Data-driven Identification of a Non-homogeneous Inverted Pendulum Model for Enhanced Humanoid Control

Ernesto Hernandez Hinojosa*, and Pranav A. Bhounsule

Abstract— **This work aims to enhance the linear inverted pendulum model (LIPM) for bipedal robot control. While the LIPM simplifies the dynamics by assuming homogeneity, it fails to capture important nonlinear dynamics observed in real-world scenarios. To address this limitation, we propose the non-homogeneous LIPM (NH-LIPM), which incorporates a non-homogeneous term in the traditional LIPM dynamics. The NH-LIPM is augmented with controllable inputs, allowing for greater parameter control compared to the LIPM. Through regression analysis and the use of the Recursive Least Squares algorithm with forgetting, we extract and adaptively tune the NH-LIPM parameters. Evaluation through high-fidelity simulation and experimentation on a 30-degree-of-freedom humanoid demonstrates that the NH-LIPM offers improved velocity tracking control, particularly when ankle torque with damping control is added. This model provides a flexible framework for simultaneously controlling the center of mass velocity and position, enabling precise reference tracking and enhanced bipedal locomotion. A video is linked here: `http://tiny.cc/NHLIPM`**

I. Introduction

Bipedal robots, such as Agility Robotics's Digit, strive to replicate human-like balance, walking control, and adaptability. However, achieving robust walking capabilities for these robots remains a significant challenge. Factors such as high dimensionality, nonlinear dynamics, and under-actuation due to limited ankle torques contribute to the complexity of bipedal robot control.

In the field of bipedal robot control, two main approaches have emerged. The first approach, the linear inverted pendulum model (LIPM), uses a point mass model, ignoring most of the nonlinearities, to develop controllers [1]. This controller is then transferred to a full robot model using force and/or position control. However, this approach overlooks the influence of angular momentum and center of mass differences, limiting its ability to capture the full dynamics of the system. The second approach involves developing a trajectory tracking controller using a full-order robot model that behaves as a low-order model, typically through the concept of hybrid zero dynamics [2]. This approach faces challenges when the system deviates significantly from the reference trajectory, leading to stabilization difficulties.

In this study, we propose an approach that leverages the strengths of both approaches. We employ tight inverse kinematics trajectory tracking on the feet and an inverse-dynamics-based torso orientation control such that the dynamics of the center of mass (COM) of the robot are predictable. We then use the LIPM as the base function of the Step-to-Step (S2S) dynamics. The S2S dynamics refer to the time-evolving behavior and state transition of the system within the time-scale of a step. The LIPM exhibits modeling inaccuracies that contribute to errors in the S2S control of the robot. To mitigate such errors, we propose adding a non-homogeneous equation to the homogeneous LIPM dynamics producing the non-homogeneous linear inverted pendulum model (NH-LIPM). Two methods are used to extract the solution to the non-homogeneous equation. The first is by extracting an analytical polynomial regression on the LIPM S2S error, and the other is utilizing the recursive least squares (RLS) algorithm to adaptively find the optimal parameters of the non-homogeneous equation. The main contributions of this research are the 1. data-derived formulation of the particular solution of the NH-LIPM to reduce the modeling error displayed in the homogeneous LIPM, 2. capturing the effects of forced inputs on the NH-LIPM to develop the F-NH-LIPM and using the new model for multi-input control and 3. utilizing the RLS algorithm to iteratively adapt the coefficients of the particular solution in real time.

II. Background and Related Work

The control of bipedal robots poses significant challenges due to their high degree of freedom and underactuation at the ankle joint. Achieving stability in bipedal locomotion often relies on orbital stability, where control is achieved at the time-scale of each step [3]. Passive dynamic walkers, for example, demonstrate orbital stability by utilizing natural dynamics to walk stably down slopes [4][5]. Actuators can be employed to achieve similar behavior on level ground by replacing gravitational energy with joint torques [6]. Stability evaluation of such cyclic systems can be performed using a S2S map or Poincaré map, which relates states at one instance of the cycle to the same state in the next cycle [7]. Stability analysis based on eigenvalue analysis of the map helps determine the stability of the system and design control strategies to achieve desired stable eigenvalues [8].

Hybrid zero dynamics is an alternative method for achieving S2S stability by utilizing virtual constraints that track controlled degrees of freedom as a function of uncontrolled degrees of freedom [9], [10]. Although this reduces the dimensionality of the Poincaré map, it does not guarantee S2S stability. Various methods, such as finding controls with eigenvalues less than 1, modifying virtual constraints,

Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, 842 W. Taylor St., Chicago, IL 60607 USA. `eherna95@uic.edu`, `pranav@uic.edu`. The work is funded by US National Science Foundation through grant 2128568, E.H.H. was supported by the American Heart Association predoctoral fellowship.
*Corresponding author

or introducing event-based stabilization, can be employed to achieve S2S stability [11]. However, linearized stability achieved through these methods may be compromised in the presence of significant perturbations.

To simplify control and develop orbitally stable controllers, several simple models have been proposed, such as the Linear Inverted Pendulum Model (LIPM) for walking and the Spring-Loaded Inverted Pendulum Model for running [12][13][14]. These models provide insights into the dynamics and appropriate foot placement for achieving desired walking velocities. However, limitations arise from modeling errors due to nonlinearities that are not captured by these simplified models [14].

Previous studies have explored enhancements to the LIPM by incorporating time-varying dynamics and forcing functions derived from ankle torque inputs. For example, an angular momentum-based LIP (ALIP) model with a forcing function was developed to simulate walking over dynamic surfaces [15]. In our previous work, polynomial regression analysis was employed to extract an analytical S2S map of the COM position and velocity, improving dynamic predictions and velocity tracking in 2D simulations [14]. This study expands upon the previous work by extending it to the 3D case. This study introduces the concept of a NH-LIPM that captures the time and state dependency of the dynamics. It incorporates forcing functions derived from ankle torque inputs, enabling further control enhancement.

Other research efforts have explored adaptive control approaches to address model mismatches and improve control robustness [16]. These include using neural networks for adaptive control and developing adaptive virtual models to respond to parameter variations and external disturbances [17]. However, these approaches focus on adaptation at the controller level and may not yield significant improvements in forward speed control or continuous dynamics in the x-axis.

In this study, we propose the use of the recursive least squares algorithm with forgetting to adaptively tune the parameters of the discrete NH-LIPM S2S model in real-time for forward and lateral walking. The forgetting factor in the controller allows us to select a model that adapts quickly to changes or at a slower rate, retaining the memory of previously sampled data.

III. Models

A. Robot model

Digit is an advanced bipedal robot with impressive capabilities. It has 30 degrees of freedom and features 20 actuated joints, as illustrated in Figure 1 (refer to [18] for more comprehensive information on Digit). The design of Digit's legs draws inspiration from its predecessor, Cassie [19], but incorporates notable enhancements. These enhancements include the addition of a roll joint at the toe and the inclusion of an upper body comprising of a torso and two arms.

To understand the leg kinematics of Digit, we can visualize it as two interconnected closed-loop chains. These chains are represented by distance constraints between the heel spring and the hip pitch joints. During the swing leg phase, for the sake of simplicity, the knee and tarsus joint positions are inversely coupled as explained in [14].

B. Step-to-step (S2S) Models

The general form of the S2S models developed in this study is illustrated in [20] and the general S2S map is shown in Eqn. (1). During the initiation of the single support phase (SSP), the COM position and velocity relative to the stance foot are denoted as x_k^- and v_k^-, respectively, where k represents the current step number. As the current SSP concludes, these states transition to x_k^+ and v_k^+. The dynamics model, represented as ζ, captures the S2S relationship, mapping the initial states $\mathbf{s_k^-} = \{x_k^-, v_k^-\}$ and input toe pitch torque, τ_p, to the resulting states $\mathbf{s_k^+} = \{x_k^+, v_k^+\}$.

Fig. 1: **Robot model:** Digit bipedal robot with 30 degrees of freedom and 20 actuated joints.

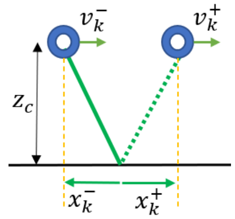


Fig. 2: The S2S model maps the state at s_k^+ with the state at s_k^- and control during the step u_k.

$$\mathbf{s_k^+} = \zeta\left(\mathbf{s_k^-}, \tau_\mathbf{p}\right) \tag{1}$$

1) Analytical map from the LIPM: The LIPM maintains the height of the COM at an approximately constant level by controlling the hip, knee, and ankle joints. The LIPM equations are linear and involve parameters such as gravity (g), the constant height of the COM (z_c), and the mass of the robot (m).

$$\ddot{x} = \frac{g}{z_c}x + \frac{1}{mz_c}\tau_p, \qquad \ddot{y} = \frac{g}{z_c}y + \frac{1}{mz_c}\tau_r \tag{2}$$

The input roll torque is denoted as τ_r. Assuming no input torques, the equations can be integrated analytically from time 0 to T_s, resulting in an equation that conforms to the form of Eqn. (1).

$$\zeta\left(\mathbf{s_k^-}\right) = \begin{bmatrix} x_k^{L+} \\ v_k^{L+} \end{bmatrix} = \begin{bmatrix} C_T \cdot x_k^- + T_c S_T \cdot v_k^- \\ S_T/T_c \cdot x_k^- + C_T \cdot v_k^- \end{bmatrix} \tag{3}$$

Here, $S_T = \sinh(T_s/T_c)$, $C_T = \cosh(T_s/T_c)$, and $T_c = \sqrt{z_c/g}$. The superscripts L and N appended to the state variables indicate whether they belong to the LIPM or the regression model, respectively. It is important to note that the equations apply to both the x (frontal) axis and y (lateral) axis.

C. System homogeneity and state-independence

The homogeneity property of the LIPM, displayed in Eqn. 2 when $\tau = 0$, enables analytical solutions and simplifications, making it a viable model for investigating bipedal walking dynamics. Additionally, the pendulum dynamics exhibit state independence and symmetry, relying

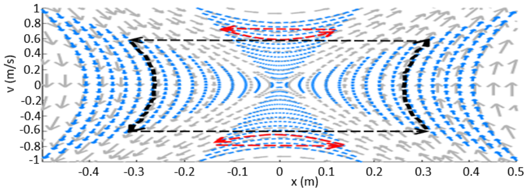


Fig. 3: Phase portrait of the LIPM with two P1 orbits shown in red and one P2 orbit shown in black.

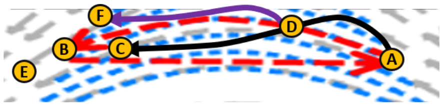


Fig. 4: Phase asymmetry displayed in NH-LIPM dynamics (black and purple trajectories)

solely on the COM height. This behavior is illustrated in the phase portrait shown in Fig. 3, where the blue lines depict possible system dynamics at a given stepping frequency and COM height. The dotted red lines represent the forward and backward walking of the LIPM, completing a cycle in a single step, while the dotted black lines demonstrate lateral walking, completing a cycle in two steps. Fig. 4 illustrates the phase trajectories for the LIPM (red) and a NH-LIPM (purple and black). The NH-LIPM displays asymmetry. An initial state A would normally lead to the final state B in the LIPM. However, in a NH-LIPM the final state might be at C. Similarly, a starting state at D would take the same path as starting state A in the LIPM and end at state E, but would take the purple path in the NH-LIPM to F.

1) Non-homogeneous LIPM: The LIPM falls short of capturing the time and state-dependent variations and non-linearities present in real-world dynamics as illustrated in Fig 4. The NH-LIPM incorporates a time-varying and state-dependent component that captures the deviations from the idealized linear inverted pendulum dynamics. To derive the time and state-dependent equation, data was collected to conduct a regression analysis and extract the non-homogeneous term that compliments the LIPM, also known as the particular solution.

$$\mathbf{s}' = \mathbf{A}\mathbf{s} + \mathbf{q}(t, \mathbf{s}) \tag{4}$$

$$\mathbf{s}(t) = e^{\mathbf{A}t}\mathbf{s}(\mathbf{0}) + \int_0^t e^{\mathbf{A}(t-\gamma)}\mathbf{q}(\gamma, \mathbf{s}(\gamma))d\gamma \tag{5}$$

$$\boldsymbol{\eta}^N(t) = \zeta(\mathbf{s}^-, t) + \mathbf{h}(\mathbf{s}^-, t) \tag{6}$$

Eqn. 4 is the general dynamics equation of the NH-LIPM where $\mathbf{q}$ is the time-variant nonhomogeneous term. Eqn. 5 is the system equation of motion which can be expressed as the sum of Eqn. 1 and the particular solution $\mathbf{h}$ as shown in Eqn. 6. The non-homogeneous model incorporates a time-varying component that captures the deviations from the idealized linear inverted pendulum dynamics. The objective is to find the analytical equation, $\mathbf{h}$, that captures the time-variant dynamics of the system and is a solution to the nonhomogeneous system in Eqn. 4.

$$\begin{aligned} h_1(s_k^-, t) &= x_k^{N+} = \alpha_0 + \alpha_1 x_k^- + \alpha_2 v_k^- + \alpha_3 t.. && (7)\\ &... + \alpha_4(x_k^-)^2 + \alpha_5(v_k^-)^2 + \alpha_6 t^2 + ... + \alpha_7 x_k^- v_k^-.. \\ &... + \alpha_m(x_k^-)^n(v_k^-)^n t^n \\ h_2(s_k^-, t) &= v_k^{N+} = \beta_0 + ... && (8) \end{aligned}$$

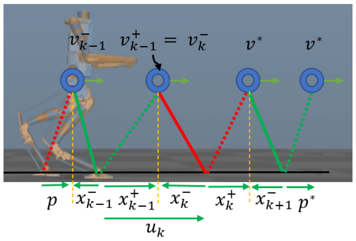


Fig. 5: Model-based stepping. The stepping controller dictates the step size u_k to achieve desired velocity $v*$.

The polynomial regression equation of degree n will fit the form in Eqns. 7 and 8 where α and β are coefficients.

D. Enhancing the NH-LIPM with Forced Inputs

The NH-LIPM's control method is limited to foot placement and COM height. To further enhance the model, a forcing term, $\mathbf{g}$, is introduced to account for input parameters, such as ankle torque/damping, which are not considered in the traditional LIPM. Regression analysis is applied to extract a function incorporating these additional parameters. This refined model called the forced NH-LIPM (F-NH-LIPM), offers an improved representation of the bipedal robot's dynamics.

$$\boldsymbol{\eta}^F(t) = \zeta(\mathbf{s}^-, t) + \mathbf{h}(\mathbf{s}^-, t) + \mathbf{g}(\mathbf{s}^-, t, \tau) \tag{9}$$

where τ are control inputs that may be desirable for better continuous inter-step control.

IV. Methods

A. Stepping Controllers

The S2S map from the analytical LIPM and data-driven NH-LIPM were used to develop stepping controllers for stable walking. We assumed there is no double support phase during walking. Inverse dynamics is used to apply a force on the base to ensure the hip position is at a desired constant height z_c.

1) Model-based Stepping: The stepping controllers inputs are step lengths u_k^L for the LIPM and u_k^N for the NH-LIPM.

$$u_k^L = x(t)_{k-1}^{+R} - x_k^{L-} = x(t)_{k-1}^{+R} - (v_{des}^+ - C_T v_k^-)T_c/S_T \tag{10}$$

$$u_k^N = x(t)_{k-1}^{+R} - x_k^{N-} = x(t)_{k-1}^{+R} - \hat{h}_2^-(v_k^-, x_k^{L-}) \tag{11}$$

where the superscript R in $x(t)_{k-1}^{+R}$ indicates the robot's continuous state, $\hat{h}_2^-$ is the inverse function of $\hat{h} = h_2(s^-, t) + x_k^{L-}$ used to solved for x_k^- given a desired $v_k^+ = v_{des}$.

Feedback may be added to the controller to add stability, as shown below.

$$u_x^R = {u_k}^N + \mathbf{K}\ \left(\mathbf{s(t)_k^R} - \mathbf{s_k^N}\right) \tag{12}$$

where u_x^R is the updated controller.

Along the lateral direction, the controller u_y^R is employed.

$$r_p = (w_s)/2 \tag{13}$$

$$r_v = (v_s) + j(r_p) \tag{14}$$

$$\begin{aligned} u_y^R &= y(t)_{k-1}^{+R} + w_s + \mathbf{K_1}\ \left(\mathbf{s(t)_k^R} - \mathbf{s_k^N}\right) ... \\ &... + \mathbf{K_2}\ \left(\mathbf{s(t)_k^R} - \mathbf{r_k}\right) && (15) \end{aligned}$$

where w_s is the desired step width of the robot, r_v is the velocity vy_k^+ when walking in a cyclic gait with y_k^- equal to r_p. The function j gives the cyclic velocity as a function of the desired state y^- which in this case is half the step width, $w_s/2$. To move laterally, the term v_s is added and, therefore, can be used to control lateral walking speed. The step width w_s should be negative during the left stance and positive during the right stance. More details on LIPM-based stepping control can be found in [14] and [21].

B. Data Collection

To gather the necessary training data, we employed a high-fidelity simulator, which emulates the behavior and characteristics of the robot. This simulator is proprietary to Agility Robotics. During these data collection trials, we followed a procedure akin to what would be done on the actual hardware. Running every trial at close to real-time speed and starting and stopping each trial individually.

The simulator was used to run forward and lateral walking simulations, which allowed us to gather S2S data in both discrete and continuous formats. Similarly, on hardware, a study was done on forward/backward (frontal) walking. By adjusting the desired walking speed during training, with control inputs x_k^- and v_k^-, we inherently introduced variability across different trials (see video [22]). This variation enabled us to examine the impact of different walking speeds on the robot's behavior. For every step taken x_k^-, v_k^-, $x(t)$, and $v(t)$ were stored parameters.

To derive the equation of the particular solution of the NH-LIPM, regression analysis was employed on the error between the LIPM dynamics and actual robot dynamics shown in Eqn. 16. The NH equation will then capture some of the dynamics not captured by the LIPM and the summation of the LIPM and the NH term will better approximate the robot dynamics as shown in Eqn. 17.

$$\mathbf{e}_k^L = \mathbf{s}(t)_k^{+R} - \mathbf{s}(t)_k^{+L} \tag{16}$$

$$\mathbf{s}(t)_k^{+R} \approx \mathbf{s}(t)_k^{+L} + \mathbf{h}(s_k^-, t) \tag{17}$$

Data was collected using the Agility Robotics simulator. For every training trial, the robot walked using LIPM-based control at various speeds between 0 and 0.7 m/s, ensuring every speed, with 0.1 increments, was trialed for at least 5 seconds. The regression analysis was done with initial states (x^-, v^-) and time (t) as parameters and the continuous state errors $(\mathbf{e}_k^L(t))$ as regression outputs.

The F-NH-LIPM equation was formulated by fitting a function to the model error with inputs to the equation as shown in Eqn. 19.

$$\mathbf{e}_k^N = \mathbf{s}(t)_k^{+R} - \mathbf{s}(t)_k^{+L} - \mathbf{h}(\mathbf{s}_k^-, t) = \mathbf{g}(\mathbf{s}_k^-, t, \tau) \tag{18}$$

$$\mathbf{s}(t)_k^{+R} \approx \mathbf{s}(t)_k^{+L} + \mathbf{h}(\mathbf{s}_k^-, t) + \mathbf{g}(\mathbf{s}_k^-, t, \tau) \tag{19}$$

where $\mathbf{e}_k^N$ is the model error of the NH-LIPM and $\mathbf{g}$ captures the dynamics associated with the force inputs τ.

The LIPM and NH-LIPM controllers can control v_k^+ or x_k^+, but not both. The F-NH-LIPM (Eqn. 19) has two control inputs, u_k^N and τ, so it can be used to control the states v_k^+ and either x_k^- or x_k^+. We use τ to control v_k^+ and leave x_k^- as a free parameter to which we may assign an arbitrary value.

Two control inputs were tested: ankle damping and ankle torque. The ankle torque was a ramp torque applied during the first quarter of the cycle. For every training trial, the robot walked using LIPM-based control at speeds between 0 and 0.7 m/s with seven different force input values. Every combination was tested for at least ten steps. The regression analysis was done with initial states (x^-, v^-), inputs τ, and time (t) as parameters and the continuous state errors $(\mathbf{e}_k^N(t))$ as regression outputs.

C. Recursive Least Squares (RLS)

The RLS algorithm was utilized in this study to enable real-time adaptation of the discrete S2S model parameters.

The nonhomogeneous term (h in Eqn. 6) of the NH-LIPM can be represented as $\hat{c}$ in the linear form.

$$\hat{\boldsymbol{c}}^T = [\boldsymbol{\theta}\,\boldsymbol{b}^T] \tag{20}$$

$$\boldsymbol{c} = [p_{k+1}^e, v_{k+1}^e] \tag{21}$$

$$\boldsymbol{b} = [p_k^e, v_k^e] \tag{22}$$

$$\boldsymbol{\theta} = \begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{bmatrix} \tag{23}$$

$\mathbf{c}$ contains the predicted center of mass position and velocity error states $[p_{k+1}^e, v_{k+1}^e]$, $\mathbf{c}$ are the current error states, and θ are the model parameters. The RLS LIPM would then take the discrete form of $\boldsymbol{\eta}_k^{rls} = \zeta_k(\mathbf{s}^-) + \hat{\boldsymbol{b}}^T$.

The algorithm consists of an initialization state where the error covariance matrix $\mathbf{P}$ is computed, often initialized as an identity matrix, and the initialization parameters $\boldsymbol{\theta}$ are set. The coefficients of the NH function are initialized to zero. For illustration purposes, the RLS algorithm below predicts the state v where a_{21} and a_{22} are the linear parameters of the non-homogeneous term, h_2, in the NH-LIPM. This method can similarly be applied to the state p to predict parameters a_{11} and a_{12} in h_1.

$$\text{Initialize}\begin{cases} \boldsymbol{P} = \dfrac{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}{\lambda} \\ \boldsymbol{\theta_2} = \begin{bmatrix} a_{21} & a_{22} \end{bmatrix}^T \end{cases} \tag{24}$$

$$\text{Iterate}\begin{cases} \boldsymbol{\phi} = [p_k, v_k] & (25) \\ \mathbf{K} = \dfrac{\boldsymbol{P\phi}^T}{\Delta + \boldsymbol{\phi P \phi}^T} & (26) \\ e = v_{k+1}^e - \boldsymbol{\theta}_2^T \boldsymbol{\phi}^T & (27) \\ \boldsymbol{\theta}_2 = \boldsymbol{\theta}_2 + Ke\sigma & (28) \\ \boldsymbol{P} = \dfrac{\boldsymbol{P} - \mathbf{K}\boldsymbol{\phi P}}{\lambda} & (29) \end{cases}$$

The algorithm is iterated at every footstep where the regressor, $\boldsymbol{\phi}$, consisting of the inputs, is updated as shown in Eqn. 25, and the Kalman gain, $\mathbf{K}$, is computed from Eqn 26. Δ is the regularization parameter used to prevent overfitting, where larger values reduce the variance but increase the bias. In Eqn. 27, the prediction error, e, is computed by subtracting the current state error, v_{k+1}^e, from the predicted state error. The parameters are then updated in Eqn. 28 where σ is the error factor used to smoothen out parameter outputs. Finally, in Eqn. 29 the error covariance matrix is updated by implementing a forgetting factor λ, a value between zero and one where a higher value means less forgetting.

The RLS algorithm operates by iteratively updating the model parameters using a weighted least squares approach. The objective is to minimize the difference between the predicted outputs of the model and the actual measurements obtained during walking as shown in Eqn. 27.

By applying the forgetting factor in Eqn. 29, older data points are gradually assigned less weight in the parameter update process. This allows the model to adapt more quickly to changes occurring in the immediate past while still retaining a degree of information from earlier steps. The forgetting factor can be adjusted to strike an appropriate balance between adaptability and stability, ensuring that the model remains robust despite variations in the robot's dynamics.

To evaluate the performance of the RLS controller, we intentionally varied several parameters that impact the dynamics of the robot. This deliberate variation was carried out to create scenarios where the LIPM no longer accurately captures the system dynamics. Specifically, we adjusted the robot's height by reducing it by 0.1 m, doubled the step width, and modulated the toe damping between 20% and 60% of the maximum damping value of 28.5 Ns/m.

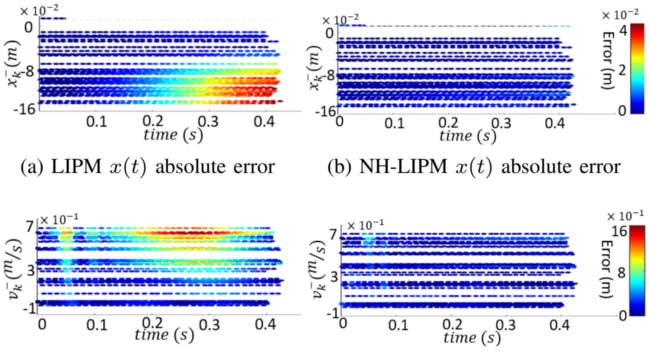
To assess the RLS controller's adaptive tuning capability for model parameters, we conducted a velocity tracking trial and compared its performance to that of the LIPM.

V. Results and Discussion

A. Regression Analysis (Simulation)

The non-homogeneous regression equation consists of a 3rd-order polynomial function of the x^+ (Eqn. 7) and a 7th-order polynomial function of the v^+ map (Eqn.8). The adjusted R-squared value for the regression analysis was 0.81 for x_k^+ and 0.79 for v_k+. The mean absolute error (MAE) of the continuous NH-LIPM regression was 0.0022 m for x and 0.0116 m/s for v. A reduction of 80% and 6%, respectively, when compared to the LIPM. The MAE of the discrete NH-LIPM regression was 0.0039 m for x and 0.012281 m/s for v. A reduction of 28% and 82%, respectively, when compared to the LIPM. Fig. 6 shows the continuous S2S modeling absolute errors of the training set using the LIPM (left) and the NH-LIPM (right). The y-axis on the plot depicts the initial states x_k^- and v_k^- of the tested data. The S2S COM position is captured poorly at faster speeds and towards the end of the step, as denoted by the large error seen at wider x_k^- and higher time (t) in Fig. 6(a). Conversely, the NH-LIPM displays errors below 1 cm, as seen in Fig. 6(b). The S2S continuous COM velocity prediction does poorly at higher v_k^- and around the middle of the cycle. However, the discrete prediction, marked by the prediction made at time $t = 0.4$ in Fig. 6(c) displays a lower error.

Fig. 7 depicts the phase portraits of the LIPM and the NH-LIPM. Although the LIPM phase portrait is state-independent, it can only predict the states of the model loosely. However, it is evident that for the given step frequency, the portrait trajectory, and dynamics closely intersect at the end of the cycle. This might be why it is possible to employ LIPM-based stepping for our system, as the error of the discrete S2S map is low. Fig. 7(b) shows the phase portrait of the NH-LIPM, which better tracks the continuous dynamics of the system. It is worth mentioning that, unlike the LIPM portrait, the phase lines in the NH-LIPM portrait will change depending on the initial states x^- and v^-. This is because the actual dynamics of the robot are state-dependent and nonlinear, which the LIPM does not account for.



(a) LIPM $x(t)$ absolute error (b) NH-LIPM $x(t)$ absolute error

(c) LIPM $v(t)$ absolute error (d) NH-LIPM $v(t)$ absolute error

Fig. 6: Surface plots of the S2S modeling errors seen in the regression

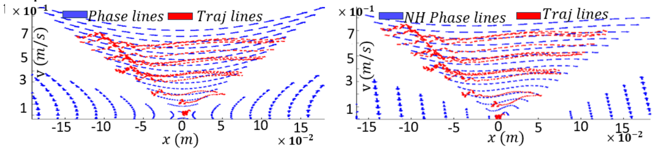


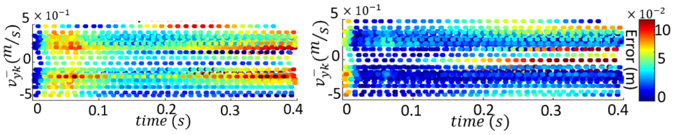
Fig. 7: (a) Test data overlayed on the LIPM phase portrait. (b)Test data overlayed on the NH-LIPM phase portrait.

The regression analysis of the lateral dynamics yielded a 3rd-order time polynomial function for the v^+ map with an adjusted R-squared value of 0.62. The lateral continuous NH-LIPM displayed a mean absolute error of 0.026 m/s, a reduction of 46% when compared to the LIPM. The discrete NH-LIPM displayed a mean absolute error of 0.0408 m/s, a reduction of 41%. Fig. 8 shows the absolute error in the continuous velocity models.

This polynomial regression analysis demonstrated the effectiveness of the NH-LIPM in capturing the dynamics of the robot. The R-squared values indicate a high correlation between the regressed parameters and the dynamics. Furthermore, the NH-LIPM showed greater accuracy in predicting the dynamics of the system compared to the LIPM. The model may be further enhanced by employing advanced learning techniques such as Gaussian Processes or neural networks.

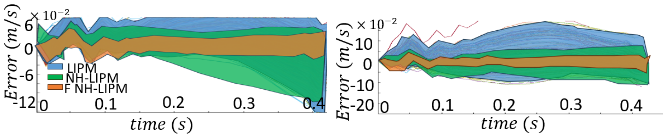
B. Regression Analysis on External Forces

Regression analysis was conducted to extract the F-NH-LIPM with two separate external forces as control. The joint-damping-control model (g in Eqn.9) is a 7th-order time polynomial function with squared states and 1st-order damping . The R-squared value for the regression was 0.82. The MAE of the continuous velocity dynamics was 0.008859



(a) LIPM $v_y(t)$ absolute error (b) NH-LIPM $v_y(t)$ absolute error

Fig. 8: Surface plots of the S2S $v_y(t)$ modelling error seen in the regression testing.



(a) Toe damping input (b) Toe force input

Fig. 9: $v_x(t)$ error band comparison.

m/s. This is an improvement of 57% when compared against the NH-LIPM and 76% against the LIPM. The MAE of the discrete model yielded an improvement of 74% and 75% compared to the NH-LIPM and LIPM, respectively.

The ankle-torque-control model is a 7th-order time polynomial function with squared states and 1st-order damping. The R-squared value for the regression was 0.86. The MAE of the continuous velocity dynamics was 0.012147 m/s. This is an improvement of 60% when compared against the NH-LIPM and 78% against the LIPM. The MAE of the discrete model yielded an improvement of 54% and 77% compared to the NH-LIPM and LIPM, respectively. Fig. 9(b) shows the test set continuous $v(t)$ error bands of the three models.

Although the NH-LIPM was improved for the general case, it showed significant prediction errors for the case when the model experiences force inputs. The F-NH-LIPM is able to capture these forces as input parameters in the model, showing improved prediction accuracy. This layered model can be expanded to include other favorable control parameters, such as a toe push-off force which may give the controller greater flexibility and the capability of performing task-specific control.

C. *Stepping Control: tracking a reference velocity*

We evaluated the performance of the model and controller framework in tracking a reference velocity profile, denoted as v_{des}^{+}. The MAE was utilized as a performance metric, calculated by subtracting the reference velocity from the actual velocity at each step and taking the mean.

When considering the frontal (x) axis, the LIPM stepping controller, as represented by Eqn. (10), yielded a MAE of 0.165 m/s. In contrast, the NH-LIPM stepping controller, described by Eqn. (11), achieved a MAE of 0.01 m/s, demonstrating a 91% improvement over the LIPM. These results are illustrated in Fig. 10a.

For the lateral (y) axis, velocity tracking was performed on the mid-cycle state $v(T_s/2)$, where $v(T_s/2)$ is employed to solve for v_s in Eqn. 14. The LIPM stepping controller exhibited a mean absolute error of 0.02 m/s, whereas the NH-LIPM stepping controller achieved a mean absolute error of 0.016 m/s. An improvement of 21% over the LIPM can be observed in Fig. 10b.

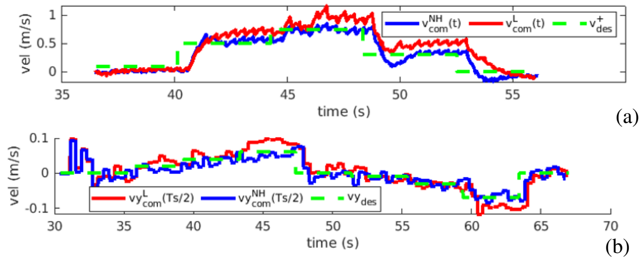


Fig. 10: **NH-LIPM**: Velocity tracking model comparison along the fore-aft (x) direction **(a)**, and the lateral (y) direction **(b)** .

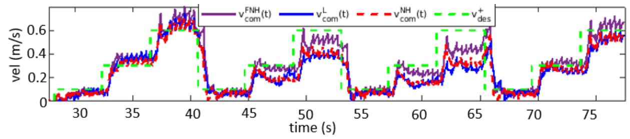


Fig. 11: **F-NH-LIPM:** Velocity tracking at 2.85, 11.42, 17.13, and 5.7 Ns/m ankle damping input.

The NH-LIPM improved velocity tracking for both the continuous and discrete models. This improvement is due to the increased model prediction in the NH-LIPM, making it better suited not only for steady-state control but also for model predictive control requiring task-specific acyclic gaits.

D. *F-NH-LIPM: tracking a reference velocity with varying ankle damping*

The F-NH-LIPM was employed to achieve velocity tracking by adjusting the damping coefficient on the stance ankle joint. The bipedal robot underwent walking experiments at three different speeds, while the toe actuators of the stance foot were augmented with four random damping coefficients (2.85, 11.42, 17.13, and 5.7 Ns/m). By incorporating the damping value as inputs to the F-NH-LIPM model, the system demonstrated superior tracking performance, as depicted in Fig. 11, exhibiting a MAE of 0.1 m/s. This represented an improvement of 21% and 16% compared to the performance of the conventional LIPM and NH-LIPM models, respectively.

The F-NH-LIPM controller with ankle-damping input tracks the reference velocity better than the NH-LIPM and LIPM. It contains parameter information that the other models don't have. This extra control parameter can expand the feasible stepping space and walking velocity achieved by the robot, as adding damping to the ankle can result in slow walking with wider steps. However, as seen in Fig. 11, even the F-NH-LIPM tracks poorly with high damping values. The model could be further enhanced by adding a higher-order polynomial to the parameter to capture the nonlinearities better.

E. *F-NH-LIPM: velocity and step width control*

The F-NH-LIPM was utilized to develop a two-input controller capable of controlling both the desired COM velocity (v^{+}) and position (x^{-}). The control inputs for this controller are the foot location at the footstrike (x_k^{-}) and the

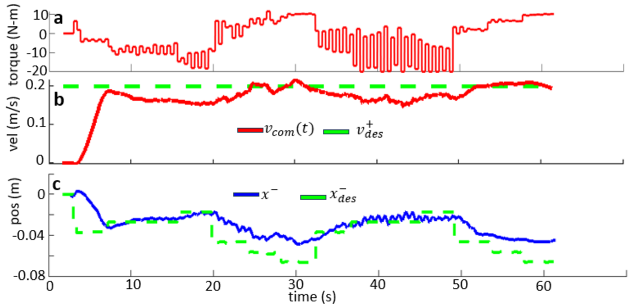


Fig. 12: **F-NH-LIPM:** Using step length and ankle torque (a) for control of v^+ (b) and x^- (c) using the F-NH-LIPM controller.

ankle torque (τ). Throughout the trial, the desired velocity (v^+_{des}) was maintained at a constant value of 0.2 m/s, while the desired position (x^-_{des}) was modulated.

As shown in Fig. 12, the robot's COM velocity, $v_{com}(t)$, closely follows the desired velocity (v^+_{des}), with a MAE of 0.032 m/s. Similarly, the state x^- successfully tracks the trajectory of the desired position (x^-_{des}), achieving a MAE of 0.016 m.

These results demonstrate the effectiveness of the F-NH-LIPM controller in accurately controlling both v^+ and x^-. This additional control input can help navigate a constrained environment where several robot states must be achieved for proper navigation. The controller is in its most basic form and can be improved with feedback control to achieve more stability and fewer steady-state errors. It is worth noting that not every control pair combination will yield a feasible response, and as such, it is essential to have boundary conditions on the controller inputs.

F. Real-Time Adaptation of Linear S2S Model

Implementing the RLS controller involved several tuning trials to determine suitable parameter values. For the frontal axis, the forgetting factor (λ) was set to 0.95, and the regularization parameter (Δ) was set to 0.1. Along the lateral axis, λ was adjusted to 0.97, and Δ was set to 0.99.

Fig. 13 illustrates the velocity tracking performance of the RLS controller compared to the LIPM controller. In the forward direction, the MAE achieved by the RLS controller was 0.0604 m/s, which is 43% lower than the MAE obtained by the LIPM controller. In the lateral direction, the MAE achieved by the RLS controller was 0.0648, an improvement of 21% over the LIPM. During lateral walking using the LIPM controller, the model discrepancy caused the robot to tip over, as shown in Fig. 13. These results demonstrate the superior tracking capabilities of the RLS controller in comparison to the traditional LIPM controller.

Additionally, Fig. 14 showcases the adaptive tuning of the RLS parameters a_{21} and a_{22} along the x and y axes, respectively, as the damping is varied. This adaptive tuning capability allows the RLS controller to dynamically adjust its model parameters to handle changes in the damping conditions effectively. Parameter convergence is achieved in less than 5 seconds on average.

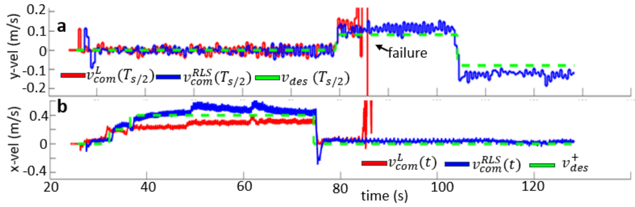


Fig. 13: **RLS vs LIPM:** Lateral (a) and forward (b) walking velocity tracking comparison between the adaptive RLS and the LIPM stepping controllers. Failure occurs during lateral walking with LIPM controller.

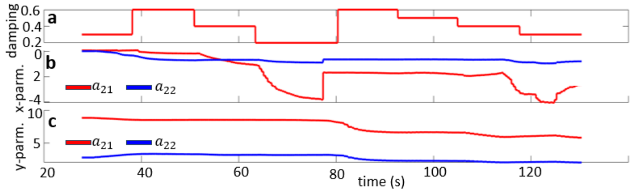


Fig. 14: **RLS controller:** (a) Deliberate damping variations were used in the trial. (b)-(c) the RLS controller performed online tuning of the model parameters

These findings highlight the effectiveness of the RLS controller in achieving improved velocity tracking and its ability to adaptively tune its parameters based on the changing conditions of the system. With an adaptive model, a ”close-enough” representation of the robot's dynamics can be extracted, which can then be utilized to develop stepping controllers. This eliminates the need for a time-consuming trial-and-error process typically associated with developing accurate models for walking controllers.

G. Preliminary Hardware Results: Extracting the Non-homogeneous LIPM

The regression analysis on hardware yielded an R-squared value of 0.87 with a MAE of the continuous NH-LIPM of 0.0156 m/s, which is an improvement of 55% compared to the LIPM. The MAE of the discrete NH-LIPM was 0.01369 m/s, an improvement of 75% compared to the LIPM.

The results of the velocity tracking experiment is shown in Fig. 15. The tracking MAE of the NH-LIPM controller was 0.043 m/s, an improvement of 48% compared to the LIPM controller. This preliminary experiment shows the effectiveness of the regression-based NH-LIPM on hardware. The tracking MAE was reduced when compared to the traditional LIPM controller. The NH-LIPM controller, however, displayed less steady-state stability due to the use of a nonlinear model. The stability will be addressed via feedback control and controller tuning in future studies.

VI. Conclusion and Future Work

Our findings demonstrate the enhanced performance and effectiveness of the NH-LIPM in predicting the dynamics of the bipedal robot Digit. The NH-LIPM introduces a time and state-dependent component that enhances the accuracy of the robot's S2S dynamics. By extracting the solution to the non-homogeneous term through regression analysis, we account for the time/state-varying aspects of the robot's behavior, resulting in a more comprehensive model that better captures the real-world dynamics. Hardware testing yielded promising results.

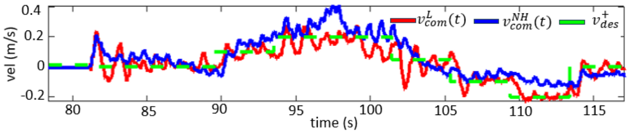


Fig. 15: **Hardware:** Velocity tracking comparison between NH-LIPM and the LIPM performed on Digit.

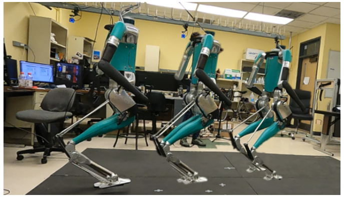


Fig. 16: v^+_{des} tracking hardware experiment using the NH-LIPM controller.

The NH-LIPM was supplemented with a forced term, with ankle damping and torque as input parameters, to develop the F-NH-LIPM. The F-NH-LIPM controller accurately and simultaneously controlled the COM velocity and COM position of the robot with respect to the stance foot. This refined model enables a more effective control strategy facilitating navigation in environments that require simultaneous control of multiple states, such as foot placement and COM velocity.

The RLS algorithm was used to adaptively tune the parameters of the particular solution to the NH-LIPM. The RLS controller with forgetting, effectively modulated the parameters and reacted to deliberate changes in the robot dynamics like height, step width, and ankle damping. A velocity-tracking experiment showed the RLS controller's ability to track a reference velocity with changing ankle damping.

Overall, the findings of this study support the use of the NH-LIPM as a valuable model for understanding and controlling bipedal walking. The incorporation of non-homogeneous terms through regression analysis improves the accuracy of the model, enabling more precise predictions of the COM position and velocity.

Future work will center on implementing the discussed methodologies on hardware, with a key focus on enhancing the stability of the hardware NH-LIPM. To bolster the predictive capabilities, a more extensive training set will be utilized, augmenting the convex hull of the regression function. Experiments will be conducted to examine the NH-LIPM controller on lateral stepping. Additionally, the integration of ankle damping and torque as control inputs to the F-NH-LIPM will be explored on hardware. Lastly, to validate the model's practical applicability, a model predictive stepping controller will be deployed, employing the F-NH-LIPM to navigate challenging terrains with obstacles. This comprehensive evaluation of real-world hardware will further enhance the model's efficacy and facilitate its integration into bipedal robotic research.

References

[1] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-theory and basic experiment," in *Proc. of the IEEE International Conference on Robotics and Automation, Sacramento, California, USA*, 1991, pp. 741–746.

[2] J. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Transactions on Automatic Control*, vol. 46, no. 1, pp. 51–64, 2001.

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

[4] T. McGeer, "Passive dynamic walking," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[5] S. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.

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

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

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

[9] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, pp. 42–56, 2003.

[10] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "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, 2016, pp. 1447–1454.

[11] C. Chevallereau, J. Grizzle, and C. Shih, "Asymptotically stable walking of a five-link underactuated 3-d bipedal robot," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 37–50, 2009.

[12] J. Seipel, M. Kvalheim, S. Revzen, M. A. Sharbafi, and A. Seyfarth, "Conceptual models of legged locomotion," in *Bioinspired Legged Locomotion*. Elsevier, 2017, pp. 55–131.

[13] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.

[14] E. H. Hinojosa, D. Torres, and P. A. Bhounsule, "Quadratically constrained quadratic programs using approximations of the step-to-step dynamics: application on a 2d model of digit," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 96–103.

[15] Y. Gao, Y. Gong, V. Paredes, A. Hereid, and Y. Gu, "Time-varying alip model and robust foot-placement control for underactuated bipedal robotic walking on a swaying rigid surface," in *2023 American Control Conference (ACC)*, 2023, pp. 3282–3287.

[16] J. Hrr, J. Pratt, C.-M. Chew, H. Herr, and G. Pratt, "Adaptive virtual model control of a bipedal walking robot," in *Proceedings. IEEE International Joint Symposia on Intelligence and Systems (Cat. No.98EX174)*, 1998, pp. 245–251.

[17] V. C. Paredes and A. Hereid, "Resolved motion control for 3d underactuated bipedal walking using linear inverted pendulum dynamics and neural adaptation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6761–6767.

[18] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Robust feedback motion policy design using reinforcement learning on a 3d digit bipedal robot," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5136–5143.

[19] X. Xiong and A. D. Ames, "Orbit characterization, stabilization and composition on 3d underactuated bipedal walking via hybrid passive linear inverted pendulum model," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4644–4651.

[20] M. Ackermann and A. Van den Bogert, "Optimality principles for model-based prediction of human gait." *Journal of biomechanics*, vol. 43, pp. 1055–1060, 2010.

[21] X. Xiong and A. Ames, "3d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization," 2021. [Online]. Available: https://arxiv.org/abs/2101.09588

[22] [Online]. Available: http://tiny.cc/NHLIPM