EMTH171 Case Study 2 Report:

Introduction

The purpose of this case study is to produce an accurate model of the Fuji Conductor 2.1+ electrical bicycle that allows for graphs depicting its key variables under a variety of conditions to be produced. To model the key variables of the bicycle, the angular acceleration is found from a derived version of the equation below. This equation is the rearranged formula for moment of inertia.

$$\frac{d\omega}{dt} = \frac{T}{J} \tag{1}$$

The effective moment of inertia of the bicycle (J_{ef}) is given by the below equation. This is the moment of inertia of the wheel and motor in addition to the moment of inertia of the total mass around the wheel.

$$J_{ef} = J_w + {r_w}^2 M \tag{2}$$

The net torque (T) is the combined sum of the forwards torque produced by the motor and pedals, and the backwards torque. The backwards torque is from wind resistance (βu^2) and the downwards force on slopes $(Mg \sin \alpha)$. Rolling resistance and other small forces are considered a constant (f_{d0}) to simplify the model. Below is the equation for net torque.

$$T = T_m + \left(\frac{r_s}{r_p}\right) T_p - r_w(f_{d0} + \beta u^2 + Mg \sin \alpha) \tag{3}$$

The combination of equations (1), (2), and (3) derive the equation below.

$$\frac{d\omega_m}{dt} = \frac{1}{J_{ef}} [T_m + \left(\frac{r_s}{r_p}\right) T_p - r_w(f_{d0} + \beta u^2 + Mgsin\alpha)] \tag{4}$$

With angular acceleration, Euler's method can be used to find the angular velocity in steps of Δt.

$$\omega_{mi+1} = \omega_{mi} + \Delta t \left(\frac{d\omega_m}{dt}\right)_i \tag{5}$$

This allows for the velocity to be found, as the radius of the wheel is known. Using the trapezium rule, the distance travelled can be found through step size of Δt.

$$s_{i+1} = s_i + \frac{\Delta t[r_w(\omega_{mi+1} + \ \omega_{mi})]}{2} \tag{6}$$

Hence with angular acceleration, velocity, and distance, an accurate model of the bicycles path can be plotted and found. This report presents the results gained from modelling performance of the bicycle and discuss improvements to accuracy made through decreasing the step size of Euler's method.

Results

Task 0:

The aim of Task 0 was to model a flat road with no pedalling and no wind resistance over a 30 second interval. Equation (4) was simplified to:

$$\frac{d\omega_m}{dt} = \frac{1}{J_{ef}} (T_m - r_w f_{d0}) \tag{7}$$

A definite integral was found to confirm the reliability of the code when it was run. Below is the working used to derive the final velocity and distance through this method.

$$\omega_m(t) = \int_0^{30} \frac{1}{J_{ef}}(T_m - r_w f_{d0})\, dt \tag{8}$$

$$\omega_m(t) = 30\left[\frac{1}{0.5\ kgm^2 + (0.35\ m)^2 \times 95\ kg}(3.2\ NmA^{-1} \times 4\ A - 0.35\ m \times 2\ N)\right] = 29.9073\ rads^{-1} \tag{9}$$

$$v_m = \omega_m r_w = 29.9073\ rads^{-1} \times 0.35\ m = 10.4676\ ms^{-1} \tag{10}$$

To find the final distance the trapezium rule was used.

$$s_f = s + \Delta t\left(\frac{v_f - v_i}{2}\right) = 0\ m + 30\ s\left(\frac{10.4676\ ms^{-1} - 0\ m}{2}\right) = 157.0140\ m \tag{11}$$

The given script for Task 0 with additions made in appendix A was run to generate the plots displayed in Figure 1.

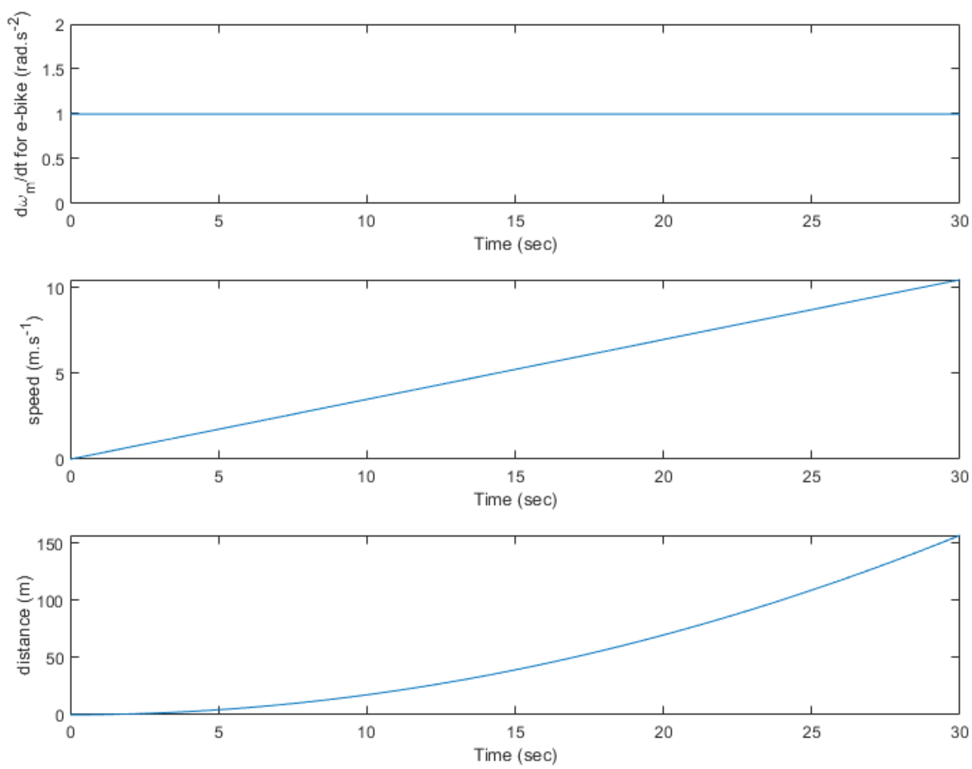

Figure 1 – Graphs depicting the angular acceleration (in $rads^{-2}$), the linear acceleration (in ms^{-2}), and the linear velocity (in ms^{-1}) respectively for an bicycle and rider on a flat surface with no wind resistance.

The angular acceleration of the wheel remains constant when there is no wind resistance. The linear velocity increases at a constant rate, resulting in a final linear velocity of $10.4676\ ms^{-1}$ at a distance of $157.0134\ m$ from the starting point.

As the angular acceleration was constant a large step size did not reduce the accuracy of the results found of angular acceleration and linear velocity but did decrease the accuracy of the trapezium rule distance which used the trapezium rule with the same step size and so a step size of $0.01\ s$ was used.

During this simulation, a fully charged battery of charge $Q_{Total} = 18\ Ah$ was used to provide power to the bicycle. To calculate the charge expended over the $30\ s$ period with a constant amperage of $4\ A$, the equation below was used.

$$Q_{Used} = I_m t = 4\ A \times \frac{1}{120}\ h = \frac{1}{30} Ah \tag{12}$$

This value was then used to find the percentage of the battery used during the trial.

$$\%Q_{Used} = \frac{Q_{Used}}{Q_{Total}} \times 100\% = \frac{{}^{1}\!/_{30}\ Ah}{18\ Ah} \times 100\% = 0.185\% \tag{13}$$

Therefore, the percentage of charge expended during the $30\ s$ trial was 0.185%.

Task 1:

The conditions for Task 1 were the same as Task 0 with the addition of wind resistance which was included in the calculations made. Equation (3) was simplified to only include relevant variables for this task.

$$\frac{d\omega_m}{dt} = \frac{1}{J_{ef}}[T_m - r_w(f_{d0} + \beta u^2)] \tag{14}$$

The script for Task 1 in Appendix A which included wind resistance was used to obtain graphs of the key variables of the bicycle during the simulation. These graphs can be seen in Figure 2.

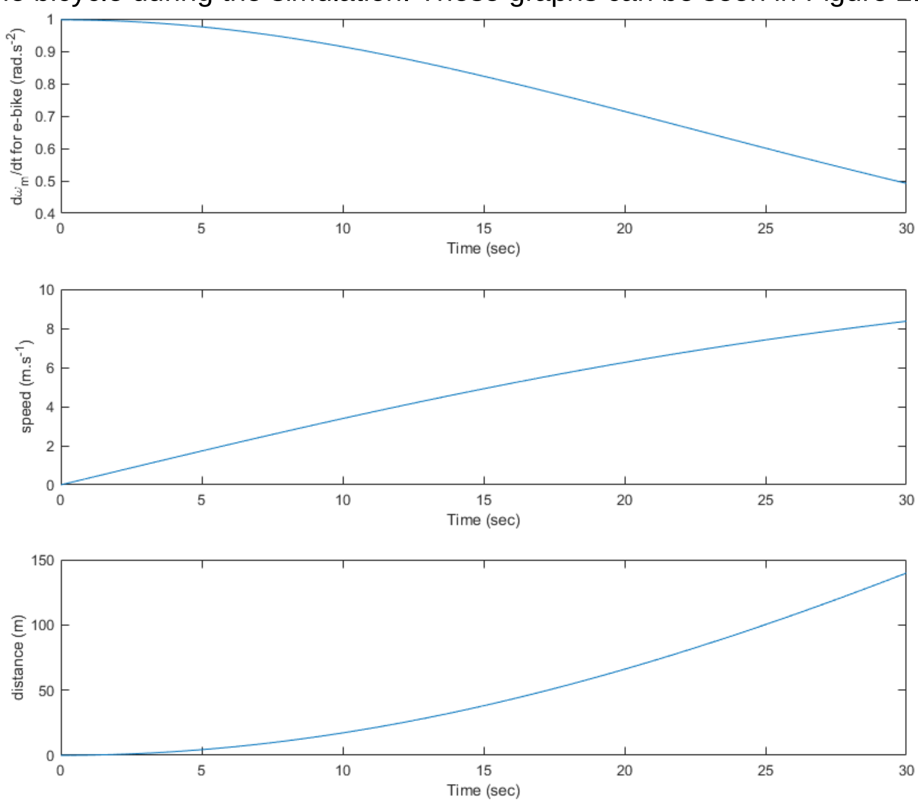

Figure 2 – Graphs depicting the angular acceleration ($rads^{-2}$), linear acceleration (ms^{-2}), and linear velocity (ms^{-1}) respectively for a bicycle experiencing wind resistance riding along a flat road.

When wind resistance was added in the simulation, the angular acceleration began to decrease over the 30 second interval towards $0\ rads^{-2}$. This led the linear velocity to reach an asymptotic velocity as there was no acceleration. This asymptotic velocity was calculated below.

$$u = \sqrt{\frac{1}{\beta}\left(\frac{I_m k_m}{r_w} - f_{d0}\right)} = \sqrt{\frac{1}{0.25\ Nm^{-2}s^2}\left(\frac{4\ A \times 3.2\ NmA^{-1}}{0.35\ m} - 2\ N\right)} = 11.7595\ ms^{-1} \tag{15}$$

This occurs when the torque from the motor is opposed by an equal and opposite torque from wind resistance and other forces. So, there is no torque driving the angular acceleration. The asymptote was not reached during the 30 second simulation, instead the final linear velocity was found to be $8.3670\ ms^{-1}$, and the bicycle travelled $139.8282\ m$.

The step size used to plot these graphs was $\Delta t = 0.01\ s$. As the angular acceleration was not constant to better approximate the graph smaller intervals of time should be used. Values greater than $1\ s$ did not provide an accurate graph and curve and therefore Δt was chosen as it was small enough to allow of an accurate graph to be obtained.

The battery charge used was the same as that calculated in Task 0 as neither the current nor time interval changed. The power delivered by the motor at 30 s was calculated below. Where $\omega_m = 24.0793\ rads^{-1}$ is angular velocity at 30 s.

$$P_m = T_m\omega_m = I_m k_m \omega_m = 4\ A \times 3.2\ NmA^{-1} \times 24.0793\ rads^{-1} = 308.2150\ W \tag{16}$$

Task 2:

The aim of Task 2 was to model the performance of the bicycle as it moved from a flat road to an incline. Table 1 displays the new data used for this simulation at both points.

Table 1 – The data used for the simulation script for Task 2.

	Mapped Distance (m)	Slope Angle (θ)	Motor Current (A)	Gear Radius (m)	Pedal Torque (Nm)
Flat Road	2000	0	2.0	0.030	8
Slope Road	4000	0.07	6.5	0.045	15

To reflect this change, a new script was developed for Task 2 as seen in Appendix A. As Task 2 provided no set time frame, the For loop used prior in Task 0 and 1 was changed for a While loop that ran until the necessary total distance had been travelled. A check was also made to determine if the bicycle had reached the slope, and upon doing so, the slope's variables were used.

Throughout the calculations a check to ensure the motor power was below the maximum of $500\ W$ was made. If it was above the maximum power the motor torque was decreased to its maximum possible. Because the torque was no longer a constant value the script implemented code to calculate the battery used over the time steps of Δt and found the total sum and hence total battery usage.

The time interval for Euler's Method used in this task was $0.01\ s$. This was found to be the most effective step size as it was small enough to accurately display the key variables over the period, while providing better performance for the script.

The script for Task 2 in Appendix A was run to produce the plots seen in Figure 3.

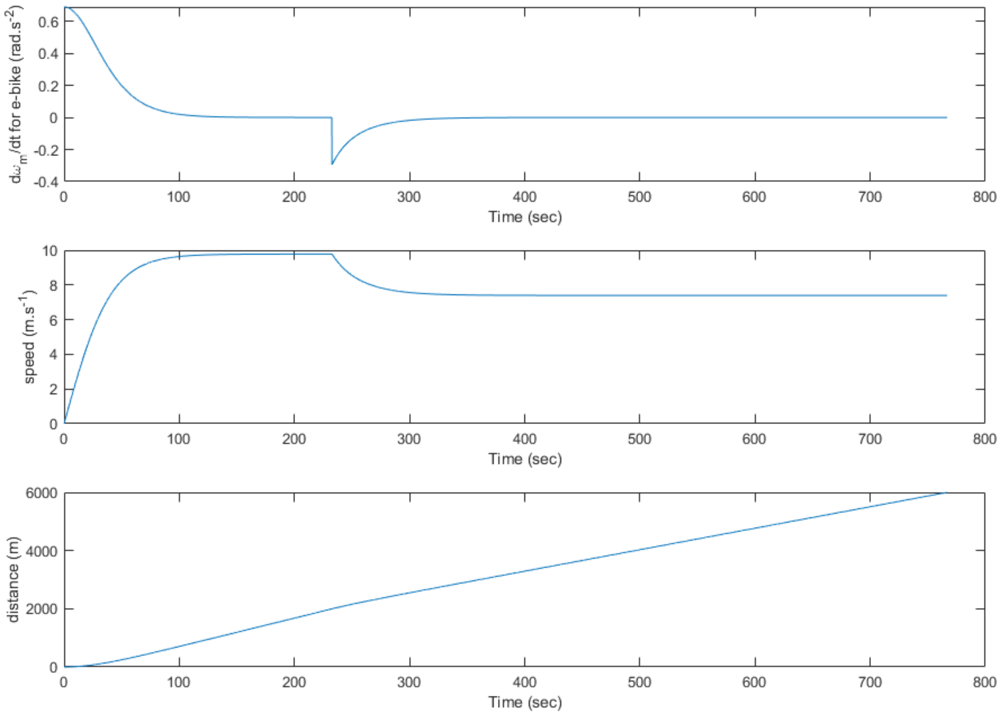

Figure 3 – Graphs displaying the angular acceleration ($rads^{-1}$), linear acceleration (ms^{-2}), and linear velocity (ms^{-1}) for a bicycle and rider travelling initially along a flat road, before climbing a slope of $0.07\ rad$.

The rider takes $766.81\ s$ to travel a mapped distance $6\ km$ from the starting distance, to the top of the rise. As the rider reaches the rise, their linear velocity is $9.7783\ ms^{-1}$. Upon reaching the top of the rise, the rider has decelerated to a final velocity of $7.3956\ ms^{-1}$. The total battery usage was found to be $6.0763\ \%$.

To find the power supplied by the rider at the beginning of the rise, the angular velocity of the pedals was found as seen below.

$$\omega_{p\ (Flat)} = \frac{\omega_m r_{s\ (High)}}{r_p} = \frac{27.938\ rads^{-1} \times 0.03\ m}{0.09\ m} = 9.3126\ rads^{-1} \tag{17}$$

This value was then used in the equation for power, alongside the torque applied to the pedals.

$$P_{Flat} = T_{p\ (Flat)}\omega_{p\ (Flat)} = \ 8\ Nm \times 9.3126\ rads^{-1} = \ 74.5013\ W \tag{18}$$

This process was repeated to find the power supplied by the rider during the sloped segment of the journey.

$$\omega_{p\ (Slope)} = \frac{\omega_m r_{s\ (Low)}}{r_p} = \frac{21.1304\ rads^{-1} * 0.045\ m}{0.09\ m} = 10.5652\ rads^{-1} \tag{19}$$

$$P_{Slope} = T_{p\ (Slope)}\omega_{p\ (Slope)} = \ 15\ Nm * 10.5652\ rads^{-1} = 158.4780\ W \tag{20}$$

To find the total power exerted by the rider during the simulation, these two values found for P_{Flat} and P_{Slope} were combined.

$$P_{Rider} = P_{Flat} + P_{Slope} = 74.5013\ W + 158.4780\ W\ = 233.9793\ W \tag{21}$$

The same process was used to find the power of the motor during the simulation.

$$P_{Flat} = T_{m\,(Flat)}\omega_{m\,(Flat)} = 6.4 * 27.938 = 178.8032\ W \tag{22}$$

$$P_{Slope} = T_{m\,(Slope)}\omega_{m\,(Slope)} = 20.8 * 21.1304 = 439.5123\ W \tag{23}$$
$$P_{Motor} = P_{Flat} + P_{Slope} = 178.8032 + 439.5123 = 618.3155\ W \tag{24}$$

As power is limited by the motor, $P_{motor} = 500\ W$, using this the percentage can be found.

$$\%\ P_{rider} = \frac{P_{rider}}{P_{rider} + P_{Motor}} * 100 = \frac{233.9793}{233.9793 + 500} * 100 = 31.88\ \% \tag{25}$$

Task 3:

The method used to find four effective variables, used assumptions made about the velocity during each section of the journey. The first step in the method used an equation expressing the fact the individual times for each section must be less than or equal to the maximum time allowed of $1800\ s$. This equation is shown below.

$$1800 \geq \frac{d_1}{v_1} + \frac{d_2}{v_2} \tag{26}$$

Two assumptions were made to derive suitable velocity values. Firstly, it was assumed both distances were $5\ km$ this was done to simplify the values used. It was also assumed that as $E = (T\omega)t$ and the torque needed to overcome opposing forces was greater on the slope than flat road as there was also the opposing force of gravity. Therefore, the gradient is greater during this section and so to minimize total energy the slope's velocity should be double that of the flat road, which has a lower gradient.

With these two assumptions the potential velocities chosen were $v_1 = 4\ ms^{-1}$ and $v_2 = 10\ ms^{-1}$ as they gave a time of $1750\ s$ which is below but close to the maximum time. For the equation $t = d/v$ the two times of each section were found below.

$$t_1 = \frac{25}{72} h \text{ and } t_2 = \frac{5}{36} h \tag{27}$$

Next as only $5\ \%$ of an $18\ Ah$ battery could be used, only $0.9\ Ah$ was available. An equation expressing this restriction and its relationship to the charges used during each section is shown below.

$$0.9 = t_1 I_{m1} + t_2 I_{m2}$$

To find values of current it was assumed that $I_{m1} = 0\ A$ as this allowed for more current to be used during the slope and therefore reduce the time and hence energy expended by the rider as the gradient of the flat road was lower and therefore had a relatively lower effect on the energy used. Using the times derived above the current for I_{m2} was found to be $I_{m2} = 6.48\ A$.

Finally, the velocities and currents derived above were used in the equation below to find the values of T_{p1} and T_{p2}.

$$T_m + \left(\frac{r_s}{r_p}\right) T_p = r_w(f_{d0} + \beta u^2 + Mg \sin \alpha) \tag{28}$$

$$T_{p1} = 6.3\ Nm \text{ and } T_{p2} = 10.03\ Nm \tag{29}$$

The code was run with these values. It was found that the chosen values were slightly too slow and therefore T_{p2} was increased by $1.57\ Nm$ so $T_{p2} = 11.6\ Nm$. With this adjustment made the final energy was found using the code for Task 3 in appendix A to be $E = 113.8\ kJ$. The journey of the bicycle is modelled below.

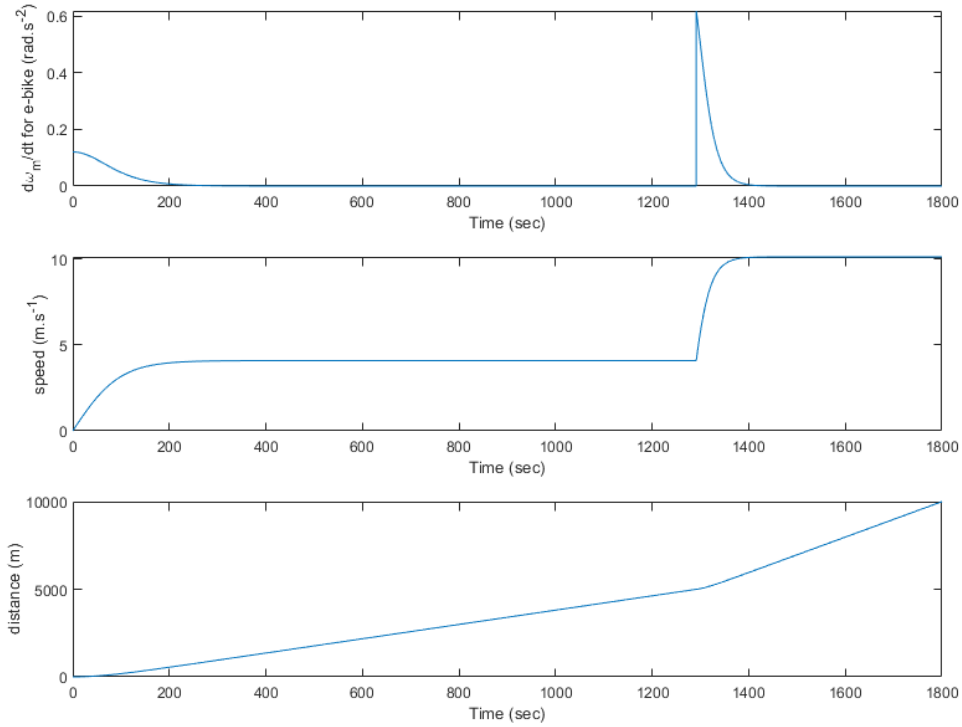

Figure 4 – Graphs displaying the angular acceleration ($rads^{-2}$), linear acceleration (ms^{-2}), and linear velocity (ms^{-1}) during a simulation for the model performance of the bicycle for Task 3.

Conclusion

The performance of the Fuji Conductor 2.1+ electric bicycle on flat roads and inclines was modelled in 4 different situations to find and plot key variables such as time, velocity, and distance. Below is a brief overview of these results found.

Task 0 found that over a $30\ s$ interval the bicycle could travel $157.0140\ m$ along a flat road when there was no wind resistance. A constant angular acceleration of $0.9969\ rads^{-1}$ and final velocity of $10.4676\ ms^{-1}$ was found.

In Task 1 with the addition of wind resistance, the acceleration decreases over the $30\ s$ interval, resulting in the velocity approaching an asymptote of $11.7595\ ms^{-1}$. This asymptote was not reached during the simulation however, and the final velocity was $8.3670\ ms^{-1}$ with the final distance from the start point being $139.8282\ m$.

Task 2 modelled a mapped $6\ km$ journey, it was found that the bicycle steadily decelerates to reach an asymptote at a velocity of $9.7783\ ms^{-1}$. Upon reaching a hill the velocity of the bicycle decreased to reach a new asymptotic velocity of $7.3956\ ms^{-1}$. The total time the journey took was $766.81\ s$.

Task 3's aim was to minimize the energy expended by the rider while riding a total mapped distance of $10\ km$ in $30\ min$ while using less than 5% of the total battery. Using a method of assumptions made about velocity and current over the journey values for current and pedal torque where found where the total energy expended by the rider was $E = 113.8\ kJ$.

Appendix A:

Task 0:

```
%{
Case Study #2 EMTH171  2020
E-Bike
Task 0 Description:
For this task there is no wind resistance. From a standstill, what final velocity
and distance travelled will the e-bike reach on a level road with no pedal
assistance and a constant 4A motor current over a total time of 30 secs?

Original code by: P.J. Bones   UCECE
Code edited by: John Elliott and Samuel Vallance
Last modified:  08/10/2020
%}
clear
clc
close all

%{
===========================================================================
                            | Variable Setup |
===========================================================================
%}
% … Code assigning values for Task 0 omitted
%{
===========================================================================
                 | Euler's Method and Speed Calculations |
===========================================================================
%}
% Iteratively compute u and s as the e-bike proceeds, advancing tdelta sec each
index = 1;
for t = 0:tdelta:(tf - tdelta)
   % Compute the motor acceleration at the start of the step
   domdt = (Tm - rw * fd0) / Jef;

   % Estimate the state at the end of the time step (n = 'next')
   omn = om + tdelta * domdt;       % Estimate om by Euler's method
   un = omn * rw;                   % Estimate u at end of step
   sn = s + tdelta * (un + u) / 2;  % Estimate s at end of step (Trap.)

   domdtarray(index) = domdt;       % Store current values
   uarray(index) = u;
   sarray(index) = s;
   index = index + 1;

   % Advance to next time step
   om = omn;
   u = un;
   s = sn;

end
% Store final values
domdtarray(index) = domdt;
uarray(index) = un;
sarray(index) = s;
```

```
%{
============================================================================
                      | Display Results and Graphs |
============================================================================
%}
% … Code plotting values for Task 0 omitted
```

Task 1:

```
% … Only Euler's method section was changed from task 0
%{
============================================================================
                 | Euler's Method and Speed Calculations |
============================================================================
%}
% Iteratively compute u and s as the e-bike proceeds, advancing tdelta sec each
index = 1;
for t = 0:tdelta:(tf - tdelta)
   % Compute the motor acceleration at the start of the step
   domdt = (Tm - rw * (fd0 + beta*u^2)) / Jef;

   % Estimate the state at the end of the time step (n = 'next')
   omn = om + tdelta * domdt;        % Estimate om by Euler's method
   un = omn * rw;                    % Estimate u at end of step
   sn = s + tdelta * (un + u) / 2;   % Estimate s at end of step (Trap.)

   domdtarray(index) = domdt;        % Store current values
   uarray(index) = u;
   sarray(index) = s;
   index = index + 1;

   % Advance to next time step
   om = omn;
   u = un;
   s = sn;
end
% Store final values
domdtarray(index) = domdt;
uarray(index) = un;
sarray(index) = s;
```

Task 2:

```
%{
Case Study #2 EMTH171  2020
E-Bike
Task 2 Description:
From a standing start on a level road. A motor current of Im = 2 A and a pedal
torque of Tp = 8 Nm (using high gear) is applied until the e-bike
reaches the start of a hill after 2 km. The road now slopes upward with Alpha =
0.07 rad, so the rider applies T = 15 N.m and m I = 6.5 A (using low gear). Mapped
distance for the slope is 4 km.

Original code by: P.J. Bones   UCECE
Code edited by: John Elliott and Samuel Vallance
Last modified:  09/10/2020
%}
clear
```

```matlab
clc
close all

%{
===========================================================================
                             | Variable Setup |
===========================================================================
%}
%… Simple variable assignment omitted
% Distances
flatr = 2000;    % Flat road distance in m
slopem = 4000;   % Slope mapped distance in m
sloper = slopem/cos(alphaf);
                 % Slope road distance in m

% Battery parameters
Qtotal = 18;     % Total battery capacity in Ah
Qused = 0;       % Total battery used in Ah

%{
===========================================================================
                 | Euler's Method and Speed Calculations |
===========================================================================
%}
% Continually compute u and s as the e-bike proceeds, advancing time
% by tdelta sec each step until required distance is travelled
index = 1;
while (s <= (flatr + sloper))
   % Compute the motor acceleration at the start of the step
   windf = beta * u^2;
   gravityf =  M * g * sin(alpha);
   domdt = (Tmc + (rsc / rp) * Tp - rw * (fd0 + windf + gravityf)) / Jef;

   % Estimate the state at the end of the time step (n = 'next')
   omn = om + tdelta * domdt;       % Estimate om by Euler's method
   un = omn * rw;                   % Estimate u at end of step
   sn = s + tdelta * (un + u) / 2;  % Estimate s at end of step (Trap.)

   domdtarray(index) = domdt;       % Store current values
   uarray(index) = u;
   sarray(index) = s;
   tarray(index + 1) = tarray(index) + tdelta;
                                    % Stores new time step taken
   Qused = Qused + (Tmc / km) * tdelta / 3600;
                                    % Stores total battery used
   index = index + 1;

   % Advance to next time step
   om = omn;
   u = un;
   s = sn;

   % Checks to ensure power is below maximum output.
   if (Tmc * omn) > pmax
       Tmc = (pmax / omn);        % Power is at max, so torque is limited
   else
       Tmc = Tmh;                 % Returns torque to fixed value
   end
```

```
   % Checks if next step is at the slope so that variables can be updated.
   if (s >= flatr)
       Tmc = Tmf;           % Changes motor torque to sloped value
       Tmh = Tmf;           % Holder motor torque, for when power > 500 W
       Tp = Tpf;            % Changes pedal torque to sloped value
       alpha = alphaf;      % Changes angle to sloped angle
       rsc = rs1;           % Changes gear from high to low gear
   end
end
% Store final values
domdtarray(index) = domdt;
uarray(index) = un;
sarray(index) = s;

% Finds battery usage percentage
Qpercent = Qused/Qtotal * 100;

%{
============================================================================
                      | Display Results and Graphs |
============================================================================
%}
% … Code plotting values for Task 2 omitted
```

Task 3:

```
%{
Case Study #2 EMTH171  2020
E-Bike

Task 3 Description:
The rider travelled on a flat road with wind resistance after 5 km the rider
travels another 5 km mapped distance up a slope with an angle of 0.05 rad.
The parameters for pedal torque and motor current must be chosen for each section
of the journey so to minimize energy expended by the rider.

Original code by: P.J. Bones   UCECE
Code edited by: John Elliott and Samuel Vallance
Last modified:  09/10/2020
%}
clear
clc
close all
%{
============================================================================
                            | Variable Setup |
============================================================================
%}
%… Simple variable assignment omitted
% Simulation chosen parameters
Tp = 6.48;      % Flat road pedal torque in Nm
Im = 0;         % Flat road motor current in A

Tpf = 11.6;     % Sloped road pedal torque in Nm
Imf = 6.3;      % Sloped road motor current in A
```

```matlab
%{
===========================================================================
                  | Euler's Method and Speed Calculations |
===========================================================================
%}
% Continually compute u and s as the e-bike proceeds, advancing by tdelta each
% step until required distance is travelled or battery is flat.
index = 1;
while (s <= (flatr + sloper)) && (Qused <= Qtotal)
   % Compute the motor acceleration at the start of the step
   windf = beta * u^2;
   gravityf =  M * g * sin(alphac);
   domdt = (Tmc + (rsc / rp) * Tpc - rw * (fd0 + windf + gravityf)) / Jef;

   % Estimate the state at the end of the time step (n = 'next')
   omn = om + tdelta * domdt;        % Estimate om by Euler's method
   un = omn * rw;                    % Estimate u at end of step
   sn = s + tdelta * (un + u) / 2;   % Estimate s at end of step (Trap.)
   p = (Tpc * ((omn * rsc) / rp));   % Estimate power used at end of step

   domdtarray(index) = domdt;        % Store current values
   uarray(index) = u;
   sarray(index) = s;
   earray(index + 1) = earray(index) + (p * tdelta);
                                     % Stores new total energy used
   tarray(index + 1) = tarray(index) + tdelta;
                                     % Stores new time step taken
   Qused = Qused + (Tmc / km) * tdelta / 3600;
                                     % Stores total battery used
   index = index + 1;

   % Advance to next time step
   om = omn;
   u = un;
   s = sn;

   % Checks to ensure power is below maximum output.
   if (Tmc * omn) > pmax
       Tmc = (pmax / omn);        % Power is at max, so torque is limited
   else
       Tmc = Tmh;                 % Returns torque to fixed value
   end

   % Checks if next step is at the slope so that variables can be updated.
   if (s >= flatr)
       Tmc = Tmf;
       Tmh = Tmf;
       Tpc = Tpf;
       alphac = alphaf;
       rsc = rs1;
   end
end
%{
===========================================================================
                       | Display Results and Graphs |
===========================================================================
%}
% … Code plotting values for Task 3 omitted
```