Optimum Allocation of TCAT Hybrid Buses to Maximize Fuel Efficiency

Raghav Batra (rb698), Michael Galbato (mag386), Red Giuliano (rg586)

November 13, 2017

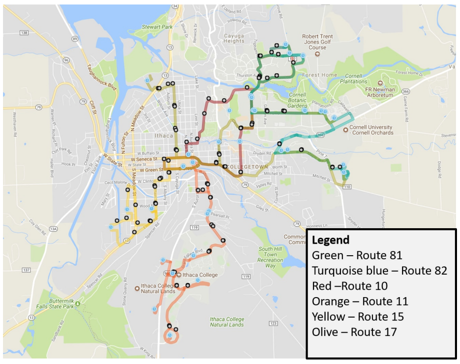
Abstract

In this manuscript, we present a model that simulates the diesel fuel usage of hybrid and diesel buses on certain TCAT bus routes in Ithaca. Our model makes use of elevation data to calculate the physical power exerted by the engine at a series of discrete points along the route. When modeling the hybrid bus, we take into consideration the proper conditions for battery power to be used, and use this information to determine the total energy consumed by hybrid and diesel buses along each route. We then solve the allocation problem using linear programming to find the optimal distribution of hybrid buses in order to maximize fuel savings and maintain the current number of buses along each route. We determined that the optimal allocation is two hybrid buses on route 10, two buses on route 11, one bus on route 15, one bus on route 17, and two buses on route 81. TCAT can save \$28,630 when compared to an allocation of no hybrid buses to these routes and can save \$12,411 when compared to the least optimal allocation of hybrid buses. Finally, we discuss further extensions and improvements to our model to improve its accuracy.

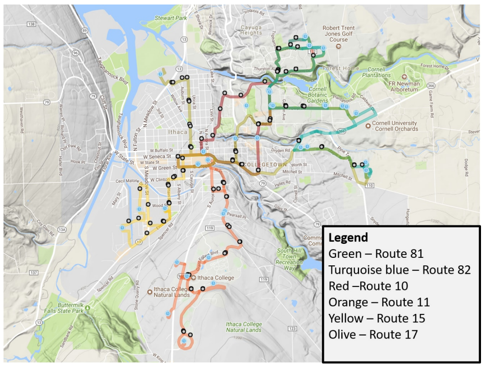
Contents

1 Introduction

The Tompkins Consolidated Area Transit (TCAT) is a private, non-profit public transportation that serves the City of Ithaca and surrounding urban area. Running on 33 routes with its 54-bus fleet, the TCAT has a 4.13 million annual trips. Currently, TCAT has 8 hybrid buses that it wants to allocate on 6 of its routes, 10, 11, 15, 17, 81, and 82 to maximize fuel efficiency. Hybrid buses are attractive in the sense that they both pollute less and are more fuel efficient, so they should be properly allocated to maximize fuel efficiency. We attempt a bottom up approach to modeling the problem, using equations from classical mechanics to simulate the amount of power used by diesel and hybrid buses along each of the considered routes. Given this information, we then determine an optimal allocation of hybrid using linear programming.



(a) Without terrain



(b) With terrain

Figure 1: Bus routes

2 Assumptions

2.1 Assumptions about the physical world

- There is no friction.
- Over short distances, each stretch of road has a constant slope can be represented by a right triangle.

2.2 Assumptions about the buses (diesel and electric hybrid), engines and battery

- Our model does not take into account weights of the passengers, or the mass of the fuel in the tank at any given time. These values vary by route and by time, but should not have too great of an effect the fuel efficiency.
- The bus never stops to refuel.
- We assume that there is no traffic on any of the routes.
- The batteries do not lose efficiency over the course of the simulation..
- The bus gains its charging for the electric motor only from the regenerative brakes.
- We set the minimum state of charge of the battery to 60% (as shown in section 3) in order to keep enough charge for applications such as heating, display illuminations, and ignition [6]. Due to the discrete nature of our simulation, the battery power can drop slightly below the minimum state of charge if the electric engine outputs too much energy over a given interval.

2.3 Assumptions about the simulation

- The driver maintains a constant velocity v throughout the route and does not stop.
- Each route is assumed to be a loop, here onward called a *cycle* (this is the case for almost all routes).
- There is a constant integer number of buses on the cycle during the operating hours of the route.
- There is no lag between cycles.

3 Parameter Values

Finding exact value for most of these parameters is a bit overoptimistic, so the values reported below have been chosen by taking the average of several measures from the references. Care has been taken to ensure that the units of all parameters are consistent (here on called "standard units").

- Values for the bus:

Measurement	Value	Value in Standard Units	References
Bus mass, M	30000 pounds	13409 kg	[1] [2]
Bus velocity, v	30 miles/hour	13.5 m/s	[4]
Acceleration due to gravity, g	9.8 m/s	9.8 m/s	Elementary physics

- Values for the battery, charging, and motors:

Measurement	Value	Value in Standard Units	References
Battery capacity, C_{max}	25 kWh	25 kWh	[5]
Minimum operating battery level (State of Charge) C_{min}	60%	60%	[5] [6]
Charge gained from regenerative braking, r	20% of power	20% of power	[9] [10]
Electric motor maximum power,	150kW	150 kW	[11]
Diesel engine maximum power P_{max}	388 hp	290 kW	[12]

- Other values:

Measurement	Value	Value in Standard Units	References
Electric engine slope threshold, Max_{slope} (% grade)	8.56%	8.56%	*
Chemical energy of gas converted to kinetic energy of motion	7%	7%	[11]
Electric energy from battery converted to kinetic energy of motion	23%	23%	[11]
Conversion from kWh to gallons of diesel, C_{diesel}	$\frac{1}{40.28}$	$\frac{1}{40.28}$	[14]

* The slope threshold is calculated using the formula $P_i = M * g * (h_i/d) * v$ given the maximum electric motor power, the acceleration of gravity, the mass of the bus, and the velocity of the bus.

4 Model

To help the reader better understand the model, We illustrate the model through the use of a particular route, Route 10.

For elevation data (CSV files) about the other routes, consider the link to the GitHub repository in the codebase at the end of the manuscript.

4.1 Data collection

As can be seen in Figure 1, the hilly terrain of Ithaca affects the routes in consideration. We decide to make use of the slope at various points along each route in our models.

Using `geocontent.org` [1], we were able to plot the bus routes by hand and obtain the elevation profiles. An example that explains this process is included below for reference:

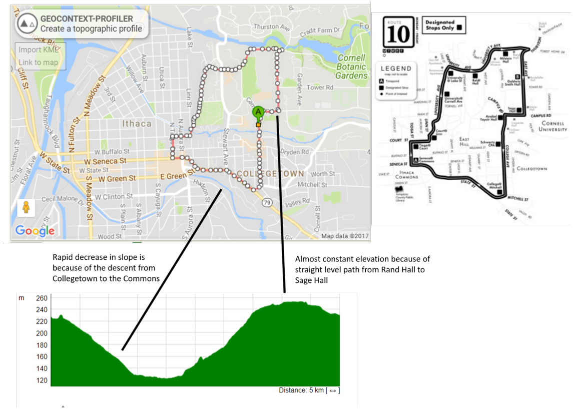


Figure 2: Route 10

What you see above on the far right is the route map for Route 10. On its immediate left is the set of plotted points that closely mirrors the path. Below this is the elevation profile, mapping points from the beginning at point A and ending right before it (this is a loop, like most TCAT routes). This matches our expectations for the declines and inclines for this route, given the large elevation change between campus and the Commons.

4.2 Data cleaning

These "curves" were downloaded as CSV files, and after some manipulation, we can obtain a 2 column matrix, with columns for the distance and the corresponding elevations. The corresponding graph for Route 10 is illustrated below:

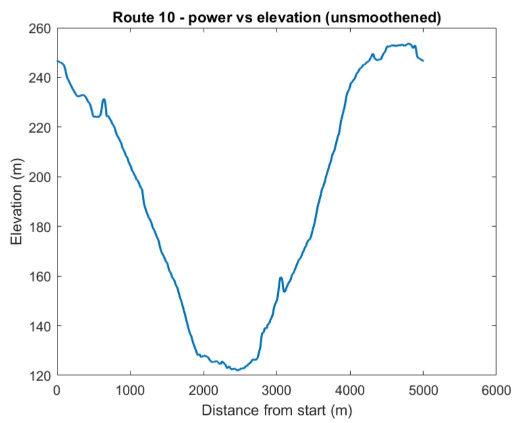
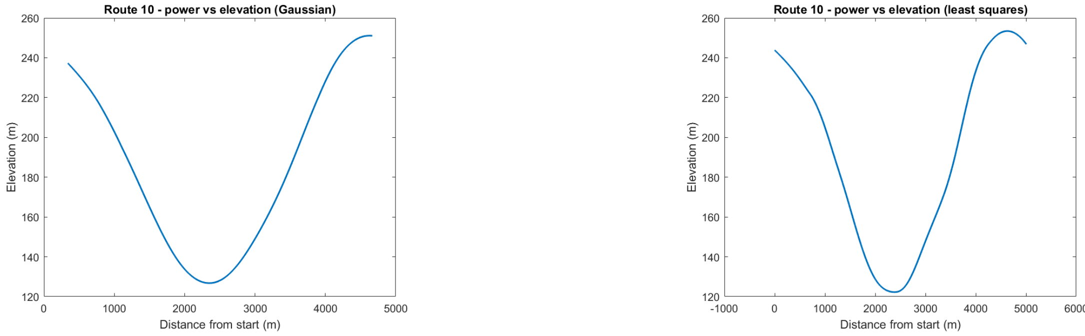


Figure 3: Route 10: Elevation vs distance from start

Since in the real worlds, roads have a continuous slope, we would like a smoother curve. To that extent we smoothen the elevation profiles using a Gaussian weighted average and least squares regression [2] using MATLAB's built-in functionality.

[1]The website allows you to make topographic profiles anywhere on the Earth

[2] For the function `smoothdata`, the method `'gaussian'` refers to the Gaussian-weighted moving average, over each interval while `'loess'` refers to the use of quadratic least-squares regression over each interval of the curve.



(a) Gaussian (b) Least squares

Figure 4: Smoothening the curve

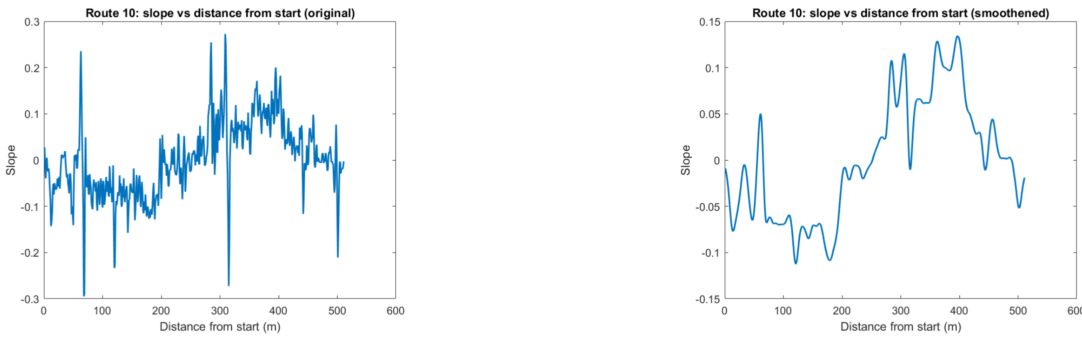
As can be seen from the above curves, while the Gaussian weighted average renders the curve too smooth, and local perturbations are lost. This would be inaccurate, however the curve produced by the least squares method also smoothens the curve, while preserving local disturbances (to check this notice the difference in the graphs from x = 2000 and x = 3000).Thus, it makes sense to use the least squares method to smoothen the graph.

4.3 Extracting the terrain gradient

We had started this data collection process to obtain the slope for each interval. Now that we have elevation data, we can do this very easily by using the concept of the slope:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

where m is the required slope, y_1 and x_1 are the elevation and distance components of the first point respectively, while y_2 and x_2 are the elevation and distance components of the next point respectively.
The components are smoothened beforehand (as shown above) to obtain a nice slope curve. This is once again plotted against the original curve to show the differences in the curve:



(a) Original (b) Smoothened

Figure 5: Smoothening the curve

4.4 Calculating the power at each interval

Now, that we have the intervals and their slopes, we can explain our graphically. We represent each discrete interval as a free body diagram, like the one pictured below. The bus is the block, and the information about the triangle is given by the data we have collected about each route.

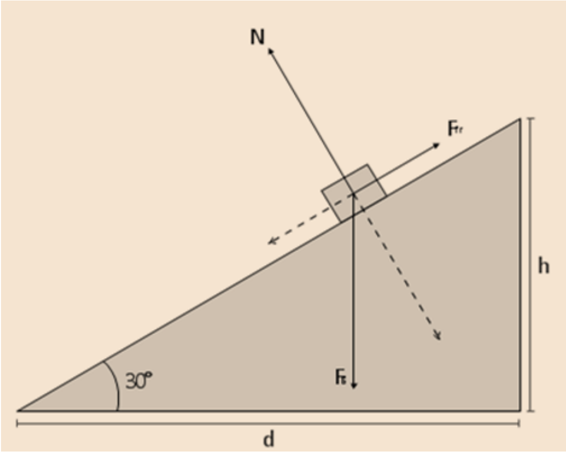


Figure 6: Free body diagram

We make the following assumptions to aid our calculations: there is no friction, and the bus travels at a constant velocity v (in $\frac{m}{s}$) regardless of the slope of the triangle. Let's start by calculating the time taken over each interval, since that will be useful in our calculations. We know the velocity v and the distance covered d we can solve for the time taken t_i on the interval d_i with the formula $t_i{=}\frac{d_i}{v_i}$. Given these assumptions we know from physics that the power P_i in Watts (W) needed to move our bus of mass M (kg) up a hill of height h_i and length d (m) at a constant velocity v ($\frac{m}{s}$), given that the acceleration due to gravity is g ($\frac{m}{s^2}$) is given by

$$P_i = M * g * \frac{h_i}{d} * v$$

Since M,v and g are constants over the entire loop and h_i is constant over each interval i, we know that P_i is constant over each interval.

4.5 Calculating the Total Energy used

Given information about the power outputted over each interval, we are interested in the energy E_i in kWh needed to power the bus along each block. We convert t_i from seconds to hours and P_i from W to kW to find energy in kWh:

$$E_i = \int_0^{\frac{t_i}{3600}} \frac{P_i}{100} dt = \frac{P_i}{1000} * (\frac{t_i}{3600})$$

.

Let us first consider the case where the bus is traveling on an incline with positive slope.

We are now interested in how much fuel a bus running on diesel would need to cover an interval. If we assume (as per our parameter regarding conversion of chemical energy to kinetic energy of motion) that 7% of the energy produced by the engine on the interval i, E_{ei} goes towards powering the motion of the car then we know that over an interval i, $E_{ei} = \frac{E_i}{0.07}$. Assuming the conversion rate from energy to diesel in gallons is $C_{diesel} = 1/40.28$, we know the total number of gallons used over that interval is given by

$$G_i = E_{ei} * C_{diesel}$$

If a hybrid bus is running on battery, we are interested in how much energy is lost from the battery. The battery is more efficient, and 23% of power generated from the battery powers the motion of the car. Therefore the total amount of energy in kWh produced from the battery on the interval i is given by $E_{bi} = \frac{E_i}{0.23}$. This is also the amount by which the battery of the electric vehicle is depleted on the interval i. Therefore, if the battery power at time i is given by B_i, the battery at time $i+1$ is given by

$$B_{i+1} = B_i - E_{bi}$$

.

Let us now consider the case when the bus is traveling down an incline with negative slope.

Since there is no friction, the bus would immediately accelerate down the hill because of gravity. However we assume that velocity is constant, which means that the acceleration and therefore the power needed to travel along this interval is negative. To see this notice that $P_i = M * g * \frac{h_i}{d} * v$, with h_i<0. This means that in order to keep a constant velocity v, we must decelerate by using the brakes to absorb the excess power produced. If the bus is a diesel bus, this means that the total number of gallons used over this interval is by $G_i = 0$, regardless of whether the bus is diesel powered or a hybrid. If the bus is a hybrid the regenerative brakes charge the battery, and transfer a fixed percentage $r = 20\%$ of the energy absorbed from the brakes. This means that for a hybrid bus traveling downhill

$$B_{i+1} = B_i + r * P_i$$

4.6 Putting it all together: simulating a single loop

Now consider which power source is used on each interval. If we are using a diesel powered bus then each interval is powered by diesel. If we are using a hybrid bus, then we will only use the battery if the slope of the interval is lower than the maximum slope threshold for electric power, and if the battery level B_i at the beginning of the interval i is above a minimum charge level C_{min}. If either of these conditions fail we use diesel.

Summarizing, we have the following information from our data at each point i: height h_i and distance d of the interval, energy required to cross the interval E_i, battery power B_i, battery power used crossing the interval E_{bi}, diesel used crossing the interval G_i. When we simulate a diesel truck the only variable we need to maintain is G_i. Therefore let $diesel_i = G_i$. We we simulate a hybrid truck we need to simulate the diesel used, the electric power used, and the total battery at each step.Let $hybrid_i = [Ghybrid_i, B_i, E_bi]$.

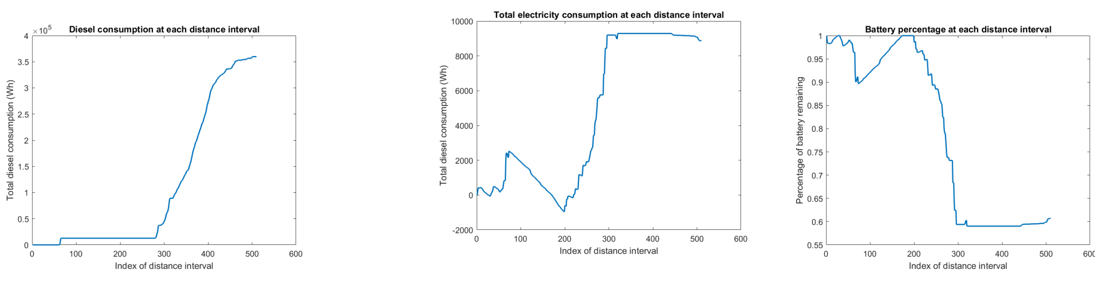
Then

if $(0 < slope_i < Max_{slope}$ and $B_i > Min_{charge})$ or $(0 > slope_i)$
——— $Ghybrid_i = 0$, $E_{bi} = E_{bi}$, $B_{i+1} = B_i - E_{bi}$.

Else
——— $Ghybrid_i = G_i$, $B_{i+1} = Bi$, $E_{bi} = 0$.

Now that we know how to calculate the energy and fuel used on each sub-interval of the route, and we have rules for when a hybrid truck uses battery power and when it uses diesel we are ready to calculate the energy and fuel used on along the entire route. In case of the diesel truck total diesel used $diesel_{tot} = \sum_{i=1}^{n} G_i$.

In case of the hybrid truck total diesel used is given by $hybrid_{tot} = \sum_{i=1}^{n} Ghybrid_i$.
Here are plots for this simulation on Route 10:

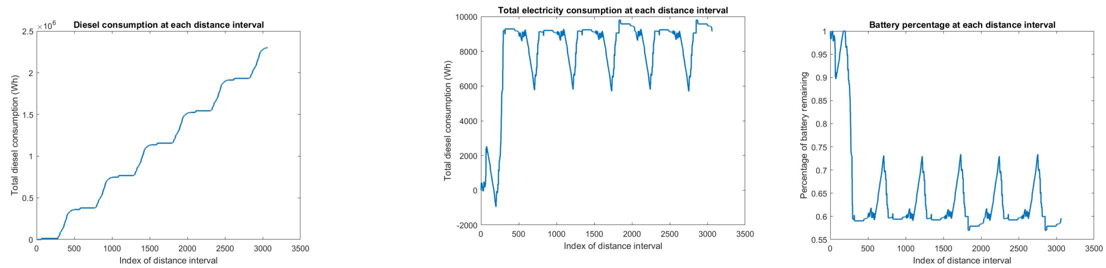


(a) Diesel energy consumption (b) Electrical energy consumption (c) Battery level

Figure 7: Route 10 - Simulation on a single loop

Some points to consider from this graph:

- Graph (7a) shows the total diesel consumption along the route up to that interval. Graph (7b) shows the same for the electricity consumption, while the last graph (7c) shows the percentage of battery remaining (the minimum level for which is 60%). Notice that while fuel used is always increasing, energy consumption and battery level both increase and decrease depending on whether the brakes are being used.
- The bus uses electricity initially, while battery power is high and gradients are low.
- At a little before 100th interval, gradients increase and the bus now switches to using diesel, which is noticeable in graph (7a) point.
- A little after the 100th interval, the battery is again in use: notice the increase in electrical energy consumption, the decrease in battery level and the. When the battery reaches its minimum level, the bus switches to diesel, and the graphs for electricity and battery used become relatively flat.



(a) Diesel energy consumption

(b) Electrical energy consumption

(c) Battery level

Figure 8: Route 10 - Simulation on 6 loops

Next, we simulate 6 loops of the same route: this route is as simple as the reader would expect! Some salient features of the above plots (6 iterations of the Route 10 trip):

- Except for the first iteration for each graph, the rest look identical! This seems to indicate the energy consumption reaches a periodic cycle. To see this, consider the first graph. When we simulated one iteration we noticed a "step-shape" in figure (7a). In figure (8a) There are 6 such "steps", resulting in 6 iterations.
- The reason the first iteration is different is because we initialize the simulation with a fully charged battery. At the end of the first iteration, the battery is almost empty. Because of this, we expect to see different behavior for the 2nd iteration. During the 2nd iteration, because the battery is not able to charge to full capacity, it is depleted and reaches its minimum state of charge. This behavior propagates throughout the other iterations, giving the periodic behavior mentioned above.

4.7 Finding the optimum allocation using Linear Programming

Allocating the hybrid electric buses between routes should exactly satisfy each route's bus requirement, while fulfilling the remaining demand with regular diesel buses. The table below summarizes the number of buses needed concurrently on each route, the operating duration in hours of each route, and the number of cycles a bus completes per day, based on TCAT's bus schedules [13].

We assume that the number of active buses is the minimum number of buses needed to fulfill the route's schedule. For example, on route 82, a bus departs the Hasbrouck Apartments at 7:43am and eventually returns at 8:33am. In that time interval, buses leave the Hasbrouck Apartments at 7:53am, 8:03am, 8:13am, 8:23am, and 8:33am. This pattern continues for the rest of the day. Therefore, there must be a minimum of five buses needed on route 82 to allow this schedule.

We also assume that the number of cycles a bus completes per day is the daily operating duration of the route divided by the cycle time. The daily operating duration is the time that the last bus of the day makes a stop minus the time that the first bus of the day makes a stop. For example, on route 10, since the first bus stops at the commons at 7:12am and the last bus of the day stops at 8:17pm, the daily operating duration is 13 hours and 5 minutes. The cycle time is determined by examining the time it takes for a bus to complete one cycle. For example, on route 10, buses that leave the Commons return 19 minutes later.

	Active Buses		Operating Hours		Cycle Time (min)		Cycles per Day	
Route	Weekday	Weekend	Weekday	Weekend	Weekday	Weekend	Weekday	Weekend
Route 10	2	0	13.08	0	19	0	41	0
Route 11	2	2	14.22	14.08	51	51	17	14
Route 15	1	1	13.57	12.57	34	24	24	20
Route 17	1	1	21.53	20.53	20	20	65	60
Route 81	2	0	12.43	0	30	0	25	0
Route 82	5	0	12.00	0	60	0	12	0

We then determine the amount of gas used by a hybrid electric bus and a regular diesel bus per cycle for each route by simulating 20 cycles and calculating the average gas usage per cycle. The following table summarizes all the variables required for the linear program.

Variable	Definition
d_i	Buses needed on route i
x_i	Hybrid buses assigned to route i
$regulargas_i$	Energy used by regular diesel bus on one cycle of route i.
$hybridgas_i$	Energy used by hybrid bus on one cycle of route i.
c_i	Cycles made each day on route i per bus.
R	$10, 11, 15, 17, 81, 82$

The optimal allocation of the eight hybrid electric buses is the one that maximizes the total daily gas savings. This is determined using **linear programming**, with the following objective function.

$$\sum_{i \in R} -savings_i(x_i)$$

$$savings_i(x_i) = x_i * (regulargas_i - hybridgas_i) * c_i$$

By minimizing the objective function, we are maximizing the total savings in gas across all routes. The linear program minimizes this objective function by changing the values for x_i $\forall i \in R$, within the constraints below.

$$0 \leq x_i \leq d_i \; \forall i \in R$$

$$\sum_{i \in R} x_i = 8$$

5 Simulation Algorithm

```
Distance = an n dimensional column vector containing information
about the length of each subinterval i.

Height = a n dimensional column vector containing information
about the height of each subinterval i.

Slope = Height/Distance

Time = Distance/v

Power_needed = (Height./Distance)*v*m*g

Battery_charge = an n dimensional column vector to be updated
in a for loop, used to keep track of battery power
```

```
For each subinterval i

..... If Power_needed > 0

..... ..... If Battery_charge(i) > Min_charge and slope < max_slope threshold

..... ..... ..... Fuel_needed=0

..... ..... ..... Electricity_needed = (Power.*Time.*1/electric_efficiency)

..... ..... ..... Battery_charge(i+1)=Battery_charge(i)-Electricity_needed

..... ..... Else

..... ..... ..... Fuel_needed = (1+battery_recharge_rate)
..... ..... .....  (Power.*Time.*1/engine_efficiency) *(conversion_rate)

..... ..... ..... Electricity_needed = 0

.....  If Power_needed < 0

..... .....  Fuel_needed=0

..... ..... Electricity_needed=(Power.*Time.*regeneration_rate)

..... .....  Battery_charge(i+1)=Battery_charge(i)-Electricity_needed
```

6 Model Predictions

The following table summarizes the diesel energy usage by route for hybrid electric buses and for regular diesel buses. The elevation gain is calculated by subtracting the lowest elevation from the highest elevation.

	Diesel Used per Cycle (gallons)				
Route	Hybrid	Regular	Distance (miles)	Elevation Gain (feet)	Energy Saved (%)
Route 10	1.9	2.0	3.1	432	6.47%
Route 11	2.3	2.5	4.6	493	6.09%
Route 15	1.0	1.1	4.3	42	8.04%
Route 17	0.7	0.8	3.3	35	9.47%
Route 81	0.9	1.0	2.8	119	8.54%
Route 82	2.3	2.4	7.5	139	6.20%

With similar elevation gains, route 10 and route 11 tend to use gas proportional to the distance of travel. The same observation holds for route 15 and route 17, as well as route 81 and route 82. When comparing route 17 and route 10, both have similar distance traveled, but the elevation gain of route 10 causes buses to use more than twice the fuel. The energy savings from switching to a hybrid bus center around 7%. The table below summarizes the fuel efficiency in miles per gallon.

	Fuel Efficiency (mpg)	
Route	Hybrid	Regular
Route 10	1.6	1.5
Route 11	2.0	1.9
Route 15	4.1	3.8
Route 17	4.4	4.0
Route 81	3.2	2.9
Route 82	3.3	3.1

These values are fairly realistic, with hybrid buses and regular diesel buses typically achieving 4.48

mpg and 3.83 mpg, respectively [3]. Next, using the estimated fuel consumed per cycle for each route, the linear program determined the following optimal allocation of hybrid buses.

Route	Optimal Weekday Hybrid Buses	Optimal Weekend Hybrid Buses
Route 10	2	0
Route 11	2	2
Route 15	1	1
Route 17	1	1
Route 81	2	0
Route 82	0	0

The fuel savings of this allocation compared to an allocation of all regular diesel buses is 27.22 gallons. This contributes to an annualized savings of 9,934.2 gallons or $28,630 of diesel [16]. We hypothesize that hybrid usage on route 82 is less optimal than the others because the energy saved is relatively low and the operating duration is the lowest among all routes. The allocation of hybrid buses on the weekend is trivial, as only route 11, route 15, and route 17 run on weekends and hybrid buses always acheive greater efficiency on these routes. The table below summarizes the worst allocation of hybrid buses.

Route	Least Optimal Weekday Hybrid Buse Allocation
Route 10	0
Route 11	0
Route 15	1
Route 17	0
Route 81	2
Route 82	5

The fuel savings of this allocation compared to an allocation of all regular diesel buses is 15.42 gallons, which is 11.80 gallons less than our optimal allocation. Reallocating from the least optimal to the most optimal allocation contributes to an annualized savings of 4,306.4 gallons or $12,411 of diesel [16]. In assessing the affect of winter on our bus allocation, we lowered the maximum electric engine slope threshold by 40%. This caused no change in our optimal bus allocation.

7 Robustness Testing

To test the sensitivity of our model to certain parameters, we conducted a set of trials in which we varied one parameter at a time by a certain percentage, computing the percentage of fuel saved by a hybrid bus. The cell values in the tables below measure

$$(1 - \frac{E_h}{E_d}) * 100$$

where E_h is the diesel energy consumed by the hybrid bus, while E_d is that consumed by the regular diesel bus.

We chose two particular routes to test this on: Route 17, which commutes around the Commons and thus has a *relatively level elevation profile*, and Route 10, which commutes from the Commons to Cornell, and has a *steeper elevation profile*. The smoothened elevation profiles are given below for reference:

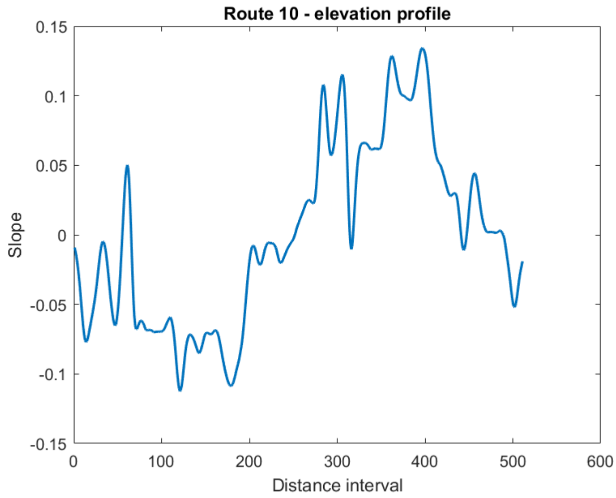


Figure 9: Route 10

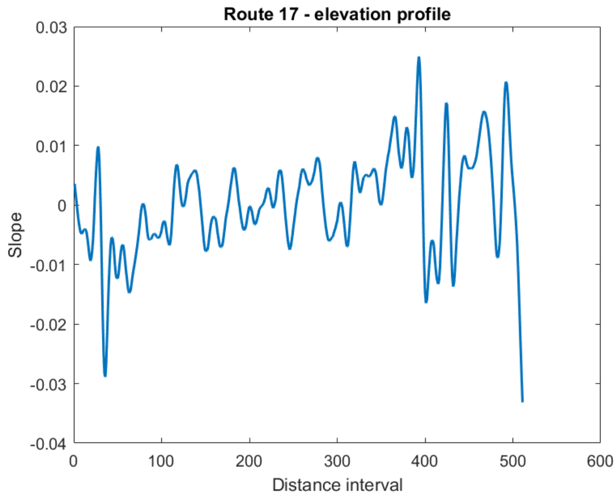


Figure 10: Route 17

As the reader can see, the magnitude of the slopes on route 10 is far greater than the magnitude of the slopes on route 17.

Route 10

Percentage Change	Slope Threshold	Brake Regeneration Rate	Electrical Engine Efficiency	Diesel Engine Efficiency
-30	5.39	4.09	3.86	5.41
-20	5.40	4.63	4.49	5.41
-10	5.41	5.15	5.11	5.41
10	5.41	5.66	5.71	5.41
20	5.41	6.18	6.31	5.41
30	5.40	6.69	6.92	5.41

Route 17

Percentage Change	Slope Threshold	Brake Regeneration Rate	Electrical Engine Efficiency	Diesel Engine Efficiency
-30	6.53	5.19	4.63	6.53
-20	6.53	5.71	5.37	6.53
-10	6.53	6.26	6.15	6.53
10	6.53	6.82	6.94	6.53
20	6.53	7.37	7.70	6.53
30	6.53	7.92	8.48	6.53

Some conclusions we can draw from the above tables:

- By looking at the table, the reader might see that percentage of fuel saved by a hybrid bus does not change as a function of Diesel Engine Efficiency. Is this a problem? No, because we are modifying the engines so that, when they are running on diesel, the efficiency of both the hybrid and diesel buses are the same. Thus, it is no surprise that the percentage of fuel saved remains the same, since the quantity of fuel used by each bus increases by the same amount in both cases.
- As expected, an increase in either the brake regeneration rate or electrical engine efficiency increases fuel savings, as both of these parameters are intricately tied to how the electrical engine operates and making these better will have a direct impact on emissions.
- Fuel efficiency as a function of the maximum slope threshold for Route 17 is constant; however, this is not the case for route 10. This is explained by the elevation profiles above. Since Route 17 does not seem to go through any large changes in elevation, it does not make a difference if we change the slope threshold because the threshold is never reached.

So far we have assumed that the effect of the diesel engine in charging the battery of a hybrid bus is negligible. Let us assume that the engine produces engine produces $\%C_{rate}$ more power than it usually would any time the diesel is being used, and that $\%C_{rate}$ power from the engine was being transferred directly to the battery. Then the percentage of fuel saved along each route for different values of C_{rate} is given by:

Route 10

Recharge Rate	Improvement
0	4.92
0.001	5.13
0.01	7.02
0.04	12.58
0.08	18.66
0.12	23.56

Route 17

Recharge Rate	Improvement
0	5.59
0.001	5.74
0.01	7.62
0.04	13.19
0.08	19.20
0.12	24.11

We notice that the more diesel is used for charging the battery, the more fuel is saved. This likely has to do with the fact that we assume that the electric engine is significantly more efficient than the electric engine at transforming power into kinetic energy. Therefore using more diesel to charge the battery at a greater rate is more efficient than using diesel to power the car.

8 Strengths and Weaknesses

8.1 Strengths

- **Solid data collection and cleaning strategies**: The data for the model has been collected very carefully using online software and tables from the TCAT website. The results have been checked against the actual map routes verified. The data is also cleaned thoroughly by smoothening it to both remove noise and simplify the trends in the data.
- **Real-life simulation**: The model considers most of the physical factors that go into determining the power and efficiency of vehicles. A lot of research has also been put into finding sound values for the parameters.

- **Simplicity**: The model has been simplified and broken down into logical blocks that build off of each other. The manuscript has also been formatted into sections in order to convey this effectively.
- **Visualizations**: We include graphs and tables, complete with explanations of what they contain whenever possible. In this way we allow the reader to interpret the results more visually.

8.2 Weaknesses

- **Too many parameters**: Our bottom-up approach to building this model includes many parameter values that each may introduce error into the actual answer. The presence of too many parameters is generally not desirable, but in-order to maintain a semblance of realism it was necessary.
- **Too many assumptions**: In order to focus on the problem at hand, we had to ignore certain factors that exist in real-life such as traffic and bus stops. While we realize that real life complications do play a role in what we are modeling, we believe that our approach considers the main factors involved and gives dependable answers.

9 Future Work

- **Varying the velocity**: Currently, we assume a fixed velocity. A better approach would be to use varying velocity using acceleration produced by the power of the engine, and speed zones. Varying velocity would also imply varying friction so adding friction to such a model would change the result both qualitatively and quantitatively.
- **Modeling bus stops**: Bus stops are a major part of any route, and our model currently ignores this. We could improve our model by considering the extra fuel needed because of starting and stopping.
- **Changing the route map**: Since we focused on optimizing the fuel efficiency on the existing routes, we did not propose any changes in the routes. However, we do have a proof of concept: Consider each stop on a route as a node of a graph and connect the stops with edges having weights as the total energy consumption between the points. We can add a "penalty" factor if this stop is too far away from the others, or at a very steep location. Given that, we can solve this problem using some standard shortest path algorithm like Dijkstra's to find the optimal path. The options can then be considered for changing the route map with minimum inconvenience.

10 Conclusion

The optimal allocation of the hybrid electric buses during the week is to place two on route 10, two on route 11, one on route 15, one on route 17, one on route 81, and none on route 82. This allocation does not change during the winter. The optimal allocation of the hybrid electric buses during the weekend is to place two on route 11, one on route 15, and one on route 17. This is a trivial solution because only five buses are required at night, and hybrid buses are more efficient than diesel buses on routes 11, 15, and 17.

According to our calculations, this optimal allocation will result in $12,411 in saving compared to the least optimal allocation of hybrid buses, and will result in $28,630 in savings when compared to an allocation of 0 hybrid buses to the routes. Furthermore, our allocation will maintain the current level of service along each route at all times, which will result in happy patrons and lower emissions!

11 Appendix A: Non-technical letter

Ithaca Journal Media
123 W. State / Martin Luther King Jr. Street
Ithaca, NY 14850

Dear readers of the Ithaca Journal,

We write to you today to further expand upon the conclusions we have recently drawn in our paper *Optimum Allocation of TCAT Hybrid Buses to Maximize Fuel Efficiency.* In the paper we look at different ways of optimally allocating our eight hybrid TCAT buses to routes 10, 11, 15, 17, 81, and 82 in order to minimize the total amount of fuel used. We did this by finding information about the distance travled and elevation gain for each route, using this information to calculate the total amount of fuel used by a hybrid bus and a diesel bus along each route.

We then used information from the TCAT schedule in order to determine the total number of buses needed on each route. In order to make sure that no passengers were left stranded or underserved, we imposed the additional requirement that no bus schedule should change as a result of the optimization. Our model suggested placing the hybrid buses on Routes 10, 11, 15, 17, and 81. This will save $28,630 in gasoline while reducing our greenhouse gas emissions.

While this is our recommendation, it is important to mention that modeling problems are not a precise science. The real world is extremely complicated and our simulations make many assumptions. That being said, we are proud of the work we've done and this is the best model we have devised in our research. We hope it helps the TCAT to optimize its fuel expectations.

Best,

Raghav Batra, Michael Galbato, Red Giuliano

12 Appendix B: Codebase

GitHub link: to take a closer look or look at the CSV data.

```
%Parameters: mass (m, kg), acceleration due to gravity (g, m(s^-2)), maximum power
%(p_max, W), max_speed (v_max, ms^-1), max_electric slope (max_s, W), brake
%conversion (c_rate)
%(GOING TO ADD LATER) maximum charge (c_max, J or MWH), minimum charge
%(c_min J or MWH), regeneration rate (r, % of power),
% fuel conversion parameters (d_conv, e_conv, %of power), parameters (c_min J or MWH)

m=13409; %kg, hybrid; regular 11364

m_regular=11364;

g=9.8; %ms^-2

p_max=290*10^3;%Watts, max power that can be outputted total

a_max=0;

v_max=48000/3600;%meters per second

params=[m,g,p_max,a_max,v_max];
%engine parameters

c_max=25*10^3; %kwh
c_min=0.6*c_max; %kwh
d_conv=0.07;
e_conv=0.23;
max_s=0.0856;
r=0.2;
c_rate=0;
engine_params=[c_max, c_min, d_conv, e_conv, max_s, r, c_rate];
%Equations: velocity, acceleration, power, position, slope. Position and
%slope are known

%Intiial parameters velocity, accelleration, power, slope and distance of interval at an index (t)
%velocity is initialiazed by velocity after one second of acceleration.

t=1;
v_t=v_max;
p_t=v_t*m*g*slope(t,1);
a_t=0;


%Final computations for fuel consumption, total fuel used for hybrids are in
%dieselstats(1,:). total fuel used for diesel busses are in
%dieselstats(2,:). Columns represent routes in ascending order.

dieselstats=zeros(2,6);

[avg_diesel_hybrid, avg_electricity, avg_diesel_diesel]=loop(params,  engine_params, 20, '10route.
dieselstats(1,1)=avg_diesel_hybrid;
dieselstats(2,1)=avg_diesel_diesel;
[h11, e11, d11]=loop(params, engine_params, 20,'11route_day_regular.csv' );
dieselstats(1,2)=h11;
dieselstats(2,2)=d11;
[h15, e15, d15]=loop(params, engine_params, 20,'15route.csv' );
```

```
dieselstats(1,3)=h15;
dieselstats(2,3)=d15;
[h17, e17, d17]=loop(params, engine_params, 20,'17route.csv' );
dieselstats(1,4)=h17;
dieselstats(2,4)=d17;
[h81, e81, d81]=loop(params, engine_params, 20,'81route.csv' );
dieselstats(1,5)=h81;
dieselstats(2,5)=d81;
[h82, e82, d82]=loop(params, engine_params, 20,'82route.csv' );
dieselstats(1,6)=h82;
dieselstats(2,6)=d82;

dieselstats=real(dieselstats)/1000


%Sensitivity/Robustness calculations
sensitivity_matrix=zeros(6,4,2);
testing=zeros(6,5);

testing(:,1)=linspace(0.7*engine_params(1,5),1.3*engine_params(1,5),6).';%max_slope
testing(:,2)=linspace(0.7*engine_params(1,6),1.3*engine_params(1,6),6).';%max_brakeregeneration
testing(:,3)=linspace(0.7*engine_params(1,4),1.3*engine_params(1,4),6).';%battery_efficiency
testing(:,4)=linspace(0.7*engine_params(1,3),1.3*engine_params(1,3),6).';%engine_efficiency
testing(:,5)=[0, 0.001, 0.01, 0.04, 0.08, 0.12];

slope_test=repmat(engine_params,6);
slope_test(:,5)=testing(:,1);
max_regeneration=repmat(engine_params,6);
max_regeneration(:,6)=testing(:,2);
battery_e=repmat(engine_params,6);
battery_e(:,4)=testing(:,3);
engine_e=repmat(engine_params,6);
engine_e(:,3)=testing(:,4);
charging_r=repmat(engine_params,6);
charging_r(:,7)=testing(:,5);

for i=1:6
    [dh, ae, dd]=loop(params,  slope_test(i,:), 20, '10route.csv');
    sensitivity_matrix(i, 1, 1)=dh/dd;
    [dh1, ae1, dd1]=loop(params,  slope_test(i,:), 20, '17route.csv');
    sensitivity_matrix(i, 1, 2)=dh1/dd1;
    [dh2, ae2, dd2]=loop(params,max_regeneration(i,:)  , 20, '10route.csv');
    sensitivity_matrix(i, 2, 1)=dh2/dd2;
    [dh21, ae21, dd21]=loop(params,  max_regeneration(i,:), 20, '17route.csv');
    sensitivity_matrix(i, 2, 2)=dh21/dd21;
    [dh3, ae3, dd3]=loop(params,  battery_e(i,:), 20, '10route.csv');
    sensitivity_matrix(i, 3, 1)=dh3/dd3;
    [dh13, ae13, dd13]=loop(params,  battery_e(i,:), 20, '17route.csv');
    sensitivity_matrix(i, 3, 2)=dh13/dd13;
    [dh4, ae4, dd4]=loop(params,  engine_e(i,:), 20, '10route.csv');
    sensitivity_matrix(i, 4, 1)=dh4/dd4;
    [dh14, ae14, dd14]=loop(params,  engine_e(i,:), 20, '17route.csv');
    sensitivity_matrix(i, 4, 2)=dh14/dd14;
    [dh5, ae5, dd5]=loop(params,  charging_r(i,:), 20, '10route.csv');
    sensitivity_matrix(i, 5, 1)=dh5/dd5;
    [dh15, ae15, dd15]=loop(params,  charging_r(i,:), 20, '17route.csv');
    sensitivity_matrix(i, 5, 2)=dh15/dd15;
```

```
end

sensitivity_matrix
%function that continues running the bus on a loop


function [avg_diesel_hybrid, avg_electricity, avg_diesel_diesel]=loop(params,  engine_params, num_
%engine parameters
c_max=engine_params(1,1); %kwh
c_min=engine_params(1,2); %kwh
d_conv=engine_params(1,3);
e_conv=engine_params(1,4);
max_s=engine_params(1,5);%maximum slope
r=engine_params(1,6);
c_rate=engine_params(1,7);

%other parameters
m=params(1,1);

g=params(1,2);

p_max=params(1,3);

a_max=params(1,4);

v_max=params(1,5);


[slope, position, distance]=finds_slope(filename);
slope1=slope;
position1=position;
distance1=distance;
% figure
% plot(slope)
% legend('slope')
t=1;
v_t=v_max;
p_t=v_t*m*g*slope(t,1);
a_t=0;


for trial=1:num_trials
    slope1=vertcat(slope1,slope);
    position1=vertcat(position1,position+position(1,1));
    distance1=vertcat(distance1, distance);
end


[v a p]=progression(v_t, a_t, p_t, slope1, distance1, params);


[diesel, electricity, current]=hybrid_conversion(v, a, p, distance1, slope1, engine_params);
[diesel_mm]=diesel_conversion(v, a, p, distance1, engine_params);

% figure
% plot(diesel)
% legend('diesel')
% figure
```

```
% plot(electricity)
% legend('elec')
% figure
% plot(current)
% legend('batt')
avg_diesel_hybrid=diesel(1,end-1)/(num_trials+1);
avg_electricity=electricity(1,end-1)/(num_trials+1);
avg_diesel_diesel=diesel_mm(1,end-1)/num_trials;
end

%function that converts data into fuel used for a diesel truck
function [diesel_mm]=diesel_conversion(v, a, p, distance, engine_params)
%calculate time taken per each step, given distance velocity and
%acceleration
time=zeros(1,length(a));
for i=1:length(a)-1
    time(1,i)=distance(i,1)/((v(1,i)+v(1,i+1))/2);
%     if a(1,i)>0
%        time(1,i)=(-v(1,i)+(v(1,i).^2-4*(1/2)*(a(1,i)).*distance(i,1)).^1/2)./a(1,i);
%     else
%        time(1,i)=distance(i,1)./v(1,i);
%     end
end

d_conv=engine_params(1,3);
c_rate=engine_params(1,7);


%total diesel used so far
diesel_mm=zeros(1,length(p));%total diesel at each time step

%for each discrete segment
for i=1:length(time)-1
    %if the slope is low enough use electric power
    if p(1,i)>0
        diesel_mm(1,i+1)=diesel_mm(1,i)+(1/d_conv)*p(1,i)*(time(1,i)/3600);
    end
    if p(1,i)<0
        diesel_mm(1,i+1)=diesel_mm(1,i);
    end
end


end

%function that converts power into fuel used, and electric power used for
%hybrid
function [diesel_mm, electricity_mm, c_current_mm, time]=hybrid_conversion(v, a, p, distance, slop
%calculate time taken per each step, given distance velocity and
%acceleration
time=zeros(1,length(a));
for i=1:length(a)-1
    time(1,i)=distance(i,1)/((v(1,i)+v(1,i+1))/2);
%     if a(1,i)>10^-1
%        time(1,i)=(-v(1,i)+(v(1,i).^2-4*(1/2)*(a(1,i)).*distance(i,1)).^1/2)./a(1,i);
%     else
%        time(1,i)=distance(i,1)./v(1,i);
%     end
```

```
end
%restating engine parameters

c_max=engine_params(1,1); %kwh
c_min=engine_params(1,2); %kwh
d_conv=engine_params(1,3);
e_conv=engine_params(1,4);
max_s=engine_params(1,5);%maximum slope
r=engine_params(1,6);
c_rate=engine_params(1,7);

slopes=slope(1:length(slope)-1);

slopes_logical=slopes<max_s;

%total diesel used so far
diesel_mm=zeros(1,length(p));
%total electricity used so far
electricity_mm=zeros(1,length(p));
%total battery charge left
c_current_mm=zeros(1,length(p));
c_current_mm(1,1)=c_max;

%for each discrete segment
for i=1:length(slopes_logical)-1
    %don't allow battery power to go over max power
    for j=1:length(c_current_mm)-1
        if c_current_mm(1,j+1)>c_max
            c_current_mm(1,j+1)=c_max;
        end
    end
    %if the slope is low enough use electric power
    if slopes_logical(i,1)==1 && c_current_mm(1,i)>c_min
        if p(1,i)>0
            %increment total electricity used, total battery remaining, and
            %total diesel used
            electricity_mm(1,i+1)=electricity_mm(1,i)+(1/e_conv)*p(1,i)*(time(1,i)/3600);
            c_current_mm(1,i+1)=c_current_mm(1,i)-(1/e_conv)*p(1,i)*(time(1,i)/3600);
            diesel_mm(1,i+1)=diesel_mm(1,i);
        end
        %if power outputted is negative charge the battery with brakes
        if p(1,i)<0
            %increment total electricity used, total battery remaining, and
            %total diesel used
            electricity_mm(1,i+1)=electricity_mm(1,i)+r*p(1,i)*(time(1,i)/3600);
            c_current_mm(1,i+1)=c_current_mm(1,i)-r*p(1,i)*(time(1,i)/3600);
            diesel_mm(1,i+1)=diesel_mm(1,i);
        end
    end
    %if no more battery or slopes are steep
    if slopes_logical(i,1)==0 || c_current_mm(1,i)<c_min
        %use diesel and charge batterry with engine if power needed is
        %positive
        if p(1,i)>0
            diesel_mm(1,i+1)=diesel_mm(1,i)+(1+c_rate)*(1/d_conv)*p(1,i)*(time(1,i)/3600);
            c_current_mm(1,i+1)=c_current_mm(1,i)+c_rate*(1/d_conv)*p(1,i)*(time(1,i)/3600);
            electricity_mm(1,i+1)=electricity_mm(1,i);
        end
```

```
        %charge batterry with engine if power needed is
        %negative
        if p(1,i)<0
            diesel_mm(1,i+1)=diesel_mm(1,i);
            c_current_mm(1,i+1)=c_current_mm(1,i)-r*p(1,i)*(time(1,i)/3600);
            electricity_mm(1,i+1)=electricity_mm(1,i)+r*p(1,i)*(time(1,i)/3600);
        end
    end
end

end


%function that generates velocity, accelaration, and power information at
%each node of interpolation

function [v a p]=progression(v_t, a_t, p_t, slope, distance, params)
%parameters

m=params(1,1);

g=params(1,2);

p_max=params(1,3);

a_max=params(1,4);

v_max=params(1,5);

%initialize vectors

v=zeros(1,length(slope));
a=zeros(1,length(slope));
p=zeros(1,length(slope));
v(1,1)=v_max;
p(1,1)=v(1,1)*m*g*slope(1,1);
a(1,1)=0;

for t=1:length(slope)-1
    values=update(v(1,t), a(1,t), p(1,t), t, slope, distance, params);
    v(1,t+1)=values(1,1);
    a(1,t+1)=values(1,2);
    p(1,t+1)=values(1,3);
end


end

%update rule function


function updated=update(v, a, p, t, slope, distance, params)

%parameters


m=params(1,1);
```

```
g=params(1,2);

p_max=params(1,3);

a_max=params(1,4);

v_max=params(1,5);

p1=slope(t,1)*v_max*m*g;
%if power outputted is max because of the slope then maintain max power and decrease velocity

% if v_max*(slope(t+1,1)*m*g)>=p_max %this will have to change when we add friction
%     p_1=p_max;
%     v_1=p_max/(slope(t+1,1)*m*g); %this will have to change when we add friction
%     a_1=0;
% end

%if power needed to get up the slope is not max then

% if v_max*(slope(t+1,1)*m*g)<p_max
%     %if velocity is 30 then allow power to vary in order to maintain velocity
%     if (v-v_max)>-1
%         v_1=v_max;
%         a_1=0;
%         p_1=m*g*slope(t+1,1)*v_1;
%     end
%     %if velocity is less than 30 the accelerate using the power not used to
%     %fight gravity
%     % in order to increase velocity to v_max
%
%     if v-v_max<-1
%         v_1=v+a*(distance(t,1)/v);
%         a_1=((p_max-v*(slope(t,1)*m*g))/((distance(t,1)/v)*2*m))^(1/2);
%         p_1=m*g*slope(t+1,1)*v_1;
%
%     end
% end

%returns velocity, acceleration and power for that time interval, and next
%time index
updated=[v_max, 0, p1, t+1];


end





function [slope, position, distance] = finds_slope(filename)
    % find slope of data in ‘filename‘

    % assumes first row contains header
    M = csvread(filename, 1);

    % first column has distances and second column has corresponding
    % elevation
```

```
    M2 = M(1:end - 1, :);
    M3 = M(2:end, :);

    delM = M3 - M2;

    % del elevation / del distance
    %slope = smoothdata(delM(:, 2), 'gaussian', 20) ./ smoothdata(delM(:, 1),'gaussian', 20);
    slope=delM(:,2)./delM(:,1);
    position=M(:,1);
    distance=(delM(:,1).^2+delM(:,2).^2).^1/2;
    % comment out if necessary
    %plot(M(:, 1), M(:, 2))
end
```

References

[1] Adrienne S
What is the Weight of a City Bus?
Retrieved from https://getawaytips.azcentral.com/what-is-the-weight-of-a-city-bus-12373480.html.

[2] Intercity Transit
Hybrid Bus Fact Sheet
Retrieved from https://www.intercitytransit.com/sites/default/files/2016-07/HybridFactSheet20120802.pdf.

[3] TCAT
TCAT 2014 Yearbook
Retrieved from https://www.tcatbus.com/content/uploads/2015/06/Download-2014-PDF.pdf.

[4] City of Ithaca
City of Ithaca
Retrieved from https://ecode360.com/8395575.
September 2017

[5] Antti Lajunen
Evaluation of battery requirements for hybrid and electric city buses
Retrieved from https://pdfs.semanticscholar.org/5484/5a891edbc79f4428d2d621f0626026b974cc.pdf.
May 2012

[6] Jürgen Garche et al
Lead-Acid Batteries for Future Automobiles
Retrieved from https://www.elsevier.com/books/lead-acid-batteries-for-future-automobiles/garche/978-0-444-63700-0.
March 2017

[7] University of Tennessee Chattanooga, Computer Science
Frequently Asked Questions (about electric & hybrid-electric buses)
Retrieved from https://www.utc.edu/college-engineering-computer-science/research-centers/cete/faq.php#top.

[8] Wikipedia
Electric bus
Retrieved from https://en.wikipedia.org/wiki/Electric_bus.

[9] Matthew Brown
Regenerative Braking Efficiency
Retrieved from http://large.stanford.edu/courses/2016/ph240/brown1/.
November 2016

[10] ProEV
Regenerative Braking Efficiency
Retrieved from http://proev.com/LLPgs/LLei0005.htm.
June 2016

[11] US Department of Energy
Where the Energy Goes: Gasoline Vehicles
Retrieved from https://www.fueleconomy.gov/feg/atv.shtml.

[12] CPTDB Wiki
Voith DIWA.5
Retrieved from https://cptdb.ca/wiki/index.php/Voith_DIWA.5.

[13] TCAT
Schedules
Retrieved from https://www.tcatbus.com/ride/schedules-fall-2017/.

[14] US Energy Information Administration
Energy Units and Calculations Explained
Retrieved from https://www.eia.gov/energyexplained/index.cfm?page=about_energy_conversion_calcul
.

[15] Benchmarking of Regenerative Braking for a Fully Electric Car
Benchmarking of Regenerative Braking for a Fully Electric Car
Retrieved from http://www.mate.tue.nl/mate/pdfs/12673.pdf.

[16] US Energy Information Administration
Weekly Retail Gasoline and Diesel Prices
Retrieved from https://www.eia.gov/dnav/pet/pet_pri_gnd_dcus_nus_w.htm.