Modeling and Simulation of a Planar, Bipedal Walker Using Hybrid Zero Dynamics Control

EECE 7398: Legged Robotics, Professor Alireza Ramezani

Chenghao Wang
Dept. Electrical and Computer Eng.
Northeastern University
wang.chengh@northeastern.edu

Noah Ossanna
Dept. Mechanical Eng.
Northeastern University
ossanna.n@northeastern.edu

Dev Vaibhav
Dept. Electrical and Computer Eng.
Northeastern University
vaibhav.d@northeastern.edu

Siddharth Maheshwari
Dept. Electrical and Computer Eng.
Northeastern University
maheshwari.si@northeastern.edu

Michael Tang
Dept. Mechanical Eng.
Northeastern University
tang.mich@northeastern.edu

Kshama Dhaduti
Dept. Electrical and Computer Eng.
Northeastern University
dhaduti.k@northeastern.edu

I. Abstract

This report details the approach to simulating the gait and control of a planar, three-link bipedal robot utilizing a Hybrid Zero Dynamics (HZD) method. In order to simplify interactions, the group made the assumptions that one foot remains in contact with the ground at a time with a no slip condition, and that impacts were considered instantaneous. The procedure to achieve a stable, realistic gait includes generating the forward kinematics of the biped, Euler-Lagrange formulas, dynamic model, zero dynamics, optimization, and controller design. Progressively stepping through the design process allowed the group generate an efficient gait using a PD controller in feedback. The report is divided into mini projects that encapsulate the key milestones used to generate the final gait for the bipedal robot.

II. Mini-project 1:

In order to understand the behavior of the three-link biped, we first uncovered the underlying interactions between joint angles and positions of designated features. The mathematical relations between the joint angles and positions are accounted for through the formulation of the forward kinematics. It was important for the design that directions, relations, and naming conventions remained consistent with those established here in order to generate the correct corrective actions and gait.

A. Forward Kinematics Derivation

Forward Kinematics allow the derivation of point mass positions and velocities based on the relationship between joint angles and physical model constraints. Point masses specifically refer to the assumed mass distributions of the robot's legs which should be noted for simplicity of calculation. Relations were derived through homogeneous transformations that represent the rotation and translation between coordinate frames. The contact point between the stance leg and the ground would was assumed to be the (0,0) of the world coordinate frame. A figure of the given joint angles, point masses, and lengths are shown in Fig 1.

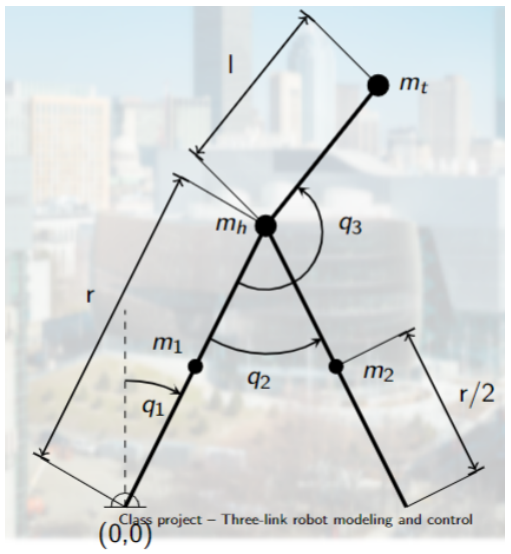


Fig. 1: Given three-link 2D bipedal robot model

The generated transformation matrices were created based on arbitrary, but consistent conventions for coordinate frame placement. The positive direction for rotation was assumed to be counter clockwise, and positive z-direction is out of the plane of paper towards us. A diagram of the local coordinate frames is shown below:

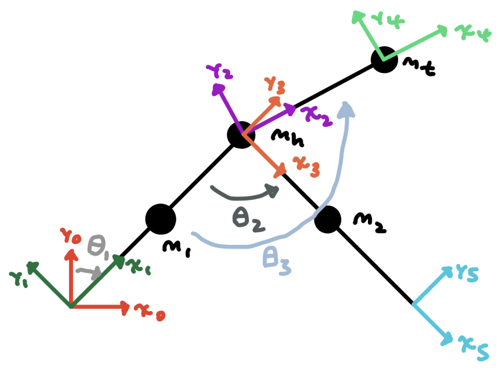


Fig. 2: Local coordinate frames placement on the three-link model

Combinations of these transformations allowed the group to calculate the position and velocity of desired positions based on joint angles and given physical parameters. To find the position of the desired points with respect to the origin, calculations were broken into 2 steps. The first step was finding the homogeneous transformation matrices between the different coordinate frames and then by multiplying these transformation matrices with the position of the desired points defined in their local coordinate frames.

$$T_1^0 = \begin{bmatrix} -\sin(q_1) & -\cos(q_1) & 0 & 0 \\ \cos(q_1) & -\sin(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} -\cos(q_3) & \sin(q_3) & 0 & r \\ -\sin(q_3) & -\cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} \cos(q_2 - q_3) & -\sin(q_2 - q_3) & 0 & 0 \\ \sin(q_2 - q_3) & \cos(q_2 - q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^2 = \begin{bmatrix} 1 & 0 & 0 & l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^3 = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The position vectors for the desired point masses: m_1, m_2, m_t, m_h are shown below

$$pos_{m_1}^1 = \begin{bmatrix} r/2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$pos_{m_h}^1 = \begin{bmatrix} r \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$pos_{m_t}^2 = \begin{bmatrix} l \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$pos_{m_2}^3 = \begin{bmatrix} r/2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The final position of these points with respect to the origin were found by multiplying combinations of these transformation matrices with the position vectors, which is shown below.

$$P_{m_1}^0 = T_1^0 \times pos_{m_1}^1 \quad (1)$$
$$P_{m_2}^0 = T_1^0 \times T_2^1 \times T_3^2 \times pos_{m_2}^3 \quad (2)$$
$$P_{m_h}^0 = T_1^0 \times pos_{m_h}^1 \quad (3)$$
$$P_{m_t}^0 = T_1^0 \times T_2^1 \times pos_{m_t}^2 \quad (4)$$

The final results of these positions are given as:

$$P_{m_1}^0 = \begin{bmatrix} -\frac{r*\sin(q_1)}{2} \\ \frac{r*\cos(q_1)}{2} \end{bmatrix}$$

$$P_{m_2}^0 = \begin{bmatrix} \frac{r*(\sin(q_1+q_2)-2*\sin(q_1))}{2} \\ -\frac{r*(\cos(q_1+q_2)-2*\cos(q_1))}{2} \end{bmatrix}$$

$$P_{m_h}^0 = \begin{bmatrix} -r*\sin(q_1) \\ r*\cos(q_1) \end{bmatrix}$$

$$P_{m_t}^0 = \begin{bmatrix} l*\sin(q_1+q_3) - r*\sin(q_1) \\ r*\cos(q_1) - l*\cos(q_1+q_3) \end{bmatrix}$$

III. Mini-project 2:

To correlate positions of interest on the robot to desired trajectories, we began by applying the Lagrange formalism. This formulation resolves the system's dynamics as a function of kinetic and potential energies as well as a set of generalized coordinates: $(q, \dot{q})$. These represent the states of the joint variables defined previously alongside their derivatives. The objective was to derive dynamical equations of the following forms:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \Gamma \tag{5}$$

$$\dot{x} = f(x) + g(x)u \tag{6}$$

where equation 5 is the set of second order equations defined by the Lagrange formalism and equation 6 is the nonlinear state variable representation of the system. Derivations of the inertia, coriolis, and gravity matrices D, C, and G are explained in subsequent sections.

A. Euler-Lagrange Formalism: Support Phase

The Lagrangian was constructed as the following function of kinetic and potential energies:

$$\mathcal{L}(q, \dot{q}) = K(q, \dot{q}) - V(q) \tag{7}$$

where $\mathcal{L}$ is the Lagrangian function of the generalized coordinates $(q, \dot{q})$, K is the kinetic energy term, and V is the potential energy term. In this case, we consider the generalized cordinates and their derivatives to represent the 3 joint variables of the planar walker: q_1, q_2, q_3 which form the basis of Lagrange's equation:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \Gamma \tag{8}$$

where equation 8 results in equation 5 above following subsequent derivations. In these equations, Γ represents a vector of generalized forces and torques. To derive the state space representation for the model from these relations, the total kinetic and potential energies of the robots links were resolved by the equations below.

$$K_i = \frac{1}{2}m_i((p^{\dot{h}}_{cm,i})^2 + (p^{\dot{v}}_{cm,i})^2) + \frac{1}{2}J_{cm,i}(\dot{\theta_i^{abs}})^2 \tag{9}$$

$$K(q, \dot{q}) = \sum_{i=1}^{N} K_i(q, \dot{q}) = \frac{1}{2}\dot{q}^T D \dot{q} \tag{10}$$

$$V(q) = \sum_{i=1}^{N} V_i(q) = \sum_{i=1}^{N} g_0 m_i p^v_{cm,i}(q) \tag{11}$$

where g_0 is the acceleration due to gravity, m_i is the point mass of the i-th link, $p^{h,v}_{cm,i}$ are the horizontal and vertical positions of that point mass, $J_{cm,i}$ is the moment of inertia of the mass of link-i, and $\dot{\theta_i^{abs}}$ is the absolute angular velocity of that link. The inertia matrix D is also derived as the following:

$$D(q) = \frac{\partial}{\partial \dot{q}}\left(\frac{\partial K}{\partial \dot{q}}\right) \tag{12}$$

and in its extended matrix form as:

$$D_e = \begin{bmatrix} D & 0 \\ 0 & m_{tot} * I \end{bmatrix} \tag{13}$$

where $q = (q_1, q_2, q_3)$ is the configuration vector/ variable consisting of joint angles, m_{tot} is the sum of all masses of the links (resolved to point masses by assumption) and D_e is the inertia matrix for the flight phase dynamics. These are calculated alongside the ongoing swing phase dynamics derived in this section. Next, the terms of the coriolis matrix were defined iteratively through summation:

$$C(q, \dot{q})_{kj} = \sum_{i=1}^{N} \frac{1}{2}\left(\frac{\partial D_{kj}}{\partial q_i} + \frac{\partial D_{ki}}{\partial q_j} - \frac{\partial D_{ij}}{\partial q_k}\right)\dot{q}_i \tag{14}$$

where the terms in the summation are Christoffel symbols that denote the centrifugal $(i = j)$ and coriolis $(i \neq j)$ components of the equation of motion. Finally, we define the gravity matrix G and input mapping matrix B as follows:

$$G(q) = \frac{\partial V}{\partial q} \tag{15}$$

$$B(q) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{16}$$

Equation 16 is defined based on the fact that q_1 is under actuated while we have two actuated control inputs: q_2 and q_3. The B matrix defined in equation 16 was used to map actuated, control inputs to the generalized vector of forces and torques, Γ:

$$\Gamma = B * u \tag{17}$$

where the vector of actuated inputs u_2, u_3 (corresponding to the inputs of joints q_2, q_3) was defined as

$$u = \begin{bmatrix} u_2 & u_3 \end{bmatrix}^T \tag{18}$$

From these derivations for the support phase of the robots gait, we define the state variable vector x:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{19}$$

which consists of a vector containing the joint angles and their derivative with respect to time for the planar walker. This nomenclature allows for the derivation of the state space form by rearranging the equations:

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix} x_2 \\ -D^{-1}(x_1)(C(x_1,x_2)x_2 + G(x_1)\end{bmatrix} \quad (20)$$

Splitting this equation into the terms defined by equation 6 yield the below expressions that could be implemented pragmatically:

$$f(x) = \begin{bmatrix} dq \\ -D(q)^{-1}(C(q,\dot{q})dq + G(q))\end{bmatrix}_{6\times 1} \quad (21)$$

$$g(x) = \begin{bmatrix} zeros(3,2) \\ D^{-1}B\end{bmatrix}_{6\times 2} \quad (22)$$

Equations 21 and 22 have 6x1 and 6x2 dimensionality respectively due to the planar walker having 3 joint configuration variables. Next, the above equations were adapted to the extended phase of the walker's gait to begin formulating a switching model to transition from swing phase to impact dynamics and vice versa.

B. *Swing Phase and Impact Mapping:*

The goal of this approach was to develop a model in which the first part of the gait is described by the single support dynamics. Then, when impact occurs, a switching condition is applied and the swing (extended state) model is applied using an impact model. For this reason, this section explores the derivation of an impact model that represents the instantaneous point in which both the stance and swing leg are in contact with the ground. The following set of coordinates are realized to begin representing this state:

$$q_e = \begin{bmatrix} q \\ p_e\end{bmatrix}_{3\times 1} = \begin{bmatrix} q \\ p^h_{cm,e} \\ p^v_{cm,e}\end{bmatrix}_{3\times 1} \quad (23)$$

where $p^{h,v}_{cm,e}$ are the horizontal and vertical positions of the robots center of mass in the extended state right before impact. Applying the Lagrangian to this state fields the relation

$$D_e(q_e)\ddot{q_e} = C_e(q_e,\dot{q_e})\dot{q_e} + G_e(q_e) = B_e(q_e)u + \delta F_{ext} \quad (24)$$

where the inertia matrix for the extended state is defined above in equation 13 and the other matrices are derived identically using the new extended variables. δF_{ext} represents a vector of external forces due to contact between the swing leg and ground surface. Equation 24 can be integrated over the infinitesimal contact time of the impact to give the force:

$$F_{ext} = D_e(q_e{}^+)\dot{q_e}{}^+ - D_e(q_e{}^-)\dot{q_e}{}^- \quad (25)$$

where $\dot{q_e}{}^{-/+}$ are the joint velocities before or after impact. Crucially, $\dot{q_e}{}^-$ is defined by the single support phase model which allows the principle of virtual work to be applied as:

$$F_{ext} = E_2(q_e^-)^T F_2 \quad (26)$$

where $E_2(q_e) = \frac{\partial p_2 q_e}{\partial q_e}$ and F_2 is the vector of forces acting at the swing leg. Then, by assuming that the swing leg does not slip or bounce when it makes an impact with the ground, the set of equations can be established below:

$$\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^1)^T \\ E_2(q_e^1) & 0_{2\times 2}\end{bmatrix}\begin{bmatrix}\dot{q_e}{}^+ \\ F_2\end{bmatrix} = \begin{bmatrix} D_e(q_e^-)\dot{q_e}{}^- \\ 0_{2\times 1}\end{bmatrix} \quad (27)$$

which can be solved for $\dot{q_e}{}^+$ and F_2 according to [1]. In MATLAB, we relabel the impact coordinates to achieve the relation $x^+ = \Delta(x^-)$ which represents the post- and pre-impact states respectively. After simplification, we combined supported and swing phase dynamics represented by the following hybrid model as defined in [1]:

$$\begin{cases} \dot{x} = f_s(x) + g_s(x)u & x^- \notin S \\ x^+ = \Delta x^- & x^- \in S \end{cases} \quad (28)$$

where S is the switching set defined in [1]. The completion of the switching model allowed for the gait design described in subsequent sections.

IV. Mini-project 3: Gait Design

A. *Zero Dynamics: Bezier Polynomial*

The gait design procedure was done by using an optimizer and reducing the dimensionality of the system to a projection onto the zero dynamics manifold. Zero dynamics equations are the internal dynamics of the system that is compatible with the output being identically zero. Since the dimension of the zero dynamics is less than the dimension of the model itself, this optimization approach is efficient in computation due to its selection requirement from fewer parameters. This approximation method retains all essential features of the bipedal model without sacrificing the accuracy of the simulation results.

The zero dynamics manifold is defined by z_1 and z_2, where z_1 is the cyclic variable (q_N or q_1) and the relation below.

$$z_2 = \dot{z_1} = \dot{q_N} = \dot{q_1}$$

Next, we parameterized the body angles ($q_b = [q_2, q_3]^T$) in terms of the cyclic variable (q_N) using a Bezier polynomial of degree $M = 4$. The Bezier polynomial ($b_i(s)$) takes an input s which, in the case of this simulation, is referred to as the gait timing variable where $s = 0$ is the start of gait and $s = 1$ is the end of gait. This variable is defined such that $0 \leq s \leq 1$ and produces a range R of degree M. We consider Bezier coefficients (α_k^i) that are M+1 in number and controlled variables q_2 and q_3 which are the body angles of the planar walker to be parameterized by the polynomial. The Bezier coefficients are equally spaced along the Bezier curve with subscript i referring to the index of the controlled variable ($i = 2$ for q_2 and $i = 3$ for q_3). The gait timing variable is defined as

$$s = \frac{q_N - q_N^+}{q_N^- - q_N^+} \tag{29}$$

where q_N^- is the pre-impact value of cyclic variable just before touchdown of stance leg and q_N^+ is the post-impact value of cyclic variable immediately after touchdown of stance leg. From these definitions, the Bezier polynomial and its derivative are constructed from the following relations:

$$b_i(s) = \sum_{k=0}^{M} \alpha_k^i \frac{M!}{k!(M-k)!} s^k (1-s)^{(M-k)} \tag{30}$$

$$\frac{\partial b_i(s)}{\partial s} = \sum_{k=0}^{M-1} (\alpha_{k+1}^i - \alpha_k^i) \frac{M!}{k!(M-k-1)!} s^k (1-s)^{(M-k-1)} \tag{31}$$

To partition the full dynamics, the time derivative of the bezier polynomial is determined by applying the chain rule as follows:

$$\frac{db_i(s)}{dt} = \frac{\partial b_i(s)}{\partial s} * \frac{\partial s}{\partial q_N} * \frac{\partial q_N}{\partial t} \tag{32}$$

where secondary terms are defined below.

$$\frac{\partial s}{\partial q_N} = \frac{1}{q_N^- - q_N^+} \quad and \quad \frac{\partial q_N}{\partial t} = \dot{q_1}$$

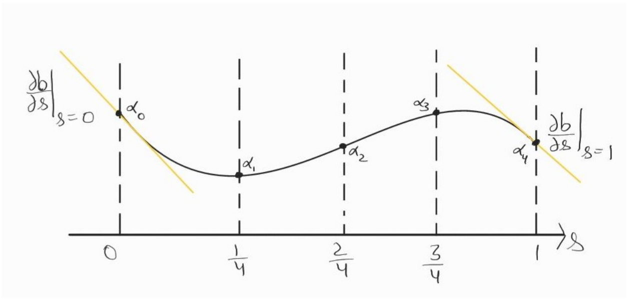
Partitioning the full dynamics described in equation 5 is accomplished by the equation

$$\begin{bmatrix} D_{11} D_{12} D_{13} \end{bmatrix} \begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \\ \ddot{q_3} \end{bmatrix} + \begin{bmatrix} C_{11} C_{12} C_{13} \end{bmatrix} \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} + G_{11} = 0 \tag{33}$$

where $\ddot{q_1}$ can be computed from the partitioned zero dynamics equations to obtain z_2 and the relation $\dot{z_2} = \ddot{z_1} = \ddot{q_1}$.

Using this formulation, the objective is to shape the Bezier polynomials to obtain feasible walking gait. An optimizer adjusts the positions of the Bezier coefficients by satisfying the declared constraints. The desired gait cycle should be periodic in nature such that the post-impact states equal the pre-impact when the impact map is applied. The optimizer must tune 8 parameters considering the Bezier coefficients and the degrees of freedom of the planar-walker system. The following properties and steps were defined in the construction of the optimizer:

1) Properties of Bezier polynomials:
   - The polynomial is confined in a convex hull of the M+1 bezier coefficients. Large oscillations do not occur during the optimization which helps in fine tuning the parameters.
   - $b_i(0) = \alpha_0^i$ and $b_i(1) = \alpha_M^i$
   - $\left.\frac{\partial b_i(s)}{\partial s}\right|_{s=0} = \left.\frac{\partial b_i(s)}{\partial s}\right|_{s=1}$ and $\alpha_0 = \alpha_M$
   - The above equalities should be satisfied for periodicity which makes α_0 and α_1 redundant, reducing the parameters required for optimization. These properties are visualized in Fig. 3 below.



$\left.\frac{\partial b}{\partial s}\right|_{s=0}$
α_0
α_1
α_2
α_3
α_4
$\left.\frac{\partial b}{\partial s}\right|_{s=1}$
0
$\frac{1}{4}$
$\frac{2}{4}$
$\frac{3}{4}$
1
s

Fig. 3: Bezier polynomial curve of degree (M) = 4

2) Steps followed for gait design using Hybrid Zero Dynamics
   a) Pre-impact state z^- is considered
   b) Bezier polynomial is exploited to enforce periodicity
   c) Post-impact state z^+ is computed using impact map
   d) Hybird zero dynamics is integrated for the gate duration
   e) Locomotion cost is checked
3) Cost Function: There are different ways by which cost function can be defined. For this project, we have used electric motor energy per distance traveled as the cost. Minimizing this function tends to reduced peak torque demands over a step.
   - Electric Motor Energy spent per distance travelled

$$Cost = \frac{1}{StepLength} \int_0^{StepDuration} |uu^T| dt \tag{34}$$

Different cost functions can produce different limit cycles but all should be periodic, symmetric and feasible.

4) Code Specifications:

a) Degree of Bezier polynomial M = 4

b) Optimizer solves for 8 parameters: the cyclic variable and its derivative as well as the 3rd, 4th and 5th coefficients for q_2 (represented by α) and q_3 (represented by γ) as shown in the vector f below:

$$f = [q_1, \dot{q_1}, \alpha_1, \alpha_2, \alpha_3, \gamma_1, \gamma_2, \gamma_3] \qquad (35)$$

c) Constraints: Swing leg end's vertical height at end of gait should be as close to contact with the ground in the y-direction as possible:

$$0 \leq P_2^v \leq 0.02 \quad meter$$

where P_2^V is the vertical position of the end of the swing leg.

Also, the stance leg should be on the left side of swing leg at gait end which is expressed by the relation below:

$$P_{m_1}(q^{-/+}) - P_{m_2}(q^{-/+}) \leq 0$$

Equality constraints: Cyclic variable (q_N) and its first derivative ($\dot{q_N}$) should be equal before and after impact.

$$norm(q_N^- - q_N^+) = 0$$

$$norm(\dot{q_N}^- - \dot{q_N}^+) = 0$$

d) The final guess for the optimizer is defined as follows:

$$f_0 = [-20, -210, 25, 40, 45, 180, 195, 200].\frac{pi}{180}$$

B. Optimizing Zero Dynamics and Bezier Coefficients Parameters

Initially the guess for $\dot{q_N}$ was $-110°/s$ with which the optimizer resulted in $\dot{q_N}$ = -0.8570 rad/s = -49.1025 $°/s$ which did not seem good enough for the robot motion as we were expecting it to be somewhere around $150°/s$. Though, desired limit cycle from Zero Dynamics looked good with this guess but when this was simulated for full dynamics, it did not make sense as shown in the plots.

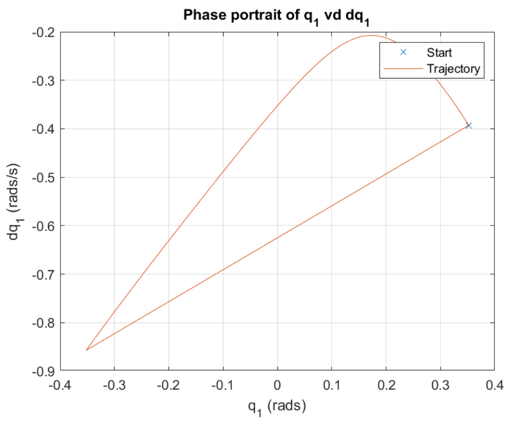


Fig. 4: Phase portrait for cyclic variable q_1 for zero dynamics looks good with bad initial guess $\dot{q_N} = -110°/s$

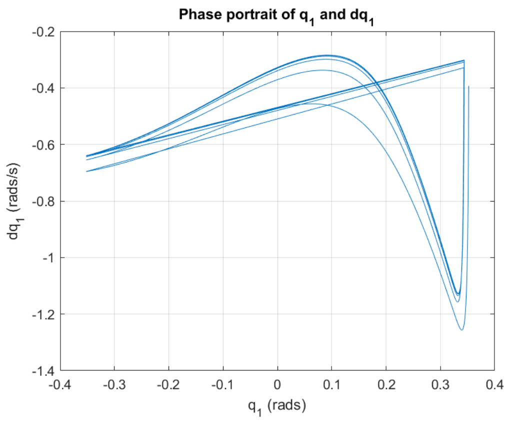


Fig. 5: Phase portrait for cyclic variable q_1 for full dynamics looks bad with bad initial guess $\dot{q_N} = -110°/s$

After correcting the guess for $\dot{q_N}$ to $-210°/s$, optimizer was able to converge and resulted in the below f which made more sense as now $\dot{q_N}$ = -2.2689 rad/s = -129.9984$°/s$, and a desired limit cycle is produced which is both symmetric and feasible.

$$f = \begin{bmatrix} -0.2996 \\ -2.2689 \\ 0.5571 \\ 1.2217 \\ 0.5992 \\ 2.9883 \\ 2.8947 \\ 2.9671 \end{bmatrix}^T \qquad (36)$$

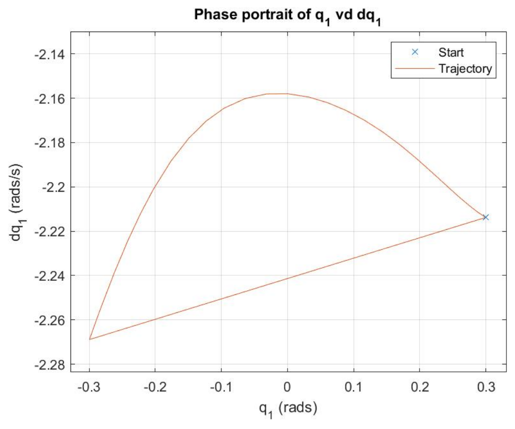


Fig. 6: Phase portrait for cyclic variable q_1 for zero dynamics with the correct initial guess

V. Mini-project 4:

A. *Feedback Control Development*

We found zero dynamics in the previous section for conducting gait design of the 3 link biped. Here we assume that the states are always in the zero dynamics manifold and never pass over. This statement is not guaranteed and feedback must be designed to provide attraction to zero dynamics manifold.
The job was completed via feedback linearization. A PD controller was created with the control action

$$u = -L_gL_fh^{-1}(\nu - L_f^2h) \tag{37}$$

where the auxiliary input is defined as

$$\nu = K_dL_fh + K_ph \tag{38}$$

The input and auxiliary inputs u, ν define control attraction while on and approaching the zero dynamics manifold respectively. Here we define appropriate gains given the results of our gait optimization:

$$K_p = \begin{bmatrix} 3200 & 0 \\ 0 & 3200 \end{bmatrix} \quad K_d = \begin{bmatrix} 320 & 0 \\ 0 & 320 \end{bmatrix} \tag{39}$$

These gains resulted from empirical experimentation to achieve the closest possible result to the desired gait. Alongside these control policies, the parameters resulting from the optimization algorithm in equation 35 above are imported into the full dynamics simulation. The numerical values acquired from the optimizer are listed above in equation 36

The simulation assigned values to the mentioned D, C, G, and B matrices using model parameters and an initial condition from a specified configuration. For the construction of the control policy (equations 37 and 38 above), develop the state space representation similarly to equation 6. Control output is correspondingly defined as follows, resulting in the desired trajectories for the joints.

$$h = \begin{bmatrix} q_2 - b2 \\ q_3 - b3 \end{bmatrix} \tag{40}$$

Subjecting h to partial differentiation with respect to states provides L_fh, the Lee derivative, and its other forms from equations 37 - 39 to define the control policy. The derivative of this term, $\frac{\partial L_fh}{\partial x}$, is also imported from an initialization script and used to define the remaining terms of the control policy below:

$$L_f^2h = \frac{\partial Lfh}{\partial x}f_x \tag{41}$$

$$L_gL_fh = \frac{\partial Lfh}{\partial x}g_x \tag{42}$$

Finally control action is computed as mentioned earlier. In the figure below we can see that the limit cycle with control action. In which, the derivative term is used to stir the velocity component and the full dynamics curve. Here proportional term is used to stabilize the system. The limit cycle for each gait follows the same kind of trajectory which shows the control action is working perfectly.

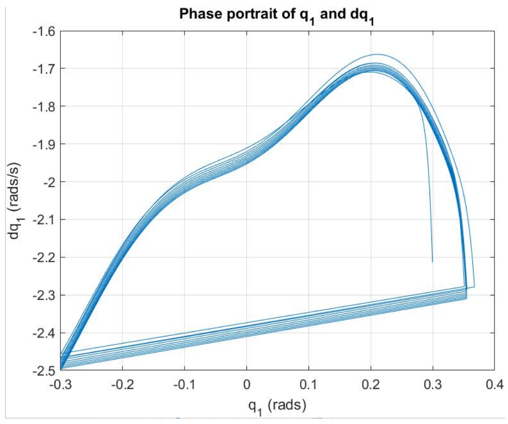


Fig. 7: Phase portrait for cyclic variable q_1 for full dynamics and feedback implemented

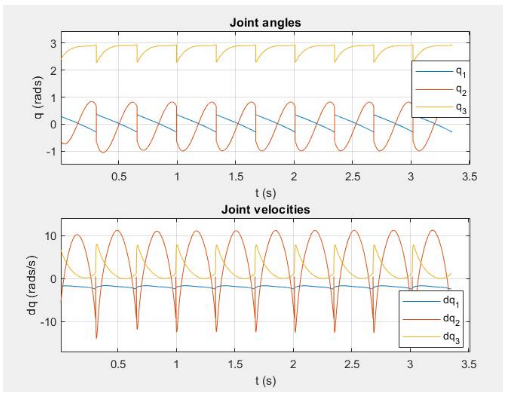


Fig. 8: Joint Trajectories for full dynamics and feedback implemented

Fig. 8 shows the trajectories obtained while walking and a point to be noted is the velocity of each joint is almost constant throughout the 10 steps provided in the code. There are some discrepancies which can be solved using further tuning of the PD controller gains.

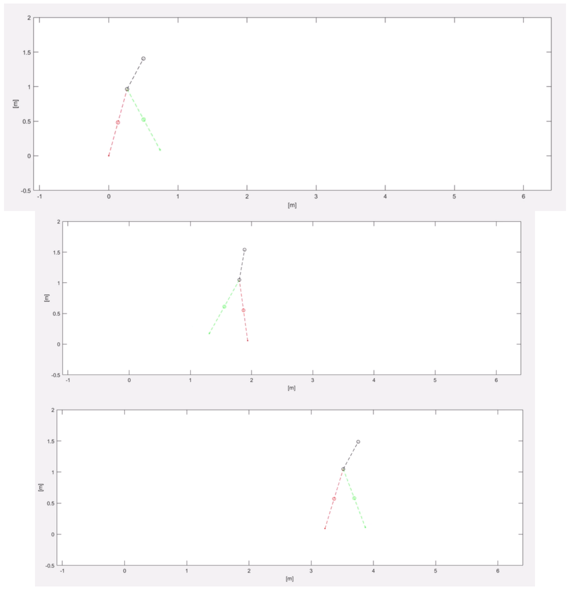


Fig. 9: Bipedal robot gait with feedback and full dynamics simulated

VI. Conclusions

In conclusion, this project gave us a high-level control overview of how to design a bipedal legged locomotion controller. By deriving the equation of motion, getting D, C, G matrix by using Euler-Lagrange formulas, we designed a Hybrid Zero Dynamics based PD controller with linear feedback, and by using a fmincon based multi-constraint optimizer, we obtained the optimal gait for 2D walking. For the simulation result, although the limit cycle derived by full dynamics is slightly different from that by zero dynamics, the gait achieved is still relatively effective and stable.

VII. References:

References

[1] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, Feedback control of dynamic bipedal robot locomotion. CRC Press, 2018.