3 and corresponding capture point locations 1 to 3. To obtain the N-step capture regions, the locations of all possible instantaneous capture points bounded by $\Delta t_{\text{s,min}}$ are surrounded by bands of width d'_N, given by (25). For this figure, $r_{\text{max}} = 0.2$.

capture point locations is bounded by the 'lines of sight' from the instantaneous capture point to the base of support (shown as dashed lines in Fig. 6), resulting in a wedge-shaped region.

The effect of the minimum step time constraint is again evaluated by considering where the instantaneous capture point can be located at the minimum step time if the CoP may be placed anywhere inside the base of support. Theorem 2 and Corollary 1 reveal that only constant CoP positions within the base of support need to be considered. The near boundary of the set of possible instantaneous capture point locations is found by evaluating (13) as before, using CoP locations that lie along the part of the boundary of the base of support that 'can be seen' from the instantaneous capture point. The near boundary turns out to be a scaled point reflection of this part of the base of support across the instantaneous capture point. The scaling factor depends on the minimum step time only.

To find N-step capture regions, we follow the same procedure as in Section II-G, *i.e.*, we create nested regions around the set of possible instantaneous capture point locations. This time, the greatest allowed distance to the possible instantaneous capture point locations is computed using (25). This method assumes that the foot will be oriented optimally when the step is taken. Note that for this model, $d'_0 = r'_{\text{max}} > 0$. Of course, points that are outside the maximum step length circle need to be discarded.

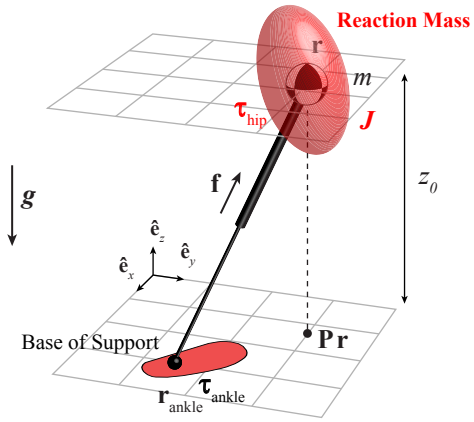

Fig. 7. The 3D-LIPM with finite-sized foot and reaction mass. The 3D-LIPM with finite-sized foot (Fig. 5) is extended with a non-zero mass moment of inertia tensor $\mathbf{J}$ and the ability to apply hip torques $\boldsymbol{\tau}_{\text{hip}}$ to obtain the 3D-LIPM with finite-sized foot and reaction mass.

IV. 3D-LIPM WITH FINITE-SIZED FOOT AND REACTION MASS

We now extend the 3D-LIPM with finite-sized foot by modeling not just a point 'body' mass at the end of the leg, but a rigid body possessing a non-zero mass moment of inertia. Actuators in the hip can exert torques on this reaction mass in all directions, enabling lunging motions in 3D. The model, depicted in Fig. 7, is a 3D version of the Linear Inverted Pendulum plus Flywheel Model presented in [17]. It can also be considered a linear version of the Reaction Mass Pendulum [18] with a constant mass moment of inertia.

To make the analysis tractable, we specify several constraints. We place limits on the allowable angles of the reaction mass. At the start of our analysis, we assume that all angles and angular velocities are zero. Hip torques can be used to accelerate the reaction mass, followed by decelerating torques to prevent the reaction mass from exceeding angle limits. Furthermore, we assume that the robot can only lunge once, in only one direction, similar to a human using a single impulsive lunging response in an attempt to regain balance after a severe perturbation. Besides angle limits, we place limits on the allowable hip torque. The hip torque component around the z-axis is determined by the requirement of no yaw of the reaction mass. This requirement makes the equations of motion linear. For the horizontal torque components, we assume a bang-bang input profile, as used in [6], [19]. We require that the execution time of the profile is less than the minimum step time.

A. Equations of Motion

The equations of motion for the reaction mass are

$$m\ddot{\mathbf{r}} = \mathbf{f} + m\mathbf{g} \tag{27a}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}_{\text{hip}} - \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) \tag{27b}$$

where $\boldsymbol{\omega} = \begin{pmatrix}\omega_x & \omega_y & \omega_z\end{pmatrix}^T$ is the angular velocity vector of the upper body, expressed in the inertial reference frame, $\boldsymbol{\tau}_{\text{hip}} = \begin{pmatrix}\tau_{\text{hip},x} & \tau_{\text{hip},y} & \tau_{\text{hip},z}\end{pmatrix}^T$ is the hip torque vector, $\mathbf{J}$ is the mass moment of inertia in the body-fixed frame, and m, $\mathbf{r}$, $\mathbf{f}$ and $\mathbf{g}$ are as defined in Section II-A.

Assuming that $\omega_z = 0$, that $\tau_{\text{hip},z}$ is such that $\dot{\omega}_z = 0$, and that $\mathbf{J}$ is diagonal, (27b) can be rewritten as

$$\begin{aligned} J_{xx}\dot{\omega}_x &= \tau_{\text{hip},x} \\ J_{yy}\dot{\omega}_y &= \tau_{\text{hip},y} \\ 0 &= \tau_{\text{hip},z} + (J_{xx} - J_{yy})\omega_x\omega_y. \end{aligned}$$

This last equation specifies the hip torque about the z-axis that is required to keep the reaction mass from yawing. Note that no hip torque about the z-axis is required if $J_{xx} = J_{yy}$.

The moment balance for the massless leg link is

$$-(\mathbf{r} - \mathbf{r}_{\text{ankle}}) \times \mathbf{f} - \boldsymbol{\tau}_{\text{hip}} + \boldsymbol{\tau}_{\text{ankle}} = \mathbf{0}. \tag{28}$$

Keeping the mass at $z = z_0$ means that $f_z = mg$, as before. This fact and (28) can be used to find the reaction forces f_x and f_y, and the ankle torque $\tau_{\text{ankle},z}$:

$$\begin{aligned} f_x &= m\omega_0^2(x - x_{\text{CoP}}) - \frac{\tau_{\text{hip},y}}{z_0} \\ f_y &= m\omega_0^2(y - y_{\text{CoP}}) + \frac{\tau_{\text{hip},x}}{z_0} \\ \tau_{\text{ankle},z} &= \frac{\tau_{\text{hip},x} - \tau_{\text{ankle},x}}{z_0}(x - x_{\text{ankle}}) \\ &\quad + \frac{\tau_{\text{hip},y} - \tau_{\text{ankle},y}}{z_0}(y - y_{\text{ankle}}) + \tau_{\text{hip},z}. \end{aligned}$$

We can now rewrite (27) to obtain the equations of motion,

$$\ddot{\mathbf{r}} = \omega_0^2(\mathbf{P}\mathbf{r} - \mathbf{r}_{\text{CMP}}) \tag{29a}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\mathbf{P}\boldsymbol{\tau}_{\text{hip}} \tag{29b}$$

where $\mathbf{r}$, ω_0 and $\mathbf{P}$ are as defined in Section II-A, and

$$\mathbf{r}_{\text{CMP}} = \mathbf{r}_{\text{CoP}} + \frac{1}{mg}\begin{pmatrix} \tau_{\text{hip},y} \\ -\tau_{\text{hip},x} \\ 0 \end{pmatrix} = \mathbf{r}_{\text{CoP}} + \frac{\boldsymbol{\tau}_{\text{hip}} \times \hat{\mathbf{e}}_z}{mg} \tag{30}$$

is the Centroidal Moment Pivot (CMP) as defined in [20]. Here we have used the fact that the CoP is equal to the Zero Moment Point when the ground is flat and horizontal [20].

The equations of motion are again linear. Note the similarity to the equations of motion for the previous models, which allows us to reuse most results obtained for the previous models.

B. Allowable Control Inputs

The actuation limits of the 3D-LIPM with finite-sized foot are extended to include the hip torque profile. The set of allowable hip torque profiles is the set of bang-bang torque profiles for which the torque and angle limits are not exceeded at any time.

A hip torque τ_{hip} is first applied in the direction of the unit vector $\hat{\mathbf{e}}_{\boldsymbol{\tau}}$ for a time interval Δt_1, then in the opposite direction for Δt_2, and the torque is zero afterwards. Since the angle limit may not be exceeded at any time, the reaction mass angular velocity after the hip torque profile must be zero. Since the reaction mass angular velocity is also zero before the start of the torque profile, we define the reaction mass torque time interval Δt_{RM}, as $\Delta t_{\text{RM}} = \Delta t_1 = \Delta t_2$. The torque profile is thus

$$\boldsymbol{\tau}_{\text{hip}} = \tau_{\text{hip}}\hat{\mathbf{e}}_{\boldsymbol{\tau}}[u(t) - 2u(t - \Delta t_{\text{RM}}) + u(t - 2\Delta t_{\text{RM}})] \tag{31}$$

where $u(\cdot)$ is the Heaviside step function.

For a 2D version of the presented model, [6] and [17] have shown that the time interval has a maximum value $\Delta t_{\text{RM,max}} = \sqrt{J\theta_{\max}/\tau_{\text{hip,max}}}$, given the scalar mass moment of inertia J, the angle limit $\theta_{\max}$ with respect to vertical and hip torque limit $\tau_{\text{hip,max}}$. The appropriate scalar inertia value for the model presented here can be obtained from the mass moment of inertia tensor and the torque direction as $J = \hat{\mathbf{e}}_{\boldsymbol{\tau}}^T\mathbf{J}\hat{\mathbf{e}}_{\boldsymbol{\tau}}$.

C. Dimensional Analysis

Additional dimensionless quantities are needed to nondimensionalize the equations of motion. We define the dimensionless mass moment of inertia $\mathbf{J}'$, angular velocity $\boldsymbol{\omega}'$, and hip torque $\boldsymbol{\tau}'_{\text{hip}}$ as

$$\mathbf{J}' = \frac{\mathbf{J}}{mz_0^2} \qquad \boldsymbol{\omega}' = \frac{\mathbf{J}'\boldsymbol{\omega}}{\omega_0} \qquad \boldsymbol{\tau}'_{\text{hip}} = \frac{\boldsymbol{\tau}_{\text{hip}}}{m\omega_0^2 z_0^2}.$$

Like the dimensionless CoM position, the dimensionless angular velocity $\boldsymbol{\omega}'$ is differentiated with respect to dimensionless time t' to obtain dimensionless angular acceleration:

$$\dot{\boldsymbol{\omega}}' = \frac{d}{dt'}\boldsymbol{\omega}' = \frac{\mathbf{J}'\dot{\boldsymbol{\omega}}}{\omega_0^2}.$$

The dimensionless version of the CMP is

$$\mathbf{r}'_{\text{CMP}} = \frac{\mathbf{r}_{\text{CMP}}}{z_0} = \mathbf{r}'_{\text{CoP}} + \boldsymbol{\tau}'_{\text{hip}} \times \hat{\mathbf{e}}'_z. \tag{32}$$

These quantities can be used to rewrite the equations of motion, (29), as

$$\ddot{\mathbf{r}}' = \mathbf{P}\mathbf{r}' - \mathbf{r}'_{\text{CMP}} \tag{33a}$$

$$\dot{\boldsymbol{\omega}}' = \mathbf{P}\boldsymbol{\tau}'_{\text{hip}}. \tag{33b}$$

D. Effect of the hip torque profile

To analyze capturability, we first examine how the hip torque profile influences the instantaneous capture point motion. We will determine the instantaneous capture point location after the application of the torque profile, *i.e.*, at the final time $t'_{\text{f}} = 2\Delta t'_{\text{RM}}$, given its location at $t' = 0$ and the torque profile (31).

As noted in Section IV-A, the instantaneous capture point dynamics are described by (12) with $\mathbf{r}'_{\text{ankle}}$ replaced by $\mathbf{r}'_{\text{CMP}}$. To find the final location of the instantaneous capture point given a time-variant hip torque, we transform (12) into the Laplace domain:[6]

$$\tilde{\mathbf{r}}'_{\text{ic}}(s) = \frac{\mathbf{r}'_{\text{ic}}(0) - \tilde{\mathbf{r}}'_{\text{CMP}}(s)}{s - 1}. \tag{34}$$

The CMP motion of (32) translates to the Laplace domain as

$$\tilde{\mathbf{r}}'_{\text{CMP}}(s) = \frac{\mathbf{r}'_{\text{CoP}}}{s} + \tilde{\boldsymbol{\tau}}'_{\text{hip}}(s) \times \hat{\mathbf{e}}'_z \tag{35}$$

[6]Variables in the Laplace domain are marked with a tilde: $\tilde{\cdot}$.

The hip torque profile, (31), is given by

$$\tilde{\boldsymbol{\tau}}'_{\text{hip}}(s) = \frac{\tau'_{\text{hip}}\hat{\mathbf{e}}'_{\boldsymbol{\tau}}}{s}[1 - 2e^{-\Delta t'_{\text{RM}}s} + e^{-2\Delta t'_{\text{RM}}s}] \quad (36)$$

We can now substitute (35) and (36) into (34) to find $\tilde{\mathbf{r}}'_{\text{ic}}(s)$. Taking the inverse Laplace transform and evaluating the resulting function of time at the final time t'_{f} results in the desired expression for the final instantaneous capture point location:

$$\mathbf{r}'_{\text{ic}}(t'_{\text{f}}) = \{\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{CoP}} - \Delta\mathbf{r}'_{\text{RM}}\}e^{t'_{\text{f}}} + \mathbf{r}'_{\text{CoP}} \quad (37)$$

where

$$\Delta\mathbf{r}'_{\text{RM}} = \tau'_{\text{hip}}(\hat{\mathbf{e}}'_{\boldsymbol{\tau}} \times \hat{\mathbf{e}}'_z)[1 - 2e^{-\Delta t'_{\text{RM}}} + e^{-2\Delta t'_{\text{RM}}}] \quad (38)$$

expresses the contribution of using the reaction mass to the final location of the instantaneous capture point. The requirement for 0-step capturability may be readily determined now that the effect of the torque profile on the instantaneous capture point location is known.

E. Capturability

With a reaction mass, the legged system can be 0-step capturable even if the instantaneous capture point is not initially located inside the base of support. In fact, the requirement for 0-step capturability is that the instantaneous capture point should be inside the base of support *after the application of the torque profile*. The goal will thus be to push the instantaneous capture point inside the base of support using suitable ankle and hip torques.

We examine a case, called the boundary case, for which the instantaneous capture point can only just be pushed from outside of the base of support to the edge of the base of support. Ankle torques should be such that the CoP is placed as close to the instantaneous capture point as possible to minimize the rate of diversion of the instantaneous capture point away from the base of support. Hip torques should be such that they maximally influence the final location of the instantaneous capture point. From (38), we see that this is achieved when lunging as hard as possible and as long as possible in the direction of the instantaneous capture point. The maximum distance over which the instantaneous capture point can be changed due to lunging is

$$\left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\| = \left\| \Delta\mathbf{r}'_{\text{RM}}|_{\Delta t'_{\text{RM}}=\Delta t'_{\text{RM,max}},\tau'_{\text{hip}}=\tau'_{\text{hip,max}}} \right\|. \quad (39)$$

Since the CoP position will be the closest point to the instantaneous capture point within the base of support, it is also the position where the instantaneous capture point should end up at t'_{f}: $\mathbf{r}_{\text{ic}}(t'_{\text{f}}) = \mathbf{r}'_{\text{CoP}}$. Therefore, the requirement for reaching a captured state in the boundary case is:

$$\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{CoP}}\| = \left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\|. \quad (40)$$

The distance $\left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\|$ acts as an upper boundary for 0-step capturability: the legged system is 0-step capturable if the distance between the initial instantaneous capture point and the base of support is smaller than or equal to this distance.

For N-step capturability, we restrict the analysis to initial states in which a step has just been taken, the foot is optimally oriented, and the reaction mass is in the upright position with zero angular velocity. For these initial states, the requirement for 0-step capturability can be written in terms of distance to the contact reference point $\mathbf{r}_{\text{ankle}}$ as

$$\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\| \leq r'_{\text{max}} + \left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\| = d'_0. \quad (41)$$

The limits of capturability for higher N are calculated similarly to those in Section II-F and Section III-E. As mentioned before, we assume that the hip torque profile can only be applied once. It should be applied as soon as possible to be most effective, since waiting longer simply results in an initial instantaneous capture point location that is farther removed from the foot. After execution of the torque profile, the model essentially reduces to the 3D-LIPM with finite-sized foot, as the CMP coincides with the CoP. Our assumption was that the execution time of the torque profile is less than the minimum step time. Hence, only the first step that this model takes is different when compared to the previous model. As such, the requirement for N-step capturability is that the distance $\left\|\mathbf{r}'_{\text{ic}}(\Delta t'_{\text{s,min}}) - \mathbf{r}'_{\text{ankle}}\right\|$, is less than $d'_{N-1} + l'_{\text{max}}$ *for the model without a reaction mass*, i.e., according to (25). For clarity, we will relabel the d'_N for the 3D-LIPM with finite-sized foot as $\bar{d}'_N$.

The distance $\left\|\mathbf{r}'_{\text{ic}}(\Delta t'_{\text{s,min}}) - \mathbf{r}'_{\text{ankle}}\right\|$ is readily obtained by substituting the optimal hip torque profile, CoP placement, and stepping strategy in (37):

$$\begin{aligned}\left\|\mathbf{r}'_{\text{ic}}(\Delta t'_{\text{s,min}}) - \mathbf{r}'_{\text{ankle}}\right\| &= r'_{\text{max}} \\ &+ \{\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\| - r'_{\text{max}} - \left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\|\}e^{\Delta t'_{\text{s,min}}}. \quad (42)\end{aligned}$$

Noting that the left hand side and hence the right hand side must be less than or equal to $\bar{d}'_{N-1} + l'_{\text{max}}$, we find that the requirement on the initial distance is $\|\mathbf{r}'_{\text{ic}}(0) - \mathbf{r}'_{\text{ankle}}\| \leq d'_N$, with

$$d'_N = (l'_{\text{max}} - r'_{\text{max}} + \bar{d}'_{N-1})e^{-\Delta t_{\text{s,min}}} + r'_{\text{max}} + \left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\| \quad (43)$$

for all $N \geq 1$. Comparing these results with the results of Section III-E, we see that the relation between the d'_N for the current model and the $\bar{d}'_N$ for the model without a reaction mass is:

$$d'_N = \bar{d}'_N + \left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\|, \quad \forall N \quad (44)$$

which shows that $\left\|\Delta\mathbf{r}'_{\text{RM,max}}\right\|$ is the margin that is gained by adding a reaction mass.

F. Capture regions

The N-step capture regions for the 3D-LIPM with finite-sized foot and reaction mass are shown in Fig. 8.

Similar to the previous models, the first step to finding the capture regions is to find the set of possible future instantaneous capture point locations. This problem can be solved using (37). Comparing (37) to (13), we can rewrite (37) as

$$\mathbf{r}'_{\text{ic}}(t'_{\text{f}}) = \mathbf{r}'_{\text{ic}}(t'_{\text{f}})|_{\boldsymbol{\tau}'_{\text{hip}}=0} - \Delta\mathbf{r}'_{\text{RM}}e^{t'_{\text{f}}} \quad (45)$$

where $\mathbf{r}'_{\text{ic}}(t'_{\text{f}})|_{\boldsymbol{\tau}'_{\text{hip}}=0}$ is the instantaneous capture point location at t'_{f} when no hip torque is applied. The difference due to the use of the reaction mass is bounded as $\left\|\Delta\mathbf{r}'_{\text{RM}}e^{t'_{\text{f}}}\right\| \leq$

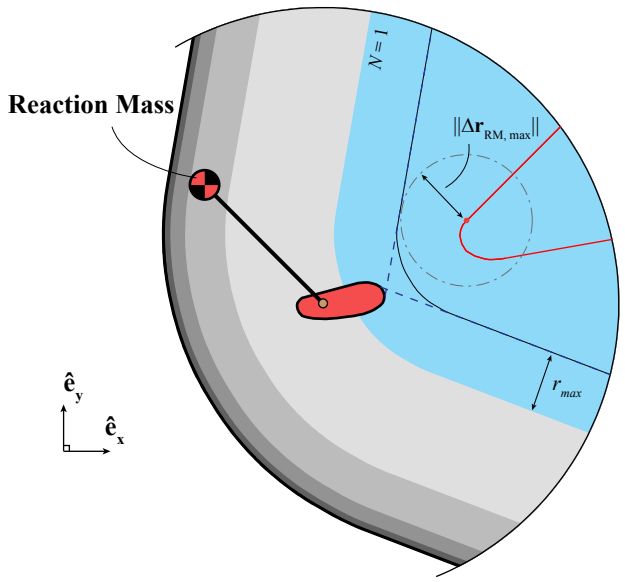

Fig. 8. Top view of the 3D-LIPM with finite-sized foot and reaction mass, with a schematic representation of the N-step capture regions. The figure is an extension of Fig. 6: state parameters $\mathbf{r}_{\text{ankle}}$, $\mathbf{r}$ and $\dot{\mathbf{r}}$ are identical. We have omitted labels that were already shown in Fig. 6 to avoid cluttering. The 1-step capture region is constructed using all possible instantaneous capture point locations of the 3D-LIPM with finite-sized foot (Fig. 6). One of these possible instantaneous capture point locations is indicated by the dot on the solid line inside the 1-step capture region. Using the reaction mass, all possible instantaneous capture point locations can be offset over distance $\|\Delta \mathbf{r}_{\text{RM,max}}\|$. For this figure, $\boldsymbol{\tau}_{\text{max}}$ is set to 0.5 and $\theta_{\text{max}} = 1/8$, which results in a total lunge time ($2\Delta t_{\text{RM,max}}$) of 1.

$\left\|\Delta \mathbf{r}'_{\text{RM,max}}\right\| e^{t'_{\text{f}}}$, and may have any direction. Therefore, the set of possible instantaneous capture point locations at the specific time t'_{f} consists of all points that lie at most $\|\Delta \mathbf{r}_{\text{RM,max}}\|$ from possible instantaneous capture point locations at t'_{f} for the model without reaction mass. Since we assume that after t'_{f} no hip torque is applied, the CMP coincides with the CoP for time greater than t'_{f}, and the instantaneous capture point will move on a line through itself and the CoP. Bounds on reachable instantaneous capture point locations are therefore found in a manner similar to that given in Section III-F, by constructing lines of sight from the base of support to the set of possible instantaneous capture point locations at t'_{f} (the dashed lines in Fig. 8). The near boundary is determined by the minimum step time constraint in conjunction with (45).

Finally, we can construct capture regions exactly as in Section III-F. After the first step is taken, no hip torque is applied anymore and the model essentially reduces to the 3D-LIPM with finite-sized foot. We should hence construct nested regions around the set of possible future instantaneous capture point locations using the d'_N for that model, i.e., those calculated using (25), not the ones from (44).

V. Capturability Comparison

For all three models, we determined which states in a subset of state space are N-step capturable, and derived descriptions of the N-step capture regions. The N-step capture regions of Fig. 4, 6 and 8 clearly showed that an increase in the number of possible stabilizing mechanisms leads to an increase in region size. This result implies that there is more freedom to choose foot placements that keep the legged system capturable, or we could say that the 'level of capturability' increases. To express this level of capturability in a more formal and quantitative manner, we introduce two capturability metrics that apply specifically to the models presented in this part:

1) Let $\mathbf{x}$ be the state of a legged system. The legged system's *capturability margin* is the area of the ∞-step capture region in state $\mathbf{x}$.
2) Let K be a subset of a legged system's state space X, and let $C(K)$ denote the subset of K that is capturable. For each capturable initial state $\mathbf{x} \in C(K)$, we can compute the distance between the instantaneous capture point and the contact reference point. The d_∞ *capturability level* of the legged system with respect to K is defined as the largest such distance over all states $\mathbf{x} \in C(K)$.

The first metric was first mentioned in Part 1. Since the ∞-step capture region region is two-dimensional for the models in this part, we have chosen the area as a measure of its size. The first metric expresses how close a specific state of a legged system is to not being capturable. It also gives an indication of the input deviations and disturbances that are allowed while executing a given evolution. A small size of the ∞-step capture region, for example, indicates that there is little room for disturbances.

In the previous sections, we graphically depicted the influence of the various model parameters on the ∞-step capture region for a given initial state. Fig. 9 combines these results and displays the size of the three ∞-step capture regions. For the selected initial state, the addition of a finite-sized foot caused the metric to increase by 260%. Another increase of 130% was found for the addition of the reaction mass.

Expressions for the dimensionless version of the d_∞ capturability level, d'_∞, were derived for all three models, where the subset of state space K for each model is as specified in Sections II-F, III-E and IV-E respectively. Since it considers a set of states K instead of just a single state, it gives an indication of the overall legged-system stability and allows for comparison between different legged systems. In terms of the original physical quantities, d_∞ for the 3D-LIPM with finite-sized foot is expressed as

$$d_\infty = \underbrace{l_{\text{max}} \frac{e^{-\omega_0 \Delta t_{\text{s,min}}}}{1 - e^{-\omega_0 \Delta t_{\text{s,min}}}}}_{\text{3D-LIPM, Section II}} + \underbrace{r_{\text{max}}}_{\text{3D-LIPM, Section III}} + \underbrace{\frac{\tau_{\text{hip,max}}}{m\omega_0^2 z_0}\left[1 - 2e^{-\omega_0 \Delta t_{\text{RM,max}}} + e^{-2\omega_0 \Delta t_{\text{RM,max}}}\right]}_{\text{3D-LIPM, Section IV}}. \quad (46)$$

VI. Discussion and Future Work

To analyze capturability, we defined the instantaneous capture point, which is determined only by the CoM position and velocity. This gave us a dimensionally-reduced description of the dynamics of three walking models. We showed how this resulted in relatively simple and comprehensible expressions, and enabled calculation and visualization of capture regions

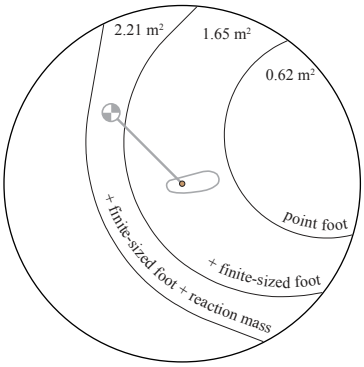

Fig. 9. Three superimposed ∞-step capture regions of all three models as previously presented in Fig. 4, 6 and 8. The size of each ∞-step capture region is shown.

and viable-capture basins. We will comment on how these expressions and regions are useful in the analysis and control of legged locomotion.

A. Insights gained from the presented models

The point foot model showed that the velocity of the instantaneous capture point is directed away from the foot. It was shown to be proportional to the distance to the foot and to incresae exponentially with time. This suggests that in order to remain capturable, the foot should be placed sufficiently quickly in the direction of the instantaneous capture point. This simple stepping strategy was used to create a variety of stable locomotion patterns in simulation [12], [21] and was also found to be a good predictor of stable foot placement locations in the analysis of human walking [15], [22]–[24].

The model with finite-sized foot demonstrated how CoP modulation can change the location and velocity of the instantaneous capture point. This property can be used, for example, to change the walking direction. The expression for an equivalent constant CoP enabled the definition of a fixed CoP location that results in an identical instantaneous capture point as given by a time-variant CoP location. We showed how this can further reduce the complexity of a capturability analysis.

The model with finite-sized foot and reaction mass showed that lunging as soon as possible in the direction of the instantaneous capture point maximally increases the level of capturability. We believe that, if lunging is constrained by angle and torque limits, bang-bang control achieves the optimal effect. This reasoning implies that the acceleration time interval and deceleration time interval of the reaction mass should be maximized and equal. Note that in general it is not straightforward to relate the effect of the angular momentum generated by the simple reaction mass to the effect generated by all individual links of a complex multibody system [18], [25]. However, we were able to demonstrate the conceptual contribution of angular momentum to the stability of locomotion, represented mathematically by (46) and graphically by Fig. 9.

The three models also revealed the relation between the location of the point foot, the CoP and the CMP in the analysis of capturability. Despite time variant inputs, the dynamics of the instantaneous capture point remains easy to predict for all three models: the instantaneous capture point diverges away from the CMP along a straight line at a velocity proportional to the distance to the CMP. The CMP reduces to the CoP if no reaction mass is present or actuated. The CoP reduces to the point foot location if the base of support is infinitesimally small.

Furthermore, the models are not only valuable for the analysis and control of walking, but also for example for regaining upright balance after a perturbation. For such a situation, we can approximate the limits of 0-step capturability for a particular robot, and determine appropriate step locations, ankle torques and hip torques to recover postural equilibrium.

B. Limitations of the presented models

The use of these models as a representation of legged locomotion has a number of limitations.

The models discard many aspects of legged locomotion. Height variations of the CoM during legged locomotion were not considered. Internal forces generated by lunging or swing leg dynamics were discarded. Slippage or losses at the change of support were not considered. The existence of a double stance phase in case of walking was also not taken into account. Consequently, using these simple models to approximate the capturability of a real robot will lead to discrepancies between the approximated and true values.

Furthermore, the limitations on the stabilizing control inputs were modeled simplistically. For example, consider the limitations on the stepping performance of the model. Stepping speed was constrained by enforcing a constant minimum step time, independent of the step location. Step location was constrained by limiting the maximum step length, irrespective of the current CoM position or direction of motion. The expressions for capturability in this paper rely strongly on these simplistically modeled limitations.

We see an advantage in the simplicity of the presented models however. Comparable studies demonstrated that even slightly more complex models can result in expressions that are less comprehensible and require complex numerical methods to be solved [26], [27]. This decreases understanding and increases the computational burden. Although the models are very elementary, they are still useful for the analysis and control of legged locomotion. From a viewpoint of analysis, the models exposed fundamental principles of locomotion: the basic influence of the stabilizing mechanisms on N-step capturability was clearly demonstrated. Control algorithms can be based on the results obtained for the simple models. We can use the models offline, where the results serve as a starting point for a learning algorithm [28]. Or we can use the models online, where the models serve as an integral part of the control algorithm of a complex robot. Although the models give approximations, the results can be sufficient for control purposes, as we will demonstrate in Part 3.

C. Insights gained from capturability analysis

For each model, we chose a subset of the state space and examined which states in that subset are capturable. These

results were used to compute N-step capture regions for N up to and including ∞. The instantaneous capture point, which combines the CoM position and velocity, was of great use in the capturability analysis.

Our analysis of capturability took both the dynamics and actuation limits of the legged system into account, while no specific control law was assumed *a priori*. This approach allows us to make some strong conclusions concerning capturability. For example, if there is no ∞-step capture region, then it is impossible to make the legged system come to a stop without falling, no matter what control law is used.

We also introduced two capturability metrics that enable a quantitative comparison between models. The capturability margin expresses the level of capturability in a single state, whereas the d_∞ capturability level is based on a subset of state space. Human subject studies already demonstrated that the CoM position and velocity in relation to the base of support is a good indicator of the ability to maintain balance and the number of steps required to do so [29]–[33]. Hof *et al.* [13] were the first to formally define the distance between the instantaneous capture point (which these authors call the 'extrapolated CoM') and the base of support as a 'margin of stability'. We see an advantage to using our metrics, since they take the effects and limits of the stabilizing control inputs (foot location, ankle torque and hip torque) into account.

The influence of each stabilizing mechanism on capturability was demonstrated; see (46). As an application of capturability analysis, we estimated its implications for human locomotion. Based on estimates for anthropomorphic model parameters (see Appendix), the d_∞ capturability level has a magnitude of 0.7 meters. The contribution of the foot length to the d_∞ capturability level is about 25%, and the contribution of lunging is about 10%. This suggest that, not surprisingly, the ability to perform rapid steps is most important to remain capturable. This suggestion is also expressed by the metric being most sensitive to changes in minimum step time. A variation in minimum step time can be compensated by another stabilizing mechanism to retain the same level of capturability. However, a 10% increase in step duration already requires a 17% longer step or a 30% longer foot. For humans, selection of the appropriate step speed and length may be a trade-off between the required muscle strength to perform a quick step [34], [35] and the perceived level of stability or safety of the selected step length [31], [36], [37].

D. Limitations of the capturability analysis

The presented method of analyzing capturability has provided a great deal of insight, but it directly applies to only a small class of models. For example, it is not straightforward to apply this analysis to models with non-linear dynamics, since the analysis depends heavily on the linear and first order dynamics of the instantaneous capture point. Because of linearity, capture regions could be computed using geometric relations. Computing capture regions for non-linear models is computationally much more expensive [28] and remains a topic for future research. Another unexplored area is the consideration of keep out regions in the computation of capture regions.

As noted, we have not computed complete viable-capture basins, because the computation of the N-step capture regions and the presented capturability metrics does not require this information. Instead, we have only considered a subset of the state space of each model, to simplify the analysis. Complete knowledge of the viable-capture regions would however still be useful.

Plots of the capture regions, such as presented in Fig. 4, 6 and 8, do not directly give information about which actions should be taken to come to a stop using a given capture point. However, the underlying analysis that allowed us to construct these plots does provide that information. Moreover, the plots themselves can be used to exclude stepping locations that will inevitably lead to a fall.

We chose to define the d_∞ capturability level, described in Section V, with respect to a subset of state space, denoted K. There seems to be no completely unambiguous choice of K that results in a fair comparison between the different models, which each have different state spaces. One could argue that, for each model, K should be the model's entire state space. However, that choice would make the value of the metric for the 3D-LIPM with finite-sized foot and reaction mass unrealistically high. For that model, the value of the metric, i.e., the largest distance between the contact reference point and the instantaneous capture point for which the legged system is capturable, would correspond to an initial state in which the legged system leans all the way back from the instantaneous capture point, prepared to lunge as long as possible in its direction. Such a state is rarely encountered in a realistic scenario. This line of reasoning leads to our choice of K as the set of states for which the reaction mass angle is zero.

We feel that the analysis of the simple models presented in this part of our three-part paper can play an important role in the future development of more versatile, stable and natural robot motions. We have already used these models to design foot planning algorithms and to synthesize robot balance-recovery strategies in response to a push, which will be described in detail in Part 3.

APPENDIX

We estimated anthropomorphic model parameters for the 3D-LIPM with finite size foot and reaction mass, see Table I. Mass and length parameters are based on a typical human 1.75 m tall and with a mass of 70 kg. Gait parameters are based on experimental studies on human trip recovery.

TABLE I
ESTIMATES OF ANTHROPOMORPHIC MODEL PARAMETERS.

Parameter	Symbol	Value	Units	Ref.
Step length	$l_{\max}$	0.7	m	[38], [39]
Time between steps	$\Delta t_{s,\min}$	0.3	s	[38], [39]
Ankle to toe length	$r_{\max}$	0.2	m	[40]
CoM height	z_0	0.95	m	[40]
HAT segment max. angle	$\theta_{\max}$	0.5	rad	[9], [38]
Moment of inertia of HAT w.r.t. body CoM	J	8	kg m^2	[40]
Body mass	m	70	kg	[40]
Hip torque	$\tau_{\max}$	100	Nm	[41]

ACKNOWLEDGMENT

The authors gratefully acknowledge helpful discussions with D. Karssen, whose ideas contributed to the work. We would like to thank E. Westervelt for his helpful comments.

REFERENCES

[1] S. Kajita and K. Tanie, "Study of dynamic biped locomotion on rugged terrain—derivation and application of the linear inverted pendulum mode—," in *Proc. 1991 IEEE Int. Conf. Robot. Automat.*, vol. 2. IEEE Comput. Soc. Press, Apr. 1991, pp. 1405–1411.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *Proc. 2001 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, vol. 1, Oct./Nov. 2001, pp. 239–246.

[3] F. Horak and L. Nashner, "Central programming of postural movements: adaptation to altered support-surface configurations," *J. Neurophysiol.*, vol. 55, no. 6, p. 1369, Jun. 1986.

[4] M. Guihard and P. Gorce, "Dynamic control of bipeds using ankle and hip strategies," in *Proc. 2002 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, vol. 3, Sep./Oct. 2002.

[5] D. N. Nenchev and A. Nishio, "Ankle and hip strategies for balance recovery of a biped subjected to an impact," *Robotica*, vol. 26, no. 05, pp. 643–653, Sep. 2008.

[6] B. J. Stephens, "Integral control of humanoid balance," in *Proc. 2007 IEEE/RSJ Int. Conf. Intell. Rob. Syst.* IEEE, Oct. 2007, pp. 4020–4027.

[7] M. Abdallah and A. Goswami, "A Biomechanically Motivated Two-Phase Strategy for Biped Upright Balance Control," in *Proc. 2005 IEEE Int. Conf. Robot. Automat.* IEEE, Apr. 2005, pp. 1996–2001.

[8] S.-H. Hyon, J. Hale, and G. Cheng, "Full-Body Compliant Human-Humanoid Interaction: Balancing in the Presence of Unknown External Forces," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 884–898, Oct. 2007.

[9] J. van der Burg, M. Pijnappels, and J. van Dieën, "Out-of-plane trunk movements and trunk muscle activity after a trip during walking," *Exp. Brain Res.*, vol. 165, no. 3, pp. 407–412, Sep. 2005.

[10] P. E. Roos, M. P. McGuigan, D. G. Kerwin, and G. Trewartha, "The role of arm movement in early trip recovery in younger and older adults," *Gait & Posture*, vol. 27, no. 2, pp. 352–356, Feb. 2008.

[11] M. Pijnappels, I. Kingma, D. Wezenberg, G. Reurink, and J. van Dieën, "Armed against falls: the contribution of arm movements to balance recovery after tripping," *Exp. Brain Res.*, vol. 201, no. 4, pp. 689–699, Apr. 2010.

[12] J. E. Pratt and R. Tedrake, "Velocity-based stability margins for fast bipedal walking," in *Fast Motions in Biomechanics and Robotics*, ser. Lecture Notes in Control and Information Sciences, M. Diehl and K. Mombaur, Eds. Springer Berlin Heidelberg, 2006, vol. 340, ch. 14, pp. 299–324.

[13] A. L. Hof, M. Gazendam, and W. Sinke, "The condition for dynamic stability," *J. Biomech.*, vol. 38, no. 1, pp. 1–8, 2005.

[14] A. L. Hof, R. M. van Bockel, T. Schoppen, and K. Postema, "Control of lateral balance in walking: Experimental findings in normal subjects and above-knee amputees," *Gait & Posture*, vol. 25, no. 2, pp. 250–258, Feb. 2007.

[15] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking," *Hum. Movement Sci.*, vol. 27, no. 1, pp. 112–125, Feb. 2008.

[16] A. Goswami, "Postural stability of biped robots and the foot rotation indicator (FRI) point," *Int. J. Robot. Res.*, vol. 18, no. 6, pp. 523–533, 1999.

[17] J. E. Pratt, J. Carff, S. V. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Proc. 2006 IEEE-RAS Int. Conf. Humanoid Rob.* IEEE, Dec. 2006, pp. 200–207.

[18] S.-H. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *Proc. IEEE Int. Conf. Robot. Automat.* IEEE, Apr. 2007, pp. 4667–4672.

[19] B. J. Stephens, "Humanoid push recovery," in *Proc. 2007 IEEE-RAS Int. Conf. Humanoid Rob.* IEEE, Nov. 2007, pp. 589–595.

[20] M. B. Popovic, A. Goswami, and H. M. Herr, "Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications," *Int. J. Robot. Res.*, vol. 24, no. 12, pp. 1013–1032, 2005.

[21] K. Yin, K. Loken, and M. van de Panne, "SIMBICON: Simple Biped Locomotion Control," in *Proc. SIGGRAPH 2010*. New York, NY, USA: ACM Press, 2007, p. 105.

[22] M. A. Townsend, "Biped gait stabilization via foot placement," *J. Biomech.*, vol. 18, no. 1, pp. 21–38, 1985.

[23] M. Millard, D. L. Wight, J. McPhee, E. G. Kubica, and D. W. Wang, "Human foot placement and balance in the sagittal plane," *J. Biomech. Eng.*, vol. 131, no. 12, p. 121001, 2009.

[24] A. L. Hof, S. M. Vermerris, and W. A. Gjaltema, "Balance responses to lateral perturbations in human treadmill walking," *J. Exp. Biol.*, vol. 213, pp. 2655–2664, 2010.

[25] D. E. Orin and A. Goswami, "Centroidal Momentum Matrix of a humanoid robot: Structure and properties," in *Proc. 2008 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, Sep. 2008, pp. 653–659.

[26] D. L. Wight, E. G. Kubica, and D. W. Wang, "Introduction of the Foot Placement Estimator: A Dynamic Measure of Balance for Bipedal Robotics," *J. Comput. Nonlinear Dynam.*, vol. 3, no. 1, p. 011009, 2008.

[27] J. E. Pratt and S. V. Drakunov, "Derivation and Application of a Conserved Orbital Energy for the Inverted Pendulum Bipedal Walking Model," in *Proc. 2007 IEEE Int. Conf. Robot. Automat.*, 2007.

[28] J. R. Rebula, F. Cañas, J. E. Pratt, and A. Goswami, "Learning capture points for humanoid push recovery," in *Proc. 2007 IEEE-RAS Int. Conf. Humanoid Rob.*, 2007.

[29] Y.-C. Pai and J. Patton, "Center of mass velocity-position predictions for balance control," *J. Biomech.*, vol. 30, no. 4, pp. 347–354, Apr. 1997.

[30] Y.-C. Pai, M. W. Rogers, J. Patton, T. D. Cain, and T. A. Hanke, "Static versus dynamic predictions of protective stepping following waist-pull perturbations in young and older adults," *J. Biomech.*, vol. 31, no. 12, pp. 1111–1118, 1998.

[31] B. E. Maki and W. E. McIlroy, "The control of foot placement during compensatory stepping reactions: does speed of response take precedence over stability?" *IEEE Trans. Rehabil. Eng.*, vol. 7, no. 1, pp. 80–90, Mar. 1999.

[32] E. T. Hsiao and S. N. Robinovitch, "Elderly subjects' ability to recover balance with a single backward step associates with body configuration at step contact," *J. Gerontol. A Biol. Sci. Med. Sci.*, vol. 56, no. 1, pp. M42–47, 2001.

[33] Z. Aftab, P.-B. Wieber, and T. Robert, "Comparison of Capture Point estimation with human foot placement : Applicability and Limitations," oral presentation at 5èmes Journées Nationales de la Robotique Humanoïde, Poitier, Jun. 2010.

[34] D. G. Thelen, L. A. Wojcik, A. B. Schultz, J. A. Ashton-Miller, and N. B. Alexander, "Age differences in using a rapid step to regain balance during a forward fall," *J. Gerontol. A Biol. Sci. Med. Sci.*, vol. 52, no. 1, pp. M8–13, 1997.

[35] C. Smeesters, W. Hayes, and T. McMahon, "The threshold trip duration for which recovery is no longer possible is associated with strength and reaction time," *J. Biomech.*, vol. 34, no. 5, pp. 589–595, May 2001.

[36] E. T. Hsiao-Wecksler and S. N. Robinovitch, "The effect of step length on young and elderly women's ability to recover balance," *Clin. Biomech.*, vol. 22, no. 5, pp. 574–580, Jun. 2007.

[37] V. Weerdesteyn, B. Nienhuis, T. Mulder, and J. Duysens, "Older women strongly prefer stride lengthening to shortening in avoiding obstacles," *Exp. Brain Res.*, vol. 161, no. 1, pp. 39–46, Feb. 2005.

[38] M. J. Pavol, T. M. Owings, K. T. Foley, and M. D. Grabiner, "Mechanisms leading to a fall from an induced trip in healthy older adults," *J. Gerontol. A Biol. Sci. Med. Sci.*, vol. 56, no. 7, pp. M428–437, 2000.

[39] A. Forner Cordero, H. Koopman, and F. van der Helm, "Multiple-step strategies to recover from stumbling perturbations," *Gait & Posture*, vol. 18, no. 1, pp. 47–59, Aug. 2003.

[40] D. A. Winter, *Biomechanics and Motor Control of Human Movement*, 2nd ed. Wiley, 1990.

[41] L. A. Wojcik, D. G. Thelen, A. B. Schultz, J. A. Ashton-Miller, and N. B. Alexander, "Age and gender differences in peak lower extremity joint torques and ranges of motion used during single-step balance recovery from a forward fall," *J. Biomech.*, vol. 34, no. 1, pp. 67–73, Jan. 2001.

Twan Koolen received his B.Sc. degree from the faculty of mechanical engineering at the Delft University of Technology in the Netherlands in 2009. He is currently pursuing an M.Sc. degree in mechanical engineering at the Delft Biorobotics Laboratory and working as a Research Associate at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include robustness in legged locomotion, multibody dynamics and robust control.

Tomas de Boer received the M.Sc. degree in mechanical engineering from Eindhoven University of Technology of, Eindhoven, The Netherlands, in 2006. He is currently working toward the Ph.D. degree with Delft University of Technology, Delft, The Netherlands. His current research interests include bioinspired robotics, legged locomotion, mechatronics, (multibody) dynamics and (compliant) control.

John Rebula received his S.B. degree in Mechanical Engineering from MIT in 2006, and is currently in the Ph.D. program in the Mechanical Engineering Department at the University of Michigan. He is interested in analysis, planning, and control of human and robotic locomotion.

Ambarish Goswami has been with Honda Research Institute in California, USA, for the past nine years, where he is currently a Principal Scientist. His field is dynamics and control, and his main research is in balance maintenance and fall for the Honda humanoid robot ASIMO. He received the Bachelors degree from Jadavpur University, Kolkata, India, the Masters degree from Drexel University, Philadelphia, PA, and the Ph.D. degree from Northwestern University, Evanston, IL, all in Mechanical Engineering.

Ambarish Goswamis Ph.D. work, under Prof. Michael Peshkin, was in the area of automated assembly and robot-assisted surgery. For four years following his graduation he worked at the INRIA Laboratory in Grenoble, France, as a member of the permanent scientific staff (Charge de Recherche). He became interested in human walking and in biomechanics while working on the first anthropomorphic biped robot "BIP" in France. This interest in gait study subsequently brought him to the Center for Human Modeling and Simulation (as an IRCS Fellow) of the University of Pennsylvania, Philadelphia, and a three year position as a core animation software developer for 3D Studio Max at Autodesk. Ambarish Goswami has held visiting researcher positions at the Ohio State University and the University of Illinois at Urbana-Champaign for short periods.

Jerry Pratt received his Ph.D. degree in Computer Science from MIT in 2000. He is currently a research scientist at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include the analysis and control of bipedal and quadrupedal walking and running, humanoid robots, and exoskeletons. Some of the robotic projects he has been involved in include Spring Turkey, Spring Flamingo, M2, the RoboKnee, tBot, LittleDog algorithms, M2V2, and the IHMC ROAM Exoskeleton.

Capturability-Based Analysis and Control of Legged Locomotion, Part 3: Application to M2V2, a Lower Body Humanoid

Jerry Pratt, Twan Koolen, Tomas de Boer, John Rebula, Sebastien Cotton, John Carff, Matthew Johnson, Peter Neuhaus

***Abstract*—This three-part paper discusses the analysis and control of legged locomotion in terms of N-step capturability: the ability of a legged system to come to a stop without falling by taking N or fewer steps. We consider this ability to be crucial to legged locomotion and a useful, yet not overly restrictive criterion for stability.**

Part 1 provided theoretical background on capturability. Part 2 showed how to obtain capture regions and control sequences from simplified gait models. In Part 3, we describe an algorithm that uses these results as approximations to control a complex humanoid robot. This algorithm was tested using M2V2, a 3D force-controlled humanoid robot with 12 actuated degrees of freedom in the legs, both in simulation and in real-world experiments. Two control tasks were defined: 1) balancing on one leg and stepping to regain balance when necessary, and 2) walking. While performing the balancing task, the real robot was able to recover from forward and sideways pushes of up to 21 Ns. The simulated version of the robot was able to recover from sideways pushes of up to 15 Ns while walking, and walked across randomly placed stepping stones.

***Index Terms*—**

Capture point, Motion control, Balance control, Bipedal robots, Push Recovery, Capturability

I. Introduction

Making humanoid robots useful in complex environments requires attaining good disturbance rejection properties while performing other tasks, such as walking. Current robots have not sufficiently demonstrated these properties. In Part 1, we proposed to approach this problem using the concept of capturability, roughly defined as the ability to come to a stop. Part 2 provided a capturability analysis of three simplified gait models, which allowed insight into the contributions of three different stabilizing mechanisms to capturability: stepping, ankle control and lunging. This part presents capturability-based control algorithms for balancing and walking while being robust to pushes and unexpected ground variations. These control algorithms were implemented on M2V2, a 12 degree of freedom force controlled 3D lower body humanoid robot.

In the presented control algorithms, we make use of the concepts derived in the previous parts of this three-part paper, such as the instantaneous capture point and the 1-step capture region, which are approximated using the 3D Linear Inverted Pendulum Model (3D-LIPM) with finite-sized foot, as described in Part 2. Parts 1 and 2 also introduced the *capture margin*, which will be used in the current part to evaluate the performance of the presented control algorithms.

To date, we have achieved push recovery during one-legged balancing on the real robot. In simulation, we have also achieved push recovery while walking and walking over stepping stones. These experimental results demonstrate that capturability-based control using simplified dynamic models may be useful in developing bipedal walking control algorithms that are robust to disturbances. While performing one-legged balancing, the real M2V2 was able to recover from forward and sideways pushes of up to 21 Ns. The simulated version of the robot was able to recover from sideways pushes of up to 15 Ns while walking without stepping stones.

The remainder of this part is structured as follows. Section II presents related literature. In Section III we describe the M2V2 robot. Section IV describes the simulation environment for the robot. In Section V we describe the control tasks that we are interested in achieving with the robot. Section VI describes some of the control concepts that we employ in developing control algorithms. Section VII presents implementation details of our control algorithms. Section VIII presents results. Finally, in Section IX we discuss capturability-based analysis and control and suggest future work.

II. Background

The literature on control algorithms for humanoid robots is extensive. Here we provide a brief survey of some widely used control techniques and focus on their disturbance rejection properties.

a) ZMP-based trajectory tracking control: The Zero Moment Point (ZMP) is the point about which the resultant ground reaction torque has no horizontal component [1]. ZMP-based trajectory tracking control usually encompasses choosing a desired ZMP trajectory based on available footholds and desired gait properties, and calculating the center of mass

Manuscript received March 2011. This work was funded through the Army Tank and Automotive Research and Development Command, the Defense Advanced Research Projects Agency, the Office of Naval Research, NASA, and the Honda Research Institute.

J. Pratt, T. Koolen, J. Carff, M. Johnson, and P. Neuhaus are with the Institute for Human and Machine Cognition, Florida

T. Koolen, and T. de Boer are with the Delft University of Technology, the Netherlands

J. Rebula is with the University of Michigan, Ann Arbor

(CoM) motion that results in that ZMP trajectory [2], [3]. It is widely used in legged robot control since maintaining the ZMP strictly inside the support polygon at all times guarantees that it is physically possible to track the reference joint trajectories using conventional control tools. The distance from the ZMP to the edge of the support polygon can be used as a measure of robustness. Modifying the reference ZMP trajectory online has also been explored, including both small local ZMP changes and larger step placement changes [4]. Another ZMP control approach treats the ZMP as a control input, which is manipulated to produce a desired motion of the CoM. For example, central pattern generators have been used to calculate the reference ZMP trajectory to produce walking [5].

Typical ZMP-based gait generation techniques cannot be used to generate gaits for which the stance foot rolls from heel to toe, as observed in fast human walking, since the ZMP is undefined when the robot rotates about an edge of the support polygon [1]. Also, the reference joint trajectories themselves might lead the robot to a fall by design, even if the ZMP is kept inside the support polygon at all times. Hence, the ZMP criterion is not a necessary condition to avoid falling [6]. In addition, the ZMP criterion is not applicable to non-flat terrain [7]. Most importantly for the present work, ZMP analysis does not answer the crucial question of where to step to recover from large disturbances.

b) Passive Dynamics Based Control: Another approach to walking control explicitly relies upon the passive behavior of the robot's mechanical components. Simple walking has been demonstrated for purely passive devices walking down a slope [8] [9]. Adding limited actuation to machines designed for passive walking can yield a controlled, efficient gait [10]. Rejection of small disturbances has been shown for planar walkers under limited control [11], [12], as well as locally stable gaits with purely reflexive control [13]. A major focus of our current work is the ability to recover from disturbances large enough to require significant actuation, so we cannot rely on passive dynamics alone to avoid falling.

c) Hybrid Zero dynamics: Another approach to locomotion control identifies relationships between the degrees of freedom of a robot that lead to a steady gait [14]. These relationships are then enforced by a feedback controller, yielding a locally stable, periodic gait. This method has been shown to reject small disturbances to terrain [15] [16]. More recently, this method has been used to generate three dimensional walking [17]. However, it requires off-line computation of a repetitive gait, and therefore it currently has no mechanism for explicitly handling rough terrain with impassable areas. Also, it is still unclear how a robot using this method will handle large pushes that significantly disturb the state of the machine from the preplanned gait.

d) Compliant Strategies for Force Controllable Robots: Force controllable robots have lead to the development of compliant, full body control strategies. Virtual Model Control and other intuitive control strategies were used on the 2D walking robot Spring Flamingo [18]. Coros *et al.* [19] combine several control techniques, including Jacobian transpose, joint PD, and gravity compensation control with step planning based on an inverted pendulum model to obtain a walking controller that works for a range of simulated characters while performing secondary tasks. Stephens and Atkeson [20] introduced an algorithm that combines joint PD control, Virtual Model Control and Dynamic Balance Force Control, an inverse dynamics approach based on the contact forces obtained from a CoM planner. Hyon *et al.* achieved disturbance rejection using a passivity-based controller [21], later complemented by CoM control using a Dynamic Balancer [22].

Compliant strategies can enhance the robustness of a walking algorithm since they focus on interaction forces with the environment in order to achieve higher level goals, instead of relying on high gain position control and extremely accurate ground models to achieve perfect kinematic trajectories. The presented work utilizes compliant control strategies, and extends some of the control strategies from previous work [18] to 3D.

III. Description of M2V2 Robot

M2V2 (Fig. 1) is a twelve degree of freedom lower-body humanoid [23]. See Appendix A for the joint layout and inertia parameters. It is a second generation redesign of M2, a robot developed at the MIT Leg Laboratory [24], [25]. Each degree of freedom is driven by a force controllable Series Elastic Actuator (SEA) [26], [27]. These actuators use a spring in series between the drive train and the load. By measuring the spring deflection, the force of the actuator can be measured. Using a feedback controller, the actuator force can be accurately and quickly controlled. For M2V2, each actuator can produce a force of up to 1.3 kN, with a smallest resolvable force of approximately 4.4 N, giving it a $300:1$ dynamic range. The force controllable bandwidth of each actuator is approximately 40 Hz.

M2V2 has two U.S. Digital EM1-0-500 linear encoders and LIN-500 encoder strips at each Series Elastic Actuator, one to measure position and one to measure spring deflection. Onboard computation is provided by a PC104 with a dual core Pentium-M processor. Sensor reading is done by several AccesIO 104-Quad8 encoder input boards. Desired current is output as a PWM signal through two Real Time Devices 6816 PWM boards. Body orientation and angular rate is measured using a MicroStrain 3DM-GX3-25 inertial measurement unit. Current control is provided through twelve Copley Controls Accelnet module ACM-180-20 amplifiers. The PC104, I/O cards, and current amplifiers are all located in the body of the robot. In addition, a custom designed push stick equipped with a digital Loadstar ILoad Pro load cell was constructed for measuring pushing forces applied to the robot. This load cell is connected to to the robot to eliminate data synchronization issues, but the control algorithm does not have access to its output.

IV. Simulation environment

We have developed a simulation model of M2V2 using the Yobotics Simulation Construction Set software package [28]. This software package allows for rigid body dynamic modeling and simulation using the Articulated Body Algorithm [29],

Fig. 1. M2V2, a twelve degree of freedom lower-body humanoid robot. The robot uses twelve identical Series Elastic Actuators in each of its joints. There are three degrees of freedom at each hip, one at each knee, and two at each ankle.

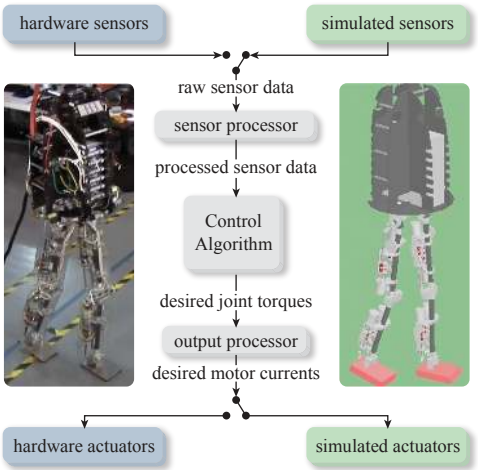

Fig. 2. High level overview of the software architecture. The majority of the software runs both in simulation and on the real robot, eliminating the need to maintain separate versions, and allowing for significant development and testing in simulation.

[30] and a fourth order Runge-Kutta integrator. The integration step size we use is 0.1 milliseconds. Ground contact is simulated by attaching ground contact points to locations on the feet and modeling the ground as a surface via a function from location to height and surface normal. Ground contact forces are determined using spring-damper ground models (penalty-based methods).

Pushing disturbances are modeled as high intensity impulse forces of constant magnitude for a short duration. The force is applied to the midpoint between the hip joints. Stepping stones are modeled as polygons.

Our software architecture (see Fig. 2) is designed to have a common control algorithm that is used both in simulation and on the real robot. The only differences between the simulation and the real robot are the source of the raw sensor data and the destination of the desired motor currents. The control algorithm threads run at the same rate for the simulation as they do in the real robot. By having the exact same code base for the simulation and the real robot, we eliminate the effort and bugs that are typical when porting from a simulation environment to real hardware.

The simulated sensors include noise and discretization error. The actuators are simulated as low pass filters to simulate the bandwidth of the Series Elastic Actuators and have maximum output force limits.

Despite efforts to minimize the differences between the simulated and real robots, there is lingering discrepancy between simulated and real motion. To address this problem, additional tuning on the real robot is required before an algorithm that generates performs well in simulation can also perform well in real experiments.

V. Control Tasks

The controller was designed for two separate control tasks: 1) balancing on one leg and 2) walking. These tasks require slightly different implementations of some of the modules in the controller, although the main control algorithm is the same.

A. Balancing

Balancing on one leg (subsequently called 'balancing') entails going from a double support configuration to single support, and remaining in single support for as long as possible. If the robot is significantly perturbed while in single support, it will need to take a step in order to prevent a fall. Possible disturbances include moderate size pushes and, to a lesser extent, sensor and actuator imperfections. Pushes of a magnitude requiring the robot to take a step will be assumed to have a direction that does not require a cross-over step. For example, if the robot is balancing on its left leg, then significant pushes could be directed forward, backward or to the right, but not to the left. This task was chosen as a precursor to walking because it is challenging and requires good foot placement, while it is not as hard as walking due to the lack of the requirement of sustained forward progression. To date we have achieved balancing on the real robot, recovering from forward and sideways pushes up to 21 Ns.

B. Walking

The second control task is to walk, with the same external disturbances as listed for the balancing task. The walking task requires the robot to move forward and manage slow changes in desired walking direction. In addition, the controller should be able to handle keep out regions, i.e. regions on the ground that the robot cannot step to. To date, walking on flat ground without pushes has been achieved on the real robot. Walking in the presence of pushes up to 15 Ns and walking over stepping stones have been achieved in simulation.

VI. Control Concepts

This section describes some of the key concepts we use in our control strategies, regardless of the particular control task, namely 1) capturability-based control using an approximate model, 2) force control and 3) Virtual Model Control.

A. Capturability-based Control using an Approximate Model

We consider bipedal locomotion to be an inherently robust control problem, which does not require great accuracy in controller design. This motivates our use of an approximate model. Out of the three simplified gait models described in Part 2, we have chosen to use the 3D-LIPM with finite-sized foot for approximations. See Section IX-A for the motivation for this choice. A full capturability analysis for this model was presented in Part 2, and its results are used to great effect in the presented controller. In particular, we base our control strategy on the instantaneous capture point and the approximated 1-step capture region. Although no guarantees on capturability can be made for the robot using this approximation, we have found that it works well in practice.

Considering the 3D-LIPM with finite-sized foot, the instantaneous capture point is the point on the ground where the Center of Pressure (CoP) should be placed instantaneously and maintained in order to come to a rest with the CoM directly above the CoP. We do not specify a desired CoM trajectory; instead, desired instantaneous capture point paths are used as a basis for control. The linear dynamics of the instantaneous capture point allows us to find a desired CoP location within the base of support that 'pushes' the instantaneous capture point along the desired path.

For the balancing task, we determine whether taking a step is necessary based on whether the instantaneous capture point has left the base of support (Corollary 2 in Part 2, [31]). For both locomotion tasks described in this part, the controller will attempt to step to a desired step location in the 1-step capture region. If the robot is significantly disturbed in mid-swing, the desired step location will be adjusted so that it always lies in the 1-step capture region. Section IX-B provides a discussion on why we chose to base the controller on the 1-step capture region, as opposed to using an N-step capture region with $N > 1$.

B. Force Control

Force controllable actuators are very useful for controlling a biped to walk smoothly and naturally. These actuators allow for compliant control methods that are forgiving to external forces and unknown terrain. Traditionally, many humanoid robots use high-gain position control to track prescribed joint trajectories using non-backdrivable actuators. This approach typically requires near perfect knowledge of the terrain, a near perfect dynamic model of the robot, and no external forces. When pushed or encountering unexpected terrain, these robots may no longer be able to follow the prescribed joint trajectories, and either a new trajectory must be computed on the fly, or the robot falls.[1]

Our control approach avoids the use of desired joint trajectories, especially for the stance leg. Instead, we use low-impedance feedback controllers that control the fundamental aspects of walking (foot placement and body height, orientation and speed), rather than attempting to rigidly control each degree of freedom. Force control provides some robustness to rough terrain since the exact foot/ground contact configuration is less important than the interaction forces between the feet and the ground. In addition, force controllable actuators simplify control of the CoP location, and allow compliant control techniques, such as Virtual Model Control.

C. Virtual Model Control

Virtual Model Control is a tool that allows a designer to control a robot by choosing virtual components, such as springs and dampers, to intuitively achieve task goals [18], [32]–[35]. Once these components are chosen, the kinematic model of the system and additional user defined constraints allow direct calculation of the joint actuation required to simulate the effect of the desired components.

For example, in previous work on Spring Turkey, a planar walking robot with 4 actuated degrees of freedom, we decomposed the requirements of walking and designed simple virtual components to achieve each one [18]. CoM height was maintained by a virtual vertical spring-damper "granny walker" and forward travel was achieved with a virtual "track bunny" with constant forward velocity connected to the robot's body by a virtual horizontal damper. The joint actuation was calculated using the transpose of a Jacobian that spans the joints between two virtually controlled points. While kinematic singularities remove degrees of freedom from the possible actuation, the joint torques for the remaining degrees of freedom can still be calculated. Similar techniques were used in the control of Spring Flamingo, a planar walker with 6 actuated degrees of freedom. Virtual Model Control is used in the control of M2V2 to maintain CoM height and body orientation, and to achieve approximate CoP control.

VII. Controller Implementation

We now present a detailed description of the balancing and walking controller. The controller's input is comprised of joint angles and angular velocities, and the orientation and orientation rates of the upper body. The controller's output is comprised of desired torques at each joint. The controller roughly consists of five parts:

1) the state machine, which keeps track of the gait phase and acts as a supervisory system that calls the appropriate lower level routines (Section VII-A).
2) the capture region calculator, which determines the instantaneous capture point and the 1-step capture region (Section VII-B).

[1]Trajectory tracking bipedal robots may use compliant foot pads, force sensors, and real time modification of ankle trajectories in order to control the CoP location. This technique essentially converts the ankle actuators to lower impedance force controllable actuators.

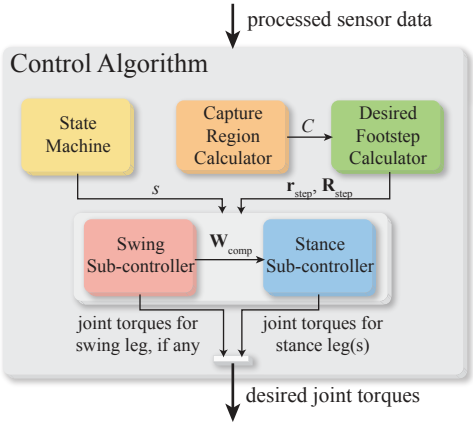

Fig. 3. High level overview of the control architecture. Arrows represent data flow. Arrows that point to the edge of a block signify that the corresponding information is available to all subblocks. The state machine produces the controller state s. The capture region calculator approximates the 1-step capture region C, represented as a 2D convex polygon on the ground. The desired footstep calculator uses the capture region to compute a step location $\mathbf{r}_{\text{step}}$ and orientation $\mathbf{R}_{\text{step}}$. The swing sub-controller computes torques for the swing leg if the robot is in a single support state. It also produces a wrench $\mathbf{W}_{\text{comp}}$ that the stance sub-controller uses to compensate the swing motion. The stance sub-controller computes the torques for both legs in double support, or just the stance leg in single support.

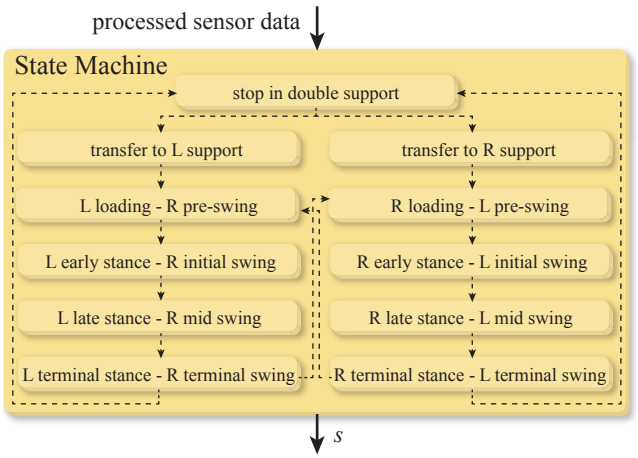

Fig. 4. The state machine, which produces the controller state s. Blocks represent states and dashed lines represent available state transitions. When the control algorithm is started, the robot is in the 'stop in double support' state.

3) the desired footstep calculator, which determines where to step to next (Section VII-C).
4) the swing sub-controller, which computes the torques for the swing leg joints, if any (Section VII-D).
5) the stance sub-controller, which computes the torques for the stance leg joints (Section VII-E);

See Fig. 3 for a high level overview of the control architecture. The same state machine structure is used for both control tasks (balancing and walking), but with different control actions and transition conditions for each task. These control actions and transition conditions are defined by the swing sub-controller and the stance sub-controller. While these sub-controllers are different for each task, they reuse many of the underlying control modules. The desired footstep calculator is also task-specific.

The capture region calculator is task independent and is the module most linked with capturability-based analysis and control. This module may also be useful in other legged robot control architectures and with other walking control techniques.

A. State Machine

The state machine structure, shown in Fig. 4, is based on the gait phases that a *single leg* goes through during human walking, as described in the biomechanics literature [36]. See Fig. 5 for a graphical depiction of these gait phases. The gait phases can roughly be grouped into stance phases and swing phases. Below we provide a short description of each gait phase. See [36] for more detailed descriptions.

1) Stance phases: During *loading response*, the shock of initial ground contact is absorbed by bending the knee and using the heel as a rocker. The leg is loaded, while the trunk is kept upright. Once the opposite leg is lifted, the robot transitions into *mid stance*, in which the CoM moves forward over the stance foot as the leg is straightened. The robot transfers into *terminal stance* when the CoM is aligned over the forefoot. During this state, the heel rises and the knee is first straightened further and then begins to flex slightly.

2) Swing phases: When the opposite leg makes contact with the ground, the robot transitions into *pre-swing*. The leg is unloaded and bent more in preparation of the swing phase. *Initial swing* begins as the foot lifts off the floor. Foot clearance is achieved and the leg is swung forward. When the swinging limb is opposite the stance limb, the robot enters *mid swing*. The hip is flexed further and the knee is allowed to extend in response to gravity. Finally, the robot transitions into *terminal swing*, in which the knee is extended as limb advancement is completed.

Transitions between gait phases for the right leg are directly coupled to those for the left leg. The states shown in Fig. 4 were hence created by combining one gait phase for the left leg and one gait phase for the right leg, *e.g.* 'left early stance - right initial swing'. In addition to the eight walking states, there is also a state in which the robot is stopped in double support and states in which weight is transferred to one leg, allowing the robot to start from a stop.

While this state machine is based on walking, it is easily adapted to one-legged balance, through appropriate selection of control actions in each state and transition conditions between states.

B. Capture Region Calculator

Below is a step-by-step description of how we approximate the 1-step capture region, C. This description is based on Part 2, using the 3D-LIPM with finite-sized foot. It assumes that the ground is flat, but that it may contain keep-out regions. We assume that the support polygon, the reachable region of the swing leg, and the available foothold regions are all represented as polygons.

- Determine the convex support polygon on the ground.

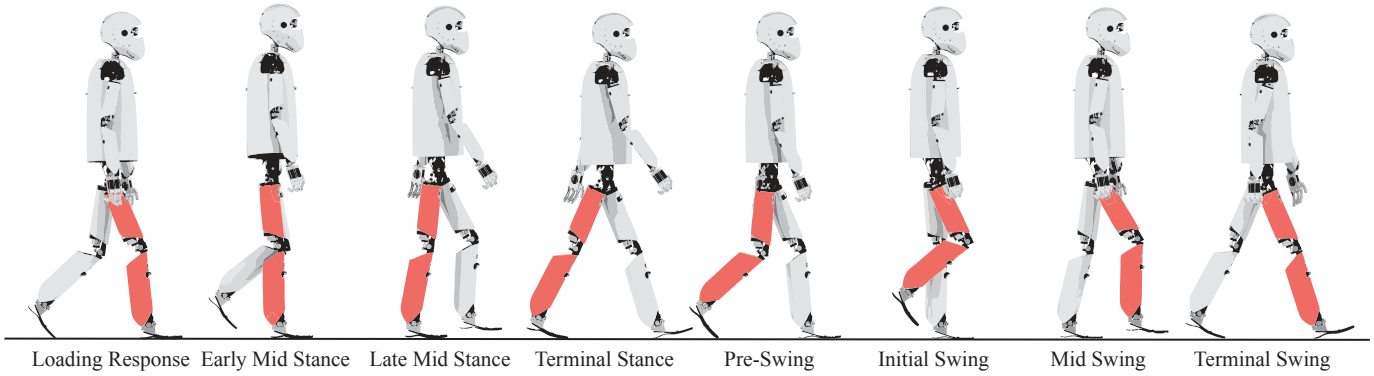

Fig. 5. The phases of gait upon which the state machine for the walking task is based. The labels correspond to the phase of the highlighted right leg. Adapted from [36], which describes the phases of gait for human walking.

- Determine the reachable region of the swing leg as a set of polygons on the ground.
- Determine the available foothold region as a set of polygons on the ground.
- Given the CoM position, and CoM velocity, determine the instantaneous capture point.
- If the instantaneous capture point is inside the convex support polygon, then the 1-step capture region is the entire reachable region of the foot intersected with the available foothold region.
- Otherwise, find the set of vertices visible from the instantaneous capture point, from the subset of support polygon vertices. A visible vertex is one in which a line segment can be drawn from the instantaneous capture point to the visible vertex without intersecting the interior of the support polygon.
- From the set of visible vertices, find the two line of sight vertices. These are the furthest outside vertices and have the property that a ray from the instantaneous capture point through a line of sight vertex would not intersect the interior of the support polygon.
- For each visible vertex, find a corresponding projected point, such that if the CoP where placed at the visible vertex, the instantaneous capture point would move from its current location to the projected point after the minimum swing time remaining.
- For each of the two line of sight vertices, find a capture region boundary ray by forming a ray from the line of sight vertex and through the projection point, and removing the line segment from the line of sight vertex to the projection point.
- Form the "unlimited foothold and reach capture region" by combining the projection points and the two capture region boundary rays, and then enlarge this region to account for the size of the swing foot.
- Form the "unlimited foothold capture region" by intersecting the unlimited foothold and reach capture region with the reachable region of the swing leg.
- Form the approximated capture region by intersecting the unlimited foothold capture region with the available foothold region.

C. Desired Footstep Calculator

The desired footstep calculator (see Fig. 6) determines a desired footstep, consisting of the desired position of the swing ankle $\mathbf{r}_{\text{step}}$ and orientation of the swing foot $\mathbf{R}_{\text{step}}$ at the end of the upcoming step. The desired position and orientation are expressed in a frame fixed to the supporting foot. Because footstep planning depends greatly on the control task, we have created two separate implementations of this module: one for the balancing task and another for the walking task. The general pattern used for both implementations is to choose a good initial footstep $(\mathbf{r}_{\text{step,init}}, \mathbf{R}_{\text{step,init}})$ at the start of the swing phase and adjust it during the swing phase if necessary, e.g. if the robot is significantly perturbed, to obtain the final output $(\mathbf{r}_{\text{step}}, \mathbf{R}_{\text{step}})$.

1) Balancing: The desired footstep calculator for the balancing task is very rudimentary. The initial desired step location is computed using a fixed distance along a ray that starts at the centroid of the support foot and points in the direction of the instantaneous capture point. The initial orientation is chosen to be the same as the stance foot orientation and is never adjusted. The direction in which the robot steps is adjusted according to changes in instantaneous capture point location during the first 0.1s after the instantaneous capture point has left the foot polygon. After that it remains fixed for the remainder of swing.

2) Walking: A desired footstep for the walking task should be chosen in such a way that forward progression is made, while the robot maintains 1-step capturability. The initial step length is determined to match a desired walking velocity while the initial step width is set to a constant value. On flat ground, the step height is set to zero but may be changed to any feasible desired value when necessary, for instance to climb a slope. Footstep yaw is set equal to the desired walking direction. Footstep pitch and roll are set to zero on flat ground but may be used to walk on rough terrain to reduce the need for compliance in the joint. More details on the initial footstep calculation will be published in [37]. If the initial desired step location ceases to be well within the capture region at any instant during the swing phase, an adjusted desired step location is computed by projecting the initial location inside the 1-step capture region by a margin. The orientation is not

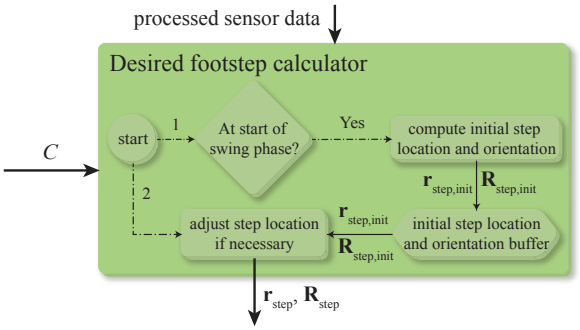

Fig. 6. General behavior of both implementations of the desired footstep calculator. The step location calculator determines the desired position $\mathbf{r}_{\text{step}}$ and orientation $\mathbf{R}_{\text{step}}$ of the foot for the upcoming step. Solid lines represent data flow and dash-dotted lines represent flow of control. The numbers 1 and 2 signify order of evaluation.

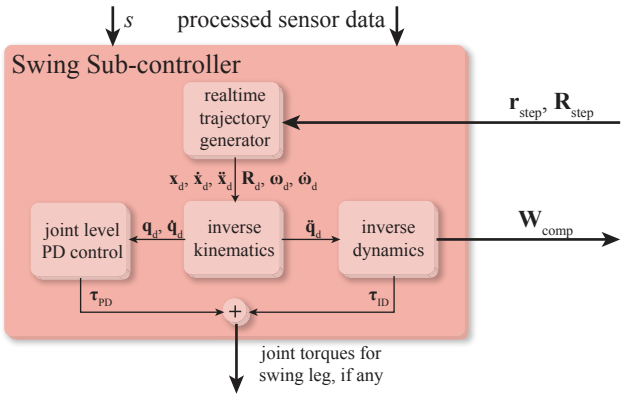

Fig. 7. Swing sub-controller. A real-time trajectory generator computes the desired position, linear velocity and acceleration ($\mathbf{x}_{\text{d}}$, $\dot{\mathbf{x}}_{\text{d}}$, $\ddot{\mathbf{x}}_{\text{d}}$ respectively), as well as the desired orientation, angular velocity and angular acceleration ($\mathbf{R}_{\text{d}}$, $\boldsymbol{\omega}_{\text{d}}$, $\dot{\boldsymbol{\omega}}_{\text{d}}$ respectively), expressed in world frame. Inverse kinematics is used to translate this trajectory from Euclidean space to desired joint positions, velocities and accelerations ($\mathbf{q}_{\text{d}}$, $\dot{\mathbf{q}}_{\text{d}}$ and $\ddot{\mathbf{q}}_{\text{d}}$ respectively). This information is used by an inverse dynamics algorithm, which computes pin joint torques $\boldsymbol{\tau}_{\text{id}}$ and an upper body compensation wrench $\mathbf{W}_{\text{comp}}$, used by the stance sub-controller. The output $\boldsymbol{\tau}_{\text{PD}}$ of a joint-space PD controller is added to $\boldsymbol{\tau}_{\text{ID}}$ to obtain the swing leg joint torques.

adjusted.

D. Swing Sub-controller

The task of the swing sub-controller (see Fig. 7) is to compute the desired torques for the swing leg joints. It can be considered fairly conventional, except for the fact that it can handle changes in the desired step location, obtained from the desired step location calculator, on the fly. It contains a trajectory generator for the position and orientation of the swing foot, which is used to compute desired joint angles, velocities and accelerations through inverse kinematics. The trajectory is tracked using inverse dynamics, augmented by PD position control in joint space.

1) Trajectory generation: To allow for changes in step location in reaction to external disturbances, it should be possible to dynamically update the swing foot trajectory, even in mid-swing. We have chosen to use a simple second order dynamic system which determines the desired linear position $\mathbf{x}_{\text{d}} \in \mathbb{R}^3$ and velocity $\dot{\mathbf{x}}_{\text{d}}$ of the ankle of the swing foot at each control time step by integrating an appropriately chosen desired acceleration $\ddot{\mathbf{x}}_{\text{d}}$. Initial values for $\mathbf{x}_{\text{d}}$ and $\dot{\mathbf{x}}_{\text{d}}$ are set equal to the actual position and velocity of the swing foot ankle.

The dynamic system goes through three phases: take-off, cruise, and landing. Limits are placed on the magnitudes of acceleration and velocity, and we specify a clearance height and a take-off and landing slope. During the take-off phase, the desired acceleration $\ddot{\mathbf{x}}_{\text{d}}$ is chosen to have maximal magnitude until the velocity limit is reached, and be directed at the current desired step location $\mathbf{r}_{\text{step}}$, while moving upwards using the specified take-off slope. Once the minimum clearance height is reached, the dynamic system transitions into the cruise phase, in which $\ddot{\mathbf{x}}_{\text{d}}$ is chosen to maintain constant height while still adjusting for changes in the desired step location. The transition into the landing phase takes place based on the landing slope, and the desired position decelerates maximally while the height is gradually reduced according to the landing slope.

For the orientation trajectory of the swing foot, we use a simple interpolation between the measured orientation at the start of the swing phase and the desired final foot orientation, $\mathbf{R}_{\text{step}}$. The interpolation parameter is obtained from a quintic spline starting and ending with zero velocity and acceleration, and is not dynamically updated.

2) Position control: The desired position and orientation at every control time step are used to compute the corresponding joint angles using inverse kinematics. The desired linear and angular velocity and acceleration of the swing foot specify corresponding desired joint velocities and accelerations through the inverse of the swing leg Jacobian.

An inverse dynamics algorithm [38], augmented by PD position control in joint space is used to compute the desired torques for the swing leg joints. Problems due to the singularity that occurs when the knee is stretched are circumvented by gradually scaling the desired joint velocities and accelerations back to zero (pure damping) based on the value of the Jacobian determinant. We omit the stance leg joints in computing the inverse dynamics. The desired spatial acceleration of the upper body is set to zero. In addition to the torques across the swing leg pin joints, the inverse dynamics algorithm also returns a wrench that should be exerted across the 'floating joint' that connects the upper body to the world in order to achieve the desired zero spatial acceleration. This wrench will be used in the stance sub-controller as a feed-forward term to compensate the swing leg torques and reduce upper body oscillations.

Previously, we have also implemented a Virtual Model Control-based swing sub-controller, but the current implementation outperformed the Virtual Model Control implementation in terms of accuracy and swing speed, which we consider to be key ingredients for dynamic walking and push recovery. Future work will include swing techniques that are less dependent on joint space tracking.

E. Stance Sub-controller

The stance sub-controller controls balance by computing desired torques for the stance leg(s). The goals of the stance sub-controller are to control 1) instantaneous capture point location, 2) upper body orientation, and 3) upper body height. It does so by first computing a desired wrench on the upper body that satisfies these control goals, and then using Jacobian transpose control to find desired leg torques that realize this desired wrench. The stance sub-controller is an implementation of Virtual Model Control. It consists of multiple control modules, as shown in Fig. 8. The following sections will describe these modules in more detail.

1) Instantaneous Capture Point Control Module: The goal of the instantaneous capture point control module is to regulate the location of the instantaneous capture point by determining the desired position of the CoP. This control module switches between two modes of operation, depending on whether the instantaneous capture point is inside the support polygon.

When the instantaneous capture point is inside the support polygon, a *desired instantaneous capture point* is determined as a function of the state and the control task. For the balancing task, the desired instantaneous capture point coincides with the centroid of the support polygon during the double support state. When the robot is commanded to start balancing on one leg, the desired instantaneous capture point is moved to the centroid of the upcoming support foot, where it remains as long as the robot is able to maintain its balance without taking a step. This location maximizes robustness against external disturbances from unknown directions. For the walking task, the desired instantaneous capture point is located near the toes of the leading foot during the double support states, promoting forward motion. At the start of the swing phase, the desired capture point is moved outside the stance foot in the direction of the upcoming step location.

Given the current location of the instantaneous capture point, $\mathbf{r}_{\text{ic}}$, and the desired location $\mathbf{r}_{\text{ic,des}}$, a simple proportional control law with an added feed forward term is used to obtain the tentative location of the desired CoP, $\bar{\mathbf{r}}_{\text{CoP,des}}$:

$$\bar{\mathbf{r}}_{\text{CoP,des}} = \mathbf{r}_{\text{ic}} + k_{\text{ic}}(\mathbf{r}_{\text{ic}} - \mathbf{r}_{\text{ic,des}}) \tag{1}$$

where k_{ic} is the proportional gain. This proportional control law is motivated by the linear instantaneous capture point dynamics for the 3D-LIPM with finite-sized foot described in Part 2. The tentative desired CoP $\bar{\mathbf{r}}_{\text{CoP,des}}$ is then projected onto the edge of the support polygon if it lies outside to obtain the final output $\mathbf{r}_{\text{CoP,des}}$.[2]

When the instantaneous capture point is not inside the support polygon (*i.e.* when the robot is pushed significantly or after the instantaneous capture point has been driven outside the stance foot polygon during the walking task), it is not possible to track a desired location, since the instantaneous capture point will always exponentially diverge away from the stance foot. We therefore only control the direction in which it moves away from the foot. This is done by specifying a *guide line*, along which the instantaneous capture point should move. The guide line is defined by the final location of the desired CoP before the capture point moves outside the support polygon, and the desired step location. Finally, the desired CoP is computed by finding the intersection of the stance foot polygon and a virtual *control line*, which is parallel to the guide line as shown in Fig. 9a. The distance $d(\cdot,\cdot)$ between the guide line L_{g} and the control line L_{c} is determined as

$$d(L_{\text{g}}, L_{\text{c}}) = k_{\text{ic}} d(L_{\text{g}}, \mathbf{r}_{\text{ic}}). \tag{2}$$

2) Virtual Toe Point Calculator: The virtual toe point calculator uses the desired CoP to compute the *leg support fractions* and a *virtual toe point* [27] for each leg.

The leg support fractions are two scalars, denoted w_0 and w_1, in the interval $[0, 1]$ that sum to one, and describe which fraction of the desired wrench on the upper body is to be exerted by each stance leg. Virtual toe points are similar to the centers of pressure for each foot, except that a virtual toe point is a commanded quantity, not a measured one, and is only based on a static analysis. The virtual toe point on a foot is the point about which no torque is commanded. Details on how the virtual toe points are used are given in Section VII-E5. Controlling virtual toe point locations and leg support fractions results in approximate control of the overall CoP of the robot.

In single support, the virtual toe point for the stance leg is placed at the location of the desired CoP and the leg is assigned a leg support fraction of 1. In double support, we use a heuristic based on geometric relations to determine a virtual toe point $\mathbf{r}_{\text{vtp},i}$, for foot $i \in \{0, 1\}$ in such a way that the desired CoP and both virtual toe points lie on one line, as

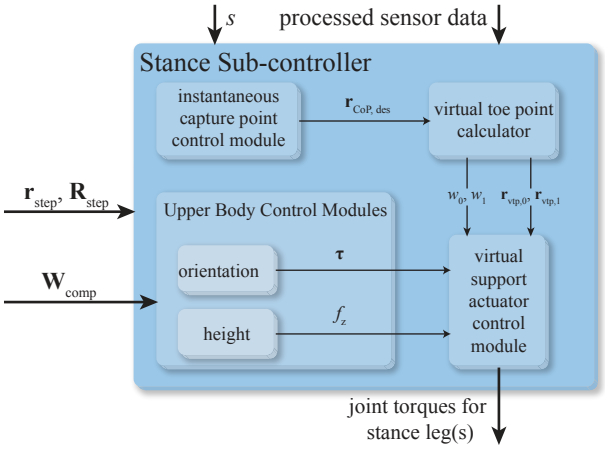

Fig. 8. Stance sub-controller. The instantaneous capture point control module computes the desired CoP $\mathbf{r}_{\text{CoP,des}}$ within the base of support. This desired CoP is used by the virtual toe point calculator to compute leg support fractions (w_0, w_1) and virtual toe points ($\mathbf{r}_{\text{vtp},0}$, $\mathbf{r}_{\text{vtp},1}$), which determine each leg's contribution to supporting the upper body. The orientation and height upper body control modules determine the torque $\boldsymbol{\tau}$ and the z-component of the force f_z to be exerted on the upper body. In single support, these modules use the swing leg compensation wrench $\mathbf{W}_{\text{comp}}$ to compensate swing leg motion. Finally, the virtual support actuator control module computes joint torques for the stance legs which result in the desired virtual toe points, leg strengths, upper body torque and force.

[2]In practice, we use a slightly smaller version of the support polygon when projecting the tentative desired CoP, in order to prevent the feet from tipping at times when this is not desired. This is necessary because of unmodeled dynamics, inability to perfectly track the desired CoP, and model uncertainty.

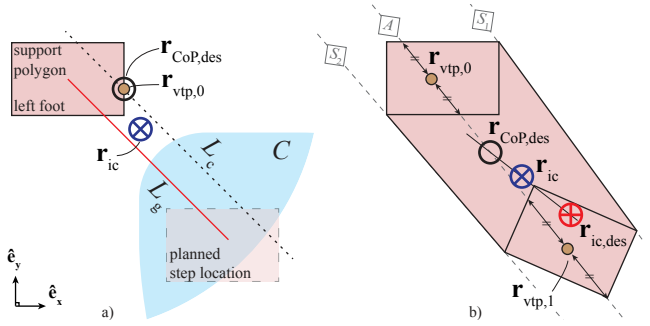

Fig. 9. Schematic overhead view of the robot, showing the workings of the stance subcontroller. **a)** Typical situation during single support when the instantaneous capture point has left the foot. A *guide line*, L_g, along which the instantaneous capture point should move, is computed based on the planned step location. The distance between the guide line and a parallel *control line*, L_c, is determined using (2). The desired CoP is chosen as the intersection of the control line and the support polygon. Since there is only one support leg, the locations of the virtual toe point and desired CoP are identical. **b)** Typical situation during double support. The desired CoP is chosen so that it pushes the instantaneous capture point towards its desired location, typically set to the inside of the future stance foot. Line A intersects $\mathbf{r}_{\text{CoP,des}}$ and its orientation is gradually morphed from that of line S_1 to that of line S_2. Virtual toe points $\mathbf{r}_{\text{vtp},0}$ and $\mathbf{r}_{\text{vtp},1}$ are then computed using geometric relations as shown in the figure.

explained in Fig. 9b. The distance between these points and the overall desired CoP determines the leg support fraction:

$$w_i = \frac{d(\mathbf{r}_{\text{CoP,des}}, \mathbf{r}_{\text{vtp},1-i})}{d(\mathbf{r}_{\text{vtp},0}, \mathbf{r}_{\text{vtp},1})} \tag{3}$$

Support is typically gradually transferred from one leg to the other in double support, corresponding to continuously changing leg support fractions.

3) Upper Body Height Control Module: The upper body height control module determines the vertical force part, f_z, of the desired wrench on the upper body, and controls the upper body height. In double support, the force is set to a constant value that slightly overcompensates the estimated gravitational force acting on the entire robot, simulating a virtual constant force spring. In single support, the force is set to the sum of the weight of the stance leg and the z-component of the force from the swing leg compensation wrench $\mathbf{W}_{\text{comp}}$, thus cancelling out some of the dynamic effects due to the swing leg motion.

4) Upper Body Orientation Control Module: The upper body orientation control module determines the torque part, $\boldsymbol{\tau}$, of the desired upper body wrench, and is used to control the orientation of the upper body with respect to the world (as perceived by the inertial measurement unit). The desired pitch of the upper body is constant and set to zero. Both the desired yaw and the desired roll depend on the gait phase and state of the support leg. Yawing and rolling are used in order to obtain a longer reach for the swing leg and to make the gait look more humanlike. In addition, the desired yaw also depends on the desired walking direction. The x, y and z components of the desired upper body torque are computed using PD control on the roll, pitch and yaw corresponding to the rotation matrix that describes the orientation of the actual upper body with respect to the desired orientation, respectively. This PD control scheme can be viewed as a set of virtual torsional springs and dampers. In single support, the torque part from the swing leg compensation wrench $\mathbf{W}_{\text{comp}}$ is added to the result, to compensate the swing leg motion.

5) Virtual Support Actuator Control Module: The upper body height- and orientation control modules provide a partial desired wrench on the upper body, consisting of the z-component of the force, f_z, and the torque $\boldsymbol{\tau}$, expressed in an upper body-fixed frame. The virtual support actuator control module distributes this partial wrench over the support leg(s) using the leg support fractions w_i as weighting factors:

$$\begin{aligned} f_{z,i} &= w_i f_z \qquad i \in \{0,1\} \\ \boldsymbol{\tau}_i &= w_i \boldsymbol{\tau} \end{aligned} \tag{4}$$

where $f_{z,i}$ and $\boldsymbol{\tau}_i$ are the z-component of the force and the torque to be exerted by leg i, respectively. We aim to compute a complete wrench $\mathbf{W}_i$ for each leg, where

$$\mathbf{W}_i = \begin{pmatrix} \mathbf{f}_i \\ \boldsymbol{\tau}_i \end{pmatrix} \text{ with } \begin{aligned} \mathbf{f}_i &= (f_{x,i}, f_{y,i}, f_{z,i})^T \\ \boldsymbol{\tau}_i &= (\tau_{x,i}, \tau_{y,i}, \tau_{z,i})^T . \end{aligned} \tag{5}$$

The remaining x- and y-components of the wrenches for each leg are computed using the virtual toe points. For leg i, joint torques that realize the wrench W_i will then be computed using Jacobian transpose control.

The virtual toe point constraint, which states that no torque should be commanded about the horizontal axes at the virtual toe point, can be enforced as follows. We consider the virtual toe point for a foot to be the intersection of the axes of two virtual pin joints, located on the sole of the foot. Their orthogonal axes of rotation lie in the plane of the foot. The virtual pin joints do not exist on the real robot, but provide a simple way of computing the x- and y-components of the force on the upper body that satisfy the virtual toe point constraint: the torques across these joints should be zero. The virtual pin joints come after the real joints of the robot in the kinematic chain from upper body to foot. Their rotation angles are set to zero, but their location on the foot changes in time, depending on the location of the virtual toe point. We use $\boldsymbol{\tau}_{\text{vtp},i} \in \mathbb{R}^2$ to denote the vector of virtual joint torques exerted at the virtual pin joints. Considering stance leg i, a static analysis results in

$$\boldsymbol{\tau}_{\text{vtp},i} = \mathbf{J}_{\text{vtp},i}^T \mathbf{W}_i. \tag{6}$$

In this equation, $\mathbf{J}_{\text{vtp},i} \in \mathbb{R}^{6\times 2}$ is the Jacobian that maps the joint velocities of the virtual pin joints to the twist of the foot with respect to a virtual body attached 'after' the virtual pin joints in the kinematic chain, expressed in an upper body-fixed frame.

Splitting the Jacobian $\mathbf{J}_{\text{vtp},i}$ into a 2×2 block $\mathbf{J}_{\text{vtp},i,2\times 2}$ and a 4×2 block $\mathbf{J}_{\text{vtp},i,4\times 2}$ and using (5), we can rewrite (6) as

$$\boldsymbol{\tau}_{\text{vtp},i} = \mathbf{J}_{\text{vtp},i,2\times 2}^T \begin{pmatrix} f_{x,i} \\ f_{y,i} \end{pmatrix} + \mathbf{J}_{\text{vtp},i,4\times 2}^T \begin{pmatrix} f_{z,i} \\ \boldsymbol{\tau}_i \end{pmatrix} \tag{7}$$

We require that the torques at a leg's virtual toe point be zero. We can then solve (7) to find the values of $f_{x,i}$ and $f_{y,i}$:

$$\begin{pmatrix} f_{x,i} \\ f_{y,i} \end{pmatrix} = -\mathbf{J}_{\text{vtp},2\times 2}^{-T} \mathbf{J}_{\text{vtp},2\times 4}^T \begin{pmatrix} f_z \\ \boldsymbol{\tau}_i \end{pmatrix} \tag{8}$$

Now that $f_{x,i}$ and $f_{y,i}$ are also known, we know the complete wrench $\mathbf{W}_i$ to be exerted on the upper body by stance leg i, and we can use a different Jacobian, $\mathbf{J}_{\text{leg},i} \in \mathbb{R}^{6\times 6}$ to find the joint torques $\boldsymbol{\tau}_{\text{leg},i}$:

$$\boldsymbol{\tau}_{\text{leg},i} = \mathbf{J}_{\text{leg},i}^T \mathbf{W}_i \tag{9}$$

where $\mathbf{J}_{\text{leg},i}$ maps the joint velocities of the *real* joints of leg i to the twist of the upper body with respect to the foot, expressed in the upper body-fixed frame.

Computing $f_{x,i}$ and $f_{y,i}$ based on virtual toe points instead of specifying these forces directly has as an advantage that virtual toe points are closely related to the CoP, which plays a major role in the instantaneous capture point dynamics described in Part 2. This relation to the CoP also means that limits due to the finite-sized support polygon are easier to take into account. We simply make sure that each foot's virtual toe point lies inside its convex polygon.

VIII. Results

This section presents results obtained for both the balancing task and the walking task. Balancing and walking without pushes was achieved on the real M2V2 robot. Walking while recovering from pushes and walking over stepping stones was achieved on the simulated M2V2 robot. Note that figures are labeled either [REAL] if the data is from the real robot or [SIM] if from the simulated robot. A video of results from both real-world experiments and simulations is available online with the paper.

In this section, we use a slightly modified version of the capturability margin introduced in Parts 1 and 2. Instead of using the area of the ∞-step capture region, we use the area of the 1-step capture region. The reason for this is that the entire algorithm is based on 1-step capturability.

A. Balancing task

On the real M2V2 robot we achieved balancing on one leg and recovering from sideways and forward pushes. Fig. 10 shows time-elaspsed images of M2V2 recovering from a push. Fig. 11 shows the norm of the disturbance force, as recorded from the push stick, and the capture margin, measured as the area of the approximated 1-step capture region. We see that the robot was able to recover from pushes of approximately 21 Ns.

B. Walking task

On the real M2V2 robot we achieved flat ground walking without disturbances. On the simulated M2V2 robot we achieved walking while recovering from pushes and walking over stepping stones.

Fig. 12 shows time elapsed images of M2V2 walking on flat ground. For this walk, the robot uses a constant step length and width. During the walk, the capture region is computed, but since there are no pushes, the robot does not have to change where it steps.

Fig. 13 through 16 show plots of a single data set obtained from simulation for walking on flat ground while recovering

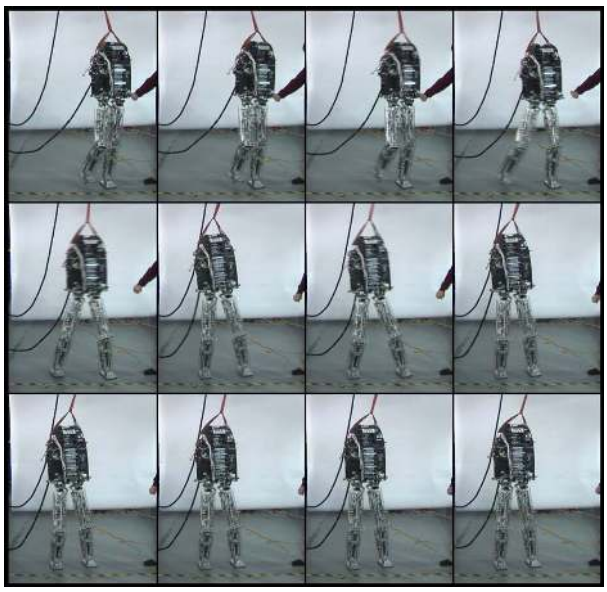

Fig. 10. [REAL] M2V2 recovering from a push while standing on one leg. Images are from left to right starting at the top left.

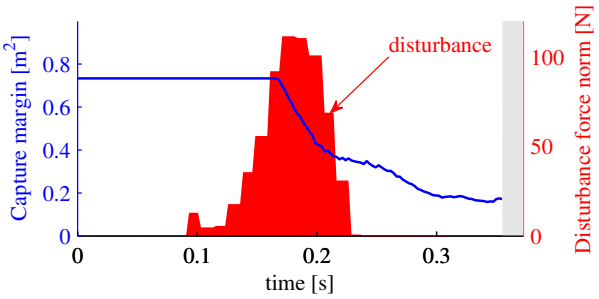

Fig. 11. [REAL] Norm of disturbance force and capture margin of M2V2 recovering from a push while standing on one leg. The capture margin is not shown after transition to the 'stop in double support' state (gray area) to avoid cluttering.

from pushes. Three different pushes to the left occur at approximately 4, 8.5, and 13 seconds. These pushes are modeled as forces applied to the midpoint between the hip joints and are 300 N in magnitude for a duration of 0.05 seconds in various directions. This corresponds to an impulse of 15 Ns. Note that pushes were to the left while the left foot was swinging. Pushes to the opposite side would require either a cross-over step or two quick steps, both of which are more difficult to achieve and an area of future work.

Fig. 13 shows side and overhead views of the robot. Each time a push occurs, the robot steps to the left to recover from the push. Also plotted in the overhead view are the CoM path and the paths of each ankle.

Fig. 14 shows a time-lapsed overhead view of the robot. In the first two frames the capture region is quite large during the beginning of swing. The robot gets pushed between the second and third frame, decreasing the size of the capture region and requiring the robot to choose a different place to step. In frames 3-6 the robot steps further to the left than

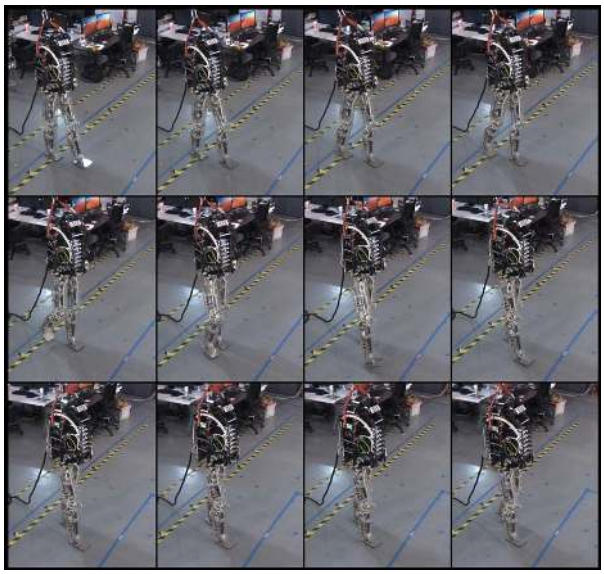

Fig. 12. [REAL] M2V2 robot walking on flat ground. Images are from left to right starting at the top left. In this walk, the robot uses a constant step length and width. The algorithm used is very close to that described in this paper, but slightly older. We are currently working on getting the robot to walk using the exact algorithm described in this paper, but are delayed due to hardware issues.

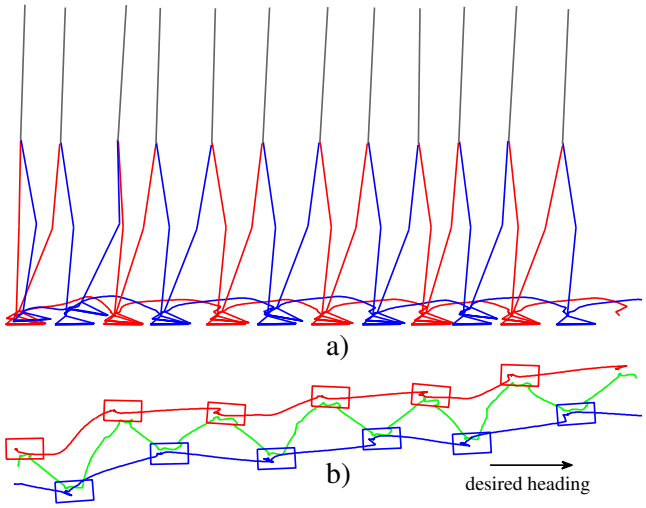

Fig. 13. [SIM] Schematic view of the robot as it is walking while being pushed laterally on every second step. Both a) and b) show the same data. **a)** Side view. Actual trajectories of the ankles as simulated are shown. **b)** Overhead view. Actual ankle trajectories connect the sequence of footprints for each foot. The actual instantaneous capture point trajectory is shown, zig-zagging between the feet.

originally intended, landing in the 1-step capture region, and successfully recovering from the push.

Fig. 15 shows the sideways pushes applied to the M2V2 simulation, and the resulting change in velocity while recovering to pushes during walking. Since the pushes were mostly to the side, the change in lateral velocity is more prominent than forward velocity. After each push we see that the robot recovers with one step.

Fig. 16 shows the 1-step capture margin during walking of

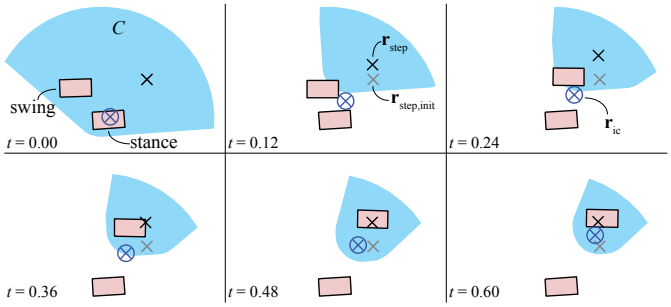

Fig. 14. [SIM] Time-lapse overhead view of the robot during the walking task, showing one step. The robot is perturbed laterally at the start of the step ($t = 0.00$). In addition to the location of both feet, the figure shows the capture region C, the initial step location $\mathbf{r}_{step,init}$, the adjusted step location $\mathbf{r}_{step}$ and the instantaneous capture point $\mathbf{r}_{ic}$.

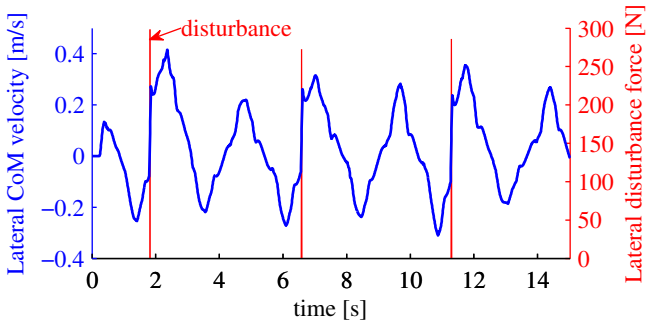

Fig. 15. [SIM] Lateral velocity and disturbances of the M2V2 simulation while recovering from pushes during walking. After each push, we see a change in the velocity up to 0.44 m/s, which then recovers during the next couple steps.

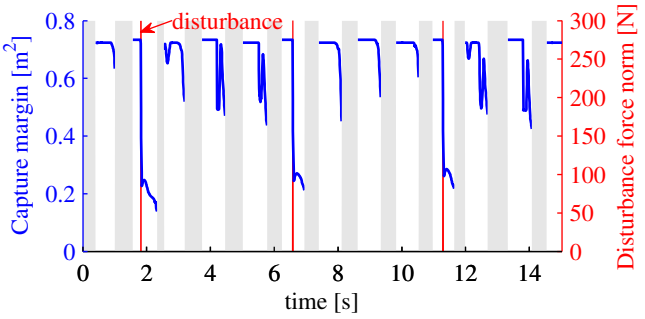

Fig. 16. [SIM] Capture margin during walking while recovering from pushes. The capture margin is not shown in double support phases (gray areas) to avoid cluttering. After each push, the 1-step capture margin significantly decreases showing that the robot is in danger of falling. After each recovery step, the capture margin recovers. Note that during double support and periods during single support when the instantaneous capture point is inside the support polygon the capture margin is not plotted since any reachable point is in the 1-step capture region when the robot is in a captured state.

the M2V2 simulation while recovering from pushes. We see that the capture margin significantly decreases after each of the 3 pushes, corresponding to the decrease in area of the capture region as seen in Fig. 14.

Walking over various stepping stones was achieved on the simulated M2V2 robot. Fig. 17 shows an example of walking over stepping stones that are clustered in groups of three. For

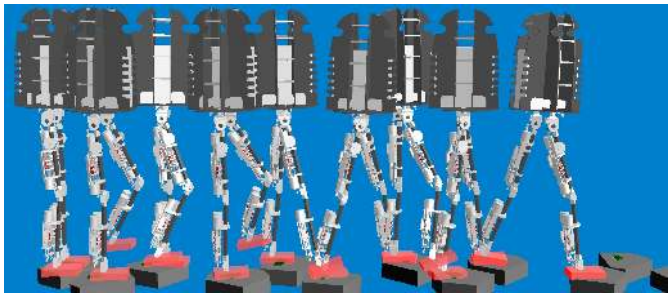

Fig. 17. [SIM] M2V2 walking over stepping stones that are clustered in groups of three. Snapshots are taken at 1.5 second intervals.

stepping stones, the robot was given exact knowledge of the stones. The capture region was computed in the same way as with no stepping stones, and then intersected with the available stepping stones. To ensure that the entire foot rested on each stone, the stepping stones were first "shrunken" based on the size of the foot. However, due to inaccuracies in swing, the foot would sometimes slightly overhang the edge of a stone.

IX. Discussion and Future Work

A. Using Simple Models for Complex Robots

In this part we used the 3D-LIPM with finite-sized foot, described in Part 2, in order to estimate the 1-step capture region, which was then used to help control walking in a complex 3D lower body humanoid. This simple model was sufficient for controlling balancing on one foot, walking, and recovering from pushes while walking. The simplified model was sufficient for several reasons:

- The model accounts for the dynamics of the CoM with respect to the CoP, which are the key dynamics of walking. Other things, like internal angular momentum play a less critical role.
- The model allows for the use of feet, and the modulation of the CoP in real time. This adds robustness, as opposed to an algorithm that predetermines a CoP trajectory.
- The 1-step capture region tends to be fairly large for moderate speed walking with M2V2 parameters, so there is a large degree of robustness to modeling errors.

In this part, none of the control algorithms utilized angular momentum of the upper body. As we start to address more challenging tasks, like walking over rough terrain and over narrow beams, we will likely need to use more complex models for computing capture regions and developing control startegies. In Part 2, we analyzed the 3D-LIPM with finite sized foot and reaction mass. This model should be useful in control when upper body angular momentum is used to prevent falling. In future work we will use this model and will investigate other strategies for using multi-joint upper body angular momentum in walking and disturbance recovery.

B. 1-Step Versus N-Step Capture Regions

In this part we developed controllers that always step into the 1-step capture region. This is overly cautious in general, and it seems as though very fast walking requires periods when the legged system is only 2-step, or perhaps even only 3-step capturable. However, M2V2 currently has a long swing time, making the 2-step capture region not much larger than the 1-step capture region.

For M2V2, the falling time constant is $1/\omega_0 = \sqrt{z_0/g} \approx 0.32$ seconds, where $g = 9.81$ is the gravitational acceleration and $z_0 \approx 1.0$ is the CoM height. The minimum step time, $\Delta t_{s,\min}$ for M2V2 is currently approximately 0.6 seconds. The geometric ratio governing the diminishing returns for the N-step capture region as N increases is $\exp(-\sqrt{z_0/g}\Delta t_{s,\min}) \approx 0.15$ (see (16) in Part 2). This means that the radius of the 2-step capture region is only about 15% larger than the radius of the 1-step capture region. Therefore, there is not much to be gained in considering 2-step capturability over 1-step capturability with M2V2, until we can get the robot to swing its leg more quickly.

With human walking, on the other hand, minimum swing time is approximately 0.3 seconds. This gives a geometric ratio of approximately 0.4. Hence, the 2-step capture region for human walking should be relatively large, and even 3-step capturability should be considered. However, for human walking, diminishing returns leave little to gain for considering beyond 3-step capturability.

C. Capture Margin

We presented an experimental evaluation of a modified version of the capture margin, introduced in Parts 1 and 2, where we used the area of the 1-step capture region instead of the area of the ∞-step capture region. Fig. 11 and Fig. 16 showed a significant decrease in this capture margin when the robot was perturbed, as expected. For both the balancing task and the walking task, the capture margin recovered completely after taking one step.

According to the theoretical considerations presented in Part 1, the capture margin should never increase unless a step is taken or an external force is exerted on the robot. Any other increases in capture margin are due to modeling errors, sensor noise and lack of exact knowledge about when the swing foot hits the ground.

D. Machine Learning

In addition to improving our models, we can apply machine learning in order to improve performance, either through improving estimates of the capture region or by directly improving control policies. In previous work [39] we showed how the balancing task could be improved by learning where to step after a push. Each time a simulated robot was pushed, it attempted to regain balance in a single step. Whenever the step was unsuccessful, a memory was updated to improve where the robot would step if encountering similar states. This memory was initialized using the Linear Inverted Pendulum Model. After a few hundred pushes and learning trials the robot significantly improved its ability to regain balance. Using machine learning to improve walking on M2V2 are two areas for future work.

E. Uneven Ground

All of the models presented in Part 2 and all of the control algorithms presented in Part 3 assumed flat ground, perhaps with keeep-out regions. We have done some preliminary simulations with moderate slopes and relatively small steps and so far it appears that the same models are applicable to these types of terrain. However, for large slopes and steps and very rough ground the models will likely need to be expanded and further control strategies developed. In addition we still need to develop capturabilty-based models and control strategies for situations where hands can push against walls or hold handrails, or where feet can be different slopes to get more interesting ground contact interaction. In these cases concepts like force closure will need to be used rather than just using the CoP on a flat ground.

F. Controlling Velocity versus Coming to a Stop

Capturability measures the ability of a legged system to come to a stop in N steps or fewer. However, we are not usually interested in coming to a stop, but rather maintaining an approximate average speed. In the M2V2 controller we do that by selecting parameters that define how control actions are taken and transition conditions occur, based on functions of the instantaneous capture point and the 1-step capture region. For example, during single support the stance leg CoP is moved to compel the instantaneous capture point to follow a guide line towards the next desired place to step. This modulates the desired CoP perpendicularly to the guide line. To control the velocity of the robot, the desired CoP can be moved along the other axis. To speed up, the desired CoP can be place further back. To slow down, it can be placed more forward. So even though capturability is based on the ability to come to a stop, using tools based on capturability do not *require* the legged system to come to a stop, but instead can be used for such things as controlling velocity.

G. Pushes Requiring Cross Over Steps

In this study, we only considered pushes during single support, which did not require a "cross over" step, i.e., when the robot was supported by its left leg, we never pushed the robot to the left. Cross over steps are challenging for a number of reasons. The swinging leg needs to make sure to not contact the support leg. In order to do that, the path of the leg may be longer, requiring longer swing time. Also, the length of the step will be smaller than the leg can swing to the outside. An alternative is to quickly step straight down with the currently swinging leg and then quickly swing the other leg to prevent a fall. This two step recovery strategy requires extra time to execute and for significant pushes will likely only be successful for robots with a relatively quick swing time, on the order of how fast humans can swing their legs.

H. Virtual Toe Points and Center of Pressure

In the presented control system, the desired CoP is used during control. However, the actual CoP is not exactly achieved. Instead, we use "virtual toe points", which can be interpreted as the attachment points of virtual actuators on the feet. The difference between the virtual toe point and the actual CoP on a foot is typically small. During single support if the vertical force of the virtual actuator equals the weight of the robot and the vertical acceleration is zero, then the two points will be theoretically identical. In simulation, the two points always remain close (within a few cm) during single support. During double support, there can be a large error between the desired and actual CoP, particularly when one of the legs loses kinematic range. For example, if the robot is far enough forward that the hind leg is completely straight, and the desired CoP is on the heel of the hind leg, then the virtual actuator on the hind leg will have a large force and its virtual toe point will be back on the heel. However, since the leg is straight, the actual joint torques that the virtual actuator produces will be low, and the CoP will be much more forward than desired.

One way to get a better match between the desired and actual CoP is to keep the knees of the robot bent to avoid losing kinematic range. However, we wish to avoid that solution since human walking does not rely on bent knees and because it requires unnecessarily high torques at the knees. Another solution, which we will investigate in future work, is to use toe off on the rear leg to better control the CoP during double support. Currently some toe off occurs at the end of the stride, but it is simply the result of the dynamics of the walk, rather than used as a control mechanism.

I. Trajectory Tracking During Swing

In order to swing as fast as possible and as accurately as possible, we are currently using traditional high-gain trajectory tracking techniques on the swing leg. However, we believe that swing can be performed in a more compliant manner and that for the most part the swing can be determined mostly by the passive pendulum dynamics of the leg. Determining swing strategies that allow for fast and accurate steps while exploiting the natural dynamics of the leg is an area of future work.

J. Difficulties with Real Hardware

Despite efforts to make the simulation accurately model the real robot, it is still typically the case that performance in the simulation is better than performance on the real robot. Here we identify a few of the difficulties in making a real robot that make this so.

The algorithm presented, as well as most feedback control algorithms for bipedal walking, rely on a good estimation of the CoM location and velocity, particularly in the horizontal plane. Getting such an estimate on a real robot is difficult for several reasons.

Knowing the CoM projection on the ground requires knowing which way is down. A small error in the perceived orientation of the body can result in a significant error in the CoM projection. For example, if the CoM is at 1 meter and there is a 0.01 radian error in the body orientation, that will result in a 1 cm error of the CoM on the ground. For 3D robots, orientation is typically done through an inertial measurement

unit (IMU), and therefore having a good IMU and related sensor processing is important.

Using kinematics through the leg to estimate the CoM velocity has the problem that one must assume that the foot is not moving. However, if the foot is slipping this will make the CoM seem to be moving in the opposite direction. Integrating an accelerometer to estimate velocity has the problem of accumulating errors. For M2V2 we have used a combination of kinematics and accelerometers. We have not yet determined how accurately we are estimating the CoM velocity, but we believe we can do much better and therefore improve performance on the real robot.

In this paper we showed how foot placement can be used to regain balance after a push. Doing so requires a fast swing that is accurate enough that the foot ends up landing in the capture region. However, due to the use of very compliant Series Elastic Actuators, we have been having difficulty quickly and accurately swinging the leg. The SEAs do a really good job allowing for compliant control, but they make traditional high-gain trajectory tracking challenging. We believe that we can achieve the same good compliant control characteristics of the actuators and better tracking by increasing the stiffness of our series springs. In addition we are investigating other improvements to the M2V2 hardware which will make swing quicker and more accurate. Based on capturability analysis, we believe that improving swing is a key improvement that can be made to improve the performance of a bipedal walking robot.

K. Application to Other Robots

We believe that capturability concepts can be applied to the analysis and control of other legged systems. Estimating capture regions and determining capturability-based robustness metrics should be possible with all legged systems. While we advocate compliant force control for legged robots, most of the techniques described in this paper should also apply to high-gain trajectory tracking robots. Stepping strategies that take into account the capture region should be applicable to any robot that can change where it steps on-the-fly. Directing the instantaneous capture point along a line should be applicable to any robot that can control its CoP location on the ground. We are currently expanding the algorithms presented in this paper and working toward there application on several different humanoid robot platforms.

APPENDIX
ROBOT PARAMETERS

Table I shows the joint layout and inertia parameters of M2V2. Each row represents a joint and its associated rigid body. We use a coordinate system in which x is forward, y is to the left, and z is up. Note that the bottom 6 rows represent the left leg joints and masses (marked with the letter 'L'). The right leg is a mirror image of the left leg, and thus is identical to the left leg except for the y values, which are all the additive inverse. For Pin-type joints, the letter following "Pin" refers to the rotational axis that the joint is aligned with.

ACKNOWLEDGMENT

The authors would like to thank A. Goswami and E. Westervelt for their helpful comments.

M2V2 is a second generation version of the M2 robot, developed by G. Pratt, D. Paluska, J. Pratt, and D. Robinson at the MIT Leg Laboratory. M2V2 was designed by B. Krupp, V. Ragusila, I. Olaru, T. Craig, and J. Pratt. M2V2 electronics and interface software was designed by G. Watkins, S. Emami, T. Hutcheson, J. Pratt, J. Smith, and S. Nayak. M2V2 was assembled and maintained by T. Craig and J. Taylor. The force sensing push stick used in push recovery experiments on M2V2 was designed and constructed by J. Joyner.

A team at Bucknell University, led by S. Shooter and K. Buffinton, have designed an improved the feet and a vision head for the robot. Students involved include M. Kandler, D. Snyder, L. Markison, J. Ricci and C. Hubicki.

Various low level control modules for M2V2 were developed by C. Shake, N. van Nieuwenhuizen, F. Cañas, D. Garg, S. Tamadoni, R. van Doesburgh, T. Koolen, J. Pratt, M. Johnson, and P. Neuhaus. A team from NASA JSC, consisting of J. Braman, D. Gooding, S. Tamblyn, M. Goza and A. Hulse, has also contributed in this area.

M2V2 filming and video editing was performed by W. Howell. Part acquisition and lab management was performed by B. Layton.

Simulation Software and user interfaces were improved by J. Carff, M. Fortenberry, D. Reyes Duran, G. Barr, B. Waxler, and P. DeMonaco.

REFERENCES

[1] M. Vukobratovic and B. Borovac, "Zero-Moment Point — Thirty Five Years of its Life," *Int. J. Human. Robot.*, vol. 1, no. 1, pp. 157–173, 2004.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, and K. Yokoi, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. 2003 IEEE Int. Conf. Robot. Automat.*, Sep. 2003, pp. 1620–1626.

[3] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue, "A Fast Dynamically Equilibrated Walking Trajectory Generation Method of Humanoid Robot," *J. Auton. Robots*, vol. 12, no. 1, pp. 71–82, 2002.

[4] K. Nishiwaki and S. Kagami, "Strategies for Adjusting the ZMP Reference Trajectory for Maintaining Balance in Humanoid Walking," in *Proc. 2010 IEEE Int. Conf. Robot. Automat.* IEEE, May 2010, pp. 4230–4236.

[5] T. Sugihara, "Consistent Biped Step Control with COM-ZMP Oscillation Based on Successive Phase Estimation in Dynamics Morphing," in *Proc. 2010 IEEE Int. Conf. Robot. Automat.* IEEE, May 2010, pp. 4224–4229.

[6] J. E. Pratt and R. Tedrake, "Velocity-based stability margins for fast bipedal walking," in *Fast Motions in Biomechanics and Robotics*, ser. Lecture Notes in Control and Information Sciences, M. Diehl and K. Mombaur, Eds. Springer Berlin Heidelberg, 2006, vol. 340, ch. 14, pp. 299–324.

[7] P.-B. Wieber, "On the stability of walking systems," in *Proc. Int. Workshop Humanoid Hum. Friendly Robot.*, Tsukuba, Japan, 2002, pp. 53–59.

[8] T. McGeer, "Passive walking with knees," in *Proc. 1990 Int. Conf. Robot. Automat.* Cincinnati, OH, USA: IEEE, May 1990, pp. 1640–1645.

[9] S. H. Collins, M. Wisse, and A. Ruina, "A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees," *Int. J. Robot. Res.*, vol. 20, no. 7, pp. 607–615, Jul. 2001.

[10] S. H. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient Bipedal Robots Based on Passive-Dynamic Walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, Feb. 2005.

TABLE I
ROBOT JOINT AND MASS LAYOUT.

Joint	Parent	Offset from Parent [m]			Joint Type	Mass [kg]	COM offset [m]			Moment of Inertia [kg m^2]		
		x	y	z			x	y	z	I_{xx}	I_{yy}	I_{zz}
Root	N/A	0.0	0.0	0.0	Floating	24.0	0.0	0.0	0.0	2.756	2.756	0.380
L Hip Yaw	Root	0.0	0.092	-0.381	Pin-Z	0.2	0.0	0.0	0.0	5.3E-4	5.3E-4	5.3E-4
L Hip Roll	L Hip Yaw	0.0	0.0	0.0	Pin-X	0.0	0.0	0.0	0.0	0.0	0.0	0.0
L Hip Pitch	L Hip Roll	0.0	0.0	-0.029	Pin-Y	4.6	-0.0318	0.006	-0.229	0.044	0.044	0.004
L Knee	L Hip Pitch	0.0	0.0	-0.450	Pin-Y	4.3	-0.025	0.0	-0.229	0.054	0.054	0.003
L Ankle Pitch	L Knee	0.0	0.0	-0.467	Pin-Y	0.11	0.0	0.0	0.0	2.6E-4	2.6E-4	2.6E-4
L Ankle Roll	L Ankle Pitch	0.0	0.0	0.0	Pin-X	0.727	0.0127	0.0	-0.051	3.63E-4	0.002	0.002

[11] F. Tan, C. Fu, and K. Chen, "Biped Blind Walking on Changing Slope with Reflex Control System," in *Proc. 2010 IEEE Int. Conf. Robot. Automat.*, May 2010, pp. 1709–1714.

[12] A. D. Kuo, "Stabilization of Lateral Motion in Passive Dynamic Walking," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 917–930, 1999.

[13] T. Geng, B. Porr, and F. Wörgötter, "Coupling of neural computation with physical computation for stable dynamic biped walking control," *Neural Comput.*, 2005.

[14] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid Zero Dynamics of Planar Biped Walkers," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 42–56, Jan. 2003.

[15] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A Compliant Hybrid Zero Dynamics Controller for achieving Stable, Efficient and Fast Bipedal Walking on MABEL," *Int. J. Robot. Res.*, submitted for publication.

[16] E. R. Westervelt, G. Buche, and J. W. Grizzle, "Experimental Validation of a Framework for the Design of Controllers that Induce Stable Walking in Planar Bipeds," *Int. J. Robot. Res.*, vol. 23, no. 6, pp. 559–582, 2004.

[17] C. Chevallereau, J. W. Grizzle, and C. Shih, "Asymptotically Stable Walking of a Five-Link Underactuated 3-D Bipedal Robot," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 37–50, Feb. 2009.

[18] J. E. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. A. Pratt, "Virtual model control: an intuitive approach for bipedal locomotion," *Int. J. Robot. Res.*, vol. 20, no. 2, pp. 129–143, Feb. 2001.

[19] S. Coros, P. Beaudoin, and M. van de Panne, "Generalized biped walking control," in *Proc. SIGGRAPH 2010*. New York, NY, USA: ACM Press, Jul. 2010, pp. 1–9.

[20] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Proc. 2010 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, Oct. 2010.

[21] S.-H. Hyon, J. Hale, and G. Cheng, "Full-Body Compliant Human-Humanoid Interaction: Balancing in the Presence of Unknown External Forces," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 884–898, Oct. 2007.

[22] S.-H. Hyon, R. Osu, and Y. Otaka, *Integration of multi-level postural balancing on humanoid robots*. IEEE, May 2009.

[23] J. E. Pratt and B. T. Krupp, "Design of a Bipedal Walking Robot," in *Proc. 2008 SPIE*, vol. 69621F, 2008.

[24] D. J. Paluska, "Design of a Humanoid Biped for Walking Research," M. Eng. thesis, Massachusetts Institute of Technology, 2000.

[25] G. A. Pratt, "Legged Robots at MIT: what's new since Raibert?" *IEEE Robot. Autom. Mag.*, vol. 7, no. 3, pp. 15–19, 2000.

[26] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 1995, pp. 399–406.

[27] J. E. Pratt and G. A. Pratt, "Intuitive control of a planar bipedal walking robot," in *Proc. 1998 IEEE Int. Conf. Robot. Automat.*, vol. 3. IEEE, 2002, pp. 2014–2021.

[28] Yobotics, Inc. (2011, Mar.) Yobotics home. legged robots, simulation, actuators. [Online]. Available: http://www.yobotics.com/

[29] R. Featherstone, *Robot dynamics algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1987.

[30] B. V. Mirtich, "Impulse-based dynamic simulation of rigid body systems," Ph.D. dissertation, University of California, Berkeley, 1996.

[31] J. E. Pratt, J. Carff, S. V. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Proc. 2006 IEEE-RAS Int. Conf. Humanoid Rob.* IEEE, Dec. 2006, pp. 200–207.

[32] J. E. Pratt, P. Dilworth, and G. A. Pratt, "Virtual model control of a bipedal walking robot," in *Proc. 1997 IEEE Int. Conf. Robot. Automat.*, 1997, pp. 193–198.

[33] J. J. Hu, J. E. Pratt, C.-M. Chew, H. M. Herr, and G. A. Pratt, "Adaptive virtual model control of a bipedal walking robot," *Proc. 1998 IEEE Int. Joint Symp. Intell. Syst.*, pp. 245–251, 1998.

[34] ——, "Virtual model based adaptive dynamic control of a biped walking robot," *Int. J. Art. Intel. Tools*, vol. 8, no. 3, pp. 337–348, 1999.

[35] J. Chen, J. Cheng, and G. Yu, "Virtual model control of a quadruped walking robot," *J. Shanghai Jiaotong Univ.*, vol. 35, no. 12, pp. 1771–1775, 2001.

[36] J. Perry, *Gait Analysis: Normal and Pathological Function*. Thorofare, NJ, USA: Slack, 1992.

[37] S. Cotton, P. D. Neuhaus, T. Koolen, and J. E. Pratt, "Where Should We Step?" 2011, in preparation.

[38] R. Featherstone, *Rigid body dynamics algorithms*. Boston, MA, USA: Springer US, 2008.

[39] J. R. Rebula, F. Cañas, J. E. Pratt, and A. Goswami, "Learning capture points for humanoid push recovery," in *Proc. 2007 IEEE-RAS Int. Conf. Humanoid Rob.*, 2007.

Jerry Pratt received his Ph.D. degree in Computer Science from MIT in 2000. He is currently a research scientist at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include the analysis and control of bipedal and quadrupedal walking and running, humanoid robots, and exoskeletons. Some of the robotic projects he has been involved in include Spring Turkey, Spring Flamingo, M2, the RoboKnee, tBot, LittleDog algorithms, M2V2, and the IHMC ROAM Exoskeleton.

Twan Koolen received his B.Sc. degree from the faculty of mechanical engineering at the Delft University of Technology in the Netherlands in 2009. He is currently pursuing an M.Sc. degree in mechanical engineering at the Delft Biorobotics Laboratory and working as a Research Associate at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include robustness in legged locomotion, multibody dynamics and robust control.

Tomas de Boer received the M.Sc. degree in mechanical engineering from Eindhoven University of Technology of, Eindhoven, The Netherlands, in 2006. He is currently working toward the Ph.D. degree with Delft University of Technology, Delft, The Netherlands. His current research interests include bioinspired robotics, legged locomotion, mechatronics, (multibody) dynamics and (compliant) control.

John Carff is a research associate at the Institute for Human and Machine Cognition in Pensacola, FL. He received his B.S. and Masters in Computer Science from The University of West Florida. Mr. Carff started working for the Institute as a research assistant in 2003. He currently is the lead developer on a project for enabling micro-air vehicle operations in urban environments through human-machine teaing. Mr. Carff's research interests include user interfaces, Java programming, 3D simulations, and robotics.

John Rebula received his S.B. degree in Mechanical Engineering from MIT in 2006, and is currently in the Ph.D. program in the Mechanical Engineering Department at the University of Michigan. He is interested in analysis, planning, and control of human and robotic locomotion.

Sebastien Cotton received his Ph.D. degree in Robotic from University of Montpellier 2, France in 2010. He is currently a research scientist at the Institute for Human and Machine Cognition in Pensacola, FL. His research interests include modeling and control of legged robots and biomechanical analysis of walking.

Matthew Johnson has worked at the Institute for Human and Machine Cognition in Pensacola Florida since 2002. He received his B.S. in Aerospace Engineering from the University of Notre Dame in 1992 and a M.S. in Computer Science from Texas A&M Corpus Christi in 2001. Prior to working for IHMC, he spent ten years in the Navy flying both fixed and rotary wing aircraft. He has worked on numerous projects including the Oz flight display for reducing the cognitive workload in the cockpit, Augmented Cognition for improving human performance, and several human-robot coordination projects for both NASA and the Department of Defense. Most recently he has worked on the Little Dog project developing walking algorithms for a quadruped robot on rough terrain and development of the Yobotics-IHMC lower body humanoid. Matthews research interests focus on improving performance in human-machine systems and include the areas of teamwork, coordination and human-robot interaction.

Peter Neuhaus received his Ph.D. from U.C. Berkeley in mechanical engineering. He is currently as Research Scientist at the Florida Institute for Human and Machine Cognition in Pensacola, FL. His research interests include robotic orthoses for mobility assistance, medical rehabilitation, and human strength augmentation. His projects include the Human Assisted Walking Machine, DARPA Learning Locomotions LittleDog, PISCES, tBot, and the ROAM.