AM170B Project Proposal

Nicolas Vaillancourt

April 2025

Abstract

Autonomous vehicle navigation requires precise control over a car's motion to reach desired destinations while respecting physical constraints and dynamic limitations. A key challenge in this domain is computing the control inputs—or forcing terms—necessary to transition a vehicle from its current state to a specified target location. These inputs must be consistent with the vehicle's kinematics or dynamics and must yield feasible, smooth trajectories. In this paper, we propose a method for computing these forcing terms using a sequential refinement multi-headed multilayer perceptron (MLP) model. Our approach iteratively improves the predicted control signals through a structured learning process, enabling accurate and efficient maneuvering across a variety of driving scenarios.

1 Equations of Motion

In this section we will discuss the governing equations of motion for a simple 2D car simulation. The car drives along the xy-plane with an angle θ, forwards speed s and angular velocity ω. The car is controlled by 2 forcing terms, linear acceleration ϕ (pressing the gas pedal) and angular acceleration ψ (turning the steering wheel). To estimate future positions of the car's trajectory in time, a second order Taylor series approximation is used. This requires the 0^{th}, 1^{st} and 2^{nd} derivatives of the cars trajectory corresponding to position, velocity and acceleration.

1.1 Position

To model the car's position, motion and orientation, locations are stored as a 5-dimensional vector. This takes the form $\vec{V} = \langle x, y, \theta, s, \omega \rangle$ where x and y are the location in $\mathbb{R}^2$, $\theta \in [-\pi, \pi)$ is the direction of the car, s is the signed speed in the direction of travel and ω is the angular velocity. Writing the position of the car as a function of time yields:

$$\vec{V}(t) = \underbrace{\langle x, y, \theta, s, \omega \rangle}_{\vec{A}_1} \tag{1}$$

$$\vec{V}(t) = \vec{A}_1 \tag{2}$$

1.2 Velocity

The derivative of the car's position vector $\vec{V} \in \mathbb{R}^5$ is the velocity denoted as $\vec{V}'$. Computing each component of the velocity and factoring out the forcing terms results in:

$$\vec{V}'(t) = \langle s\cos\theta, s\sin\theta, \omega, \phi, \psi\rangle \tag{3}$$

$$\vec{V}'(t) = \underbrace{\langle s\cos\theta, s\sin\theta, \omega, 0, 0\rangle}_{\vec{A_2}} + \phi \underbrace{\langle 0,0,0,1,0\rangle}_{\vec{B_2}} + \psi \underbrace{\langle 0,0,0,0,1\rangle}_{\vec{C_2}} \tag{4}$$

$$\vec{V}'(t) = \vec{A_2} + \phi\vec{B_2} + \psi\vec{C_2} \tag{5}$$

1.3 Acceleration

Taking the derivative again yields the acceleration vector $\vec{V}''$, and factoring as before gives:

$$\vec{V}''(t) = \langle \phi\cos\theta - s\omega\sin\theta, \phi\sin\theta + s\omega\cos\theta, \psi, 0, 0\rangle \tag{6}$$

$$\vec{V}''(t) = \underbrace{\langle -s\omega\sin\theta, s\omega\cos\theta, 0, 0, 0\rangle}_{\vec{A_3}} + \phi \underbrace{\langle \cos\theta, \sin\theta, 0, 0, 0\rangle}_{\vec{B_3}} + \psi \underbrace{\langle 0,0,1,0,0\rangle}_{\vec{C_3}} \tag{7}$$

$$\vec{V}''(t) = \vec{A_3} + \phi\vec{B_3} + \psi\vec{C_3} \tag{8}$$

2 Computing a Trajectory

Approximating a discretized trajectory using a second order Taylor series is important for planning the car's course of motion. At each timestep in the trajectory, the next position is determined by the following equation:

$$\vec{V}(t+\Delta) \approx \vec{V}(t) + \Delta\vec{V}'(t) + \tfrac{\Delta^2}{2}\vec{V}''(t) \tag{9}$$

$$\vec{V}(t+\Delta) \approx \vec{A_1} + \Delta(\vec{A_2} + \phi\vec{B_2} + \psi\vec{C_2}) + \tfrac{\Delta^2}{2}(\vec{A_3} + \phi\vec{B_3} + \psi\vec{C_3}) \tag{10}$$

$$\vec{V}(t+\Delta) \approx \underbrace{(\vec{A_1} + \Delta\vec{A_2} + \tfrac{\Delta^2}{2}\vec{A_3})}_{\vec{A}} + \phi \underbrace{(\Delta\vec{B_2} + \tfrac{\Delta^2}{2}\vec{B_3})}_{\vec{B}} + \psi \underbrace{(\Delta\vec{C_2} + \tfrac{\Delta^2}{2}\vec{C_3})}_{\vec{C}} \tag{11}$$

$$\vec{V}(t+\Delta) \approx \vec{A} + \phi\vec{B} + \psi\vec{C} \tag{12}$$

Lets define a trajectory as the set of $n+1$ points starting at $(\vec{V_0}, t_0)$ and ending at $(\vec{V_n}, t_n)$. At each point the forcing terms applied are ϕ_i and ψ_i. By using evenly spaced timesteps, $t_j = t_0 + j\Delta$ where $\Delta = \frac{t_n - t_0}{n}$ a recurrence relation can be defined.

$$\vec{V}_{j+1} = \vec{A}_j(\vec{V}_j) + \phi_j\vec{B}_j(\vec{V}_j) + \psi_j\vec{C}_j(\vec{V}_j) \tag{13}$$

2.1 Trajectory Optimization

For the following section we will rewrite the trajectory and forcing terms as matrices of the row vectors:

$$\mathbf{V} = \left[\vec{V}_0, \cdots, \vec{V}_n\right]_{n+1\times 5} \tag{14}$$

$$\mathbf{F} = \left[\vec{F}_0, \cdots, \vec{F}_{n-1}\right]_{n\times 2} \tag{15}$$

If we define a target ending point $\vec{T}$ which we want the trajectory to end at, let the loss of a trajectory be the mean squared error (MSE) between the trajectories endpoint $\vec{Y} = \vec{V_n(\mathbf{F})}$ and the target:

$$\mathcal{L}(\mathbf{F}, \vec{T}) = \tfrac{1}{5}\sum_{j=1}^{5}(\vec{Y}_j - \vec{T}_j)^2 \tag{16}$$

Now our goal becomes finding a set of forcing terms that minimize the loss function:

$$\mathbf{F} = \underset{\mathbf{F}}{\operatorname{argmin}}\left[\mathcal{L}(\mathbf{F}, \vec{T})\right] \tag{17}$$

A common approach to finding a value that minimizes a complicated function is gradient descent. We can apply this by defining $\mathbf{F}^i$ as the i^{th} descent step. Using a step size of η, the recurrence relation is:

$$\mathbf{F}^{i+1} = \mathbf{F}^i - \eta\nabla_{\mathbf{F}^i}\mathcal{L}(\mathbf{F}, \vec{T}) \tag{18}$$

3 Gradient Flow

Solving for the gradient of the loss function with respect to the $(n, 2)$ forcing terms $\nabla_{\mathbf{F}^i}\mathcal{L}(\mathbf{F}^i, \vec{T})$ can be done by separating the gradient into a product 2 terms.

$$\nabla_{\mathbf{F}^i}\mathcal{L}(\mathbf{F}^i, \vec{T}) = \left[\frac{\partial \mathcal{L}}{\partial \mathbf{F}^i_{j,k}}\right]_{n\times 2} \tag{19}$$

$$\nabla_{\mathbf{F}^i}\mathcal{L}(\mathbf{F}^i, \vec{T}) = \left[\nabla_{\vec{Y}}\mathcal{L} \times \nabla_{\mathbf{F}^i_{j,k}}\vec{Y}\right]_{n\times 2} \tag{20}$$

The first term has a fairly simple closed form solution:

$$\nabla_{\vec{Y}}\mathcal{L} = \left[\ \tfrac{\partial \mathcal{L}}{\partial \vec{Y}_1},\ \cdots\ ,\tfrac{\partial \mathcal{L}}{\partial \vec{Y}_5}\ \right]^\top \tag{21}$$

$$\nabla_{\vec{Y}}\mathcal{L} = \tfrac{1}{5}\left[\ 2(\vec{Y}_1 - \vec{T}_1),\ \cdots\ ,2(\vec{Y}_5 - \vec{T}_5)\ \right]^\top \tag{22}$$

$$\nabla_{\vec{Y}}\mathcal{L} = \tfrac{2}{5}(\vec{Y} - \vec{T})^\top \tag{23}$$

The second term requires a different approach. When holding all else constant each forcing term $\mathbf{F}_{j,k}$ applied along the path at $\vec{V}_j$ causes a slight change in the position of the next point $\vec{V}_{j+1}$. This perturbation propagates forward in time and has an effect on the final position $\vec{V}_n$. Without integrating this perturbation for all forcing terms there is a local approximation for small changes leveraging a cumulative product of Jacobians. Lets define the change in a future position $\vec{V}_{j+1}$ as a function of the previous position $\vec{V}_j$ as the j^{th} Jacobian J_j:

$$J_j = \nabla_{\vec{V}_j} \vec{V}_{j+1} = \begin{bmatrix} \frac{\partial \vec{V}_{j+1,x}}{\partial \vec{V}_{j,x}} & \frac{\partial \vec{V}_{j+1,x}}{\partial \vec{V}_{j,y}} & \frac{\partial \vec{V}_{j+1,x}}{\partial \vec{V}_{j,\theta}} & \frac{\partial \vec{V}_{j+1,x}}{\partial \vec{V}_{j,s}} & \frac{\partial \vec{V}_{j+1,x}}{\partial \vec{V}_{j,\omega}} \\ \frac{\partial \vec{V}_{j+1,y}}{\partial \vec{V}_{j,x}} & \frac{\partial \vec{V}_{j+1,y}}{\partial \vec{V}_{j,y}} & \frac{\partial \vec{V}_{j+1,y}}{\partial \vec{V}_{j,\theta}} & \frac{\partial \vec{V}_{j+1,y}}{\partial \vec{V}_{j,s}} & \frac{\partial \vec{V}_{j+1,y}}{\partial \vec{V}_{j,\omega}} \\ \frac{\partial \vec{V}_{j+1,\theta}}{\partial \vec{V}_{j,x}} & \frac{\partial \vec{V}_{j+1,\theta}}{\partial \vec{V}_{j,y}} & \frac{\partial \vec{V}_{j+1,\theta}}{\partial \vec{V}_{j,\theta}} & \frac{\partial \vec{V}_{j+1,\theta}}{\partial \vec{V}_{j,s}} & \frac{\partial \vec{V}_{j+1,\theta}}{\partial \vec{V}_{j,\omega}} \\ \frac{\partial \vec{V}_{j+1,s}}{\partial \vec{V}_{j,x}} & \frac{\partial \vec{V}_{j+1,s}}{\partial \vec{V}_{j,y}} & \frac{\partial \vec{V}_{j+1,s}}{\partial \vec{V}_{j,\theta}} & \frac{\partial \vec{V}_{j+1,s}}{\partial \vec{V}_{j,s}} & \frac{\partial \vec{V}_{j+1,s}}{\partial \vec{V}_{j,\omega}} \\ \frac{\partial \vec{V}_{j+1,\omega}}{\partial \vec{V}_{j,x}} & \frac{\partial \vec{V}_{j+1,\omega}}{\partial \vec{V}_{j,y}} & \frac{\partial \vec{V}_{j+1,\omega}}{\partial \vec{V}_{j,\theta}} & \frac{\partial \vec{V}_{j+1,\omega}}{\partial \vec{V}_{j,s}} & \frac{\partial \vec{V}_{j+1,\omega}}{\partial \vec{V}_{j,\omega}} \end{bmatrix} \tag{24}$$

If we consider a small perturbation ϵ_j to the $j+1^{\text{th}}$ point on the trajectory caused by the forcing term $\mathbf{F}_{j,k}$ an approximation for the displacement of the endpoint can be written as a product. The perturbation at the next timesteps approximately follow the locally linear Jacobian transformation:

$$\epsilon_{j+1} \approx J_j \times \epsilon_j \tag{25}$$

$$\epsilon_{j+2} \approx (J_{j+1} \times J_j) \times \epsilon_j \tag{26}$$

$$\epsilon_n \approx \underbrace{\prod_{n-1}^{j} J_j}_{J_j^c} \times \epsilon_j \tag{27}$$

$$\epsilon_n \approx J_j^c \times \epsilon_j \tag{28}$$

Where J_j^c is the backward cumulative product of Jacobians and J_n^c is defined to be the identity matrix I_5. Writing this out in terms of the gradient of the endpoint:

$$\nabla_{\mathbf{F}^i_{j,k}} \vec{Y} \approx J^c_{j+1} \times \nabla_{\mathbf{F}^i_{j,k}} \vec{V}_{j+1} \tag{29}$$

With equations for each of the terms, the gradient of the loss function with respect to each of the forcing terms can be written as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{F}^i_{j,k}} = \nabla_{\vec{Y}} \mathcal{L} \times \nabla_{\mathbf{F}^i_{j,k}} \vec{Y} \tag{30}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{F}^i_{j,k}} \approx \tfrac{2}{5} (\vec{Y} - \vec{T})^\top \times J^c_{j+1} \times \nabla_{\mathbf{F}^i_{j,k}} \vec{V}_{j+1} \tag{31}$$

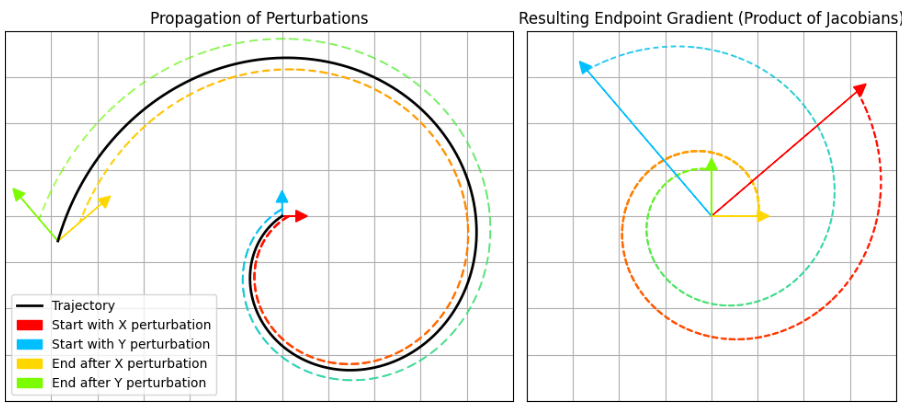

Figure 1:

Going back to the formula for computing the next point in a trajectory, the gradient of $\vec{V}_{j+1}$ with respect to $\mathbf{F}^i_{j,k}$ can be broken into 2 parts:

$$\frac{\partial \vec{V}_{j+1}}{\partial \mathbf{F}^i_{j,1}} = \vec{B}(\vec{V}_j) \tag{32}$$

$$\frac{\partial \vec{V}_{j+1}}{\partial \mathbf{F}^i_{j,2}} = \vec{C}(\vec{V}_j) \tag{33}$$

Putting this all together, an analytical approximation for the gradient of the loss function with respect to the forcing terms is:

$$\nabla_{\mathbf{F}^i} \mathcal{L} = \frac{2}{5} \begin{bmatrix} (\vec{Y} - \vec{T})^\top J_1^c \vec{B}(\mathbf{V}_0) & (\vec{Y} - \vec{T})^\top J_1^c \vec{C}(\mathbf{V}_0) \\ \vdots & \vdots \\ (\vec{Y} - \vec{T})^\top J_n^c \vec{B}(\mathbf{V}_{n-1}) & (\vec{Y} - \vec{T})^\top J_n^c \vec{C}(\mathbf{V}_{n-1}) \end{bmatrix}_{n \times 2} \tag{34}$$

4 ML Approach

When analytic solutions fail, general function approximators can help find satisfactory solutions.

$$\mathbf{F}^{i+1} = M(F^i, V^i) \tag{35}$$