10 Steps Forward: Simulation Of A 3-Link Bipedal Robot

Abhijit Kaluri (kaluri.a@northeastern.edu)
Haoxin Liu (liu.haox@northeastern.edu)
Kevin Wu (wu.kevi@northeastern.edu)
Nagavarun Samala (samala.n@northeastern.edu)

***Abstract*—The control of bipedal systems poses a complex challenge in robotics, necessitating robust and sophisticated solutions. This report which is a combination of 4 mini projects synthesizes an approach to model and control a three-link biped robot, primarily leveraging the Hybrid Zero Dynamics (HZD) framework developed at the University of Michigan. The kinematics, dynamic modeling, and controller design are thoroughly developed to implement the hybrid zero dynamics of the biped. Utilizing zero dynamics, we have engineered an energy-efficient gait, incorporating virtual holonomic constraints controlled as a function of an unactuated cyclic variable through a feedback linearizing controller. A Bezier polynomial is optimized to design this gait, defined by a monotonically increasing gait timing variable. Further, the full dynamics are integrated with a Proportional-Derivative (PD) controller to enhance system stability and ensure the biped adheres to the zero dynamics manifold. Simulation results verify the effectiveness of the designed autonomous gait, highlighting the model's ability to maintain stability with a single foot contact and instantaneous impacts, fulfilling the conditions for optimal gait design and energy efficiency in legged locomotion.**

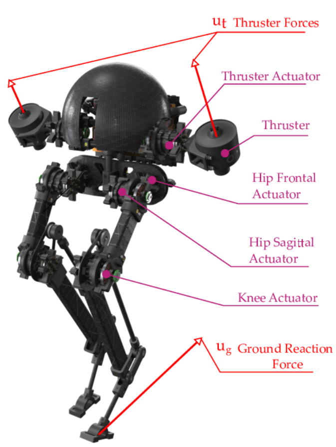

Fig. 1. Harpy - A Thruster Assisted Bipedal robot developed at Northeastern University

I. MINI PROJECT 1: FORWARD KINEMATIC ANALYSIS

A. *Symbolic Variables and Equations for a Three-Link Bipedal Robot*

This section outlines the symbolic mathematics used to model the dynamics of a three-link bipedal robot. The conventions and equations described here assume that negative angles are measured clockwise from the vertical, adhering to the standard mathematical convention used in robotics.

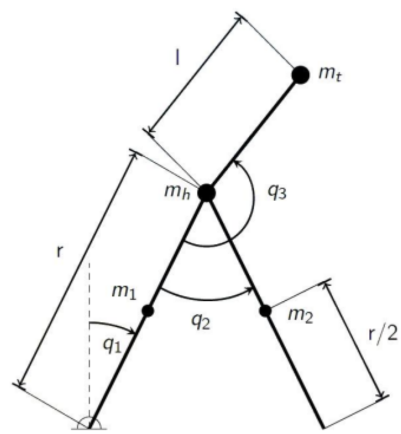

Fig. 2. 3-link-Biped Diagram

B. *Model Parameters*

The following table summarizes the physical parameters used in the dynamic modeling of the three-link bipedal robot:

TABLE I
MODEL PARAMETERS

Parameter	Value	Description
m_1	5 kg	Mass of swing leg
m_2	5 kg	Mass of stance leg
m_h	15 kg	Mass of hip
m_t	10 kg	Mass of torso
l	0.5 m	Length of torso
r	1 m	Length of each leg

C. *Definition of Symbolic Variables*

Generalized Coordinates and Velocities:

- q_1, q_2, q_3: Angular positions of each link.
- $\dot{q}_1, \dot{q}_2, \dot{q}_3$: Angular velocities of each link.

Physical Parameters:

- r, m, M_h, M_t, l, g: Represent physical characteristics of the robot:

- r: Radius or effective length of the links.
- m: Mass of each link (assumed uniform).
- M_h, M_t: Masses of the hip and torso, respectively.
- l: Length between the hip and the torso.
- g: Acceleration due to gravity.

D. Position Vectors

Hip Position (p_{Mh}):

$$p_{Mh} = \begin{bmatrix} r\sin(-q_1) \\ r\cos(-q_1) \end{bmatrix} \tag{1}$$

Torso Position (p_{Mt}):

$$p_{Mt} = p_{Mh} + \begin{bmatrix} l\sin(-q_1 + \pi - q_3) \\ l\cos(-q_1 + \pi - q_3) \end{bmatrix} \tag{2}$$

Mass Centers (p_{m1}, p_{m2}):

$$p_{m1} = 0.5 \times r \times p_{Mh}, \tag{3}$$
$$p_{m2} = p_{Mh} + \frac{r}{2} \times \begin{bmatrix} \sin(q_2 + q_1) \\ -\cos(q_2 + q_1) \end{bmatrix} \tag{4}$$

Center of Mass of the Entire System (p_{cm}):

$$p_{cm} = \frac{m \times p_{m1} + m \times p_{m2} + M_h \times p_{Mh} + M_t \times p_{Mt}}{2m + M_h + M_t} \tag{5}$$

Position of the Second Link (P_2):

$$P_2 = p_{Mh} + \begin{bmatrix} r\sin(q_2 + q_1) \\ -r\cos(q_2 + q_1) \end{bmatrix} \tag{6}$$

II. MINI PROJECT 2: DYNAMICS MODEL

This section details the symbolic computations used to derive the equations of motion, kinetic and potential energies, and other control matrices for the three-link bipedal robot, using the symbolic variables and relations defined earlier.

A. Velocity Computations

The velocities of each mass center are calculated using the Jacobian of their position vectors with respect to the generalized coordinates:

$$\mathbf{v}_{Mh} = J_{pMh}(\mathbf{q}) \cdot \dot{\mathbf{q}}, \tag{7}$$
$$\mathbf{v}_{Mt} = J_{pMt}(\mathbf{q}) \cdot \dot{\mathbf{q}}, \tag{8}$$
$$\mathbf{v}_{m1} = J_{pm1}(\mathbf{q}) \cdot \dot{\mathbf{q}}, \tag{9}$$
$$\mathbf{v}_{m2} = J_{pm2}(\mathbf{q}) \cdot \dot{\mathbf{q}}, \tag{10}$$
$$\mathbf{v}_{cm} = J_{pcm}(\mathbf{q}) \cdot \dot{\mathbf{q}}. \tag{11}$$

B. Kinetic and Potential Energy

The kinetic energy for each component is computed as follows:

$$K_{Mh} = \frac{1}{2} M_h (\mathbf{v}_{Mh}^\top \mathbf{v}_{Mh}), \tag{12}$$
$$K_{Mt} = \frac{1}{2} M_t (\mathbf{v}_{Mt}^\top \mathbf{v}_{Mt}), \tag{13}$$
$$K_{m1} = \frac{1}{2} m (\mathbf{v}_{m1}^\top \mathbf{v}_{m1}), \tag{14}$$
$$K_{m2} = \frac{1}{2} m (\mathbf{v}_{m2}^\top \mathbf{v}_{m2}), \tag{15}$$
$$K = K_{m1} + K_{Mh} + K_{Mt} + K_{m2}. \tag{16}$$

The potential energy for each component is defined as:

$$V_{Mh} = pMh_2 \cdot M_h \cdot g, \tag{17}$$
$$V_{Mt} = pMt_2 \cdot M_t \cdot g, \tag{18}$$
$$V_{m1} = pm1_2 \cdot m \cdot g, \tag{19}$$
$$V_{m2} = pm2_2 \cdot m \cdot g, \tag{20}$$
$$V = V_{m1} + V_{Mh} + V_{Mt} + V_{m2}. \tag{21}$$

C. Inertia, Coriolis, and Gravity Matrices

The inertia matrix D, Coriolis matrix C, and gravity vector G are derived as follows:

$$D = (\text{jacobian}(\text{jacobian}(K, \dot{\mathbf{q}})^\top, \dot{\mathbf{q}})), \tag{22}$$
$$C(k,j) = \sum_{i=1}^{N} \left(\frac{1}{2} (\frac{\partial D_{kj}}{\partial q_i} + \frac{\partial D_{ki}}{\partial q_j} - \frac{\partial D_{ij}}{\partial q_k}) \dot{q}_i \right), \tag{23}$$
$$G = (\text{jacobian}(V, \mathbf{q})^\top). \tag{24}$$

The control input matrix B is defined statically in the system as:

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{25}$$

D. Impact Map and Enhanced Dynamic Modeling

Post-impact equations are derived to model the dynamics immediately after an impact:

$$p_e = \begin{bmatrix} p_h \\ p_v \end{bmatrix}, \tag{26}$$
$$\mathbf{q}_e = \begin{bmatrix} \mathbf{q} \\ p_h \\ p_v \end{bmatrix}, \tag{27}$$
$$\dot{\mathbf{q}}_e = \begin{bmatrix} \dot{\mathbf{q}} \\ dp_h \\ dp_v \end{bmatrix}, \tag{28}$$
$$\mathbf{v}_{Mh\_e} = J_{pMh_e}(\mathbf{q}_e) \cdot \dot{\mathbf{q}}_e, \tag{29}$$
$$K_e = K_{m1e} + K_{Mhe} + K_{Mte} + K_{m2e}, \tag{30}$$
$$D_e = (\text{jacobian}(\text{jacobian}(K_e, \dot{\mathbf{q}}_e)^\top, \dot{\mathbf{q}}_e)), \tag{31}$$
$$E = (\text{jacobian}(P2_e, \mathbf{q}_e)). \tag{32}$$

These equations are fundamental for further analysis in control design, ensuring that the biped robot's motion is optimized for stability and efficiency. The transition from pre-impact to post-impact conditions involves calculating new velocities and positions, critical for continuous motion simulation and analysis.

III. MINI PROJECT 3: GAIT DESIGN

A. Zero-Dynamics Design

This section details the implementation of Zero Dynamics (ZD) for the three-link bipedal robot, integrating the dynamic modeling and control aspects to design an optimized walking gait.

B. Dynamic Equations

The mathematical model for the biped's walking gait encompasses key dynamic equations essential for simulation and control:

- $H(q)$: Mass matrix, dependent on the generalized coordinates q.
- $C(q, \dot{q})$: Coriolis and centripetal forces matrix.
- $G(q)$: Gravitational force vector.
- τ: Generalized torque vector.
- q_i^+: Updated coordinates post-leg switch.
- $f(q, \dot{q})$: Nonlinear function for leg switching dynamics.

These components facilitate transitions between single-support and double-support phases during the gait.

C. Hybrid Model

The hybrid model describes the biped dynamics through various walking gait phases, categorized into:

- Single-support phase: Only one leg contacts the ground.
- Double-support phase: Both legs make ground contact.

Transitions between these phases are governed by the nonlinear function $f(q, \dot{q})$, determining leg switch events. Upon a leg switch, coordinates are relabeled as q_i^+.

This comprehensive model encapsulates the forward kinematics, motion equations, and a hybrid model outlining the gait's phases. The implemented MATLAB code includes the mass, Coriolis and centripetal matrices, gravitational forces vector, and the generalized torque vector, along with the nonlinear function for leg switching. The model successfully simulates the biped's motion during the walking gait, capturing the leg switch events during the double-support phase effectively.

D. Equations from MATLAB Code

The mathematical model for the biped's walking gait incorporates several key equations, represented in MATLAB code as follows:

- $H(q)$: The mass matrix, depicting the mass and inertia distribution across the biped, dependent on the generalized coordinates q.
- $C(q, \dot{q})$: The Coriolis and centripetal forces matrix, accounting for velocity and acceleration effects, reliant on both the generalized coordinates q and velocities $\dot{q}$.
- $G(q)$: The gravitational force vector, illustrating the gravity's impact on the system, based on the generalized coordinates q.
- τ: Represents the generalized torque vector, indicating external torques applied to the system.
- q_i^+: New generalized coordinates post-leg switch, ensuring model continuity by relabeling coordinates q_i.
- $f(q, \dot{q})$: A nonlinear function delineating leg switching during the walking gait, as a function of q and $\dot{q}$.

These equations enable the model to simulate the biped's walking gait comprehensively, covering transitions between single-support and double-support phases, as well as leg switches during the double-support phase.

E. Swing Phase Model

The approach for the swing phase model involves leveraging the previously derived extended positions and velocities of the biped's links. The methodology focuses on implementing the zero dynamics for gait planning, utilizing the inherent constraints provided by the zero dynamics to partition the complete dynamic model comprehensively.

The dynamic equation characterizing the swing phase is formally represented as:

$$D(q, \ddot{q}) + C(q, \dot{q})\dot{q} + G(q) = Bu \tag{33}$$

This model represents a pinned kinetic chain, emphasizing symmetric gaits. Consequently, the specific leg serving as the stance leg is interchangeable, with the impact model accounting for the alternation of leg roles.

The state-space representation of the entire dynamics is articulated as:

$$\dot{x} = f(x) + g(x)u \tag{34}$$

where $x = [q, \dot{q}]$ and u denotes the control input applied by actuators. The functions $f(x)$ and $g(x)$ are defined as:

$$f(x) = \begin{bmatrix} \dot{x} \\ -D^{-1}C(q, \dot{q})\dot{q} - D^{-1}G \end{bmatrix}, \tag{35}$$

$$g(x) = \begin{bmatrix} 0 \\ -D^{-1}B \end{bmatrix}. \tag{36}$$

Dynamic model partitioning yields:

$$\begin{bmatrix} D11 & D12 \\ D21 & D22 \end{bmatrix} \begin{bmatrix} \ddot{q}_n \\ \ddot{q}_b \end{bmatrix} + \begin{bmatrix} H1 \\ H2 \end{bmatrix} = Bu, \tag{37}$$

where $q_b = [q_2, q_3]^T$, $q_n = q_1$, and matrix components are specifically delineated as $D11 = D_{1,1}$, $D12 = [D_{1,2}, D_{1,3}]$, and so forth.

For the purpose of gait design, a normalized variable s is introduced as:

$$s = \frac{q_1 - q_1^+}{\Delta q_1^{\max}}, \tag{38}$$

where $\Delta q_1^{\max} = q_1^- - q_1^+$. The output functions y_1 and y_2 are defined to adhere to Bezier curve polynomials:

$$y_1 = q_2 - h_1(s), \tag{39}$$

$$y_2 = q_3 - h_2(s). \tag{40}$$

Considering the Zero Dynamics: For the gait planning, Bezier curve polynomials $h_i(s)$ are utilized. In the scenario of zero dynamics, both $y = 0$ and $\dot{y} = 0$ are satisfied, leading to the equations:

$$q_2 = h_1(s), \tag{41}$$
$$q_3 = h_2(s). \tag{42}$$

Given $h_i(s)$ represents the Bezier curve polynomials, the second-order derivatives of q_2 and q_3 with respect to time are zero, indicating a steady state during zero dynamics. Thus, the equations simplify to:

$$\ddot{q}_2 = 0, \tag{43}$$
$$\ddot{q}_3 = 0. \tag{44}$$

Substituting the values of $\ddot{q}_b = [\ddot{q}_2, \ddot{q}_3]$ into the previously defined dynamics equation, we obtain:

$$(D_{11} + D_{12}\beta_1)\ddot{q}_1 + D_{12}\beta_2 + H_1 = 0, \tag{45}$$

where β_1 and β_2 are coefficients derived from the dynamics partitioning process and are crucial for achieving the desired zero dynamics during the swing phase of the biped's gait.

F. Zero-Dynamics Equations for the Three-Link Biped

For a three-link biped traversing the sagittal plane, the zero-dynamics equations are articulated as follows:

$$q_1 = z(1), \tag{46}$$
$$q_2 = \text{bezier}(s; M; \alpha_2), \tag{47}$$
$$q_3 = \text{bezier}(s; M; \alpha_3), \tag{48}$$
$$\dot{q}_1 = z(2), \tag{49}$$
$$\dot{q}_2 = \frac{d}{ds}\text{bezier}(s; M; \alpha_2) \times \frac{\dot{q}_1}{\delta q}, \tag{50}$$
$$\dot{q}_3 = \frac{d}{ds}\text{bezier}(s; M; \alpha_3) \times \frac{\dot{q}_1}{\delta q}, \tag{51}$$

where s symbolizes the normalized gait timing variable, $\alpha_{i,j}$ are the Bezier coefficients, and $B_{j,M}(s)$ is the j^{th} Bezier polynomial of order M.

G. Dynamics Partitioning for Zero-Dynamics Equations

The zero-dynamics equations are derived through the partitioning of dynamics as follows:

- $D_1 = D_{11}$, which signifies the top-left entry of the full dynamics matrix D.
- $D_2 = D_{12:13}$, representing the combination of top-right and bottom-left entries of D.
- $H_1 = C_{11} \times \dot{q}_1 + G_{11}$, where C_{11} and G_{11} correspond to the first entries of the Coriolis and gravity vectors, respectively.
- $\beta_1 = \text{func\_compute\_}\beta_1(s; [\dot{q}_1; z_{\max} - z_{\min}]; [\alpha_2; \alpha_3])$, a function that calculates the β_1 term for the zero-dynamics equations.
- $\ddot{q}_1 = (D_1 + D_2 \times \beta_2) \cdot (D_2 \times \beta_1 - H_1)$, where β_2 is another function determining a term for the zero-dynamics equations.

H. Impact Phase Model

The impact phase model delineates the transformation of the states from pre-impact to post-impact conditions, taking into account potential energy losses during the ground contact moment. By defining the Cartesian coordinates of the robot's center of mass, we establish:

$$q_e = [q, p_h, p_v]. \tag{52}$$

I. Hypotheses for the Impact Model

The model is predicated on several key hypotheses:

1) The swing leg, upon ground contact, neither rebounds nor slips.
2) The stance leg disengages from the ground without interaction.
3) The double support phase is considered instantaneous, allowing the impact to be modeled as a rigid contact event.
4) Post-impact, the roles of the stance and swing legs are interchanged, reflective of a symmetric gait.

J. Model Definition

The center of mass (cm) of the robot is denoted in Cartesian coordinates as $p_e = [p_{cm1}, p_{cm2}]^T$. The generalized coordinates in a single support model are denoted as q_s, and for the unpinned model, we utilize $q_e = [q_s, p_e]^T$. Let $p_e = \gamma_e(q_s)$, which leads to the pre-impact state equations:

$$q_-^e = \begin{bmatrix} q_-^s \\ \gamma_e(q_-^s) \end{bmatrix}, \tag{53}$$

$$\dot{q}_-^e = \begin{bmatrix} I_{N\times N} \\ \frac{\partial}{\partial q_s}\gamma_e(q_-^s) \end{bmatrix} \cdot \dot{q}_-^s. \tag{54}$$

The Jacobian matrix E_2, mapping external forces from joint space to task space, and the force vector F_2 acting at the swing leg's end are defined as follows:

$$E_2 = \frac{\partial}{\partial q_s} p_2(q_e), \tag{55}$$

$$F_2 = \begin{pmatrix} F_{T_2} \\ F_{N_2} \end{pmatrix}. \tag{56}$$

Here, $p_2(q_e)$ represents the position of the swing leg's tip with respect to the inertial frame, encapsulating the impact dynamics succinctly as $x_+ = \Delta(x_-)$.

K. Zero Dynamic Model

The zero dynamic model is derived from the full dynamics model, which is essential for gait planning. The full dynamics model integrates the dynamic equations and constraints into a unified framework. This model allows the simulation of the robot's motion and control through the gait cycle. The full dynamics model, inclusive of the impact phase and the swing phase dynamics, is formalized as:

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u, & x_- \notin S, \\ x_+ = \Delta(x_-), \end{cases} \tag{57}$$

where the set S is defined as $S := \{(q_s, \dot{q}_s) \in TQ_s | p_{v2}(q) = 0, p_{h2} > 0\}$, representing the conditions for the transition from one phase of the gait to another.

The zero dynamics model, encapsulating the inherent dynamics of the system without external inputs, is a subset of the full dynamics, focusing on the natural motion patterns of the biped. It is expressed by amalgamating equations (14) and (20), highlighting the robot's behavior under zero control input. This model plays a crucial role in understanding the intrinsic properties of the biped's motion and in designing control strategies that leverage these natural dynamics for efficient locomotion.

L. Gait Optimization

The optimization of the biped's gait utilizes a fourth-order Bézier curve, defined for joint variables q_2 and q_3 as follows:

$$b_i(s) = \sum_{k=0}^{4} \alpha_{i,k} \binom{4}{k} s^k (1-s)^{4-k}, \quad i \in \{2,3\}, \tag{58}$$

where $s \in [0,1]$ represents the gait timing variable, transitioning from the start to the end of the gait. The Bézier coefficients $\alpha_{i,k}$ are optimized to ensure an energy-efficient gait by minimizing the mechanical cost.

- $q_1, \dot{q}_1$: Pre-impact conditions, both negative:

$$\mathbf{z}^- = \begin{bmatrix} -0.2618 \\ -1.2 \end{bmatrix}$$

- Bézier coefficients:

$$\boldsymbol{\alpha} = [0.3, 0.6, 0.51] \text{ for } q_2 \text{ with additional } \alpha_{3-5}$$

- Coefficients for the optimization cost:

$$\boldsymbol{\gamma} = [2.40, 2.50, 2.58]$$

- Epsilon for angle:

$$\epsilon = \frac{15\pi}{180}$$

- Epsilon for velocity:

$$\epsilon_v = \frac{75\pi}{180}$$

- Cost = sum of norms of control action:

$$J = J + \|u\|^2$$

The second-order derivatives yield $\dot{q}_i = 0$ and $\ddot{q}_i = 0$ for $i = 2, 3$. Substituting $\ddot{q}_b = [\ddot{q}_2, \ddot{q}_3]$ into the zero dynamics equation facilitates an expression of the form:

$$(D_{11} + D_{12}\beta_1)\ddot{q}_1 + D_{12}\beta_2 + H_1 = 0. \tag{59}$$

This equation underpins the subsequent gait design process, incorporating the zero dynamics framework to achieve an optimized walking gait.

M. Plot

The optimized values obtained through MATLAB's optimizer are encapsulated in:

$$f = [0.2819 -1.8221 0.0414 0.3391 0.5341 2.6617 2.4783 2.3184]^\top \tag{60}$$

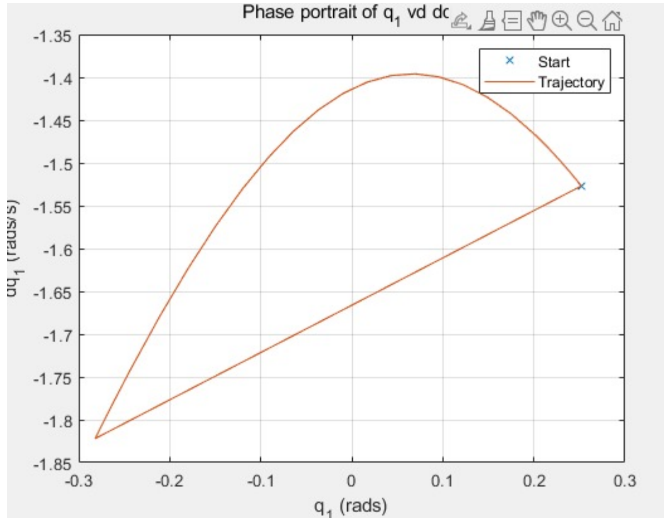

Fig. 3. Output of the optimizer

IV. MINI PROJECT 4: DESIGNING NON-LINEAR CONTROLLER

A. Control Action Implementation Using Feedback Linearization

This section outlines the mathematical formulation of the control strategy implemented in MATLAB for a three-link biped robot, utilizing feedback linearization to compute the control actions needed for achieving desired gait dynamics.

B. State Variables and Parameters

- **Generalized Coordinates:** $\mathbf{q} = [q_1, q_2, q_3]^T$
- **Generalized Velocities:** $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$
- **Bezier Coefficients for q_2 and q_3:** α_2 and α_3
- **Gait Timing Parameters:** $q_{1_{\min}}$, Δq

C. Control Inputs Calculation

The control inputs are calculated as follows:

$$s = \frac{q_1 - q_{1_{\min}}}{\Delta q} \tag{61}$$

D. System Dynamics

The dynamics of the robot are captured by the following state-space representation:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \tag{62}$$

where:

$$\mathbf{D}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} = \mathbf{B}\mathbf{u} \tag{63}$$

and decomposed into:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{\mathbf{q}} \\ -\mathbf{D}^{-1}(\mathbf{C}\dot{\mathbf{q}} + \mathbf{G}) \end{bmatrix}, \quad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}^{-1}\mathbf{B} \end{bmatrix} \tag{64}$$

E. Output Function and its Derivatives

Define the output function $\mathbf{y}$ related to Bezier curves:

$$\mathbf{y} = \begin{bmatrix} q_2 - \text{bezier}(s, 4, \alpha_2) \\ q_3 - \text{bezier}(s, 4, \alpha_3) \end{bmatrix} \tag{65}$$

Linearized dynamics around the output function:

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{L}_f \mathbf{h}(\mathbf{x}) \tag{66}$$

Second derivative (for feedback linearization):

$$\ddot{\mathbf{y}} = \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{L}_2 \mathbf{f} \mathbf{h} + \mathbf{L}_g \mathbf{L}_f \mathbf{h} \cdot \mathbf{u} \tag{67}$$

F. Proportional-Derivative (PD) Controller

The control action computed to drive $\mathbf{y}$ to zero:

$$\mathbf{v} = \frac{1}{\epsilon^2} \mathbf{K}_p \mathbf{y} + \frac{1}{\epsilon} \mathbf{K}_d \dot{\mathbf{y}} \tag{68}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are diagonal matrices containing proportional and derivative gains respectively.

G. Final Control Law Using Feedback Linearization

The control input $\mathbf{u}$ is designed to cancel out the nonlinearities and drive the error dynamics:

$$\mathbf{u} = -\mathbf{L}_g \mathbf{L}_f \mathbf{h}^{-1} (\mathbf{L}_2 \mathbf{f} \mathbf{h} + \mathbf{v}) \tag{69}$$

V. Simulation of Dynamics and Impact Mapping

This section elaborates on the preparation, execution, and analysis of dynamics simulations for a three-link biped robot, utilizing precomputed Bezier coefficients and the impact map to simulate the walking gait.

A. Extraction of Parameters and Pre-Impact Conditions

Firstly, parameters and initial conditions are extracted and prepared for the simulation:

- **Bezier Coefficients Extraction:** The Bezier coefficients for the trajectories of joints q_2 and q_3 are calculated from a predefined array f:

$$\text{alpha2} = [-f(5), -f(4), f(3:5)];$$
$$\text{alpha3} = [-f(8) + 2 * f(6), -f(7) + 2 * f(6), f(6:8)];$$

- **Initial Conditions:** The initial conditions for q_1 and its derivative $\dot{q}_1$ are set based on the extracted values:

$$q1\_minus = f(1);$$
$$dq1\_minus = f(2);$$

- **Range Calculation for Gait:** The maximum and minimum angles for q_1 during a single gait are determined to ensure symmetry within the gait:

$$x\_max = q1\_minus;$$
$$x\_min = -q1\_minus;$$
$$delq = x\_max - x\_min;$$

B. Impact Mapping and Simulation Setup

The impact dynamics are evaluated using the pre-impact states and Bezier coefficients:

- **Impact States Calculation:** Using the last Bezier coefficient for immediate pre-impact positions and velocities:

$$q2\_minus = \text{alpha2}(5);$$
$$q3\_minus = \text{alpha3}(5);$$
$$dq2\_minus = 4 \cdot (\text{alpha2}(5) - \text{alpha2}(4)) \cdot \frac{dq1\_minus}{delq};$$
$$dq3\_minus = 4 \cdot (\text{alpha3}(5) - \text{alpha3}(4)) \cdot \frac{dq1\_minus}{delq};$$

Pre-Impact Conditions Vector:

$$x\_minus = [q1\_minus, q2\_minus, q3\_minus, dq1\_minus, dq2\_minus, dq3\_minus]; \tag{70}$$

- **Impact Map Calculation:** The function `func_impact_map` calculates the states right after impact:

$$[x\_plus0, \sim] = \texttt{func\_impact\_map}(x\_minus); \tag{71}$$

- **Simulation Parameters Setup:** The starting conditions for the simulation are set, taking the output of the impact map as the initial conditions.

$$s\_params = [x\_plus0(\text{end}, 1), x\_max]; \tag{72}$$

C. Execution of Dynamics Simulation

The dynamics of the biped robot are simulated over multiple steps, integrating the dynamics using an ODE solver:

- **ODE Solver Setup:** An ODE solver is configured with event detection settings to simulate the walking gait over multiple steps.
- **Loop for Multiple Steps:** The simulation is run over a specified number of steps, with the state updated after each step based on the impact dynamics.

D. Results Visualization

Upon completion of the simulation, trajectories are plotted, and animations are generated to visualize the behavior of the biped robot throughout its gait cycle:

- **Plotting Trajectories:** `plot_trajectories(t_tot, x_tot);`
- **Animating Results:** `animate_results(t_tot, x_tot);`

This section encapsulates the entire process from parameter extraction through to the dynamic simulation and visualization, critical for evaluating the designed control strategy and the robot's performance.

The optimized values obtained through MATLAB's optimizer are encapsulated in:

$$f = [0.2819 -1.8221 0.0414 0.3391 0.5341 2.6617 2.4783 2.3184]^\top \tag{73}$$

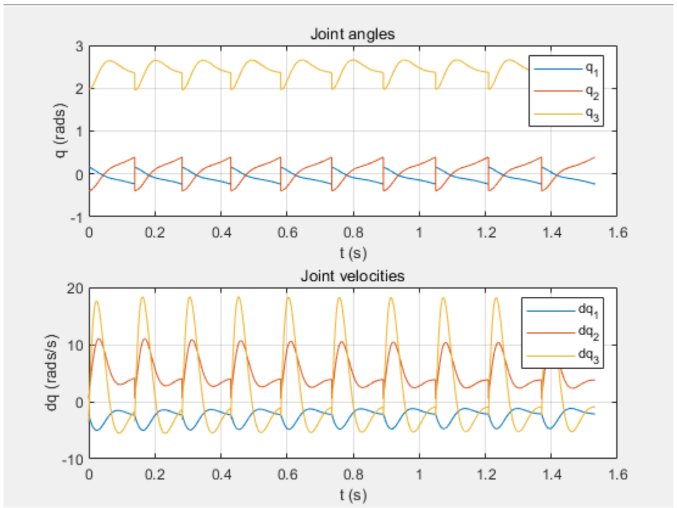

Fig. 4. Joint Angles and Joint Velocity Plots

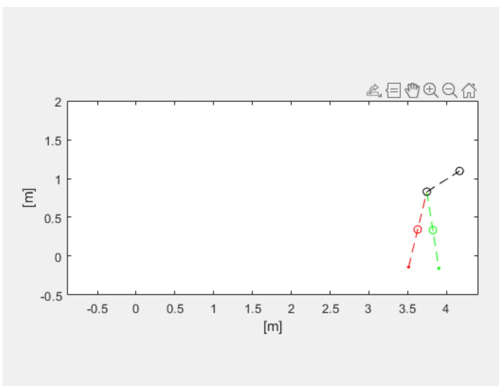

Fig. 5. Animation

E. *Poincare Analysis*

By analyzing the phase portrait of the joint angle q_1 and its velocity $\dot{q}_1$, we observe the limit-cycle behavior indicative of a stable gait. This outcome is the result of careful tuning of the gait parameters within the zero dynamics framework. The Poincaré analysis further confirms this stability, as the model's trajectories converge to a single closed loop after a few steps, despite minor variations in the initial conditions. This demonstrates that the feedback linearization controller and the Hybrid Zero Dynamics approach used in this study are effective in producing a stable autonomous walking gait for the bipedal robot.

VI. Conclusion

This report has presented a comprehensive approach to the design and control of a three-link bipedal robot, employing the Hybrid Zero Dynamics (HZD) framework and feedback linearization techniques. The integration of these advanced control strategies has resulted in a robust model capable of simulating and controlling a biped robot with precision and energy efficiency.

Throughout the four mini-projects, we have successfully:

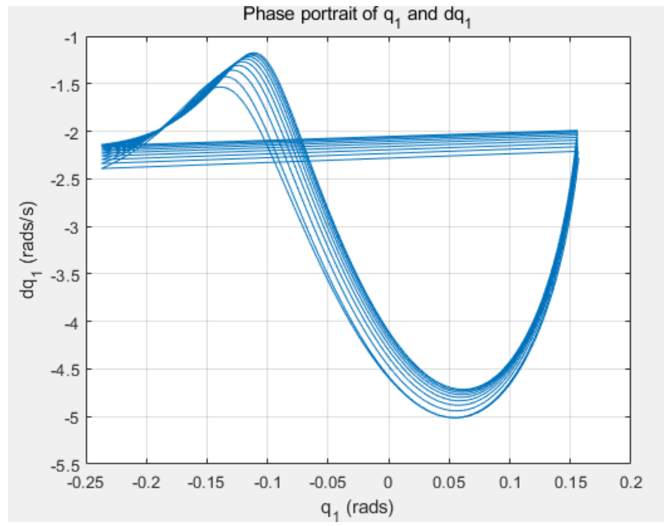

Fig. 6. Phase portrait of q1 vs dq1

- Developed and validated the kinematic models that provide a fundamental understanding of the robot's potential movements.
- Implemented dynamic modeling that captures the energies acting on the robot, allowing for the simulation of realistic gait dynamics.
- Applied zero dynamics design to optimize gait patterns that adhere to predefined dynamic constraints, significantly enhancing the efficiency and stability of the robot's locomotion.
- Engineered a control system that utilizes both PD controllers and feedback linearization to maintain the robot's trajectory firmly within the zero dynamics manifold, thereby ensuring stable and consistent walking patterns.

The simulations and optimizations performed have underscored the effectiveness of the control strategies, demonstrating their capability in maintaining stability even under the constraints of single-foot contact and instantaneous impacts. The success of these simulations suggests potential applications in more complex terrains and dynamic environments, where robust and adaptive control strategies are crucial.