Optimization of Vehicle Fuel Consumption

Brett Siddoway | Tanner Jasperson | Titus Quah | Vivian D'Souza

Introduction

Vehicle automation and optimization is a rapidly growing and increasingly important research field. Rising fuel costs and environmental concerns have directed much of that research towards reducing emissions and fuel consumption. This project focuses on studying the fuel consumption of a light vehicle over a set two-dimensional distance with randomized stop signs placed along the route. The optimal solution will minimize both the vehicle fuel consumption and total time required to make the trip. The solution will be compared to a base case where the vehicle accelerates rapidly to the speed limit and minimizes only travel time.

Literature Review

Much of our project mirrors work done by Tu Luu et al.[1], but our methods of model creation and some optimization objectives differ. The paper uses principles of machine learning to process the vehicle information when creating the digital vehicle model. We chose to first model our process with semi-empirical methods, taking data first then fitting equations to that data by varying parameters within those equations. Our model also used varying parameters with changing velocity to account for gear shift rather than utilizing vehicle specific gear ratios as was done by Guang and Jin[2]. We found that by changing parameters with set speed ranges, or adjusting for the vehicle gear as a function of velocity, allowed us to more accurately model the vehicle performance as shown by the data collected. After attempting to implement MHE and MPC, however, we found that this initial model was lacking in appropriate model dynamics such as the gears and vehicle inertia, so we had to alter which drive-cycle we used and apply a more involved approach as was done by our group member Vivian in prior work experience [3].

[1] Tu Luu, H., Nouvelière, L., & Mammar, S. (2010). Dynamic programming for fuel consumption optimization on light vehicle doi://doi.org/10.3182/20100712-3-DE-2013.00097
[2] Guang, H., & Jin, H. (2019). Fuel consumption model optimization based on transient correction doi://doi.org/10.1016/j.energy.2018.12.067
[3] Cyber physical modeling of Automotive Control Systems For Engine, Driveline, and Vehicle by Uwe Kiencke and Lars Nielsen

Model Description

Our model was based off the following equations:

$$ma = F_E - F_D - F_B - mg\,sin(\theta) \quad (1)$$
$$F_D = C_d\, v^2 \quad (2)$$
$$F_E = k_1 G + k_2/(G+1) + k_3 G^{0.5} \quad (3)$$

For this simulation, our manipulated variables are the vehicle fuel consumption rate (G) and the braking force (F_B). The braking force was not required to create the digital twin, but will be taken into account when simulating and optimizing the vehicle performance over different courses. The constants k_i in Equation 3, the vehicle mass m in Equation 1, and the drag coefficient C_d were determined by fitting the data to the model. We found that the constants changed depending on the velocity that the vehicle was operating at and will be implemented in our model with a method similar to gain scheduling. These values are shown in Table 1 below.

Table 1

Velocity Range (m/s)	Parameter	Value
0 - 14	m	8.314149
	C_d	0.086232
	k_1	0.0
	k_2	17.94961
	k_3	6.852568
14-40	m	109.4097
	C_d	0.241189
	k_1	2.651655
	k_2	0.0
	k_3	8.594629
	m	64.2855

40 +	C_d	0.04953
	k_1	0.72726
	k_2	38.7965
	k_3	0.0

Using these values, we were able to fit the following curves to our data:

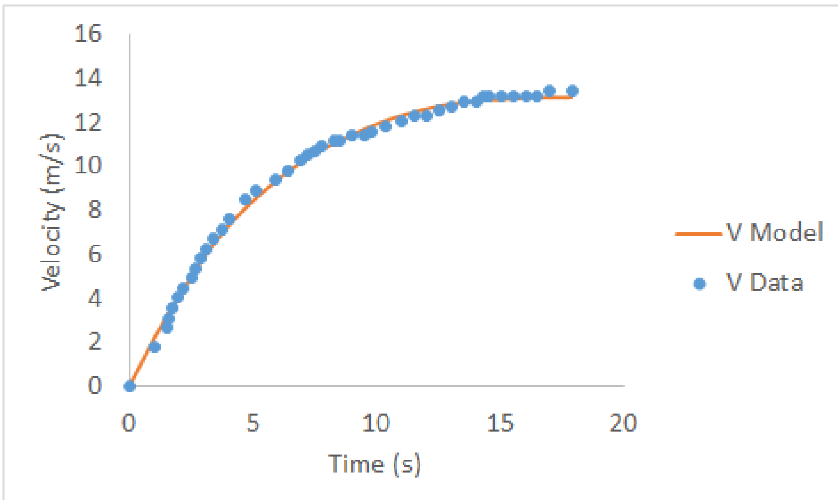

Figure 1

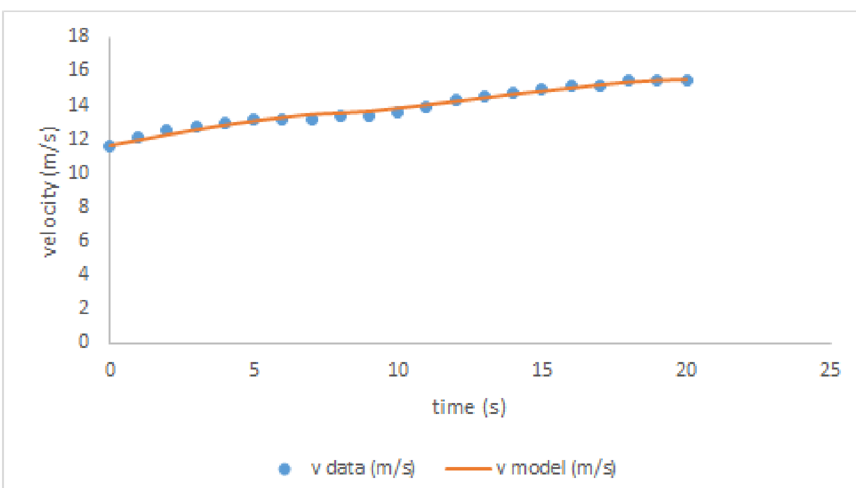

Figure 2

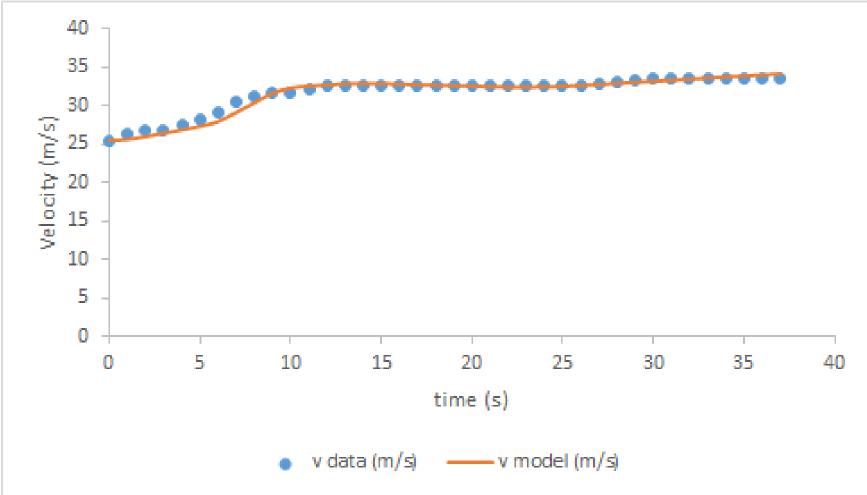

Figure 3 - Model fit to data

Taking these parameters, we were able to produce the following simulated response to a hypothetical system with stops added in. We noticed, however, that the modeled approach resulted in always driving at the maximum velocity and approaching that velocity at the maximum allowable acceleration, which we did not find would lead to an optimal solution as seen in Figure 4.

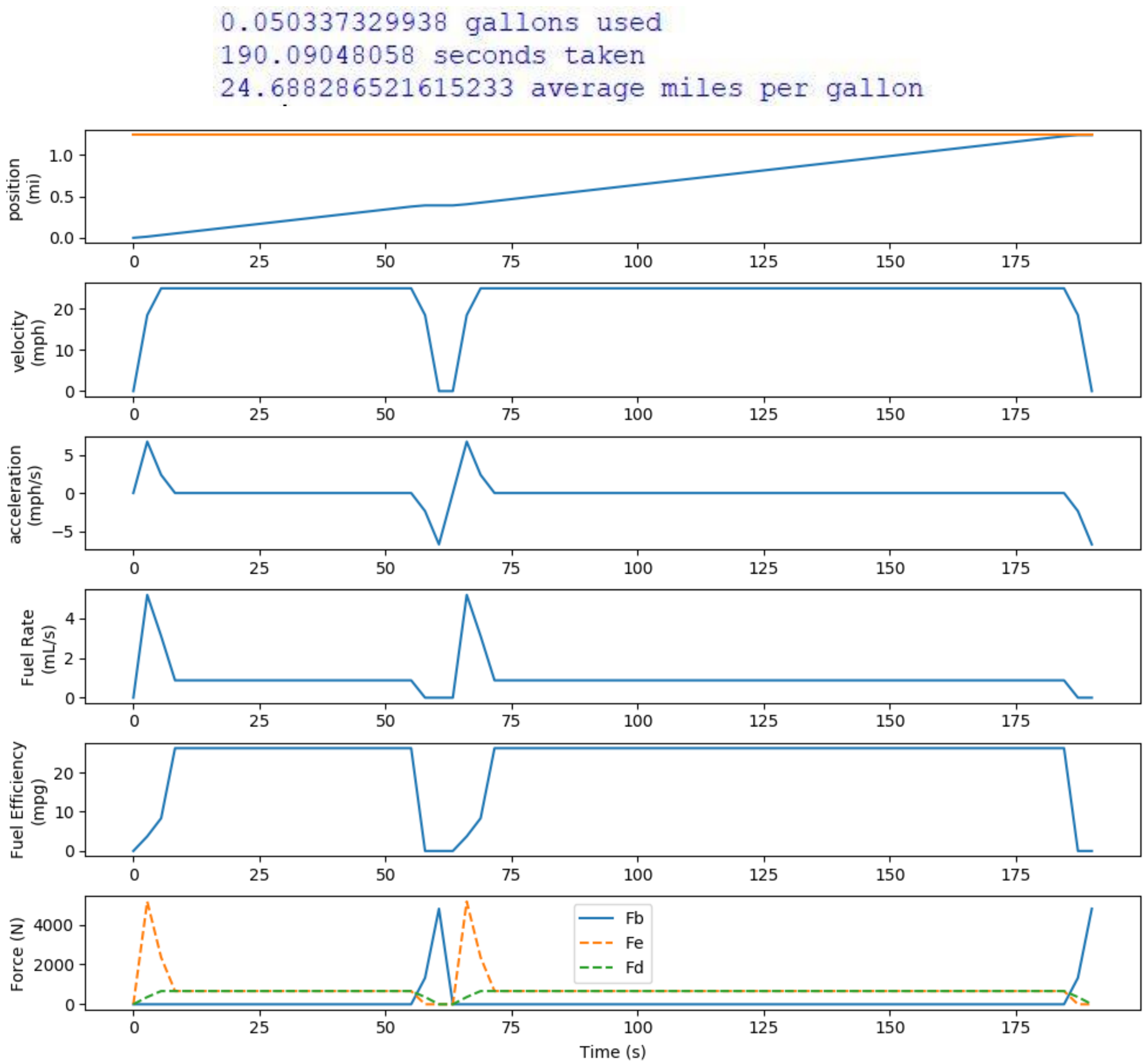

Figure 4 - Results for horizon control

This solution was due to our incorrect parameters derived from few datasets and missing a term for the forces on the car. The parameters determined were from a single drive cycle. However, when tested on other drive cycle data that was collected, the fit was bad and no parameters or velocity windows could be found to improve the fit. Furthermore, the data gathering method was flawed in that exact data could not be obtained since these measurements were made by taking data points from a rotating dial indicator. This led to bad readings when the car switched from decelerating/maintaining velocity to acceleration since the dial's fuel consumption reading is delayed which led to the model have low fuel consumption with high acceleration. The rolling resistance and gear ratio of a car had also been ignored since it was assumed these terms may merge with the drag coefficient and be accounted for with the inclusion of the velocity windows for different parameters. Due to the delayed fuel consumption meter and difficulty in determining velocity windows for the current vehicle and data, it was concluded a model based on a more valid source (industrial data) would be used. The model equations were modified to account for gear change and rolling resistance as shown in Eqns. 4

$$F_E = \begin{cases} F_p f_{dr}(\text{gear})(u) & u < 0 \\ F_b(u) & u \geq 0 \end{cases} \tag{4a}$$

$$u = \begin{cases} \text{brake position} & u < 0 \\ \text{pedal position} & u \geq 0 \end{cases} \tag{4b}$$

$$\text{gear} = \begin{cases} 1.0 & v \leq 0.2 \\ 3.65 & 0.2 < v \leq 7.2 \\ 2.15 & 7.2 < v \leq 11.2 \\ 1.45 & 11.2 < v \leq 17.9 \\ 1.0 & 17.9 < v \leq 20.6 \\ 0.83 & 20.6 < v \end{cases} \tag{4c}$$

$$F_D = 0.5\rho_{air} C_d A v^2 \tag{4d}$$

$$F_{rr} = (m + \text{load}) C_{rr} \cos(\text{grade}) \tag{4e}$$

$$F_g = 9.81(m + \text{load}) \sin(\text{grade}) \tag{4f}$$

$$(m + \text{load})\frac{dv}{dt} = F_e - F_d - F_{rr} - F_g \tag{4g}$$

F_p = Engine power plant force $\quad f_{dr}$ = Final drive ratio $\quad A$ = Cross sectional area of vehicle

F_{rr} = Rolling resistance force $\quad C_{rr}$ = Rolling resistance coefficient $\quad F_g$ = Gravity force

Figure 3 shows the model of the vehicle running on a driving cycle with a proportional integral controller. As seen in the figure, the gear can change more than one level in the model so to fix this, the vehicles inertia will be taken into account and if the model becomes too convoluted, a time delay on gear changing will be added. Furthermore, the pedal position will be related to fuel consumption with empirical data. This model will be implemented into the horizon control with objectives minimize final time and minimize overall fuel consumption. This will similarly be done with model predictive control.

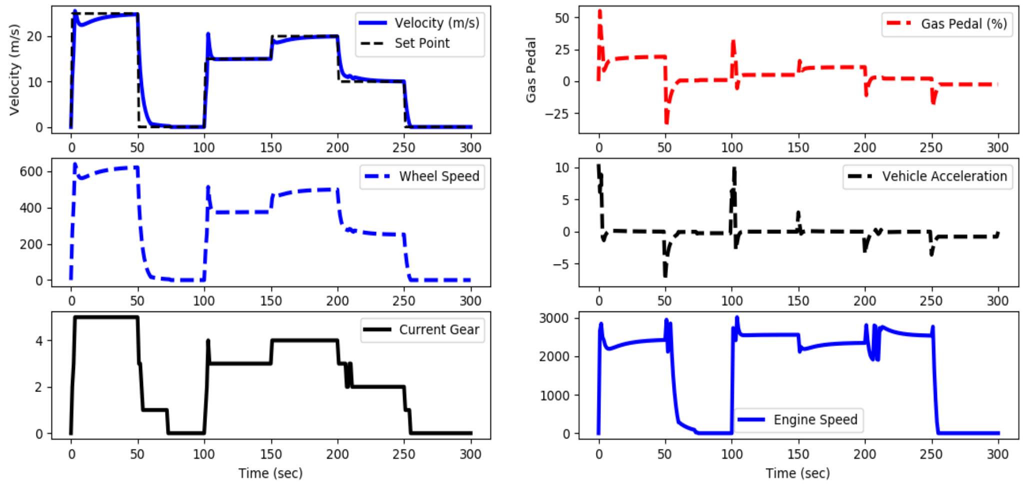

Figure 5 - Model of a motorcycle with proportional integral control

Preliminary results from the new model are promising. We will continue to work on developing a good relationship between engine speed and fuel consumption, but in the meantime we integrated the more recent model equations with our optimizer. For the optimization calculations, we assumed a linear relationship between the accelerator (pedal) and the fuel consumption. Figures 6 and 7 show two results predicted by the optimization code.

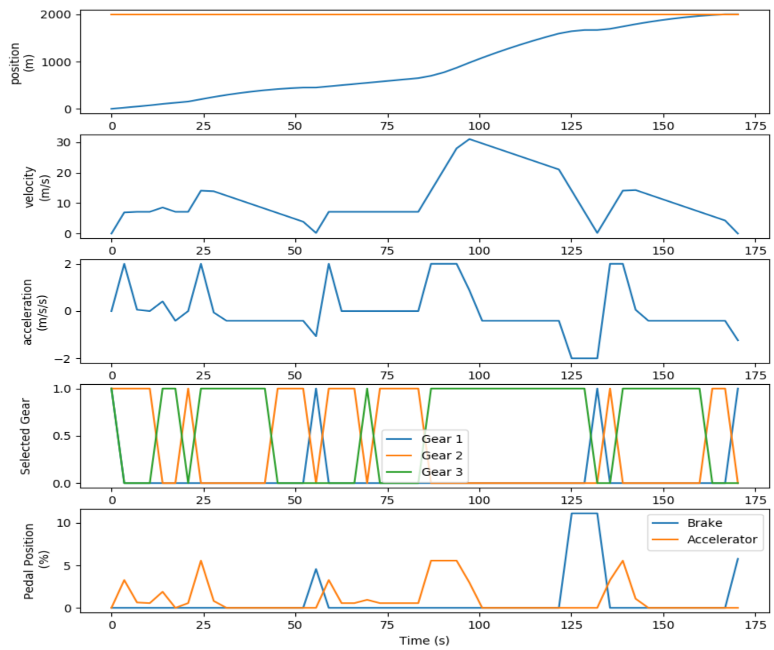

Figure 6 - Preliminary results of updated model being run with no speed limit.

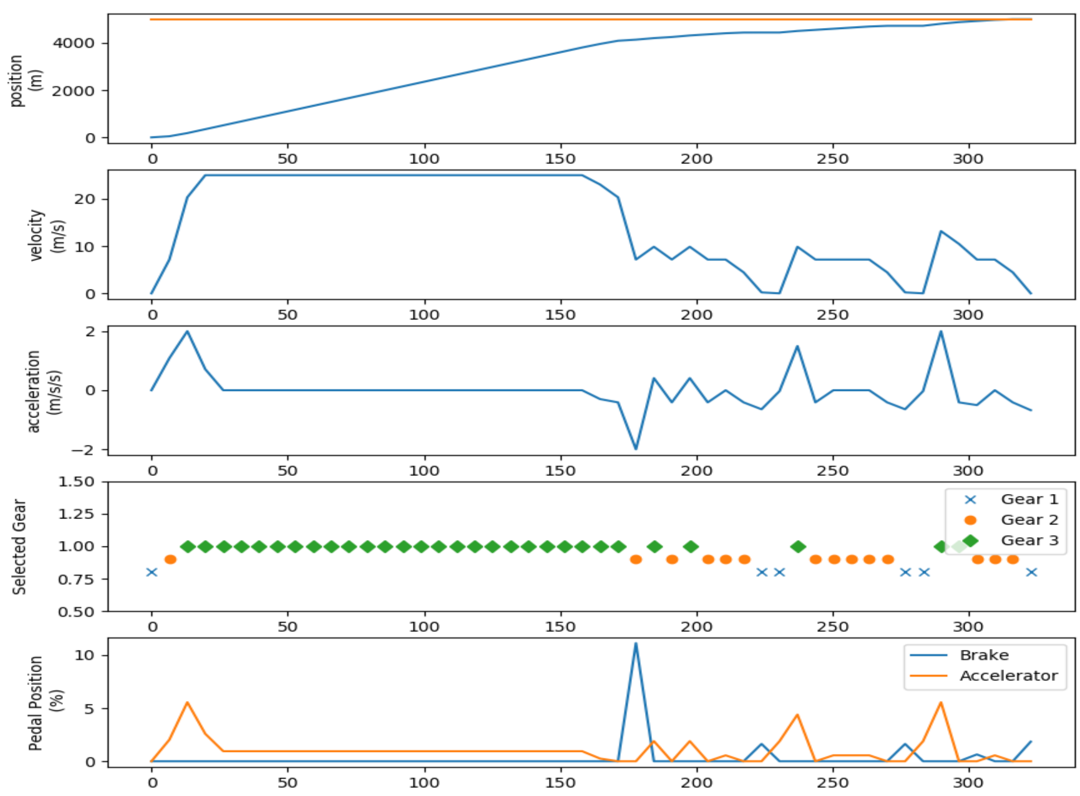

Figure 7 - Preliminary results of updated model being run with a 25 m/s speed limit

These results are promising because we see that the 'optimal' solution is no longer to accelerate as quickly as possible. Along with those mentioned above, notable changes to our model in this version include the addition of a gear box, an accelerator pedal, and a brake pedal.

Parameter Estimation and Discussion

To tune our controller, we first needed to have a method of estimating model parameters based on system data. We decided to implement Moving Horizon Estimation (MHE) into our model over a ten second horizon. The moving parameters we are estimating are the gear the vehicle is in based on velocity and fuel consumption performance, the gear efficiency, and a factor to account for the braking resistance if velocity is less than 0.1.

First Attempts

We first attempted to implement model the vehicle as it was with all its gears switching, brake logic and other defining equations. This led to a model predictive controller with over 200 degrees of freedom since there were many integer choices that went into the developing gear logic, velocity logic and brake logic.

We then attempted to implement the MHE with a simplified system assuming a first order process. Due to complications with the solver and non-convergence issues, we were not able to successfully compile a solution, but we suspect that the issue came from inconsistencies in how our model "creating" the data treated engine resistance when the accelerator pedal was not being pressed and the MHE system lacking relationships to account for this resistance. We decided it would be more valuable to go straight into implementing the MHE with our full system.

Successful Solution

Figure 8 shows our successful solution for estimating the moving parameters over the horizon. Notice how closely our model fits the data over the entire time span. There are some instances where the model proves physically incorrect, such as when the velocity goes negative when approaching a zero velocity. The data shows the vehicle stopping at these points, but our estimator gave parameters leading to the vehicle moving backwards, which would not physically happen unless the vehicle were put in reverse. This is assumed to be insignificant and will be modified with the controller to just make it a zero velocity if it would otherwise be negative. This negative velocity is likely due to the brake fudge factor which is meant to be 1 when the velocity is greater than 0.1 and 0 when the opposite is true. This switch is meant to stop the braking from turning into a reverse gear. This factor also takes care of engine braking that the simulation includes as well as engine box and vehicle inertia effects on the gears. Since these would either introduce mixed integers or a state space problem, we decided we would hold off to add these features and decide whether the effects were significant enough to add in (or when we would have time).

The model has been modified to limit the MHE solver to only select parameters that would yield velocities greater than zero. An issue with this MHE is the parameters vary a lot with the noise as seen in Figure 2. This is a difficult issue to resolve since the WMODEL should not be increased and DMAX limited too much since the parameters are meant to change when the velocity of the car changes. These variations appear to be the cause of an incorrect calculation of the engine speed. The parameter for gear does not ever change from 0 and so the engine speed is always read at 1000 when it should vary up to 9000. We are not sure how to force the controller to do so, but one possibly is start adding some mixed integer choices where it is a certain gear for specified velocities. This would increase the complexity of the MHE with 15 mixed integers and likely would make this estimator obsolete as it would take more than 0.1 seconds to solve.

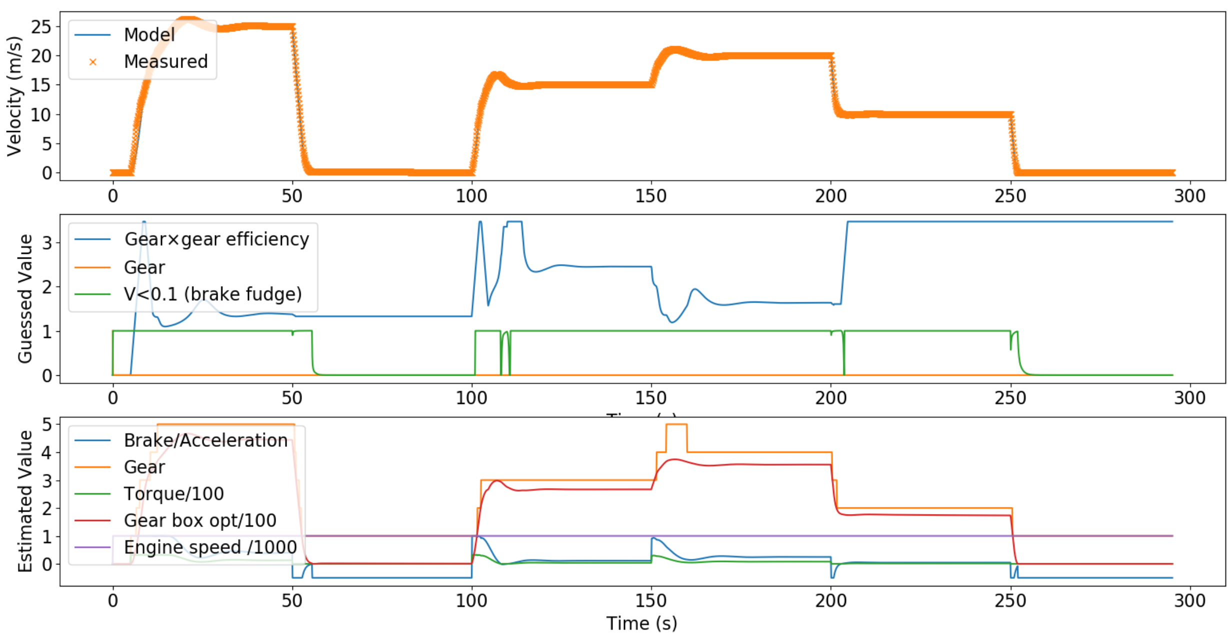

Figure 8 - Model performance data with estimated gear parameters vs noiseless data

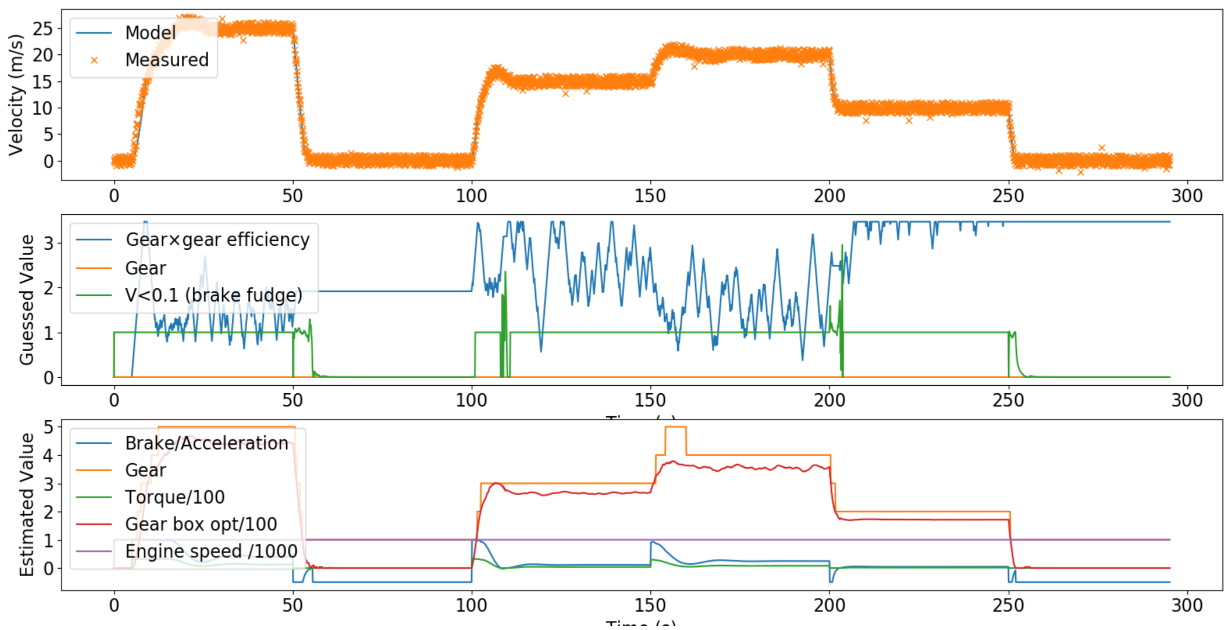

Figure 9 - Model performance data with estimated gear parameters vs noisy data (±1 with occasional ±2.5)

Control and Optimization Results and Discussion

Using an improved MHE model by estimating velocity with a first order process with dead time, the model was able to better predict velocity, engine speed, engine torque and fuel consumption rate as seen with Figure 10. The engine gears, integer values, were approximated by a cubic spline function of velocity. The engine speed and torque were already cubic splines and engine torque had a sign function to decide between the engine torque or brake torque. The fuel flow was approximated by a basic spline as a function of torque and engine speed.

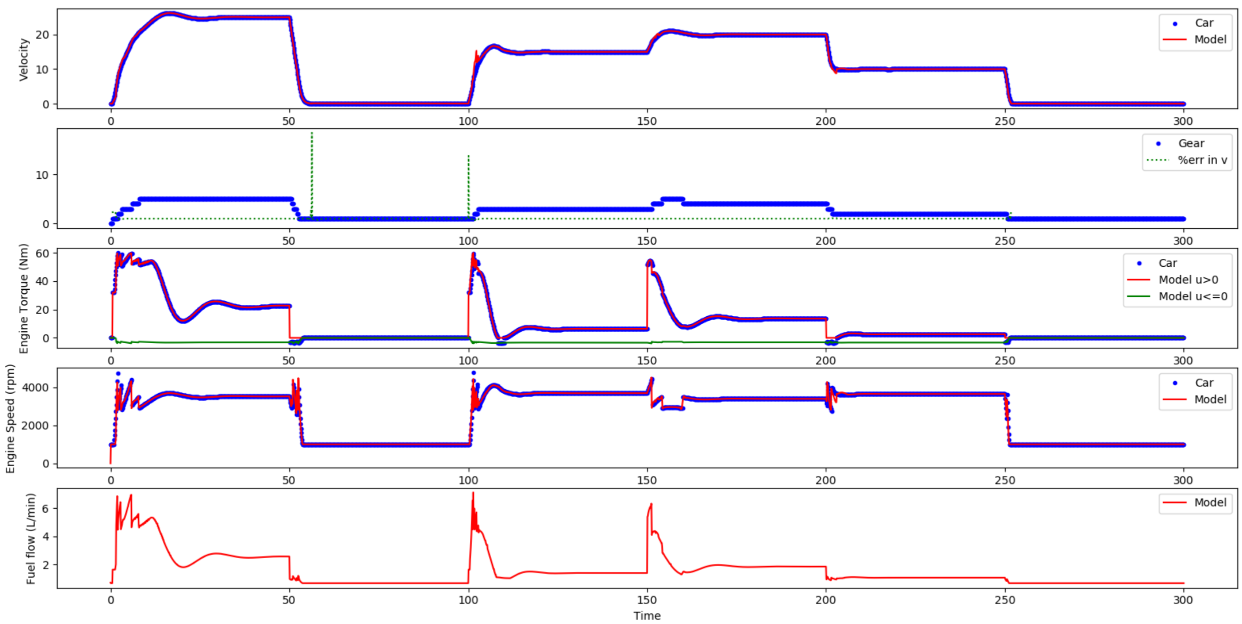

Figure 10 - FOPDT model of car velocity

This model was then input into a MPC to reach certain distances and stop with a constant speed limit. At first, the goal was to minimize final time every 100 meters, but this resulted in too many degrees of freedom and failed a lot. The MPC was then configured to minimize the distance between itself and the goal as well as take in consideration the fuel consumption. Since this did not lend itself well to new goal inputs, the final model has the position as a controlled variable that aims to reach the goal with an objective function minimizing the integral of the fuel usage. Once it passes the goal and has a velocity below 0.1 m/s, a new set point is given and the process repeats. There is a very high weight for exceeding the setpoint high since the goals are treated as stop signs. Also, in the case of MPC failure, the model would lightly step on the brakes and stop accelerating. This model unfortunately is very unstable and depends heavily on the weight values assigned to being below or above the set point and how much it wants to save fuel. As seen with Figure 11, it is possible that the weight set point must change depending on the goal it needs to reach since it reaches the first and second set point, but it does not attempt to reach the third set point. Also, this model predictive controller is currently not viable since to simulate 15 seconds takes around 30 minutes. Due to this long run time, it was difficult to get significant results for how much fuel the MPC saves. Figure 11 shows the model traveling to 10 meters and stopping then travels to 50 meters and stops and then fails to proceed to 100 meters. This example was tested without the fuel minimizer.

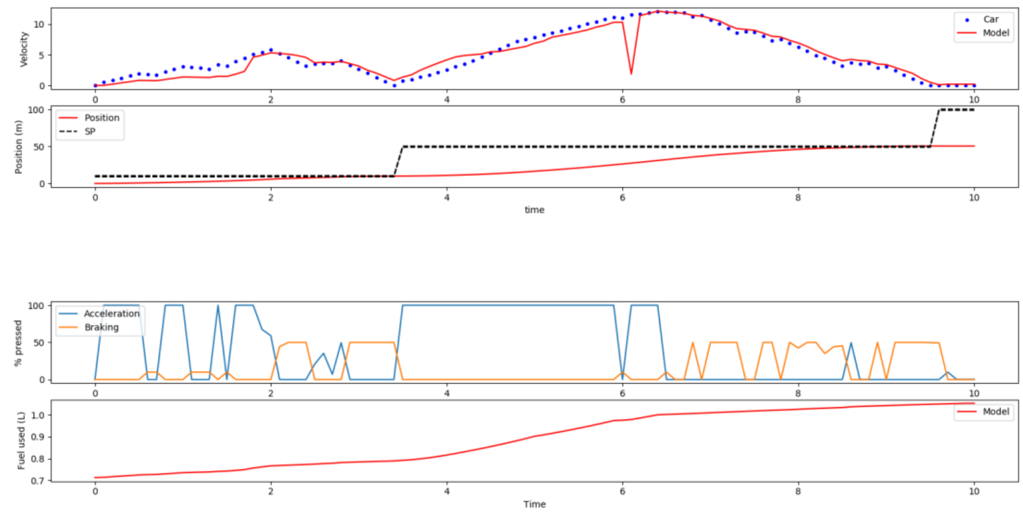

Figure 11 - MPC of a vehicle without fuel minimization

The MPC fails quite a bit and creates some disturbances when the accelerator suddenly goes to zero and braking begins. Also the MHE makes some serious errors at times as well. This is possibly due to erroneous guess values and the lack of tuning performed on these parameters so far. We do not understand why the controller does not attempt to accelerate to meet the last set point since the only objective in this model is from the set points. It may be due to the MHE giving bad values that would indicate no acceleration would occur if the acceleration pedal were pressed. There are other tests where the model completely ignores the set point high and cruises through the stop sign and does not brake as seen in Figure 12. Again this depends on the weights assigned to the SPHI, SPLO or objective function, but it may be due to the fuel objective function overrides the control variable objectives and thus the vehicles does not brake or the MPC keeps failing and sends it a low braking value. The latter is the case for Figure 12. There was one trial that worked well with the fuel optimizer as seen in Figure 13. Unfortunately, due to the short distance it travels, it is difficult to notice a large difference in fuel consumption. However, compared to the non fuel optimizing MPC, the acceleration and braking periods seems to be longer. None use less that 100% of the allowed usage though. Strangely, the fuel optimizer brakes at times when attempting to accelerate to a higher velocity. In all, this MPC has a lot of work to be done. First, it must be determined how to reliably solve the MPC and have it attempt to reach the next set point. This has many points of investigation such as the MHE values, possibly changing weights, and also considering using a different objective method such as a objective function.

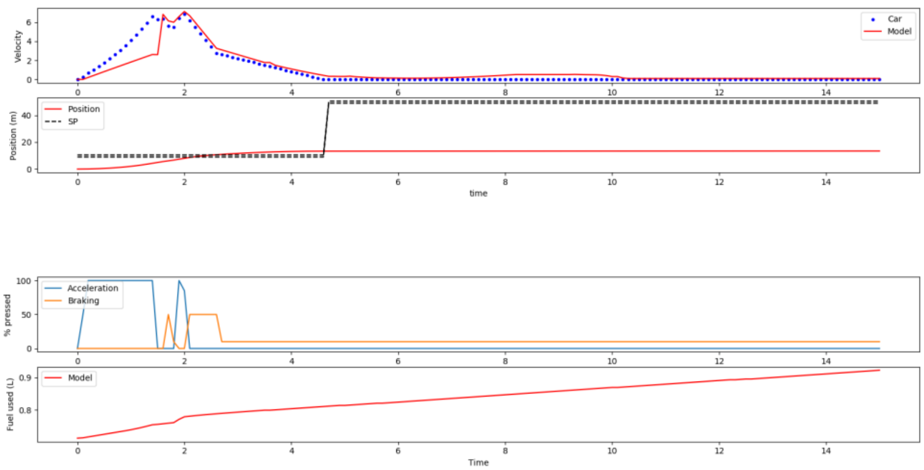

Figure 12 - Fuel optimizer MPC malfunctioning

Another point would be to make the drive more realistic such as adding a DMAX high to the variables to prevent from moving from 100 to 0 back up to 100 and also adding changing speed limits or road grade. This would need to be done after the MPC works reliably.

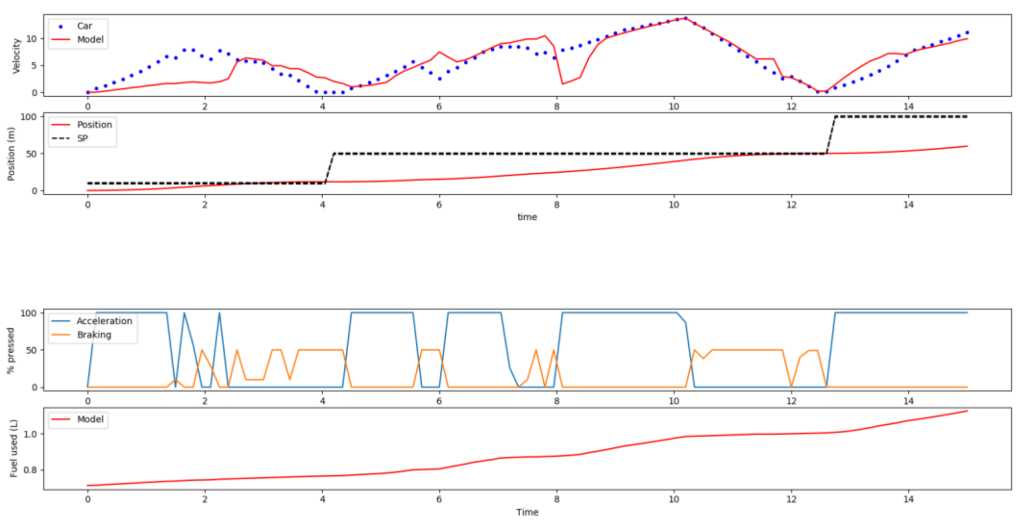

Figure 13 - MPC of a vehicle with fuel optimization

Conclusions

Using our final model, we were able to minimize both fuel consumption and travel time over a predetermined course. Even this model, however, did have some limitations, namely that it would become unstable if the simulation time was increased to allow for further future prediction. In the future, we could determine an optimal horizon to look back to as well to get even better parameter for our vehicle model.