8

Electric Vehicle Modelling

8.1 Introduction

With all vehicles the prediction of performance and range is important. Computers allow us to do this reasonably easily. Above all, computer-based methods allow us quickly to investigate aspects of the vehicle, such as motor power, battery type and size, weight, and so on, and see how the changes might affect the performance and range. In this chapter we will show how the equations we have developed in the preceding chapters can be put together to perform quite accurate and useful simulations. Furthermore, we will show how this can be done without using any special knowledge of programming techniques, as standard mathematical and spreadsheet programs such as MATLAB® and Excel make an excellent basis for these simulations. We will also see that there are some features of electric vehicles that make the mathematical modelling of performance easier than for other vehicles.

The first parameter we will model is vehicle *performance*. By performance we mean acceleration and top speed – an area where electric vehicles have a reputation of being very poor. It is necessary that any electric vehicle has a performance that allows it, at the very least, to blend safely with ordinary city traffic. Many would argue that the performance should be at least as good as current IC engine vehicles if large-scale sales are to be achieved.

Another vitally important feature of electric vehicles that we must be able to predict is their *range*. This can also be mathematically modelled, and computer programs make this quite straightforward. The mathematics we will develop will allow us to see the effects of changing things like battery type and capacity, as well as all other aspects of vehicle design, on range. This is an essential tool for the vehicle designer.

We will go on to show how the data produced by the simulations can also have other uses in addition to predicting performance and range. For example, we will see how data about the motor torque and speed can be used to optimise the compromises involved in the design of the motor and other subsystems.

Electric Vehicle Technology Explained, Second Edition. James Larminie and John Lowry.

8.2 Tractive Effort

8.2.1 Introduction

The first step in vehicle performance modelling is to produce an equation for the required 'tractive effort'. This is the force propelling the vehicle forward, transmitted to the ground through the drive wheels.

Consider a vehicle of mass m, proceeding at a velocity v, up a slope of angle ψ, as in Figure 8.1. The force propelling the vehicle forward, the tractive effort, has to accomplish the following:

- overcome the rolling resistance;
- overcome the aerodynamic drag;
- provide the force needed to overcome the component of the vehicle's weight acting down the slope;
- accelerate the vehicle, if the velocity is not constant.

We will consider each of these in turn.

8.2.2 Rolling Resistance Force

The rolling resistance is primarily due to the hysteresis losses in the vehicle tyres. Friction in bearings and the gearing system also play their part. The rolling resistance is approximately constant, and hardly depends on vehicle speed. It is proportional to vehicle weight. The equation is

$$F_{rr} = \mu_{rr} m g \tag{8.1}$$

where μ_{rr} is the coefficient of rolling resistance. The main factors controlling μ_{rr} are the type of tyre and the tyre pressure. Any cyclist will know this very well – the freewheeling performance of a bicycle becomes much better if the tyres are pumped up to a high pressure, though the ride may be less comfortable.

The value of μ_{rr} can reasonably readily be found by pulling a vehicle at a steady very low speed, and measuring the force required.

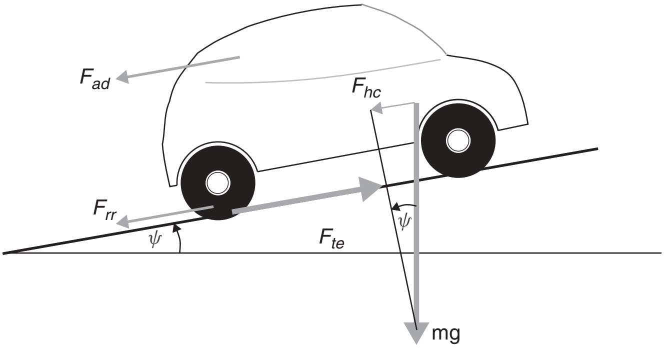

Figure 8.1 The forces acting on a vehicle moving up a slope

Typical values of μ_{rr} are 0.015 for a radial ply tyre, down to about 0.005 for tyres developed especially for electric vehicles.

8.2.3 Aerodynamic Drag

This part of the force is due to the friction of the vehicle body moving through the air. It is a function of the frontal area, shape, protrusions such as side mirrors, ducts and air passages, spoilers and many other factors. The formula for this component is

$$F_{ad} = \tfrac{1}{2}\rho A C_d v^2 \tag{8.2}$$

where ρ is the density of the air, A is the frontal area and v is the velocity. C_d is a constant called the 'drag coefficient'.

The drag coefficient C_d can be reduced by good vehicle design. A typical value for a saloon car is 0.3, but some electric vehicle designs have achieved values as low as 0.19. There is greater opportunity for reducing C_d in electric vehicle design because there is more flexibility in the location of the major components, and there is less need for cooling air ducting and under-vehicle pipework. However, some vehicles, such as motorcycles and buses, will inevitably have much larger values, and C_d figures of around 0.7 are more typical in such cases.

The density of air does of course vary with temperature, altitude and humidity. However a value of $1.25\,\mathrm{kg\,m^{-3}}$ is a reasonable value to use in most cases. Provided that SI units are used (m^2 for A, $\mathrm{m\,s^{-1}}$ for v) then the value of F_{ad} will be given in newtons.

8.2.4 Hill Climbing Force

The force needed to drive the vehicle up a slope is the most straightforward to find. It is simply the component of the vehicle weight that acts along the slope. By simple resolution of forces we see that

$$F_{hc} = mg\sin\psi \tag{8.3}$$

8.2.5 Acceleration Force

If the velocity of the vehicle is changing, then clearly a force will need to be applied in addition to the forces shown in Figure 8.1. This force will provide the *linear acceleration* of the vehicle, and is given by the well-known equation derived from Newton's third law,

$$F_{la} = ma \tag{8.4}$$

However, for a more accurate picture of the force needed to accelerate the vehicle we should also consider the force needed to make the rotating parts turn faster. In other words, we need to consider *rotational* acceleration as well as *linear* acceleration. The main issue here is the electric motor – not necessarily because of its particularly high moment of inertia, but because of the higher angular speeds.

Referring to Figure 8.2, clearly the axle torque equals $F_{te}r$, where r is the radius of the tyre and F_{te} is the tractive effort delivered by the powertrain. If G is the gear ratio

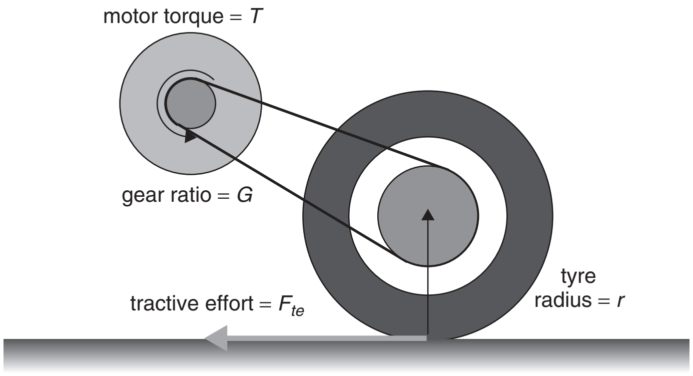

Figure 8.2 A simple arrangement for connecting a motor to a drive wheel

of the system connecting the motor to the axle and T is the motor torque, then we can say that

$$T = \frac{F_{te}r}{G} \quad \text{and} \quad F_{te} = \frac{G}{r}T \tag{8.5}$$

We will use this equation again when we develop final equations for vehicle performance. We should also note axle angular speed equals v/r radians per second, so motor angular speed is

$$\omega = G\frac{v}{r}\text{rad s}^{-1} \tag{8.6}$$

and, similarly, motor angular acceleration is

$$\dot{\omega} = G\frac{a}{r}\text{rad s}^{-2}$$

The torque required for this angular acceleration is

$$T = I \times G\frac{a}{r}$$

where I is the moment of inertia of the rotor of the motor. The force at the wheels needed to provide the angular acceleration ($F_{\omega a}$) is found by combining this equation with Equation (8.5), giving

$$F_{\omega a} = \frac{G}{r} \times I \times G\frac{a}{r} \quad \text{or} \quad F_{\omega a} = I\frac{G^2}{r^2}a \tag{8.7}$$

We must note that in these simple equations we have assumed that the gear system is 100% efficient – it causes no losses. Since the system will usually be very simple, the efficiency is often very high. However, it will never be 100%, and so we should refine the equation by incorporating the gear system efficiency η_g. The force required will be slightly larger, so Equation (8.7) can be refined to

$$F_{\omega a} = I\frac{G^2}{\eta_g r^2}a \tag{8.8}$$

Typical values for the constants here are 40 for G/r and 0.025 kg m^2 for the moment of inertia. These are for a 30 kW motor, driving a car which reaches 60 kph at a motor speed of 7000 rpm. Such a car would probably weigh about 800 kg. The IG^2/r^2 term in Equation (8.8) will have a value of about 40 kg in this case. In other words, the angular acceleration force given by Equation (8.8) will typically be much smaller than the linear acceleration force given by Equation (8.4) In this specific (but reasonably typical) case, it will be smaller by the ratio

$$\frac{40}{800} = 0.05 = 5\%$$

It will quite often turn out that the moment of inertia of the motor I will not be known. In such cases a reasonable approximation is simply to increase the mass by 5% in Equation (8.4), and to ignore the $F_{\omega a}$ term.

8.2.6 Total Tractive Effort

The total tractive effort is the sum of all these forces

$$F_{te} = F_{rr} + F_{ad} + F_{hc} + F_{la} + F_{\omega a} \tag{8.9}$$

where:

- F_{rr} is the rolling resistance force, given by Equation (8.1)
- F_{ad} is the aerodynamic drag, given by Equation (8.2)
- F_{hc} is the hill climbing force, given by Equation (8.3)
- F_{la} is the force required to give linear acceleration given by Equation (8.4)
- $F_{\omega a}$ the force required to give angular acceleration to the rotating motor, given by Equation (8.8)

We should note that F_{la} and $F_{\omega a}$ will be negative if the vehicle is slowing down, and that F_{hc} will be negative if it is going downhill.

8.3 Modelling Vehicle Acceleration

8.3.1 Acceleration Performance Parameters

The acceleration of a car or motorcycle is a key performance indicator, though there is no standard measure used. Typically the time to accelerate from standstill to 60 mph, or 30 or 50 kph, will be given. The nearest to such a standard for electric vehicles are the 0–30 and 0–50 kph times, though these times are not given for all vehicles.

Such acceleration figures are found from simulation or testing of real vehicles. For IC engine vehicles this is done at maximum power. Similarly, for electric vehicles performance simulations are carried out at maximum torque.

We have already seen in Chapter 7 that the maximum torque of an electric motor is a fairly simple function of angular speed. In most cases, at low speeds, the maximum torque is a constant, until the motor speed reaches a critical value ω_c after which the torque falls.

In the case of a 'brushed' shunt or permanent magnet (PM) DC motor the torque falls linearly with increasing speed. In the case of most other types of motor, the torque falls in such a way that the power remains constant.

The angular velocity of the motor depends on the gear ratio G and the radius of the drive wheel r as in Equation (8.6) derived above. So, we can say that

$$\text{For } \omega < \omega_c, \text{ or } v < \frac{r}{G}\omega_c, \text{ then } T = T_{\max}$$

Once this constant torque phase is passed, that is $\omega \geq \omega_c$, or $v \geq (r/G)\,\omega_c$, then *either* the power is constant, as in most brushless type motors, and we have

$$T = \frac{T_{\max}\omega_c}{\omega} = \frac{rT_{\max}\omega_c}{Gv} \tag{8.10}$$

or the torque falls according to the linear equation we met in Section 7.1.2:

$$T = T_0 - k\omega$$

which, when Equation (8.6) is substituted for angular speed, gives

$$T = T_0 - \frac{kG}{r}v \tag{8.11}$$

Now that we have the equations we need, we can combine them in order to find the acceleration of a vehicle. Many of these equations may look quite complex, but nearly all the terms are constants, which can be found or estimated from vehicle or component data.

For a vehicle on level ground, with air density $1.25\,\text{kg}\,\text{m}^{-3}$, Equation (8.9) becomes

$$F_{te} = \mu_{rr}mg + 0.625AC_dv^2 + ma + I\frac{G^2}{\eta_g r^2}a$$

Substituting Equation (8.5) for F_{te}, and noting that $a = dv/dt$, we have

$$\frac{G}{r}T = \mu_{rr}mg + 0.625AC_dv^2 + \left(m + I\frac{G^2}{\eta_g r^2}\right)\frac{dv}{dt} \tag{8.12}$$

We have already noted that T, the motor torque, is either a constant or a simple function of speed [Equations (8.10) and (8.11)]. So, Equation (8.13) can be reduced to a differential equation, of first order, for the velocity v. Thus the value of v can be found for any value of t.

For example, in the initial acceleration phase, when $T = T_{\max}$, Equation (8.12) becomes

$$\frac{G}{r}T_{\max} = \mu_{rr}mg + 0.625AC_dv^2 + \left(m + I\frac{G^2}{\eta_g r^2}\right)\frac{dv}{dt} \tag{8.13}$$

Provided all the constants are known, or can reasonably be estimated, this is a very straightforward first-order differential equation, whose solution can be found using many modern calculators, as well as a wide range of personal computer programs. This is also possible for the situation with the larger motors. Two examples will hopefully make this clear.

Figure 8.3 Electric scooter of the type simulated at various points in this chapter. The photograph was taken in a Berlin car park

8.3.2 Modelling the Acceleration of an Electric Scooter

For our first example we will take an electric scooter. No particular model is being considered, but the vehicle is similar to the electric scooters made by Peugeot and EVS, an example of which is shown in Figure 8.3.

- The electric scooter has a mass of 115 kg, with a typical passenger of mass 70 kg, so total mass $m = 185\,\text{kg}$.
- The moment of inertia of the motor is not known, so we will adopt the expedient suggested at the end of Section 8.2.5, and increase m by 5% in the linear acceleration term only. A value of 194 kg will thus be used from m in the final term of Equation (8.13).
- The drag coefficient C_d is estimated as 0.75, a reasonable value for a small scooter, with a fairly 'sit-up' riding style.
- The frontal area of vehicle and rider is $0.6\,\text{m}^2$.
- The tyres and wheel bearings give a coefficient of rolling resistance, $\mu_{rr} = 0.007$.
- The motor is connected to the rear wheel using a 2:1 ratio belt system, and the wheel diameter is 42 cm. Thus $G = 2$ and $r = 0.21\,\text{m}$.
- The motor is an 18 V 'Lynch'-type motor, of the type discussed in the previous chapter, Section 7.1.2. Equation (7.8) has been recalculated for 18 V, giving

$$T = 153 - 1.16\,\omega \tag{8.14}$$

- As in Section 7.1.2, the maximum current is controlled by the maximum safe current, in this case 250 A, so, as shown in Equation (7.9), the maximum torque $T_{\max}$ is 34 N m.

- The critical motor speed, ω_c, after which the torque falls according to Equation (8.14), occurs when

$$34 = 153 - 1.16\,\omega \qquad \therefore \omega = \frac{153-34}{1.16} = 103\,\text{rad}\,\text{s}^{-1}$$

- The gear system is very simple, and of low ratio, so we can assume a good efficiency. A value of η_g of 0.98 is estimated. An effect of this will be to reduce the torque, and so this factor will be applied to the torque.

When the torque is constant, as in Equation (8.13), we have

$$\frac{2}{0.21} \times 0.98 \times 34 = 0.007 \times 185 \times 9.8 + 0.625 \times 0.6 \times 0.75v^2 + 194\frac{dv}{dt}$$

$$317 = 12.7 + 0.281v^2 + 194\frac{dv}{dt}$$

$$\text{thus} \quad 194\frac{dv}{dt} = 304 - 0.281v^2$$

$$\text{and so} \quad \frac{dv}{dt} = 1.57 - 0.00145v^2 \tag{8.15}$$

This equation holds until the torque begins to fall when $\omega = \omega_c = 103\,\text{rad}\,\text{s}^{-1}$, which corresponds to $103 \times 0.21/2 = 10.8\,\text{m s}^{-1}$. After this point the torque is governed by Equation (8.14). If we substitute this, and the other constants, into Equation (8.12) we obtain

$$\frac{2}{0.21} \times 0.98 \times \left(153 - 1.16\frac{2}{0.21}v\right) = 0.007 \times 185 \times 9.8$$

$$+ 0.625 \times 0.6 \times 0.75v^2 + 194\frac{dv}{dt}$$

$$1428 - 103v = 12.7 + 0.281v^2 + 194\frac{dv}{dt}$$

$$\text{and so} \quad \frac{dv}{dt} = 7.30 - 0.53v - 0.00145v^2 \tag{8.16}$$

There are many practical and simple ways of solving these differential equations. Many modern calculators will solve such equations, remembering that there is a simple initial condition that $v = 0$ when $t = 0$. However, the most versatile next step is to derive a simple numerical solution, which can then easily be used in Excel or MATLAB®. The derivative of v is simply the difference between consecutive values of v divided by the time step. Applying this to Equation (8.15) gives us

$$\frac{v_{n+1} - v_n}{\delta t} = 1.57 - 0.00145v_n^2$$

For a program such as Excel or MATLAB® we need to rearrange this equation to obtain the value of the *next* velocity from the *current* velocity. This is done as follows:

$$v_{n+1} = v_v + \delta t \times \left(1.57 - 0.00145v_n^2\right) \tag{8.17}$$

This equation holds for velocities up to the critical velocity of 10.8 m s^{-1}, after which we have to use Equation (8.16), approximated in exactly the same way as we have just done for Equation (8.15), which gives

$$v_{n+1} = v_v + \delta t \times \left(7.30 - 0.53v - 0.00145v_n^2\right) \tag{8.18}$$

The MATLAB® script file below shows how to solve these equations using this program. Figure 8.4 is a plot of the solution using a time step δt of 0.1 seconds. Exactly the same result can be obtained with almost equal ease using Excel. It is left as an exercise for the reader to produce an Excel spreadsheet or MATLAB® script file where many of the machine parameters, such as the gear ratio G, are left as easily altered variables, so that the effect of changing it on the vehicle's performance can be noted.

```
% ScootA - electric scooter acceleration.
t=linspace(0,50,501); % 0 to 50 s, in 0.1 s steps
vel=zeros(1,501); % 501 readings of velocity
d=zeros(1,501);% Array for storing distance travelled
dT=0.1; % 0.1 second time step
for n=1:500
% Now follow equations (8.17) & (8.18)
if vel(n)<10.8 % Torque constant till this point
      vel(n+1)= vel(n) + dT*(1.57 - (0.00145*(vel(n)^2)));
elseif vel(n)>=10.8
vel(n+1)=vel(n)+dT*(7.30-(0.53*vel(n))-(0.00145*(vel(n)^2)));
end;
d(n+1)=d(n) + 0.1*vel(n); % Compute distance travelled.
end;
vel=vel.*3.6; % Multiply by 3.6 to convert m/s to kph
plot(t,vel); axis([0 30 0 50]);
xlabel('Time / seconds');
ylabel('Velocity / kph');
title('Full power (WOT) acceleration of electric scooter');
```

The result of this simulation is shown in Figure 8.4, and shows that the performance is somewhat as might be expected from a fairly low-power motor. The acceleration is unspectacular, and the top speed is about 30 mph or 48 kph, on level ground. However, this is reasonably compatible with safe city riding. The acceleration of such vehicles is sometimes given in terms of the standing start 100 m times, and the power of such MATLAB® script files is that they can very easily be changed to produce such information. If the plot line in the file above is changed as follows, then Figure 8.5 is obtained.

```
plot(t,d); axis([0 15 0 100]);
```

While we are not claiming that our model exactly represents any particular commercial designs, it is worth noting the following points from the specification of the Peugeot Scoot'Elec performance specification:

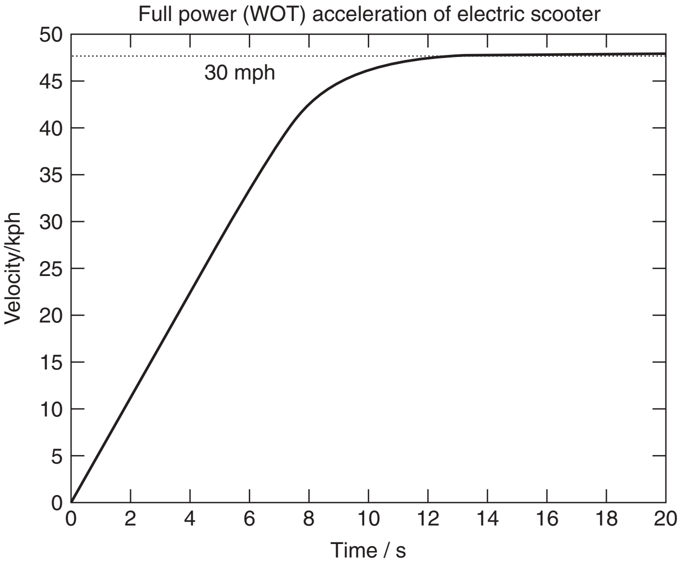

Figure 8.4 A graph showing the acceleration of a design of electric scooter, being the solution of Equations (8.15) and (8.16), as approximated by Equations (8.17) and (8.18), with a 0.1 second time step

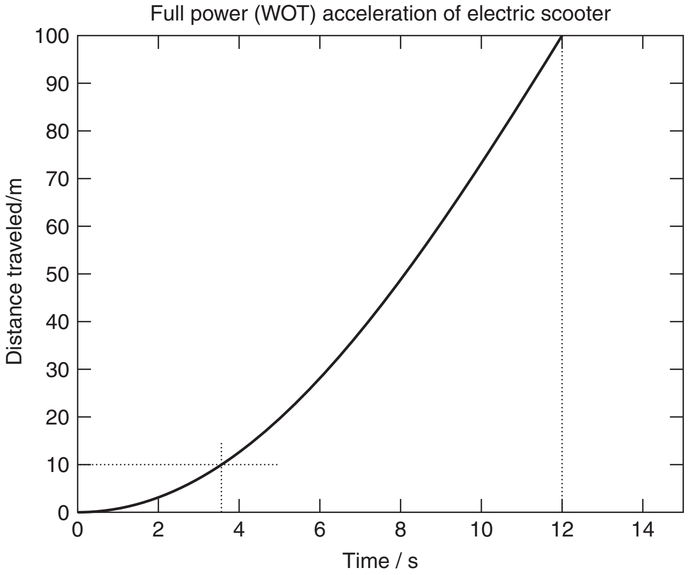

Figure 8.5 Distance/time graph for an electric scooter, showing the time to cover 10 and 100 m from a standing start

- Maximum speed: 45 kph (28 mph)
- 10 m from standing start time: 3.2 seconds
- 100 m from standing start time: 12 seconds.

It is very clear, from Figures 8.4 and 8.5, that the performance of our simulated vehicle is remarkably similar.

8.3.3 Modelling the Acceleration of a Small Car

For our second example we will use a vehicle that had an important impact on the recent development of electric cars. The GM EV1 was arguably the first modern electric car from one of the really large motor companies. It incorporated technologies that were quite novel when it was introduced. Several views of this vehicle are shown in Figure 14.5. Further details of this car are given in Section 14.3, but as far as simulating its performance, the main features are:

- an ultra-low drag coefficient C_d of 0.19;
- a very low coefficient of rolling resistance, μ_{rr}, of 0.0048;
- the use of variable frequency induction motors, operating at very high speed – nearly 12 000 rpm at maximum speed.

Further data is taken from company information[1] about the vehicle:

- Vehicle mass $= 1400\,\mathrm{kg}$.[2] Then add a driver and a passenger each weighing 70 kg, giving $m = 1540\,\mathrm{kg}$.
- The motor's moment of inertia is not known. However, compared with the mass of such a heavy vehicle this will be very low. The wheels are also very light. We will approximate this term by increasing the mass very modestly to 1560 kg in the final term of Equation (8.12).
- The gear ratio is 11:1, thus $G = 11$. The tyre radius is 0.30 m.
- For the motor, $T_{\max} = 140\,\mathrm{N\,m}$ and $\omega_c = 733\,\mathrm{rad\,s^{-1}}$ (Note that this means $T = T_{\max}$ till $v = 19.8\,\mathrm{m\,s^{-1}}$ (71.3 kph)).
- Above $19.8\,\mathrm{m\,s^{-1}}$ the motor operates at a constant 102 kW, as this is a WOT test. So

$$T = \frac{102000}{37 \times v} = \frac{2756}{v}$$

- The frontal area $A = 1.8\,\mathrm{m}^2$.
- The efficiency of the single-speed drive coupling between motor and axle is estimated as 95%, so $\eta_g = 0.95$. The values of the torque T will be reduced by a factor of 0.95. This slightly lower figure is because there is a final drive and a higher ratio gearbox than in the last example.

[1] The two sources are Shnayerson (1996) and the official GM EV1 website at www.gmev.com.

[2] It is interesting to note that 594 kg, or 42%, of this is the lead acid batteries!

These values can now be put into Equation (8.12), giving, for the first phase when the motor torque is constant,

$$0.95 \times 37 \times 140 = 72.4 + 0.214v^2 + 1560\frac{dv}{dt}$$

so

$$\frac{dv}{dt} = 3.11 - 0.000137v^2 \tag{8.19}$$

Once the speed has reached 19.8 m s^{-1} the velocity is given by the differential equation

$$0.95 \times 37 \times \frac{2756}{v} = 72.4 + 0.214v^2 + 1560\frac{dv}{dt}$$

so

$$\frac{dv}{dt} = \frac{62.1}{v} - 0.046 - 0.000137v^2 \tag{8.20}$$

The procedure for finding the acceleration is very similar to the first example; the only extra complication is that when the velocity reaches 35.8 m s^{-1} it stops rising, because at this point the motor controller limits any further acceleration.

Before any program such as Excel or MATLAB® can be used the key equations, Equations (8.19) and (8.20), must be put into 'finite difference' form. This is done exactly as we did for Equations (8.15) and (8.16). The two equations become

$$v_{n+1} = v_n + \delta t\left(3.11 - 0.000137v_n^2\right) \tag{8.21}$$

$$v_{n+1} = v_n + \delta t\left(\frac{62.1}{v_n} - 0.046 - 0.000137v_n^2\right) \tag{8.22}$$

The MATLAB® script file for these equations is very similar to that for the electric scooter given above, so it is not given here in the main text, but can be found in Appendix 1. The plot of velocity against time is shown in Figure 8.6. Looking at Figure 8.6, we can see that the time taken to reach 96 kph, which is 60 mph, is just under 9 seconds. Not only is this a very respectable performance, but it is also exactly the same as given in the official figures for the performance of the real vehicle.

We have thus seen that, although not overly complex, this method of modelling vehicle performance gives results that are validated by real data. We can therefore have confidence in this method. However, vehicles are required to do more than just accelerate well from a standing start, and in the next section we tackle the more complex issue of range modelling.

8.4 Modelling Electric Vehicle Range

8.4.1 Driving Cycles

It is well known that the range of electric vehicles is a major problem. In the main this is because it is so hard to store electrical energy efficiently. In any case, this problem is certainly a critical issue in the design of any electric vehicle. There are two types of calculation or test that can be performed with regard to the range of a vehicle.

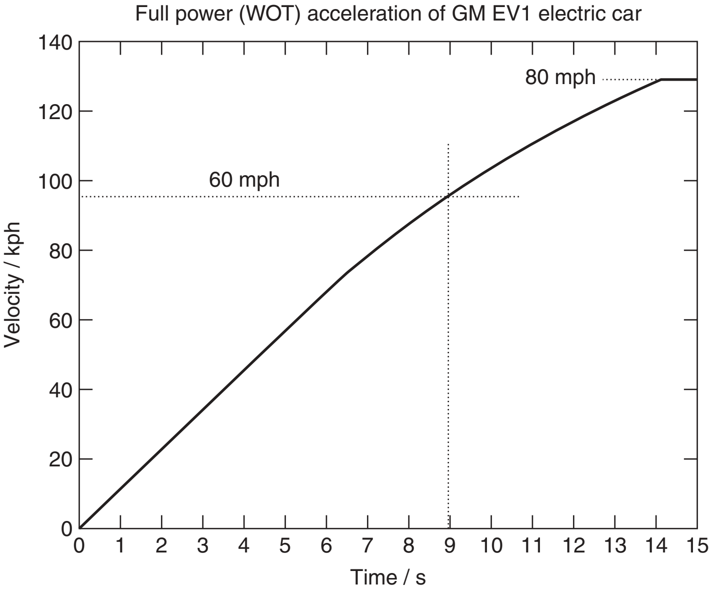

Figure 8.6 A graph of velocity against time for a GM EV1 at full power. This performance graph, obtained from a simple mathematical model, gives very good agreement with published real performance data

The first, and much the simplest, is the constant velocity simulation. Of course no vehicle is really driven at constant velocity, especially not on level ground, and in still air, which are almost universal further simplifications for these tests. However, at least the rules for the test are clear and unambiguous, even if the simulation is unrealistic. It can be argued that they do at least give useful comparative figures.

The second type of test – more useful and complex – is where the vehicle is driven, in reality or in simulation, through a profile of ever-changing speeds. These test cycles have been developed with some care, and there are (unfortunately) a large number of them. The cycles are intended to correspond to realistic driving patterns in different conditions. During these tests the vehicle speed is almost constantly changing, and thus the performance of all the other parts of the system is also highly variable, which makes the computations more complex. However, modern computer programs make even these more complex situations reasonably straightforward.

These driving cycles (or schedules) have primarily been developed in order to provide a realistic and practical test for the emissions of vehicles. One of the most well known of the early cycles was one based on actual traffic flows in Los Angeles, and is known as the LA-4 cycle. This was then developed into the Federal Urban Driving Schedule, or FUDS. This is a cycle lasting 1500 seconds, and for each second there is a different speed, as shown in Figure 8.7. There is also a simplified version of this cycle known as SFUDS, shown in Figure 8.8, which has the advantage that it only lasts 360 seconds, and so has only 360 data points. This has the same average speed, the same proportion of time stationary, the same maximum acceleration and braking, and gives very similar results when used for simulating vehicle range.

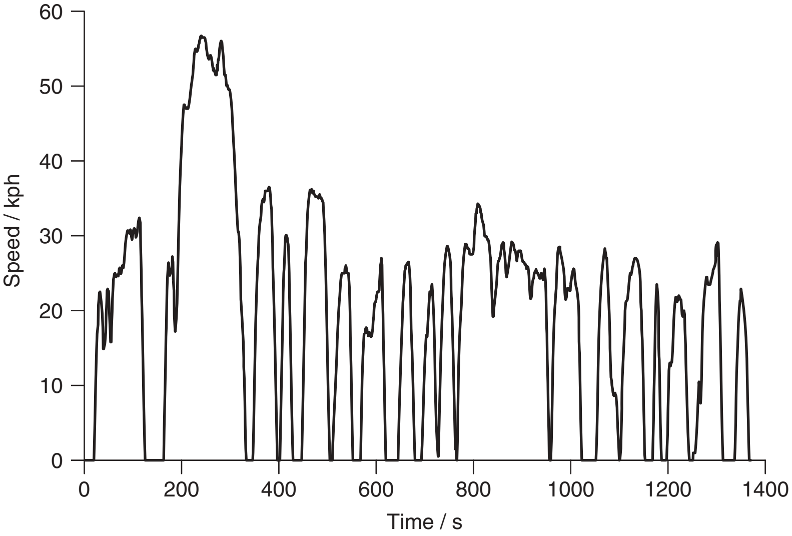

Figure 8.7 The Federal Urban Driving Schedule, as used for emission testing by the United States Environmental Protection Agency

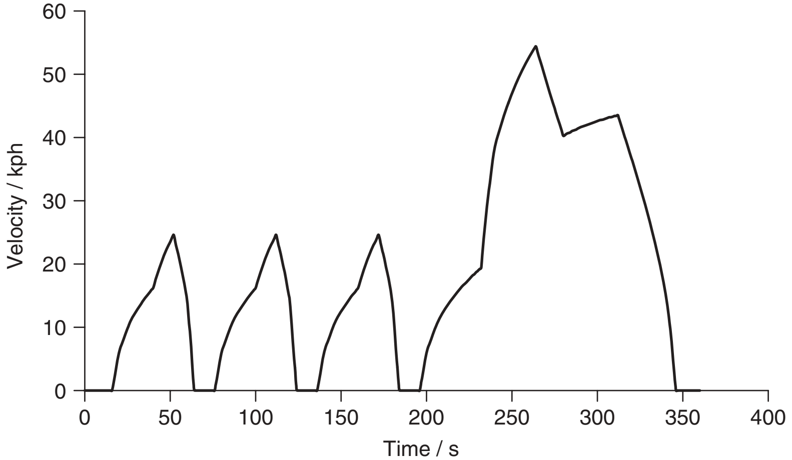

Figure 8.8 Graph of speed against time for the Simplified Federal Urban Driving Schedule

These cycles simulate urban driving, but other cycles are used to simulate out-of-town or highway driving. Two notable examples of these are the Federal Highway Driving Schedule, or FHDS, shown in Figure 8.9. Although widely used, this cycle has a rather unrealistic maximum speed for highway driving, and the newer US06 standard, is now becoming more widely used instead.

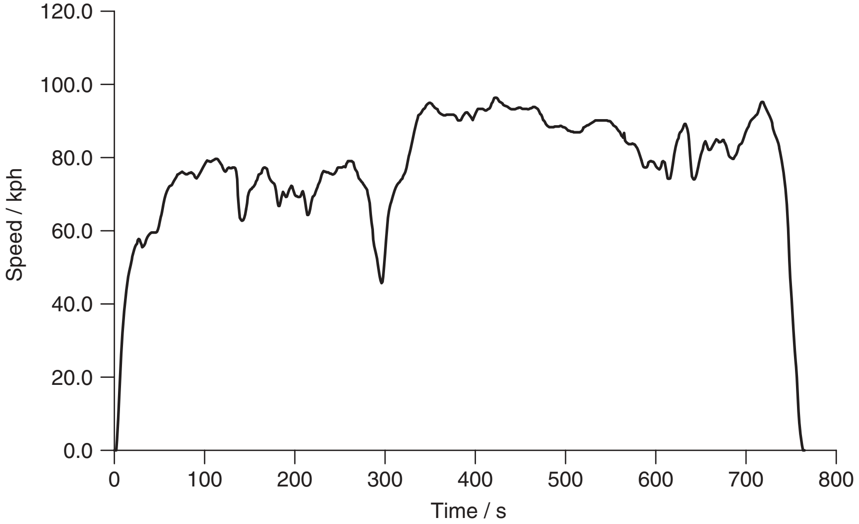

Figure 8.9 Graph of speed against time for the 765 second Federal Highway Driving Schedule

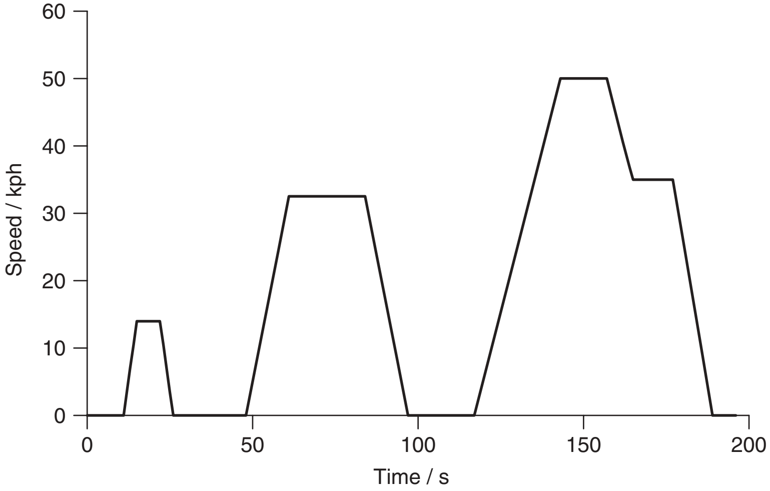

Figure 8.10 European urban driving schedule ECE-15

In the European scene the cycles tend to be rather simpler, with periods of constant acceleration and constant velocity. Of particular note is the ECE-15 drive cycle, shown in Figure 8.10, which is useful for testing the performance of small vehicles such as battery electric cars. In EC emission tests this has to be combined with the Extra-Urban Driving Cycle (EUDC), which has a maximum speed of 120 kph.

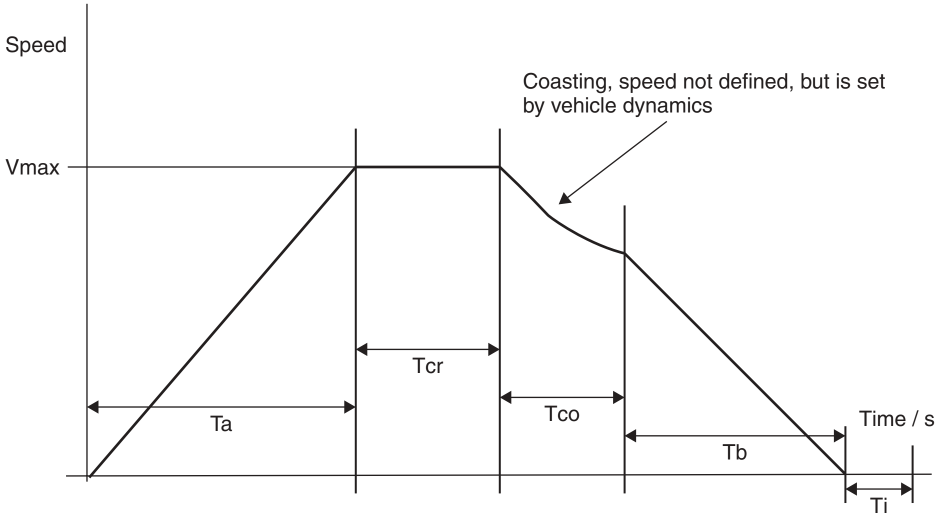

Figure 8.11 Diagram for SAE J227a cycle. The figures for the various times are given in Table 8.1

Currently the most widely used standard in Asia is the Japanese 10-15 Mode Cycle. Like the European cycles, this involves periods of constant velocity and acceleration. It is not unlike a combination of the European ECE-15 urban driving cycle and the EUDC. At the time of writing, this cycle must be used in stating ranges for vehicles in Japan, as well as for emission tests.

All these standards have maximum speeds in the region of 100 kph. For several important types of electric vehicle, including the electric delivery vehicle and the electric motor scooter, this is an unrealistic speed, which can often not be achieved. To simulate these vehicles other standard cycles are needed. A fairly old standard, which was developed specifically for electric vehicles in the 1970s, is the SAE J227a driving schedule. This has four versions, with different speeds. Each cycle is quite short in time, and consists of an acceleration phase, a constant velocity phase, a 'coast' phase and a braking phase, followed by a stationary time. The coasting phase, where the speed is not specified, but the tractive effort is set to zero, is somewhat of a nuisance to model.[3] The general velocity profile is shown in Figure 8.11, and the details of each of the four variants of this cycle are given in Table 8.1. The most commonly used cycle is SAE J227a-C, which is particularly suitable for electric scooters and smaller city-only electric vehicles. The A and B variants are sometimes used for special purpose delivery vehicles.

Another schedule worthy of note for low-speed vehicles is the European ECE-47 cycle, which is used for the emission testing of mopeds and motorcycles with engine capacity less than 50 cm^3. It is also widely used for the range simulation of electric scooters. Like the SAE J227 cycle it can be a little complicated to run the simulation, as the speed is not specified at all times. Instead the vehicle is run from standstill at WOT for 50 seconds. The vehicle is then slowed to 20 kph over the next 15 seconds, after which this

[3] It is not uncommon to get around this difficulty by simply putting in likely figures for a somewhat gentle period of deceleration.

Table 8.1 Nominal parameters for the four variations of the SAE J227a test schedule. These figures should be read in conjunction with Figure 8.10

Parameter	Unit	Cycle A	Cycle B	Cycle C	Cycle D
Maximum speed	kph	16	32	48	72
Acceleration time, Ta	s	4	19	18	28
Cruise time, Tcr	s	0	19	20	50
Coast time, Tco	s	2	4	8	10
Brake time, Tb	s	3	5	9	9
Idle time, Ti	s	30	25	25	25
Total time	s	39	72	80	122

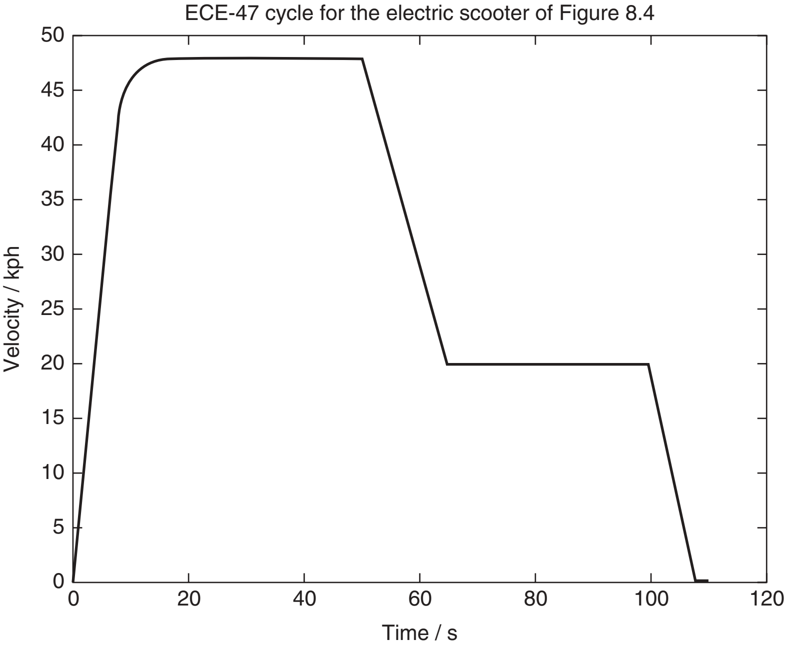

Figure 8.12 ECE-47 cycles for the same electric scooter as in Figure 8.4

speed is maintained for 35 seconds. Finally the vehicle is brought to a halt, at constant deceleration, over the next 8 seconds. This cycle has been created for the same electric scooter, an approximation to the Peugeot Scoot’Elec, and is shown in Figure 8.12. The MATLAB® script file for creating this is very straightforward, and is given in Appendix 2. This cycle has the benefit that it probably does quite well model the way such vehicles might be used – considering the age of the typical rider, that is full speed for a good deal of the time!

There are many other test cycles which can be found in the literature, and some companies have their own ‘in-house’ driving schedules. Academics sometimes propose new ones that, they suggest, better imitate real driving practice. There are also local driving cycles, which reflect the particular driving patterns of a city. A noteworthy example is the New York City Cycle, which has particularly long periods of no movement, and low average speeds, reflecting the state of the roads there. This cycle is sometimes used when simulating hybrid electric/IC engine (ICE) vehicles, as it shows this type of vehicle in a particularly good light.

The actual figures for the speed at each second, which are needed to run a simulation, can sometimes be deduced from the figures given above. However, in the case of the US cycles, which consist of a specific speed at each time, it is more convenient to load data files downloaded from web sites. These can readily be found using the normal Internet search engines, though several have been supplied as MATLAB® script files at a dedicated website.[4]

In the sections that follow it will be explained how a driving schedule can be simulated for different types of electric vehicle.

8.4.2 Range Modelling of Battery Electric Vehicles

8.4.2.1 Principles of Battery Electric Vehicle Modelling

The energy flows in a classical battery electric vehicle are shown in Figure 8.13. To predict the range, the *energy* required to move the vehicle for each second of the driving cycle is calculated, and the effects of this energy drain are calculated. The process is repeated until the battery is flat. It is important to remember that if we use time intervals of 1 second, then the *power* and the *energy consumed* are equal.

The starting point in these calculations is to find the tractive effort, which is found from Equation (8.9) The power is equal to the tractive effort multiplied by the velocity. Using the various efficiencies in the energy flow diagram, the energy required to move the vehicle for 1 second is calculated.

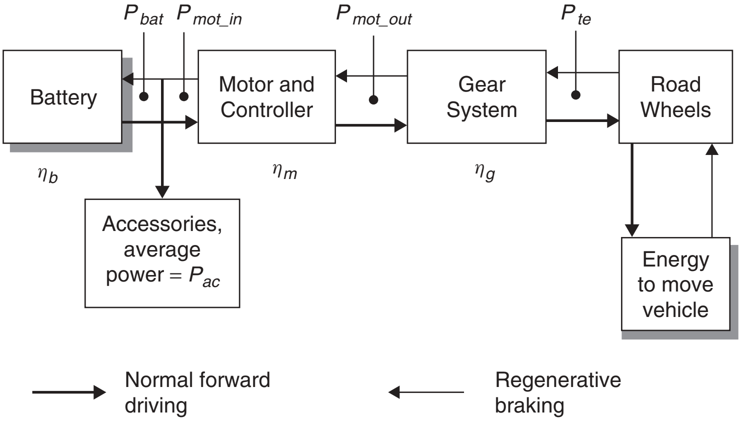

Figure 8.13 Energy flows in the 'classic' battery-powered electric vehicle, which has regenerative braking

[4] www.wiley.com/go/electricvehicle2e.

The energy required to move the vehicle for 1 second is the same as the power, so

$$\text{Energy required each second} = P_{te} = F_{te} \times v \tag{8.23}$$

To find the energy taken from the battery to provide this energy at the road we clearly need to be able to find the various efficiencies at all operating points. Equations that do this have been developed in the previous chapters, but we will review here the most important system modelling equations.

8.4.2.2 Modelling Equations

The efficiency of the gear system η_g is normally assumed to be constant, as in electric vehicles there is usually only one gear. The efficiency is normally high, as the gear system will be very simple.

The efficiencies of the motor and its controller are usually considered together, as it is more convenient to measure the efficiency of the whole system. We saw in Chapter 7 that motor efficiency varies considerably with power, torque and also motor size. The efficiency is quite well modelled by the equation

$$\eta_m = \frac{T\omega}{T\omega + k_c T^2 + k_i \omega + k_w \omega^3 + C} \tag{8.24}$$

where k_c is the copper losses coefficient, k_i is the iron losses coefficient, k_w is the windage loss coefficient and C represents the constant losses that apply at any speed. Table 8.2 shows typical values for these constants for two motors that are likely candidates for use in electric vehicles.

The inefficiencies of the motor, the controller and the gear system mean that the motor's power is not the same as the traction power, and the electrical power required by the motor is greater than the mechanical output power according to the simple equations

$$P_{mot\_in} = \frac{P_{mot\_out}}{\eta_m} \qquad P_{mot\_out} = \frac{P_{te}}{\eta_g} \tag{8.25}$$

Equations 8.25 are correct in the case where the vehicle is being driven. However, if the motor is being used to slow the vehicle, then the efficiency (or rather the inefficiency) works in the opposite sense. In other words, the electrical power from the motor is reduced, and we must use these equations:

$$P_{mot\_in} = P_{mot\_out} \times \eta_m \qquad P_{mot\_out} = P_{te} \times \eta_g \tag{8.26}$$

So, Equations 8.25 or 8.26 are used to give us the electrical and mechanical power to (or from) the motor. However, we also need to consider the other electrical systems of the vehicle, the lights, indicators, accessories such as the radio, and so on. An average power will need to be found or estimated for these, and added to the motor power, to give the total power required from the battery. Note that when braking, the motor power will be negative, and so this will reduce the magnitude of the power:

$$P_{bat} = P_{mot\_in} + P_{ac} \tag{8.27}$$

Table 8.2 Typical values for the parameters of Equation (8.23)

Parameter	‘Lynch’-type PM motor, with brushes, 2–5 kW	100 kW, high-speed induction motor
k_c	1.5	0.3
k_i	0.1	0.01
k_w	10^{-5}	5.0×10^{-6}
C	20	600

The meaning of these various powers, in and out of the motor, traction power and so on, is shown in Figure 8.13.

The simulation of battery behaviour was explained in Chapter 3, Section 3.12. But to summarise, the procedure now is:

1. Calculate the open-circuit battery voltage, which depends on the state of charge of the battery.
2. Calculate the battery current[5] using Equation (3.20), unless P_{bat} is negative, in which case Equation (3.22) should be used.
3. Update the record of charge removed from the battery, correcting high currents using the Peukert coefficient, with Equation (3.18). However, if the battery power is negative, and it is being charged, Equation (3.23) should be used instead.
4. The level of discharge of the battery should then be updated, using Equation (3.19)

Provided that the battery is not now too discharged, the whole process should then be repeated 1 second later, at the next velocity in the cycle.

8.4.2.3 Using MATLAB® or Excel to Simulate an Electric Vehicle

In the previous section we saw how the various equations we have derived can be used to calculate what goes on inside an electric vehicle. To see how far a vehicle can go before the battery is flat, we do this in a step-by-step process through the driving cycle. The way this is done is represented by the flowchart shown in Figure 8.14.

The first stage is to load the velocity data for the driving cycle to be used. This is usually done as by a separate MATLAB® script file. The way of doing this is explained in Appendices 1 and 2.

The next stage is to set up the vehicle parameters such as the mass, the battery size and type, and so on. The electrical power taken by the accessories P_{ac} should be set at this point.

Having done that, data arrays should be created for storing the data that needs to be remembered at the end of each cycle. These could be called ‘end of cycle arrays’. The most important data that needs to be kept is a record of the charge removed from the battery, the depth of discharge of the battery and the distance travelled.

[5] If the vehicle uses a conventional DC motor, it might be more convenient to calculate the current using the more or less linear relationship between torque and current. If the connection is known, this can occasionally be a useful simplification.

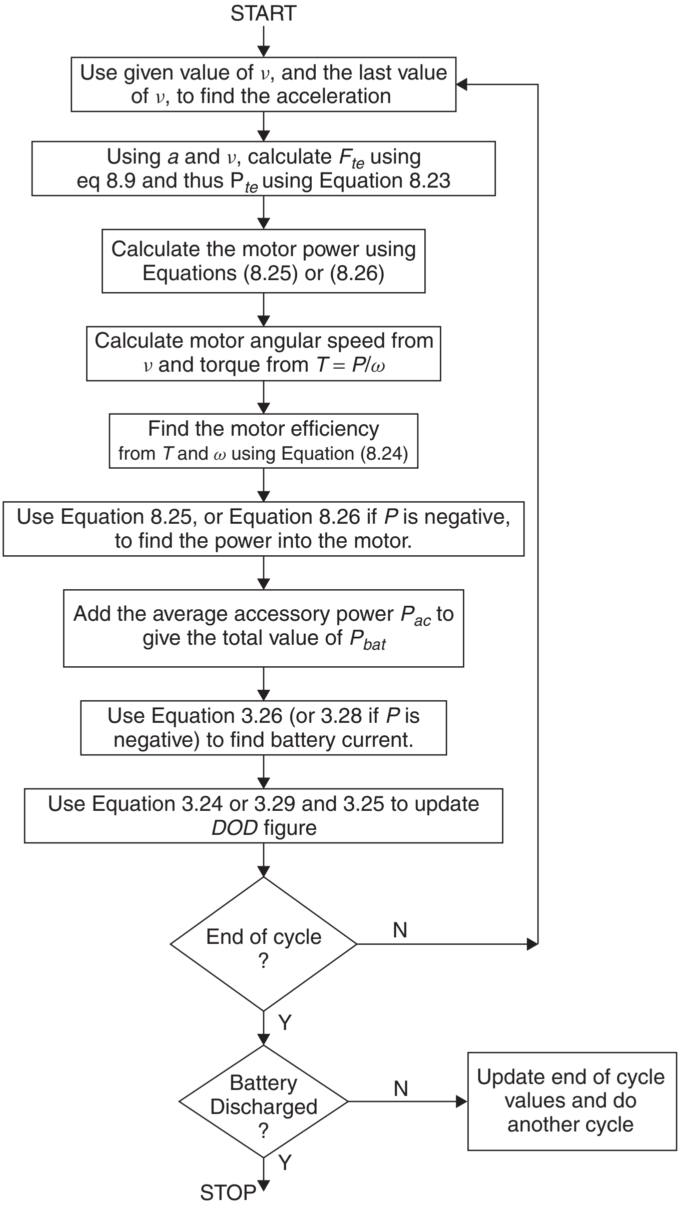
START
Use given value of v, and the last value of v, to find the acceleration
Using a and v, calculate F_{te} using eq 8.9 and thus P_{te} using Equation 8.23
Calculate the motor power using Equations (8.25) or (8.26)
Calculate motor angular speed from v and torque from $T = P/\omega$
Find the motor efficiency from T and ω using Equation (8.24)
Use Equation 8.25, or Equation 8.26 if P is negative, to find the power into the motor.
Add the average accessory power P_{ac} to give the total value of P_{bat}
Use Equation 3.26 (or 3.28 if P is negative) to find battery current.
Use Equation 3.24 or 3.29 and 3.25 to update DOD figure
End of cycle ?
N
Y
Battery Discharged ?
N
Update end of cycle values and do another cycle
Y
STOP

Figure 8.14 Flowchart for the simulation of a battery-powered electric vehicle

The next stage is to set up arrays for the data to be stored just for one cycle; this data can be lost at the end of each cycle. This is also the charge removed, depth of discharge and distance travelled, but we might also save other data, such as information about torque, or motor power, or battery current, as it is sometimes useful to be able to plot this data for just one cycle.

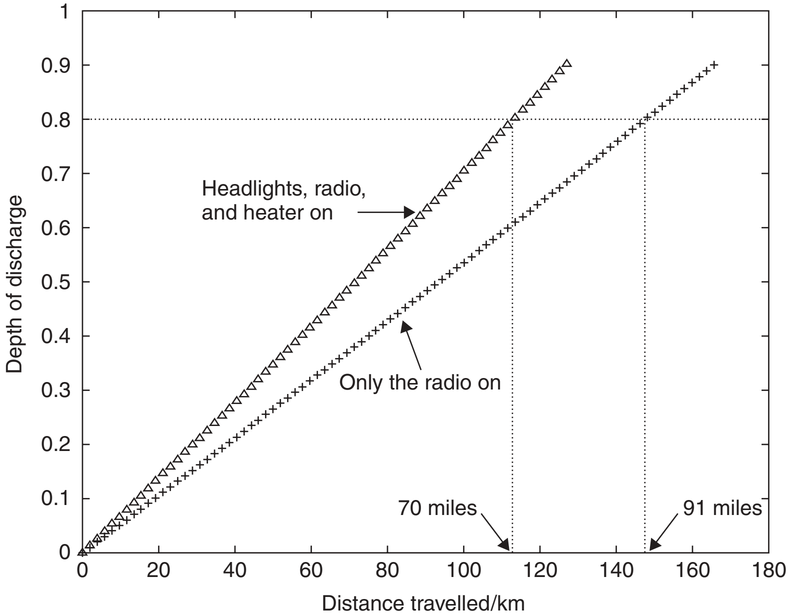

Figure 8.15 A graph of depth of discharge against distance travelled for a simulated GM EV1 electric car on the SFUDS driving cycle. In one case the conditions are benign: no lights, heating or air-conditioning are in use. In the other case the battery is degraded slightly by cold weather, and all the vehicle's headlights are on

Having set the system up, the vehicle is put through one driving cycle, using the velocities given to calculate the acceleration, and then the tractive effort, and thus the motor power, torque and speed. This is used to find the motor efficiency, which is used to find the electrical power going into the motor. Combined with the accessory power, this is used to find the battery current. This is then used to recalculate the battery state of charge. This calculation is repeated in 1 second steps[6] until the end of the cycle.

The end of cycle data arrays are then updated, and if the battery still has enough charge, the process is repeated for another cycle. This is process is shown in the flowchart of Figure 8.14.

MATLAB® lends itself very well to this type of calculation. Appendices 3 and 4 contain example MATLAB® script files that find the range for a model of the famous GM EV1 vehicle. It should be easy enough to relate these to the text and all the equations given above. The main complications relate to zero values for variables such as speed and torque – which need careful treatment to avoid dividing by zero. The vehicle is running an urban driving cycle.

The file prints a graph of the depth of battery discharge against distance travelled, and this is shown in Figure 8.15, for two different situations. It can be seen that the vehicle range is about 130 km.

[6] Steps of 1 second are the most convenient, as most driving cycles are defined in terms of 1 second intervals. Also, many of the formulae become much simpler. However, it is quite easy to adapt any of the programs given here for different time steps, and shorter steps are sometimes used.

One of the very powerful features of such simulations is that they can be used to see very quickly and easily the effect of changing certain vehicle parameters on the range. For example, it is the work of a moment to change the program so that the conditions are different. For example, we can 'put the headlights on' by increasing the value of the average accessory power P_{ac}. We can also simulate colder weather by increasing the internal resistance by 25% or so, raising the Peukert coefficient and reducing the battery voltage very slightly. The simulation can then be rerun. This has been done with Figure 8.15. This shows how the depth of discharge rises under normal clement weather, daytime conditions and also under colder conditions when in the dark. We can instantly see that the range, usually given when 80% discharge is reached, drops from a little over 90 miles to about 70 miles. The official stated range, in the GM literature, is '50–90 miles, depending on conditions'. Our simulation confirms this. We could further adapt the program to include hills, or more demanding driving, which would bring it below the 70 mile figure.

The ECE-47 driving cycle was explained in the previous section. This can equally well be used for such range testing. In Appendix 5 we have included another MATLAB® script file for the same electric scooter that was used for Figures 8.4 and 8.12. This vehicle has been set up with an NiCad battery, unlike the GM EV1, which uses lead acid. If the MATLAB® script file in the appendix is studied, it will be seen that the vehicle has been fitted with three 100 Ah batteries, with the same properties as the NiCad batteries simulated in Section 3.11.3. Some range data, taken to 80% discharged, is given in Table 8.3.

The range of the scooter appears to be about 50 km, which is longer than the 40 km in 'urban nominal mode' claimed by the Peugeot Scoot'Elec, which uses the same batteries. This is probably due to the fact that, as we will see in the next section, this ECE-47 driving cycle seems very well suited to the 'Lynch'-type motor we are using in our model. It may also be due to conservative claims in the vehicle specification.

Table 8.3 is also another demonstration of the power of simulations like these to find quickly the effect of changing vehicle parameters. In this case we have changed the proportion of the 'braking power' that is handled by the motor. In other words, we have changed the degree of regenerative braking performed. It is sometimes thought that this makes a huge difference to battery vehicle range. In the case of a scooter, it clearly does not. With no regenerative braking at all, the range is 48.82 km. Fifty per cent is probably

Table 8.3 The simulated range of an electric scooter running the ECE-47 driving cycle with different degrees of regenerative braking

Percentage regenerative braking	Range on ECE-47 cycle (km)
75% (Note: not possible in practice)	51.30
50%	50.47
25%	49.59
None	48.82

the highest practical possible motor braking, and this extends the range less than 2 km, or 4%. This does make some difference, but we should note that it is not a very great improvement. It is left as an exercise for the reader to do the same for the GM EV1 – here the difference will be much greater, because it is a heavy vehicle and well streamlined.

8.4.3 Constant Velocity Range Modelling

Compared with the modelling of the driving cycles we have just achieved, constant velocity simulation is much easier. However, the basic round of calculations is the same as those outlined in the previous section. The system is simpler since the values of speed and torque are never negative or zero.

It should be possible to write a new and much shorter MATLAB® script file for such simulation. However, a quicker and easier solution, which makes use of the programs already written, is to create a 'driving cycle' in which the velocity is constant. This can be done in one line in MATLAB® thus:

```
linspace(12.5, 12.5, 100);
```

This creates an array of 100 values, all equal to 12.5, which corresponds to 45 kph. A line like this, at any desired velocity, can replace the lines `ECE_47`, or `SFUDS`, at the beginning of the simulations given in the appendix. This may not be the most elegant method, but it is probably the quickest. Constant velocity simulations are clearly very unrealistic, and so are of limited use.

8.4.4 Other uses of Simulations

The data produced during these simulations has many more uses than just predicting the range of a vehicle. At each 1 second step of the cycle many variables were calculated, including:

- vehicle acceleration
- tractive effort
- motor power
- motor torque
- motor angular speed
- motor power
- motor efficiency
- current out of (or into) the battery.

All of these variables are of interest, and it is instructive to plot them over one cycle. This can be done with great simplicity in MATLAB®, and gives very useful results. The basic principle is to create two arrays, with names such as XDATA and YDATA, and allocate then values during a cycle. For example, if the loop counter is C, as it is in the examples of the appendix, then C will have the same value as the time in seconds. If we wanted to plot the value of the motor power during one cycle, then we would include the lines

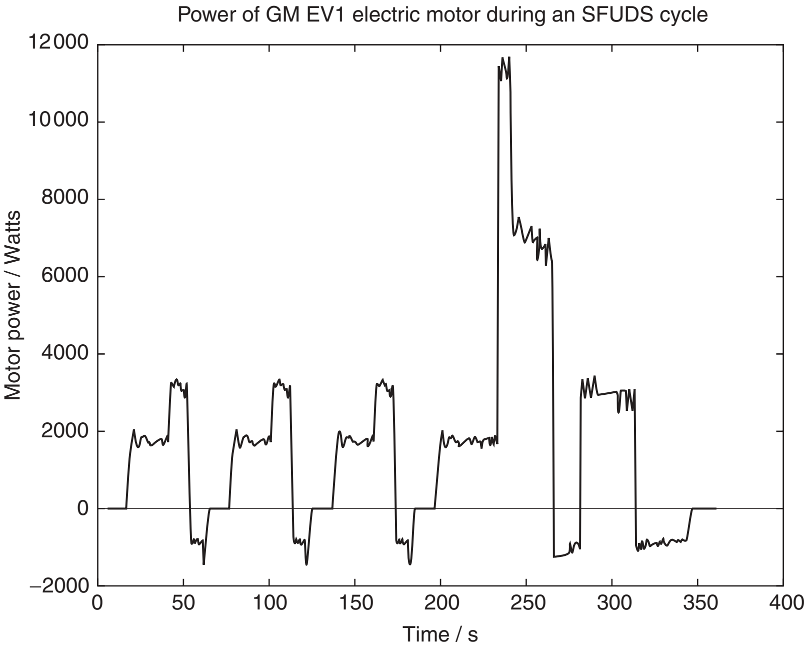

Figure 8.16 A graph of the power of the electric motor in a simulated GM EV1 electric car during one run of the SFUDS driving cycle

```
XDATA(C) = C;
YDATA(C) = Pmot_out;
```

into the code for a cycle. This can be seen very near the end of the script file given in Section A.3. Near the end of the program, the main program that uses `one_cycle`, we would include the line

```
plot(XDATA, YDATA);
```

For this type of plot the program could be simplified so that only *one* driving cycle is performed. An example of this type of plot output is shown in Figure 8.16. This shows the motor output power. It can be seen that the motor power is very modest, with a maximum of only about 12 kW. The motor has a maximum power of about 100 kW, so the SFUDS driving cycle is not testing the vehicle at all hard.[7]

Another example of a particularly useful plot is that of motor torque against motor angular speed. To produce this graph only two lines of the program need changing. The X and Y data lines become:

[7] This can be confirmed by looking for the maximum acceleration during the SFUDS cycle, which is only about $1\,\mathrm{m\,s^{-2}}$, whereas the GM EV1 is capable of over three times this value.

```
XDATA(C) = omega;
YDATA(C) = Torque;
```

For this type of plot the points should be left as disconnected points – they should not be joined by a line. MATLAB® easily allows this, and a suitable plot command is given near the end of the script file of Section A.5. In Figure 8.17 this has been done for the electric scooter simulation. This maps the operating points of the motor. This graph should be carefully compared with Figure 7.7 in the previous chapter. It can be seen that, at least with this driving cycle, the motor is frequently operating in the region of about 120 rad s^{-1} speed, and low (~10 N m) torque. From Figure 7.7 we can see that this is *precisely the area where the motor is most efficient*. The motor is thus extremely well matched to this particular driving cycle. This probably explains why the range simulation results were rather better than given in the specification for the Peugeot Scoot'Elec, to which our model is quite similar.

8.4.5 Range Modelling of Fuel Cell Vehicles

The principal energy flows in a fuel-cell-powered vehicle are shown in Figure 8.18. The energy required to drive the various fuel cell ancillaries that were discussed in Chapter 5 have not been explicitly shown, but these can be accounted for by adjustments to the value of the fuel cell efficiency.

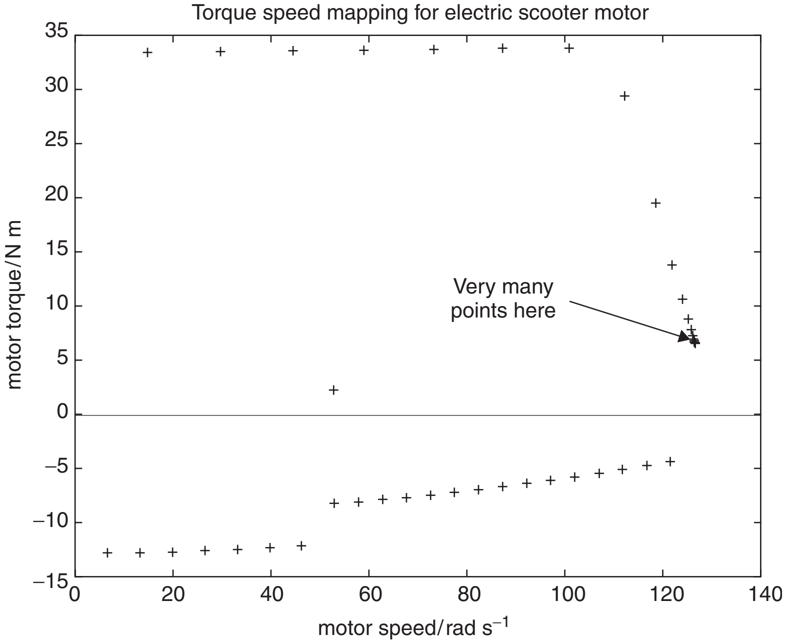

Figure 8.17 A plot of the torque/speed operating points for the electric motor in an electric scooter during the ECE-47 test cycle. In the indicated region, many points are superimposed, as the vehicle is at a constant velocity

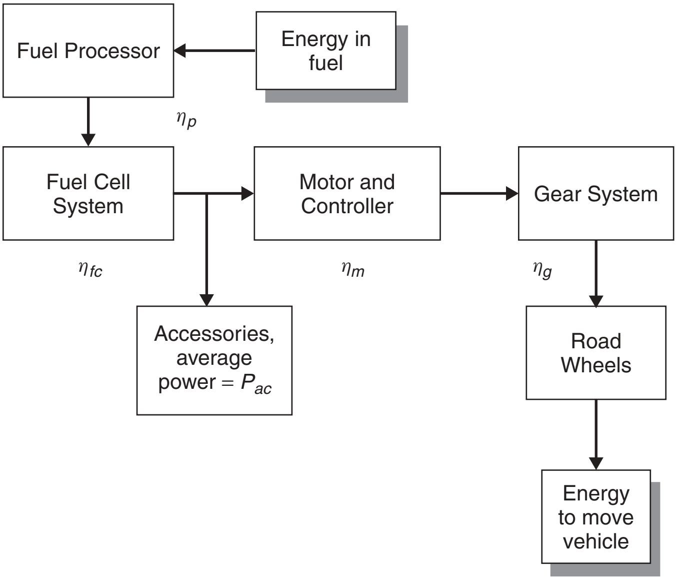

Figure 8.18 Energy flows in a fuel-cell-powered electric vehicle

The modelling of such a system is extremely complex, largely because of the fuel processor system. This has very many sub-processes with highly variable time constants, some quite long. The simulation of such fuel processing systems is extremely important – but too complex for an introductory text such as this. In addition, most of the important data is highly confidential to the companies developing these systems.

However, the simulation of a system running directly off onboard stored hydrogen is not nearly so complex. Indeed in many ways it can be less difficult than for battery vehicles, at least to a first approximation. The efficiency of a fuel cell is related, as we saw in Chapter 5, to the average voltage of each cell in the fuel cell stack, V_c. If the efficiency is referred to the lower heating value (LHV) of hydrogen, then

$$\eta_{fc} = \frac{V_c}{1.25} \tag{8.28}$$

Now, we know from Chapter 5 that at lower currents the fuel cell voltage rises, and thus the efficiency. However, we also saw in Chapter 5 that a fuel cell system will also have many pumps, compressors, controllers and other ‘balance of plant’ that use electrical power. This use of electrical power is higher, as a proportion of output power, at lower currents. The result is that, in practice, the efficiency of a fuel cell is, to quite a good approximation, more or less constant at all powers. (Note that this contrasts with an IC engine, whose efficiency falls very markedly at lower powers.)

At the present time, a good target value for the efficiency of a fuel cell operating off pure hydrogen is 38% referred to the LHV. So, from Equation (8.28), we have

$$V_c = 0.38 \times 1.25 = 0.475\text{V} \tag{8.29}$$

Note that the fuel cell will probably, in fact, be running at about 0.65 V, but the difference between this and 0.475 represents the energy used by the balance of plant.

This value of average cell voltage can then be used in the formula[8] for the rate of use of hydrogen in a fuel cell:

$$H_2 \text{ rate of usage } \dot{m} = 1.05 \times 10^{-8} \times \frac{P}{V_c} = 2.21 \times 10^{-8} \times P \tag{8.30}$$

Notice that this formula does not require us to know any details about the fuel cell, such as the number of cells, electrode area, or any details at all. It allows us very straightforwardly to calculate the mass of hydrogen used each second from the required electrical power. Indeed, this simulation is a great deal easier than with batteries because:

- there is no regenerative braking to incorporate;
- there are no currents to calculate;
- there is no Peukert correction of the current to be done.

By way of example, we could take our GM EV1 vehicle, and remove the 594 kg of batteries. In their place we could put the fuel cell system shown in Figure 5.23 and the hydrogen storage system outlined in Table 6.3. The key points are:

- Mass of hydrogen stored: 8.5 kg.
- Mass of storage system: 51.5 kg.
- Mass of 45 kW fuel cell system: 250 kg (estimate, not particularly optimistic).
- Total mass of vehicle is now $(1350 - 594) + 8.5 + 51.5 + 250 = 1066$ kg.

Appendix 6 contains the MATLAB® script file for running the SFUDS driving cycle for this hypothetical vehicle. It can be seen that the simulation is simpler. Some example results, *which the reader is strongly encourage to confirm*, are given in Table 8.4. In both cases 80% discharge is taken as the end point, that is 1.7 kg of hydrogen remaining in the case of the fuel cell.

An alternative approach that might well be found helpful, and that certainly results in a much simpler MATLAB® program, is to compute the energy consumed in running one cycle of the driving schedule being used. The distance travelled in one cycle should also be found. The number of cycles that can be performed can then be computed from the available energy and the overall efficiency. This approach obviously gives the same result.

Table 8.4 Simple fuel cell simulation results, showing great improvement in range over a battery vehicle

Simulation, SFUDS driving cycle in both cases	Range (km)
GM EV1 with standard lead acid batteries, good conditions	148
GM EV1 with a fuel cell and cryogenic H_2 store	485

[8] Derived in Appendix 2 of Larminie and Dicks (2003).

8.4.6 Range Modelling of Hybrid Electric Vehicles

All the modelling we have done so far has involved equations which the system followed in a more or less predetermined way. However, when we come to a hybrid electric vehicle, then this is no longer so. Hybrid electric vehicles involve a controller, which monitors the various power in the system and the state of charge of the battery, and makes decisions about the power to be drawn from the engine, battery, and so on. Very little about the energy flow is inevitable and driven by fixed equations. Furthermore, the strategy will change with time, depending on issues such as when the vehicle was last used, the temperature, the need to equalise the charge in the batteries from time to time, and a host of other criteria. The decision making of these controllers is not at all simple to simulate.

Another complication is that there are so many different configurations of hybrid electric vehicles.

The result is that the simulation of these vehicles cannot be attempted or explained in a few pages of a book like this. Indeed, the use of a simple program such as MATLAB® on its own is probably not advisable. At the very least the Simulink extension to MATLAB® should be used. There are a number of vehicle simulation programs available or described in the literature, for example Bolognesi *et al.* (2001). Among the most well known of these is ADVISOR (Wipke *et al.*, 1999), which is MATLAB® based.

The program for the energy controller in a hybrid system, which is a subsystem of vital importance, will often be written in a high-level language such as C. It makes sense to incorporate the simulator in the same language, and then it should be possible to use the very same control program that is being written for the controller in the simulation. This has obvious reliability and efficiency benefits.

8.5 Simulations – A Summary

We have seen in this chapter how to begin the simulation or modelling of the range and performance of electric vehicles. With most vehicles the simulation of performance, by which we usually mean acceleration, is fairly straightforward. Mathematical software such as MATLAB® or Excel lends itself very well to this.

In the case of the classical battery-powered electric vehicle, and fuel cell vehicles using stored hydrogen, the modelling of range, though considerably more complex, is not difficult. For hybrid vehicles a great deal of care and thought is needed in setting up a simulation.

We have shown the fact that a vehicle mathematical model is crucially important in the design of electric vehicles, as it allows the designer very quickly to try out different design options, and at virtually no cost. By using examples of the GM EV1 electric car, and an electric scooter, we have shown how even a quite simple mathematical model is validated by the performance of the real vehicle.

It is almost impossible to overemphasise the importance of this modelling process in electric vehicle design. Hopefully these explanations, and the example MATLAB® script files we have given, will enable the reader to make a start in this process. Chan and Chau (2001) have described the use of an extension of this sort of programming in their simulation program EVSIM, and this could be studied to see how these ideas could be taken further.

References

Bolognesi, P., Conte, F.V., Lo Bianco, G. and Pasquali, M. (2001) Hy-Sim: a modular simulator for hybrid-electric vehicles. Proceedings of the 18th International Electric Vehicle Symposium (CD-ROM).

Chan, C.C. and Chau, K.T. (2001) *Modern Electric Vehicle Technology*, Oxford University Press, Oxford.

Larminie, J. and Dicks, A. (2003) *Fuel Cell Systems Explained*, 2nd edn, John Wiley & Sons, Ltd, Chichester.

Shnayerson, M. (1996) *The Car that Could*, Random House, New York.

Wipke, K., Cuddy, M., Bharathan, D. *et al*. (1999) ADVISOR 2.0: A Second Generation Advanced Vehicle Simulator for Systems Analysis. Report no. TP-540-25928, NREL, Golden, CO.