Optimal Control of a Mobile Manipulator with 3D Obstacle Avoidance

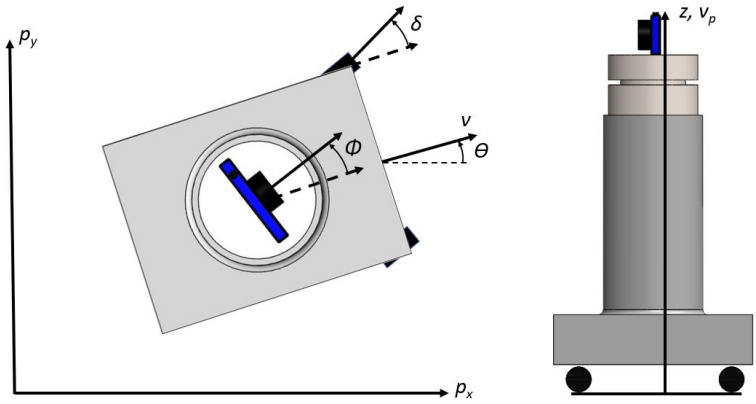
Spencer Powers and Trevor Schwehr

December 22, 2021

1 Problem Formulation

This project focuses on optimal control of a mobile manipulator with a car-like base and an attached robotic arm. The arm includes a vertical prismatic joint and a revolute joint attached to the top of the base as shown in Figure 1a, resulting in a total of four degrees of freedom for the system. A tool can then be attached to the revolute joint at the end of the prismatic joint. The following analysis assumes that the tool is mounted on the top and is relatively small, and thus only the height of the tool above the robot is accounted for during tracking. One such real-life configuration is where the tool is a small camera or other sensing device.

(a) Mobile manipulator with a camera as the tool.



(b) State description of mobile manipulator. See Section 2 for complete descriptions of state and control variables.

Figure 1: Mobile Manipulator

This configuration allows for simulation of a variety of real-world problems. For example, the mobile manipulator could be used in a medical setting to provide a safer means of access to diagnose high risk or infectious patients. In an industrial setting, this mobile manipulator could be used to safely inspect collapsed mining tunnels or other environments that are too dangerous for workers.

The primary goal of this project was to optimize the path between a given initial state and a desired final state using minimum control effort while satisfying state and control constraints and avoiding multiple 3D obstacles of different shapes, sizes, and positions. In addition, this project investigated the effects of varying the control penalties in the running cost on the optimal path chosen by the solver. Specifically, we studied the optimal paths for a more mobile base versus a more mobile arm.

2 Methods

The state and control variables are summarized in the table below alongside the system dynamics:

Symbol	Quantity	Limits
p_x	x-position of the car	$[-10, 10]$
p_y	y-position of the car	$[-10, 10]$
z	arm height	$[1, 3.5]$
θ	car heading	$[-\pi, \pi]$
v	car speed	$[-5, 5]$
δ	car steering rate	$[-\pi, \pi]$
v_p	arm prismatic joint velocity	$[-1, 1]$
ϕ	arm revolute joint angle	$[-\pi, \pi]$

$$\begin{bmatrix} \dot{p_x} \\ \dot{p_y} \\ \dot{z} \\ \dot{\theta} \\ \dot{v} \\ \dot{\delta} \\ \dot{v_p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ v_p \\ \delta \\ u_a \\ u_\delta \\ u_z \\ u_\phi \end{bmatrix}$$

Differential dynamic programming (DDP) was leveraged to determine the optimal trajectory and control sequence because it is an efficient and versatile solver that handles obstacles, control constraints, and state constraints. The initial and final states of the mobile manipulator are specified, as is the time horizon, control constraints, and a variety of obstacles in 3D space. This project supports overhead hemispheres, pillars through the X-Y plane, and pillars through the X-Z plane.

The cost function takes the following form:

$$J = \frac{1}{2}x(t_f)^T P_f x(t_f) + \int_{t_0}^{t_f} \frac{1}{2}\left(x^t Q x + u^T R u\right) \tag{1}$$

In this project, the running state cost matrix, Q, is set to 0 and the terminal cost weights, P_f, were typically stronger than the running cost weights on the controls, R, although the weights of R were varied to understand the effects of different weightings on the optimal path. Specifically, we study the effect of increasing the penalty on the control of the prismatic joint relative to those of the mobile base to encourage diverse behavior when encountering overhead obstacles. To incentivize obstacle avoidance, a steep quadratic cost term was added whenever the base or the arm tip collided with an obstacle. While the base was treated as a point in this analysis, the next simplest model treats the car as a circle or a set of circles, which is effectively equivalent to increasing the radii of the planar obstacles without changing the rest of the codebase.

3 Results and Discussion

3.1 Test Scene 1: Two Obstacle Types

3.1.1 Uniform Control Costs

One test environment is shown in an isometric view and an overhead view in Figures 2 and 3, respectively. A vertical column centered at $[-2, -3]^T$ with radius 1 and an overhead hemisphere attached to the ceiling with center $[0, -2.5, 3.5]^T$ and radius 1 obstruct the path between the start and desired configurations. The starting states are $[-5, -5, 2.8, 0, 0, 0, 0, \pi]^T$, the desired end states are $[4, 0, 3, 0, 0, 0, 0, \pi/2]^T$, and the time horizon is 40 seconds.

The candidate trajectories shown in Figures 2 and 3 represent the case where all controls are penalized equally in the running cost, i.e. R has identical diagonal elements of 0.5. Note that the candidate trajectories are in blue and the optimal trajectory is in green. Dashed lines represent ground tracks of the 3D trajectories.

Due to the fact that the controls are uniformly penalized and the terminal cost on the final state error is high, the optimal trajectory is comprised of the optimal X-Y path around the pillar and ducking under the hemisphere along the route. This is visible in the overhead view (Figure 3) and also in the plot of arm height over time (Figure 4). The non-Cartesian state variables and controls for the optimal trajectory are shown in Figures 5 and 6, respectively. Because this is a minimum control effort problem, it is not expected that the controls will be maxing themselves out over the course of the trajectory, and Figure 6 confirms this expected behavior. Note that the u_ϕ control is

a constant because it is moving the revolute joint to its desired state over the time horizon, but since the tool is assumed to be relatively small, it does not impact the collision avoidance part of the optimization problem.

Finally, the cost of each candidate trajectory is plotted with the distance of closest approach to both obstacles in Figure 7, where the vertical line represents the radius of the obstacle. As expected, the lowest cost is associated with just grazing both obstacles as this is the most direct route. The optimal cost and error metrics are summarized in Figure 8.

3.1.2 Non-Uniform Control Costs

Consider the same scene and conditions, but now the penalty for the z-axis acceleration of the prismatic joint is 20 times higher than the other controls in the running cost. It is expected that the robot will want to avoid the overhead obstacle by navigating around it in the X-Y plane instead of expending Z-control to duck under it. This behavior is demonstrated in the overhead view of the scene and the plot of arm height over time in Figures 10 and 11, respectively. Note that while the arm avoids expending control effort to duck under the overhead obstacle, it still has to go from its starting height of $z = 2.8$ to $z = 3$ over the duration of the trajectory, so it grazes the hemisphere on the way up to its final position.

The non-Cartesian states and controls for this problem are shown in Figures 12 and 13, respectively, while the costs of each candidate trajectory as a function of distance of closest approach to each obstacle and iteration count are shown in Figure 14. Note that as expected, the lowest cost trajectory is further from the pillar obstacle but still tangential to the overhead hemisphere. The optimal cost and error metrics are summarized in Figure 15.

3.2 Test Scene 2: Three Obstacle Types

A more complex test environment is shown in an isometric view and an overhead view in Figures 16 and 17 respectively. This differs from Scene 1 because it additionally has an overhead pillar in the X-Z plane that the mobile manipulator consequently cannot navigate around in the X-Y plane. The vertical pillar has the same parameters, the overhead pillar has constant X and Z center coordinates of 0 and 3.5 respectively with radius 1, and the overhead hemisphere is now centered at $[3, 0, 3.5]^T$ with radius 1.5. The initial states are the same as in Scene 1, and the final states are now $[6, 1, 3, 0, 0, 0, 0, \pi/2]^T$. The time horizon was increased to 60 seconds to account for the longer Cartesian distance between start and end states.

The same behavioral analysis was performed in this more complex scenario, namely observing whether the optimal trajectory chosen by the solver followed the behavior expected by intuition. When the control costs were uniformly set to 0.5, the solver indeed chose the optimal trajectory to be comprised of the optimal X-Y path around planar obstacles while maneuvering below overhead obstacles when necessary on the path to the final states. The candidate trajectories, non-Cartesian states, controls and costs are shown in Figures 16, 17, 18, 19, 20, and 21. When the u_z control is 20 times higher than the other controls in the running cost, the optimal trajectory was still forced to duck under the overhead pillar, but intuitively chose to swing out wide early in the trajectory to give it a longer route before encountering each consecutive obstacle. As a result, this extra distance between obstacles lets the arm keep its vertical acceleration as small as possible. The candidate trajectories, non-Cartesian states, controls and costs are shown in Figures 22, 23, 24, 25, 26, and 27.

4 Conclusion and Future Work

The simulation results from these two test environments indicate that the DDP implementation yields intuitive solver behavior in complex scenarios. When tasked with solving the three-dimensional optimal control problem for the mobile manipulator shown in Figure 1a under different cost function parameter configurations, the solver adjusts its selection of the optimal trajectory in understandable ways. A uniform penalty on controls in the running cost and a high terminal cost yields trajectories that are comprised of a minimum control effort X-Y ground track and minimum effort Z-control to avoid overhead obstacles. On the other hand, drastically increasing the penalty on Z-control in the running cost relative to the other controls while keeping a dominant terminal cost necessarily affects the X-Y behavior along with the vertical behavior; to keep Z-control effort as low as possible, the mobile base now diverges from the optimal X-Y ground track to help the arm avoid the overhead obstacle at the cost of additional control effort.

Future work includes considering analysis for a tool with significant length which would thus impact the collision avoidance optimization. Initial work to implement this feature was completed after the presentation but the additional functionality could not be completed in time for submission and is thus included in future work.

A Figures

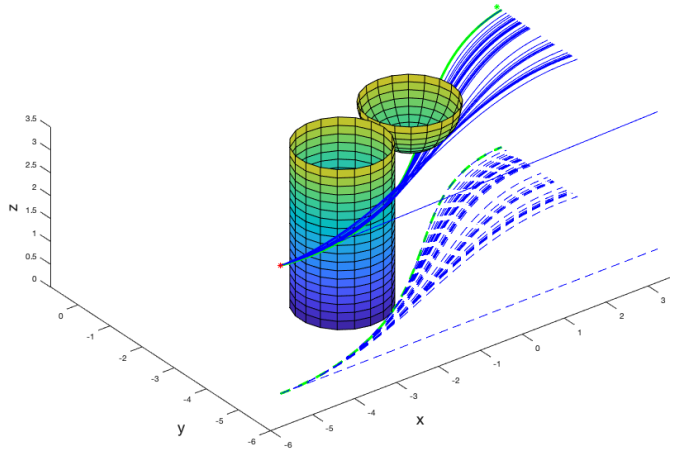


Figure 2: Scene 1 isometric view. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

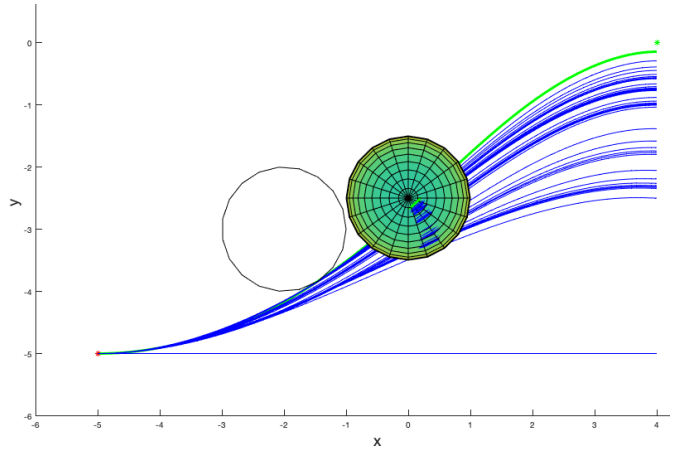


Figure 3: Scene 1 overhead view. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

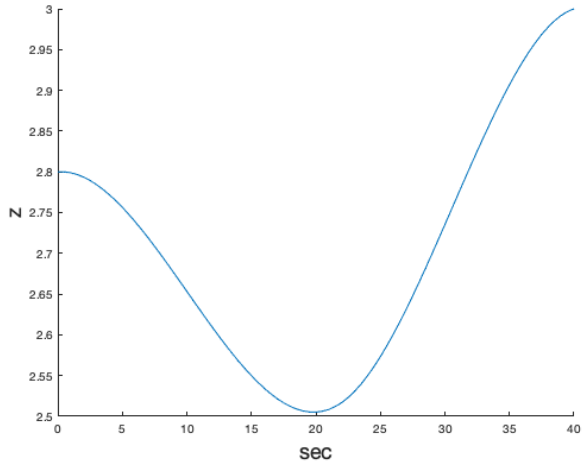


Figure 4: Scene 1 arm height over time.

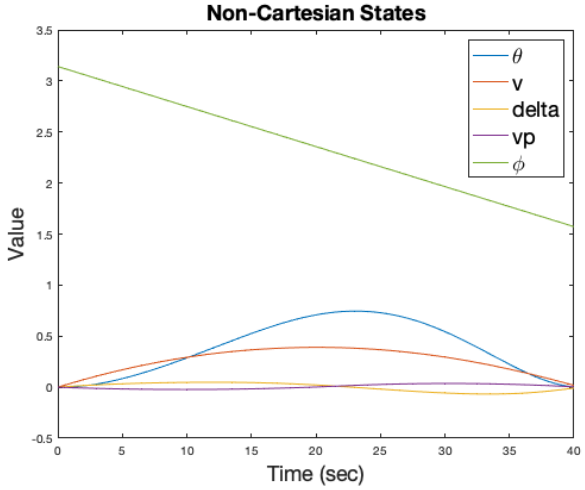


Figure 5: Scene 1 non-Cartesian states for the optimal trajectory.

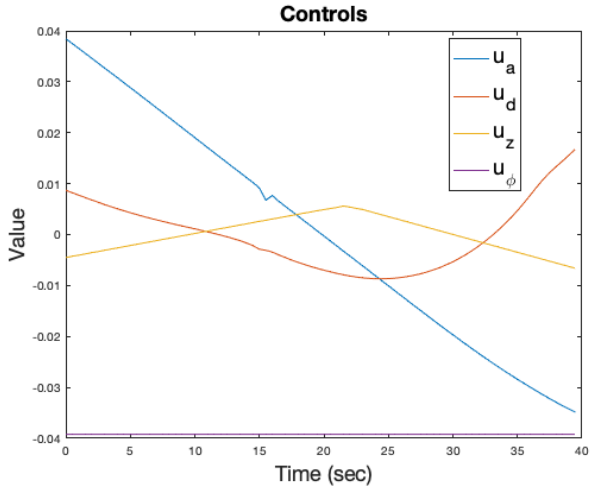


Figure 6: Scene 1 controls for the optimal trajectory.

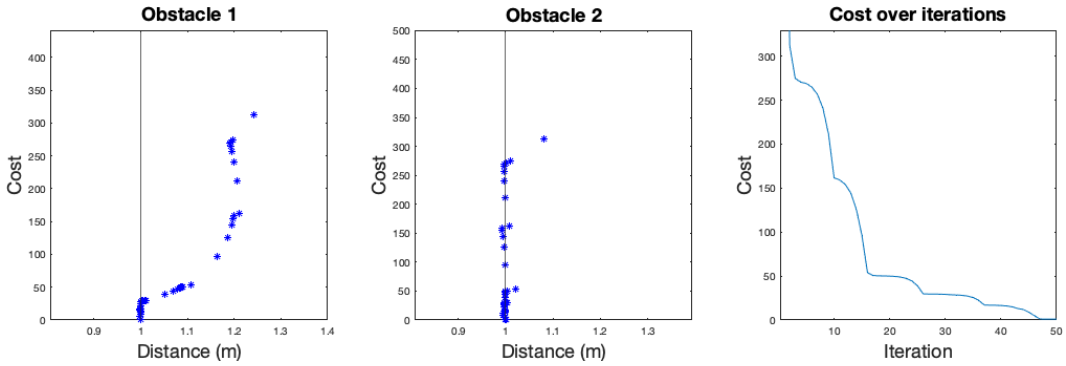


Figure 7: Scene 1 costs for each candidate trajectory plotted against the distance of closest approach to each obstacle, as well as against iteration count.

Metric	Optimal Value
J	1.1140
$\|\|\mathbf{x_f} - \mathbf{x_d}\|\|$	0.1463 m
$\phi_f - \phi_d$	0.0020 rad

Figure 8: Scene 1 optimal cost and error metrics.

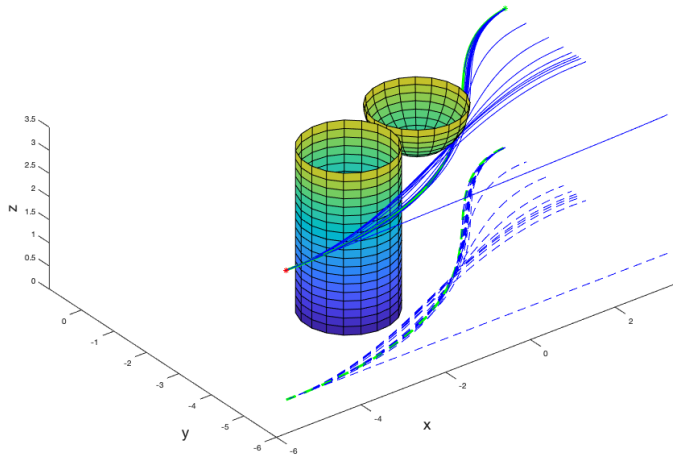


Figure 9: Scene 1 isometric view, but with non-uniform control penalties. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

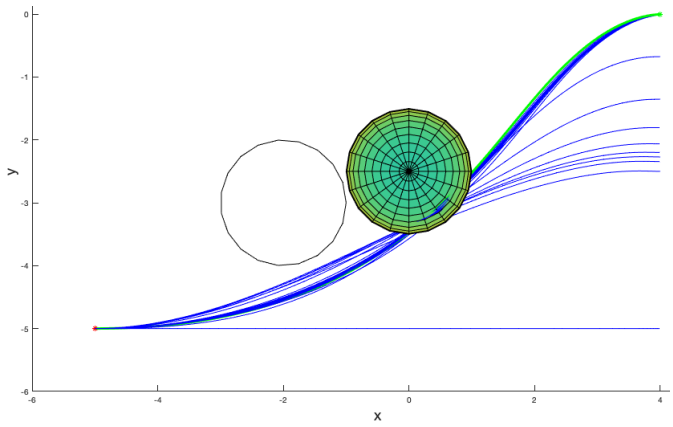


Figure 10: Scene 1 overhead view, but with non-uniform control penalties. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

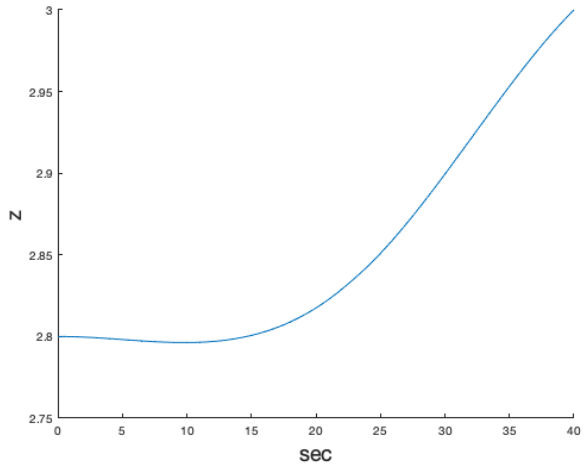


Figure 11: Scene 1 arm height over time, but with non-uniform control penalties.

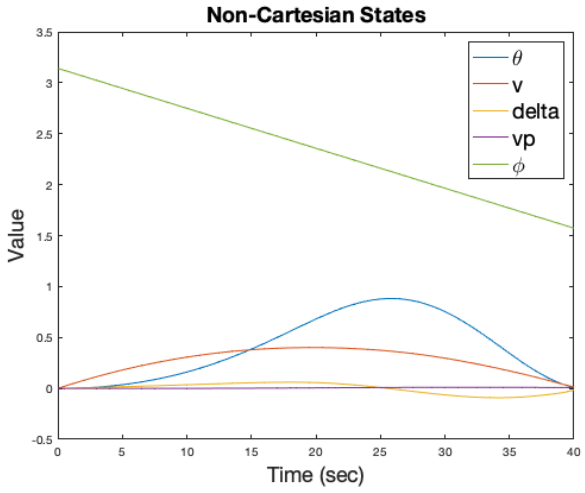


Figure 12: Scene 1 non-Cartesian states for the optimal trajectory, but with non-uniform control penalties.

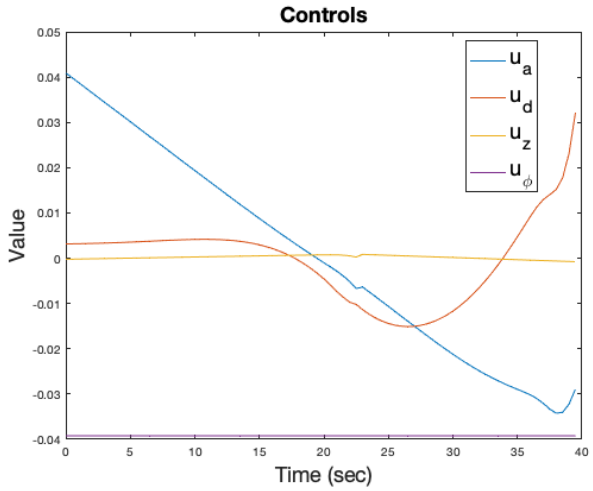


Figure 13: Scene 1 controls for the optimal trajectory, but with non-uniform control penalties.

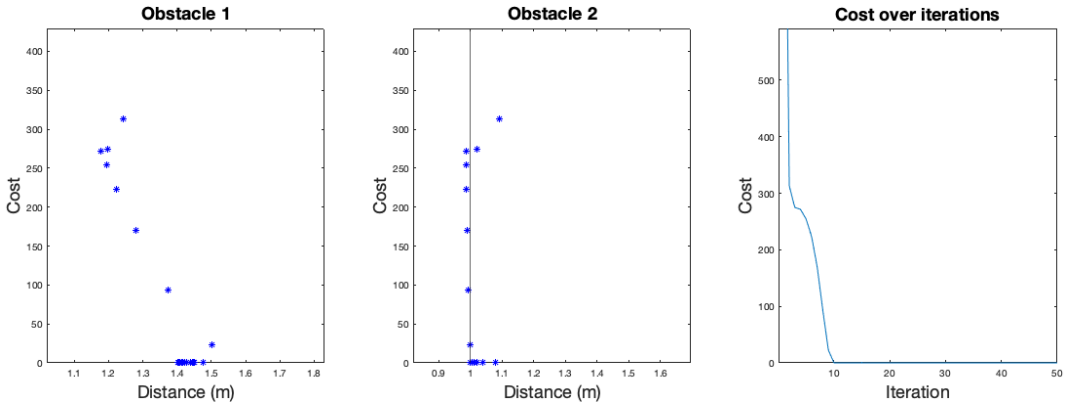


Figure 14: Scene 1 costs for each candidate trajectory plotted against the distance of closest approach to each obstacle, as well as against iteration count. However, the control penalties are non-uniform.

Metric	Optimal Value
J	0.0221
$\|\|\mathbf{x_f} - \mathbf{x_d}\|\|$	0.0026 m
$\phi_f - \phi_d$	0.0020 rad

Figure 15: Scene 1 optimal cost and error metrics, but with non-uniform control penalties.

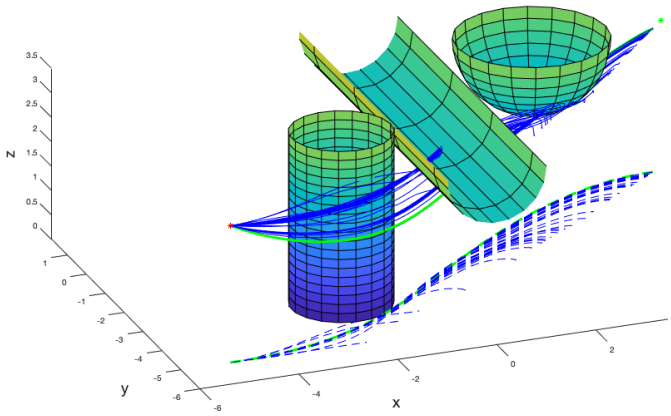


Figure 16: Scene 2 isometric view. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

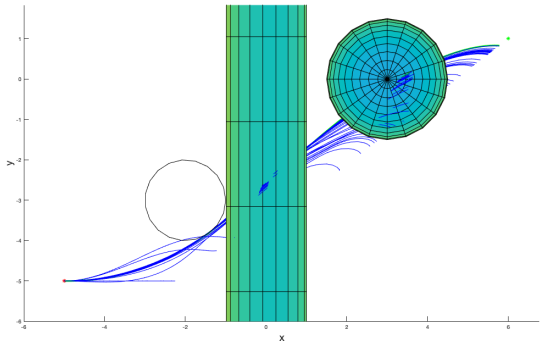


Figure 17: Scene 2 overhead view. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

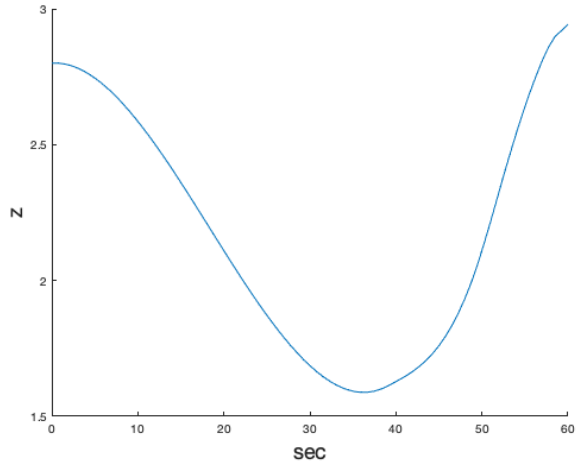


Figure 18: Scene 2 arm height over time.

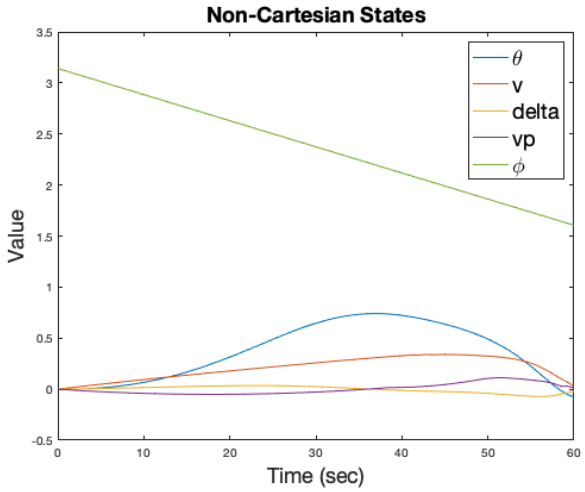


Figure 19: Scene 2 non-Cartesian states for the optimal trajectory.

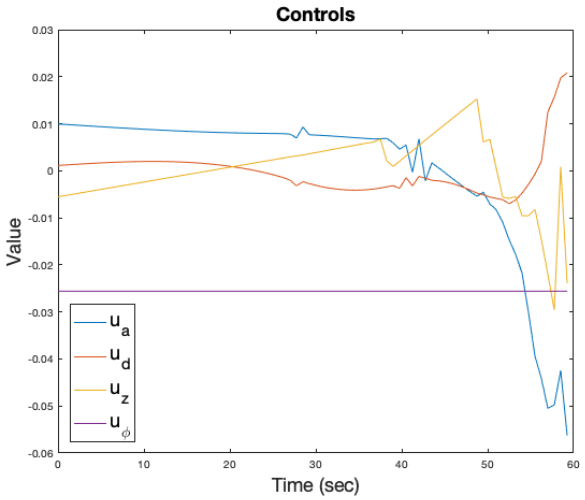


Figure 20: Scene 2 controls for the optimal trajectory.

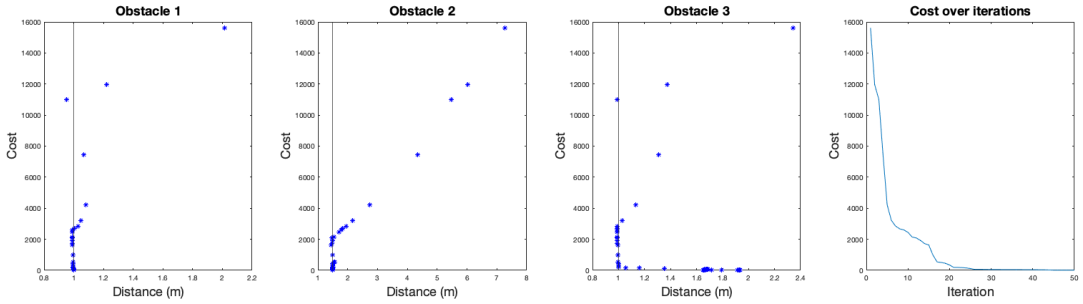


Figure 21: Scene 2 costs for each candidate trajectory plotted against the distance of closest approach to each obstacle, as well as against iteration count.

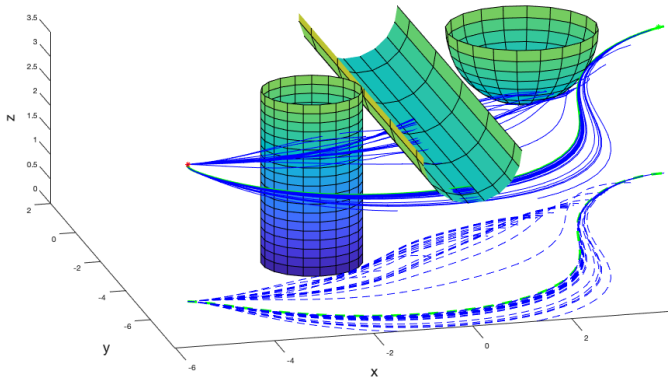


Figure 22: Scene 2 isometric view, but with non-uniform control penalties. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

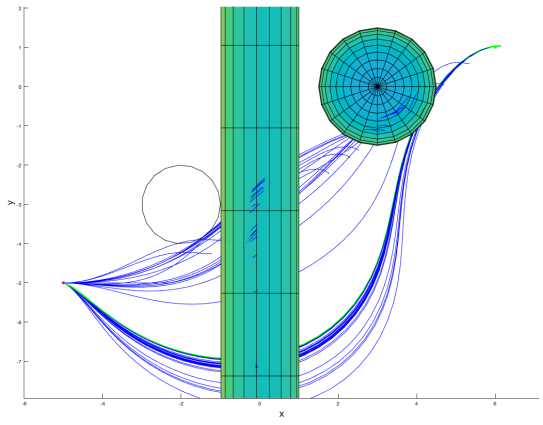


Figure 23: Scene 2 overhead view, but with non-uniform control penalties. Candidate trajectories are shown in blue and the optimal trajectory is shown in green.

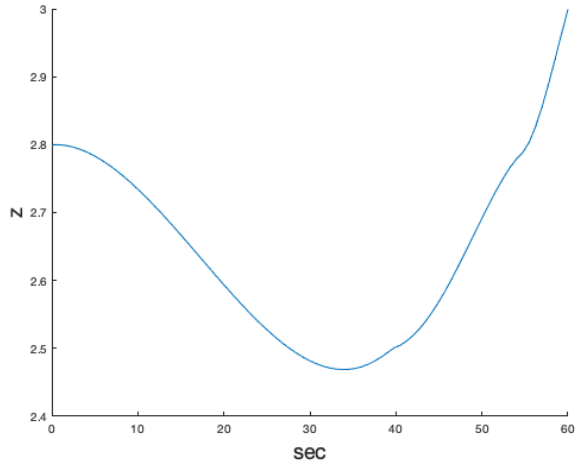


Figure 24: Scene 2 arm height over time, but with non-uniform control penalties.

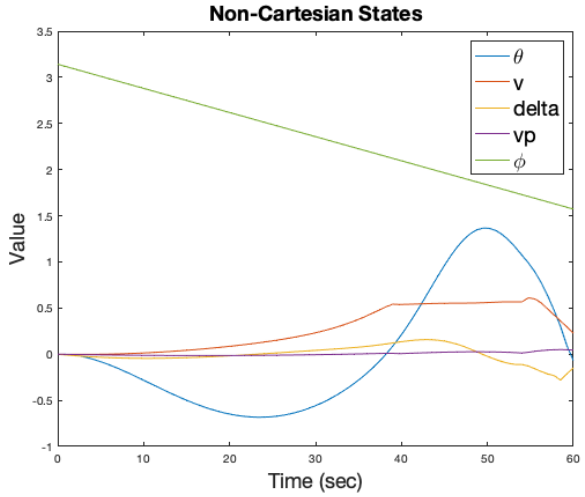


Figure 25: Scene 2 non-Cartesian states for the optimal trajectory, but with non-uniform control penalties.

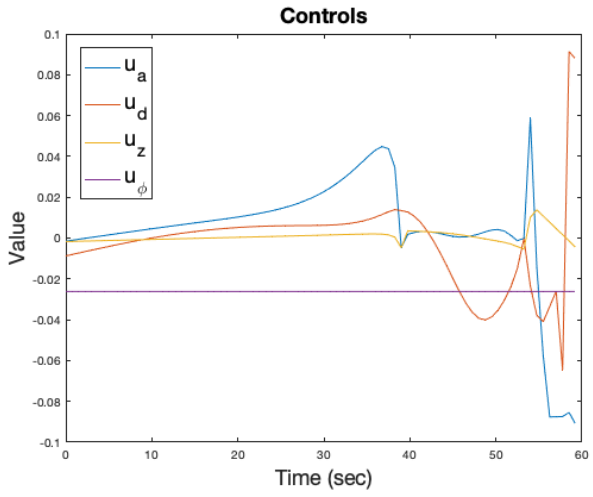


Figure 26: Scene 2 controls for the optimal trajectory, but with non-uniform control penalties.

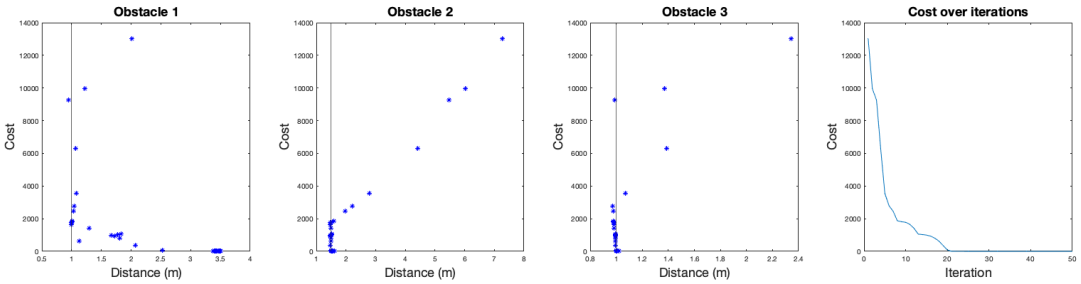


Figure 27: Scene 2 costs for each candidate trajectory plotted against the distance of closest approach to each obstacle, as well as against iteration count. However, the control penalties are non-uniform.