Assignment 2: Optimization and Optimal Control

Due Mar. 5 at 23:59

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use Google to find solutions for assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment.

Submitting Your Assignment

The assignment must be submitted on Canvas. You must submit one zip file (student_number_a2.zip) containing:

1. An assignment report in **PDF format**, named `student_number_a2.pdf`. This report should contain your solutions to questions 1-4.
2. Your code for question 1, named `a2_q1.py` or `a2_q1.m`.
3. Your code for question 1, named `a2_q2.py` or `a2_q2.m`.
4. Your code for question 4, named `a2_q3.py` or `a2_q3.m`.
5. Your code for question 4, named `a2_q4.py`.

---

1 Sequential Quadratic Programming

Notation:

- The decision variable is $x = (x_1, x_2)$.
- Subscripts denote vector components, and superscripts denote the iteration number.

a) Since our constraints are already linear, in quadratic subproblem we only need to quadratize the objective. Thus, the quadratic subproblem is given by

$$\begin{aligned} \text{minimize} \quad & \nabla f(x_1^k, x_2^k)^\top d^k + \frac{1}{2}(d^k)^\top \mathrm{H}f(x_1^k, x_2^k) d^k && \text{(1a)} \\ \text{subject to} \quad & -x_1 - 1 \leq 0 && \text{(1b)} \\ & x_1 - 1 \leq 0 && \text{(1c)} \\ & -x_2 - 1 \leq 0 && \text{(1d)} \\ & x_2 - 1 \leq 0 && \text{(1e)} \end{aligned}$$

Since $x = x^k + d^k$ where $d^k = (d_1^k, d_2^k)$ is the step direction, we can rewrite the quadratic subproblem as follows:

$$\begin{aligned} \text{minimize} \quad & \nabla f(x_1^k, x_2^k)^\top d^k + \frac{1}{2}(d^k)^\top \mathrm{H}f(x_1^k, x_2^k) d^k && \text{(2a)} \\ \text{subject to} \quad & -x_1^k - d_1^k - 1 \leq 0 && \text{(2b)} \\ & x_1^k + d_1^k - 1 \leq 0 && \text{(2c)} \\ & -x_2^k - d_2^k - 1 \leq 0 && \text{(2d)} \\ & x_2^k + d_2^k - 1 \leq 0 && \text{(2e)} \end{aligned}$$

Now, we compute the gradient and Hessian. First, let $f(x_1^k, x_2^k) = \sin(\pi x_1)\sin(2\pi x_2)$, then

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \pi\cos(\pi x_1)\sin(2\pi x_2) \\ 2\pi\sin(\pi x_1)\cos(2\pi x_2) \end{bmatrix} \tag{3}$$

$$\mathrm{H}f(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 f}{(\partial x_1)^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{(\partial x_2)^2} \end{bmatrix} = \begin{bmatrix} -\pi^2\sin(\pi x)\sin(2\pi y) & 2\pi^2\cos(\pi x)\cos(2\pi y) \\ 2\pi^2\cos(\pi x)\cos(2\pi y) & -4\pi^2\sin(\pi x)\sin(2\pi y) \end{bmatrix} \tag{4}$$

b) Please see `sqp.m`, in the function `solve_quad_subproblem`.

c) Please see `sqp.m`. Plots are shown in Figure 1.

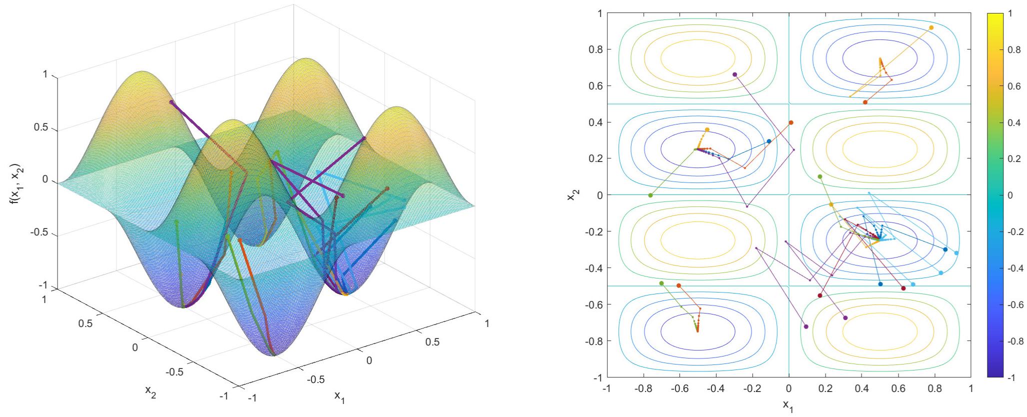

Figure 1: Visualization of solutions to Question 1.

2 Differential Flatness

a) For the function β, we need to write each of the state variables as a function of z and its derivatives only, or x and y and their derivatives only.

First, similar to our discussion in class, we have

$$\frac{\dot{y}}{\dot{x}} = \tan\theta \tag{5a}$$

$$\theta = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \tag{5b}$$

Next, $v = \sqrt{(\dot{x})^2 + (\dot{y})^2}$, as discussed in class.

The difference now is that v is a state instead of a constant. Here, note that θ can be written as a function of $\dot{x}$ and $\dot{y}$, so v can also be written as a function of derivatives of x and y – namely $\dot{x}$ and $\dot{y}$. Summarizing, we have

$$\begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \\ \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \\ \sqrt{(\dot{x})^2 + (\dot{y})^2} \end{bmatrix}. \tag{6}$$

Again, note that the entire right-hand side can be written as a function of $x, y, \dot{x}, \dot{y}$.

To obtain function γ, we can simply take the derivative of θ and v, respectively. Since θ and v can be written in terms of $x, y, \dot{x}, \dot{y}$, so too can $\dot{\theta}$ and $\dot{v}$. Thus, without writing out ω and a explicitly in terms of x, y and their derivatives, we simply have

$$\begin{bmatrix} \omega \\ a \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \dot{v} \end{bmatrix}, \tag{7}$$

where θ and v can be written in terms of x, y and their derivatives.

b) For clarity, we first explicitly write the parametrizations of $x(t)$ and $y(t)$:

$$x(t) = b_{00} + b_{01}t + b_{02}t^2 + b_{03}t^3 \tag{8a}$$
$$y(t) = b_{10} + b_{11}t + b_{12}t^2 + b_{13}t^3 \tag{8b}$$

Taking their derivatives, we get

$$\dot{x}(t) = b_{01} + 2b_{02}t + 3b_{03}t^2 \tag{9a}$$
$$\dot{y}(t) = b_{11} + 2b_{12}t + 3b_{13}t^2 \tag{9b}$$

The initial and final conditions are given in the original state space, and we need to first write them in the flat output space involving only x, y and their derivatives. The conditions on x and y are straight forward, since they are the flat outputs. We can obtain the conditions on $\dot{x}$ and $\dot{y}$ from the dynamics:

$$\dot{x}(0) = v(0)\cos(\theta(0)) = 1 \tag{10a}$$
$$\dot{x}(T) = v(T)\cos(\theta(T)) = 0 \tag{10b}$$
$$\dot{y}(0) = v(0)\sin(\theta(0)) = 0 \tag{10c}$$
$$\dot{y}(T) = v(T)\sin(\theta(T)) = 1 \tag{10d}$$

Thus, to summarize, the constraints representing initial and final conditions in the flat output space are as follows:

$$\begin{bmatrix} x(0) \\ \dot{x}(0) \\ x(T) \\ \dot{x}(T) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} y(0) \\ \dot{y}(0) \\ y(T) \\ \dot{y}(T) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

Incorporating the parametrizations of $x, y, \dot{x}, \dot{y}$, we have the following.

$$\text{For } x \text{ and } \dot{x}\text{:} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^3 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ b_{03} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

$$\text{For } y \text{ and } \dot{y}\text{:} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^3 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} b_{10} \\ b_{11} \\ b_{12} \\ b_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{13}$$

Alternatively, the following is also equivalent:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & T^2 & T^3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ b_{03} \\ b_{10} \\ b_{11} \\ b_{12} \\ b_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{14}$$

c) To obtain the state and control trajectories from the flat outputs and their derivatives we use the result from part a, in Eq. (6).

One caveat here is division by zero. For $\arctan\left(\frac{\dot{y}}{\dot{x}}\right)$, a proper arctangent function (eg. `atan2` in Matlab) would handle division by zero.

For the controls, we start from

$$\begin{bmatrix} \omega \\ a \end{bmatrix} = \begin{bmatrix} \frac{d}{dt} \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \\ \dot{v} \end{bmatrix} \tag{15}$$

and take derivatives, since $\omega = \dot{\theta}$:

$$\begin{aligned} \omega(t) &= \frac{d}{dt} \arctan\left(\frac{\dot{y}}{\dot{x}}\right) && (16a) \\ &= \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{(\dot{x})^2 \left(1 + \left(\frac{\dot{y}}{\dot{x}}\right)^2\right)} && (16b) \\ &= \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{(\dot{x})^2 + (\dot{y})^2} && (16c) \\ &= \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{v^2} && (16d) \\ &= \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{(\dot{x})^2 + (\dot{y})^2} && (16e) \end{aligned}$$

where $\ddot{x} = 2b_{02} + 6b_{03}t$ t, and $\ddot{y} = 2b_{12} + 6b_{13}t$.

Lastly, for a, we will differentiate $v^2 = (\dot{x})^2 + (\dot{y})^2$ implicitly and isolate v. Since $a = \dot{v}$, we have the following:

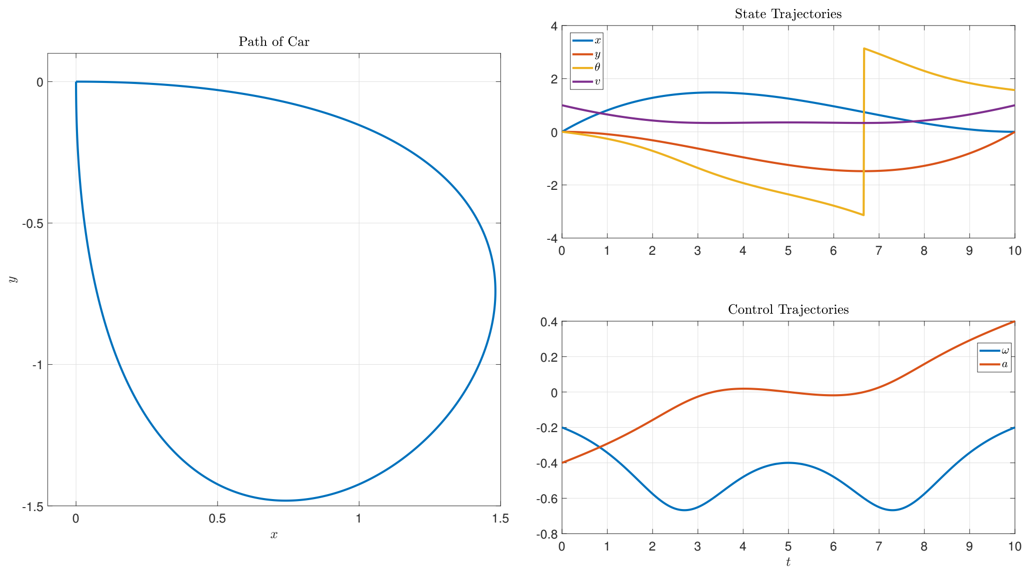

Figure 2: Visualization of feasible trajectory.

$$v^2 = (\dot{x})^2 + (\dot{y})^2 \tag{17a}$$
$$2v\dot{v} = 2\dot{x}\ddot{x} + 2\dot{y}\ddot{y} \tag{17b}$$
$$va = \dot{x}\ddot{x} + \dot{y}\ddot{y} \tag{17c}$$
$$a(t) = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{v} \tag{17d}$$
$$a(t) = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{\sqrt{(\dot{x})^2 + (\dot{y})^2}} \tag{17e}$$

Please see `diff_flat.m` for a MATLAB implementation. State and control trajectories are shown in Figure 2.

3 Multiple Shooting with Casadi

a)

$$
\begin{aligned}
\text{minimize} \quad & \int_{t=0}^{5} F^2(\tau) d\tau && \text{(18a)} \\
\text{subject to} \quad & x(0) = 0 && \text{(18b)} \\
& v(0) = 0 && \text{(18c)} \\
& \theta(0) = 0 && \text{(18d)} \\
& \omega(0) = 0 && \text{(18e)} \\
& x(5) = 0 && \text{(18f)} \\
& v(5) = 0 && \text{(18g)} \\
& \theta(5) = \pi && \text{(18h)} \\
& \omega(5) = 0 && \text{(18i)} \\
& \dot{x}(t) = v(t) && \text{(18j)} \\
& \dot{\theta}(t) = \omega(t) && \text{(18k)} \\
& \begin{bmatrix} M + m & ml\cos\theta \\ ml\cos\theta & I + ml^2 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -bv + ml\omega^2 \sin\theta + F \\ -mgl\sin\theta \end{bmatrix} && \text{(18l)} \\
& |x(t)| \leq 1 && \text{(18m)} \\
& |u(t)| \leq 0.2 && \text{(18n)}
\end{aligned}
$$

b) Let $N = 50, h = \frac{T}{N}$, and we get the following nonlinear optimization program:

$$
\begin{aligned}
\underset{\{(x_k,v_k,\theta_k,\omega_k)\}_{k=0}^{N},\{F_k\}_{k=0}^{N-1}}{\text{minimize}} \quad & h\sum_{k=0}^{N-1} F_k^2 && \text{(19a)}\\
\text{subject to} \quad & x_0 = 0 && \text{(19b)}\\
& v_0 = 0 && \text{(19c)}\\
& \theta_0 = 0 && \text{(19d)}\\
& \omega_0 = 0 && \text{(19e)}\\
& x_N = 0 && \text{(19f)}\\
& v_N = 0 && \text{(19g)}\\
& \theta_N = \pi && \text{(19h)}\\
& \omega_N = 0 && \text{(19i)}\\
& \forall k \in \{0,1,\ldots,N\}, && \text{(19j)}\\
& \quad x_{k+1} = x_k + hv_k && \text{(19k)}\\
& \quad v_{k+1} = v_k + h\dot{v}_k && \text{(19l)}\\
& \quad \theta_{k+1} = \theta_k + h\omega_k && \text{(19m)}\\
& \quad \omega_{k+1} = \omega_k + h\dot{\omega}_k && \text{(19n)}\\
& \quad \begin{bmatrix} M+m & ml\cos\theta_k \\ ml\cos\theta_k & I+ml^2 \end{bmatrix}\begin{bmatrix}\dot{v}_k \\ \dot{\omega}_k\end{bmatrix} = \begin{bmatrix} -bv_k + ml\omega_k^2\sin\theta_k + F_k \\ -mgl\sin\theta_k \end{bmatrix} && \text{(19o)}\\
& \quad |x_k| \le 1 && \text{(19p)}\\
& \quad |F_k| \le 0.2 && \text{(19q)}
\end{aligned}
$$

Note that to make implementation simpler and avoid taking matrix inverses, we introduced extra variables $\dot{v}_k$ and $\dot{\omega}_k$. The resulting extra linear equations (19o) can be written using a matrix left division operator (see line 30 in `a2_q3_sol.m`). It isn't necessary to avoid taking the inverse of such a small matrix, but this "trick" is good to keep in mind for larger problems.

c) Please see an implementation in `a2_q3_sol.m`. The state and control trajectories are shown in Figure 3.

4 Robotic Safety via Reachability Analysis

a) Assuming all distance units are in metres: $\mathcal{T} = \{(x,y,\theta,v) : \sqrt{x^2+y^2} \le 1000\}$.

b) $l(x,y,\theta,v) = \sqrt{x^2+y^2} - 1000$

c) For notation simplicity, we will simplify how we write partial derivatives with $V_x := \frac{\partial V}{\partial x}, V_y := \frac{\partial V}{\partial y}, V_\theta := \frac{\partial V}{\partial \theta}, V_v := \frac{\partial V}{\partial v}$.

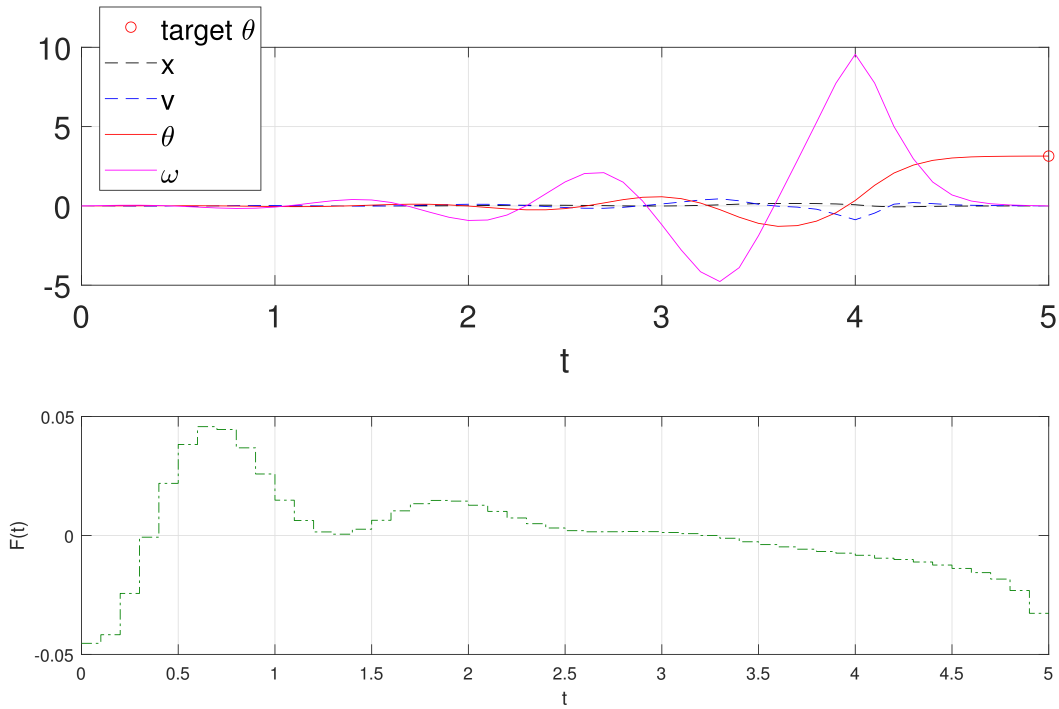

Figure 3: State and control trajectories for the cart pole problem.

For the optimal controls, we focus on terms that depend on (ω, a):

$$\frac{\partial V}{\partial z}(t,z)^\top f(z,u,d) = V_x(v\cos\theta + d_x) + V_y(v\sin\theta + d_y) + V_\theta\omega + V_v a \tag{20a}$$

$$= \text{terms independent of } (\omega, a) + V_\theta\omega + V_v a \tag{20b}$$

$$\tag{20c}$$

This allows us to analytically compute the optimal control:

$$u^*(t,x,y,\theta,v) = \arg\max_{u\in\mathcal{U}}\min_{d\in\mathcal{D}} \frac{\partial V}{\partial z}(t,z)^\top f(z,u,d) \tag{21a}$$

$$= \arg\max_{u\in\mathcal{U}}\min_{d\in\mathcal{D}} (\text{terms independent of } (\omega, a) + V_\theta\omega + V_v a) \tag{21b}$$

$$= \arg\max_{|\omega|\le 0.5, |a|\le 10} (V_\theta\omega + V_v a) \tag{21c}$$

Since ω and a are not coupled together in 21c, we can determine them separately:

$$\omega^*(t,x,y,\theta,v) = \begin{cases} -0.5, & V_\theta(t,x,y,\theta,v) < 0 \\ 0.5, & V_\theta(t,x,y,\theta,v) \geq 0 \end{cases} \tag{22a}$$
$$= 0.5\text{sign}(V_\theta(t,x,y,\theta,v)) \tag{22b}$$
$$a^*(t,x,y,\theta,v) = \begin{cases} -10, & V_v(t,x,y,\theta,v) < 0 \\ 10, & V_v(t,x,y,\theta,v) \geq 0 \end{cases} \tag{22c}$$
$$= 10\text{sign}(V_v(t,x,y,\theta,v)) \tag{22d}$$

In Eq. (22), we emphasize that the partial derivatives V_θ and V_v are functions of (t,x,y,θ,v).

For the disturbances, we focus on terms that depend on (d_x, d_y):

$$\frac{\partial V}{\partial z}(t,z)^\top f(z,u,d) = V_x(v\cos\theta + d_x) + V_y(v\sin\theta + d_y) + V_\theta\omega + V_v a \tag{23a}$$
$$= \text{terms independent of } (d_x, d_y) + V_x d_x + V_y d_y \tag{23b}$$

Note that there is no disturbance on the control in Eq. (23). This allows us to analytically compute the optimal disturbance:

$$d^*(t,x,y,\theta,v) = \arg\min_{d\in\mathcal{D}} \frac{\partial V}{\partial z}(t,z)^\top f(z,u^*,d)) \tag{24a}$$
$$= \arg\min_{d\in\mathcal{D}} (\text{terms independent of } (d_x, d_y) + V_x d_x + V_y d_y) \tag{24b}$$
$$= \arg\min_{|d_x|,|d_y|\leq 25} (V_x d_x + V_y d_y) \tag{24c}$$
$$\tag{24d}$$

Again, d_x and d_y are not coupled together in 24d, and we can determine them separately:

$$d_x^*(t,x,y,\theta,v) = \begin{cases} 25, & V_x(t,x,y,\theta,v) < 0 \\ -25, & V_x(t,x,y,\theta,v) \geq 0 \end{cases} \tag{25a}$$
$$= -25\text{sign}(V_x(t,x,y,\theta,v)) \tag{25b}$$
$$d_y^*(t,x,y,\theta,v) = \begin{cases} 25, & V_y(t,x,y,\theta,v) < 0 \\ -25, & V_y(t,x,y,\theta,v) \geq 0 \end{cases} \tag{25c}$$
$$= -25\text{sign}(V_y(t,x,y,\theta,v)) \tag{25d}$$

Note that in Eq. (25), the optimal disturbance actually takes the opposite sign of the corresponding partial derivatives, since we're *minimizing* the pre-Hamiltonian.

d) Please see `a2_q4.py` for the coding parts of this question.

- The dynamical system is defined by the class `DubinsCar4D`.

- Control and disturbance bounds are specified by the `uMin`, `uMax`, `dMin`, `dMax` properties. These should be set according to what is being asked in the question. In this part, `dMin` and `dMax` should be `[0,0]`
- The properties `uMode` and `dMode` specify whether the control and disturbance are trying to maximize or minimize the objective. In this problem, the control is maximizing, and disturbance minimizing, so the correct settings are `uMode="max", dMode="min"`.
- The `Grid` function specifies the grid; make sure the number of grid points follows the suggestion in the assignment, and the bounds are chosen to contain the entire backward reachable tube, and no part of it is outside of the grid.
- The set representing the no-fly zone can be set using the `CylinderShape` function.
- The variable `lookback_length` specifies the value of T. It should be chosen to be at least 4.
- The backward reachable tube for this part is shown in Figure 4.

e) The sets representing speed limits, which are $\{(x, y, \theta, v) : v \leq 80\}$ and $\{(x, y, \theta, v) : v \geq 300\}$ can be set using the `Lower_Half_Space` and `Upper_Half_Space` functions. Taking the minimum of these functions and the function representing the no-fly zone is equivalent to taking the union, and allows one to build up a target set that encodes a union of multiple danger regions.

Results are shown in Figure 5. Adding the additional constraints involving speed enlarges the backward reachable tube slightly, since the aircraft becomes less capable of maneuvering around the no-fly zone.

f) In this part, we set the properties `dMin=[-25,-25], dMax=[25,25]` in the `DubinsCar4D` object.

Results are shown in Figure 6. Adding disturbances further enlarges the backward reachable tube, again since the aircraft becomes less capable of maneuvering around the no-fly zone.

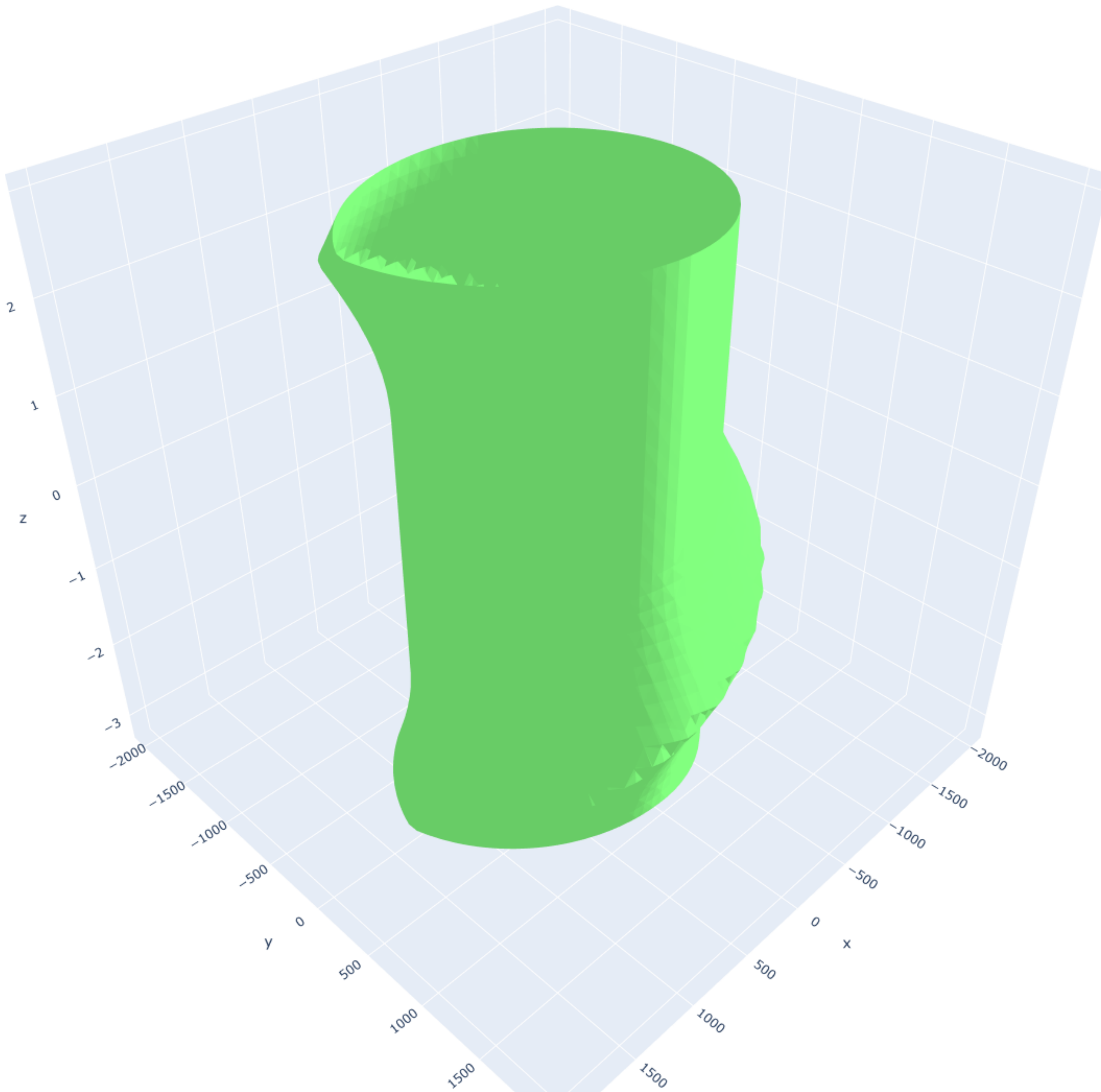

Figure 4: Backward reachable tube from the no-fly zone.

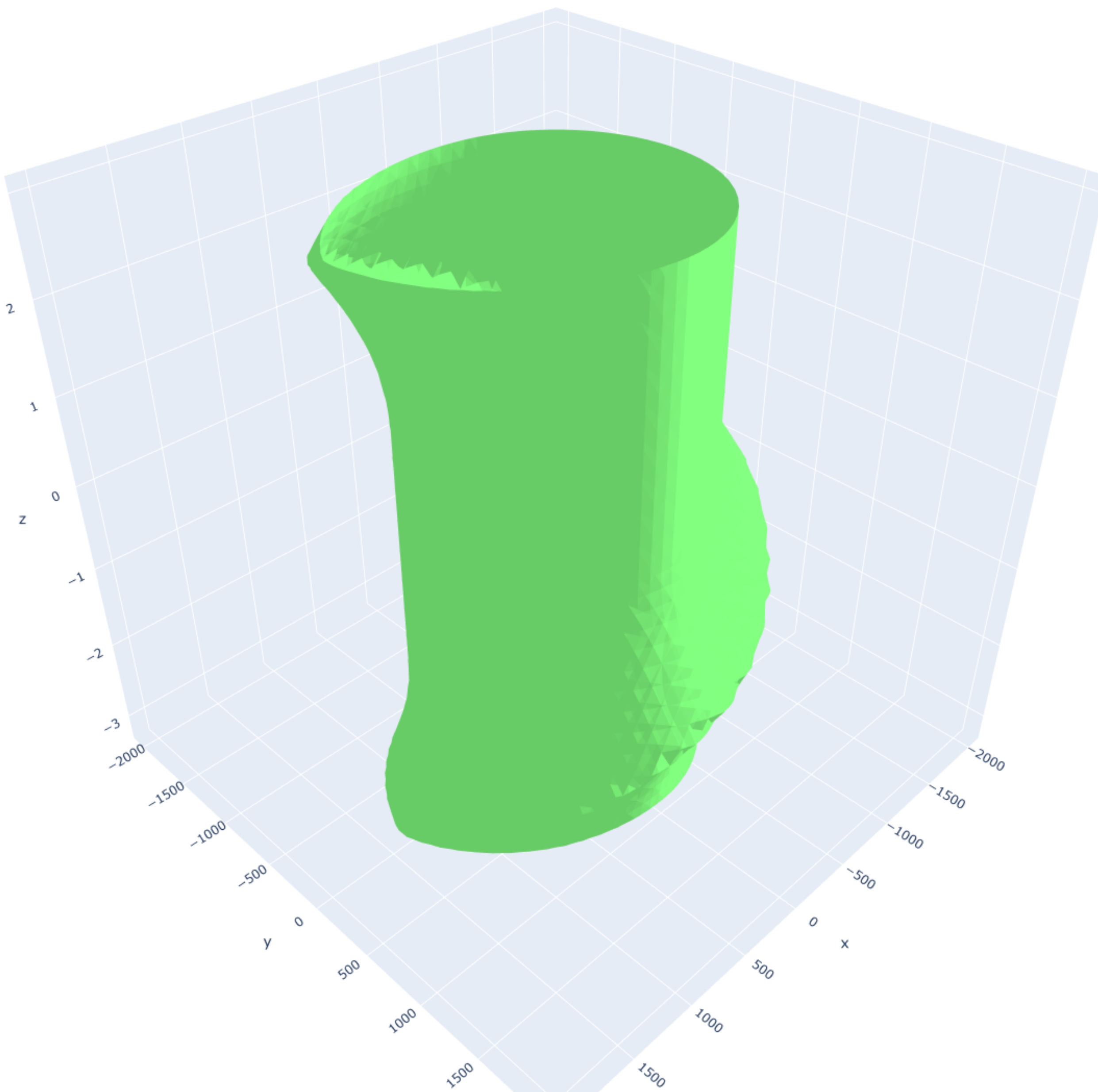

Figure 5: Backward reachable tube from the union of no-fly zone, violation of maximum speed limit, and violation of minimum speed limit.

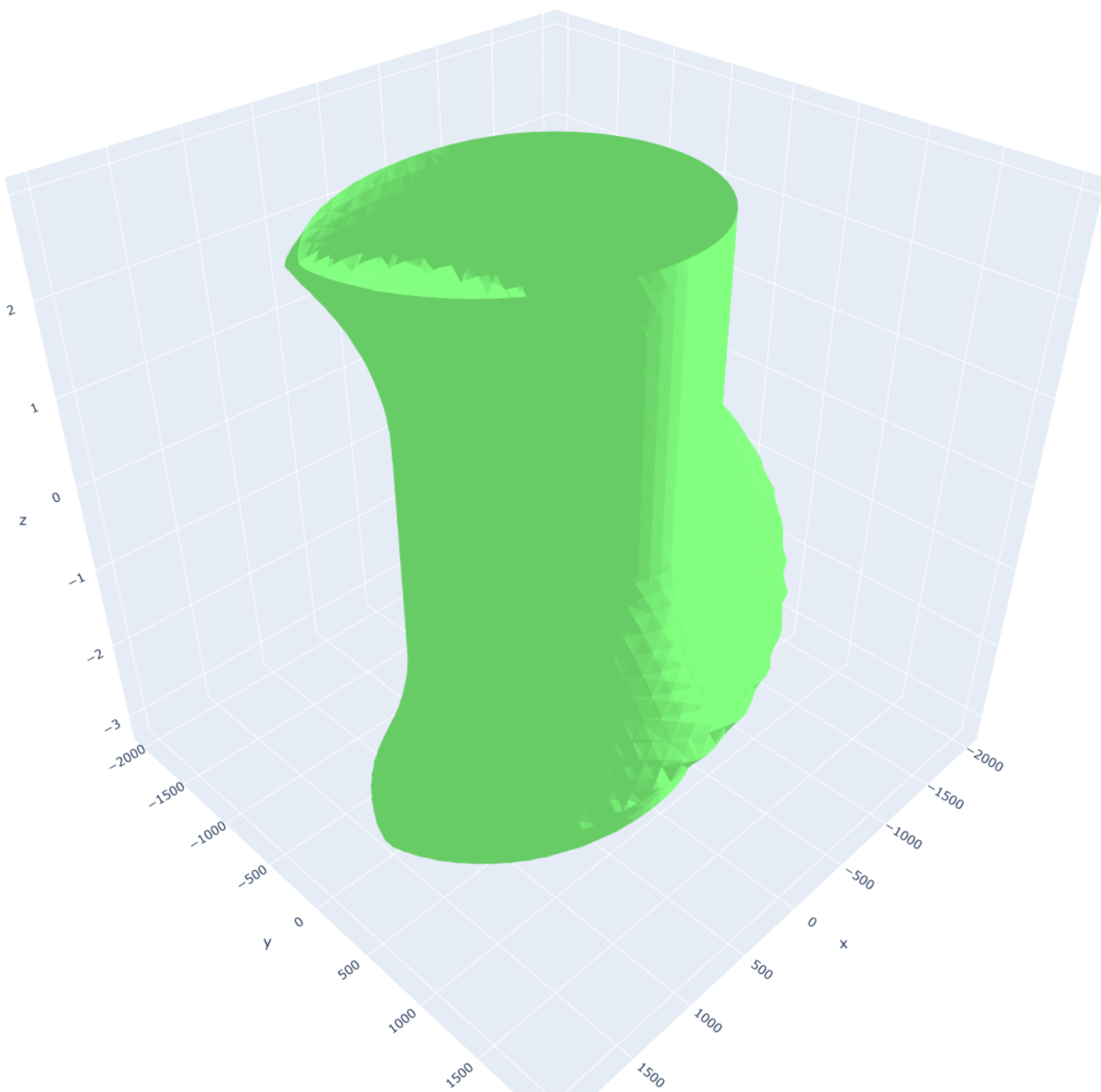

Figure 6: Backward reachable tube that accounts for disturbances.