Provably Stable Walking Gaits for a 3-link Robot with Point Feet

1 Introduction

Walking robots have been a popular science fiction fantasy since the mid 20th century. The control community has been actively working on this problem for several decades. Various methods are used; one of the easiest to implement in practice is a robot with 4-6 legs to solve the balancing problem. Bipedal, human-like walking is much more difficult.

Human walking is fundamentally difficult to emulate because it is a highly dynamic process with many states and little static stability. Many existing bipedal robots get around this problem by altering the gait properties to eliminate the dynamic balancing problem; Honda's Asimo is perhaps the most famous example. This method is termed the "Zero Moment Point" method and is discussed in Section 2.

The goal of this project is to study the bipedal locomotion problem using nonlinear control techniques in a systematic way. Stable walking gaits correspond to stable periodic orbits in the state space. By using a variety of techniques, control is applied to create these orbits, specific orbits are identified, and their stability can be proven. These controllers are then simulated on the full-dimensional robot model and shown to create the desired walking gait.

These methods have been extensively developed and applied to complicated robots [2], including hardware testing on a 5-link robot. The work presented here applies the methods of [2] to a simple 3-link walker; All theorems and methods presented here can be found in [2]. The main techniques used are Lagrangian Dynamics, Poincaré maps, hybrid systems, zero dynamics, and feedback linearization.

The outline of this report is as follows: The robot continuous dynamics and impact map are derived. A simple walking gait is chosen. A periodic orbit is found and proved stable for two different types of controllers. The zero dynamics are studied. Finally, a more complicated, energy-efficient gait is studied.

2 Zero Moment Point Heuristic

The basic idea of the Zero Moment Point Heuristic is to use actuated feet with small, flat-footed steps. In this way, the robot is statically stable at all points throughout a

step and the control problem is simplified. The interaction forces between the robot and the ground are lumped into a single force with no moment acting at the "zero moment point". If one wanted to support the robot with a point force, it must act at this point.

By designing the controller to keep this "zero moment point" within the support polygon of the robot, the robot will be statically stable. With one foot on the ground, the support polygon is the outline of the foot in contact with the ground. With both feet on the ground, it is the area between the outlying ground contact points, the front toe and back heel. This method is graphically illustrated in Figure 1.

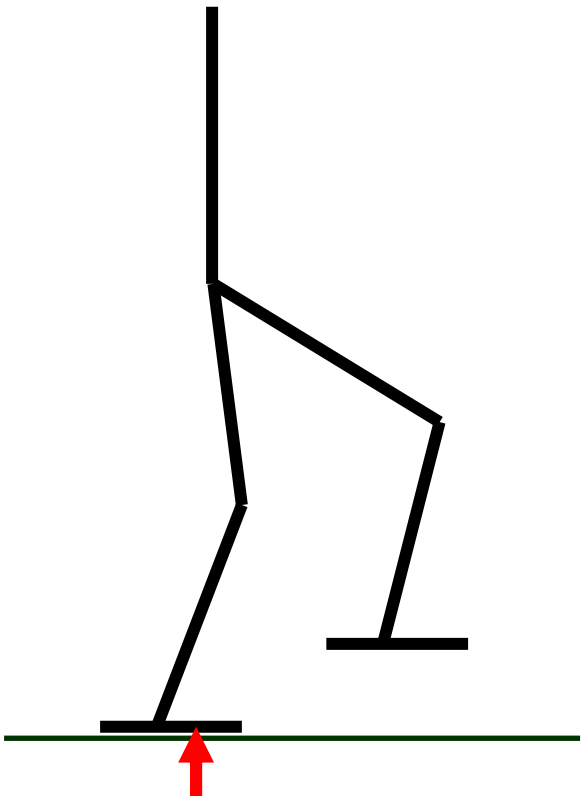

Figure 1: The Zero Moment Point Heuristic. The arrow shows the location of the Zero Moment Point, where the forces on the foot sum to zero moment.

This method is based on static stability, so it works for slow, near-static gaits. For dynamic gaits, the results break down. This method does not provide a rigorous stability proof for dynamic gaits.

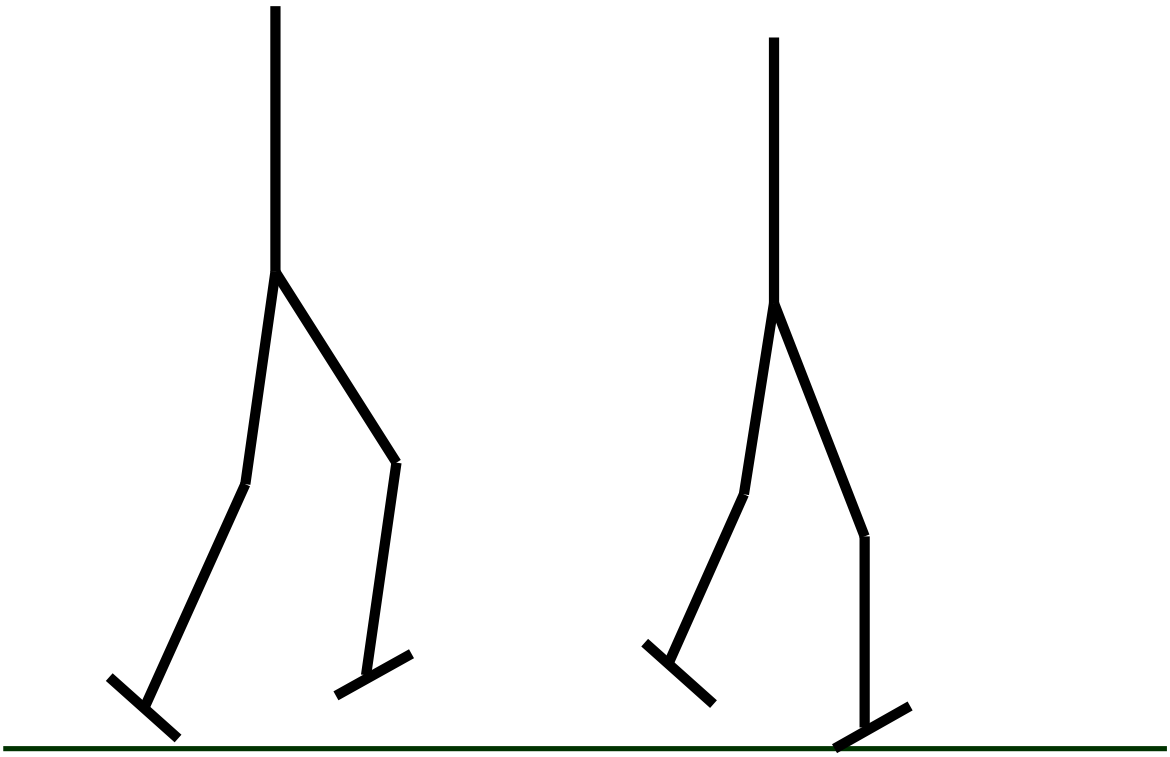

Figure 2: Toe Roll and Heel Strike are points of underactuation.

3 Motivation for Point Feet

During a typical walking gait, humans have large phases of underactuation as shown in Figure 2. During a typical step, the period during heel strike (start of a footfall) and toe roll (end of a footfall) consists of nearly point contact between the foot and the ground. During these phases the ankle and all other joints have no authority to impart moments between the robot and the ground. The zero moment point method cannot address this problem of point contact, and typical control methods will not work due to the unactuated degree of freedom. Therefore some sense of dynamic stability is required.

To explicitly address this problem, the robot studied here has point feet. This forces the controller the continuously deal with the underactuation. Once methods are developed to deal with underactuated phases, they can be applied to more complicated robots with actuated ankles.

4 Robot Model

The robot model studied for this project is the three-link walker shown in Figure 3. The robot has two legs of length r and a torso of length L. Each leg has a lumped mass m at the leg midpoint. There is a lumped hip mass M_H and a lumped torso mass M_T. The

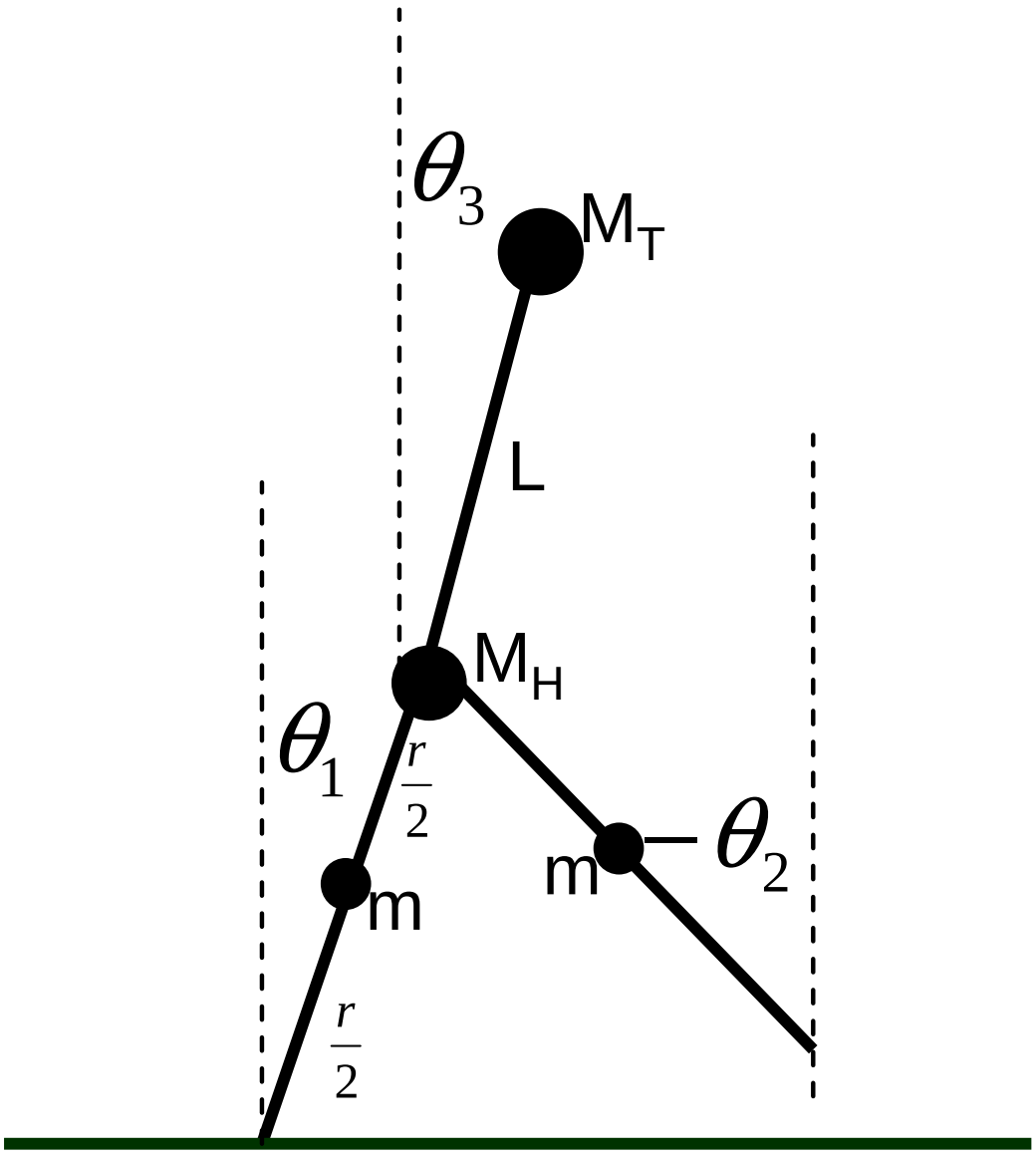

Figure 3: The 3-link walker model.

Table 1: Parameters used to model the 3-link walker

Parameter	Units	Value
Torso Length, L	m	0.5
Leg Length, r	m	1.0
Torso Mass, M_T	kg	10
Hip Mass, M_H	kg	15
Leg Mass, m	kg	5
Gravity Accel, g	m/s^2	9.81

angles used to define the robot geometry are absolute angles measured with respect to the inertial frame. θ_1 is the angle of the stance leg, θ_2 is the angle of the swing leg, and θ_3 is the angle of the torso. The parameters used are shown in Table 1.

5 Robot Model

Through the course of the system analysis and design, there is significant symbolic math, even for this simple robot. While the calculations could be done by hand, most terms are evaluated using the symbolic toolbox in Matlab.

The robot is described by a set of generalized coordinates $q = (q_1; ...; q_{\bar{N}})$. During the stance phase, the number of coordinates $\bar{N}$ is the number of links N. During flight the robot is not constrained by the ground and there are two additional degrees of freedom representing the (x, y) coordinates of the robot, so $\bar{N} = N + 2$. The robot dynamics are computed using the method of Lagrange. The Lagrangian is the real-valued function of total kinetic energy K minus total potential energy V,

$$L(q, \dot{q}) := K(q, \dot{q}) - V(q). \tag{1}$$

Lagrane's equation is

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \Gamma \tag{2}$$

where Γ is a vector of generalized forces and torques. If the kinetic energy is quadratic it can be represented as

$$K(q, \dot{q}) = \frac{1}{2}\dot{q}' D(q)\dot{q} \tag{3}$$

and (2) takes the form

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \Gamma \tag{4}$$

where $G(q) = \frac{\partial V}{\partial q}$, and $C(q, \dot{q}) = \left(\frac{\partial}{\partial q}(D(q)\dot{q})\right)\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}(D(q)\dot{q})\right)'\dot{q}$. From the principle of virtual work, for a force acting at a point $p_i = (p_i^h, p_i^v)$,

$$\Gamma_i = \left(\frac{\partial p_i}{\partial q}\right)' F. \tag{5}$$

Similarly, for a torque τ acting at a joint,

$$\Gamma_i = \left(\frac{\partial \theta_i^{abs}}{\partial q}\right)' \tau. \tag{6}$$

To begin, the kinetic and potential energies are calculated by hand based on the system geometry. The only other quantity that must be calculated manually is the vector of generalized forces Γ, which is very simple for this problem. From this point onward, all other quantities are derived symbolically. For simplicity, the robot dynamic equations are listed here. All other quantities are derived from these basic dynamics, and will not be written out as they can be quite complicated.

First, the equations of motion are derived for the stance phase, when one leg is always in contact with the ground. These dynamics are denoted with the subscript "s". The foot of the stance leg is motionless, so there are three degrees of freedom. The kinetic energy matrix denoted $D_s(q)$ is

$$D_s(q) = \begin{bmatrix} (\frac{5}{4}m + M_H + M_T)r^2 & -\frac{1}{2}mr^2\cos(\theta_1 - \theta_2) & M_T rl\cos(\theta_1 - \theta_3) \\ -\frac{1}{2}mr^2\cos(\theta_1 - \theta_2) & \frac{1}{4}mr^2 & 0 \\ M_T rl\cos(\theta_1 - \theta_3) & 0 & M_T l^2 \end{bmatrix}. \tag{7}$$

The potential energy $V_s(q)$ is

$$V_s(q) = g\left((\frac{r}{2}m + M_H r)\cos\theta_1 + M_T(r\cos\theta_1 + \cos\theta_3) + m(r\cos\theta_1 - \frac{r}{2}\cos\theta_2)\right). \tag{8}$$

The matrix $C_s(q, \dot{q})$ is derived from D_s as previously mentioned to yield

$$C_s(q, \dot{(q)}) = \begin{bmatrix} 0 & -\frac{1}{2}mr^2\sin(\theta_1 - \theta_2)\dot{q_2} & M_T rl\sin(\theta_1 - \theta_3)\dot{q_3} \\ \frac{1}{2}mr^2\sin(\theta_1 - \theta_2)\dot{q_1} & 0 & 0 \\ -M_T rl\sin(\theta_1 - \theta_3)\dot{q_1} & 0 & 0 \end{bmatrix}. \tag{9}$$

The matrix $G_s(q)$ is derived as $\frac{\partial V}{\partial q}$,

$$G_s(q) = \begin{bmatrix} -\frac{g}{2}(2M_H + 3m + 2M_T)r\sin\theta_1 \\ \frac{g}{2}mr\sin\theta_2 \\ -gM_T l\sin\theta_3 \end{bmatrix}. \tag{10}$$

Finally, the vector of generalized forces is

$$\Gamma(q) = B_s u = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} u \tag{11}$$

where u represents the torque applied to the two legs.

The stance phase dynamics of the robot are now completely described by

$$D_s(q)\ddot{q} + C_s(q, \dot{q})\dot{q} + G_s(q) = B_s u \tag{12}$$

For future use in deriving the impact model, the kinetic energy of the robot is also derived for the flight phase when there is no contact with the ground. There are now two additional degrees of freedom representing the horizontal and vertical positions of the robot, making five total degrees of freedom. The two additional states are appended to the end of the state vector q. The top left corner of the flight phase kinetic energy matrix $D_f(q)$ is simply the stance phase matrix $D_s(q)$. The matrix is symmetric, so for space considerations those entries below the diagonal are omitted. This, $D_f(q)$ is

$$\begin{bmatrix} D_{s11} & D_{s12} & D_{s13} & \left(\frac{3}{2}mr + M_H r + M_T r\right)\cos\theta_1 & -\left(\frac{3}{2}mr + M_H r + M_T r\right)\sin\theta_1 \\ \ddots & D_{s22} & D_{s23} & -\frac{1}{2}mr\cos\theta_2 & -\frac{1}{2}mr\cos\theta_2 \\ \ddots & \ddots & D_{s33} & M_T L\cos\theta_3 & -M_T L\sin\theta_3 \\ \ddots & \ddots & \ddots & 2m + M_H + M_T & 0 \\ \ddots & \ddots & \ddots & \ddots & 2m + M_H + M_T \end{bmatrix}. \tag{13}$$

6 Impact Model

Now that the continuous dynamics are solved for the stance phase, an impact model must be developed to describe what happens at foot impact. The development is sketched here, the reader is invited to consult [2] for technical proofs. The fundamental assumption used here is that the forces applied to the swing foot when it hits the ground are impulsive. With a relatively hard walking surface and rigid robot, these forces are very fast in comparison to the rest of the dynamics, so this assumption is reasonable. The main result of this assumption is that these instantaneous forces can produce a step change in the velocities of the robot, but not the configuration. Therefore, $q^- = q^+$.

The first step is to augment the stance coordinates q_s with two additional states to describe the horizontal and vertical position of the robot. The dynamics under these new coordinates $q_e = (q_s; x; y)$ describe the full free-body "flight" dynamics of the robot without constraints on the stance foot. In the flight coordinates, the dynamics are:

$$D_e(q_e)\ddot{q_e} + C_e(q_e, \dot{q_e})\dot{q_e} + G_e(q_e) = B_e(q_e)u + \delta F_{ext} \tag{14}$$

By "integrating" (14) over the instantaneous impact [2], the impact event must satisfy

$$D_e(q_e^+)\dot{q_e}^+ - D_e(q_e^-)\dot{q_e}^- = F_{ext}. \tag{15}$$

To relate forces in the inertial frame $F_2 = (F_2^T, F_2^N)$ at the swing leg end position $p_2(q_e)$ to the $F_e xt$ in the model, use the principle of virtual work. Then

$$F_{ext} = E_2(q_e^-)'F2 \tag{16}$$

where $E_2(q_e) = \frac{\partial}{\partial q_e} p_2(q_e)$. The condition of no slip and no rebound at foot contact can be written as

$$E_2(q_e^-)q_e^+ = 0. \tag{17}$$

These equations are simultaneously solved as

$$\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^-)' \\ E_2(q_e^-) & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \dot{q_e}^+ \\ F_2 \end{bmatrix} = \begin{bmatrix} D_e(q_e^-)\dot{q_e}^- \\ 0_{2\times 1} \end{bmatrix} \tag{18}$$

Solving these equations yields the velocities after the impact, $\dot{q_e}^+ = \Delta_{\dot{q}}(q_s^-, \dot{q_s}^-)$. Note that this method is basically conservation of momentum about the swing leg impact point. The other change that takes place with each step is a relabeling of the coordinates to exchange the stance and swing legs, described by the coordinate change (rotation) matrix R. These changes are lumped together, and the net result is the impact, or reset, map $\delta(x-)$ that describes the state transition from the end of one step to the start of the next step

$$\begin{bmatrix} q_s^+ \\ \dot{q_s}^+ \end{bmatrix} = \Delta(q_s^-, \dot{q_s}^-) = \begin{bmatrix} Rq_e^- \\ \Delta_{\dot{q}}(q_s^-, \dot{q_s}^-) \end{bmatrix}. \tag{19}$$

6.1 Reset Surface

Normally, the switching surface in the state space corresponds to the swing leg foot impacting the ground. However, in a robot without knees, the swing leg will always scuff the ground when it crosses the other leg.

For this case, ground impact is assumed to be initiated by the controller. The swing leg is assumed to touch the ground only when allowed by the controller. One way to do this is to have the swing leg pivot slightly outward; this puts it outside of the walking plane and it can rotate forward without ground contact. The leg is brought back inline to initiate contact. Another method is to have a small portion of the leg that can "pick up" to avoid ground contact. Both methods have been shown to work in hardware versions of robots without knees.

7 Specifying a walking motion

The control designer must somehow specify the type of motion the robot will walk. Rather than specify time trajectories to follow, this is done by assigning a relationship between the various body configuration variables called a "virtual constraint". The controller then works to enforce these virtual contstraints between various joint angles. When these constraints are correctly enforced, the controlled joint states are defined, and the dynamics that remain are the "zero dynamics".

To execute this method, first choose a generalized coordinate that is monotonically increasing during a step. For this robot, the angle of the stance leg θ_1 is convenient. The other joint angles are then assigned a configuration based on θ_1. Let the desired torso angle be described by $\theta_3^d = h_{d,1}(\theta_1)$ and the desired swing leg angle by $\theta_2^d = h_{d,2}(\theta_1)$.

Now, define the output functions $y = h(q)$ such that zeroing the output means the constraints are enforced

$$y(q) := \begin{bmatrix} \theta_3 - h_{d,1}(\theta_1) \\ \theta_2 - h_{d,2}(\theta_1) \end{bmatrix}. \tag{20}$$

When the output y is zero, dynamics have only 2 states, θ_1 and $\dot{\theta}_1$. All other states are defined as functions of these two states. These are the zero dynamics of the system. For this particular system, the θ_1 dynamics are enforcing conservation of momentum about the stance foot.

8 Feedback Control

In order to enforce the virtual constraints, a feedback controller is used to drive the output y to zero. Feedback linearization and coordinate transformations are used to simplify the control problem. Several of the available stability theorems require the trajectories to be exactly on the zero dynamics manifold. Therefore a controller is required that zeros the output in finite time (within one step). In practice, a sufficiently fast exponential controller also works, so this type of controller is used as well.

8.1 Finite-Time Controller

Assign the functions

$$\Psi(y, \dot{y}) := \begin{bmatrix} \frac{1}{\epsilon^2}\psi_\alpha(y_1, \epsilon\dot{y}_1) \\ \frac{1}{\epsilon^2}\psi_\alpha(y_2, \epsilon\dot{y}_2) \end{bmatrix}. \tag{21}$$

where

$$\psi_\alpha(x_1, x_2) = -\operatorname{sign}(x_2)|x_2|^\alpha - \operatorname{sign}(\phi_\alpha(x_1, x_2))|\phi_\alpha(x_1, x_2)|^{\frac{\alpha}{2-\alpha}} \tag{22}$$

and

$$\phi_\alpha(x_1, x_2) := x_1 + \frac{\alpha}{2-\alpha}\operatorname{sign}(x_2)|x_2|^{2-\alpha}. \tag{23}$$

The parameters α and ϵ are tuning parameters. Now, the control input is

$$u(x) := (L_g L_f h(x))^{-1}(\Psi(h(x), L_f h(x)) - L_f^2 h(x)) \tag{24}$$

where $L_b c(x) := \frac{\partial c}{\partial x} b(x)$ is the Lie derivative. With this controller, the output y will converge to zero in finite time.

8.2 Exponential Controller

An easier controller implementation is an exponential controller for the input-output linearized system. For this method, the control input is

$$u(x) := -(L_g L_f h(x))^{-1}(L_f^2 h(x)) + K_1 L_f^h(x)) + K_2 h(x)) \tag{25}$$

where K_1 and K_2 are tuning matrices.

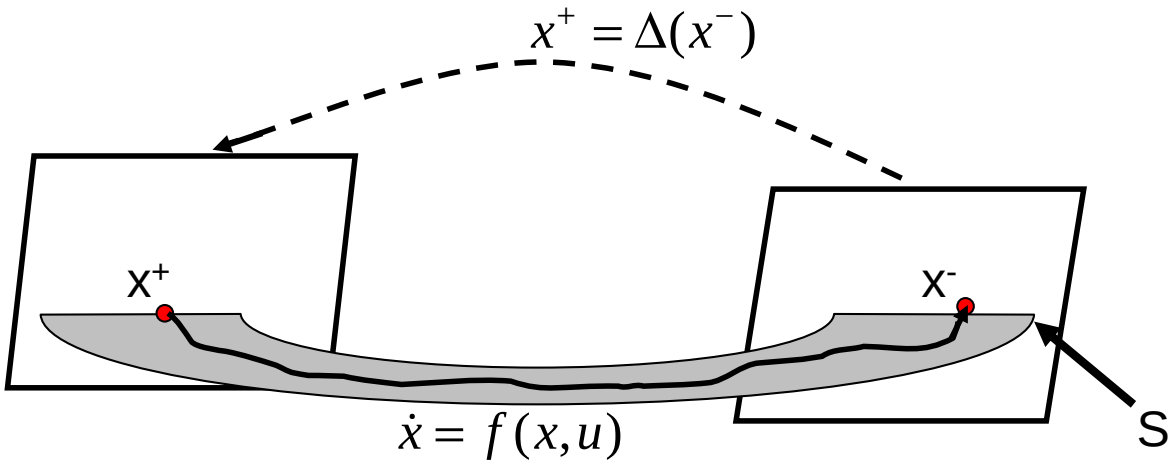

Figure 4: The Zero Dynamics Manifold of a System with Impulse Effects.

9 Simulations

To simulate the robot dynamics, two distinct steps are required. Starting at some initial condition, the continuous dynamics evolve according to (12) until the trajectory intersects the switching surface S. At this point, the reset/impact map Δ is applied as in (19) to find a new initial condition for the continuous dynamics. When the output y is zero, the continuous dynamics evolve along the zero dynamics manifold. Note that the reset map does not necessarily map the continuous state back onto the zero dynamics manifold, but the feedback controller will drive the state back to the zero dynamics manifold after each reset. For the gaits studied with this robot, the reset map does not map the state back onto the zero dynamics manifold. This process is illustrated in Figure 4.

A very simple walking gait with a stable orbit generated by assigning the virtual constraints as $\theta_3 = \frac{\pi}{6}$ and $\theta_2 = -\theta_1$. This means the torso has constant angle and the swing leg mirrors the stance leg. The swing leg impact is chosen to occur when $\theta_2 = \frac{\pi}{8}$. This orbit generates a fixed point of the switching surface (which is also the Poincaré map) of

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{8} \\ -\frac{\pi}{8} \\ \frac{\pi}{6} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 1.6 \\ -1.6 \\ 0 \end{bmatrix} (end\ of\ step) \tag{26}$$

and, after applying the reset map

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{8} \\ \frac{\pi}{8} \\ \frac{\pi}{6} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0.94 \\ -0.28 \\ 2.04 \end{bmatrix} (start\ of\ step). \tag{27}$$

Notice that the virtual constraint $\theta_3 = \frac{\pi}{6}$ implies that $\dot{\theta}_3$ is zero if the constraint is enforced, as it is at the end of the step. Immediately after impact, $\dot{\theta}_3$ is non-zero and

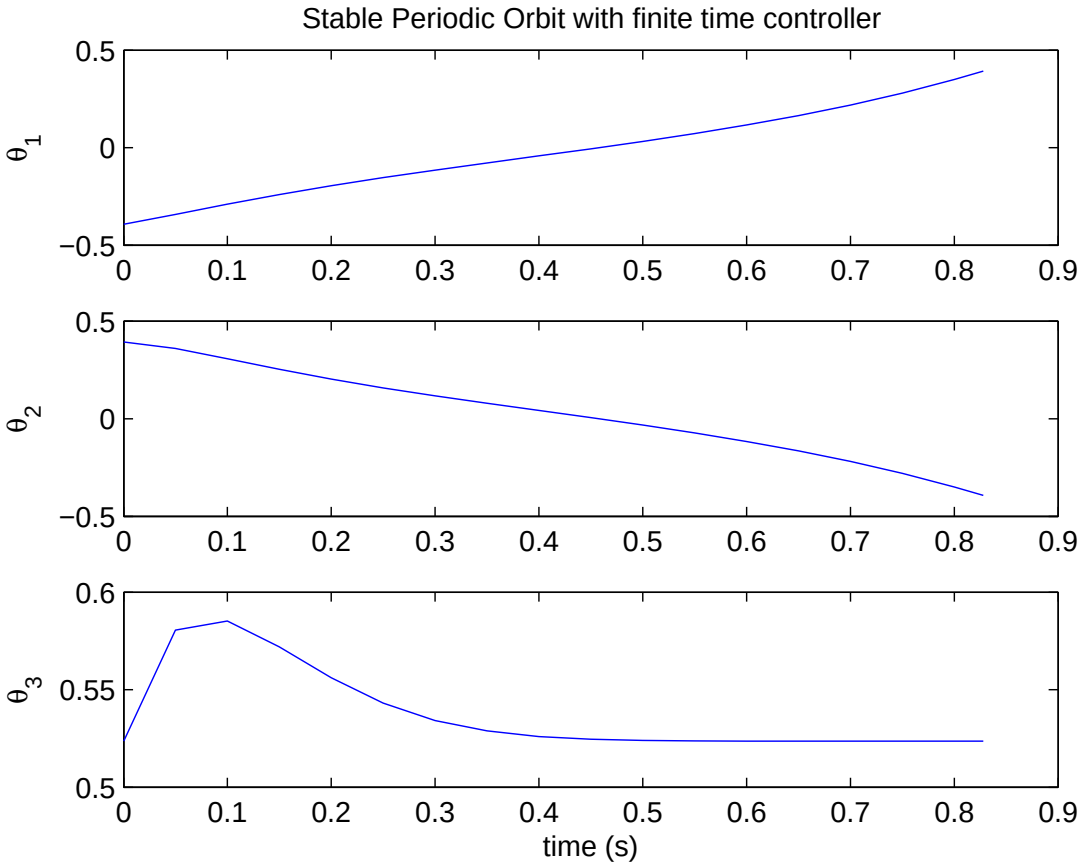

Figure 5: Joint angles for the stable periodic orbit with finite-time controller.

thus not on the zero dynamics manifold. The controller drives the trajectory back to the zero dynamics manifold before the end of the step. The time history of the joint angles is shown in Figure 9. The convergence to a stable orbit from an initial condition is shown in Figure 14. The stable orbit is shown by the thick lines where many orbits have converged. The discontinuous jump is caused by the reset map. Note that the impact maps $\dot{\theta}_3$ (z-axis) to a nonzero value, but it is driven back to zero by the controller. The x-y plane, when $\dot{\theta}_3 = 0$, can be viewed as the zero dynamics manifold in this figure.

To study how the virtual constraints are implemented, the output function $y = h(x)$ is plotted in Figure 7. Each peak is caused by the reset map with each step. Note that it takes a few steps to converge to the stable orbit, and that the output is driven to zero with each step because of the finite-time converging controller.

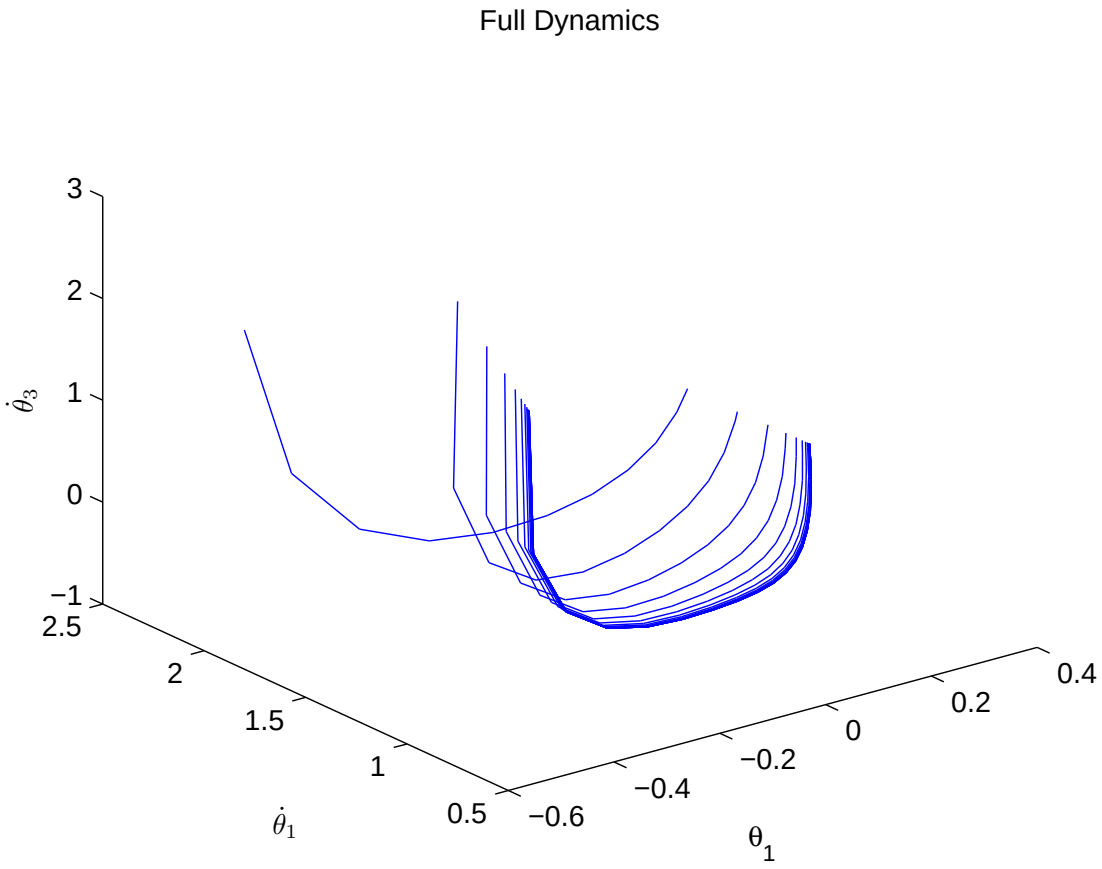

Figure 6: Stable Walking Gait with convergence from an initial condition for finite-time controller. The discontinuities are caused by the reset map.

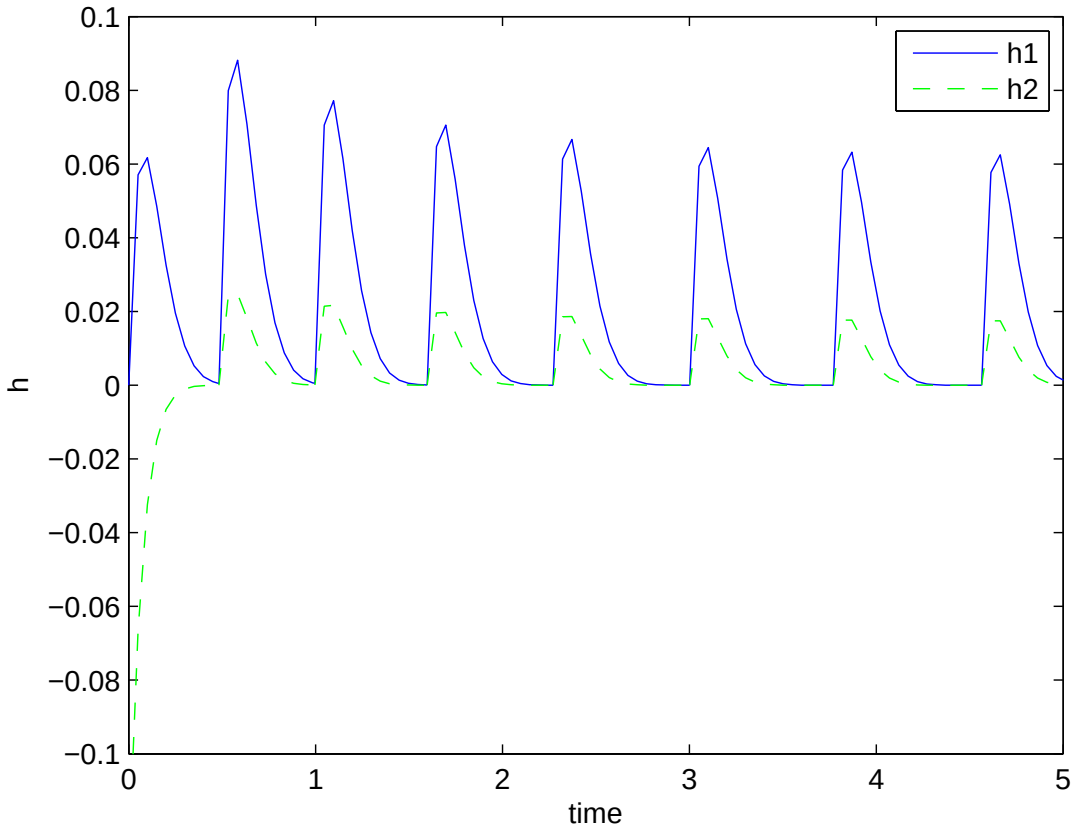

Figure 7: Output Function $y = h(x)$.

10 Stability Proofs for Finite-Time Converging Controller

The walking dynamics of the robot can be modeled as an autonomous system with impulse effects Σ

$$\Sigma : \begin{cases} \dot{x} = f(x(t)) & x^-(t) \notin S \\ x^+(t) = \Delta(x^-(t)) & x^-(t) \in S \end{cases} \tag{28}$$

.

The following hypotheses are assumed about system Σ.

HSH1) X is a smooth embedded submanifold of R^n.

HSH2) $f : X \to TX$ is continuous and a solution of $\dot{x} = f(x)$ from a given initial condition is unique and depends continuously on the initial condition.

HSH3) S is nonempty and there exist an open set $\check{X} \subset X$ and a differentiable function $H : \check{X} \to R$ such that $S := \{x \in \check{X} | H(x) = 0\}$; moreover, for every $s \in S, \frac{\partial H}{\partial x}(s) \neq 0$.

HSH4) $\Delta : S \to X$ is continuous, where S is given the subset topology from X.

HSH5) $\overline{\Delta(S)} \cap S = \emptyset$ where $\overline{\Delta(S)}$ is the set closure of $\Delta(S)$.

In addition, the following hypotheses are applied to an invariant submanifold of the system Σ

HInv1) Z is an embedded submanifold of X.

HInv2) $S \cap Z$ is an embedded submanifold with dimension one less than the dimension of Z.

HInv3) Z is locally continuously finite-time attractive.

Define the restricted Poincaré map

$$\rho : \hat{S} \cap Z \to S \cap Z \text{ by } \rho(x) := P(x) \tag{29}$$

where $\hat{S}$ is the set of points in S for which the solution returns to the zero dynamics manifold before the next impact. Then, the stability of the system can be analyzed on the zero-dynamics manifold.

Theorem: Low-Dimensional Stability Test-I Assume that the autonomous system with impulse effects (28) satisfies the hypotheses HSH1-HSH5. Suppose furthermore that $Z \subset X$ satisfies Hinv1-Hinv3. Then

1. A periodic orbit is transversal to $\hat{S}$ if, and only if, it is transversal to $\hat{S} \cap Z$.
2. $x^* \in \hat{S} \cap Z$ gives rise to a periodic orbit of (28) if, and only if, $\rho(x^*) = x^*$.

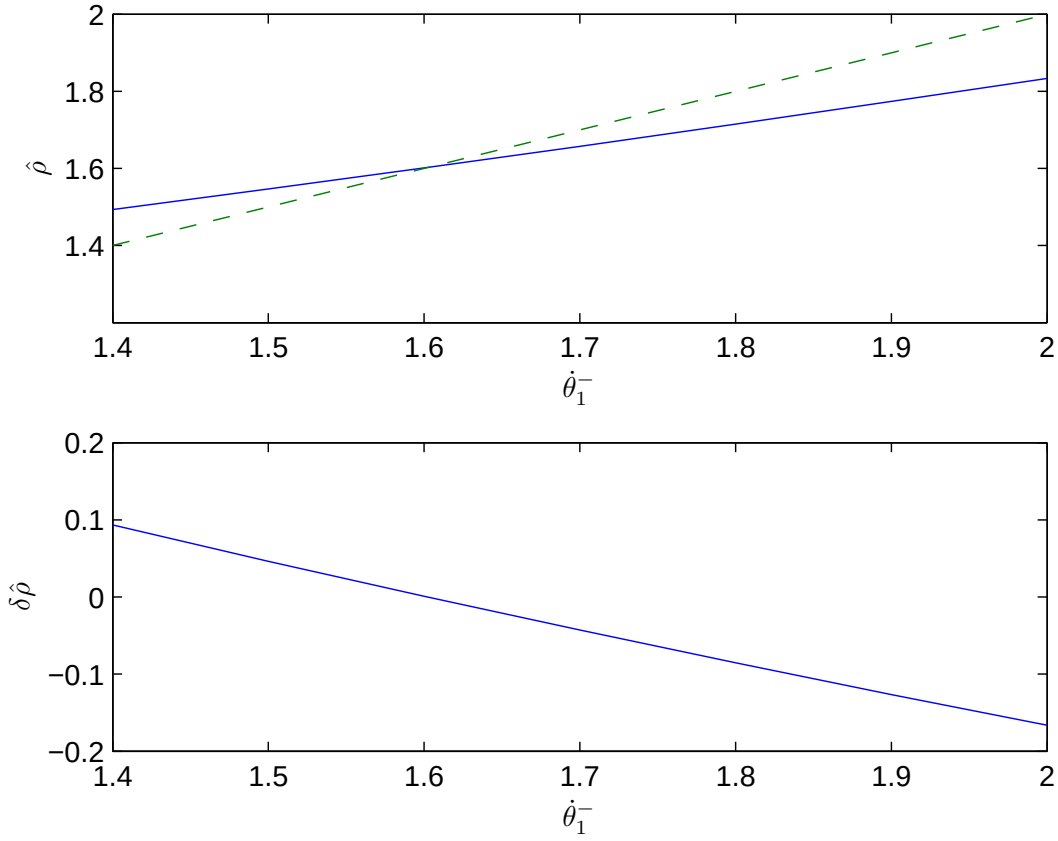

Figure 8: The Return Map of the Zero Dynamics. The dashed line in the top figure is the identity function

3. $x^* \in \hat{S} \cap Z$ gives rise to a stable (resp., asymptotically stable) periodic orbit of (28) if, and only if, x^* is a stable (resp., asymptotically stable) equilibrium point of ρ.

This theorem is used to prove the stability of the orbits described in Section 11 by checking the stability of the Poincaré return map. The main benefit of this theorem is that the stability check is carried out on the zero dynamics manifold rather than the full system dynamics. For this robot, this benefit means checking a one-dimensional system rather than the five-dimensional stability map of the full system. The zero dynamics of this model are the θ_1 and $\dot{\theta}_1$ variables, and the switching surface is defined by a constant value of stance leg angle θ_1. The value of $\dot{\theta}_1$ just before the impact is varied about the fixed point (26) to study the stability of the orbit. The top plot in Figure shows the value of $\dot{\theta}_1$ after the orbit travels through the impact map and the next step and returns to S. This is the Poincaré map (29). The dashed line is the identity function. The bottom plot shows the difference between the initial $\dot{\theta}_1$ and the return value after one step $\rho(\dot{\theta}_1)$. From either plot it is clear that the fixed point is stable, thus the periodic orbit is stable. The fixed point is also clearly identified at $\dot{\theta}_1 = 1.6$, as in (26).

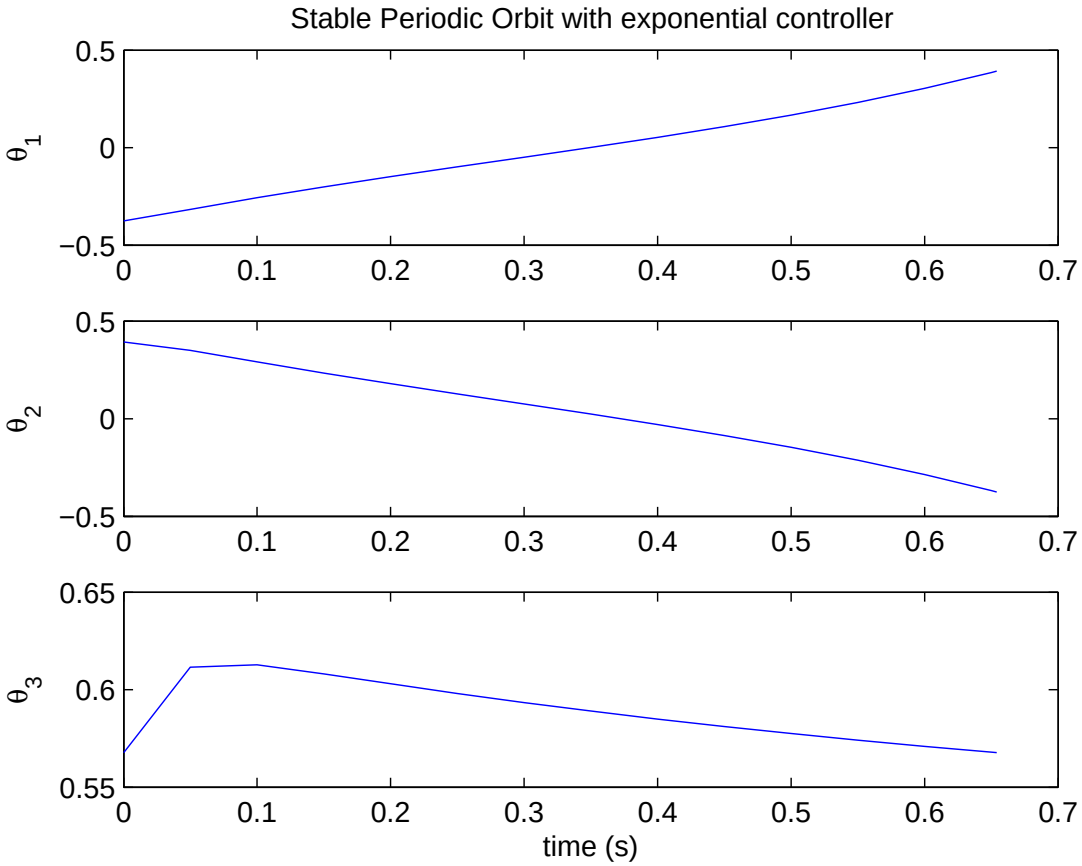

Figure 9: Joint angles for the stable periodic orbit with exponential controller.

11 Simulations with Exponential Controller

Next, an exponential controller is used to zero the output function h as described in Section 8.2. Using the exponential controller rather than the finite-time version means that the dynamics are no longer guaranteed to exactly converge to the zero dynamics manifold with each step. As long as the exponential controller is "fast enough", the resulting gait is very similar to that obtained with the finite-time controller. The same figures are shown here as for the finite time controller. The output h in plotted in Figure 11 for both finite-time and exponential controllers. Note that the output h does not reach zero with each step with the exponential controller, as it does with the finite time controller. Videos for this "simple gait" are available at [1].

12 Stability Proofs for Exponential Controller

The hypotheses HS2 and HSH4 are strengthened to make the autonomous system with impulse effects (28) continuously differentiable.

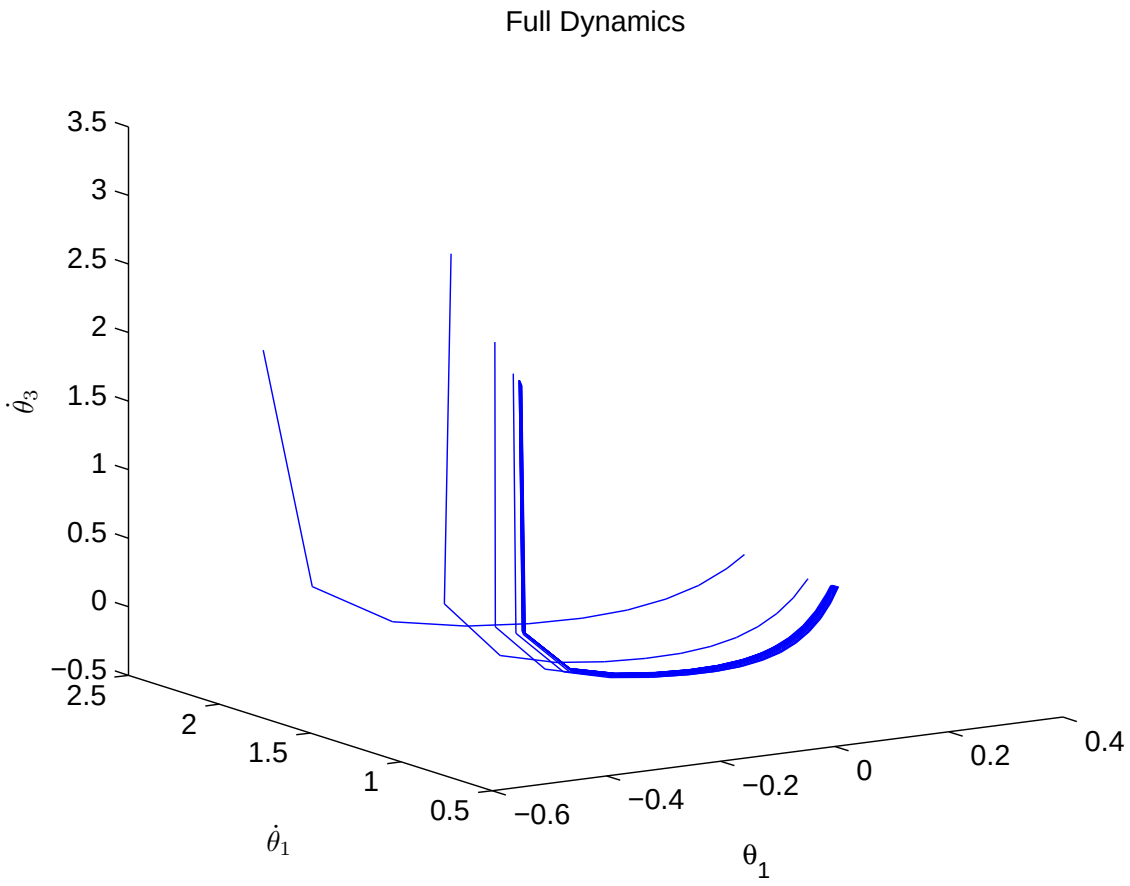

Figure 10: Stable Walking Gait with convergence from an initial condition for exponential controller. The discontinuities are caused by the reset map.

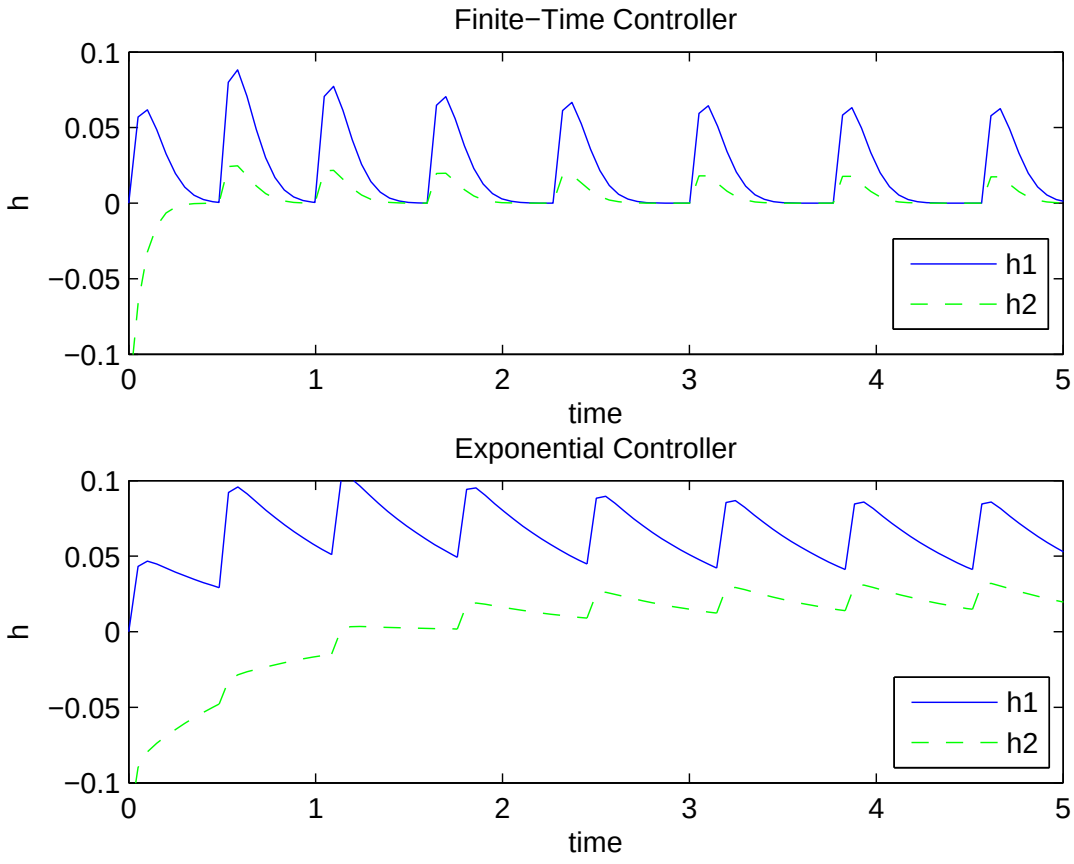

Figure 11: Output Function $y = h(x)$ for the finite-time and exponential controllers.

HSH2') $f : X \to TX$ is continuously differentiable.

HSH4') $\Delta : S \to X$ is continuously differentiable.

Without explicit mathematical definitions, let T_I be the time until impact from any state x. Let $\tilde{S}$ be the points in S that map to an x that eventually intersects the switching surface again. The following Corollary can then be used to prove stability of the system with an exponential controller.

Corollary (Method of Poincaré Sections for Differentiable Systems with Impulse Effects) Consider Hypotheses HSH1-HSH5 and assume that HSH2 and HSH4 are strengthened to HSH2' and HSH4'. Then $T_I : \tilde{X} \to R$ and $P : \tilde{S} \to S$ are continuously differentiable, and, consequently,

- $x^* \in \tilde{S}$ is an exponentially stable equilibrium point of $x[k+1] = P(x[k])$ if, and only if, the eigenvalues of $D_x P(x^*)$, the jacobian linearization of P at x^*, have magnitude strictly less than one.

By perturbing the initial conditions around the fixed point, the jacobian of the Poincaré map is numerically evaluated. Note that this is the jacobian on the switching surface and thus has five dimensions rather than the full six. The jacobian matrix is 5 x 5, and its eigenvalues are

$$\begin{bmatrix} 0.85 \\ 0.52 \\ 0.38 \\ 0.00 \\ -0.00 \end{bmatrix}. \tag{30}$$

All the eigenvalues have magnitude less than one, so the fixed point for the exponential controller is stable. If the gains of the exponential controller are decreased, the eigenvalues become unstable, as does the orbit. This happens when the controller is no longer fast enough to bring the dynamics back to the zero dynamics manifold between steps.

13 Zero Dynamics

The dynamics of the system can also be studied by restricting the dynamics to the zero dynamics manifold. The virtual constraints are assumed to be enforced exactly, making θ_2 and θ_3 functions of θ_1. The system can then be studied using the dynamics of θ_1 and $\dot{\theta}_1$. The choice of control (finite time vs. exponential) is then irrelevant as the dynamics are evolving exactly along the $h(x) = 0$ and no corrective control action is required. The zero dynamics are exactly the conservation of angular momentum about the stance foot. There are some elegant coordinate transformations in [2] that make this fact obvious. The question then arises of how to apply the reset map for the zero dynamics. When an impact occurs, angular momentum is conserved about the swing foot. Due to the "impulse" assumption, the robot configuration does not change, but the velocities do. Conservation of angular momentum L_{ang} takes the form of

$$L_{ang}(q, \dot{q})^+ = L^-_{ang}(q, \dot{q}). \tag{31}$$

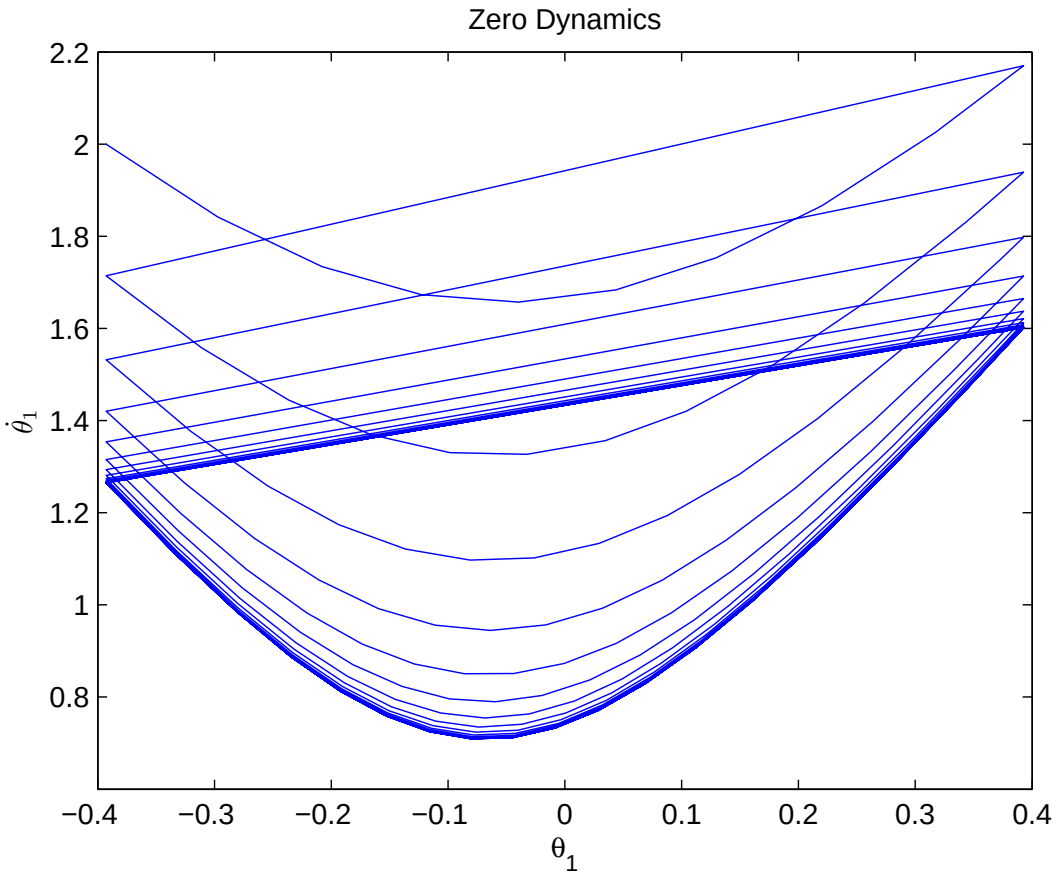

Figure 12: The zero dynamics.

This can be reduced to a reset map that determines the velocity on the zero dynamics manifold after impact

$$\dot{\theta_1}^+ = \delta_{zero}(\theta_1, \dot{\theta_1}^-). \tag{32}$$

For this robot and the gaits considered here, the function δ decreases the velocity $\dot{\theta}_1$ by about 20% with each impact.

The zero dynamics are simulated by making $\theta_2, \dot{\theta}_2, \theta_3$ and $\dot{\theta}_3$ explicit functions of θ_1 and $\dot{\theta}_1$. There are then only two continuous states. The reset map 32 is applied to yield the system with impulse effects. The results of the low-dimensional system are shown in Figure 12. Note that the zero dynamics simulations correctly identify the fixed point at $\dot{\theta}_1 = 1.6$. A stability test was conducted, and the results are the same as shown in Figure 10. This is not surprising, as the previous result was based on the dynamics exactly converging to this manifold. The 3-D figures like Figure collapse to 2-D because the output function h is perfectly satisfied.

By studying the stable orbits of the zero dynamics, one can find orbits of the full system by searching in a lower-dimensional space. Though not repeated here, theorems are available in [2] to prove that stable orbits of the zero dynamics are stable orbits of the

Table 2: Parameters for an optimized gait

i	a_0^i	a_1^i	a_2^i	a_3^i
1	0.512	0.073	0.035	-0.819
2	-2.27	3.26	3.11	-1.89

full system under certain conditions. Results are available both for finite-time controllers and for "fast enough" exponential controllers.

14 An Additional Gait

A more complicated gait was also simulated that is more energy efficient. This gait is defined by the virtual constraints $\theta_3^d = h_{d,1}(\theta_1)$ and $\theta_2^d = h_{d,2}(\theta_1)$ in (20) by

$$\begin{aligned} h_{d,1}(\theta_1, a) &:= a_1^0 + ... + a_1^3(\theta_1)^3 \\ h_{d,2}(\theta_1, a) &:= -\theta_1 + (a_2^0 + ... + a_2^3(\theta_1)^3)(\theta_1 + \theta_1^d)(\theta_1 - \theta_1^d) \end{aligned} \tag{33}$$

where θ_1^d is the swing leg ground contact angle and the a_i are the parameters in Table 2.

The output function h in (20) is now defined by these more complicated functions. The impact/reset map does not change. The full-dimensional system is simulated with a finite-time controller, and the results are shown in Figures 13-15. Note that θ_3^d is now a function rather than a constant, so the zero dynamics manifold in Figure 14 is no longer simply $\dot{\theta}_3 = 0$. Videos are available at [1], you may need to save the file to a local disk first.

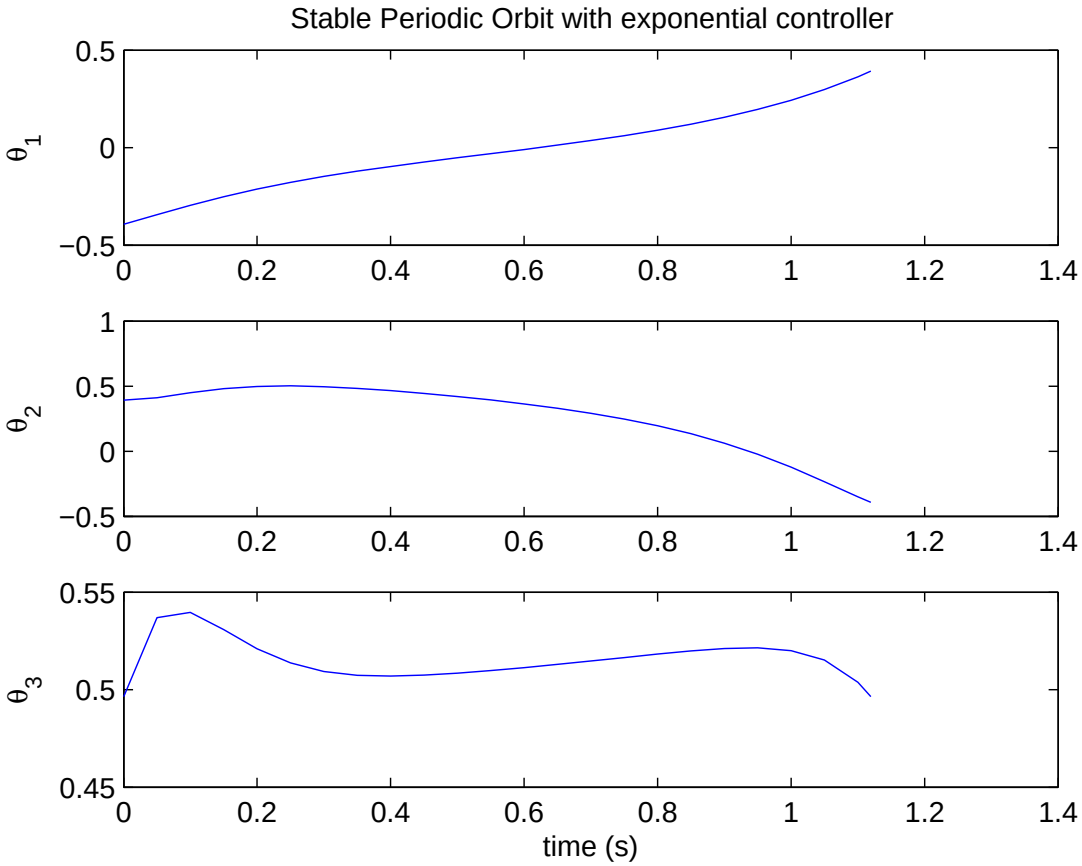

Figure 13: Joint angles for the energy-efficient gait with finite-time controller.

.

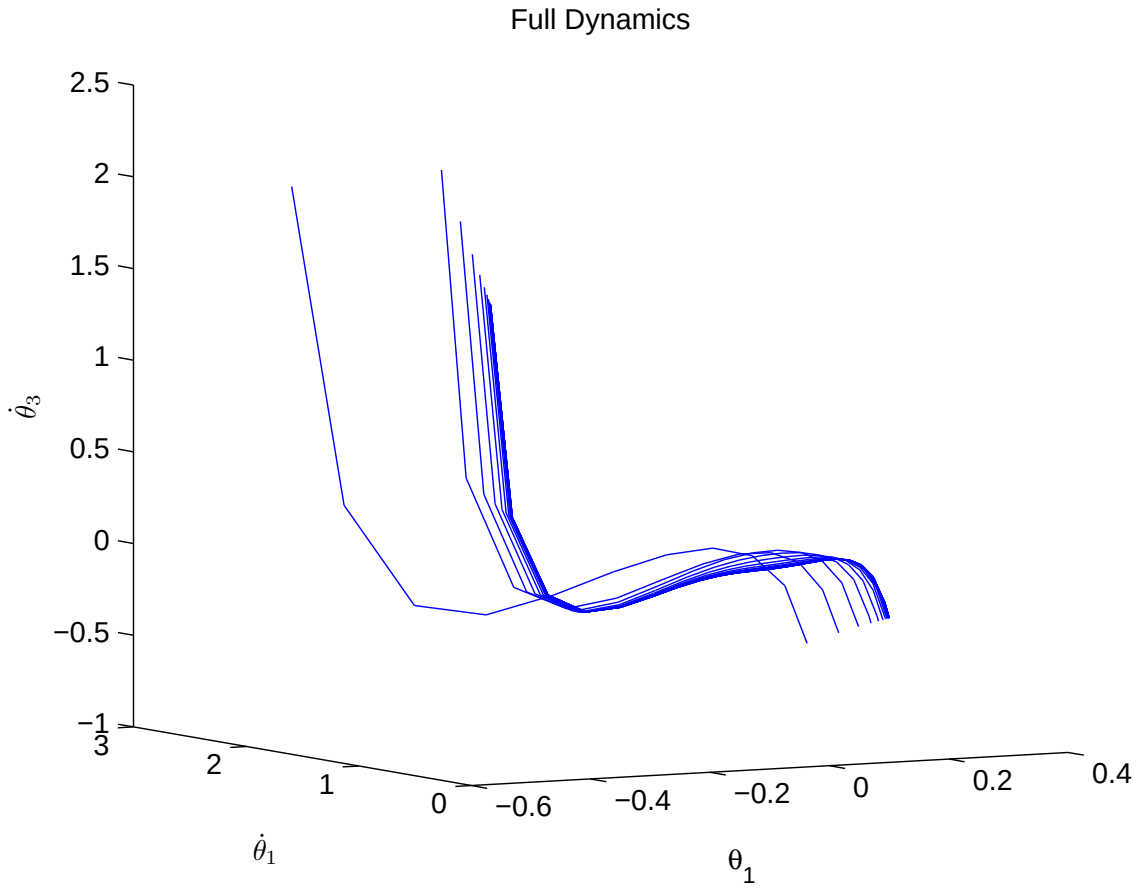

Figure 14: Stable Walking Gait with convergence from an initial condition for energy-efficient gait with finite-time controller. The discontinuities are caused by the reset map.

.

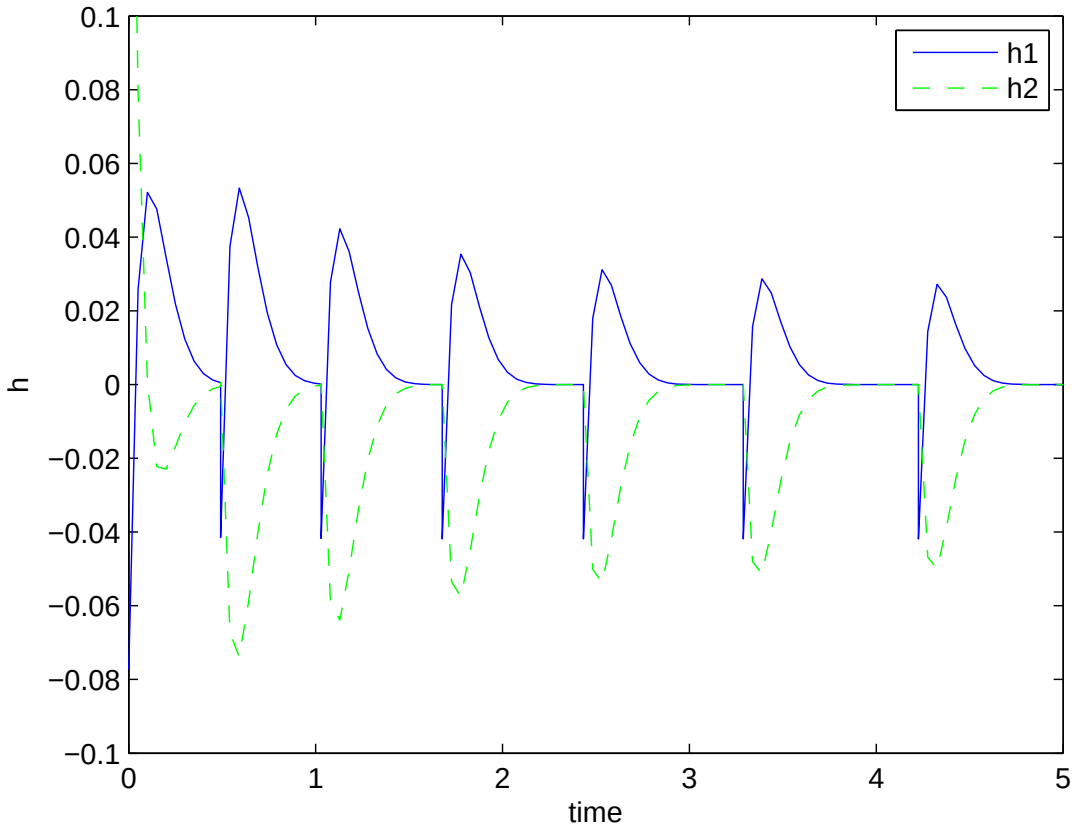

Figure 15: Output Function $y = h(x)$.

15 Conclusions

Biped walking is a challenging problem due to complicated dynamics and large state space for all but the simplest models. A fundamental challenge is the natural underactuation that occurs with walking. A useful solution to this problem is to define walking gaits by virtual constraints rather than trajectory tracking. This yields robust, provably stable orbits of the system that correspond to walking. When these virtual constraints are satisfied, the dynamics of the system can be analyzed on the zero dynamics manifold, allowing the designer to study a lower-dimensional system.

In this project, a 3-link biped walker was studied. The continuous dynamics and impact map were derived. Finite-time and exponential controllers were used to control joint angles, and the stability of the periodic orbit was proven for both cases. Two walking gaits were studied, a simple version and one that is more energy efficient.

References

[1] http://www-personal.umich.edu/ dopila/662/.

[2] E.R. Westervelt, J.W. Grizzle, C. Chevallereau, J.H Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007.