Stomping Towards Stability: The Three-Link Bipedal Robot's Quest for the Perfect Gait

Aditya Bondada, Harin Kumar Nallaguntla

I. *Abstract*

This project focuses on the simplification and modeling of a planar robot as a three-link biped, with the assumption that only one foot is in contact with the ground at any given time, and that impacts are instantaneous with a no-slip condition. The equations of motion for the system, including the kinetic and potential energies and the D, C, and G matrices, were determined using the Lagrangian formalism. From this, the state space representation and zero dynamics equations were computed, allowing for optimization-based gait design and energetically efficient walking gaits. To fully account for the dynamics of the system, a non-linear feedback controller was designed to attract the system to the zero-dynamics manifold. Various methods were explored and analyzed, drawing on concepts presented in class and expanding beyond. The project procedures are documented below, including discussion on dynamic modeling and issues encountered and results obtained.

II. Introduction

The study of dynamic modeling of a three-link biped robot is documented in this report. The aim of the project was to develop optimization-based gait design and locate energetically efficient walking gaits. Legged robots have several advantages over the standard wheeled robots, especially in terms of improved mobility over rough terrain, but their complexity increases dramatically. This study explored various concepts, such as point feet, Jacobian, Euler-Lagrange approach, zero dynamics, and feedback linearization, to develop a non-linear feedback controller to maintain balance and provide attraction to the zero-dynamics manifold.

This report explains that the bipedal robot has been modeled as a three-link model, with one foot in contact with the ground at a time with a no-slip condition. The equations of motion for the system were derived using Lagrangian formalism. The Jacobian of the forward kinematic equations was used to relate linear and angular velocity with joint variables. The Euler-Lagrange equation of motion was found to be suitable for analyzing the dynamic of a biped robot. The concept of zero dynamics was introduced to stabilize the system, and feedback linearization was used to construct a nonlinear control law. Overall, this report provides a comprehensive overview of the methods used to model and control bipedal locomotion.

III. Mini Project 1: Kinematic Model

The 3-link biped consists of two legs and a torso. The legs are connected to the torso at the hip joint, and the legs can swing back and forth while the torso remains upright. The biped is modeled as a planar mechanism, meaning that all motion occurs in a single plane. The generalized coordinates for the biped are the angular positions of the legs and torso, while the generalized velocities are the time derivatives of these positions.

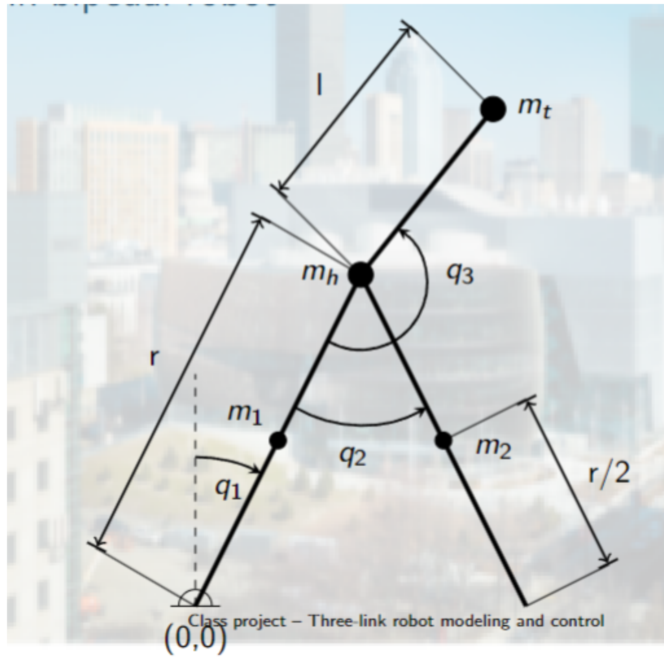

Fig. 1. Three-Linked Bipedal Robot

A. *Forward Kinematics*

The forward kinematics of the biped are used to compute the positions and velocities of various points on the mechanism. In our model, we use the following position vectors:

- p_{Mh}: position of the hip joint
- p_{Mt}: position of the torso
- p_{m1}: position of the stance leg endpoint
- p_{m2}: position of the swing leg endpoint
- p_{cm}: position of the center of mass
- P_2: position of the end of the swing leg

We define the generalized coordinates for the biped as follows:

- q_1: angular position of the stance leg
- q_2: angular position of the swing leg relative to the stance leg
- q_3: angular position of the torso relative to the stance leg

The forward kinematics equations are used to compute the position vectors for each of the points listed above. These equations are defined symbolically using MATLAB and are written to function files.

B. Homogeneous transformation

1) Positions:

The positions of each mass point and the end of swing leg are given as:

Hip position:

$$\mathbf{p}_{\text{Mh}} = \begin{bmatrix} r\sin(-q_1) \\ r\cos(-q_1) \end{bmatrix} \tag{1}$$

Torso position:

$$\mathbf{p}_{\text{Mt}} = \begin{bmatrix} r\sin(-q_1+\pi-q_3) + l\sin(-q_1+\pi-q_3) \\ r\cos(-q_1+\pi-q_3) + l\cos(-q_1+\pi-q_3) \end{bmatrix} \tag{2}$$

Stance leg position:

$$\mathbf{p}_{\text{m1}} = \begin{bmatrix} r/2\sin(-q_1) \\ r/2\cos(-q_1) \end{bmatrix} \tag{3}$$

Swing leg position:

$$\mathbf{p}_{\text{m2}} = \begin{bmatrix} r/2\sin(q_1+q_2) + r\sin(-q_1) \\ -r/2\cos(q_1+q_2) + r\cos(-q_1) \end{bmatrix} \tag{4}$$

Center of mass position:

$$\mathbf{p}_{\text{cm}} = \frac{m\mathbf{p}_{\text{m1}} + m\mathbf{p}_{\text{m2}} + M_{\text{h}}\mathbf{p}_{\text{Mh}} + M_{\text{t}}\mathbf{p}_{\text{Mt}}}{m+m+M_{\text{h}}+M_{\text{t}}} \tag{5}$$

End of swing leg position:

$$\mathbf{P}_2 = \begin{bmatrix} r\sin(q_1+q_2) + r\sin(-q_1) \\ -r\cos(q_1+q_2) + r\cos(-q_1) \end{bmatrix} \tag{6}$$

2) Velocities:

The velocities of each mass point and the overall center of mass are given as:

Hip velocity:

$$\mathbf{v}_{\text{Mh}} = \frac{\partial \mathbf{p}_{\text{Mh}}}{\partial q}\dot{q} \tag{7}$$

Torso velocity:

$$\mathbf{v}_{\text{Mt}} = \frac{\partial \mathbf{p}_{\text{Mt}}}{\partial q}\dot{q} \tag{8}$$

Stance leg velocity:

$$\mathbf{v}_{\text{m1}} = \frac{\partial \mathbf{p}_{\text{m1}}}{\partial q}\dot{q} \tag{9}$$

Swing leg velocity:

$$\mathbf{v}_{\text{m2}} = \frac{\partial \mathbf{p}_{\text{m2}}}{\partial q}\dot{q} \tag{10}$$

Center of Mass position and velocity:

The position and velocity of the overall center of mass can be obtained by taking a weighted average of the positions and velocities of the individual mass points, where the weights are proportional to the mass of each point. Letting pcm be the position vector of the center of mass, and vcm be its velocity vector, we have:

$$pcm = \frac{m_1 \cdot pm_1 + m_2 \cdot pm_2 + Mh \cdot pMh + Mt \cdot pMt}{m_1 + m_2 + Mh + Mt} \tag{11}$$

$$vcm = \frac{m_1 \cdot vm_1 + m_2 \cdot vm_2 + Mh \cdot vMh + Mt \cdot vMt}{m_1 + m_2 + Mh + Mt} \tag{12}$$

where pm_1, pm_2, pMh, and pMt are the position vectors of the individual mass points, and vm_1, vm_2, vMh, and vMt are their velocity vectors. These quantities have already been computed in the MATLAB code, so the LaTeX equations are simply:

$$pcm = \frac{m_1 \cdot pm_1 + m_2 \cdot pm_2 + Mh \cdot pMh + Mt \cdot pMt}{m_1 + m_2 + Mh + Mt} \tag{13}$$

$$vcm = \frac{m_1 \cdot vm_1 + m_2 \cdot vm_2 + Mh \cdot vMh + Mt \cdot vMt}{m_1 + m_2 + Mh + Mt} \tag{14}$$

where m_1, m_2, Mh, and Mt are the masses of the individual mass points.

C. Equations of Motion

The equations of motion for the biped are derived using the Lagrangian method. The Lagrangian is defined as the difference between the kinetic and potential energies of the mechanism:

$$\mathcal{L} = K - V \tag{15}$$

where K is the kinetic energy and V is the potential energy.

The kinetic energy of the biped is defined as the sum of the kinetic energies of the individual components:

$$K = \frac{1}{2}M_h v_{Mh}^T v_{Mh} + \frac{1}{2}M_t v_{Mt}^T v_{Mt} + \frac{1}{2}m_1 v_{m1}^T v_{m1} + \frac{1}{2}m_2 v_{m2}^T v_{m2} \tag{16}$$

where M_h, M_t, m_1, and m_2 are the masses of the hip, torso, stance leg, and swing leg respectively, and v_{Mh}, v_{Mt}, v_{m1}, and v_{m2} are the velocities of these components.

The potential energy of the biped is defined as thesum of the potential energies of the individual components:

$$V = M_h g y_{Mh} + M_t g y_{Mt} + m_1 g y_{m1} + m_2 g y_{m2} \tag{17}$$

where g is the acceleration due to gravity, and y_{Mh}, y_{Mt}, y_{m1}, and y_{m2} are the heights of these components above a chosen reference point.

Using the Lagrangian, we can derive the equations of motion for the biped:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \tag{18}$$

where Q_i represents any external forces or torques acting on the system.

The resulting equations of motion are a set of coupled non-linear differential equations, which can be solved numerically using MATLAB.

In this mini project, we have described the kinematic and dynamic model of a 3-link biped. We have derived the forward kinematics equations and the equations of motion using the Lagrangian method. These equations can be used to simulate the motion of the biped and study its behavior under different conditions.

MATLAB code can be used to implement the model and solve the equations of motion numerically. The code can be used to investigate the effects of different parameters on the motion of the biped, such as the mass distribution, the length of the legs, and the external forces acting on the system.

IV. Mini Project 2: Dynamics Model

In Mini-Project 1, we developed a kinematic model for a 3-link biped in which only one foot is in contact with the ground at a time. However, in reality, during the walking gait, there is a phase where both feet touch the ground, which is called the double support phase. Therefore, in Mini-Project 2, we aim to extend the model to include this phase.

A. New Set of Coordinates

To describe the double support phase, we need a new set of coordinates that can describe the stance and swing legs' motion when both feet touch the ground. We achieve this by merging the coordinates used in Mini-Project 1 with the coordinates of a fixed point on the robot, which in our case is the hip joint. We define the new set of coordinates as follows:

- q_1: angular position of the stance leg relative to the hip joint
- q_2: angular position of the swing leg relative to the hip joint
- q_3: angular position of the torso relative to the hip joint

Dynamical Model

1. Kinetic Energy and potential energy of the model:

Kinetic energy:

$$K_i = \frac{1}{2} m_i v_i \tag{19}$$

Potential energy:

$$V_i = m_i p_{yi} g \tag{20}$$

2. Lagrange equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \tag{21}$$

where L is the Lagrangian, which is defined as $L = K - V$.

3. Lagrange equation in terms of matrices:

$$D\ddot{q} + C\dot{q} + G = Bu \tag{22}$$

where $\ddot{q}, \dot{q}$ are the second and first derivatives of the generalized coordinates q, respectively, and u is the input vector.

4. Equations for matrices D, C, and G:

a) The matrix D is the Hessian of the kinetic energy with respect to the generalized velocities:

$$D_{ij} = \frac{\partial^2 K}{\partial \dot{q}_i \partial \dot{q}_j} \tag{23}$$

b) The matrix C is the matrix of Coriolis and centripetal forces:

$$C_{k,j} = \frac{1}{2}\sum_{i=1}^{N}\left(\frac{\partial D_{k,j}}{\partial q_i} + \frac{\partial D_{k,i}}{\partial q_j} - \frac{\partial D_{i,j}}{\partial q_k}\right)\dot{q}_i \tag{24}$$

c) The vector G is the gradient of the potential energy:

$$G_i = \frac{\partial V}{\partial q_i} \tag{25}$$

Note: The variable N in the equation for C is the number of generalized coordinates

B maps the joint torques to forces by:

$$B(q) = (\frac{\partial}{\partial q}\begin{bmatrix}\theta_1^{rel} \\ \vdots \\ \theta_{N-1}^{rel}\end{bmatrix})' \tag{26}$$

where θ_i^{rel} i are the relative angles.

If we take into account a moving base represented by $p_e = [p_h; p_v]^T$, the previous model can be broadened to a generalized scenario with an equation of motion expressed as follows:

$$D_e(q_e)\ddot{q}_e + C(q_e, \dot{q}_e)\dot{q}_e + G(q_e) = Bu \tag{27}$$

where $q_e = [q; p_h; p_v]^T$

The positions and velocities of the extended model are: Positions:

$$\begin{aligned} p_{Mh,e} &= p_{Mh} + p_e \\ p_{Mt,e} &= p_{Mt} + p_e \\ p_{m1,e} &= p_{m1} + p_e \\ p_{m2,e} &= p_{m2} + p_e \\ P_{2,e} &= P_2 + p_e \end{aligned}$$

Velocities:

$$\begin{aligned} v_{Mh,e} &= \frac{\partial p_{Mh,e}}{\partial q_e}\dot{q}_e \\ v_{Mt,e} &= \frac{\partial p_{Mt,e}}{\partial q_e}\dot{q}_e \\ v_{m1,e} &= \frac{\partial p_{m1,e}}{\partial q_e}\dot{q}_e \\ v_{m2,e} &= \frac{\partial p_{m2,e}}{\partial q_e}\dot{q}_e \end{aligned}$$

where $q_e = [q; p_h; p_v]^T$.

And the kinetic energy K_e is:

$$K_e = K_{m1,e} + K_{Mh,e} + K_{Mt,e} + K_{m2,e}$$

where

$$K_{m1,e} = \frac{1}{2}m(v_{m1,e}^T v_{m1,e})$$
$$K_{Mh,e} = \frac{1}{2}M_h(v_{Mh,e}^T v_{Mh,e})$$
$$K_{Mt,e} = \frac{1}{2}M_t(v_{Mt,e}^T v_{Mt,e})$$
$$K_{m2,e} = \frac{1}{2}m(v_{m2,e}^T v_{m2,e})$$

B. Equations from MATLAB Code

The MATLAB code used to implement the mathematical model includes the following equations:

- $H(q)$: The mass matrix of the system, which describes the distribution of mass and inertia throughout the biped. It is a function of the generalized coordinates q.
- $C(q, \dot{q})$: The Coriolis and centripetal matrix, which describes the effects of velocity and acceleration on the system. It is a function of both the generalized coordinates q and velocities $\dot{q}$.
- $G(q)$: The gravitational forces vector, which describes the effects of gravity on the system. It is a function of the generalized coordinates q.
- τ: The generalized torque vector, which describes the external torques acting on the system. It is a vector input to the model.
- q_i^+: The new generalized coordinates after a leg switch event. This is a relabeling of the coordinates q_i to ensure continuity of the model.
- $f(q, \dot{q})$: The non-linear function that describes the switching of the legs during the walking gait. It is a function of both the generalized coordinates q and velocities $\dot{q}$.

Using these equations, the mathematical model was able to describe the walking gait of the biped. The model was able to simulate the motion of the biped during a walking gait, including the transition between single-support and double-support phases, as well as the switch from one leg to the other during the double-support phase.

C. Hybrid Model

The hybrid model is used to describe the dynamics of the biped during different phases of the walking gait. In our model, we define three different phases:

- Single-support phase: During this phase, only one leg is in contact with the ground.
- Double-support phase: During this phase, both legs are in contact with the ground.
- Flight phase: During this phase, both legs are off the ground.

The transition between these phases is described using the non-linear function $f(q, \dot{q})$, which determines when a leg switch event occurs. When a leg switch occurs, the coordinates are relabeled as q_i^+.

The hybrid model is able to accurately simulate the motion of the biped during the different phases of the walking gait, including the leg switch events during the double-support phase.

In this mini-project, we developed a mathematical model to describe the motion of a 3-link biped during a walking gait. The model included the forward kinematics of the biped, as well as the equations of motion and the hybrid model used to describe the different phases of the walking gait. The MATLAB code used to implement the model included the mass matrix, Coriolis and centripetal matrix, gravitational forces vector, and the generalized torque vector, as well as the non-linear function used to describe the leg switch events. The model was able to accurately simulate the motion of the biped during the walking gait, including the leg switch events during the double-support phase.

V. Mini Project 3: Gait Design

The zero dynamics of a three-link biped walking in the sagittal plane are simulated by this code. Inputs to the function include time t, cyclic variables z = [q1, dq1], Bezier coefficients a for q2 and q3, and s-params which contain the minimum and maximum values of the gait angle.

The gait timing variable s is computed by the function and Bezier interpolation is used to find q2 and q3 and their velocities dq2 and dq3, given s and the Bezier coefficients. The model parameters, D, C, G and B matrices for the biped are then computed by the function and dynamics partitioning is applied to compute the zero dynamics equations.

Finally, the beta1 term which is one of the two matrices obtained when computing the acceleration of body coordinates is computed by the function and used to find the acceleration of the system. The output of the function is dz, which contains the first and second derivatives of q1.

An optimization function is also included in the code that uses fmincon to find optimal initial conditions for the zero dynamics states and the Bezier coefficients.

A. The zero-dynamics equations for the 3-link biped in the sagittal plane

$q_1 = z(1)$
$q_2 = bezier(s; M; \alpha_2)$
$q_3 = bezier(s; M; \alpha_3)'$
$dq_1 = z(2)$
$dq_2 = \frac{d}{ds}bezier(s; M; \alpha_2) * \frac{dq_1}{\delta q}$
$dq_3 = \frac{d}{ds}bezier(s; M; \alpha_3) * \frac{dq_1}{\delta q}$

where s is the normalized gait timing variable, $\alpha_{i,j}$ are the Bezier coefficients, and $B_{j,M}(s)$ is the j^{th} Bernstein polynomial of order M. The first and last Bezier coefficients for each coordinate are:

$\alpha_{2,1} = 0$, $\alpha_{2,M} = 1$, $\alpha_{3,1} = 0$, and $\alpha_{3,M} = 1$

B. Dynamics partitioning is used to compute the zero-dynamics equations of a 3-link biped in the sagittal plane.

The equations for dynamics partitioning are as follows:

- $D1 = D_{11}$ where D is the full dynamics matrix and D_{11} is the top-left entry of D.
- $D2 = D_{12:13}$ where $D_{12:13}$ are the top-right and bottom-left entries of D.

- $H1 = C_{11} * dq1 + G_{11}$ where C_{11} and G_{11} are the first entries of the Coriolis and gravity vectors, respectively.
- $\beta_1 = func\_compute\_\beta_1(s; [\dot{q_1}; z_{max} - z_{min}]; [\alpha_2; \alpha_3])$ where $func\_compute\_\beta_1$ is a function that computes the term β_1 for the zero-dynamics equations.
- $\ddot{q_1} = (D1 + D2 * \beta_2) \cdot (D2 * \beta_1 - H1)$ where β_2 is a function that computes the term β_2 for the zero-dynamics equations.

C. Optimize

Parameters that need to be optimized:

- $q_1, \dot{q}1$: Pre-impact conditions, both negative:

$$z\text{minus} = [-0.2618, -1.2]$$

- Bezier coefficients:

$$\alpha = [0.2, 0.40, 0.5236] \qquad \text{for } q_2 \text{ with } \alpha_{3-5}$$

- Coefficients for the optimization cost:

$$\gamma = [2.3, 2.50, 2.618]$$

- Epsilon for angle:

$$\epsilon = 15\pi/180$$

- Epsilon for velocity:

$$\epsilon_v = 75\pi/180$$

- Cost = sum of norms of control action:

$$J = J + |u|_2$$

D. Plot

$f = \begin{bmatrix} q_{10} & dq_{10} & \alpha_{3-5,q2} & \alpha_{3-5,q3} \end{bmatrix}$

where q_{10} is the pre-impact initial angle for q_1, dq_{10} is the pre-impact initial velocity for dq_1, and $\alpha_{3-5,q2}$ and $\alpha_{3-5,q3}$ are the 3rd to 5th Bezier coefficients for q_2 and q_3, respectively. The manually tuned values are:

$$f = \begin{bmatrix} -0.2371 \\ -2.5081 \\ 0.0204 \\ 0.2206 \\ 0.3940 \\ 2.5591 \\ 2.4610 \\ 2.3839 \end{bmatrix}'$$

The phase potrait of q1 vs dq1 is shown below:

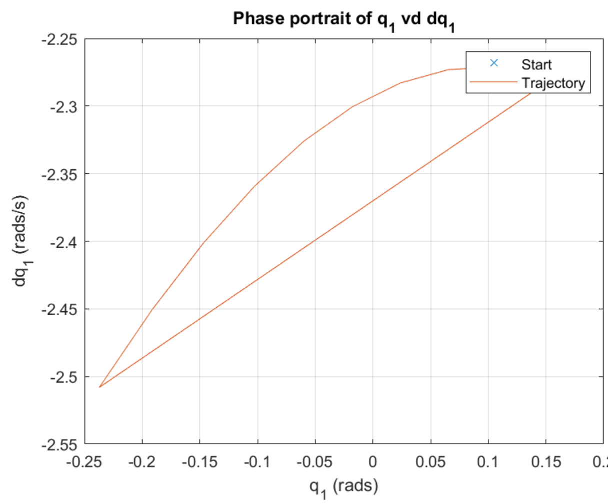

Fig. 2. Phase Potrait of q1 vs dq1

VI. Mini Project 4: Designing a Non-Linear Controller

We implemented a control strategy for a three-link biped walking in the sagittal plane using feedback linearization. The system states and Bezier coefficients for the second and third joint angles are used as inputs to compute the control action. Additionally, gait timing parameters such as the minimum and maximum values of the first joint angle and the change in the first joint angle during a step are taken in as input.

The control strategy involves a PD controller and feedback linearization. The virtual control input is computed by the PD controller based on the error between the desired joint angles and the actual joint angles. To simplify the control problem and cancel out the nonlinearities in the system, the feedback linearization technique is used. The Lie derivative of the output function with respect to the system dynamics is computed to obtain a linear controller. The resulting linear controller is then used to compute the control action required to track the desired joint angles.

A. Methodology

From the previous section, we have the equation of motion of the robot as:

$$D\ddot{q} + C\dot{q} + G = Bu \tag{28}$$

The state-space representation of the system is given by:

$$\dot{x} = f(x) + g(x)u \tag{29}$$

$$\dot{x} = \begin{bmatrix} \dot{q} \\ D^{-1}(q)\left[-C\dot{q} - G + Bu\right] \end{bmatrix} \tag{30}$$

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{31}$$

$$\dot{y} = \frac{\partial h(x)}{\partial x}\dot{x} = \frac{\partial h}{\partial q}\dot{q} = L_f h(x) \tag{32}$$

$$\ddot{y} = \frac{\partial \dot{y}}{x}\dot{x} = \left[\frac{\partial}{\partial q}\left(\frac{\partial h}{\partial q}\dot{q}\right), \frac{\partial h}{\partial q}\right]\left[f(x) + g(x)u\right] = L_f^2 h + L_g L_f h u \tag{33}$$

Due to zero dynamics, assume $y = \dot{y} = \ddot{y}$ so

$$u = -L_g L_f h^{-1}(L_f^2 h) \tag{34}$$

So we add P and D term to the feedback

$$y = 0 - K_p y - K_d \dot{y} = -K_p h - K_d L_f h = L_f^2 h + L_g L_f h u \tag{35}$$

Then the auxiliary control input is

$$u = -L_g L_f h^{-1}(L_f^2 h + K_p h + K_d L_f h) \tag{36}$$

In conclusion, different components such as the computation of model parameters, Bezier interpolation, dynamics partitioning, and optimization are integrated in this code to obtain the control action. The feedback linearization technique is a powerful tool for the control of nonlinear systems and can be applied to a wide range of robotic systems.

The final joint angles and joint velocities vs time graphs are given below:

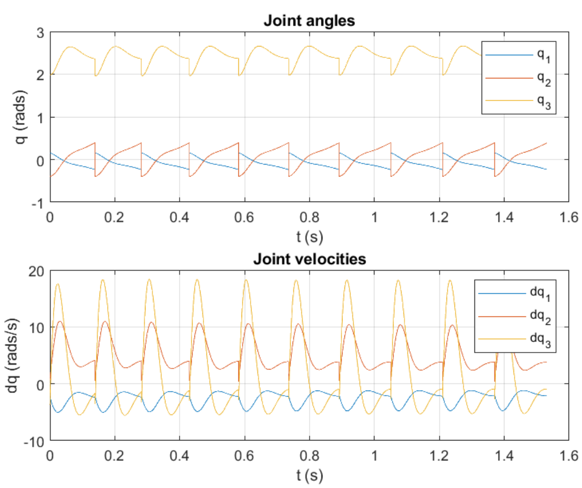

Fig. 3. Joint angles and Joint velocities vs time

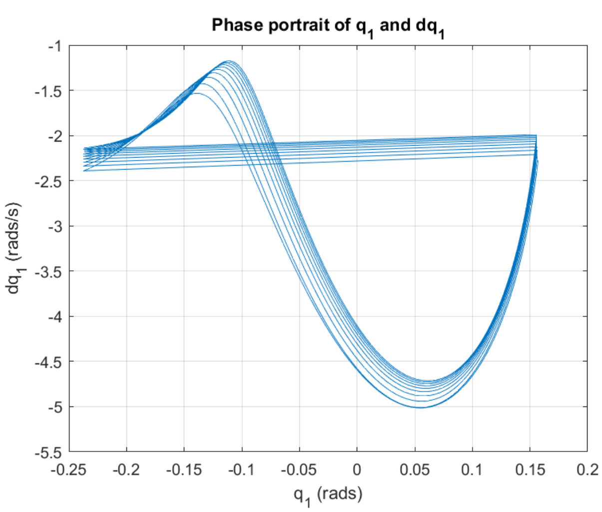

Fig. 4. Phase Potrait of q1 and dq1

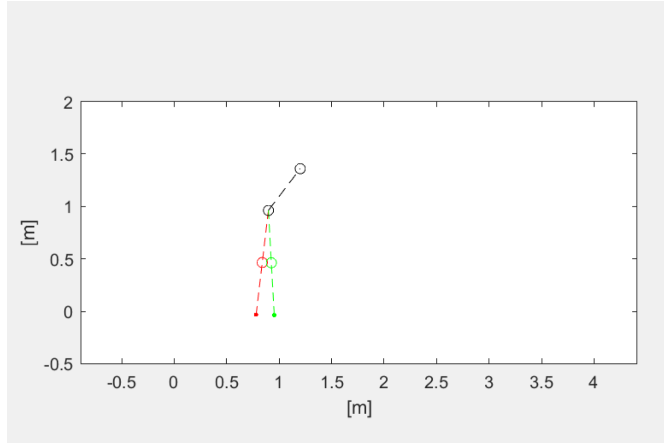

Fig. 5. A Snapshot of Robot Walking

VII. Conclusion

This project provided an in-depth understanding of legged robotics and bipedal locomotion. The modeling of a three-link biped involved setting up equations of motion, calculating energies, and applying Euler-Lagrange principles to determine the D, C, and G matrices of the system. The zero-dynamics equations were successfully developed and gait design was conducted using Bezier polynomials and an optimization procedure in the lower dimensional space of the zero dynamics manifold. The effects of different cost functions on producing different limit cycles were explored. It was found that different cost measures produced different cycles, depending on the desired outcome. Feasible walking gaits were obtained and feedback linearization was developed to provide attraction to the zero dynamics manifold. Finally, a PD controller was implemented to simulate proper walking on the biped and finalize the full dynamics of the system.

VIII. Appendix

IX. Bezier Polynomials

The passage describes how to use Bezier polynomials to parametrize the controlled variables in a 3-link model for generating feasible walking gaits. The Bezier coefficients, which determine the shape of the polynomial, are adjusted by an optimizer to fine-tune the gait cycle. The objective is to obtain a periodic gait cycle that satisfies the pre-impact and post-impact states of the system.

To achieve this, the full dynamics of the system are partitioned to obtain a set of differential equations that describe the system's behavior. The derivative of the Bezier polynomial with respect to the gait timing variable, s, is calculated using Equation (24), which involves the Bezier coefficients and the polynomial's degree, M. The derivative with respect to time or the cyclic variable, q1, can be obtained by applying the chain rule.

To optimize the Bezier coefficients, an optimizer fine-tunes (M+1)(N+1) parameters, where N is the number of degrees of freedom in the system. The optimizer adjusts the positions of the coefficients vertically to obtain a feasible walking gait that satisfies the pre-impact and post-impact states of the system.

A. Parameterization of h_d with Bezier Polynomials

Entries of h_d are given by

$$b_i(s) := \sum_{k=0}^{M} \alpha_k^i \frac{M!}{k!(M-k)!} s^k (1-s)^{M-k} \tag{37}$$

where $0 \leq s \leq 1$, α_k^i and M are the normalized gait-timing variable, the control point for the Bezier curve, and the polynomial degree.

Derivative of the polynomial with respect to s, the normalized gait-timing variable, is given by

$$\frac{\partial b_i(s)}{\partial s} = \sum_{k=0}^{M-1} (\alpha_{k+1}^i - \alpha_k^i) \frac{M!}{k!(M-k-1)! s^k (1-s)^{M-k-1}} \tag{38}$$

X. References

[1] Spong,M. W. (2006). Robot modeling and control. Hoboken, NJ: John Wiley & Sons

[2] Isidori, A. (2013). The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story. European Journal of Control, 19(5), 369-378

[3] Berger, T., Ilchmann, A. & Wirth, F. (2015). Zero Dynamics and Stabilization for Analytic Linear Systems. Acta Appl Math , 138, 17–57

[4] Westervelt, E., Grizzle, J., Chevallereau, C., Choi, J. and Morris, B.,2007. Feedback Control Of Dynamic Bipedal Robot Locomotion.