First-Order Vehicle Tracking

By: Maxwell D. Pisciotta

December 1, 2018

1 Introduction

The navigating dynamics of self-driving vehicles are crucial to the accuracy and safety of these systems. These need to be further investigated as self-driving vehicle technology continues to emerge[1]. For the navigation systems to work adequately, the steering angle and forces being applied during acceleration and braking need to be considered. The navigation, acceleration, and braking systems all work concurrently when a vehicle is turning a corner, so this is a manageable action to consider for a first-order vehicle tracking simulation.

In order to model all the subsystems of the vehicle that interact when in operation, a complex simulation is needed. In previous studies models were based on a nonlinear autonomous vehicle with two degrees of freedom[2]. These models were revised to better account for vehicle stability, tire forces, and driving effects[3]. The following model is similar, but simplified for ease and understanding. This also leaves a lot of possibility for future work to expand this model.

2 Problem Statement

This first-order vehicle tracking system uses a bicycle model[3] (seen in **Figure 1**) to simulate a vehicle traveling along a track, the optimal steering angle, and torque application on each of the tires must be determined. This system was modeled in MatLab using differential-algebraic equations and a manual time integration technique.

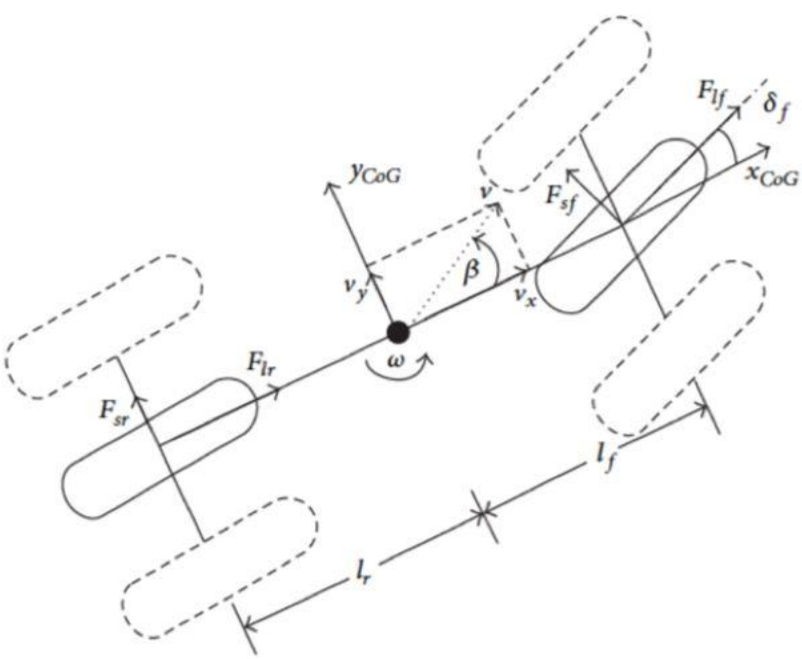

Figure 1: Simplified bicycle model for vehicle considered, 5-degree of freedom automobile model[4].

The purpose of this bicycle model is to optimize the torque being applied to the front tires to successfully complete the Colorado School of Mines (CSM) race track (seen in **Figure 2**) in the fastest time. This is a front-wheel-drive car that instead of having a braking torque applied, the driving torque is reduced. The MatLab model was coded to be robust in the sense that the driving and braking torque on the front and rear tires can be altered by changing the values of the initial guess. For the following case study, the initial driving torque was altered, but the braking torque was kept constant and equal to zero. This emphasizes the impact of the dynamic driving torque calculations built into the model.

The CSM race track can be modeled using the below equation.

$$\begin{cases} y=0 & 0m \leq x \leq 200m \\ y=0.1-(0.1x-19.9)\cos(0.01x-2) & 200m \leq x \leq 1600m \end{cases}$$

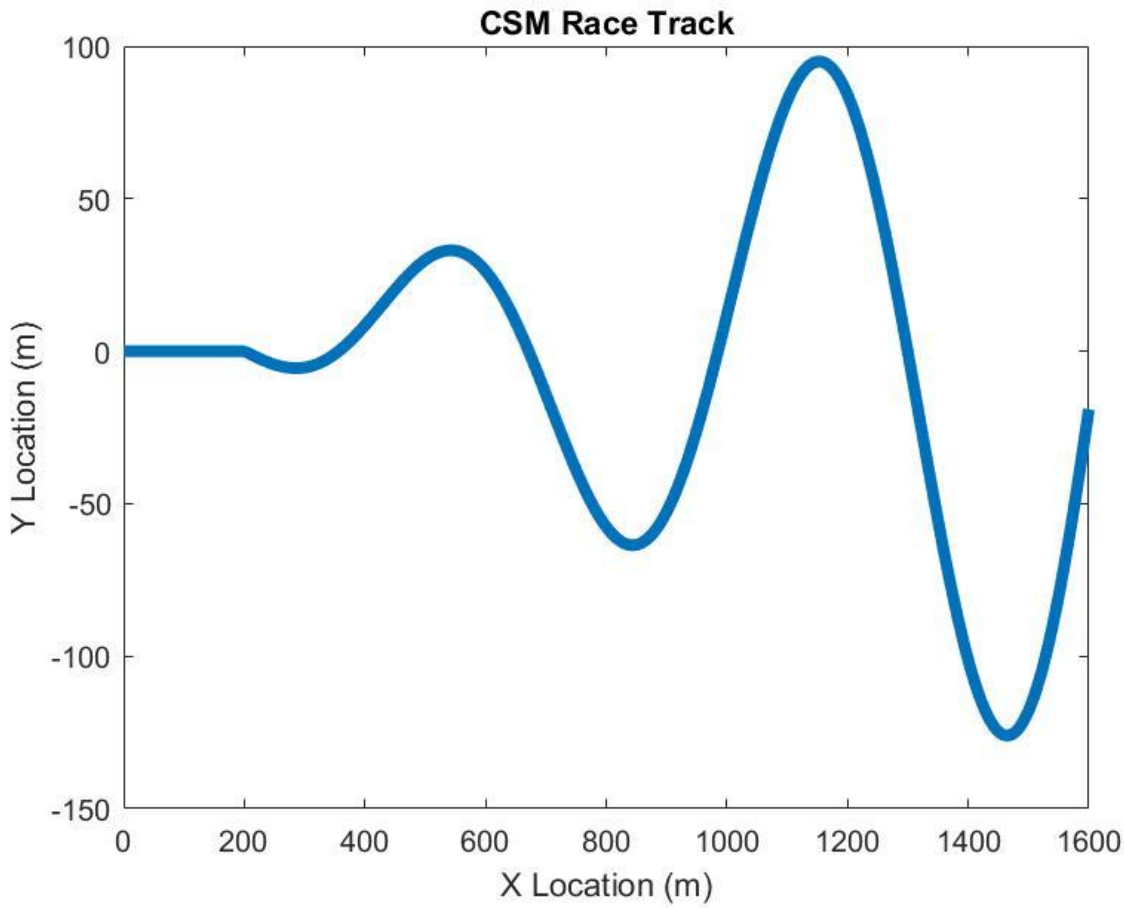

Figure 2: CSM Race Track

The assumptions that are built into this model include:

- Each front tire has the same applied torque.
- Each rear tire has the same applied torque.
- The steering angles of both front tires are equivalent.
- The forward torque on the rear tires, and braking torques on both front and rear tires are equal to zero.
- The maximum steering angle for this vehicle is 0.2π in either direction.

Further parameters and given values for this problem are found in **Appendix A**.

3 Simulation Design

In order to model the vehicle, the bicycle model was employed. This simplifies the problem, with the assumption that the front and rear wheels respond to stimulus in the same manner. A graphic of this can be seen in **Figure 1**.

The x-direction illustrated in **Figure 1** will be referenced as the longitudinal or forward direction of the vehicle in the rest of the document. The variables pertaining to this portion of the vehicle will be subscripted with an “l”. The y-direction illustrated in **Figure 1** will be referenced as the latitudinal or side direction in the rest of the document. The variables pertaining to this portion of the vehicle will be subscripted with and “s”. The variables x and y will pertain only to the absolute frame of reference in which the track and the car will be propagating through time.

3.1 Governing Equations

Below are the governing equations in which the simulation was modeled to incorporate.
The force balance on the vehicle in the longitudinal direction is as follows:

$$m\frac{dv_f}{dt} = mv_s\omega + F_{l,f}\cos(\delta_f) - F_{s,f}\sin(\delta_f) + F_{l,r} - \mathrm{sgn}(v_f)\frac{C_{D,f}A_f\rho_{air}v_f^2}{2}$$

The force balance on the vehicle in the latitudinal direction is as follows:

$$m\frac{dv_s}{dt} = -mv_f\omega + F_{l,f}\sin(\delta_f) + F_{s,f}\cos(\delta_f) + F_{s,r} - \mathrm{sgn}(v_s)\frac{C_{D,s}A_s\rho_{air}v_s^2}{2}$$

The torque balance on the overall vehicle is as follows:

$$I_z\frac{d\omega}{dt} = l_f\left(F_{l,f}\sin(\delta_f) + F_{s,f}\cos(\delta_f)\right) - F_{s,r}l_r$$

The torque balance on the front wheels is as follows:

$$J\frac{d\omega_f}{dt} = T_{d,f} - \mathrm{sgn}(\omega_f)T_{b,f} - r_eF_{l,f}$$

The torque balance on the rear wheels is as follows:

$$J\frac{d\omega_r}{dt} = T_{d,r} - \mathrm{sgn}(\omega_r)T_{b,r} - r_eF_{l,r}$$

The movement of the car in the x-direction of the absolute frame of reference is as follows:

$$\frac{dx}{dt} = v_f\cos(\theta) - v_s\sin(\theta)$$

The movement of the car in the y-direction of the absolute frame of reference is as follows:

$$\frac{dy}{dt} = v_f\sin(\theta) + v_s\cos(\theta)$$

The angle of the car in the absolute frame of reference is as follows:

$$\frac{d\theta}{dt} = \omega$$

Further equations to determine the forces on the tires can be found in **Appendix B**.

This system of eight differential equations were programmed into MatLab and solved using the MatLab command ode23s. This solver was chosen because it is a solver that can solve "stiff" differential equations and differential-algebraic equations. This could have also been solved using ode45, but the computational time would have been increased and the accuracy of the results would be decreased.

3.2 Vehicle Tracking Methods

To first determine if the vehicle is on the track, the solver is given the initial conditions that the car starts at the origin of the absolute frame of reference (0,0). These initial conditions allow for the ODE solver to project the x and y locations of the car across a time step. The time step for this simulation was set to 0.2 seconds. Larger time steps were initially chosen and the vehicle was unable to stay on the track. When smaller time steps were tested, the computational time was increased, but the results did not warrant this tradeoff. Therefore, a 0.2 second interval is near the state of mesh independence for this model.

The projected x and y locations are compared to the x and y locations of the track at distances 5 meters before and 5 meters after the current x location. The minimum absolute distance from the track and the derivative of the track with respect to space are calculated. These are used in determining the steering angle to get the vehicle closer to the track. Additionally, the derivative of the track with respect to space is determined 25 meters ahead of the current position of the vehicle to predict an upcoming turn. This is used to moderate the torque applied to the front tires as the car needs to reduce speed through turns to stay close to the track. The equation used to determine the new steering angle is as shown below.

$$\delta_f = -\theta_{err} + \operatorname{atan}\left(\frac{ke_f}{v_f}\right)$$

Where δ_f is the steering angle, θ_{err} is the difference between the two calculated derivatives, k is the gain parameter, e_f is the absolute error distances of the vehicle's current position and the minimum distance from the track, and v_f is the current velocity of the vehicle in the longitudinal direction. This equation was inspired by the Stanley Method of vehicle tracking[5]. There are conditionals to ensure the steering angle does not exceed the maximum steering angle as set by the problem statement.

When the vehicle predicts rounding a turn, the torque applied to the front tires is reduced. It is reduced using the following equation.

$$T_{d,f,new} = T_{d,f,0} - k_T m v_f^2 \theta_{turn}^2$$

Where $T_{d,f,new}$ is the torque being applied in the next iteration, $T_{d,f,0}$ is the previous torque being applied, k_T is the gain parameter, m is the mass of the vehicle, v_f is the velocity of the vehicle in the longitudinal direction, and θ_{turn} is the difference between the car's derivative and that of the path 25 meters ahead of the car's current location. This equation was derived through dimensional analysis.

Within the while loop (set to stop when the x location of the car is equal to the x location of the end of the track), the last values calculated by the ODE solver are used to guess values for the following iteration.

4 Results

When the simulation is run with lower applied torque values, the vehicle remains closer to the track, but also takes more time to finish the track distance. This indicates that the car's steering ability is increased at low torque values. The orange curve shows the time it takes to complete the track. It shows a trend of exponentially decreasing as the torque increases, but stays consistent from approximately 100 Nm to 150 Nm, at around 100 seconds. The blue curve shows that at low torque values, the vehicle remains closer to the track, up until 50 Nm, where it increases very far past the maximum desired error (red line on graph, equal to 10 meters). The maximum distance away from the track occurs at 100 Nm of torque. There is no data point for 70 Nm of applied torque, because this torque value results in a singularity point within the code, where the vehicle will is unable to return to the track. The car path graphed against the CSM Race Track for 40 Nm, 70 Nm, and 100 Nm can be seen in **Figures 4, 5, and 6**.

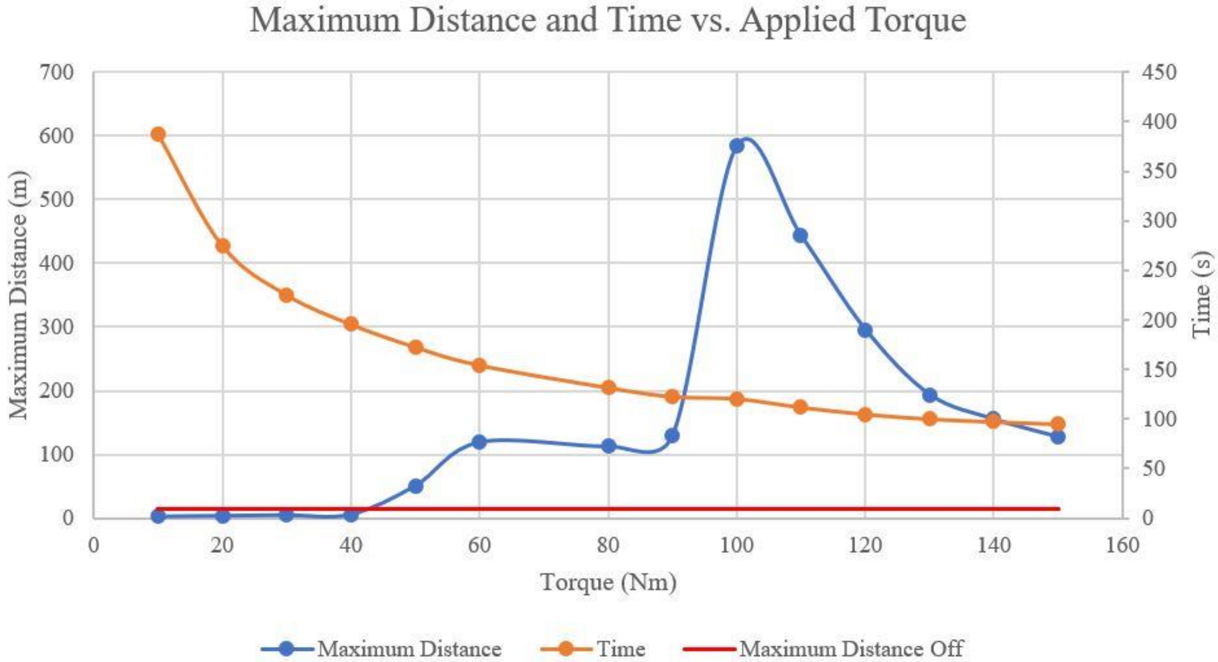

Figure 3: Maximum Distance and Time vs. Torque Applied

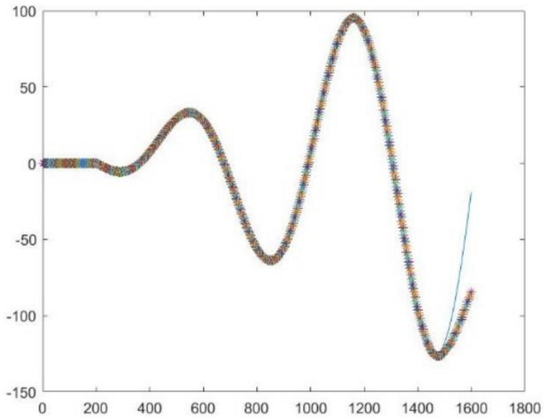

Figure 4: Car Path vs. CSM Race Track with 40 Nm of Applied Torque

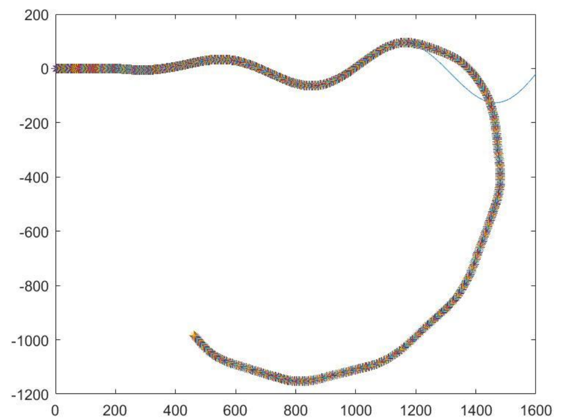

Figure 5: Car Path vs. CSM Rack Track with 70 Nm of Applied Torque

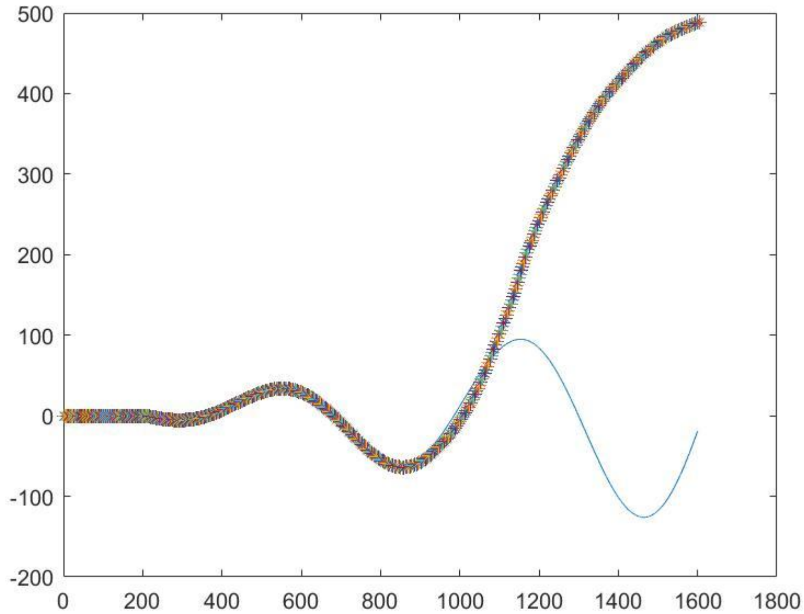

Figure 6: Car Path vs. CSM Race Track with 100 Nm of Applied Torque

5 Discussion

The graphs above show the difference in the vehicle's path based on the initially applied torque. It can be seen that at 40 Nm of applied torque, the vehicle remains on the track for the majority of the time, but the final turn is too tight for it to complete. The graph with the data from the 70 Nm trial shows that the physics of the model break down, not allowing the vehicle to return to the track. This could be due to the sampling minimum distance from the track, the sampling of the derivative terms, or a limitation within the ode23s solver. Lastly, in **Figure 6**, it can be concluded that initially applying 100 Nm of torque is too high of a value, as the vehicle can barely make it past the third curve on the track. Although the algorithm shows the vehicle making similar trajectories to the CSM Race Track, they are not precise enough for the vehicle to correct itself to return to the track.

6 Future Work

This is a first-order tracking simulation and there is a lot of future work that can be completed to improve it. This study specifically focused on the changes when the front-wheel driving torque was altered. This does not account for 4-wheel-drive or all-wheel-drive scenarios, but could be extended to do so. The equations used in the simulation did not include rear-wheel driving torque or braking force, but should be considered in the future. Additionally, different race track geometries or differences in altitude could also be investigated in the future installments of this model. Lastly, different "stiff" ODE solvers could be utilized to minimize the instability of ode23s at the input value of 70 Nm of torque. Continued work is suggested in this field as self-driving vehicle technology advances and enters the transportation market.

Bibliography

[1] "10 Million Self-Driving Cars Will be on the Road by 2020", Business Insider, June 15, 2016. https://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6

[2] U. Kiencke and L. Nielsen, Automotive Control Systems, Springer, New York, NY, USA, 2000.
[3] Wang, X., and Shi, S., "Analysis of Vehicle Steering and Driving Bifurcation Characteristics," *Mathematical Problems in Engineering*, 2015.

[4] Wang, X., Shi, S., Liu, L., and Jin, L., "Analysis of Driving Mode Effect on Vehicle Stability," *International Journal of Automotive Technology*, **14**(3), pp. 363-373, 2013.

[5] Snider, Jarrod M., "Automotive Steering Methods for Autonomous Automobile Path Tracking," *Carnegie Mellon University*, CMU-RI-TR-09-08, 2009.

Appendices

Appendix A: Input Parameters

Parameter	Symbol	MatLab Variable	Value
Mass	m	car.m	1500 kg
Moment of Inertia in z-direction	I_z	car.I_z	3000 kg*m^2
Length from Midpoint to Front Axle	l_f	car.l_f	1.2 m
Length from Midpoint to Rear Axle	l_r	car.l_r	1.3 m
Polar Moment of Inertia	J	car.J	1.0 kg*m^2
Wheel Rolling Radius	R_e	car.r_e	0.224 m
Frontal Area of Vehicle	A_f	car.A_f	1.7 m^2
Side Area of Vehicle	A_s	car.A_s	3.5 m^2
Frontal Drag Coefficient	C_{Df}	car.C_Df	0.3
Side Drag Coefficient	C_{Ds}	car.C_Ds	0.4
Density of Air	ρ_{air}	car.rho_air	1.29 kg/m^3
Steering Angle Gain Parameter	K	k	1
Decrease Torque Gain Parameter	k_T	k_T	0.001

Appendix B: Tire Force Equations

Following is the equation used to determine the force on the tires on the vehicle.

$$Force_{tire}=D\sin\left(C\tan^{-1}\left(Bx-E\left(Bx-\tan^{-1}(Bx)\right)\right)\right)$$

Where x is the longitudinal slip.

Below are the values used to determine the force on the corresponding tires as per X. Wang, 2015.

Tire	B	C	D	E
Flf	11.275	1.56	2574.8	0.4109
Flr	18.631	1.56	1749.6	0.4108
Fsf	11.275	1.56	2574.7	-1.999
Fsr	18.631	1.56	1749.7	-1.7908