Proceedings of the ASME 2021 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2021
August 17-20, 2021, Virtual Conference

IDETC2021-70733

OPTIMAL CONTROL OF A 5-LINK BIPED USING QUADRATIC POLYNOMIAL MODEL OF TWO-POINT BOUNDARY VALUE PROBLEM

Ernesto Hernandez-Hinojosa
Mechanical and Industrial Engg.
University of Illinois at Chicago
842 W. Taylor St., Chicago, IL 60607.
Email: eherna95@uic.edu

Aykut Satici
Mechanical and Biomedical Engg.,
Boise State University,
1910 University Drive, Boise, ID 83706.
Email: aykutsatici@boisestate.edu

Pranav A. Bhounsule
Mechanical and Industrial Engg.
University of Illinois at Chicago
842 W. Taylor St., Chicago, IL 60607.
Email: pranav@uic.edu

ABSTRACT

To walk over constrained environments, bipedal robots must meet concise control objectives of speed and foot placement. The decisions made at the current step need to factor in their effects over a time horizon. Such step-to-step control is formulated as a two-point boundary value problem (2-BVP). As the dimensionality of the biped increases, it becomes increasingly difficult to solve this 2-BVP in real-time. The common method to use a simple linearized model for real-time planning followed by mapping on the high dimensional model cannot capture the nonlinearities and leads to potentially poor performance for fast walking speeds. In this paper, we present a framework for real-time control based on using partial feedback linearization (PFL) for model reduction, followed by a data-driven approach to find a quadratic polynomial model for the 2-BVP. This simple step-to-step model along with constraints is then used to formulate and solve a quadratically constrained quadratic program to generate real-time control commands. We demonstrate the efficacy of the approach in simulation on a 5-link biped following a reference velocity profile and on a terrain with ditches. A video is here: `https://youtu.be/-UL-wkv4XF8`

1 Introduction

Bipedal walking systems because of their human-like morphology are perhaps more suitable for integration in human environments such as homes and warehouses. However, bipeds are yet to achieve the dexterity and nimbleness seen in human movements. Controlling bipedal systems is a formidable challenge because of their unstable inverted pendulum-like nature that is instantaneously uncontrollable because of lack of adequate actuation at the base of the feet (also known as under-actuation). For example, when a standing robot is pushed, it cannot stabilize itself in the vertical position, but needs to take a step forward to balance. These issues are further compounded by their high dimensionality (~ 20 degrees of freedom for a 3D biped).

Inspired from humans, the most successful control paradigm is to achieve balance and control over the time scale of a step, also known as step-to-step control [2]. A fundamental challenge with step-to-step control is that it is predictive: it needs to make instantaneous control decisions that affect the dynamics over the time scale of a step or the step-to-step dynamics. Thus, step-to-step control entails solving a suitable two-point boundary value problem (2-BVP). As the dimensionality of the biped increases, it becomes computationally challenging to solve this 2-BVP in real-time. Past control approaches have either used offline design with complete models [3] or used online design with simple models, which are mapped to the complete model all in real-time [4]. The former approach is not generalizable to novel scenarios while the latter is conservative because the simple models do not capture the complete dynamics. In this paper, we present an approach that enables fast computation of controllers using a complete model to enable real-time control. Our key idea is to use PFL to transform the complete model to low dimension, offline approximation of the 2-BVP using a quadratic polynomial model, and finally, online optimal control using this model.

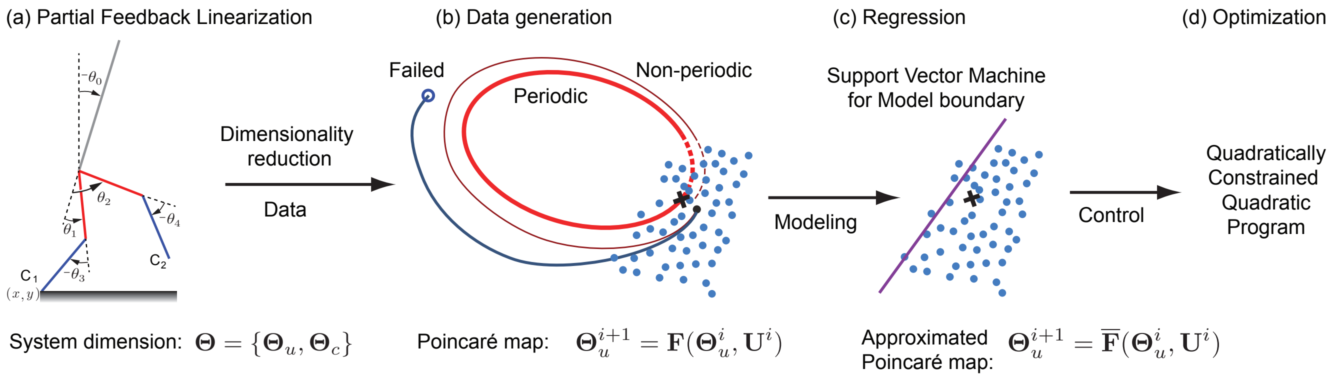
(a) Partial Feedback Linearization
(b) Data generation
(c) Regression
(d) Optimization
$-\theta_0$
θ_2
θ_1
$-\theta_4$
$-\theta_3$
C$_1$
(x, y)
C$_2$
Dimensionality reduction
Data
Failed
Periodic
Non-periodic
Support Vector Machine for Model boundary
Modeling
Control
Quadratically Constrained Quadratic Program
System dimension: $\mathbf{\Theta} = \{\mathbf{\Theta}_u, \mathbf{\Theta}_c\}$
Poincaré map: $\mathbf{\Theta}_u^{i+1} = \mathbf{F}(\mathbf{\Theta}_u^i, \mathbf{U}^i)$
Approximated Poincaré map: $\mathbf{\Theta}_u^{i+1} = \overline{\mathbf{F}}(\mathbf{\Theta}_u^i, \mathbf{U}^i)$

FIGURE 1. **Overview of the approach:** (a) PFL reduces the stance phase dynamics from $\Theta = [\Theta_u, \Theta_c]$ (10 dimensions) to $\Theta = \Theta_u$ (2 dimensions). (b) A Poincaré section is chosen at mid-stance. We generate random input state at the Poincaré section and controls at the step and simulate till the next Poincaré section to generate data for the Poincaré map given by $\mathbf{F}$, $\Theta_u^{i+1} = \mathbf{F}(\Theta_u^i, \mathbf{U}^i)$. (c) The Poincaré map is curve fitted $\Theta_u^{i+1} = \overline{\mathbf{F}}(\Theta_u^i, \mathbf{U}^i)$ where $\overline{F}$ is a quadratic polynomial model and support vector machine is used to identify the boundary of the model. (d) Nonlinear programming is used to solve a suitably formulated quadratically constrained quadratic program. For a video see [1].

2 Background and Related Work

One of the earliest demonstration of step-to-step control was through the concept of passive dynamic walking (PDW) [5]. In PDW, a walking frame resembling the human lower body settles into a cyclic or periodic gait when launched on a shallow slope with no external control. This idea has given rise to powered walkers that exhibit cyclic walking on level ground using hip and ankle actuation. Some example robots include the Cornell biped [6], Cornell Ranger [7], Delft bipedal robots [8], Michigan State synthetic wheel biped [9], and Twente Dribbel [10]. One feature of these robots is that when executing cyclic walking, they are unstable instantaneously, but are stable over the time scale of a step [11]. This cyclic stability is known as orbital stability and is a primary means of balance in humans [12].

We analyze cyclic gaits and their orbital stability using concepts from non-linear dynamics [13]. First, we find the nominal rhythmic control that achieves cyclic or periodic gait [14]. Then we use the linearization of the cyclic gait to evaluate the orbital stability. To increase the orbital stability of the system, we create a linear model of the step-to-step dynamics and then we choose a suitable linear controller using either pole placement or discrete linear quadratic regulator [15, 16]. This approach is computationally simple as it is based on a linear controller, but it limits the region of stability to a narrow region around the cyclic gait.

One way to enable stability against large perturbation is to compute controllers (e.g., using nonlinear optimization) based on the non-linear model. This approach works well for offline controller development, as the control development requires heavy computation and takes quite some time to converge to the optimum. An alternate approach is to use a simple model (e.g., linear inverted pendulum) to plan the motion over a low-dimensional space, typically the center-of-mass motion and foothold location, and then use inverse kinematics and inverse dynamics to map to the non-linear model [4, 17]. One caveat of this approach is that the results depend on the accuracy of the simple models. These simple models ignore the nonlinearities and/or the angular momentum of the upper body and have issues when planning for high speed location and/or those involving upper body movement.

The hybrid zero dynamics (HZD) approach does controller synthesis using a high-dimensional model [18, 19]. Here one defines continuous-time outputs (also known as virtual constraints) that map the actuated degrees of freedom to the unactuated degrees of freedom. One then designs a controller to drive these outputs to zero. This method is attractive because it uses control to reduce the dimensionality of the system to the unactuated degrees of freedom and is hence scalable. However, it is not very easy to find these virtual constraints that lead to acceptable performance as the system complexity increases [20]. Although one can create exponentially stable continuous time controllers, the orbital stability is only asymptotically stable [21].

We present an overview of our approach and demonstrate it on a 5-link biped. As shown in Fig. 1 (a), we use PFL to reduce the dimension from 10D to 2D. Next, as shown in Fig. 1 (b), we use a Poincaré map for modeling the step-to-step dynamics which further reduces the system dimension to 1D, the velocity of the stance leg. We note that the step-to-step dynamics has 1 state variable, the mid-stance speed, and 2 control variables, the push-off impulse and the step angle. Next, as shown in Fig. 1 (c), we approximate the step-to-step dynamics using a quadratic polynomial model and estimate its region of validity using support vector machine (SVM). Finally, as shown in Fig. 1 (d), we

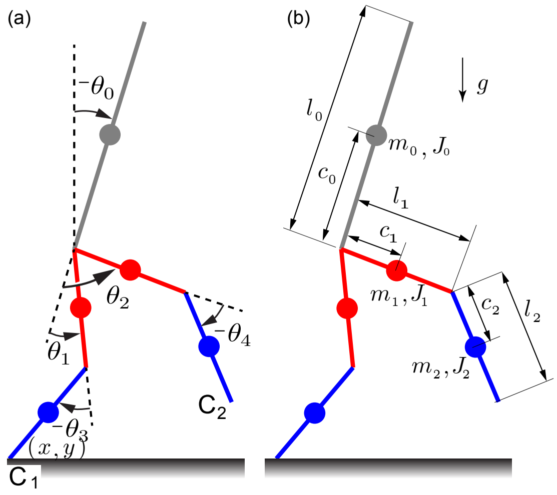

FIGURE 2. **Humanoid model:** (a) configuration variables describing the degrees of freedom, (b) mass, center of mass, inertia about center of mass, and length parameters

use nonlinear programming to solve the resulting quadratically constrained quadratic program. For a video see [1]. The novelty of our work in comparison to past approaches (e.g., [22]) is the use of data-driven methods to derive a quadratic polynomial model of the 2-BVP and identify the region of validity. Unlike past approaches that are based on linearization, our simple model covers a relatively extensive region around the cyclic gait and enables us to solve the 2-BVP in few iterations/function evaluations enabling real-time control in the future. This work builds upon our past work on control of a humanoid [23] by improving the step-to-step model approximation to compute the region of validity using SVM and subsequently we demonstrate that the resulting quadratic program can be solved in few iterations.

3 Robot model

Figure 2 shows the 2D, 5-link model used in this study. This model was previously used in our past work [23] from which this study builds from. We define the stance leg as the one in contact with the ground and the swing leg is the other leg. The foot in contact with the ground has coordinates (x, y) where the x-axis is horizontal and y-axis is vertical. The torso angle θ_0 is the angle between the torso and the vertical direction, θ_1 and θ_2 are the relative angles made by the thigh links of the stance and swing leg respectively with the torso, and θ_3 and θ_4 are the angles made by the calf links of the stance and swing leg respectively with their respective thigh links. The torso mass is $m_0 = 50$ kg, center of mass is at $c_0 = 0.5$ m, and inertia about the center of mass is $J_0 = 10$ kg-m^2. The thigh links have a mass of $m_1 = 7$ kg, center of mass is at $c_1 = 0.25$ m, and inertia about the center of mass is $J_1 = 5$ kg-m^2. The calf links have a mass of $m_2 = 5$ kg, center of mass at $c_2 = 0.25$ m, and inertia about the center of mass is $J_2 = 2$ kg-m^2. Gravity points downwards and is $g = 9.81$ m/s^2. The torso length $\ell_0 = 1$ m the thigh link and calf link lengths are equal, $\ell_1 = \ell_2 = 0.5$.

There are two sets of equations: one for the single stance phase where one foot is on the ground and the second for the foot-strike where the legs exchange roles.

3.1 Single stance equations

The state variables for derivation are defined as $\mathbf{q} = \left[x \; y \; \theta_0 \; \theta_1 \; \theta_2 \; \theta_3 \; \theta_4\right]^T$. We include the floating coordinates x and y to derive the equation, but the simplified equation has only 5 variables, θ_0, θ_1, ... θ_4. The Lagrangian $\mathscr{L} = \mathscr{T} - \mathscr{V} = 0.5\sum\left(m_i v_i^T v_i + J_i \omega_i^T \omega_i\right) - \sum\left(m_i g y_i\right)$ where v_i, ω_i, y_i are the linear velocity, angular velocity, and y-position center of mass of link i respectively. We take the summation over all the 5 links. Using the Euler-Lagrange equations gives 7 equations

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{B}\mathbf{u} + \mathbf{J}_{C_1}\mathbf{P}_{C_1} \tag{1}$$

where $\mathbf{M}$, $\mathbf{N}$, $\mathbf{B}$ are the mass matrix, accelerations due to Coriolis, centrifugal acceleration and gravity, and torque selection matrices. The control torques are $\mathbf{u} = \left[\tau_1 \; \tau_2 \; \tau_3 \; \tau_4\right]^T$ where τ_i is the torque for joint with stance calf link θ_i, $\mathbf{J}_{C_1}$ is the Jacobian of the contact point C_1 and $\mathbf{P}_{C_1}$ is the ground reaction force on the stance leg.

Without loss of generality, we can assume $x = y = 0$. Also, since C_1 is at rest, $\dot{x} = \dot{y} = \ddot{x} = \ddot{y} = 0$. Using these conditions, we use the first two equations in Eqn. 1 to find the ground reaction forces $\mathbf{P}_{C_1}$ as a function of joint angles, velocities, and acceleration. We may write the remaining 5 equations as

$$\mathbf{M}_\theta(\theta)\ddot{\theta} + \mathbf{N}_\theta(\theta,\dot{\theta}) = \mathbf{B}_\theta\mathbf{u} \tag{2}$$

where $\mathbf{M}_\theta$, $\mathbf{N}_\theta$, $\mathbf{B}_\theta$ are appropriately versions of the matrices defined earlier. We use this equation for simulating single stance phase and for controller development later.

3.2 Foot-strike equations

When the swing foot C_2 touches the ground, the single stance phase ends and the robot transitions to an instantaneous foot-strike phase. We assume that the trailing leg applies an impulsive force along the stance leg, $\mathbf{I}_{C_1} = I\left[-\sin(\theta_0+\theta_1+\theta_3),\, \cos(\theta_0+\theta_1+\theta_3)\right]^T$ where I is the scalar impulse. This force comes from the ankle motor at C_1 which is passive during the stance phase, except during the foot-strike phase. Our choice of impulsive push-off is to be able to achieve energy-efficient walking compared to hip actuation (see [24]). In this phase, angular momentum is conserved about new contact point C_2. We obtain the equations for this phase by integrating Eqn 1 and taking the limit as time goes to 0

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}^-) & -\mathbf{J}_{C_2}^T \\ \mathbf{J}_{C_2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^+ \\ \mathbf{I}_{C_2} \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q}^-)\dot{\mathbf{q}}^- + \mathbf{J}_{C_1}^T \mathbf{I}_{C_1} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

where the superscript $-$ and $+$ denote the instance before and after collision respectively.

3.3 Simulating a single step

Figure 3 shows the general equation that describes a single step, the repeating unit, that starts and ends at mid-stance. We now explain the composition of a single step. We start the step at mid-stance when stance leg thigh link is vertical, $\theta_0 + \theta_1 = 0$. Next, we use the single stance Eqn. 2 to integrate the system till foot-strike. The foot strike occurs when the swing foot C_2 touches the ground, $y_{C_2} = \ell_1 \cos(\theta_0 + \theta_1) - \ell_1 \cos(\theta_0 + \theta_2) + \ell_2 \cos(\theta_0 + \theta_1 + \theta_3) - \ell_2 \cos(\theta_0 + \theta_2 + \theta_4) = 0$. Next, we apply the foot-strike condition given by Eqn. 3. Then we swap the legs, $\theta_0^+ = \theta_0^-$, $\theta_1^+ = \theta_2^-$, $\theta_2^+ = \theta_1^-$, $\theta_3^+ = \theta_4^-$, $\theta_4^+ = \theta_3^-$. Similarly, for the angular velocities we have $\dot{\theta}_0^+ = \dot{\theta}_0^-$, $\dot{\theta}_1^+ = \dot{\theta}_2^-$, $\dot{\theta}_2^+ = \dot{\theta}_1^-$, $\dot{\theta}_3^+ = \dot{\theta}_4^-$, $\dot{\theta}_4^+ = \dot{\theta}_3^-$. Finally, we integrate the equations in single stance given by Eqn. 2 till the next mid-stance given by $\theta_0 + \theta_1 = 0$.

4 Methods

4.1 Partial feedback linearization

We use PFL to control the actuated degrees of freedom in the stance phase as described next. We invert the mass matrix from Eqn. 2 to get

$$\ddot{\theta} = \mathbf{M}_\theta^{-1}(\theta)(\mathbf{B}_\theta \mathbf{u} - \mathbf{N}_\theta(\theta, \dot{\theta})) \tag{4}$$

The system has 5 degrees of freedom, but only 4 actuators. We use PFL to decouple the 4 degrees of freedom, namely the torso θ_0, the swing leg joints θ_2 and θ_4, and the stance leg knee θ_3. Thus, if $\theta_c = \begin{bmatrix}\theta_0 & \theta_2 & \theta_3 & \theta_4\end{bmatrix}$. Then, we can find a matrix $\mathbf{S}_c$ by inspection such that $\theta_c = \mathbf{S}_c \theta$ where $\theta = \begin{bmatrix}\theta_0 & \theta_1 & \theta_2 & \theta_3 & \theta_4\end{bmatrix}$. We write

$$\ddot{\theta}_c = \mathbf{S}_c \ddot{\theta} = \mathbf{S}_c \mathbf{M}_\theta^{-1}(\theta)(\mathbf{B}_\theta \mathbf{u} - \mathbf{N}_\theta(\theta, \dot{\theta})) = \mathbf{v} \tag{5}$$

where $\mathbf{v}$ is the new control input chosen as

$$\mathbf{v} = \ddot{\theta}_c^{ref} + \mathbf{K_d}(\dot{\theta}_c^{ref} - \dot{\theta}_c) + \mathbf{K_p}(\theta_c^{ref} - \theta_c) \tag{6}$$

where θ_c^{ref}, $\dot{\theta}_c^{ref}$, $\ddot{\theta}_c^{ref}$ are the user specified reference position, velocity, and acceleration. We assume a fifth order polynomial for θ_c^{ref} such that the position, velocity, and acceleration at the start and end are specified, many of which are set to 0. The gains $\mathbf{K_p}$ and $\mathbf{K_d}$ are diagonal matrices. We choose $\mathbf{K_p} = K_p \, diag\{1,1,1,1\}$ and $\mathbf{K_d} = 2\sqrt{K_p} \, diag\{1,1,1,1\}$ to ensure critical damping. The motor torques are

$$\mathbf{u} = (\mathbf{S}_c \mathbf{M}_\theta^{-1}(\theta)\mathbf{B}_\theta)^{-1}(\mathbf{v} + \mathbf{S}_c \mathbf{M}_\theta^{-1}(\theta)(\mathbf{N}_\theta(\theta, \dot{\theta})) \tag{7}$$

The uncontrolled degree of freedom is $\theta_u = \mathbf{S}_u \theta$ where $\theta_u = \theta_1$. We can write an equation for this degree of freedom after suitably including the control input from the above equation

$$\ddot{\theta}_u = \mathbf{S}_u \ddot{\theta} = \mathbf{S}_u \mathbf{M}_\theta^{-1}(\theta)(\mathbf{B}_\theta \mathbf{u} - \mathbf{N}_\theta(\theta, \dot{\theta})) \tag{8}$$

We need to integrate this equation in the single stance phase.

4.2 Step-to-step dynamics: Poincaré map

We now introduce the idea of Poincaré section and map (see [13] for more details). The Poincaré section is a $2N-1$ (where N is the total degrees of freedom of the system) dimensional surface denoting an instance in the locomotion cycle (e.g., mid-stance, foot-strike) as shown in blue dots in Fig. 1 (b). The Poincaré map is a function $\mathbf{F}$ that maps an initial state at the Poincaré section Θ^i and controls during the step $\mathbf{U}^i$ to the state at the Poincaré section at the next step Θ^{i+1}. This map $\mathbf{F}$ describes the step-to-step dynamics and is found by integrating equations from Poincaré section to the next as shown in Fig. 3. Thus, we can write

$$\Theta^{i+1} = \mathbf{F}(\Theta^i, \mathbf{U}^i) \tag{9}$$

where i is the step number; $\Theta = \begin{bmatrix}\theta & \dot{\theta}\end{bmatrix}$ is the state, where $\theta = \begin{bmatrix}\theta_0 & \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5\end{bmatrix}$, $\mathbf{U}$ are the discrete controls that are set once per step (e.g., foot placement angle, impulsive push-off), and $\mathbf{F}$ is the Poincaré map that relates the state from one mid-stance to the next one. For most systems, it is not possible to find an analytical formula for the Poincaré map. It is obtained by numerically integrating the equations of motion and/or applying the algebraic conditions for instantaneous phases (e.g., foot-strike). Note that we define the mid-stance as $\theta_0 + \theta_1 = 0$. For a 10 degree of freedom system, the Poincaré map is 9 dimensional.

We can simplify Eqn. 9 as follows. Assuming that the PFL works as intended, the step-to-step dynamics depends only on the uncontrolled degrees of freedom, $\Theta = \begin{bmatrix}\theta_u & \dot{\theta}_u\end{bmatrix}$ (Note, we show this in the results section). Thus, for the 5-link biped, we have $\theta_u = \theta_1$. But, since the Poincaré map is at the mid-stance, there

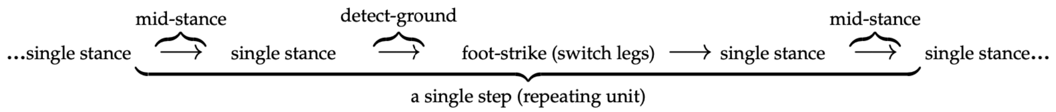

FIGURE 3. Pictorial representation of a single step

is only one degree of freedom, $\dot{\theta}_1$. We choose two controls to be the step angle at foot-strike $\theta_2 = \alpha$ and the push-off impulse I at footstrike. Thus, we write

$$ {}^m\dot{\theta}_1^{i+1} = F({}^m\dot{\theta}_1^i, \alpha^i, I^i) \qquad (10)$$

where ${}^m\dot{\theta}_1$ is the mid-stance speed of θ_1. Also, note that F is a scalar.

4.3 Approximating the Poincaré map

The Poincare map, Eqn. 10, reduces to a single equation. One caveat is that it is not possible to find an analytical solution to the step-to-step dynamics. Our goal is to find a simple approximation $\overline{F}$ with ${}^m_A\dot{\theta}_1^{i+1}$ being the approximated mid-stance speed at step $i+1$

$$ {}^m_A\dot{\theta}_1^{i+1} = \overline{F}({}^m\dot{\theta}_1^i, \alpha^i, I^i) \qquad (11)$$

4.3.1 Data generation First, we prepare the simulator to simulate a single step as shown in Fig. 3. The inputs are the mid-stance speed ${}^m\dot{\theta}_1^i$, the foot placement angle α^i, and the push-off impulse I^i and the output is the mid-stance speed at the next step ${}^m_A\dot{\theta}_1^{i+1}$. For some inputs, the reaction would cause the model to lose balance and fall before taking the next step (infeasible) while for other inputs it would successfully reach the next step (feasible). The resulting feasible/infeasible dataset is used to find the boundary of the region using support vector classification and the feasible dataset is used to find a quadratic polynomial model for the 2-BVP. These are discussed next.

4.3.2 Estimating the boundary of the Poincaré map A SVM classification model is used to estimate the boundary of the Poincaré map. The SVM binary classification algorithm from the MATLAB statistics and machine learning toolbox with a linear kernel function is used to search for an optimal hyperplane that splits the data into two classes. This allows filtering of the feasible and infeasible data. We assume that the data is linearly separable. We generate a training set of 1000 combinations of $\dot{\theta}_1$, α^i and I^i to train the SVM classifier. We show the decision boundary plane equation in Eqn. 12 where w_0 and b_0 are the coefficients of the hyperplane equation. In Eqn. 13 A^i is the Lagrange multiplier obtained from the dual cost function used in the SVM training, x^i are the support vectors, y^i is the classification of the support vectors (-1 or $+1$) and j is the number of support vectors. Eqn 15 shows the SVM classifier equations. We used a test set of 125 samples for testing.

$$ {}^m\dot{\theta}_1^i = \frac{-b_0 - w_0(1)\alpha^i - w_0(2)I^i}{w_0(3)} \qquad (12)$$

$$ w_0 = \sum_{i=1}^{j} A^i y^i x^i \qquad (13)$$

$$ b_0 = \frac{1}{j}\sum_{i=1}^{j}(y^i - w_0 \cdot x) \qquad (14)$$

$$ h(x_i) = \begin{cases} +1 \text{ (not feasible) if } w\cdot x + b \geq 0 \\ \;\;-1 \text{ (feasible)} \quad \text{if } w\cdot x + b < 0 \end{cases} \qquad (15)$$

4.3.3 Quadratic polynomial model of the Poincaré map Once we developed the classification model, we used the feasible samples to develop a polynomial regression model. Our regression model is a quadratic polynomial with regression coefficient denoted by β.

$$ {}^m_A\dot{\theta}_1^{i+1} = \beta_0 + \beta_1^m\dot{\theta}_1^i + \beta_2\alpha^i + \beta_3 I^i + \beta_4^m\dot{\theta}_1^i\alpha^i \ldots $$
$$ \ldots + \beta_5^m\dot{\theta}_1^i I^i + \beta_6\alpha^i I^i + \beta_7\alpha^i\alpha^i + \beta_8 I^i I^i \qquad (16)$$

4.4 Quadratically constrained quadratic program

We formulated an optimization problem to find the inputs that yield the approximated ${}^m_A\dot{\theta}_1^{i+1}$. The cost function is the squared error of the outputs in relation to their nominal states: Cost $= ({}^m_A\dot{\theta}_1^{i+1} -{}^m \bar{\dot{\theta}}_1^{i+1})^2 + (\alpha^i - \bar{\alpha}^i)^2 + (I^i - \bar{I}^i)^2$ where ${}^m\bar{\dot{\theta}}_1^{i+1}$, $\bar{\alpha}^i$ and $\bar{I}^i$ have the nominal values -0.97, 0.375, 0.18 respectively. These nominal values were chosen to achieve walking at human speed and step length. The constraints include Eqns. 12 - 16.

The problem can be rewritten as a quadratically constrained

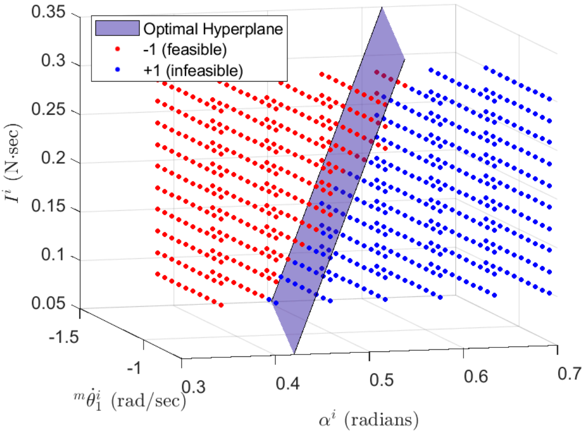

FIGURE 4. Feature space and optimal hyperplane of test set.

quadratic program as given below

$$\underset{\mathbf{y}}{\text{minimize}} \quad 0.5\mathbf{y}^T\mathbf{H}\mathbf{y} + \mathbf{f}^T\mathbf{y} + c \tag{17}$$

$$\text{subject to: } 0.5\mathbf{y}^T\mathbf{Q}_i\mathbf{y} + \mathbf{k}_i^T\mathbf{y} + d_i = 0 \tag{18}$$

$$\mathbf{p}^T\mathbf{y} < b_0 \tag{19}$$

$$\mathbf{LB} \leq \mathbf{y} \leq \mathbf{UB} \tag{20}$$

where $\mathbf{y} = \begin{bmatrix}\Theta^{i+1} & \mathbf{U}^i\end{bmatrix}^T$, $\mathbf{f}$, $\mathbf{H}$ and c are user chosen constants in the cost function. Eq. 18 is the polynomial regression equation as shown in Eq. 16 where ${}^m_A\dot{\theta}_1^i$ is treated as a known constant. The matrix $\mathbf{Q}_i$ contains the second order terms, $\mathbf{k}_i$ contains first order terms and d_i contains the constants. Eq. 19 accounts for linear inequality constraints and contains the SVM classifier constraint from Eq. 15. **LB** and **UB** are the lower bound and upper bound vectors set to be [-2.5, 0.087, 0.1] and [-0.5, 0.873, 0.4].

5 Results

5.1 SVM Classification

Fig 4 shows the feature space of the training set along with the SVM classification optimal hyperplane where the red dots are the feasible samples, and the blue are the infeasible samples. We show the optimal hyperplane in purple which corresponds to the decision boundary for the model. Fig 5 shows the confusion matrix of the SVM model. Overall, the model has an accuracy of 93.6%. As seen from the confusion matrix, the precision of predicting feasible data (-1) is 86.2% and the precision for predicting nonfeasible data (+1) is 95.8%.

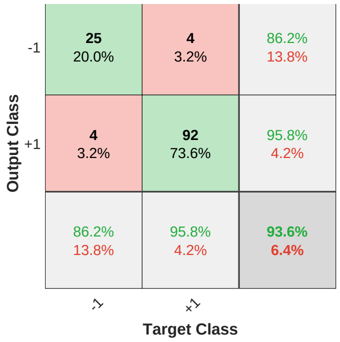

FIGURE 5. Classification confusion matrix

	Estimate	p-value
β_0	1.587	<0.01
β_1	0.910	<0.01
β_2	-10.553	<0.01
β_3	1.872	<0.01
β_4	1.429	<0.01
β_5	-2.874	<0.01
β_6	-26.937	<0.01
β_7	25.42	<0.01
β_8	6.116	<0.01

TABLE 1. Regression coefficients

5.2 Quadratic polynomial regression

We show the estimated coefficients of the regression with their respective p-values in table 1. The R-squared value and adjusted R-squared value for the regression analysis was 0.975 for both. Considering 1 rad/sec to be nominal, 93.1% of the test samples had an accuracy of 90% or better, and 100% of the samples had an accuracy of 80% or better. The mean accuracy of the regression model was 95.61%.

5.3 Optimization: Following reference velocity

We tested the model and optimization framework's ability to follow a reference velocity profile $\dot{\theta}_1^{ref}$. We chose a sinusoidal pattern with an amplitude of 0.1 rad/sec for 20 steps with an offset equal to the nominal velocity of -0.97 rad/sec. The goal of the optimization is to drive the velocity ${}^m\dot{\theta}_1^{i+1}$ to the targeted velocity $\dot{\theta}_1^{ref}$. We modified the cost function such that ${}^m\bar{\dot{\theta}}_1^{i+1} = \dot{\theta}_1^{ref}$. Fig. 6 shows the reference velocity and the actual velocity in simulation. Fig. 7 shows the behavior of the controls α^i and I^i as the velocity changes. We generate these controls from the optimization problem using the quadratic polynomial model and are inputs to the actual model, Eqn. 10. The mean absolute error was 0.0106 rad/sec. We calculated this by subtracting the reference velocity from the actual at every step and taking the mean. These optimizations took about 15 to 40 function evaluations demonstrating the feasibility for real-time control.

5.4 Optimization: Terrain with ditches

We tested the model and optimization framework's ability to plan motion on a terrain with ditches. The swing foot touchdown relative to the stance foot is $2(\ell_1+\ell_2)\sin 0.5\alpha^i \sim (\ell_1+\ell_2)\alpha^i$. Thus, swing foot touchdown is a linear constraint that preserves the quadratic form. One issue with this formulation is that we need to solve multiple quadratic programs for the foot step planning one for each feasible foot step location encompassed with ditches on either side. Here the choice of the ditches was such that we only needed to solve two quadratic programs, one for stepping before the ditch and one after the ditch.

After solving both problems, we chose the solution with the lowest cost. Fig. 9 shows the velocities ${}^m\dot{\theta}_1^{i+1}$ and ${}^m_A\dot{\theta}_1^{i+1}$ for every step. ${}^m_A\dot{\theta}_1^{i+1}$ obtained from the optimization solution and the ${}^m\dot{\theta}_1^{i+1}$ is obtained from the simulator, Eqn 10. In the optimization problem the first ${}^m\dot{\theta}_1^{i}$ input value is the nominal and for every other step the input value is the velocity ${}^m\dot{\theta}_1^{i+1}$ obtained from the Poincaré map in the previous step. The mean absolute error was 0.0077 rad/sec and was calculated by subtracting ${}^m\dot{\theta}_1^{i+1}$ from ${}^m_A\dot{\theta}_1^{i+1}$ and taking the mean. The largest absolute error was 0.0251 rad/sec. Fig. 10 shows the controller deviations from their nominal values. The humanoid from the simulation shown in Fig. 8 successfully walked over four ditches without falling. It is clear that the humanoid varies the length of its steps to properly step over the ditches. To walk over the third ditch, two steps were taken. Conversely, to walk over the fourth ditch one long step was enough. The control α was bounded by the boundaries from the training set of the regression model (0.34-0.873 radians). This limited the humanoids ability to take shorter or longer step when necessary. In these simulations, the nonlinear optimization needed 6 to 130 functions evaluations, again quite low demonstrating the feasibility for real-time control. A simulation video is shown in the reference [1].

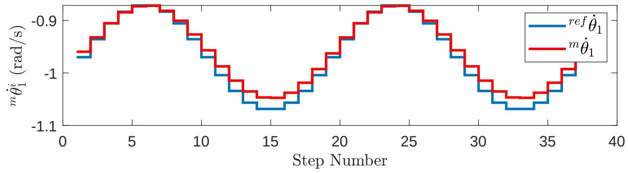

FIGURE 6. Sinusoidal velocity tracking performance

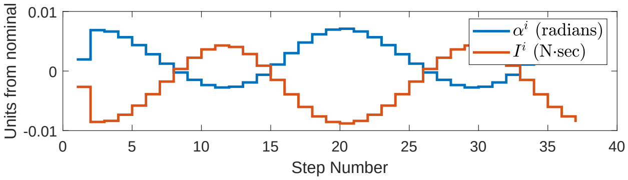

FIGURE 7. Sinusoidal velocity tracking control inputs

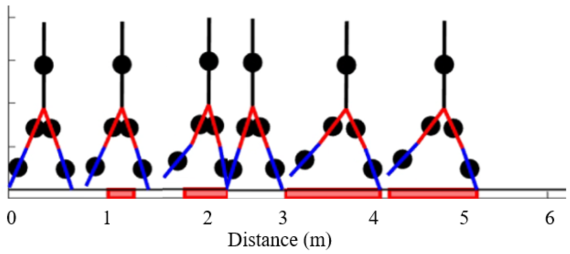

FIGURE 8. Simulation of humanoid walking over four ditches.

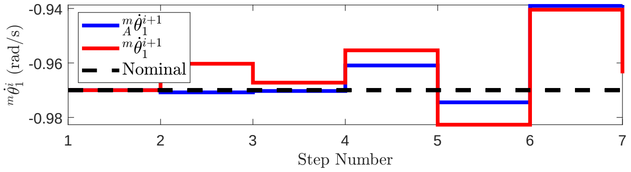

FIGURE 9. Velocity before and after every step for planning on ditches

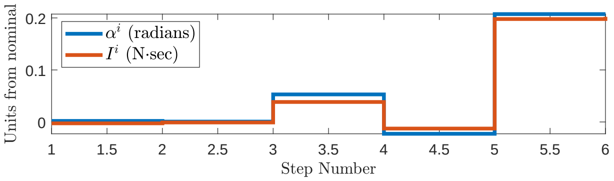

FIGURE 10. Controls at every step for planning on ditches

6 Discussion

Through numerical simulation on a 5-link biped with 10 dimensional state space, we have successfully shown that PFL reduces this to 2 dimensional state space. Furthermore, the Poincaré section reduces this to only a single dimension. Then using data-driven approach first by using extensive simulation for data generation and second by curve fitting using SVM and polynomial regression, we demonstrate a highly accurate approximation of the 2-BVP. This resulted in the formulation and solution of a suitable quadratically constrained quadratic program that achieves high accuracy and quick solution time.

PFL like the HZD formulation is a model reduction framework. However, HZD requires choosing state based output functions which can be non-intuitive and leads to a time invariant control formulation which does not need an external clock. PFL enables specifying the joint angles/velocity independently of each other, which is more intuitive and leads to a time dependent control formulation and hence needs an external clock for synchronization. Another difference between the two approaches is that

HZD is used for local stability or stability in the current step, while in our formulation the PFL with Poincaré map is used for orbital stability or stability in both the current and the future step.

Past approaches for orbital stabilization of legged robots have considered linearized control around the period walking cycle for control, which limits the approach to small disturbances. In contrast, we use data from a relatively large region near the periodic gait for fitting the Poincaré map. This enables the controller to stabilize a relatively wider range of initial conditions. These quadratic approximations of the model and linear approximation of the boundary enables us to formulate a relatively simple quadratically constrained quadratic program that is solved in less than 60 function evaluations, which is promising for real-time control.

The accuracy from the SVM classification model is good enough for our planar humanoid model, but a higher accuracy may be needed for more complex 3D models. Several ways to improve the model are discussed next. The SVM used in this method uses a hard margin and works well with linearly separable data. Using a soft margin SVM might improve the model by considering nonlinearity. One method of increasing the feasible-predicting precision is by increasing the SVM margin such that the probability of predicting +1 when it is -1 is low. In our approach, the quadratic polynomial regression was used to prevent over-fitting the data. However, a higher order polynomial regression may provide a better fit, but with possibly increased computational time. In addition, using a neural network may yield a similar or better fit. The use of a neural network was shown in [23] to be a more stable control model than a quadratic control model. Future work will investigate the use of neural networks to approximate the Poincaré map. Finally, we limited our planning horizon to a single step look-ahead. However, planning over multiple steps is more desirable for more complex terrain, and this may lead to a combinatorial explosion that would need suitable heuristics to solve quickly.

7 Conclusions and Future Work

This paper presented a fast method of computing controllers using a model that enables real-time control of a 5-link planar bipedal robot. PFL was used to transform the complete model to low dimension. Data was generated for Poincaré map. SVM was utilized to obtain a decision hyperplane which separates feasible and infeasible datapoins. A quadratic polynomial regression was used to approximate the Poincaré map leading to a closed form expression for the 2-BVP. Finally, a quadratically constrained quadratic program was formulated and solved. The results demonstrated that the approach was able to follow a reference velocity profile and plan over a terrain with ditches with high accuracy in relatively short amount of time ($<$ 130 function evaluations). To conclude, the approach presented in this paper provides a computationally efficient method of achieving precise control for complex bipedal robots. This work may ultimately be used to not only control bipedal robots but also lower limb exoskeletons and prosthesis. The evaluation of this approach on a humanoid platform will be performed in future studies. Such platform will be a 3D extension of this work where the humanoid will exhibit 3 uncontrolled degrees of freedom in a 6 dimensional state space with a 6-1 dimension of the Poincaré section.

ACKNOWLEDGMENT

The work is funded by US National Science Foundation through grant 1946282.

REFERENCES

[1] Hernandez-Hinojosa, E., 2021. Humanoid stepping. `https://youtu.be/-UL-wkv4XF8`, May 2021.

[2] Dingwell, J. B., and Kang, H. G., 2007. "Differences between local and orbital dynamic stability during human walking". *Journal of biomechanical engineering,* ***129***(4), pp. 586–593.

[3] Xie, Z., Berseth, G., Clary, P., Hurst, J., and van de Panne, M., 2018. "Feedback control for cassie with deep reinforcement learning". In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 1241–1246.

[4] Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., and Tedrake, R., 2016. "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". *Autonomous robots,* ***40***(3), pp. 429–455.

[5] McGeer, T., 1990. "Passive dynamic walking". *The International Journal of Robotics Research,* ***9***(2), pp. 62–82.

[6] Collins, S., and Ruina, A., 2005. "A bipedal walking robot with efficient and human-like gait". In Proceeding of 2005 International Conference on Robotics and Automation, Barcelona, Spain.

[7] Bhounsule, P. A., Cortell, J., Grewal, A., Hendriksen, B., Karssen, J. D., Paul, C., and Ruina, A., 2014. "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge". *International Journal of Robotics Research*.

[8] Wisse, M., and Van der Linde, R. Q., 2007. *Delft pneumatic bipeds*, Vol. 34. Springer Science & Business Media.

[9] Flynn, L., Jafari, R., and Mukherjee, R., 2010. "Active synthetic-wheel biped with torso". *IEEE Transactions on Robotics,* ***26***(5), pp. 816–826.

[10] Dertien, E., 2006. "Dynamic walking with dribbel". *Robotics & Automation Magazine, IEEE,* ***13***(3), pp. 118–122.

[11] Hobbelen, D., and Wisse, M., 2007. "Limit cycle walking". *Humanoid Robots Human-like Machines*, pp. 277–294.

[12] Dingwell, J. B., Kang, H. G., and Marin, L. C., 2007. "The effects of sensory loss and walking speed on the orbital dynamic stability of human walking". *Journal of biomechanics,* ***40***(8), pp. 1723–1730.

[13] Strogatz, S., 1994. *Nonlinear dynamics and chaos*. Addison-Wesley Reading.

[14] Garcia, M., Chatterjee, A., Ruina, A., and Coleman, M., 1998. "The simplest walking model: Stability, complexity, and scaling". *ASME J. of Biomech. Eng.,* ***120***, pp. 281–288.

[15] Kuo, A. D., 1999. "Stabilization of lateral motion in passive dynamic walking". *The International journal of robotics research,* ***18***(9), pp. 917–930.

[16] Bhounsule, P. A., Ruina, A., and Stiesberg, G., 2015. "Discrete-decision continuous-actuation control: balance of an inverted pendulum and pumping a pendulum swing". *Journal of Dynamic Systems, Measurement, and Control,* ***137***(5), p. 051012.

[17] Feng, S., Whitman, E., Xinjilefu, X., and Atkeson, C. G., 2015. "Optimization-based full body control for the darpa robotics challenge". *Journal of Field Robotics,* ***32***(2), pp. 293–312.

[18] Grizzle, J., Abba, G., and Plestan, F., 2001. "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects". *IEEE Transactions on Automatic Control,* ***46***(1), pp. 51–64.

[19] Westervelt, E. R., and Grizzle, J. W., 2002. "Design of asymptotically stable walking for a 5-link planar biped walker via optimization". In Proceeding of 2005 International Conference on Robotics and Automation, Washington DC, USA.

[20] Hereid, A., Cousineau, E. A., Hubicki, C. M., and Ames, A. D., 2016. "3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics". In 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1447–1454.

[21] Chevallereau, C., Grizzle, J., and Shih, C., 2009. "Asymptotically stable walking of a five-link underactuated 3-d bipedal robot". *IEEE Transactions on Robotics,* ***25***(1), pp. 37–50.

[22] Xiong, X., and Ames, A., 2021. "3d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization". *arXiv preprint arXiv:2101.09588*.

[23] Bhounsule, P. A., Kim, M., and Alaeddini, A., 2020. "Approximation of the step-to-step dynamics enables computationally efficient and fast optimal control of legged robots,". In ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers.

[24] Kuo, A., 2002. "Energetics of actively powered locomotion using the simplest walking model". *Journal of Biomechanical Engineering,* ***124***, pp. 113–120.