ECE1658 Final Project Report

Hao Xiang Yang

April 14, 2025

1 Part 1 Results and Discussions

The results for part 1 are as expected. When $(q_0,\ \dot{q}_0)$ are on the limit cycle, the acrobot starts of with a constant walking gait. When $(q_0,\ \dot{q}_0)$ are not on the limit cycle, but on the constraint manifold, the acrobot has to take a few steps to correct its walking gait before entering the limit cycle. This process is asymptotically stable as long as $(q_0,\ \dot{q}_0)$ is within $[(\theta_a,\ \sigma\prime(\theta_a)\dot{\theta}_a),(\theta_b,\ \sigma\prime(\theta_b)\dot{\theta}_b)]$. When $(q_0,\ \dot{q}_0)$ are not on the constraint manifold, the acrobot's behavior becomes a little more unpredictable. With small deviation in $(q_0,\ \dot{q}_0)$ (0.01 0.02), it was able to stabilize itself at a constant walking gait. However, big deviation renders the controller unable to stabilize $(q_0,\ \dot{q}_0)$.

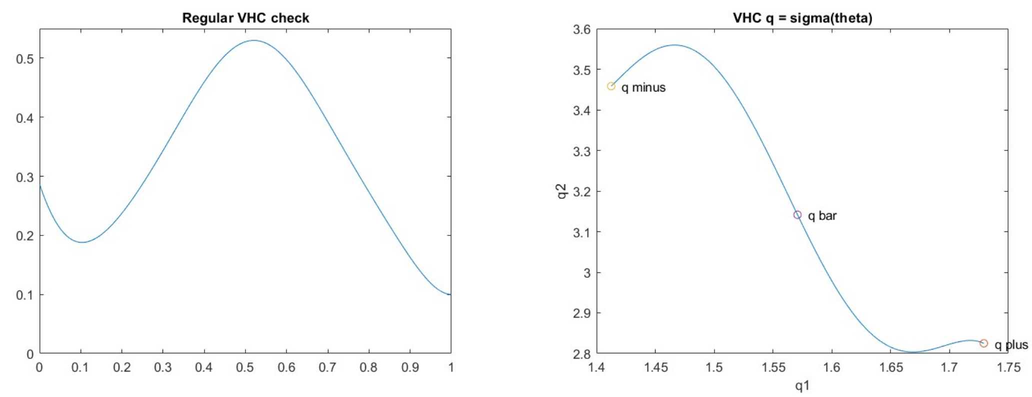

Figure 1: Transversality Condition and Hybrid Invariance Check.

The controller was set to $\tau = (\begin{bmatrix}\phi\prime(\theta)/\tilde{q}_1 & 1\end{bmatrix}\ D^{-1}B)^{-1}D^{-1}(C\dot{q}+G)+\phi\prime\prime(\theta)\dot{\theta}^2 - K_p\sin e - K_d\dot{e}$. Where $\begin{bmatrix}\phi\prime(\theta)/\tilde{q}_1 & 1\end{bmatrix}$ is the Jacobian of $\sigma(\theta)$ with respect to $\dot{q}$, $\theta = (q_1^+ - q_1)/\tilde{q}_1$ $\phi\prime\prime(\theta)\dot{\theta}^2$ is the coefficient of the $\dot{q_1}^2$, and $e = q - \sigma(\theta)$.

2 Part 2 Results and Discussions

To define the optimization problem, the constraints were separated in 4 categories, linear equalities, linear inequalities, nonlinear equalities, and nonlinear inequalities, laid out in Table 1. Having more restricting nonlinear inequalities helps with **fmincon** because it restricts the feasibility space when searching for a solution. Notice that in the code, I used an extra ϵ_{err} to account to strict inequalities. Because **fmincon** does not set up with strict inequalities, we must introduce an error tolerance to prevent the solver to consider solutions when these inequalities are = 0. I also added the nonlinear constraint that keeps q inside of W safe set to avoid the case where the optimizer settles for a solution outside of the safe set.

The first objective function observed is to minimize the length of the VHC curve $\int_0^1 ||\sigma\prime(\theta)||d\theta$. This optimization problem takes the longest to compute because the objective is a nonlinear function. The effect of this optimization is that the acrobot does the least amount of actuation to walk (See Figure (2)). This result in smallest step size and faster steps. This step is done with k = 7.

Linear Equalities	$\begin{bmatrix} 1 & 0 & 0 & ... & 0 \\ 1 & 1 & 1 & ... & 1 \\ 1 & 0.5 & 0.5^2 & ... & 0.5^{k-1} \\ 0 & 1 & 2 & ... & k-1 \\ 0 & 1 & 2(0.5) & ... & (k-1)(0.5)^{k-2} \end{bmatrix} \mathbf{a} = \begin{bmatrix} q_2^+ \\ q_2^- \\ \pi \\ v_1 \\ v_2 \end{bmatrix}$
Linear Inequalities	$v_1 \leq 2\tilde{q}_1,\ v_2 > 2\tilde{q}_1$
Nonlinear Equalities	$f(v_1) - a(2) = -\tilde{q}_1 \frac{-\mathbf{I}_{21}\tilde{q}_1 + \mathbf{I}_{22}v_1}{-\mathbf{I}_{11}\tilde{q}_1 + \mathbf{I}_{12}v_1}$ - a(2) $= 0$
Nonlinear Inequalities	Either $v_1 > \frac{\mathbf{I}_{11}}{\mathbf{I}_{12}}\tilde{q}_1$ and $v_1 \geq \tilde{q}_1 \frac{\mathbf{I}_{21}+2\mathbf{I}_{11}}{2\mathbf{I}_{12}+\mathbf{I}_{22}}$ Or $v_1 < \frac{\mathbf{I}_{11}}{\mathbf{I}_{12}}\tilde{q}_1$ and $v_1 \leq \tilde{q}_1 \frac{\mathbf{I}_{21}+2\mathbf{I}_{11}}{2\mathbf{I}_{12}+\mathbf{I}_{22}}$ $(B^{\perp} D\sigma\prime(\theta_i) \mid_{q=\sigma(\theta_i)})^2 \geq \epsilon > 0$ $0 < \frac{\delta^2}{M^-} < 1$ $\frac{V^-\delta^2}{M^- - \delta^2} + V_{max} < 0$ $q \in W$

Table 1: Optimization Constraints.

We can also minimize the v_1^2 as the objective function, which makes it quadratic and faster to solve. The resulting VHC will make the swinging leg of the acrobot to move fast at the start and slower when passing the standing leg (See Figure (3)). This step is also done with k=7.

Finally, I experimented with the number k, the order of polynomial used to represent the VHC. At higher k's, because the feasible space for the polynomial solution is bigger, the optimal solution sometimes does not stay in the W safe set. For example, as in Figure (4), the curve does not stay inside of W. The movement of the acrobot stays very close to the ground and is smoother. For even higher order polynomial, in some optimization solution, the swinging leg of the acrobot would touch the ground before the step is completed.

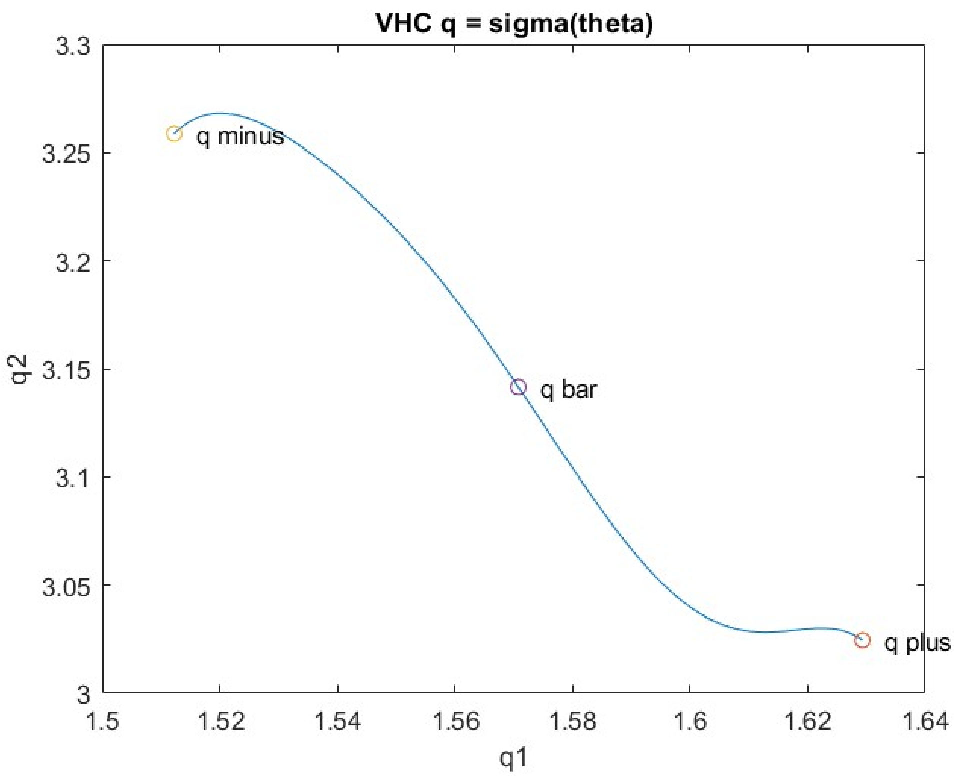

Figure 2: Optimization of Length of VHC curve.

3 Part 3 Results and Discussions

In Part 3, I decided to adapt the VHC design technique presented in part I for the 5-link robot modeled in assignment 1. We can start labeling the 5-link robot as follows in Figure (5). We can note that when the 5-link robot is at rest, we specify as a design parameter α the angle between link 1 and link 2 and between link 3 and link. This parameter lets us model the 5-link robot as an acrobot and we can reuse the techniques when modeling the acrobot.

Because the moving part that is analogous to the acrobot are q_1 and q_3, assuming stability conditions remain the same, we can use the same values as part 1 ($\beta = 0.316637, v_1 = -0.894373, v_2 = 1.9$), that correspond to the virtual acrobot. Because α is chosen to be the same for both legs, $\alpha_2 = (\pi - \alpha)/2$ on Figure (5) is also the same in the 3 labeled instances. This means we can write $q_1 = p_1 - \alpha_2 = p_1 - (\pi - \alpha)/2$ and $q_3 = p_2$, with (p_1, p_2) the joint angle of the virtual acrobot. Because we chose α for both the swinging leg and the standing leg to be equal, we want to restrict q_2 and q_4 to be the same before and after the step. Lastly, as for q_5, we can impose that the torso has to stay as a certain angle to prevent the robot from leaning towards one side when executing the movement. In Figure (5), we can clearly see that the movement ends with an additional β to q_5.

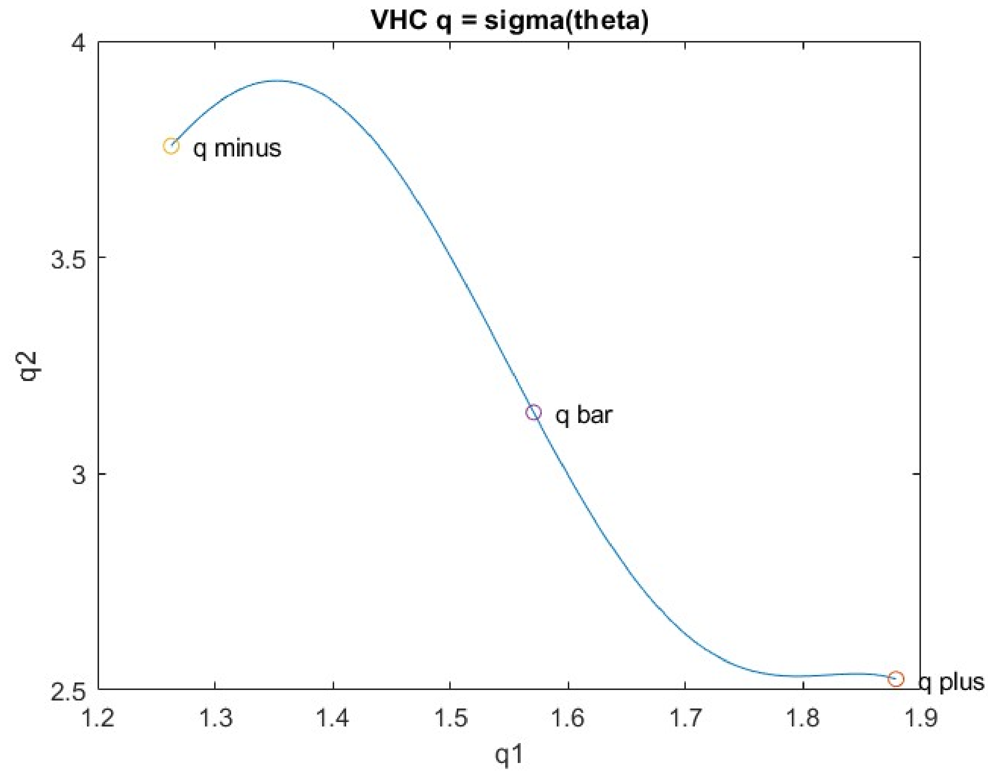

Figure 3: Optimization of V1.

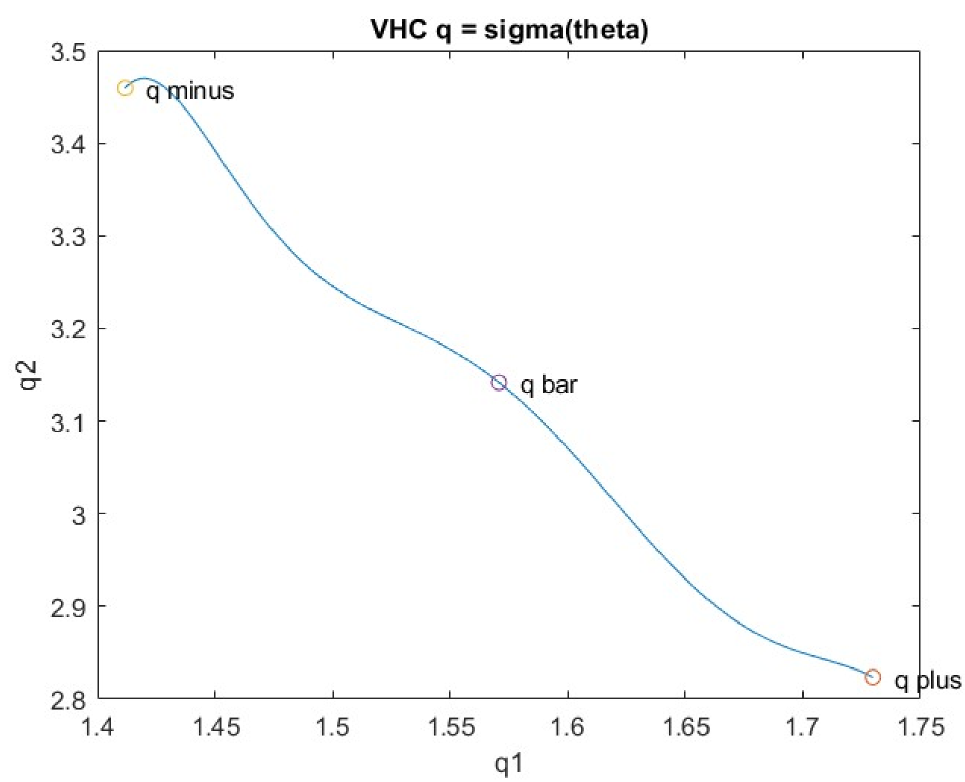

Figure 4: Optimization with K = 9.

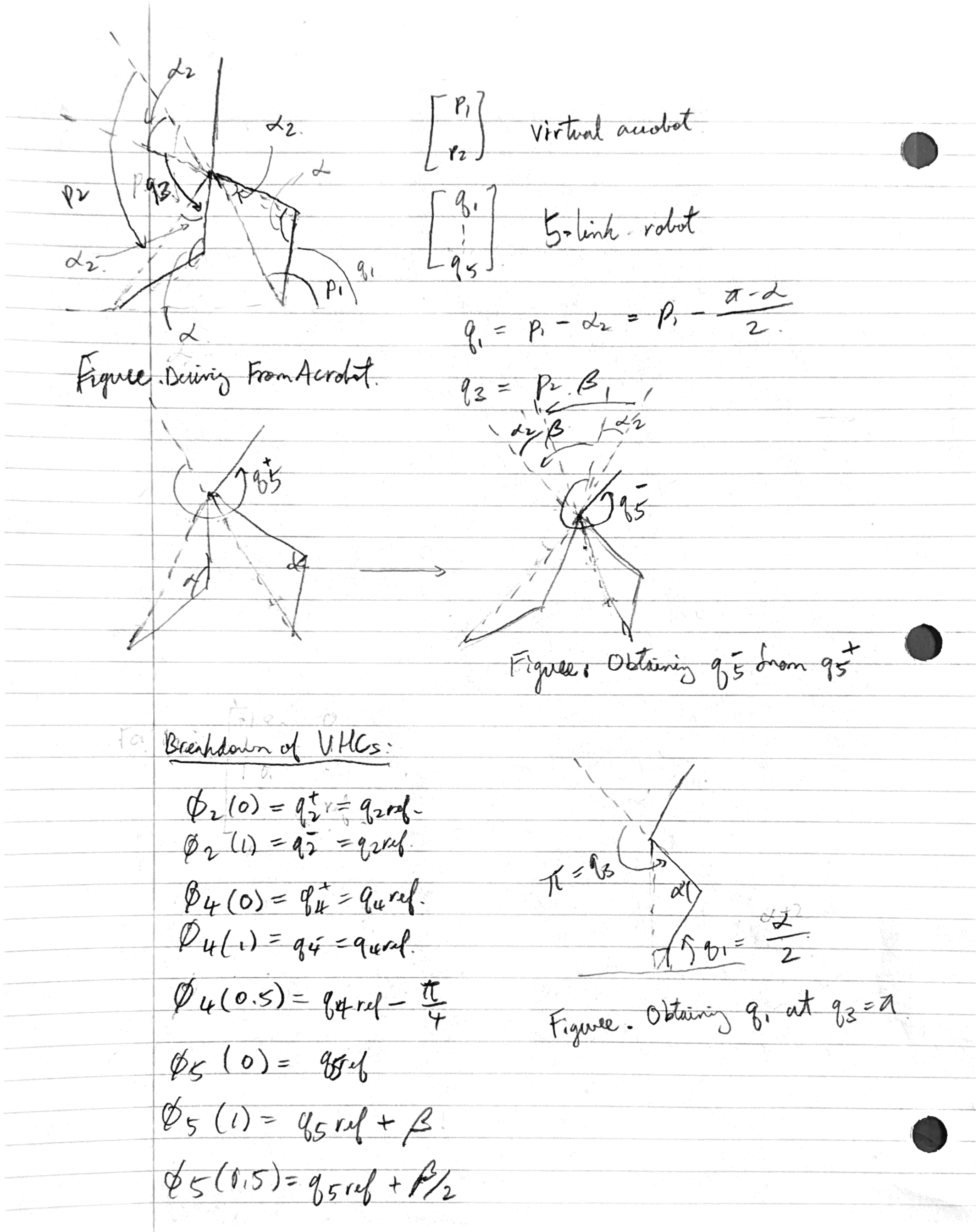

Figure 5: Diagrams for the 5-Linked Robot.

To initialize the VHC with the above observations, we can choose:

$$q^+ = \left[(\beta+\pi)/2 - (\pi-\alpha)/2 \quad q_{2ref} \quad \pi-\beta \quad q_{4ref} \quad q_{5ref}\right]^T$$
$$q^- = \left[(\beta-\pi)/2 - (\pi-\alpha)/2 \quad q_{2ref} \quad \pi+\beta \quad q_{4ref} \quad q_{5ref}+\beta\right]^T$$
$$\tilde{q} = q^+ - q^-$$

$q_{2ref} = \pi - \alpha$ and $q_{4ref} = \pi + \alpha$ are obtained based on α and the geometry of the 5-link robot in Figure (5). $q_{5ref} = -\pi/4$ as a design choice (I wanted the robot to lean slightly forward to help with the stepping and swinging movement and to move the center of gravity forward). For each actuator, a set of VHCs is assigned the same way as of part 1. We can obtain the polynomial functions $\phi_2(\theta), \phi_3(\theta), \phi_4(\theta), \phi_5(\theta)$ corresponding to q_2, q_3, q_4, q_5. The linear equation to obtain the coefficient for $\phi_3(\theta)$ is the same as the linear equations for the acrobot. A notable addition to the VHCs is that when the swinging leg is crossing the standing leg, I want the swinging leg to bend a little ($\alpha(\theta) < \alpha$) to avoid hitting the ground too early during the swing. I also wanted the torso to move linearly to avoid jerking motions that would destabilize the robot. Using $\phi_i(\theta)$'s, we can obtain

$$q = \sigma(\theta) = \begin{bmatrix} q_1^+ - \theta\tilde{q}_1 \\ \phi_2(\theta) \\ \phi_3(\theta) \\ \phi_4(\theta) \\ \phi_5(\theta) \end{bmatrix}$$

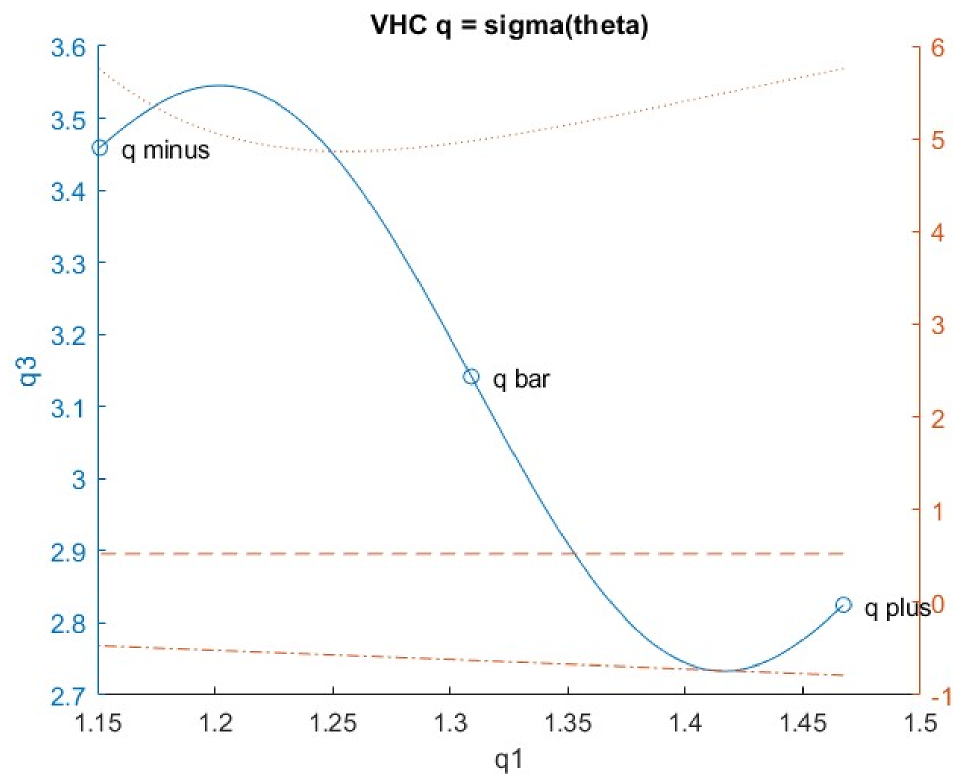

Figure 6: Behavior of q with respect to q_1. The dotted line represents $q4$, the dashed line q_2 and the dashed-dotted line q_5

In Figure 6, we see that $q3$ stayed in the W safe zone. This corresponds to the fact that the swinging leg that is following the virtual acrobot is above ground. When the swinging leg passes the standing leg (ie when $q_3 = \pi$) (see Figure (5)), we can calculate $\bar{q}_1 = \alpha/2$ as the inflection point of W. The dotted line representing q_4 behaves expectedly as we want the robot to lift its swinging left during the step. The dashed line q_2 and the dashed-dotted line q_5 are both linear as specified by the constraints. We can also verify the transversality condition $B^{\perp} D\sigma\prime(\theta_i) \mid_{q=\sigma(\theta_i)} \neq 0$ using the same method as part 1.

The last check is to ensure the existence and stability conditions:

$$0 < \frac{\delta^2}{M^-} < 1 \tag{1}$$

$$\frac{V^- \delta^2}{M^- - \delta^2} + V_{max} < 0 \tag{2}$$

My VHCs pass condition (1) but fail at condition (2). Further examinations are needed to find out what when wrong in the set up of the 5-link robot with the design parameters. My suspicions are in the values of v_1 and v_2. Unlike β which is the same parameter regardless of the model type, I suspect that a v_1 and v_2 that puts the 5-link robot on a stable limit cycle might be different than the ones of the acrobot. According to my animations, the robot settles at a constant walking gait for more than 100 steps. It is possible however, because the limit cycle conditions are not met, that at some step $N >> 1$ the robot will become unstable and falls over for example. It is also possible that the limit cycle I reached is asymptotically stable, but not exponentially stable, therefore not meeting the hybrid limit cycle conditions. This gap can potentially be bridged with future work.

Lastly, the controller for the 5-link robot is the same PID controller from Assignment 1 with few small tweaks. For one, because the input $\theta = \frac{q_1^+ - q_1}{\tilde{q}_1}$ for each $\sigma_i(\theta)$, I changed the $dh_q = H$ to include components in the first column as all q's now depends on q_1. Similarly, the error is changed to $y_i = q_i - \phi_i(\theta)$. With the introduction of a hessian, we have to include additional $ddh_q = [-\phi\prime\prime_i(\theta)/\tilde{q}_1^2]\ \forall i = 2...5$ to the controller. Then, using the initial conditions $q_0 = \sigma(0)$ and $\dot{q}_0$ the same as the acrobot for $\dot{q}_1, \dot{q}_3$ tp ensure viability, I produced a walking 5-link robot!

The behavior of the 5-link robot is also within the bounds of the VHCs. The robot starts at a very high speed, but settles at a constant walking gait after a few steps, suggesting that it has reached, at least in practice, a limit cycle. During that transition, we can see that the robot from the animation glitched due to the swinging leg hitting the floor too early. This might have to do with not meeting the hybrid limit cycle conditions. However, the controller was able to recover from this misstep.

Overall, I think it was definitely a challenge to get the robot to walk. The *ode*45 solver hit a singularity so many times that I cannot even count. Setting up the 5-link robot was tricky. I was pleasantly surprised that when everything got set up properly the robot just started walking without any major changes aside from simulating a virtual acrobot.