Optimal Planning for Off-Road Environments

Shai Bernard (shaibern@stanford.edu)

Dylan Goff (dgoff@stanford.edu)

1 Motivation

Recent advances in fields including computer vision, artificial intelligence, and integrated satellite navigation technology have made the prospect of off-road autonomous systems probable in the relatively near future. These systems hold the promise of revolutionizing areas such as delivery, transportation, and emergency services in regions lacking vehicular infrastructure without forcing a human driver to be present in the vehicle in potentially hazardous areas. However, while satellite technologies have rendered information describing surface topography and large physical obstacles readily available, there is a growing need for these autonomous systems to be able to utilize this information to calculate paths across complex terrain that optimize certain quantities of interest. In this project, we aim to demonstrate methods by which these paths can be optimized with respect to minimizing quantities such as path distance and energy usage while avoiding known obstacles in a varied surface topography.

2 Literature Review

2.1 General Path Planning

Several different approaches for optimal path generation have been explored in prior work. A variety of graph-based algorithms exist–one of which being A^*, which is guaranteed to generate an optimal path (with respect to minimizing distance) but suffers from high computational complexity and is thus not always suitable for high-dimensional problems [9]. Furthermore, while it produces an optimal path, A^* and other graph-based methods fail to take into account any physical properties (i.e. kinematics, dynamics) of the system and can only optimize for the physical distance between two points, making them of limited use if we want to optimize for other metrics such as control effort, energy usage, or passenger comfort, or if we want to model a dynamically evolving environment [9]. Population-based methods such as genetic algorithms (and more broadly, evolutionary algorithms) are capable of multi-objective trajectory optimization with potentially lower computational costs, but do not offer any guarantees of convergence and do not take kinematic or dynamic considerations into account [9], [7]. Other methods, including sampling-based methods such as the rapidly-exploring random tree (RRT) algorithm and the artificial potential field (APF) method, have also been applied to the optimal trajectory problem, but neither are able to guarantee optimal paths [9]. A variant of RRT called RRT* that can guarantee asymptotic convergence to the optimal path has also been developed, but the computational complexity associated with the sampling necessary for RRT* makes it infeasible in practice [8]. Unlike RRT or A^*, APF is capable of avoiding dynamic obstacles [3]. As such, recent work has attempted to modify the potential fields used for APF to produce optimal paths with low computational complexity similar to A^* while maintaining the dynamic obstacle avoidance capabilities of APF [4]. Work has also been done to merge A^* with APF to form a hybrid algorithm that combines the benefits of the two [3]. Each of these studies were successful in reducing the path length generated by APF while still maintaining dynamic obstacle avoidance. Optimal control methods such as model predictive control (MPC) are capable of generating optimal paths that take into account vehicle dynamics [10]. However, one issue with MPC as applied to path planning is that programming obstacles as constraints can sometimes lead to infeasible solutions [15]. To mitigate this, work has been done to combine APF and MPC by using MPC to solve the problem while modeling obstacles with a potential field [10]. Since MPC can also be computationally time consuming, population-based methods such as particle swarm optimization (PSO) and genetic algorithms have also been integrated into it to help improve computation time [14]

2.2 Multi-Objective Optimization for Path Planning

Many of the previously discussed path planning algorithms are incapable of multi-objective optimization. Of the algorithms presented, the ones most well suited to multi-objective optimization are population-based methods and MPC. Though work has also been done to modify A^* to generate energy-efficient routes, the method explored relied on accurate data from real-time traffic information [5], which may be unavailable in an off-road scenario. PSO is one population-based method that has been used for energy efficient car route generation. Prior research has investigated the use of PSO in generating energy-efficient trajectories for electric vehicles, and it was found that energy efficiency increased by more than 9.2% compared to the routes planned by services such as Google Maps [1]. Furthermore, it has been shown that PSO can be applied to energy efficient off-road path planning [6]. MPC has also been applied to energy efficient route planning, but the existing literature that we could find assumes on-road driving. It has been shown that MPC can increase the energy savings for on-road driving routes [13], and it is reasonable that this could be extended to an off-road scenario.

3 Technical Contributions

A large body of prior work exists for energy efficient route planning and several methods, including MPC and PSO, have been applied to this problem. While PSO has seen use in both the on-road and off-road case, the application of MPC to off-road energy optimal path planning appears mostly unexplored. As such, we investigate the efficacy of MPC in planning energy optimal paths in an off-road environment.

4 Project Setting

4.1 Mathematical Model

4.1.1 Overview

Path identification and optimization is critical in the transportation of goods and individuals, and the means of doing this without constant human intervention are of increasing relevance as autonomous vehicles become realizable. However, even for off-road vehicles (as we are considering here), such calculations are heavily complicated by the effects of terrain and the presence of obstacles that must be avoided. This complexity is retained even when the relevant information is known beforehand, as is often the case for terrain and large persistent obstacles, given the availability of satellite and survey data. Two frequent quantities of concern in transportation problems are transit distance and energy expenditure. Consequently, we consider how these quantities can be minimized through the calculation of an optimal path through a space of points that each have an associated elevation and viability (i.e., if a point lies within or on an obstacle, then that point is not viable).

4.1.2 The Dynamics and Terrain

We employ a simplified model of vehicle dynamics for the purposes of focusing on path optimization techniques in complex environments and due to the high degree of variation between different motors, transmission systems, vehicle loading, and environmental conditions. Specifically, we will assume that the energy expended by the vehicle per unit distance E can be expressed as

$$E = f_1 f_2 E_{ref}$$

where E_{ref} is the a normalized reference energy expenditure level (corresponding to a speed of 5 mph on a 0 degree incline), f_1 is a unit-less function of the vehicle speed that is normalized to be 1 at 15 mph, and f_2 is a unit-less function of the incline or decline over which the vehicle is driving that is normalized to be 1 at a 0 degree incline. The factors E_{ref} and f_1 are taken from publicly available data for a Tesla Roadster [12], and f_2 is taken from available data from the US Department of Energy for a Battery Electric Vehicle (BEV) Sedan [2]. For simplicity, we assume that these data can be applied to the generic electric vehicle that we are considering in our problem. Given this limited data, we also assume that the scaling factors f_1 and f_2 are independent and can be multiplied to acquire a net efficiency. We fit piecewise interpolation functions to sections of each curve following normalization.

We model the terrain as an elevation z that is a function of the x and y coordinates, i.e., $z = h(x, y)$. This is an approximation of terrain in the real world, wherein elevation is a function of longitude and latitude (which become more similar to the Euclidean coordinates x and y in our setup as the region of interest becomes small in area and in latitudinal and longitudinal spread relative to the surface of the Earth). In particular, $h(x, y)$ takes the form of a summation of 2-dimensional Gaussian distributions. This form for $h(x, y)$ is intended to provide a readily-computable approximation of a hilly landscape. This landscape will also sometimes host an obstacle (i.e. a representation of a lake or fenced-off area), which we model as a circular region within (x,y) space through which the vehicle cannot traverse.

As for the vehicle itself, we assume a modified "Dubin's Path" setup [11], wherein our state $s := [x, y, \theta]$ is governed by

$$\dot{x} = v cos(\theta) cos(\beta)$$

$$\dot{y} = v sin(\theta) cos(\beta)$$

where x and y are two orthogonal directions (in miles) perpendicular to the direction of gravity (we assume Earth's curvature can be neglected for the space considered), θ is the angle (in radians) between the vehicle's heading and the x-direction, v is the vehicle's speed (a control input), and $\dot{\theta}$ is the angular velocity of the car's heading in the anticlockwise sense when viewing the car from above (in rad/hr - our second control input). This speed is constrained to be between -50.0 mph and 50.0 mph. Note that while we have assumed that the terrain is sufficiently smooth and stable to allow for such a speed, real-world off-road electric autonomous vehicles will necessarily be significantly slower than these bounds due to terrain conditions, abundant obstacles, and potential animals and pedestrians. The angle β is the pitch of the vehicle and can be computed from the (x,y) coordinates and the heading θ.

4.1.3 The Minimization Problem

We have formulated the constrained optimization problem as follows:

$$\min_{u,\dot{\theta}}((s_N-s_{goal})^T P_\infty (s_N-s_{goal})+\sum_{n=0}^{N-1}\alpha((x_k-x_{goal})^2+(y_k-y_{goal})^2+(z_k-z_{goal})^2)+(1-\alpha)E_k+u_k^T R u_k$$

subject to

$$\dot{x} = ucos(\theta)cos(\beta) \qquad \dot{y} = usin(\theta)cos(\beta)$$

$$\beta = \arctan(u \cdot \nabla f) = \arctan(\frac{\partial f}{\partial x}\cos\theta + \frac{\partial f}{\partial y}\sin\theta)$$

$$-50 \leq v \leq 50 \; [mph]$$
$$-10 \leq u \leq 10 \; [rad/hr]$$
$$x_0 = y_0 = 1 \text{ [mi]}$$
$$x_N = y_N = 9 \text{ [mi]}$$
$$\theta_0 = \pi/4 \text{ [rad]}$$

$$\sqrt{(x_k - x_c)^2 + (y_k - y_c)^2} - r_c \geq 0$$

where

$$E_k = f_1 f_2 E_{ref}$$
$$\text{for} \quad \text{k = 1,...,N}$$

and $\alpha \in [0, 1]$ is a user-defined parameter. Note again that f_1 and f_2 are functions of vehicle speed and inclination, respectively.

Intuitively, the term $\alpha((x_k - x_{goal})^2 + (y_k - y_{goal})^2 + (z_k - z_{goal})^2)$ penalizes the total distance taken by the vehicle, where Q is a state penalty matrix. The term $(1 - \alpha)\sum_{k=0}^{N-1} E_k d_k$ penalizes the amount of energy used (recall that E_k is a measure of normalized energy expended per unit distance and not energy itself). The terminal cost on a given horizon $(s_N - s_{goal})^T P_\infty (s_N - s_{goal})$ penalizes the distance of the final position to the desired destination at (x,y) = (9,9). The matrix P_∞ is obtained via the Lyapunov equation. However, as we do not care what the vehicle's final heading is, the last row of P_∞ is then set to 0 to avoid a terminal cost on the vehicle's orientation. A value of $\alpha = 0.0$ corresponds to an optimization problem wherein we are only concerned with minimizing the energy usage, whereas a value of $\alpha = 1.0$ corresponds to a problem in which we seek to minimize the transit distance irrespective of energy usage. An intermediate value of α corresponds to a problem in which we seek to minimize both time and energy expenditure, with the relative prioritization of these two objectives determined by the value of α selected. The feasibility of a given s (determined by the feasible set of coordinates $\mathcal{S}$) is only feasible if it does not lie within the boundary of a circular obstacle with radius r_c and center $(x, y) = (x_c, y_c)$.

4.2 Affinization for SCP within MPC

To facilitate sequential convex programming (SCP) within MPC, we made our problem locally convex via linearization of the car dynamics, energy usage, and obstacle constraint. Specifically, the car dynamics f were linearized as $s_{k+1} = A_f s_k + B_f u_k + c_f$, where $A_f = \frac{\partial f}{\partial s}$, $B_f = \frac{\partial f}{\partial u}$, and $c_f = f(s_{prev}, u_{prev}) - A_f s_{prev} - B_f u_{prev}$, with s_{prev} and u_{prev} being the previous state and control, respectively. Similarly, the energy dynamics where approximated as $E_k = A_e s_k + B_e u_k + c_e$, where $A_e = \frac{\partial E}{\partial s}$, $B_e = \frac{\partial E}{\partial u}$, and $c_e = E(s_{prev}, u_{prev}) - A_e s_{prev} - B_e u_{prev}$, where E is the energy function (a function of speed and inclination). The obstacle constraint, when present, was approximated as $d_k = A_d s_k + c_d$, where $A_d = \frac{\partial D}{\partial s}$ and $c_d = D(s_{prev}, u_{prev}) - A_d s_{prev}$, with D being the nonlinear constraint violation function (returning the distance to the edge of the obstacle).

4.3 Terrain

The terrain used for simulations is shown in figures 1 and 2.

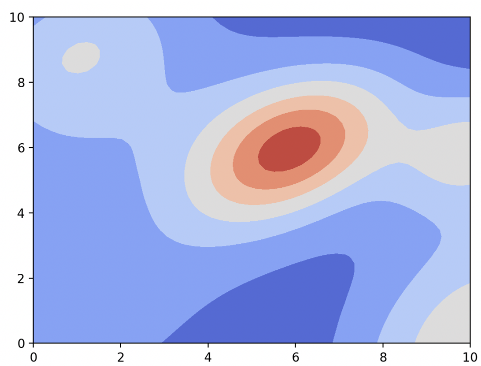

Figure 1: Level set of the terrain used in this project. Note the peak in the middle, which lies between the start point in the bottom left corner and the goal point in the top right corner.

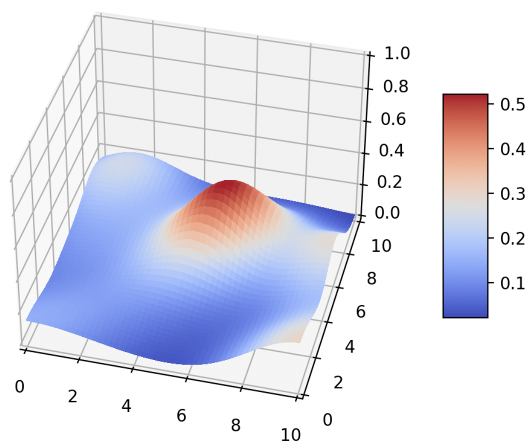

Figure 2: Three dimensional map of the terrain used in this project. Note the peak in the middle, which lies between the start point in the bottom left corner and the goal point in the top right corner.

5 Results

In order to test different levels of trade-off between reaching the goal and energy efficiency, paths were generated with MPC for a variety of different values of α. Note that we avoid cases where α = 0 or very low, as the path is unable to converge to the goal. Note that for each simulation, the MPC time horizon N is set to be equal to the simulation time span T (i.e. we assume we always have enough information to plan fully into the future).

5.1 Optimization for pure distance ($\alpha = 1$)

The test case with α=1 tests path planning that optimizes only for reaching the goal as quickly as possible in minimum distance with no concern for efficiency. We see that this configuration produces a trajectory that passes almost directly over the peak of the hill. The path produced is not straight from the start to goal because vertical distance is also considered when calculating the total distance traversed. The trajectory planned by MPC for $\alpha = 1$ traverses through some of the steepest portions of the map, while speed remains high until the goal is reached. This results in the average efficiency over the trajectory being the lowest of all the test cases, as steep terrain decreases efficiency and the velocity is not optimized for efficiency. See Figure 3 for path planning results for $\alpha = 1$.

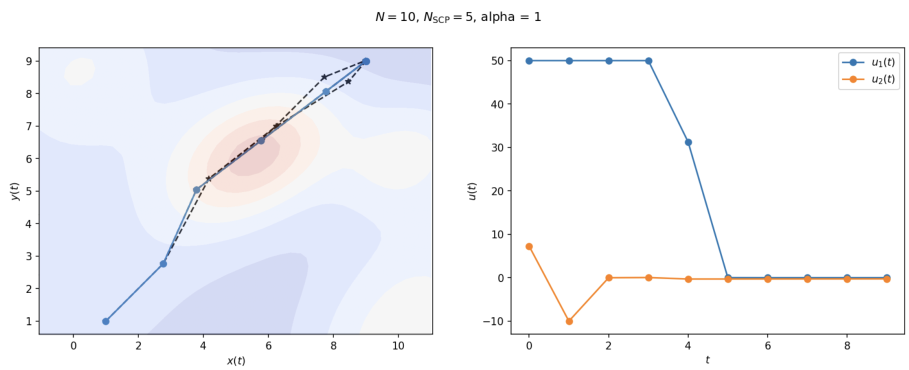

Figure 3: MPC results when $\alpha = 1$. This places no weight on energy efficiency, and instead optimizes for an optimally short path. Note that u_1 is velocity in miles per hour and u_2 is the turn rate in radians per hour.

5.2 Optimization with less weight on energy efficiency ($\alpha = 0.8$)

The test case of $\alpha = 0.8$ demonstrates path planning that weights reaching the goal higher than path efficiency. For this test case, we see the trajectory diverge slightly from the $\alpha = 1$ case, as the algorithm attempts to avoid the steep terrain near the central peak. Speed is also lowered towards the end of the trajectory, and thus approaches a more efficient value, but the algorithm noticeably struggles to converge smoothly to the desired goal point (though it does approach it). Compared to $\alpha = 1$, the mean efficiency over the course of the trajectory increased by approximately 28%. See Figure 4 for path planning results for $\alpha = 0.8$.

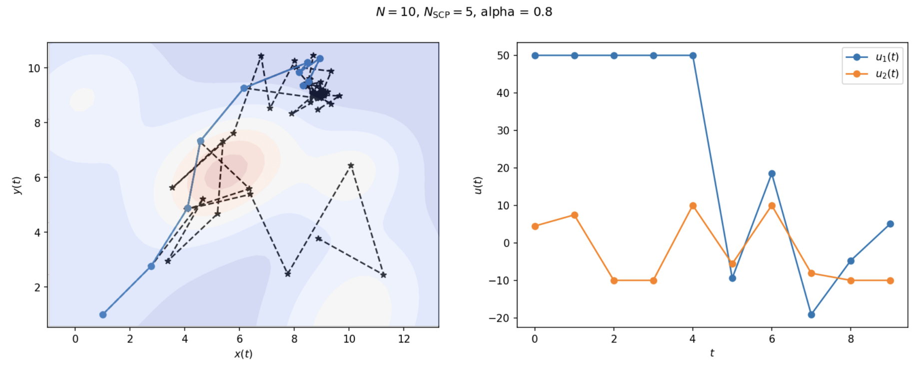

Figure 4: MPC results when $\alpha = 0.8$. Notice how the path moves further away from the peak of the hill compared to $\alpha = 1$. Speed is lowered towards the end of the trajectory, but the algorithm noticeably struggles to smoothly converge to the desired goal point. Note that u_1 is velocity in miles per hour and u_2 is the turn rate in radians per hour.

5.3 Optimization with more weight on energy efficiency ($\alpha = 0.4$)

The test case of $\alpha = 0.4$ demonstrates path planning that weights path efficiency more heavily than reaching the goal. We can see further movement away from the central peak as the algorithm is even more heavily incentived to avoid steep terrain. Due to slower speeds (which are more energy efficient than the max speed), the simulation time and time horizon were both increased to 20 in order to reach the goal point. Like the $\alpha = 0.8$ case, we see issues with smooth convergence to the goal point. Compared to $\alpha = 0.8$, the mean efficiency over the trajectory approximately doubled. See Figure 5 for path planning results for $\alpha = 0.4$.

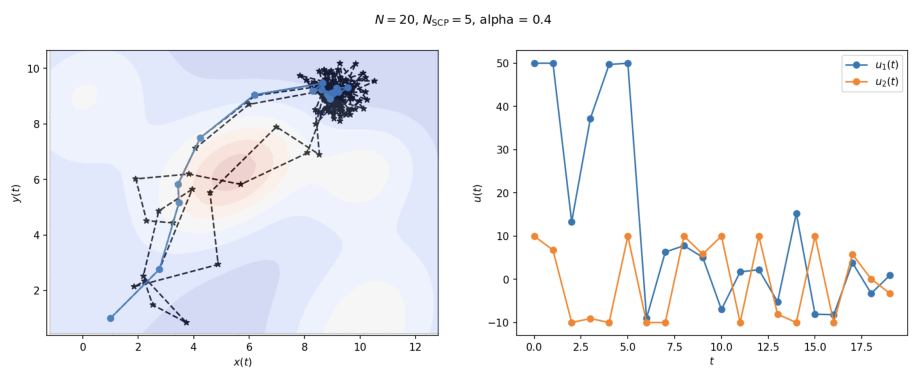

Figure 5: MPC results when $\alpha = 0.4$. Notice that the trajectory moves even further away from the hill's peak, avoiding high inclinations compared to α=1 and α=0.8. Average speed is also lowered, but the trajectory still struggles to converge smoothly to the goal point as with α=0.8. Note that u_1 is velocity in miles per hour and u_2 is the turn rate in radians per hour.

5.4 Optimization with obstacles ($\alpha = 0.4$)

In this test case, we place an obstacle in the way of the optimal path (see Figure 5). Instead of directly moving around the object, we see that the new trajectory instead goes around the other side of the mountain, showing how the path planner is capable of rerouting to significantly different energy efficient paths instead of simply moving around the edge of the obstacle. See Figure 6 for path planning results for $\alpha = 0.4$ with an obstacle.

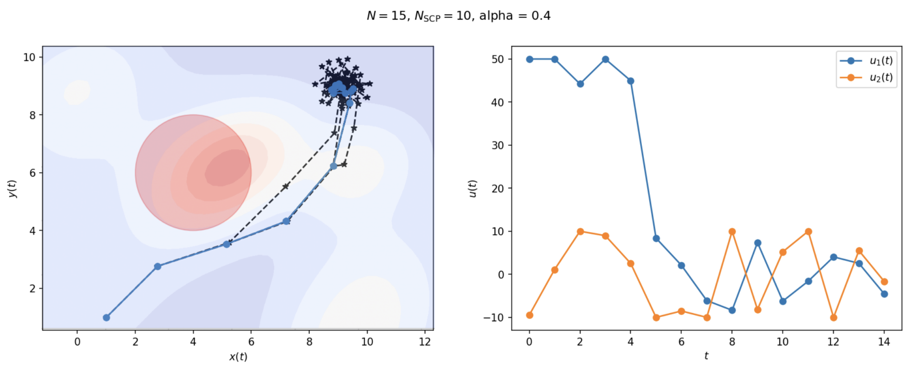

Figure 6: MPC results when $\alpha = 0.4$ with an obstacle blocking the previously calculated optimal path for $\alpha = 0.4$. Notice how, instead of moving directly around the obstacle, the algorithm searches for a new, more energy-efficient path, resulting in a path being generated that goes around the other side of the large central peak. Issues with smooth convergence to the goal point remain present. Note that u_1 is velocity in miles per hour and u_2 is the turn rate in radians per hour.

6 Conclusion/Future Work

Our results show that we are able to optimize trajectories for efficiency relatively well. As α is decreased, which corresponds to optimizing more for energy efficiency, we see that MPC trajectories move away from steep inclines and use lower speeds, which appears to increase the efficiency of the generated paths. This observation is in line with the expected trends from the efficiency curves obtained from literature. Unfortunately, there do appear to be convergence issues when close to the goal point. This is likely due to the cost of the distance to the goal being high near the start point, which causes controller to use a higher speed towards the start. As the goal point is approached, the cost associated with the distance to the goal dips dramatically, making the efficiency costs higher, resulting in the speed being lowered towards the speed for optimal efficiency. Further work should investigate this behavior and look at solutions for smoothing out speed over the course of the trajectory. Currently, we believe that this issue with convergence is due to a struggle between the cost associated with converging to the goal and the cost of picking an efficient control and trajectory. Moving forward, we would also like to look at comparing MPC results to a method like APF or to model MPC obstacle constraints using potential fields.

6.1 Project video link

Project video link: https://youtu.be/vMi4MnLwliI https://youtu.be/vMi4MnLwliI

7 References

References

[1] Rami Abousleiman and Osamah Rawashdeh. "Electric vehicle modelling and energy-efficient routing using particle swarm optimisation". In: *IET Intelligent Transport Systems* 10.2 (2016), pp. 65–72. DOI: https://doi.org/10.1049/iet-its.2014.0177. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2014.0177. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2014.0177.

[2] *Average Fuel Consumption at Increasing Road Grades*. 2014. URL: https://afdc.energy.gov/data/10601.

[3] Farid Bounini et al. "Modified artificial potential field method for online path planning applications". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 180–185. DOI: 10.1109/IVS.2017.7995717.

[4] Farid Bounini et al. "Modified artificial potential field method for online path planning applications". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 180–185. DOI: 10.1109/IVS.2017.7995717.

[5] Ing-Chau Chang et al. "A VANET-Based A* Route Planning Algorithm for Travelling Time- and Energy-Efficient GPS Navigation App". In: *International Journal of Distributed Sensor Networks* 9.7 (2013), p. 794521. DOI: 10.1155/2013/794521. eprint: https://doi.org/10.1155/2013/794521. URL: https://doi.org/10.1155/2013/794521.

[6] Mansoor Davoodi et al. "Clear and smooth path planning". In: *Applied Soft Computing* 32 (2015), pp. 568–579. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2015.04.017. URL: https://www.sciencedirect.com/science/article/pii/S1568494615002422.

[7] Mykel Kochenderfer and Tim Wheeler. In: *Algorithms for Optimization*. 2022.

[8] Chang Liu et al. "Path planning for autonomous vehicles using model predictive control". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 174–179. DOI: 10.1109/IVS.2017.7995716.

[9] Dongfang Ma et al. "An optimal control-based path planning method for unmanned surface vehicles in complex environments". In: *Ocean Engineering* 245 (2022), p. 110532. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.110532. URL: https://www.sciencedirect.com/science/article/pii/S0029801822000129.

[10] Yadollah Rasekhipour et al. "A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2017), pp. 1255–1267. DOI: 10.1109/TITS.2016.2604240.

[11] James Reeds and Lawrence Shepp. "Optimal paths for a car that goes both forwards and backwards". In: *Pacific journal of mathematics* 145.2 (1990), pp. 367–393.

[12] Rob Van Haaren. "Assessment of electric cars' range requirements and usage patterns based on driving behavior recorded in the National Household Travel Survey of 2009". In: *Earth and Environmental Engineering Department, Columbia University, Fu Foundation School of Engineering and Applied Science, New York* 51 (2011), p. 53.

[13] Dong-mei Wu et al. "Fast velocity trajectory planning and control algorithm of intelligent 4WD electric vehicle for energy saving using time-based MPC". In: *IET Intelligent Transport Systems* 13.1 (2019), pp. 153–159. DOI: https://doi.org/10.1049/iet-its.2018.5103. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2018.5103. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2018.5103.

[14] Zhiqiang Zuo et al. "Fast Nonlinear Model Predictive Control Parallel Design Using QPSO and Its Applications on Trajectory Tracking of Autonomous Vehicles". In: *2018 13th World Congress on Intelligent Control and Automation (WCICA)*. 2018, pp. 222–227. DOI: 10.1109/WCICA.2018.8630357.

[15] Zhiqiang Zuo et al. "MPC-Based Cooperative Control Strategy of Path Planning and Trajectory Tracking for Intelligent Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 6.3 (2021), pp. 513–522. DOI: 10.1109/TIV.2020.3045837.

8 Appendix

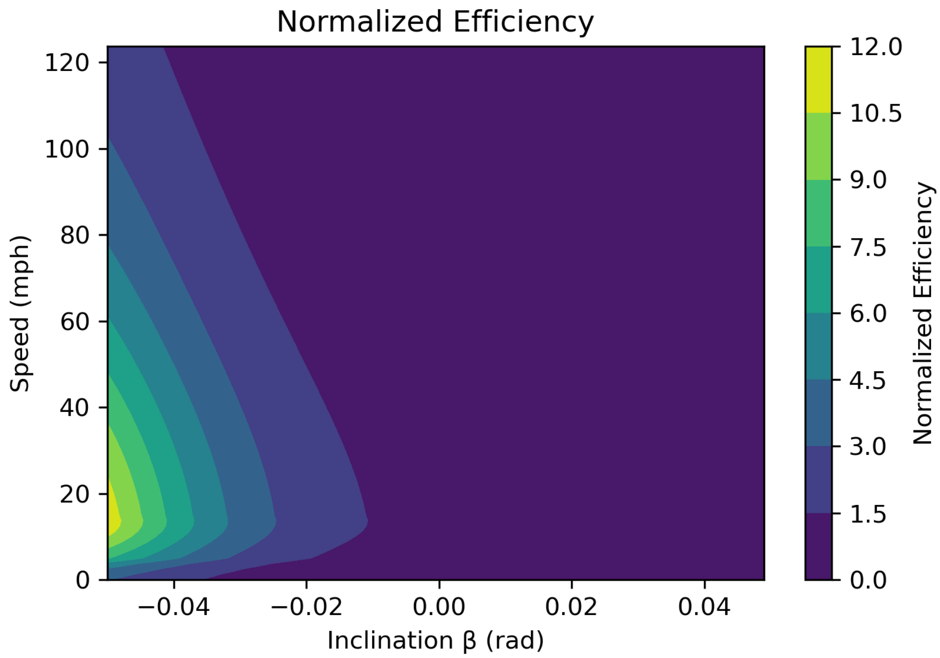

Figure 7: The normalized efficiency plot for our problem setting. Note that the normalized energy usage is the reciprocal of normalized efficiency.

8.1 Sharing with future students

We are okay with this project being shared.

8.2 Code Repository

https://github.com/SpacePanda-42/AA203_project_public.git