A model of an automatically shifted truck for prediction of longitudinal performance on an operating cycle

2nd Edition

P. Pettersson[a], B. Jacobson[a], and S. Berglund[a,b]

[a]Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Göteborg, Sweden
[b]Powertrain Engineering, Volvo AB, Göteborg, Sweden

5th November 2019

Contents

A note on the 2nd edition

This is a clean-up of the first edition where many errors and mistakes have been corrected. No new material has been added, though many updates have been made to the models; especially the implementations of the operating cycle and the driver. See [1–3] for these. The models are all available for download through `www.chalmers.se/vehprop`[1]

[1]The zip files that contain the library can also be found through `https://chalmersuniversity.app.box.com/s/f5ejzj18bh3c6z057ri71nf04er8g469`.

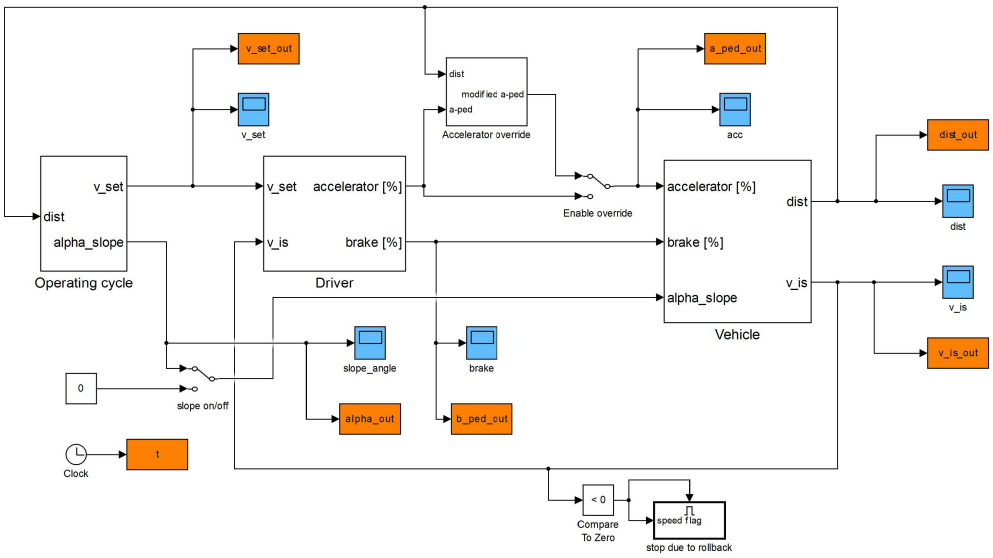

Figure 1: *Top level of VehProp*

1 Introduction

VehProp, for VEHicle PROPulsion, is a simulation model library for evaluating the longitudinal performance of a vehicle by predicting, for example; fuel consumption, productivity or clutch wear. The idea is to represent the forces on, and the torque transmitted through the powertrain during time intervals that are typical for conducting transport missions. This could mean anything from a minute to several days.

This report describes the library as parts of a full vehicle model. The model is based on a number of articles by Jacobson and Eriksson see e.g. [4, 5] as well as the master thesis by Venbrant [6]. The physical models are in turn based on work done in the PhD-thesis by Berglund [7] (engine) and the PhD-thesis by Jacobson [8] (transmission). The chassis equations follow more or less from Newton's Principia [9], but for a more recent treatment that also gives a vehicle dynamic perspective see the compendium by Jacobson [10].

2 Simulation model structure and concepts

The simulation model is largely based on a modular structure and the modules are supposed to be divided into components which are both intuitively and physically different. At top level the model is divided into the operating cycle (generalized road), the driver (who controls the vehicle) and the vehicle (the system under test), see Fig. 1.

The vehicle model here has been developed with the explicit purpose to reflect how the difference between a single clutch (SC) and dual clutch (DC) transmission influence the transport mission efficiency (e.g. average speed) and vehicle fatigue (e.g. clutch wear) during a realistic long-haul mission. As other performance measures become interesting, it could (and will) be developed in other directions to capture for these too.

It should be pointed out that the model uses forward simulation (sometimes called natural dynamic or causal) to better reflect reality. The opposite, a backward simulation (inverse dynamic or reverse causal), would be more effective but would not be able to capture gear transitions or engine dynamics in a good way. This is a major part of the model and thus a forward simulation is the canonical choice. The downside is that it makes a driver model necessary, something that adds considerable complexity.

3 Operating cycle

The operating cycle is very simple at the time of writing and uses only two road properties: speed and topography, but the principle is the same for many other effects related to road and environment.

Each property is either defined as a function of position (static properties e.g. road curvature) or time (dynamic properties, e.g. how long to stay with active PTO at a certain coordinate). The latter require a much more sophisticated treatment than the former and since no such property is used at present, it will not be treated further here.

Currently the operating cycle is a matrix OC_{ab} where the columns represent different properties and the rows each property value at the current coordinate. The current properties are: an index j (a counter that is included to easily see how big the matrix is), the position s (which is the governing variable), the target speed v (sometimes referred to as v_{set}), curvature κ, and altitude z or inclination angle θ (topography). In the implementation the continuous functions are replaced by discrete variants. In that case the operating cycle matrix can be written

$$OC_{ab} = \delta_{1b}k_a + \delta_{2b}s_a + \delta_{3b}v_{set,a} + \delta_{4b}\kappa_a + \delta_{5b}z_a \tag{1}$$

where δ_{ij} is the Kronecker delta. If the tensor notation is unfamiliar, a more convenient but less rigorous way of writing would perhaps be

$$OC_{ab} = [i, s, v, \kappa, z]_{ab} \tag{2}$$

Each column here represents a different value of b in Eq. (1). The input to the module, see Fig. 2, is the current position of the vehicle d, and a corresponding current index is calculated

$$i : d \in (s_i, s_{i+1}] \tag{3}$$

For many of the variables linear interpolation is used to find intermediary values. Other types of interpolation (or spline) methods could be used too, and what type that is most suitable is decided by the underlying physical model of each parameter. This will be further discussed in future articles specifically treating advanced operating cycle concepts. For a generic variable input variable x, using Eq. (3), the output

$$\tilde{y} = f(x, y_i) = \frac{x - x_i}{x_{i+1} - x_i}(y_{i+1} - y_i) + y_i \tag{4}$$

The implementation of the operating cycle can be seen in Fig. 2.

During a simulation, time is incremented in steps. To make the connection to the simulation more apparent, a time index k can be attached to (some of) the variables in Eq. (4). At iteration k in the simulation we would have:

$$t_k = \sum_{j=1}^{k} \Delta t_j \tag{5}$$

$$i : d_k \in (s_i, s_{i+1}) \tag{6}$$

$$\tilde{y}_k = f(x_k, y_i) \tag{7}$$

In the following there may be some slight abuse of the notation, as the value $\tilde{y}_k$ which changes in each steep and is sent around between modules in the simulation (the *signal*) may be referred to by the same name as the *parameter* y in the operating cycle, which is completely static.

3.1 Speed

The prescribed speed v_{set} is output as

$$v_{set} = f(d, v_{set,i}) \tag{8}$$

Here the notational abuse is apparent, as the left hand side is the dynamic value which changes with each time step k, but the right hand side is static. A better notation would be to make the time dependence apparent and use Eq. (1).

$$v_{set,k} = f(d_k, OC_{i3}) \tag{9}$$

In the implementation, the signal is called **v_set**.

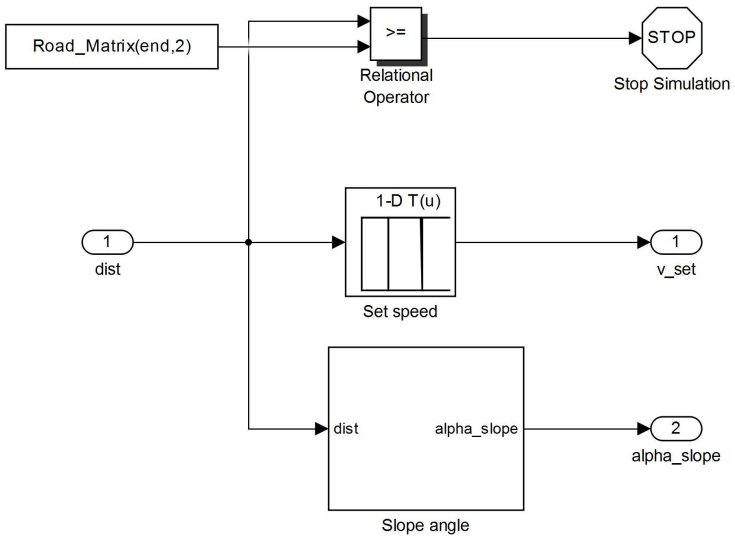

Figure 2: *The operating cycle module*

3.2 Topography

The topography can either be given as a road angle or altitude. If given as a road angle, it is represented by a function, θ, that is piecewise constant over intervals of 25 m. Thus the road angle α that is output from the operating cycle is

$$\alpha = \theta_i \tag{10}$$

One could of course outfit this with a time index too: α_k. In the simulation the signal is called **alpha_slope**.

If the topography is given in altitude z, the output road angle is calculated by looking at the elevation of the front part of the vehicle (z_f) and comparing it to the rear part (z_r)

$$z_f = f(d, z_i) \tag{11}$$
$$z_r = f(d - l, z_i), \tag{12}$$
$$\alpha = \arcsin\left(\frac{z_f - z_r}{l}\right) \tag{13}$$

where l is the wheelbase. In the implementation, the slope angle is called **alpha_slope**.

4 Driver

The driver module of course represents the human driving the vehicle. The input is the current speed v_{is} and the target speed v_{set}. The outputs are the (normalized) accelerator and brake pedal positions. A speed error is formed as

$$e_k = v_{set,k} - v_{is,k} \tag{14}$$

and a PID-regulator is used to regulate the driver pedal positions. Writing the output from this as $p(e_k)$, the pedal positions will be output as

$$p(e_k) \geq 0 \Rightarrow a_p = p,\ b_p = 0 \tag{15}$$
$$p(e_k) < 0 \Rightarrow a_p = 0,\ b_p = -p \tag{16}$$

In the implementation the Matlab/Simulink default PID-regulator is used and apart from the basics features it also includes some filtering and a useful anti-windup mechanism. A figure of the module can be seen in Fig. 3.

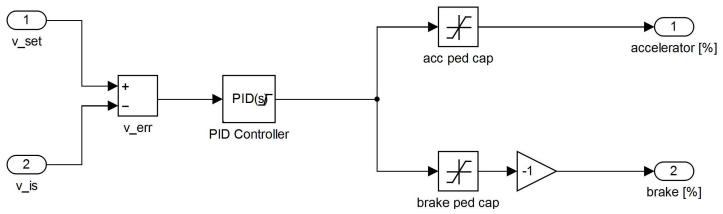

Figure 3: *The driver module*

5 Vehicle

The vehicle is by far the most complex module at the time of writing. The top level can be seen in Fig. 4 and it consists of three main sub models: engine, transmission and chassis. The inputs are the accelerator pedal position a_p, the brake pedal position b_p and the road angle α. The outputs are the travelled distance d (which for a one dimensional description with isotropic direction coincides with the position) and the current speed v_{is}. The physical vehicle model can be seen in Fig. 5.

5.1 Engine

The engine module inputs are the accelerator pedal position a_p, a torque limit request flag, torque limit specifier T_{lim} and the engine speed ω_e. The module is shown in Fig. 6. It outputs an engine torque T_e and fuel volume flow rate $\dot{V}$.

This is a simple, empirical engine model with only two dynamic states (ω_e and T_e), but still we strive to have it as physical as possible. The pedal position is reinterpreted as a torque request using a pedal map $f_{pedal}(a_p, \omega_e)$, the torque request is checked against both the maximum torque at current engine speed and the torque limit request from the transmission. The limiting value is chosen and transformed into required amount of fuel q injected into each cylinder, based on a fuel map $f_{fuel}(T, \omega_e)$. The fuel is injected into the cylinders and a steady state torque is calculated from an engine map $f_{engine}(q, \omega_e)$. Depending on its size, the torque it is either output directly or split into a base part and a top part. The former is instantaneous, and the latter is subject to a first order differential equation, to mimic a boost pressure build-up. If the requested top part is lower than the current top value, the change is immediate. The final output torque is the sum of those parts. In the form of the equations

$$T_{raw} = f_{pedal}(a_p, \omega_e) \tag{17}$$
$$T_{req} = \min(T_{req}, T_{max}(\omega_e), T_{lim}) \tag{18}$$
$$q = f_{fuel}(T_{req}, \omega_e) \tag{19}$$
$$T_{ss} = f_{engine}(q, \omega_e) \tag{20}$$
$$T_{base} = min(T_{ss}, T_{split}) \tag{21}$$
$$T_{dyn,req} = T_{ss} - T_{base} \tag{22}$$

$$T_{dyn,req} - T_{top} < 0$$
$$T_{top} = T_{dyn,req} \tag{23a}$$
$$T_{dyn,req} - T_{top} \geq 0$$
$$\dot{T}_{top} = k(T_{dyn,req} - T_{top}) \tag{23b}$$

$$T_e = T_{base} + T_{top} \tag{24}$$

where coefficient $k > 0$ decides the build-up rate. The reason for Eqs. (23a) and (23b) is that the process of reducing the boost pressure is much quicker than building it up. In principle one could handle it by Eq. (23b) only and using different coefficients depending on whether $\dot{T}$ is positive or negative.

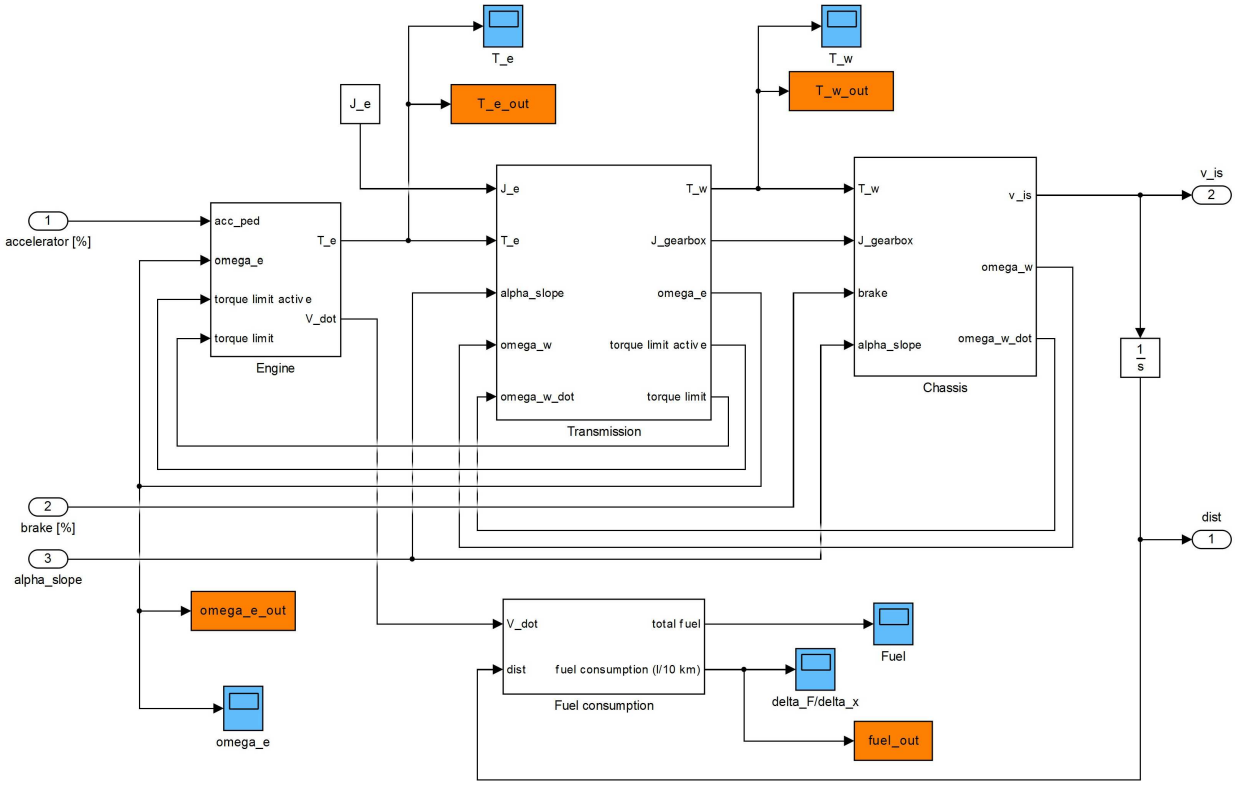

Figure 4: *The vehicle module*

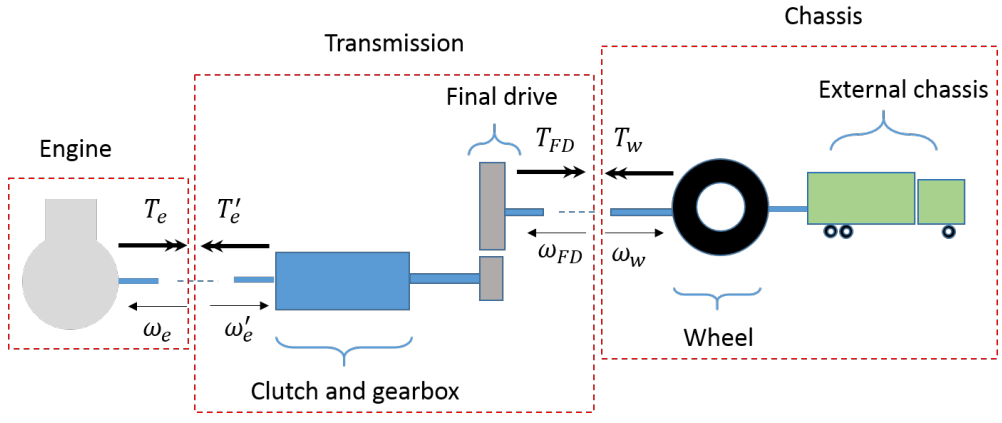

Figure 5: *The physical vehicle model. The engine and transmission together make up the powertrain model, which is the focal point in the simulation.*

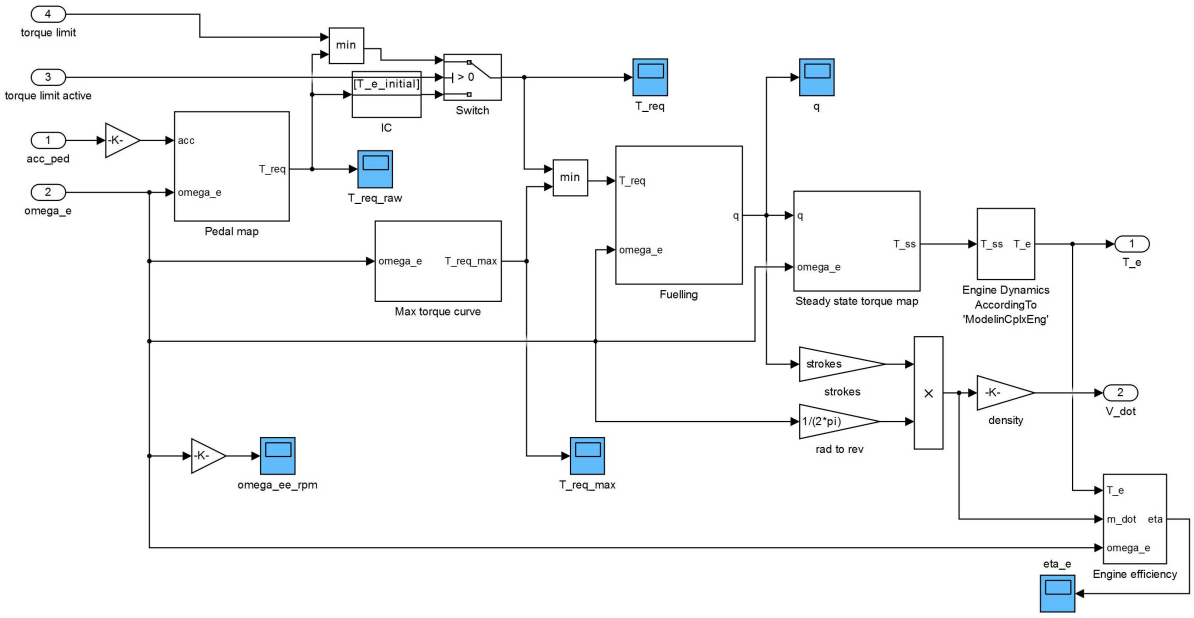

Figure 6: *The engine module*

There is also another engine layout where the engine efficiency is available. Then the fuelling can be skipped since the request can be directly translated into a steady state torque and fuel rate computed from the efficiency map. This approach requires fewer computations and is therefore quicker, but less physical. As long as the fuelling process contains no dynamics these two approaches are equivalent.

Note that in the above treatment we have not really made a clear distinction between physical components, but rather between different parts in the request chain. In reality the case is rather that the pedal actuation is received by the main engine ECU, which relates it, via a fuel and engine map, to amount of fuel to inject and then controls the injection process. Thus Eqs. (17) and (18) should really be in a control component, whereas the others belong in mechanical components. For the current implementation the distinction is irrelevant but may need to change if the complexity of the engine grows.

The variable q is the amount of fuel (in mg) to inject per cylinder (injection) stroke. In the end we are interested in the volume of fuel consumed and need to relate these two quantities. It can be done by simply using dimensional analysis: $[q] = \text{mg/stroke}$, and thus we need

$$\dot{m} = a\omega_e n_s q = c\frac{N}{n}\frac{\omega_e}{2\pi}q \tag{25}$$

ω_e has the unit rad/s and therefore we have included a numerical n_s which relates the number of strokes per second to rotation speed. This depends on the number of cylinders (N) and the type of combustion process: how many strokes (n) that are needed to complete one full thermodynamic cycle. For example: a two stroke engine need only one complete turn ($n = 1$) but a four stroke engine needs two complete turns ($n = 2$) per injection. The constant c is a simple unit conversion factor: $c = 10^{-6}$ kg/mg. So for a six cylinder, four stroke engine

$$\dot{m} = \frac{3 \cdot 10^{-6}}{2\pi} q\omega_e \tag{26}$$

with $\dot{m}$ the fuel (mass) flow rate in SI-units. Some things are of interest from this: the total volume of consumed fuel V_{fuel} and the engine efficiency η. With $\zeta = 43.1$ MJ/kg the calorific energy content of diesel and $\rho = 0.832 \cdot 10^3$ kg/m^3 the density, we have

$$V_{fuel} = \frac{1}{\rho}\int_{t_0}^{t_f} \dot{m}\,\mathrm{d}t \tag{27}$$

$$\eta = \frac{T_e\omega_e}{\zeta\dot{m}} \tag{28}$$

5.2 Transmission

The transmission is the most complex part and relies heavily on the input from various control strategies. The input variables are the engine inertia J_e, the engine torque T_e, the slope angle α, the wheel angular speed ω_w and the wheel angular acceleration $\dot{\omega}_w$. The outputs are the wheel torque T_w, the complete inertia of all components up to the final drive J_{tot}, the engine speed ω_e, a torque limit request flag and a torque limit specifier T_{lim}.

The physical model of the clutch and the gearbox is shown in Fig. 7. In addition to this there is a final gear afterwards, see Fig. 5. The cut between the engine and transmission in this figure is trivial, so

$$T_e - T_e' = 0 \tag{29}$$

$$\omega_e - \omega_e' = 0 \tag{30}$$

Since $T_e' = T_e$ and $\omega_e' = \omega_e$ the prime notation will be dropped to avoid unnecessary clutter.

Figure 7 shows a dual clutch gearbox. In this case one of the clutches controls the odd gear set and the other the even one. Both of them are dry clutches and we have chosen to model them strictly with Coulomb friction. Therefore each one operates in two regimes: one where it sticks and one where it slips. The properties change based on the discrete state. Since there are two clutches there are four different discrete states in total. The physical model of the gearbox can be seen in Fig. 8. The equations fall out as

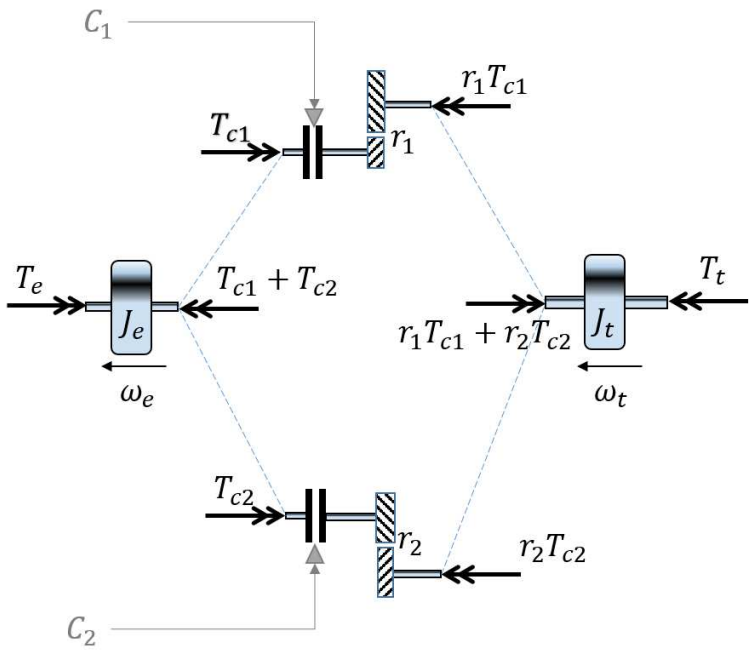

Figure 7: *The physical model of the gearbox.*

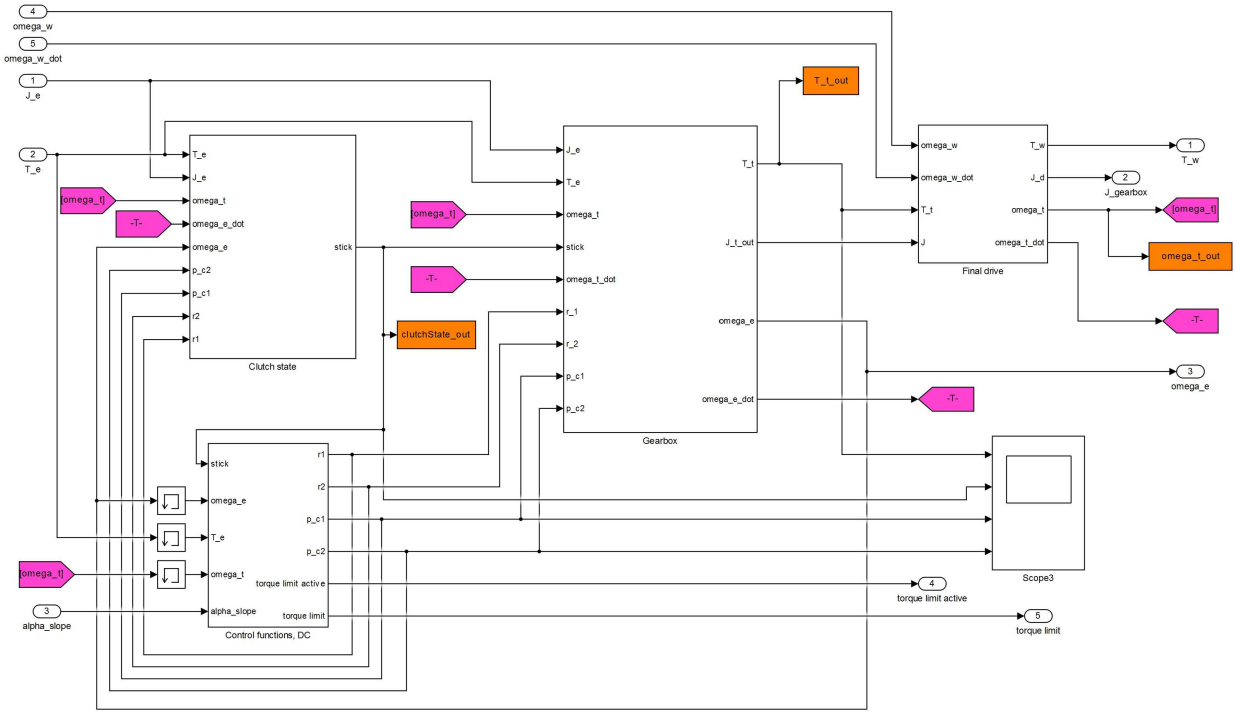

Figure 8: *The transmission module*

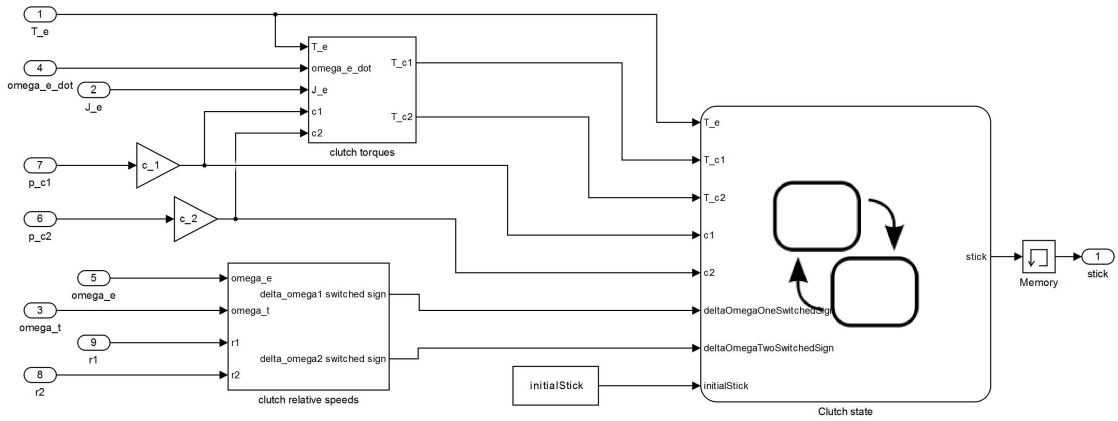

Figure 9: *The clutch module*

follows

$$J_e\dot{\omega}_e = T_e - (T_{c_1} + T_{c_2}) \tag{31}$$

$$J_t\dot{\omega}_t = -T_t + (r_1 T_{c_1} + r_2 T_{c_2}) \tag{32}$$

$$\Delta\omega_i = \omega_e - r_i\omega_t,\ i = 1, 2 \tag{33}$$

$$T_{c_i} = \begin{cases} C_i \operatorname{sgn}(\Delta\omega_i), & \text{if } c_i \text{ slip} \\ \text{from Eqs.}, & \text{if } c_i \text{ stick} \end{cases} \tag{34}$$

$$C_i = c_{max} p_i \tag{35}$$

$$\text{slip condition:} \quad T_{c_i} > C_i \tag{36}$$

$$\text{stick condition:} \quad \Delta\omega_i = 0 \tag{37}$$

One of the discrete states, the stick-stick state, is thoroughly uninteresting because it cannot happen while driving. If both of the clutches stick then the engine crankshaft and the clutch output shafts need to have the same speed: the only speed that can satisfy that equality is zero. The state could be used as a parking brake, but this is not something that we are interested in modelling.

The conditions for transitioning between the states in Eqs. (36) and (37) are indirectly controlled by the clutch pressures p_i. These are in turn controlled via some actuators depending on the engineering solution (valves for a hydraulic or pneumatic solution) by the transmission ECU.

After the gearbox there is a final drive gear, where also a total transmission efficiency η_T has been included

$$T_{FD} = \eta_T r_{FD} T_t \tag{38}$$

$$\omega_{FD} = \frac{\omega_t}{r_{FD}} \tag{39}$$

These are the same as the torque and rotational speed of the wheels, see (53) and (54).

5.2.1 Implementation of the mathematical model

The implementation separates into three main parts (the top three modules in Fig. 8): a state machine for the clutch, a module for the gearbox and a module for the final drive. The clutch module and its state machine can be seen in Figs. 9 and 10. A limitation in the implementation is that the clutch state must be delayed one time step, so the switch between states will always be one tick behind. The reason is that

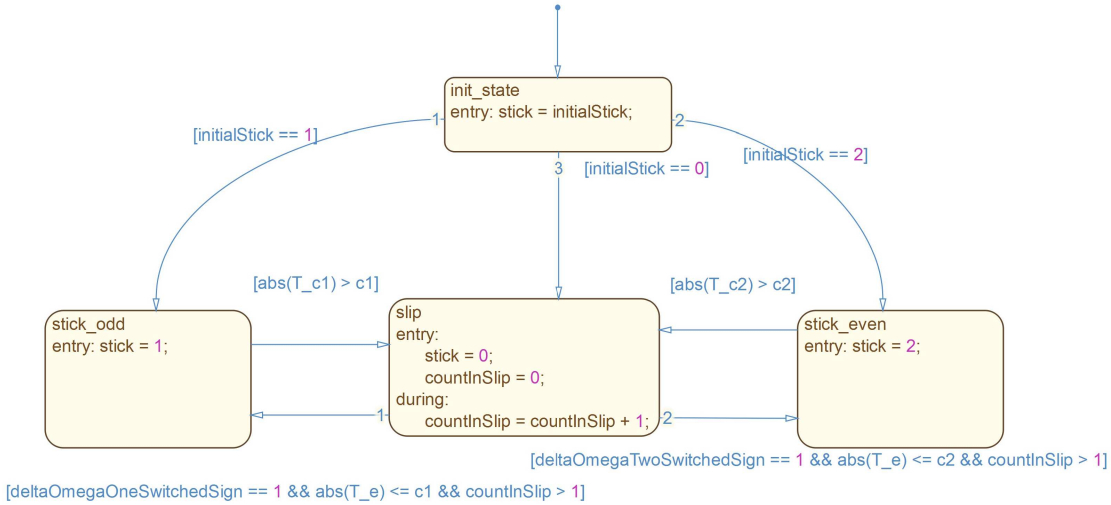

Figure 10: *The clutch state machine*

the Eqs. (31) to (36) make up a set of differential algebraic equations (DAE), and since Simulink is strictly causal it has trouble solving them. However, when the clutch state is already known the problem set reduces to a system of ordinary differential equations (ODE) instead, and those Simulink excels at handling. Also note that the stick-stick state is missing from the clutch state machine (though it could easily be introduced).

The gearbox module is shown in Fig. 11 and all the submodules can be seen in Figs. 12, 13 and 14.

There are some things that may be of note here. There is one submodule for each state: odd stick/even slip, odd slip/even slip and odd slip/even stick. All of them compute their output in each tick, and the signal router (Fig. 15) manages which of them to send further on and which ones to block. The odd stick/even slip and odd slip/even stick are identical under the transformation $1 \leftrightarrow 2$ in Eqs. (31)-(37) of course, but the slip-slip equations are a little bit different. The reason is that the number of dynamic states change here: when one of the clutches stick the input and output shaft speeds are linearly dependent (the engine speed depends on the transmission shaft speed through the gear ratio, which in the end depends on the vehicle speed), but when both clutches slip there is no such relation and these two speeds evolve independently. Thus whenever the slip-slip state is entered the engine speed needs to be decoupled and reinitialized.

Another thing that should be pointed out is that the inertias are sent onwards and the equations that are implemented are somewhat different from those presented so far. This is another relic from the choice of Simulink as implementation language: as noted above the equations are linearly dependent and, together with the relations for the chassis, must be solved as a system. Simulink cannot treat non-causalities, so we basically have to solve it by ourselves and implement it in a way so that and output (vehicle acceleration) can be computed from and input (engine torque). When doing that, all the inertias (both rotational and mass) bunch together. The final solving step is done in the chassis module when finding the linear vehicle speed, so the inertias need to be sent onwards until that point (see section 5.3).

The final module concerning the mechanical system is the final drive in Fig. 16.

5.2.2 Control modules

The control module (which would be equivalent to a transmission ECU) is shown in Fig. 17. Three functions are included: the gearshift manager (which includes the gearshift strategy and decides when a gearshift is complete), the clutch pressure control (which manages the clutches), and an engine torque limiter (which communicates with the engine module).

Gearshift manager

The gearshift manager (Fig. 18) decides when to initiate a gearshift and which gear to engage. In principle,

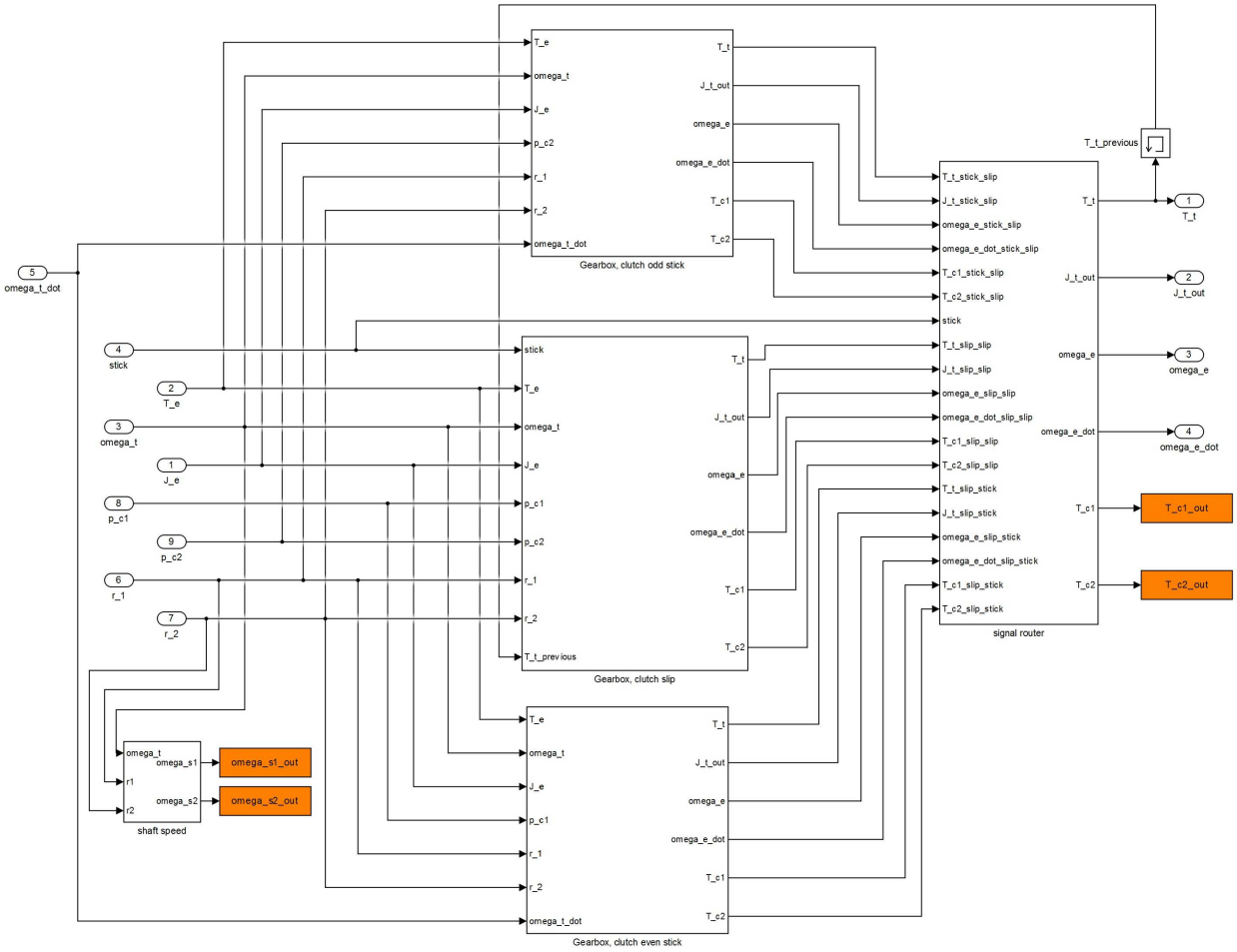

Figure 11: *The gearbox module*

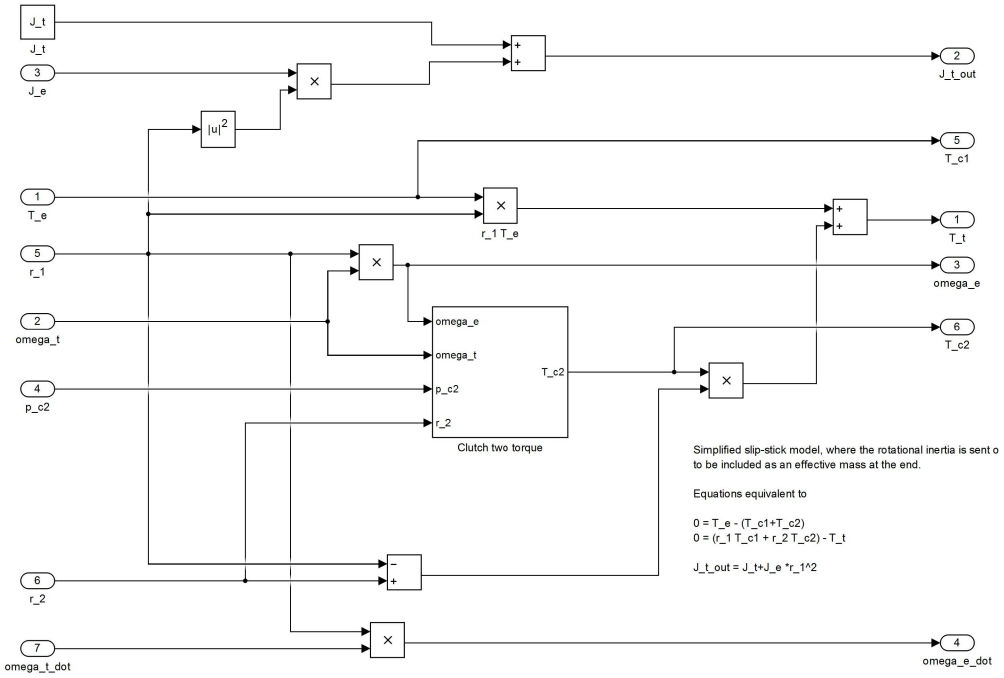

Figure 12: *The gearbox details when the odd clutch sticks ($c_1 = c_o$)*

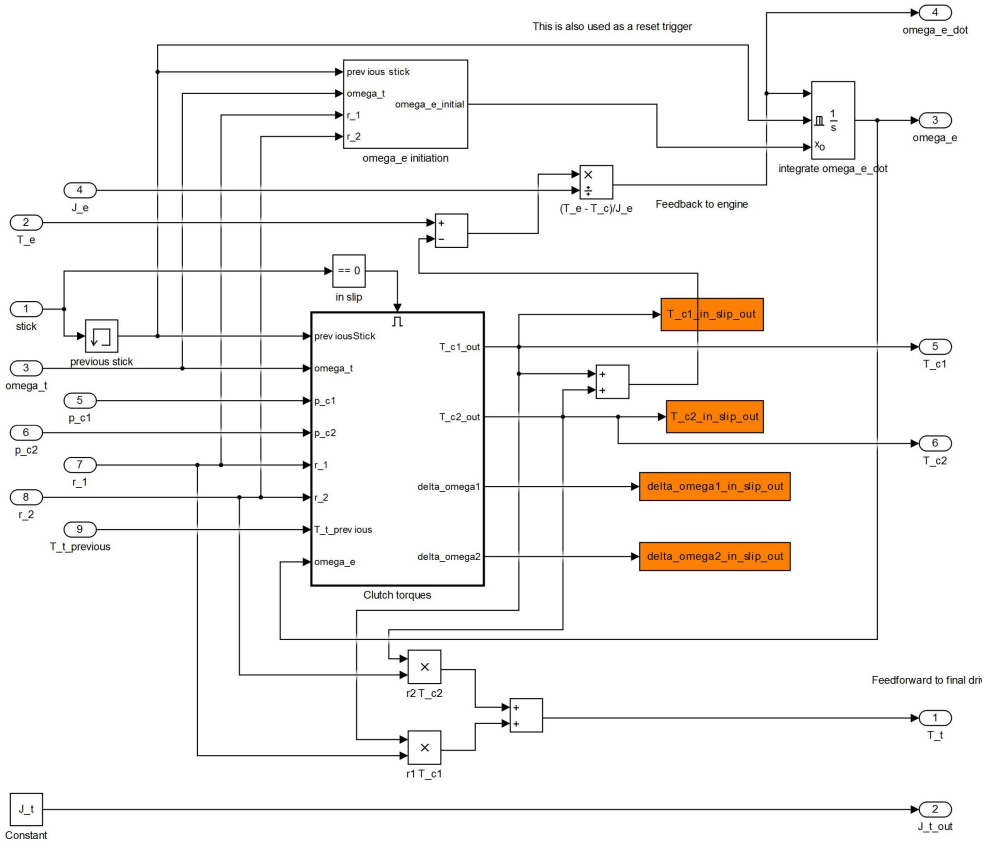

Figure 13: *The gearbox details when both clutches slip*

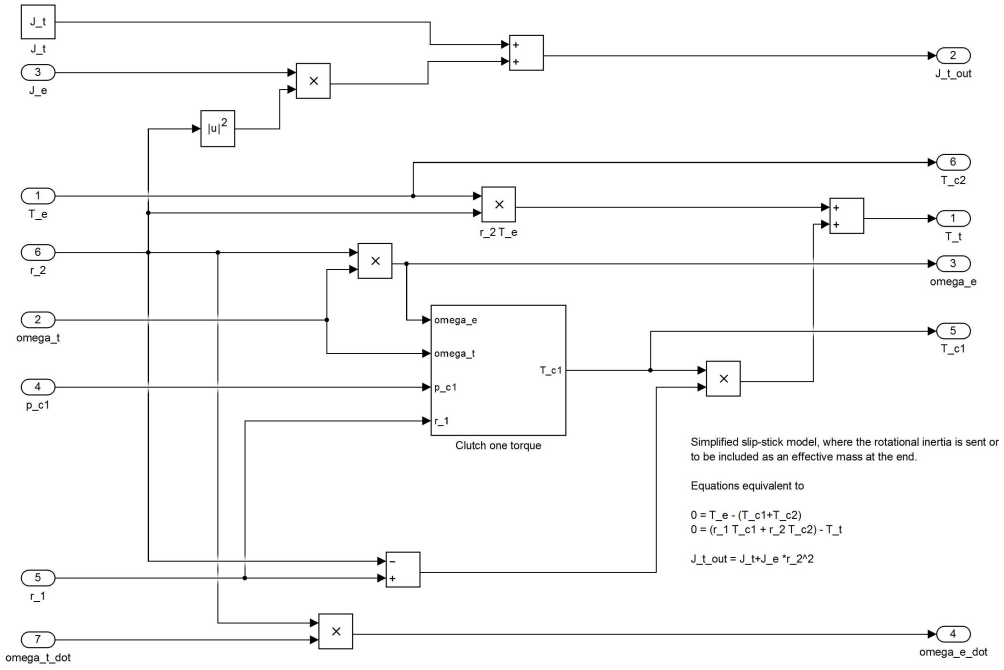

Figure 14: *The gearbox details when the even clutch sticks ($c_2 = c_e$)*

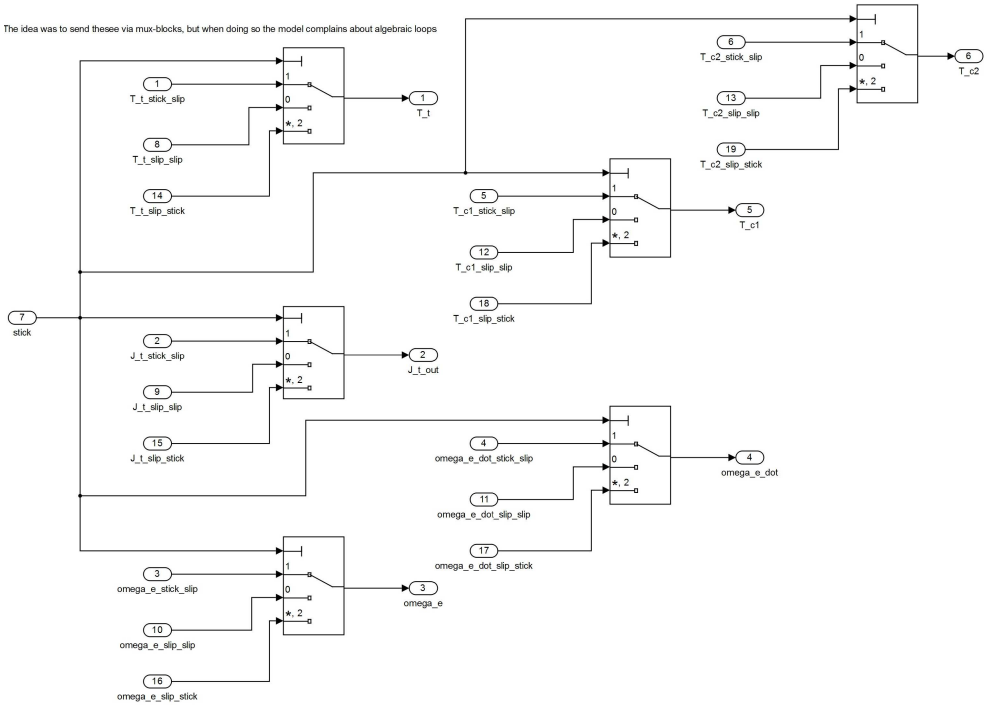

Figure 15: *The signal router*

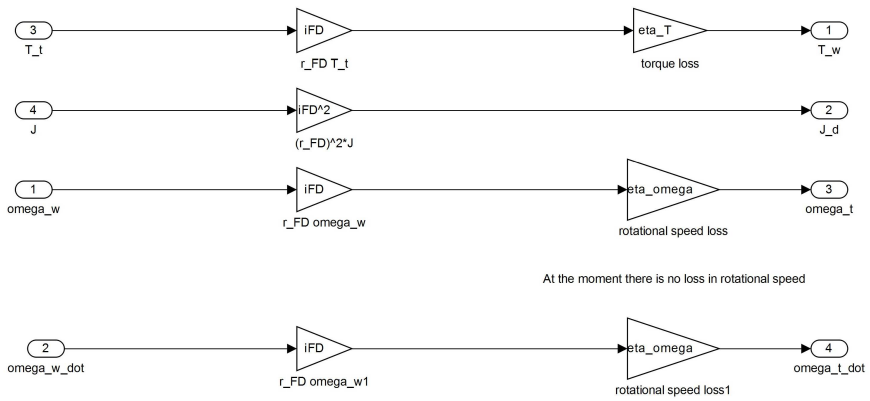

Figure 16: *The final drive module*

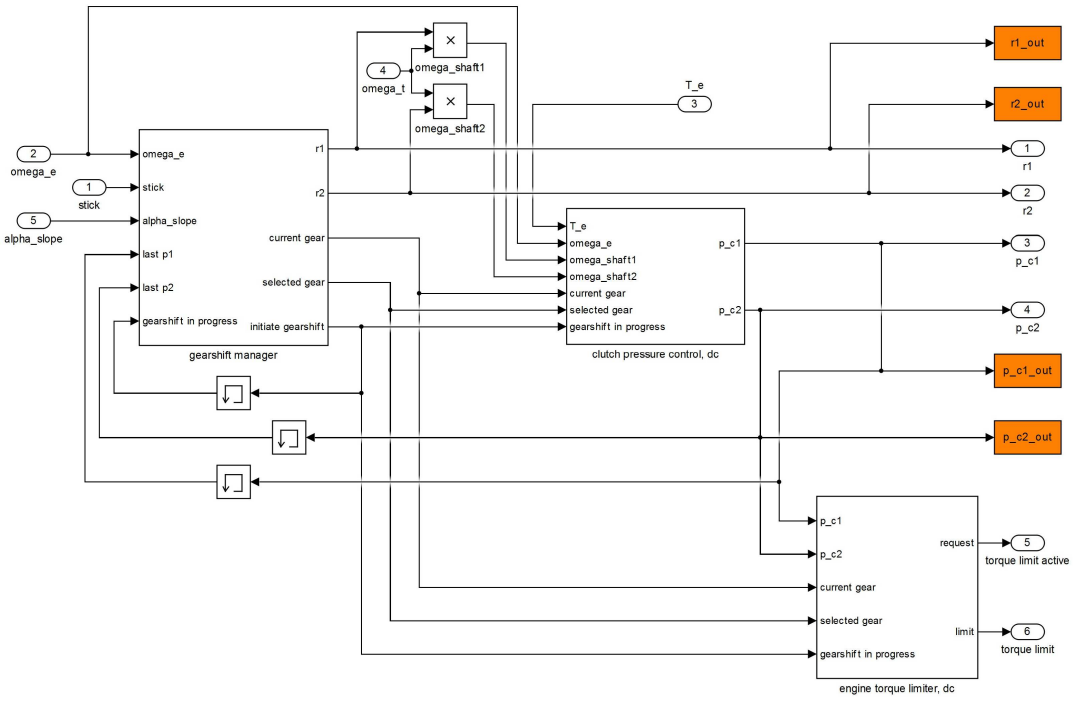

Figure 17: *The control function module*

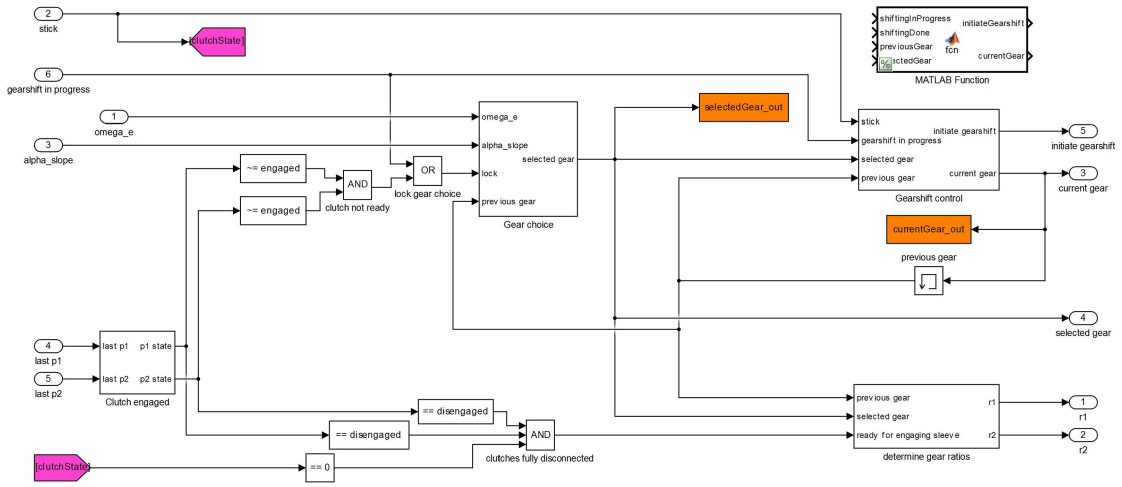

Figure 18: *The gearshift manager*

a gearshift is requested if the current gear is different from the selected (or desired, rather), no other gearshift request is active, and no shift has been performed for the past t_{min} seconds.

The gear choice (Fig. 19) is simple and depends only on engine speed and the inclination angle. If the engine speed goes below the downshift threshold a downshift is initiated and vice versa for an upshift. The inclination angle has discrete effect providing regimes in which different engine speed thresholds are utilized. Typically the engine speeds need to be higher for both up- and downshift in a downhill (meaning later upshifts and earlier downshifts, to increase the performance) and the other way around in a downhill (for fuel efficiency). The number of gears to shift is similarly dependent on current gear and inclination. If the current gear is denoted by i_c, the gear request by i_{req}, the number of gears to shift $N(i_c, \alpha)$ and the up and down limits by $\omega_{u,lim}(i_c, \alpha)$ and $\omega_{d,lim}(i_c, \alpha)$ respectively, then

$$\omega_e \geq \omega_{u,lim}(i_c, \alpha) \;\Rightarrow\; i_{req} = i_c + N_u(i_c, \alpha) \tag{40}$$

$$\omega_e < \omega_{d,lim}(i_c, \alpha) \;\Rightarrow\; i_{req} = i_c - N_d(i_c, \alpha) \tag{41}$$

For a dual clutch transmission the gearshift is typically sequential by necessity: otherwise there would need to be a disruption in the power transfer. This is not the case for an SC, where non-sequential gearshifts are used extensively to improve fuel economy. Therefore numerical representations of the functions $\omega_{u,lim}(i_c, \alpha)$, $\omega_{d,lim}(i_c, \alpha)$, $N_u(i_c, \alpha)$ and $N_d(i_c, \alpha)$ are different for SC and DC.

The gearshift controller (Fig. 20) makes the decision on when to command a gearshift and what to specify as the current gear.

Clutch pressure control

The second function, shown in Fig. 21, is the clutch manager (or clutch pressure control), and it manages the pressures in Eqs. (35). Here we imagine that the pressure is directly controlled by the module and not indirectly by some valves, but to get it somewhat more realistic the pressure actuation is linear (and not instantaneous) with specific rise and fall times.

There are two parts of it, one controller (Fig. 22) and one actuator (Fig. 23).

It should be noted that although the physical model has two clutches, the clutch control can be tailored in such a way that it behaves as a single clutch transmission too. If the disengaging clutch fully disconnects and there is a time delay before the connecting one engages, the disruption in power transfer mimics that of a single clutch. Thus the only thing that needs to be replaced when modelling a vehicle with an SC instead is the clutch control strategy.

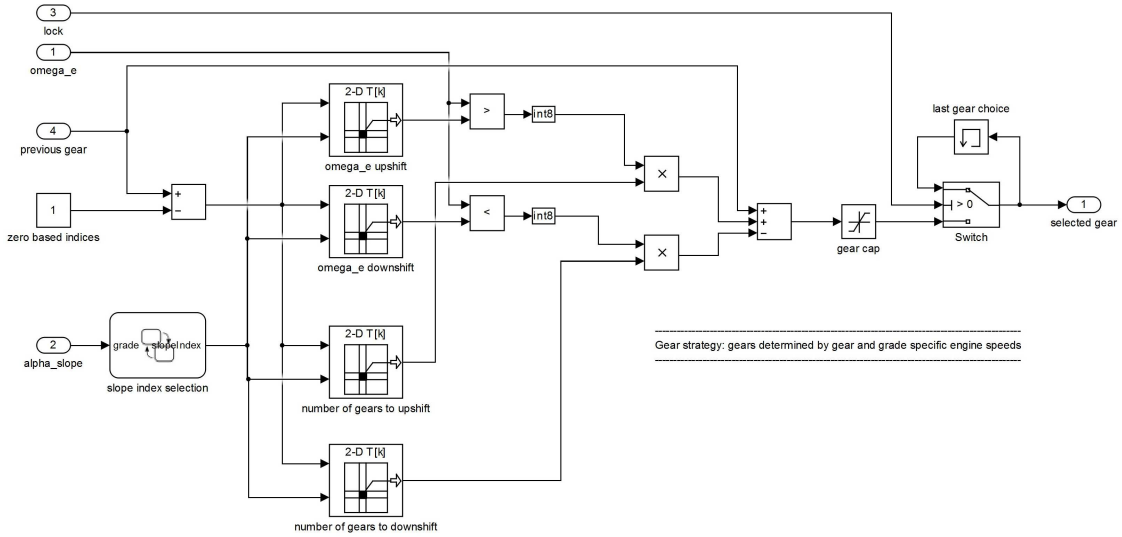

Figure 19: *The gear choice strategy*

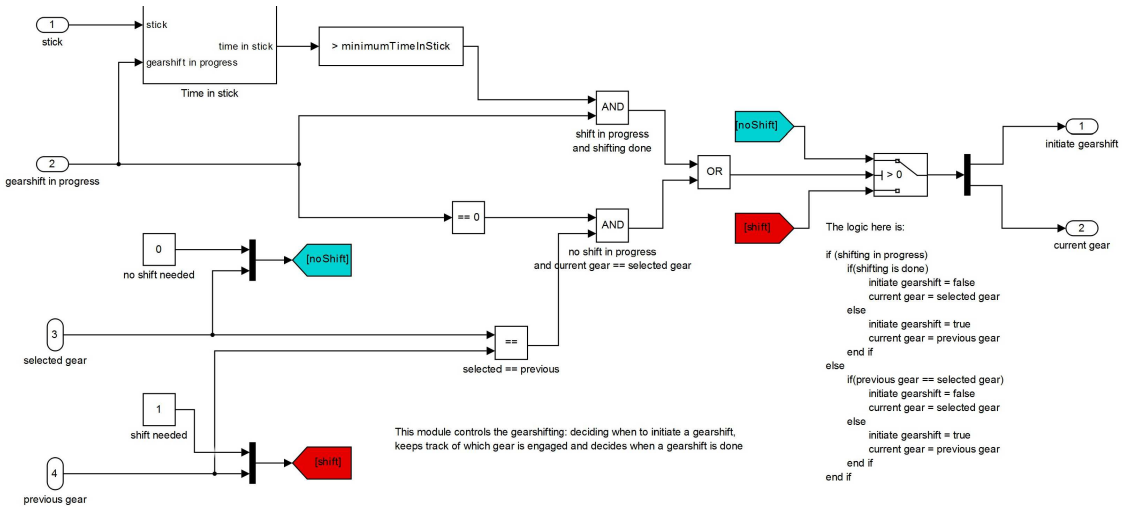

Figure 20: *The gearshift controller*

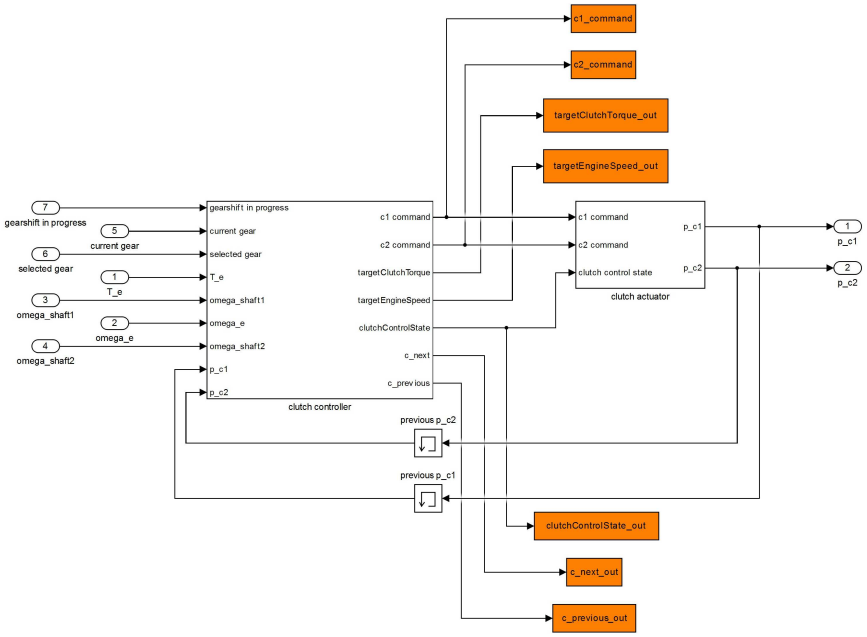

Figure 21: *The clutch pressure control*

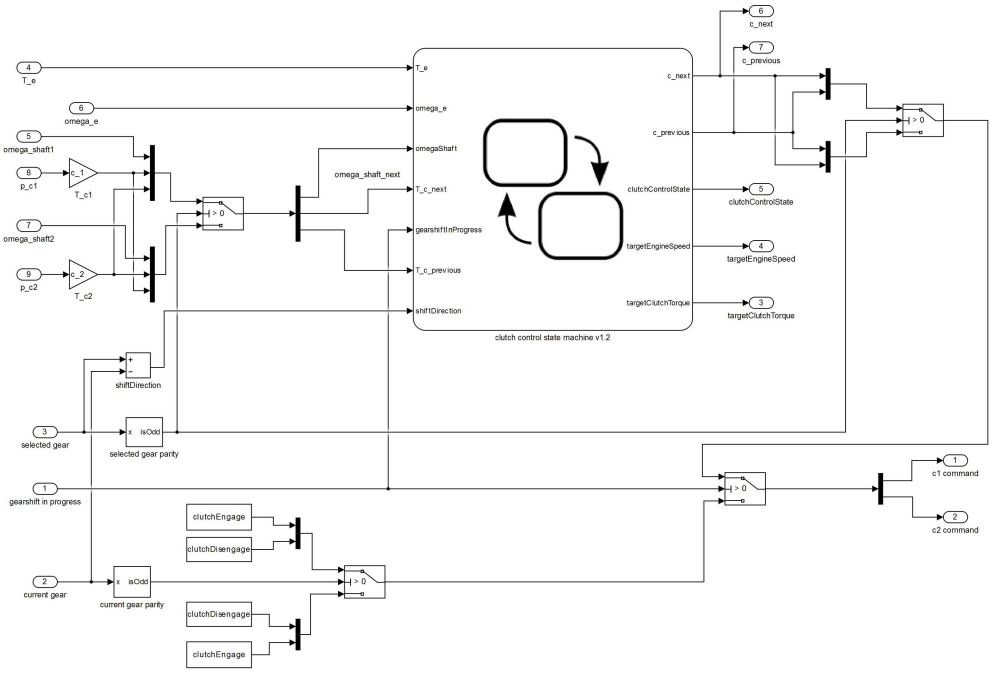

Figure 22: *The clutch controller*

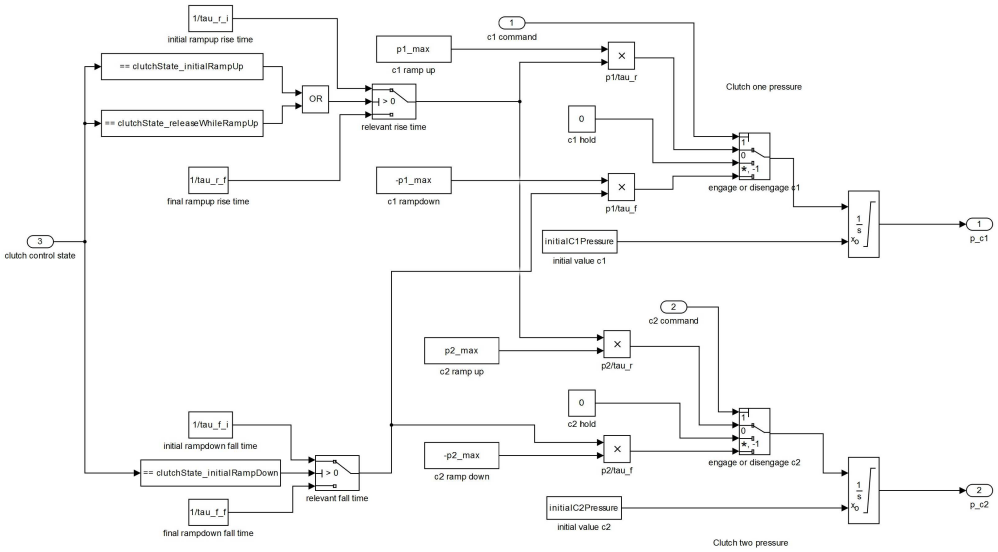

Figure 23: *The clutch actuator*

The controller (Fig. 22) is based around the clutch control state machine (Fig. 24), with some signal routing to decide which clutch is disengaging and which one is engaging, as well. The useful output is the clutch commands k_1 (odd clutch) and k_2 (even clutch), with $k_i = -1$ disengage, $k_i = 0$ hold, and $k_i = 1$ engage. The clutch state is also passed onto the actuator module.

Two different control concepts have been implemented. The first is that of a single clutch as explained above. Here, the control is very simple, whenever a gearshift is detected the disengaging clutch $k_{previous} = -1, k_{next} = -1$, and after a time delay t_d (to simulate the power disruption) $k_{previous} = -1, k_{next} = 1$. It could be done in a much more refined way: e.g. waiting for shaft synchronization before reconnecting to reduce heat generation and thus clutch wear. At the time of writing this is remains to be done. The engine speed control is performed by the engine itself, but in a controlled fashion via the torque limit module.

For the dual clutch, the strategy needs to be more refined: the engine is never fully disconnected from the transmission so it cannot be controlled in that way to the same extent. Instead the clutch pressures are steered so that the transferred torque either brakes the engine (upshift) or accelerates it (downshift). Figures 25 and 26 show the principle.

These are based on the idea of a desired gearshift time t_g and the assumption that the vehicle speed is unchanged during gearshift. Let the gear ratio of the current gear be r_c and the gear ratio of the next gear r_n. Then Eq. (31) can be used to find the transmission torque (again, assuming both this and the engine torque are constant) that achieves this in the desired time

$$\omega_e(t) = \frac{T_e - (T_{c_c} + T_{c_n})}{J_e} t + \omega_e(t_0) \tag{42}$$

$$v(t_0) = v(t_g) \Rightarrow \omega_e(t_g) = \frac{r_n}{r_c} \omega_e(t_0) \tag{43}$$

$$\Delta\omega_e = \omega_e(t_g) - \omega_e(t_0) = \left(\frac{r_n}{r_c} - 1 \right) \omega_e(t_0) \tag{44}$$

$$T_{c_c} + T_{c_n} = T_e + \frac{J_e \Delta\omega_e}{t_g} \tag{45}$$

In the case of a downshift, the engaging clutch is initially disregarded ($T_{c_n} = 0$) and disengaging clutch torque T_{c_c} is found (smaller than the engine torque) using Eq. (45). In the case of an upshift the engaging clutch torque is found (larger than the engine torque) in the same way, then the engaging clutch and the disengaging one are actuated simultaneously.

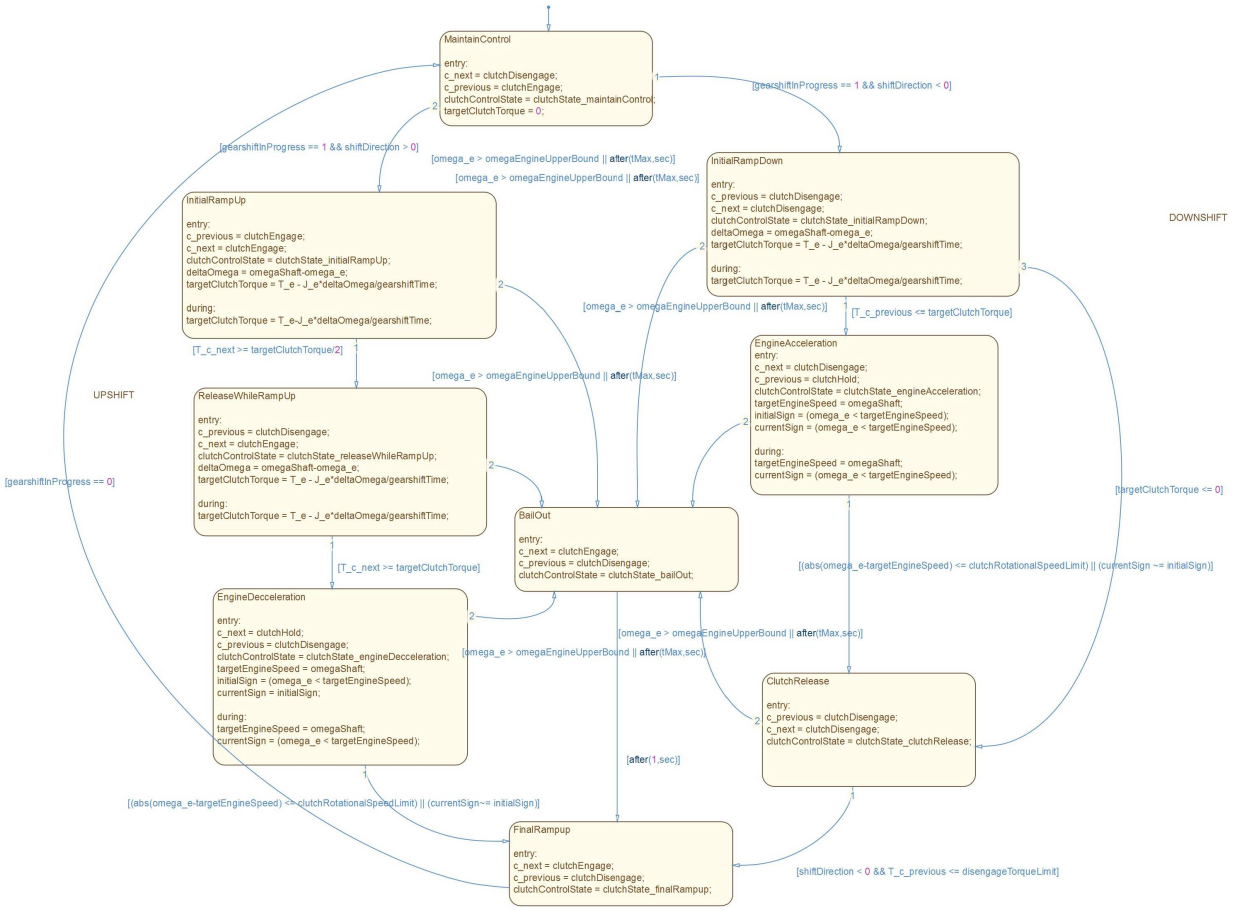

Figure 24: *The clutch control state machine*

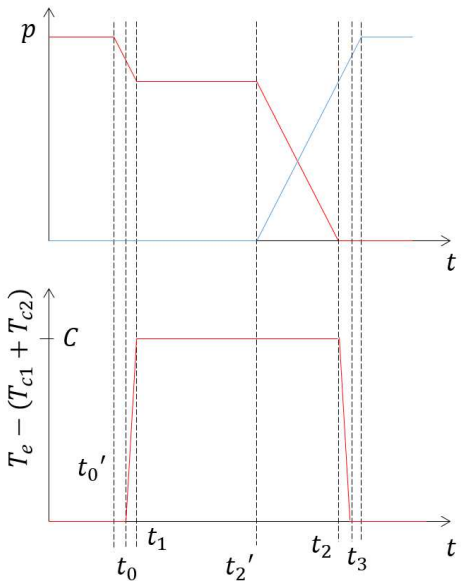

Figure 25: *The pressure control principle for a downshift. Note that $C > 0$ so the engine accelerates.*

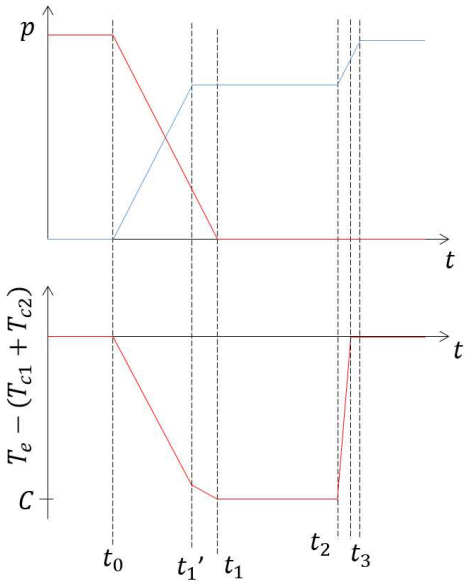

Figure 26: *The pressure control principle for an upshift. Note that $C < 0$ and the engine decelerates.*

Since the clutch actuators are not ideal, the equations above are only an approximation of the actual behaviour of the engine speed so there must also be some kind of fall back when things do not work out. That is the reason for the BailOut state.

The final component here is the clutch actuator in Fig. 23, that relates the commands from the controller into the (normalized) pressures p_i. The connection between k_i and p_i is

$$\dot{p}_i = \frac{k_i}{\tau_s},\ p_i \in [0, 1] \tag{46}$$

with τ_s being a rise or fall time, whose value depends on the clutch state and whether the pressure is to be increased or decreased. Equation (46) implies that the pressure change is not instantaneous but linear, so even though the description of the clutch pressure is not based on a physical model (e.g. hydraulic or pneumatic), it is not a fully idealized model.

Engine torque limiter

The engine torque limiter in Fig. 27 is an essential component for the single clutch transmission but not at all as important for the dual clutch, the reverse situation from the clutch control. The control principle can be seen in Figs. 28 and 29.

Both the timing and the limit levels here are fixed, depending on whether it is an upshift or a downshift and the severity of the inclination. A more realistic and effective approach would be to have these dynamic, i.e. compute both the torque request and the ramping start and end points from e.g. a desired gearshift time. This is something that remains to be done.

Currently, the torque limit is based on the clutch pressures only. For the single clutch gearshifts the torque limit request can be written as follows

$$T_{lim} = \begin{cases} \infty, & p_c > p_{break} \\ c_{max} p_c + T_s, & p_c \leq p_{break} \\ T_0, & p_c = 0,\ p_n = 0 \\ c_{max} p_n + T_s, & p_n \leq p_{break} \\ \infty, & p_n > p_{break} \end{cases} \tag{47}$$

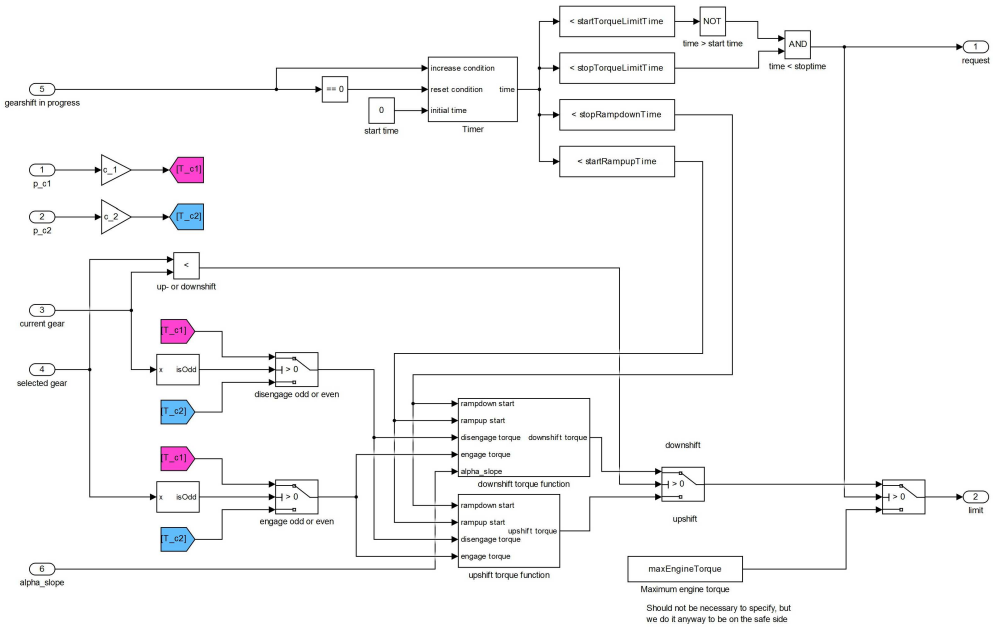

Figure 27: *The engine torque limit module*

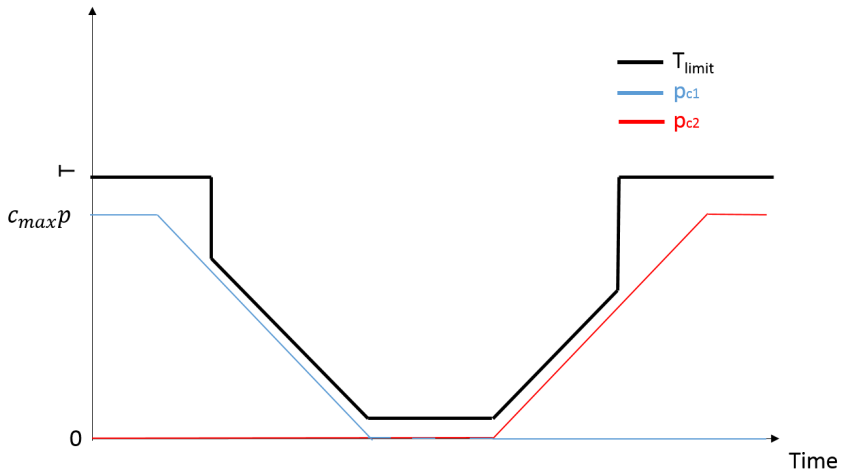

Figure 28: *The principle for the engine torque limiter during a downshift*

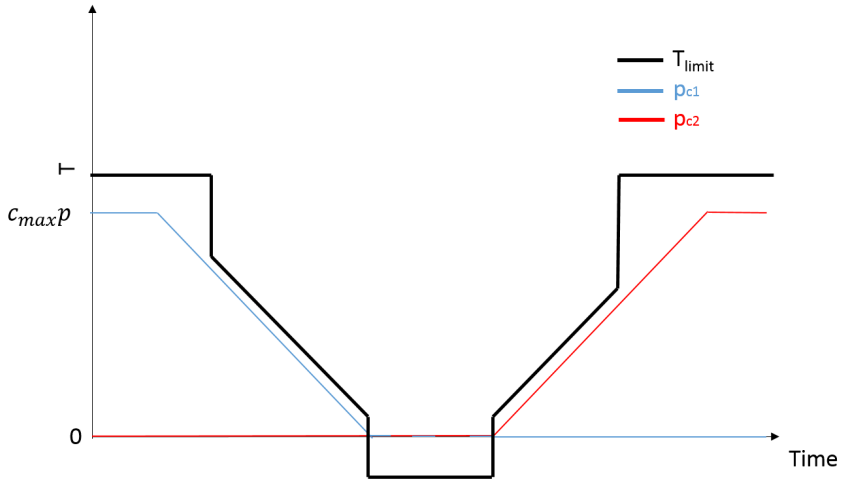

Figure 29: *The principle for the engine torque limiter during an upshift*

where we have used the current (subscript c) and next (subscript n) indices as in Eq. (42). In the end the only thing that differs between the upshift and downshift is the final torque T_0 during full declutch. For a downshift it is positive and for an upshift it is negative.

Note that this is only a limit request, then it is up to the engine to accept and perform the necessary actions to output the torque. In our case the engine is simple enough that this is never a problem, but in a real vehicle there certainly could be.

5.2.3 Off-line computations

There are lots of interesting things that can be computed using output from the simulation, but that do not necessarily need to be done during the run. The gain is both a reduced complexity and a decrease in computation time.

One such thing is the temperature in the bulk material of the clutches. Among other things this is related to the wear of the clutch material. To not confuse temperature and torque in this section, we use M to denote torque and T for temperature. The heating is due to the friction, whenever at least one of the clutches slip and there is some connection

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = |\Delta\omega_{c_1} M_{c_1}| + |\Delta\omega_{c_2} M_{c_2}| \tag{48}$$

with Q the heat energy in the clutch. The above equation could be used directly to formulate a crude measure of the clutch wear, namely the total amount of dissipated energy.

$$E_c = \int_{t_0}^{t_f} \frac{\mathrm{d}Q}{\mathrm{d}t}\,\mathrm{d}t \tag{49}$$

However, we may use Eq. (48) and consider the heat exchange with the environment via Newton's law of cooling to compute the temperature as a function of time. The heat exchange with the surroundings due to a temperature difference is

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = -q(\mathrm{T} - \mathrm{T}_0) \tag{50}$$

where q is the thermal dissipation rate and T_0 is the temperature of the surrounding. Fourier's equation can be used to supply information on the time evolution of the temperature

$$-\nabla \cdot (k\nabla \mathrm{T}) + \rho c_p \frac{\partial \mathrm{T}}{\partial t} = \frac{\mathrm{d}Q}{\mathrm{d}t} \tag{51}$$

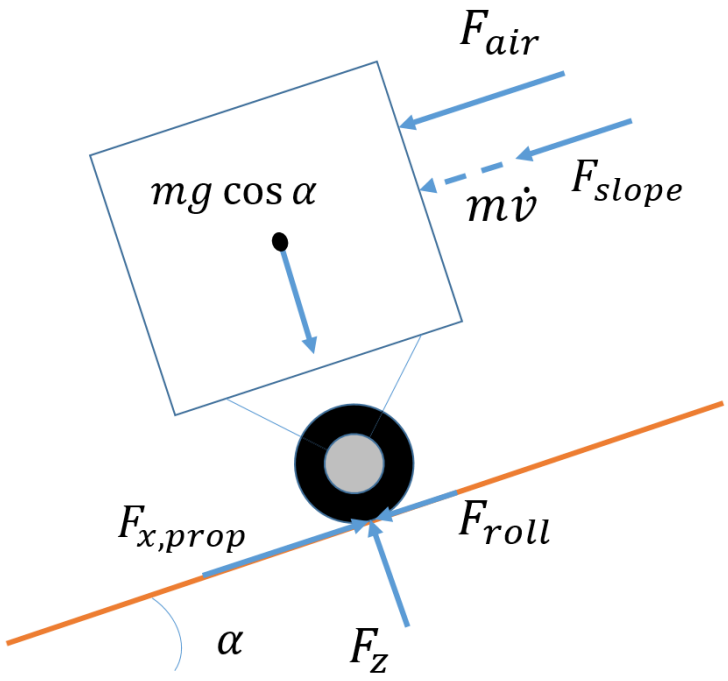

Figure 30: *A free-body diagram of the quarter car (truck) model of the vehicle chassis.*

with k the heat dispersion, ρ the volume density and c_p the specific heat capacity. In this case we will neglect the troublesome first term, by assuming that the heat travels much quicker inside the material than between the interfaces. Then we end up with a so called lumped capacitance model. Note that the heating power in Eq. (48) flows into the system, while that in Eq. (50) flows from the system (into the environment). The differential equation for the temperature becomes

$$\frac{\partial \mathrm{T}}{\partial t} = \frac{1}{\rho c_p}\left(|\Delta\omega_{c_1} M_{c_1}| + |\Delta\omega_{c_2} M_{c_2}| - q(\mathrm{T} - \mathrm{T}_0)\right) \tag{52}$$

One could also keep the terms separate, because the heating due to the clutch slip is typically a much faster process (seconds) than the cooling due to the surrounding (minutes).

Though the temperature could be tracked during the simulation, there is no real point since it doesn't affect the dynamics in the current model. To decrease the simulation time it is instead computed afterwards. One could implement a temperature dependency in the clutch capacity though (or decreased capacity due to wear etc.) and in that case the above should instead be incorporated into the model.

5.3 Chassis

The last part is the chassis that connects the longitudinal resistances and road effects to the propulsion torque. At this level the dynamics is strictly one-dimensional, i.e. neither load transfer, pitch nor heave are considered at the time of writing.

Because of these simplifications, we have used a quarter car model for the vehicle, see Fig. 30. This corresponds to the part called chassis for the total vehicle in Fig. 5. Like before, the cut between the transmission and the chassis is trivial and

$$T_{FD} - T_w = 0 \tag{53}$$
$$\omega_{FD} - \omega_w = 0 \tag{54}$$

The notation here is somewhat inconsistent: T_{FD} and ω_{FD} are entities *after* the final drive, while T_w and ω_w are entities *before* the wheel.

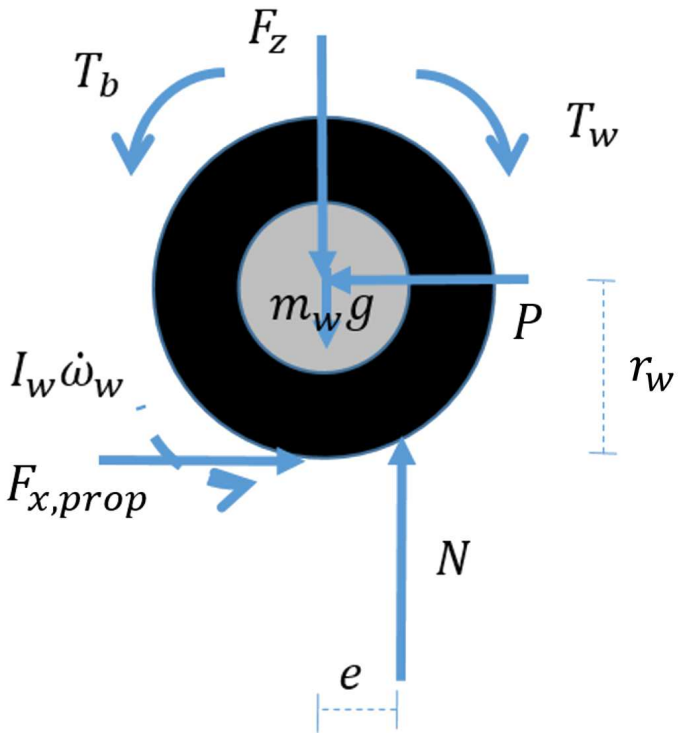

Figure 31: *A free-body diagram of the wheel and tyre on plane ground.*

In this case the tyre has been fused to the wheel and these two together make up a rigid cylinder without slip, see Fig. 31.

The figure implies the equations

$$0 = F_f - F_{x,prop} - m_w \dot{v}_w \tag{55}$$
$$0 = N - F_z - m_w g \tag{56}$$
$$0 = T_w - T_b - r_w F_f - eN - J_w \dot{\omega}_w \tag{57}$$
$$r_w \omega_w = v_w \tag{58}$$
$$v = v_w \tag{59}$$

For the chassis with no wheels (Fig. 32) we can write down the following equations

$$0 = F_{x,prop} - F_{air} - m_s \dot{v} \tag{60}$$
$$0 = F_z - m_s g \tag{61}$$
$$F_{air} = \frac{1}{2} \rho_{air} A C_d v^2 \tag{62}$$

Combining these and rotating the system to account for an inclination angle α gives the following equations for the full chassis in Fig. 30

$$\left(m_s + m_w + \frac{J_w}{r_w^2} \right) \dot{v} = \frac{1}{r_w} (T_w - T_b) - F_{air} - F_{slope} - F_{roll} \tag{63}$$
$$0 = N - (m_s + m_w) g \cos(\alpha) \tag{64}$$
$$F_{roll} = \frac{e}{r} N = f_r (m_s + m_w) g \cos(\alpha) \tag{65}$$
$$F_{slope} = (m_s + m_w) g \sin(\alpha) \tag{66}$$

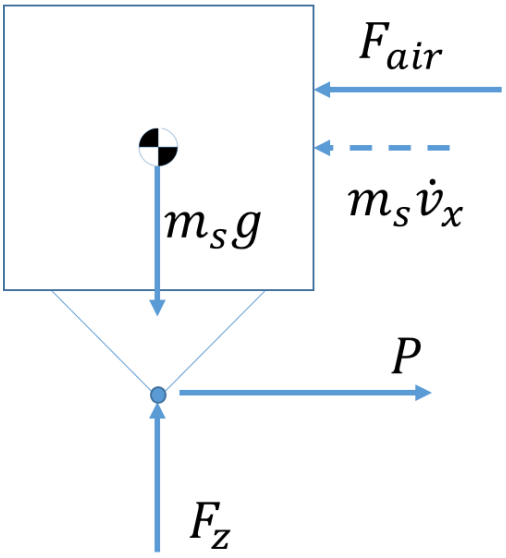

Figure 32: *A free-body diagram of the chassis without tyre on plane ground.*

The term $m_s + m_w$ is of course the total vehicle mass and we therefore write $m = m_s + m_w$. Also note that the wheel inertia comes into effect when combining the system of equations from the wheel and the raw chassis.

Doing this is very useful, as the equation can then be treated in a causal way: which means that we can prescribe a value for T_w and T_b (the input) and compute a linear acceleration $\dot{v}$ (the output). So to close the loop this reduction must be done all the way done to the engine torque (which depends on the accelerator pedal through Eqs. (17)-(24)). Since the clutch model can take three discrete states, there will be one set of equations for each such state.

For the case when clutch one sticks and clutch two slips, the reduced expression takes the form

$$\left[m + \frac{1}{r_w^2}\left(J_w + r_{FD}^2 J_t + r_{FD}^2 r_1^2 J_e\right)\right]\dot{v} = \frac{r_{FD} r_1}{r_w} T_e + \frac{r_{FD}\left(r_2 - r_1\right)}{r_w} T_{c_2} - \left(F_{air} + F_{slope} + F_{roll}\right) \tag{67}$$

The expression for clutch two sticks and clutch one slips is analogous and can be found by the transformation $1 \leftrightarrow 2$. The factor in front of $\dot{v}$ on the left hand side is sometimes called the effective mass and written m^*, γm or km. The two first terms on the right hand side can be thought of as the propulsion force

$$\begin{aligned} m^* &= m + \tfrac{1}{r_w^2}\left(J_w + r_{FD}^2 J_t + r_{FD}^2 r_1^2 J_e\right) \\ F_{x,prop} &= \tfrac{r_{FD} r_1}{r_w} T_e + \tfrac{r_{FD}(r_2 - r_1)}{r_w} T_{c_2} \end{aligned} \qquad c_1 \text{ stick, } c_2 \text{ slip}$$

When the slipping clutch is fully disconnected, it reduces to the familiar expression where the propelling force is the engine torque times the total gear ratio.

In the case where both clutches slip, the engine and chassis (everything after the clutch really: driveline and chassis) decouple and the states evolve independently of each other. The engine follows Eq. (31), while the linear vehicle speed expression reduces to

$$\left[m + \frac{1}{r_w^2}\left(J_w + r_{FD}^2 J_t\right)\right]\dot{v} = \frac{r_{FD}}{r_w}\left(r_1 T_{c_1} + r_2 T_{c_2}\right) - \left(F_{air} + F_{slope} + F_{roll}\right) \tag{68}$$

with the clutch torques given by the first case in Eq. (34). The expression for effective mass and propulsion force are different from the stick-slip system, but can still be easily identified

$$\begin{aligned} m^* &= m + \tfrac{1}{r_w^2}\left(J_w + r_{FD}^2 J_t\right) \\ F_{x,prop} &= r_1 T_{c_1} + r_2 T_{c_2} \end{aligned} \qquad c_1,\ c_2 \text{ slip}$$

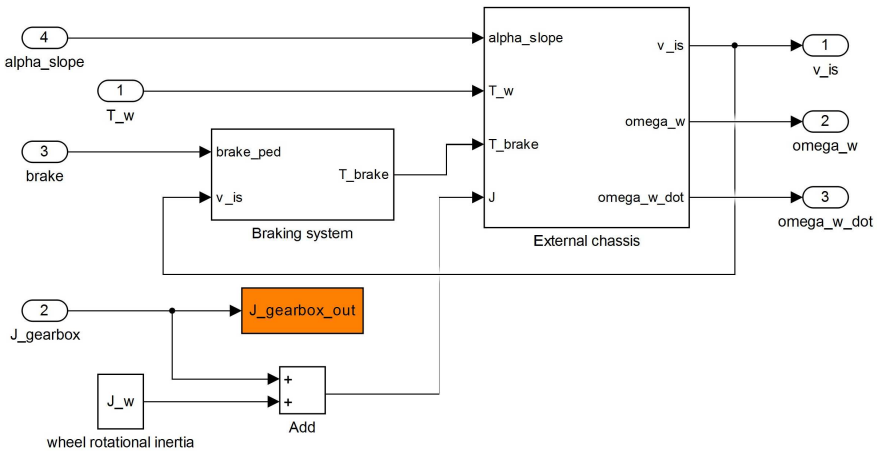

Figure 33: *The top level of the chassis module.*

Therefore a familiar expression can be written for the acceleration, independently of which state the clutch is in

$$m^* \dot{v} = F_{x,prop} - (F_{air} + F_{slope} + F_{roll}) \tag{69}$$

However, the engine speed ω_e must still be handled in the slip-slip case. The implementation of the chassis can be seen in Figs. 33 and 34. The implemented equation is the one given by Eq. (69), and so this is the reason why the inertias have been propagated from the other modules (see Figs. 4, 11 and 16).

The brake system in figure 33 is nothing sophisticated, a linear function of the brake pedal position

$$T_b = T_{b,max} b_p \cdot \tanh(v) \tag{70}$$

where $T_{b,max}$ is a constant representing the maximum torque that the brake system can deliver. The factor $\tanh(v)$ is included to get the torque direction correct: the resulting brake force should always counter the direction of motion, and to remove the brake torque at standstill.

6 Discussion and conclusion

Hopefully it can be understood from the previous sections that this model is very much a work in progress. The research in the OCEAN-project (see [11]) relates to the operating cycle module, and at the time of writing this is the least sophisticated model in VehProp. As this model becomes more and more demanding, the vehicle model (and driver too, for that matter) must be further developed to be able to account for the effects the road, environment and missions has. Some things that we already foresee will need to be implemented are

- Dry friction model for the brakes (needed for proper standstill and start-stop).
- Reversing (important for proper mission description).
- Standstill (essential for realistic driving scenarios).
- Spring and damper model for the suspension (to account for road roughness)
- Driver look-ahead (to hit the correct stop positions).
- Tyre (important all over really, but especially for starts, to account for different surfaces, and also has a small effect on fuel consumption).

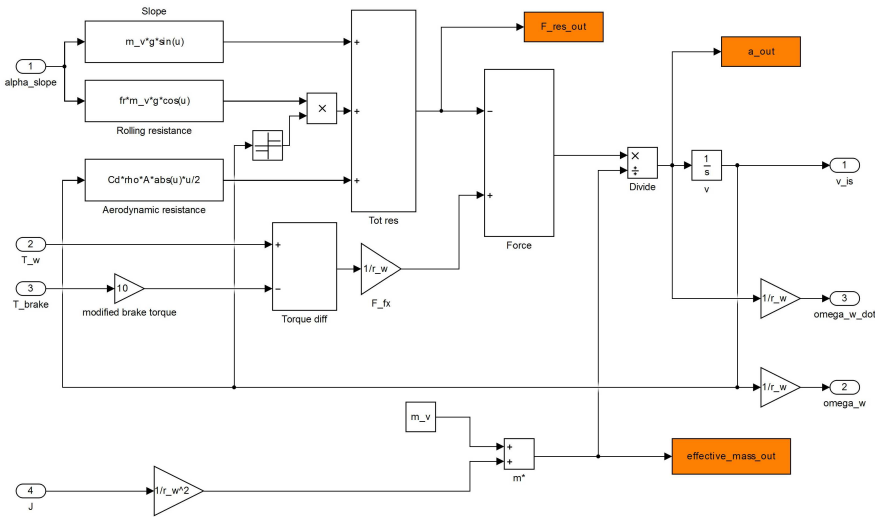

Figure 34: *The external chassis module.*

The main purpose of the model is to accurately predict fuel consumption, but there are a number of secondary properties that can be predicted or estimated at the same time.

The clutch temperature is one of them and it can be related to wear, which can be taken as another evaluation factor.

The time to finish a specific mission is another evaluation factor. This is probably most effective when comparing different vehicle configurations, e.g. the difference between a single or a dual clutch transmission. The evaluation factor measures transport productivity: finishing a mission quicker allows for more missions per time unit.

References

[1] P. Pettersson, S. Berglund, B. Jacobson, L. Fast, P. Johannesson, and F. Santandrea. "A proposal for an operating cycle description format for road transport missions". In: *European Transport Research Review* 10.31 (June 2018), pp. 1–19. DOI: 10.1186/s12544-018-0298-4.

[2] P. Pettersson. "On Numerical Descriptions of Road Transport Missions". Licentiate thesis. Göteborg, Sweden: Chalmers University of Technology, June 2017. URL: https://research.chalmers.se/publication/249525 (visited on 08/06/2018).

[3] P. Pettersson, P. Johannesson, B. Jacobson, F. Bruzelius, L. Fast, and S. Berglund. "A statistical operating cycle description for prediction of road vehicles' energy consumption". In: *Transportation Research Part D: Transport and Environment* 73 (Aug. 2019), pp. 205–229. DOI: 10.1016/j.trd.2019.07.006.

[4] B. Jacobson. *Final report from the project: Modular Simulation Tool for Vehicle Propulsion concerning Energy Consumption and Emissions*. Tech. rep. Chalmers University of Technology, Göteborg: Department of Machine and Vehicle Design, 1997.

[5] A. Eriksson and B. Jacobson. "Modular modelling and simulation tool for evaluation of powertrain performance". In: *Int. J. Vehicle Design* 21 (1999), pp. 175–189.

[6] S. Venbrant. "Towards a simulation environment for Operating Cycle Analysis of Road Transports". MA thesis. Göteborg, Sweden: Chalmers University of Technology, 2015.

[7] S. Berglund. "Modeling Complex Engines as Dynamic Powertrain Members". PhD thesis. Chalmers University of Technology, 1999.

[8] B. Jacobson. "Gear Shifting with Retained Power Transfer". PhD thesis. Chalmers University of Technology, 1993.

[9] I. Newton. *PhilosophiæNaturalis Principia Mathematica*. 3rd ed. Vol. 3. 1726.

[10] B. Jacobson. *Vehicle dynamics*. Chalmers University of Technology, 2016.

[11] *OCEAN project page*. Available at https://www.chalmers.se/en/projects/Pages/Operating-Cycle-Energy-Management.aspx. Nov. 2016.