# ECE1658 Final Report

Kevin Hu

April 22, 2024

# 1 Part 1

## 1.1 Initial VHC Design

The initial VHC design was successful. The initial VHC design satisfies all the checks required for a VHC. The resulting VHC is plotted below. The safe set $\sqsupseteq$ is highlighted in yellow. The tangent vectors are shown at $q^+$, $q^-$ and at the scuffing point.

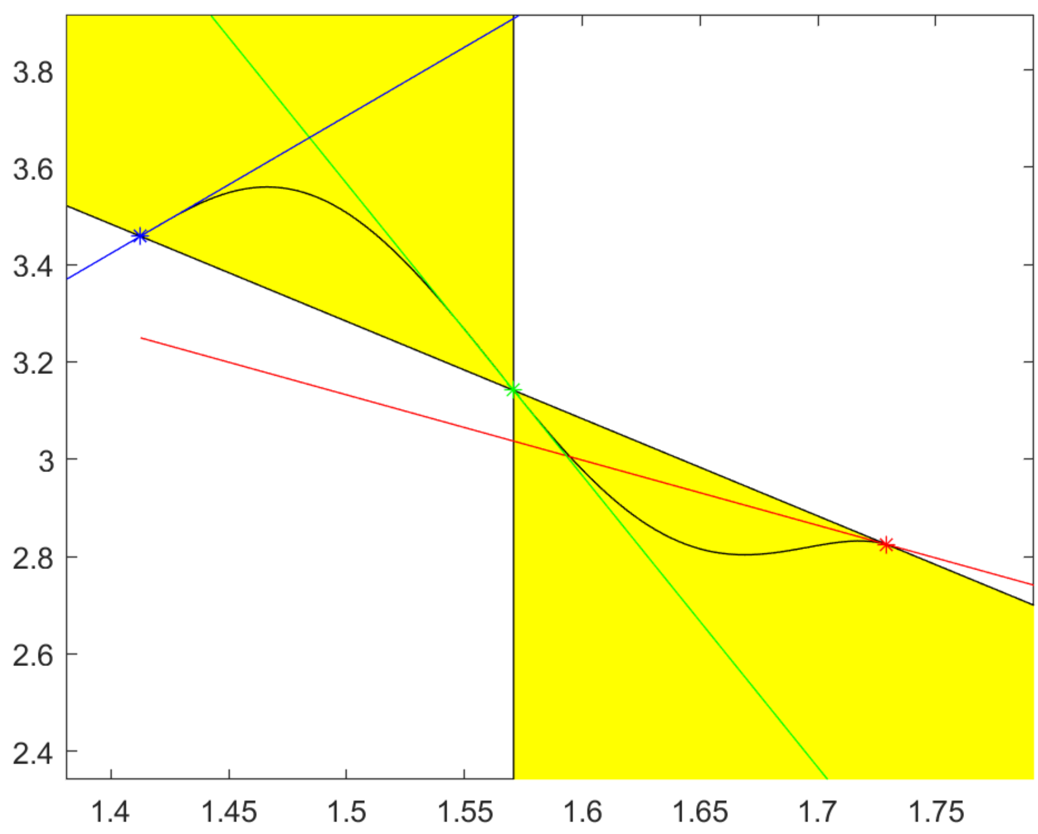


Figure 1: Initial VHC Plot

The curve is a regular VHC since it satisfies the transversality condition $B^{\perp}D\sigma'(\theta) \neq 0$ for all $\theta = [0, 1]$. The expression $B^{\perp}D\sigma'(\theta)$ is plotted for an array of values between $\theta = [0, 1]$ below. There are no zero-crossings.

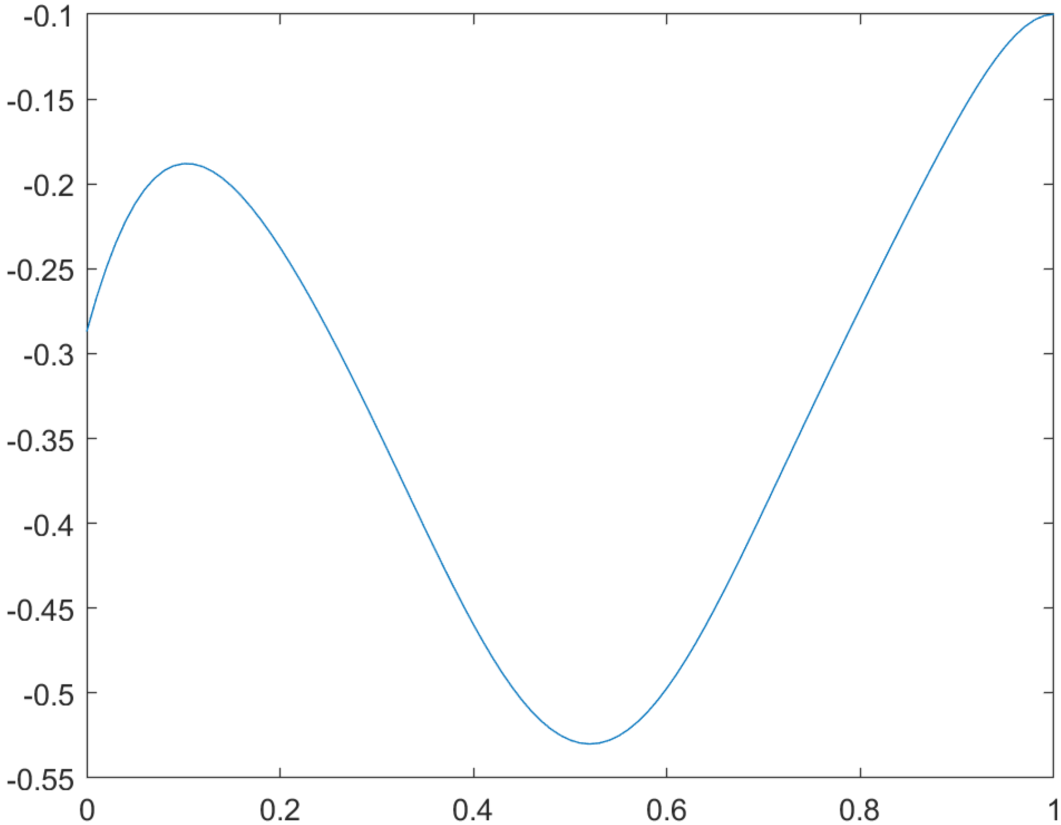


Figure 2: Transversality Condition Plot

Here is the plot of the potential functions $M(\theta)$ and $V(\theta)$:

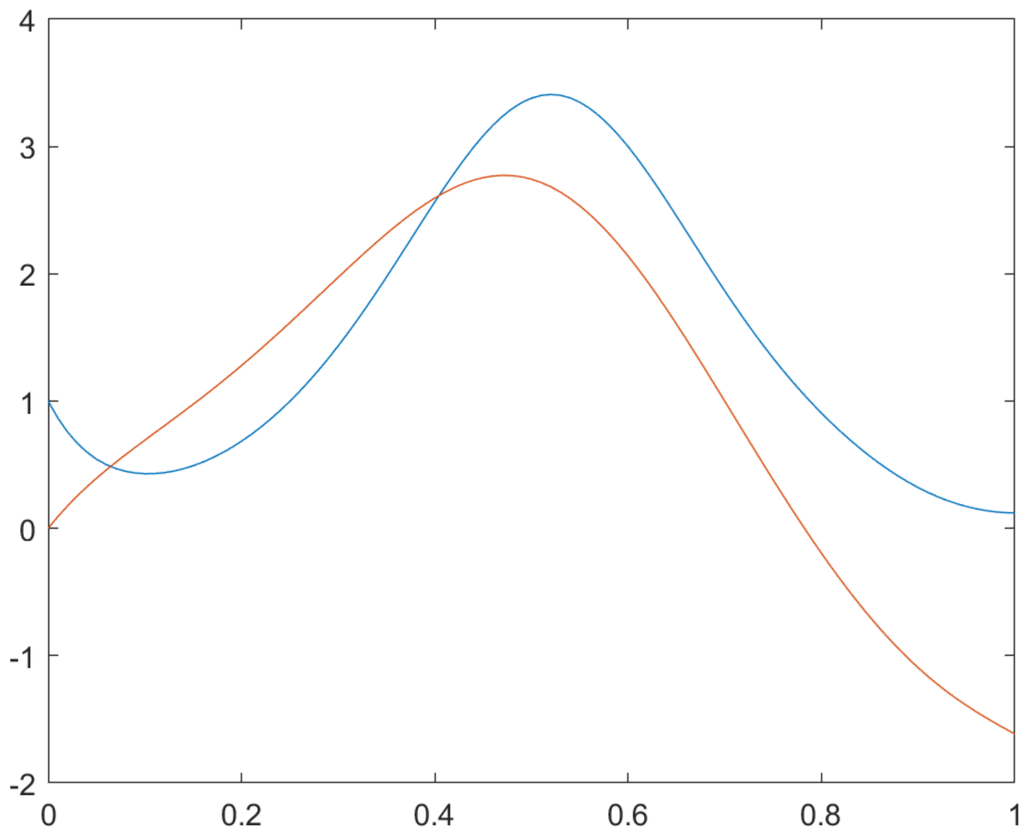


Figure 3: Potential Functions Plot

The controller for the initial VHC design by having $h = q_2(t) - \phi(\theta) = 0$. Quantities $dhq$ and $d(dhq)/dt$ were computed symbolically and turned into matlab functions. Using the equation for $\tau$ in the lecture notes for a robot, the controller was defined.

## 1.2 Optimized VHC Design

The optimized VHC problem was done in Matlab by creating matlab functions for $\phi$ and $\sigma$ and their derivatives that depended on the polynomial coefficients and other variable parameters in the optimization problem. The constraints were separated into linear equality, linear inequality, and nonlinear constraints. The majority of the constraints in this optimization problem were nonlinear constraints. The potential functions were calculated using numerical integration and fit to splines for later use.

The optimized VHC design was successful but only in certain cases. However, the code often failed to find a minimum or failed to satisfy all the constraints due to fmincon not forcing all the constraints to be satisfied at every iteration. Fmincon getting stuck in local minima was also an issue. Bounds were not added for optimization, but adding bounds may help improve the optimization result in the future since fmincon does honour bounds at every iteration.

Sometimes the constraints would be satisfied by would fail the VHC checks to make sure it was a vaild VHC curve so it is possible that there is a bug in the code.

The evaluation function that worked the best was the evaluation function that made the curve as flat as possible at the endpoint. A higher number of polynomial coefficients for the function $\phi$ were tested but 6 polynomial coefficients seemed to work the best. The optimization algorithm used was the sqp algorithm which performed better than the default interior-point algorithm.

The polynomial coefficients found were $a10 = 2.835$, $a1 = 0.4970$, $a2 = -7.3164$, $a3 = 29.0994$, $a4 = -34.3477$ and $a5 = 12.6804$. The values of beta, v1 and v2 were $beta = 0.3063$, $v1 = 0.8264$ and $v2 = 1.7939$. The value of epsilon used was $epsilon = 1e-4$ and an error term used for evaluating "greater than" or "less than" inequalities since fmincon does not accept these inequalities was $e = 1e-5$.

The optimized VHC design satisfies all the checks. The VHC plot for the optimized design is shown below:

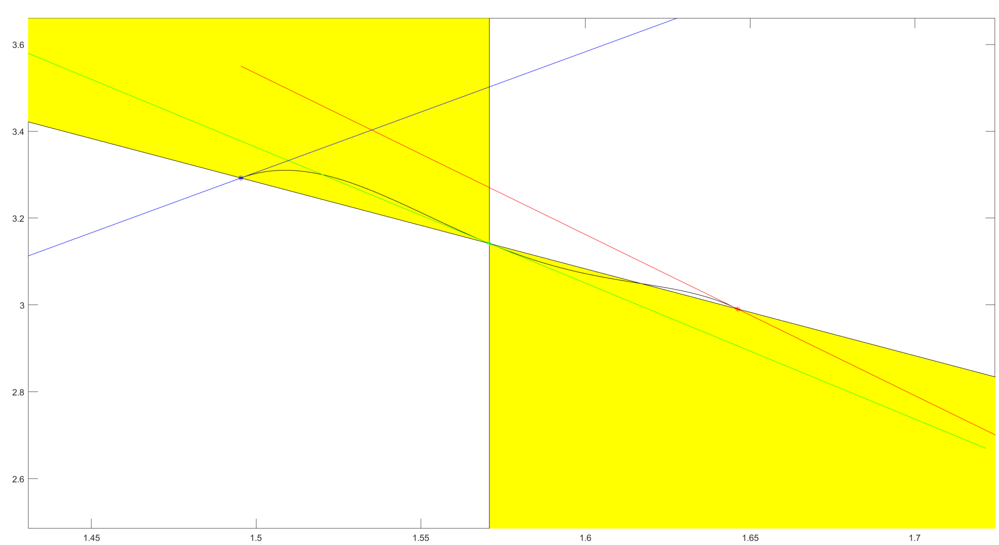


Figure 4: Optimized VHC Plot

The transversality condition plot with no zero crossings is shown below:

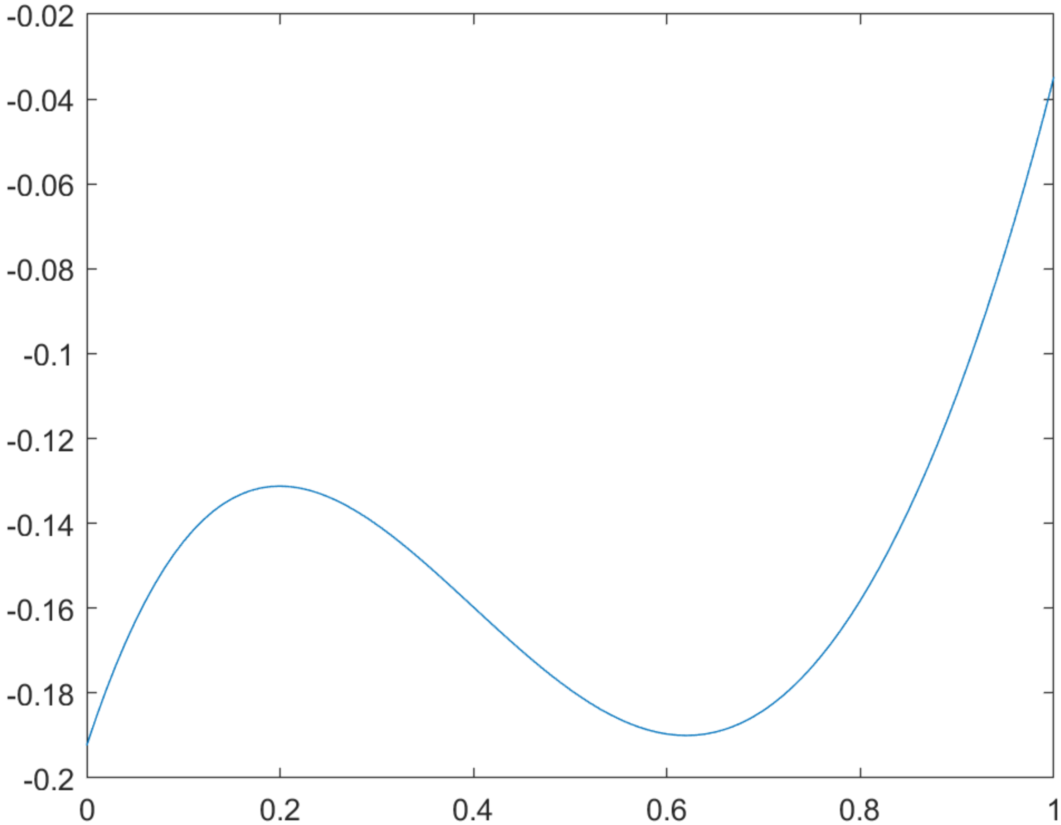


Figure 5: Optimized Transversality Condition Plot

Here is the plot of the potential functions $M(\theta)$ and $V(\theta)$:

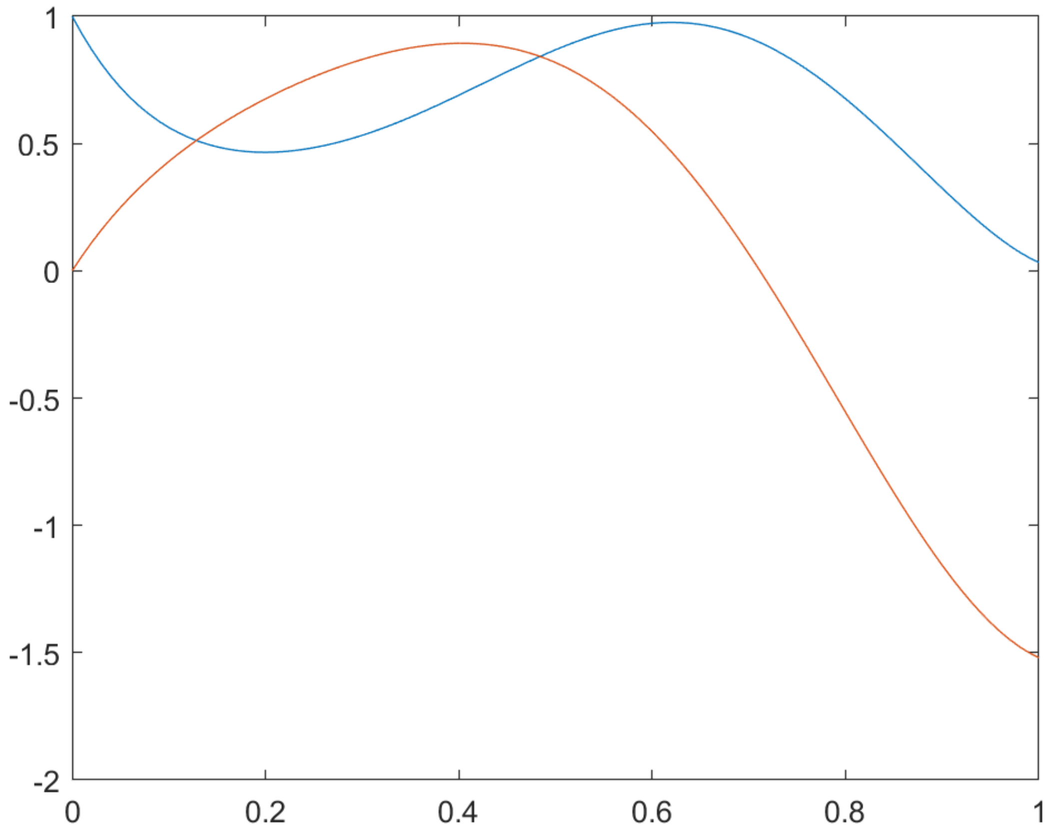


Figure 6: Optimized Potential Functions Plot

## 2 Part 2 Walkover Gait VHC Curve

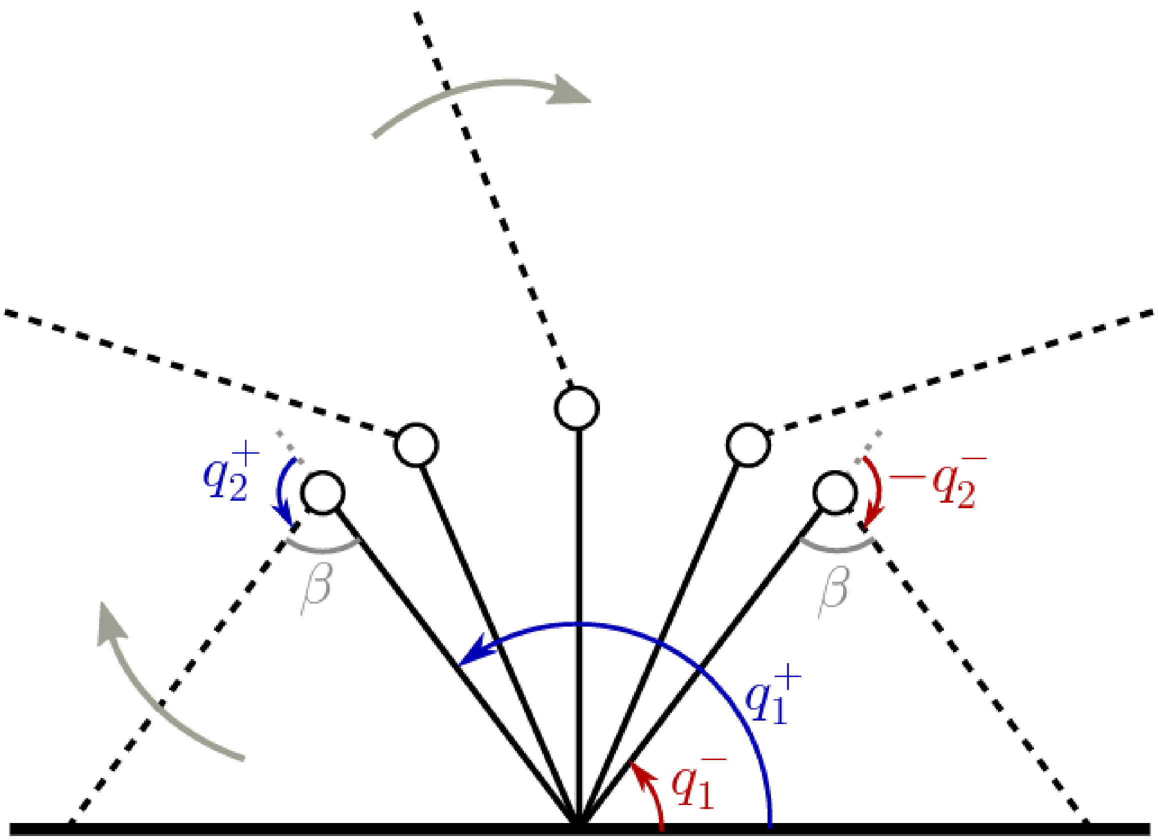


Figure 7: Walkover Gait Design

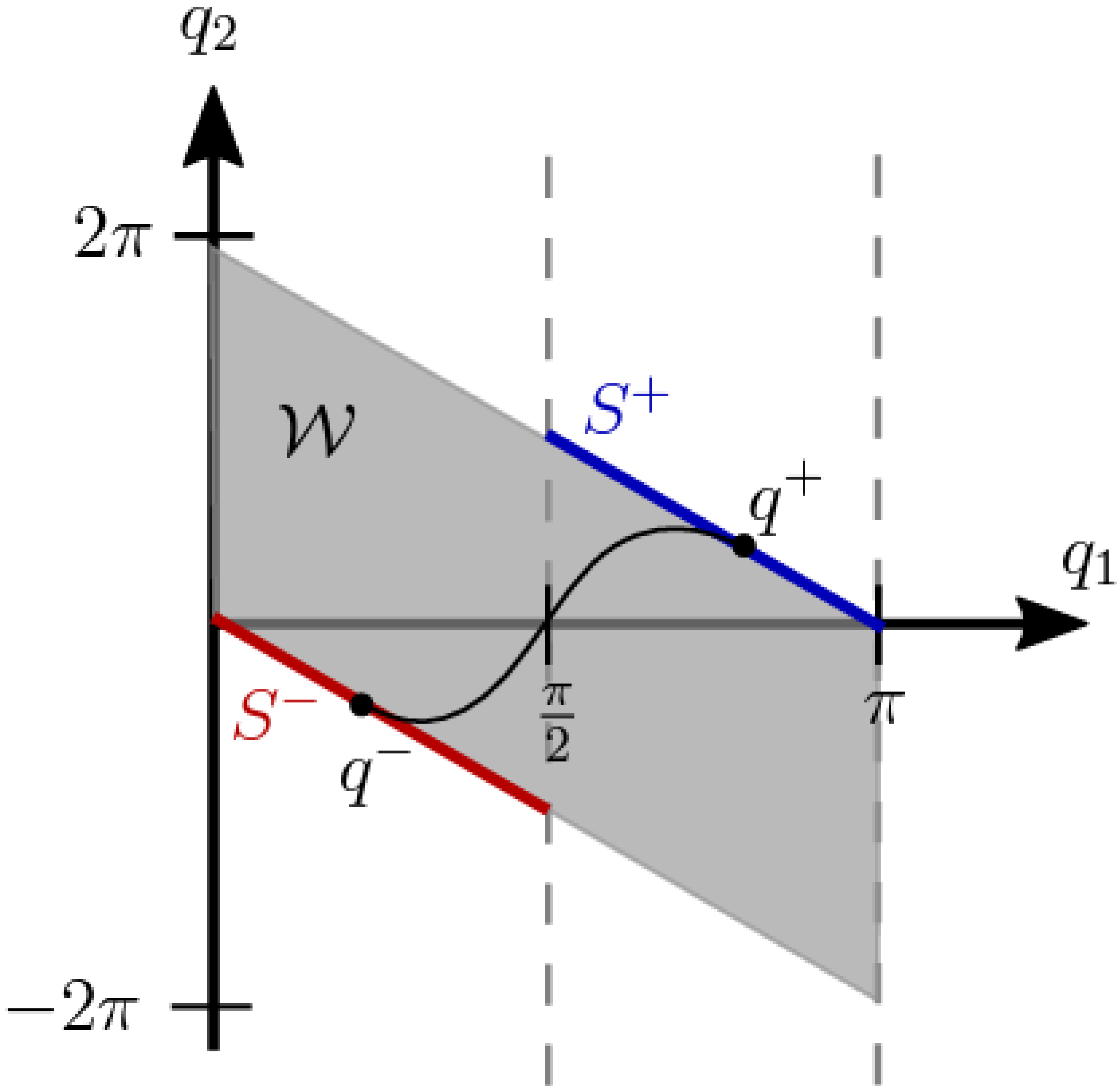


Figure 8: VHC curve for Walkover Gait

## 2.1 Problem Formulation

For the creative project, a VHC curve for the walkover gait in [1] was attempted to be found using a similar optimization problem formulation as in the paper [1] and with inspiration from Part 1.

## 2.2 Methodology for Solution

The mathematical quantities for the robot model were taken from the paper as well as the relabelling and impact maps [1]. The main difference between the optimization problem formulation here compared to the method done in Part 1 is that a single-input planar control system is used to model both the hybrid invariance and transversality conditions.

The single-input planar control system is shown below:

$$\frac{d\sigma}{d\theta} = f_{acr} + g_{acr}(\sigma(\theta))u(\theta)$$

Where u is a virtual control input. In this problem $\sigma(\theta)$ is the same parameterization as in Part 1, represented by a six coefficient Bezier polynomial.

The virtual control input is also a function parameterized by $\theta$. According to the paper [1], u is in the set of twice differentiable functions continuous in the domain $[\theta_a, \theta_b]$. In order to simplify the problem, assume u can be represented by a three coefficient Bezier polynomial.

For the optimization problem, the unit tangent vector to the curve $q = \sigma(\theta)$, $t(\theta)$ at $\theta_b$ is also being solved for. Note that $t(\theta) = \sigma'(\theta)/\|\sigma'(\theta)\|$.

There is a function $\Theta$ that can transform $t(\theta_b)$ to $t(\theta_a)$ that can be defined in terms of the matrix $\Omega(q^-)$ which is the same as matrix $I$ used in Part 1.

$$t(\theta_a) = \Theta(t(\theta_b)) = -\frac{\Omega(q^-)t(\theta_b)}{\|\Omega(q^-)t(\theta_b)\|}$$

Using this knowledge, two equality constraints are introduced into the problem,
$\sigma'(\theta_b) = \|\sigma'(\theta_b)\|t(\theta_b)$, and $\sigma'(\theta_a) = \|\sigma'(\theta_a)\|\Theta(t(\theta_b))$.

In order for the tangent vectors $t(\theta_a)$ and $t(\theta_a)$ to satisfy the transversality condition, the tangent vectors are required to satisfy the inequalities:

$$\mu det(t(\theta_b)D^{-1}(q^-)B) > 0$$
$$\mu det(\Theta(t(\theta_b))D^{-1}(q^+)B) > 0$$

where $\mu = sign(det(f_{acr}g_{acr}))$. Similar to part 1, there are conditions requiring the tangent vector $t(\theta_b)$ to point to the outside the safe set and the tangent vector $t(\theta_a)$ to point inside the safe set. Let $n = -[2 \quad 1]^\top$ be the normal vector to the bounds of the safe set $q_2 = -2q_1$ and $q2 = -2q_1 + 2\pi$ Thus we have the inequalities $<t(\theta(b)), n> \leq 0$ and $<\Theta(t(\theta(b))), n> \leq 0$.

The virtual control input u also has to follow the fact that $u(\theta_a) = u_a$ and $u(\theta_b) = u_b$ where $u_a$ and $u_b$ can be determined by solving the equations below:

$$\frac{1}{\|\sigma'(\theta_b)\|}[1 \quad u_b]^\top = [f_{acr} g_{acr}(q^-)]^{-1} t(\theta_b)$$
$$\frac{1}{\|\sigma'(\theta_a)\|}[1 \quad u_a]^\top = [f_{acr} g_{acr}(q^+)]^{-1} \Theta(t(\theta_b))$$

Finally, we have the inequalities representing the safe set.

$$I_1(q) = q_2 + 2q_1 - 2\pi$$
$$I_2(q) = -2q_1 - q2$$
$$I_3(q) = -q_1$$
$$I_4(q) = q_1 - \pi$$

The conditions for the VHC to have a stable hybrid limit cycle were also included in the problem and are the same as used in part 1.

### 2.3 Optimization Problem

A summary of the optimization problem is shown below.

$$\underset{(\theta_b,u)\in(\theta_a,\infty)\times\mathcal{U}}{\text{minimize}} \quad J(\theta_b,u)$$

$$\text{subject to } \frac{d\sigma}{d\theta} = f_{\text{acr}}(\sigma(\theta)) + g_{\text{acr}}(\sigma(\theta))u(\theta)$$

$$\sigma(\theta_a) = q^+$$

$$\sigma(\theta_b) = q^-$$

$$u(\theta_a) = u_a$$

$$u(\theta_b) = u_b$$

$$\delta^2 V^{\sigma,u}(\theta_b)/\left(M^{\sigma,u}(\theta_b) - \delta^2\right) + \max_{\theta\in[0,\theta_b]} V^\sigma < -\varepsilon$$

$$\varepsilon_1 < \delta^2/M^{\sigma,u}(\theta_b) < 1 - \varepsilon_2$$

$$I_j(\sigma(\theta)) \le 0, \;\; \theta \in (0,\theta_b), \;\; j = 1,\ldots,4,$$

Figure 9: Optimization Problem Formulation

Similar to the optimization problem in part 1, we sample N = 1000 random points from a uniform distribution for theta between 0 and 1. The 1000 values of theta are used to generate equality constraints for $\frac{d\sigma}{d\theta} = f_{acr} + g_{acr}(\sigma(\theta))u(\theta)$. These 1000 samples are also used to make sure $\sigma(\theta)$ at all times by making 4N inequality constraints for $I_j(\sigma(\theta)$. Like in part 1 $V_max$ is also determined via the 1000 sample points. The quantities being optimized include the coefficents ai for the function $\sigma$, the coefficients ui for the function u, the value $\beta$, and two values for the tangent vector $t(\theta_b)$.

There are several design parameters including $\epsilon, \epsilon_1, \epsilon_2, e > 0$, where $e$ is the error term for less than and greater than inequalities. All were set to 1e-5.

The objective function being solved is $J(\theta_b, \sigma, u) = J_1(\theta_b, \sigma, u) + \gamma J_{reg}(u)$, where

$$J_1(\theta_b, \sigma, u) = \int_0^1 \|\sigma'(\theta)\| d\theta$$

$$J_{reg}(u) = (\int_0^1 \|u''(\theta)\|^2 d\theta)^{1/2}$$

and $\gamma$ is a design parameter set to 0.5.

## 2.4 Results

Unfortunately without a good initial condition, fmincon will not converge to a solution and there are several constraint violations. Finding a good initial condition will require you to basically solve the problem itself and involves solving systems of many nonlinear equations, which there is not enough time for. So initial conditions are generated randomly within a reasonable range of values to try to find a good initial condition that will result in a minimum to be found and the constraints to be satisfied.

The optimization algorithm has some issues with numerical integration failing and matrix inversions being close to singular occasionally.

Currently, the best solution found has a 2.358 feasibility score or constraint violations. So it is probably not a valid VHC curve. The optimization algorithm interior-point was used with feasibility mode enabled.

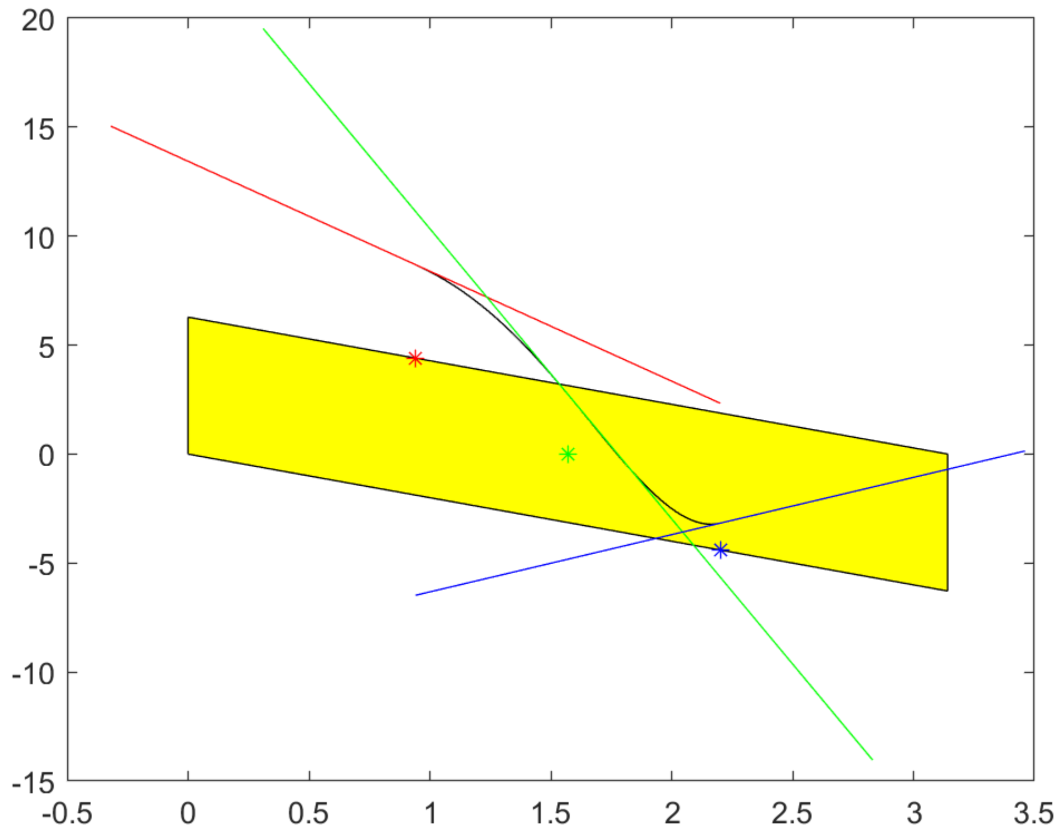


Figure 10: Plot of Curve

Here is the transversality condition:

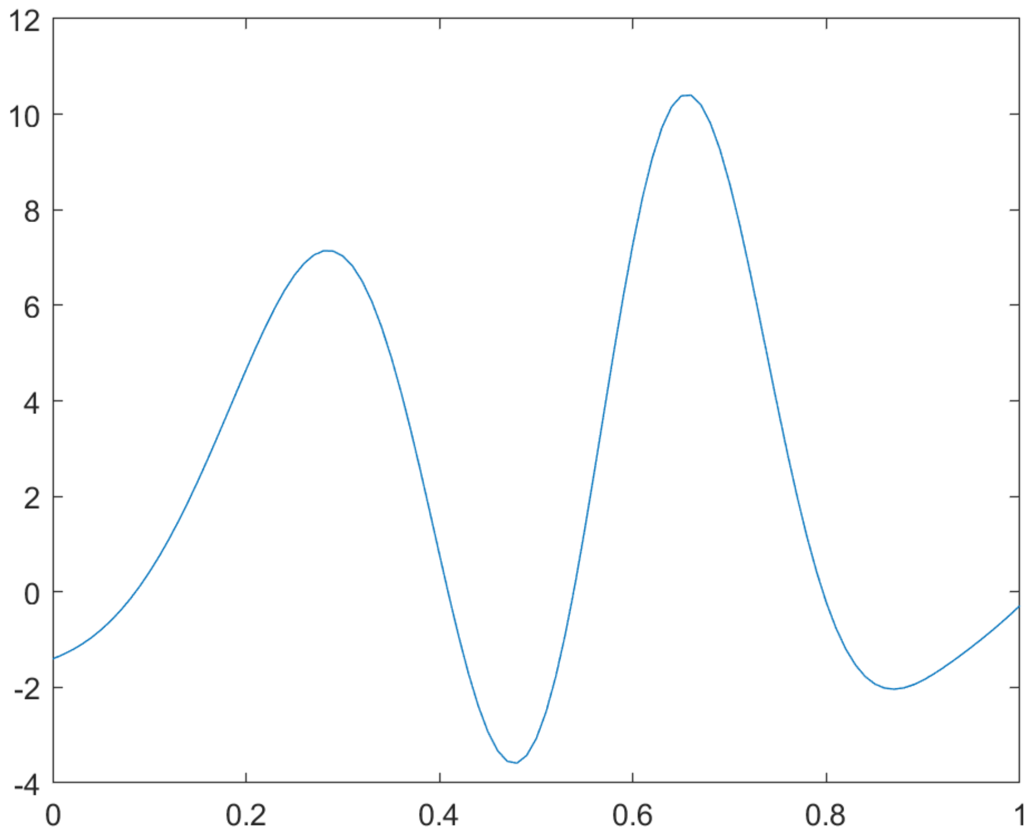


Figure 11: Transversality Condition Plot

As can be seen, there are zero crossings in the transversality condition and the VHC is not aligned properly in the plot.

### 2.5 Conclusion

Overall, the work was unsuccessful in producing a valid VHC curve. However, the method does seem promising if there was more time and a good initial condition can be found and potential bugs are fixed.

## 3 References

[1] E. Kao-Vukovich and M. Maggiore, “On the synthesis of stable walkover gaits for the acrobot,” IEEE Transactions on Control Systems Technology, vol. 31, no. 3, pp. 1379–1394, May 2023. doi:10.1109/tcst.2022.3224315