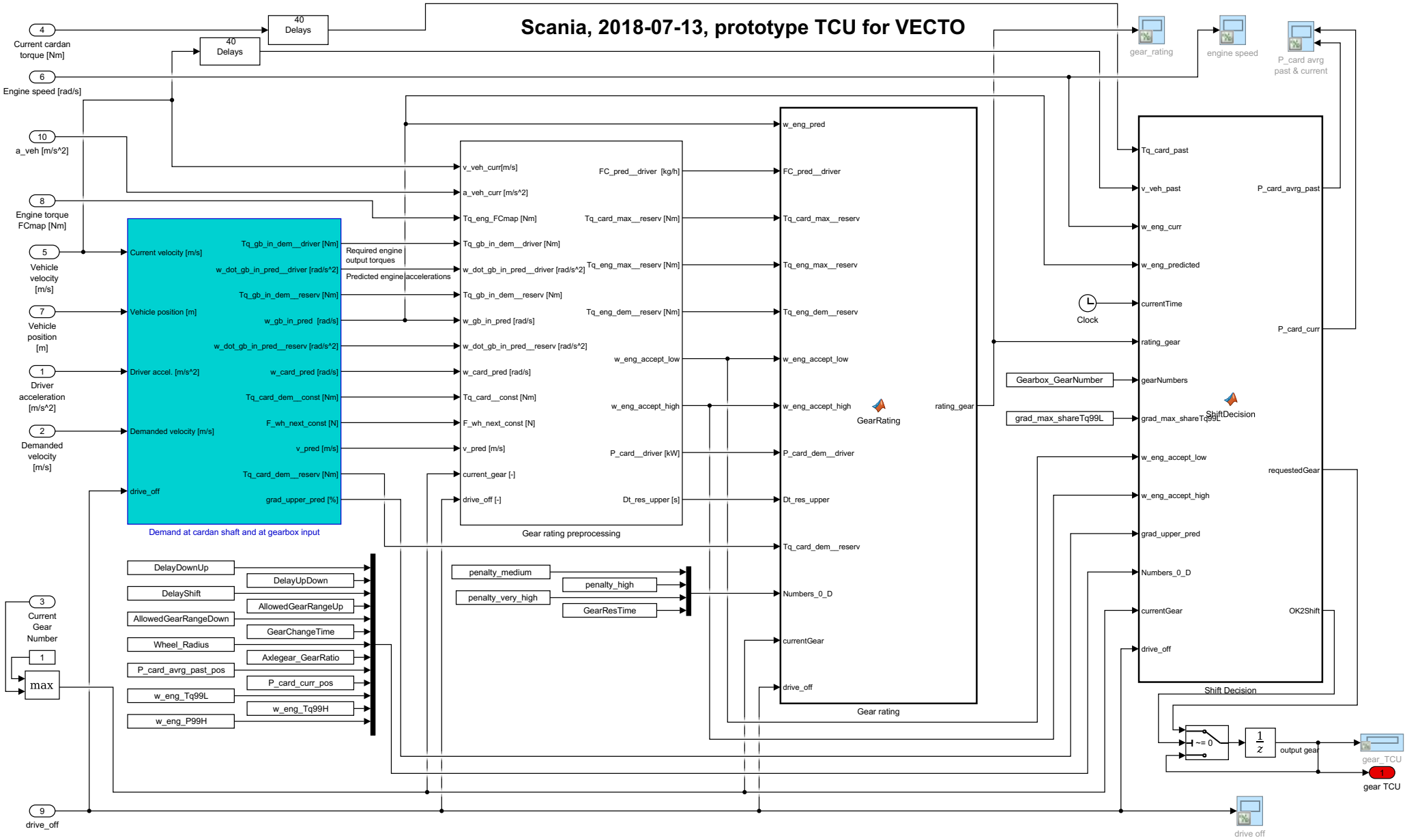
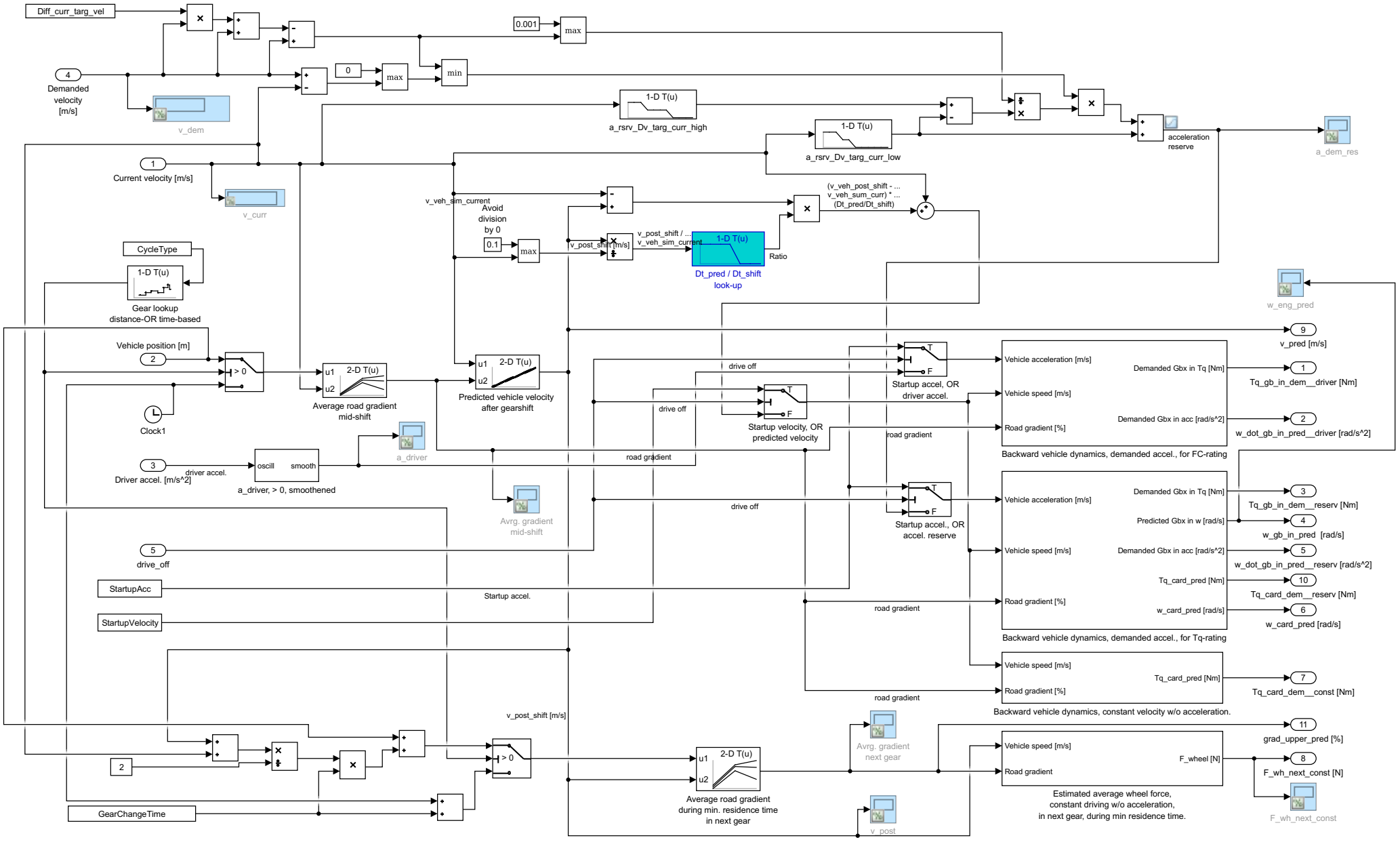
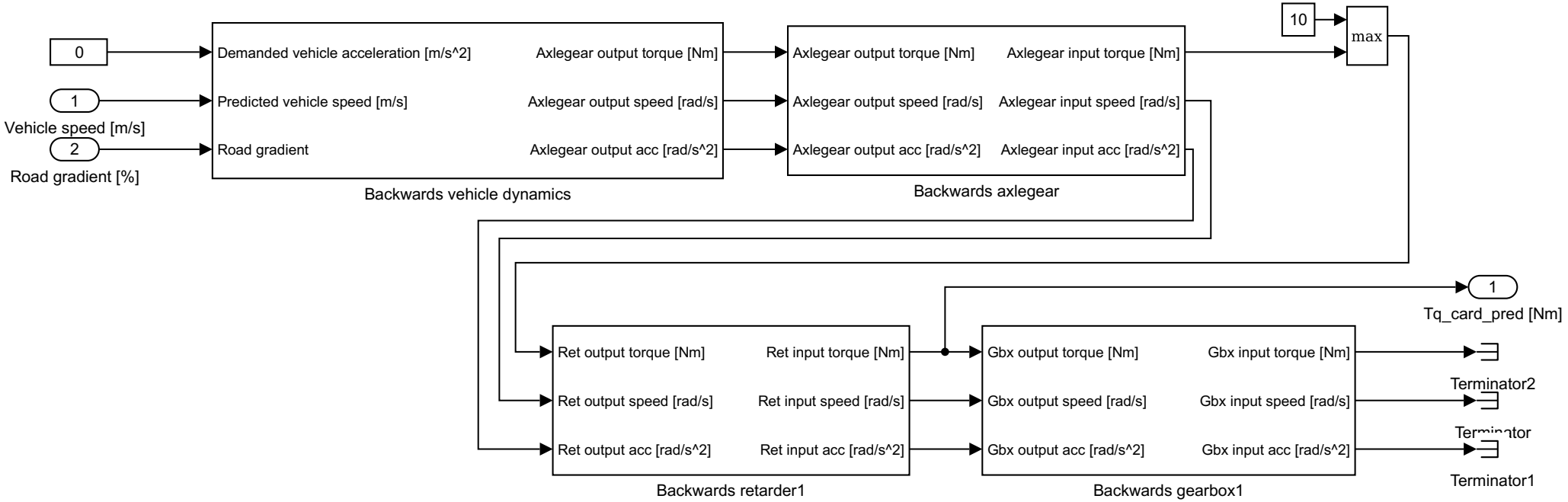
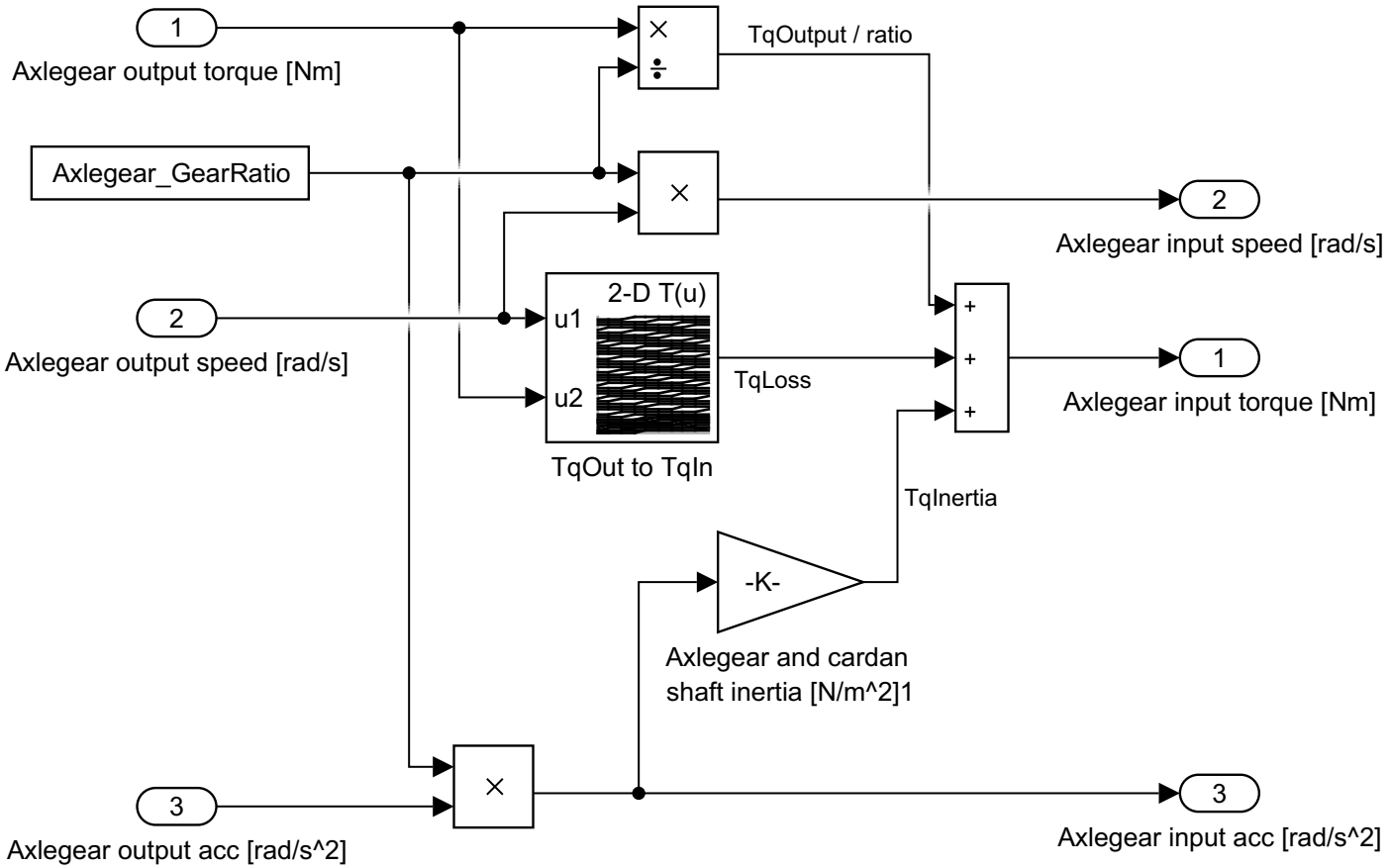
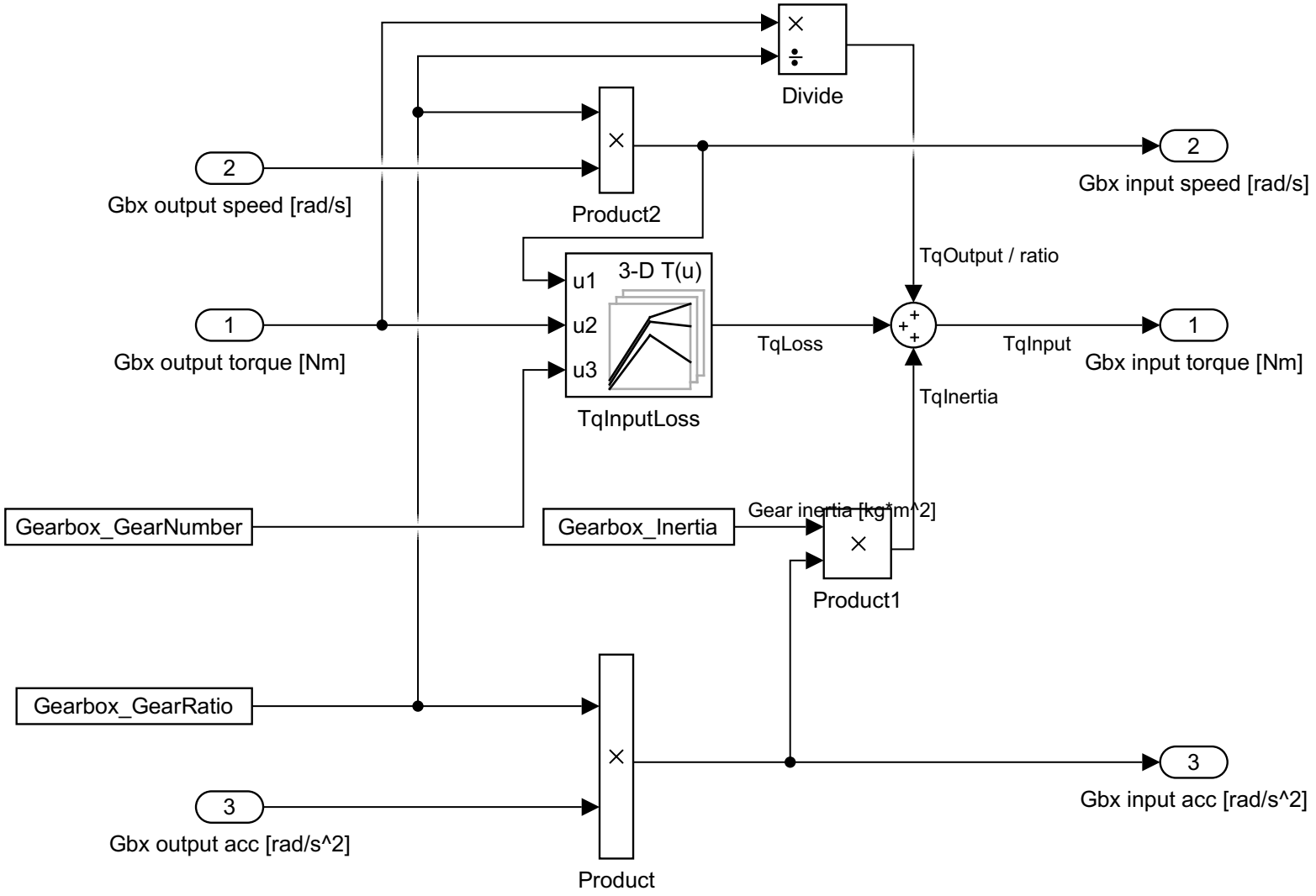

Scania, 2018-07-13, prototype TCU for VECTO
4
Current cardan torque [Nm]
40 Delays
40 Delays
6
Engine speed [rad/s]
10
a_veh [m/s^2]
8
Engine torque FCmap [Nm]
5
Vehicle velocity [m/s]
7
Vehicle position [m]
1
Driver acceleration [m/s^2]
2
Demanded velocity [m/s]
Current velocity [m/s]
Vehicle position [m]
Driver accel. [m/s^2]
Demanded velocity [m/s]
drive_off
Tq_gb_in_dem__driver [Nm]
w_dot_gb_in_pred__driver [rad/s^2]
Tq_gb_in_dem__reserv [Nm]
w_gb_in_pred [rad/s]
w_dot_gb_in_pred__reserv [rad/s^2]
w_card_pred [rad/s]
Tq_card_dem__const [Nm]
F_wh_next_const [N]
v_pred [m/s]
Tq_card_dem__reserv [Nm]
grad_upper_pred [%]
Demand at cardan shaft and at gearbox input
Required engine output torques
Predicted engine accelerations
v_veh_curr[m/s]
a_veh_curr [m/s^2]
Tq_eng_FCmap [Nm]
Tq_gb_in_dem__driver [Nm]
w_dot_gb_in_pred__driver [rad/s^2]
Tq_gb_in_dem__reserv [Nm]
w_gb_in_pred [rad/s]
w_dot_gb_in_pred__reserv [rad/s^2]
w_card_pred [rad/s]
Tq_card__const [Nm]
F_wh_next_const [N]
v_pred [m/s]
current_gear [-]
drive_off [-]
FC_pred__driver [kg/h]
Tq_card_max__reserv [Nm]
Tq_eng_max__reserv [Nm]
Tq_eng_dem__reserv [Nm]
w_eng_accept_low
w_eng_accept_high
P_card__driver [kW]
Dt_res_upper [s]
Gear rating preprocessing
w_eng_pred
FC_pred__driver
Tq_card_max__reserv
Tq_eng_max__reserv
Tq_eng_dem__reserv
w_eng_accept_low
w_eng_accept_high
P_card_dem__driver
Dt_res_upper
Tq_card_dem__reserv
Numbers_0_D
currentGear
drive_off
rating_gear
GearRating
Gear rating
DelayDownUp
DelayShift
AllowedGearRangeDown
Wheel_Radius
P_card_avrg_past_pos
w_eng_Tq99L
w_eng_P99H
DelayUpDown
AllowedGearRangeUp
GearChangeTime
Axlegear_GearRatio
P_card_curr_pos
w_eng_Tq99H
penalty_medium
penalty_very_high
penalty_high
GearResTime
3
Current Gear Number
1
max
9
drive_off
gear_rating
engine speed
P_card avrg past & current
Tq_card_past
v_veh_past
w_eng_curr
w_eng_predicted
currentTime
rating_gear
gearNumbers
grad_max_shareTq99L
w_eng_accept_low
w_eng_accept_high
grad_upper_pred
Numbers_0_D
currentGear
drive_off
P_card_avrg_past
P_card_curr
requestedGear
OK2Shift
Clock
Gearbox_GearNumber
grad_max_shareTq99L
ShiftDecision
Shift Decision
~= 0
1
z
output gear
gear_TCU
1
gear TCU
drive off


Diff_curr_targ_vel
4
Demanded velocity [m/s]
v_dem
0.001
max
0
max
min
1-D T(u)
a_rsrv_Dv_targ_curr_high
1-D T(u)
a_rsrv_Dv_targ_curr_low
acceleration reserve
a_dem_res
1
Current velocity [m/s]
v_curr
v_veh_sim_current
Avoid division by 0
0.1
max
v_post_shift [m/s]
v_post_shift / ... v_veh_sim_current
1-D T(u)
Dt_pred / Dt_shift look-up
Ratio
(v_veh_post_shift - ... v_veh_sum_curr) * ... (Dt_pred/Dt_shift)
CycleType
1-D T(u)
Gear lookup distance-OR time-based
w_eng_pred
Vehicle position [m]
2
> 0
Clock1
u1 2-D T(u) u2
Average road gradient mid-shift
u1 2-D T(u) u2
Predicted vehicle velocity after gearshift
drive off
T F
Startup accel, OR driver accel.
drive off
T F
Startup velocity, OR predicted velocity
road gradient
Vehicle acceleration [m/s]
Vehicle speed [m/s]
Road gradient [%]
Demanded Gbx in Tq [Nm]
Demanded Gbx in acc [rad/s^2]
Backward vehicle dynamics, demanded accel., for FC-rating
9
v_pred [m/s]
1
Tq_gb_in_dem__driver [Nm]
2
w_dot_gb_in_pred__driver [rad/s^2]
3
Driver accel. [m/s^2]
driver accel.
oscill smooth
a_driver, > 0, smoothened
a_driver
road gradient
Avrg. gradient mid-shift
drive off
T F
Startup accel., OR accel. reserve
Vehicle acceleration [m/s]
Vehicle speed [m/s]
Road gradient [%]
Demanded Gbx in Tq [Nm]
Predicted Gbx in w [rad/s]
Demanded Gbx in acc [rad/s^2]
Tq_card_pred [Nm]
w_card_pred [rad/s]
Backward vehicle dynamics, demanded accel., for Tq-rating
3
Tq_gb_in_dem__reserv [Nm]
4
w_gb_in_pred [rad/s]
5
w_dot_gb_in_pred__reserv [rad/s^2]
10
Tq_card_dem__reserv [Nm]
6
w_card_pred [rad/s]
5
drive_off
StartupAcc
Startup accel.
StartupVelocity
road gradient
Vehicle speed [m/s]
Road gradient [%]
Tq_card_pred [Nm]
Backward vehicle dynamics, constant velocity w/o acceleration.
7
Tq_card_dem__const [Nm]
road gradient
v_post_shift [m/s]
2
> 0
u1 2-D T(u) u2
Average road gradient during min. residence time in next gear
Avrg. gradient next gear
11
grad_upper_pred [%]
Vehicle speed [m/s]
Road gradient
F_wheel [N]
Estimated average wheel force, constant driving w/o acceleration, in next gear, during min residence time.
8
F_wh_next_const [N]
F_wh_next_const
GearChangeTime
v_post
Startup accel.

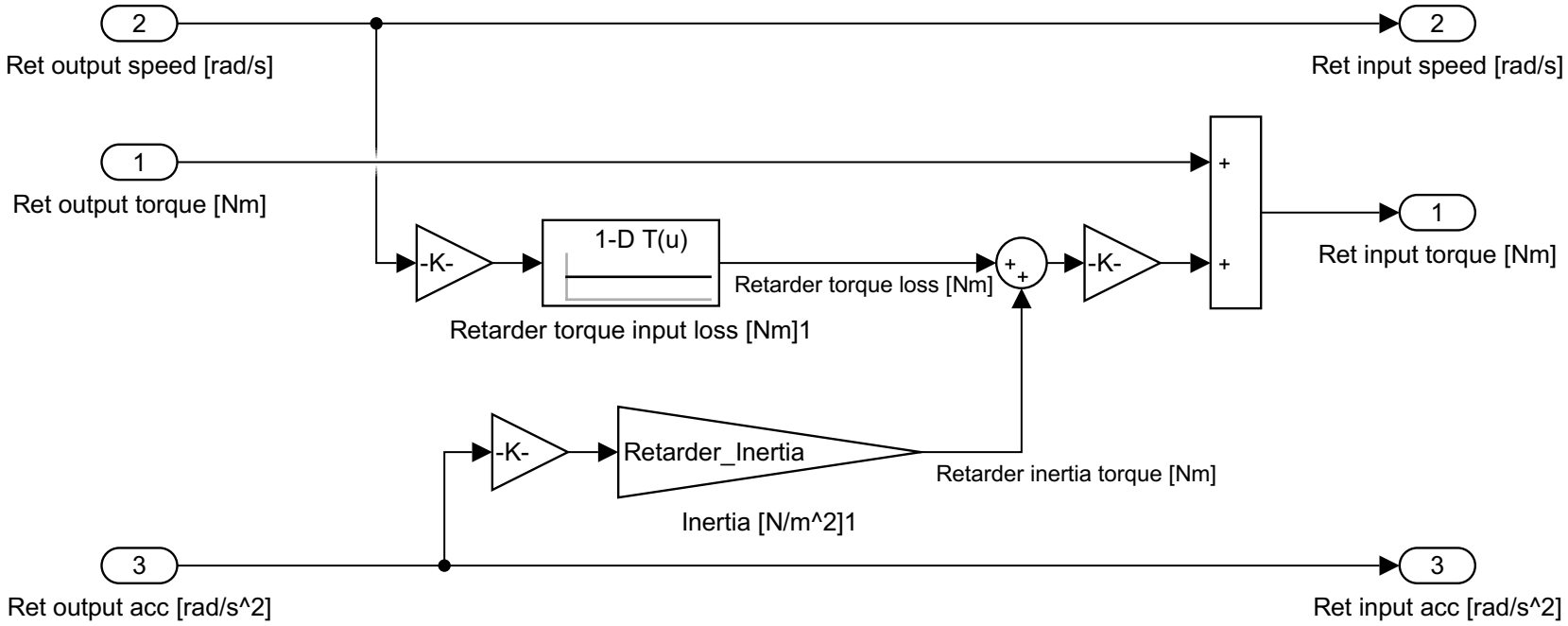

0
1
Vehicle speed [m/s]
2
Road gradient [%]
Demanded vehicle acceleration [m/s^2]
Predicted vehicle speed [m/s]
Road gradient
Axlegear output torque [Nm]
Axlegear output speed [rad/s]
Axlegear output acc [rad/s^2]
Backwards vehicle dynamics
Axlegear output torque [Nm]
Axlegear output speed [rad/s]
Axlegear output acc [rad/s^2]
Axlegear input torque [Nm]
Axlegear input speed [rad/s]
Axlegear input acc [rad/s^2]
Backwards axlegear
10
max
1
Tq_card_pred [Nm]
Ret output torque [Nm]
Ret output speed [rad/s]
Ret output acc [rad/s^2]
Ret input torque [Nm]
Ret input speed [rad/s]
Ret input acc [rad/s^2]
Backwards retarder1
Gbx output torque [Nm]
Gbx output speed [rad/s]
Gbx output acc [rad/s^2]
Gbx input torque [Nm]
Gbx input speed [rad/s]
Gbx input acc [rad/s^2]
Backwards gearbox1
Terminator2
Terminator1

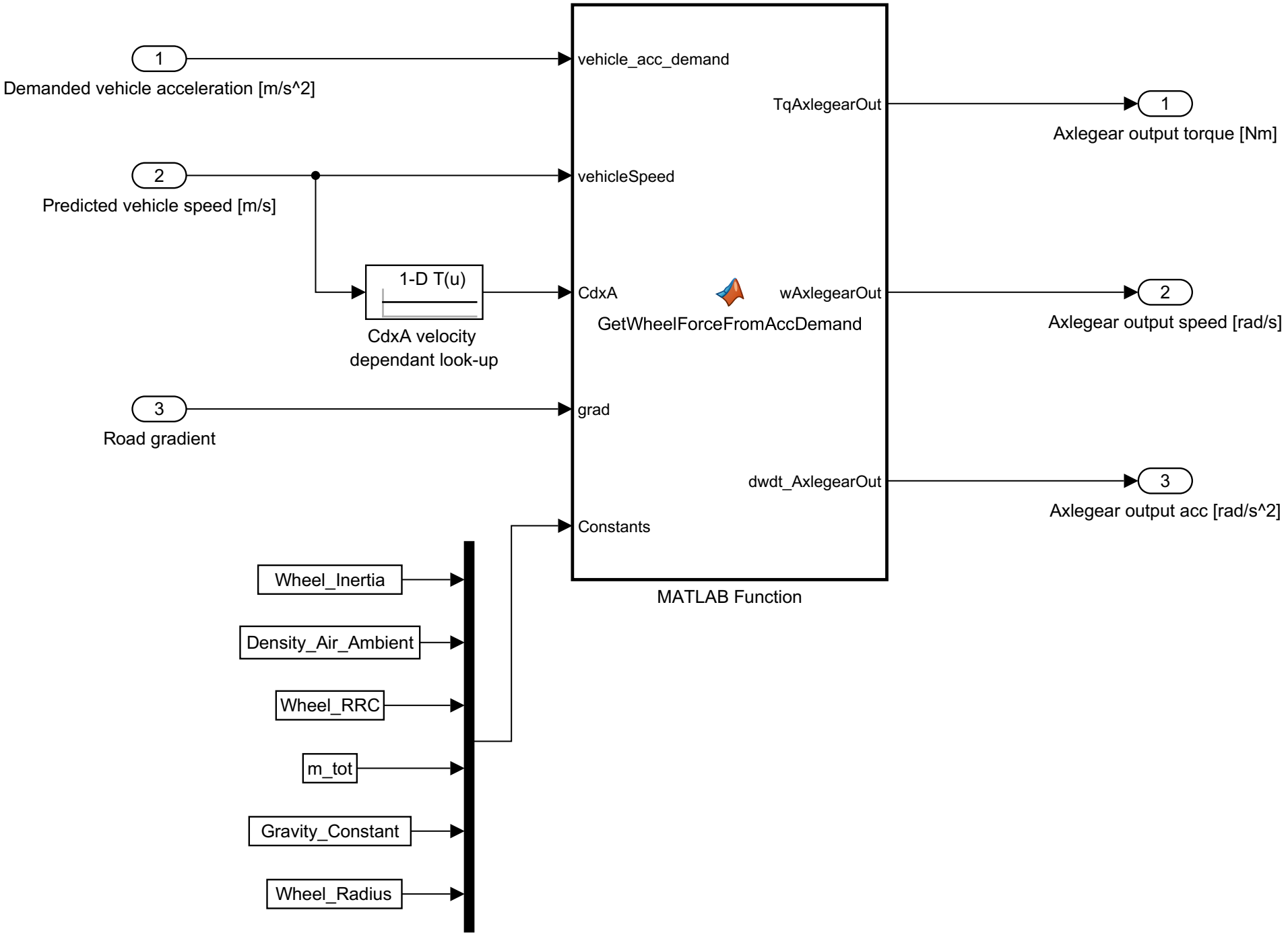
TqInput	= TqOutout / ratio + TqInputLoss + TqInertia	[Nm]
w_Input	= w_Output * ratio	[rad/s]
dwdt_Input	= dwdt_Output * ratio	[rad/s^2]



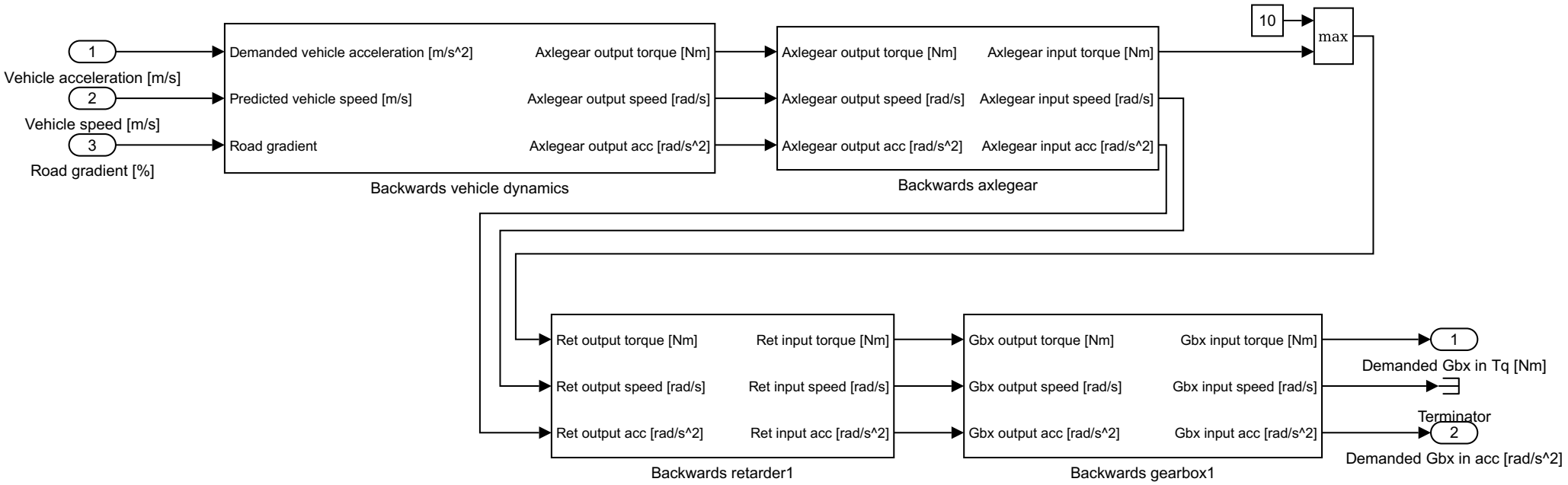

Divide
2
Gbx output speed [rad/s]
Product2
2
Gbx input speed [rad/s]
TqOutput / ratio
1
Gbx output torque [Nm]
u1
u2
u3
3-D T(u)
TqInputLoss
TqLoss
TqInput
1
Gbx input torque [Nm]
TqInertia
Gearbox_GearNumber
Gearbox_Inertia
Gear inertia [kg*m^2]
Product1
Gearbox_GearRatio
3
Gbx output acc [rad/s^2]
Product
3
Gbx input acc [rad/s^2]

TqInput	= TqOutput + ratio*(TqInertia + TqLoss)	[Nm]
w_Input	= w_Output	[rad/s]
dwdt_Input	= dwdt_Output	[rad/s^"]

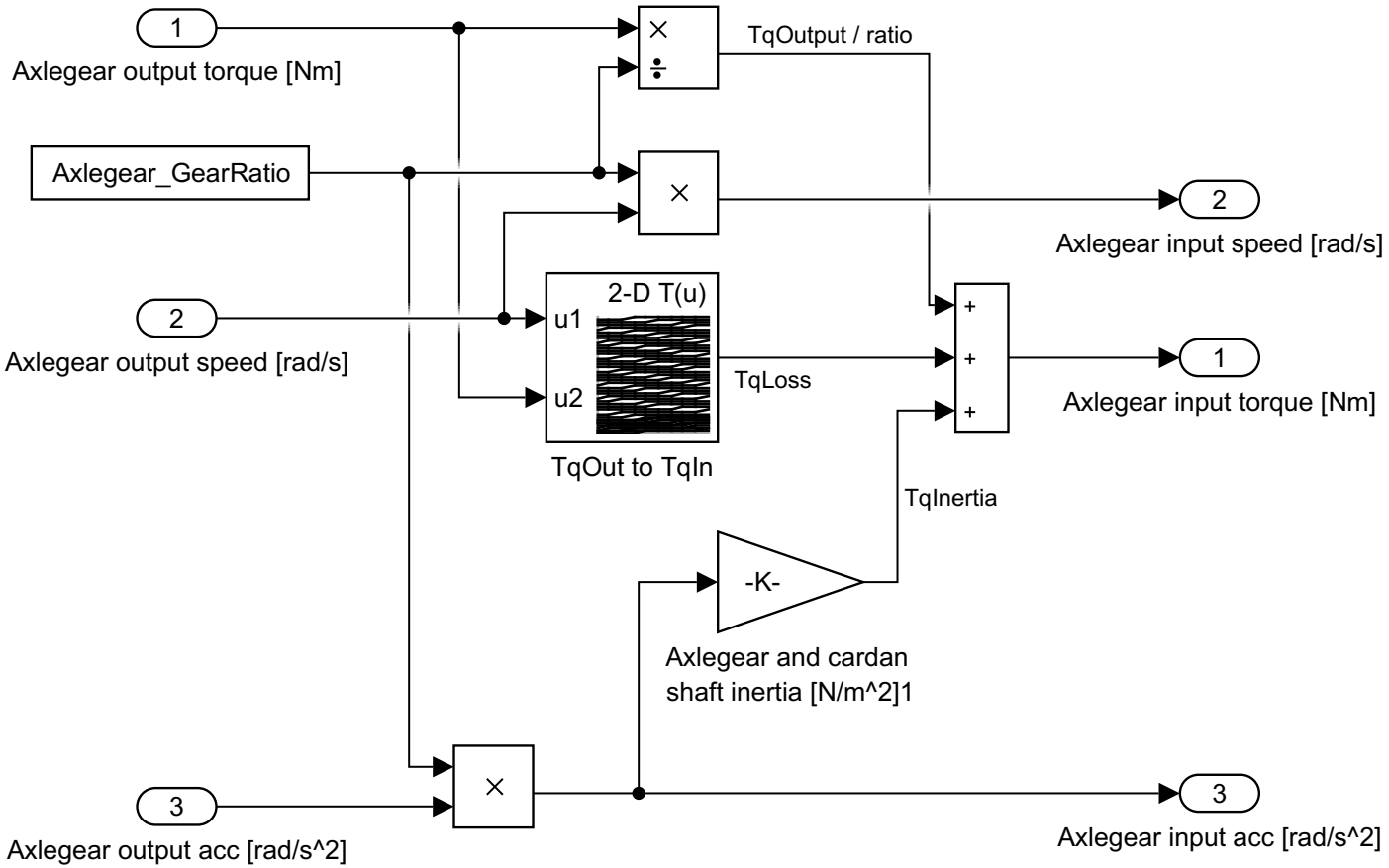


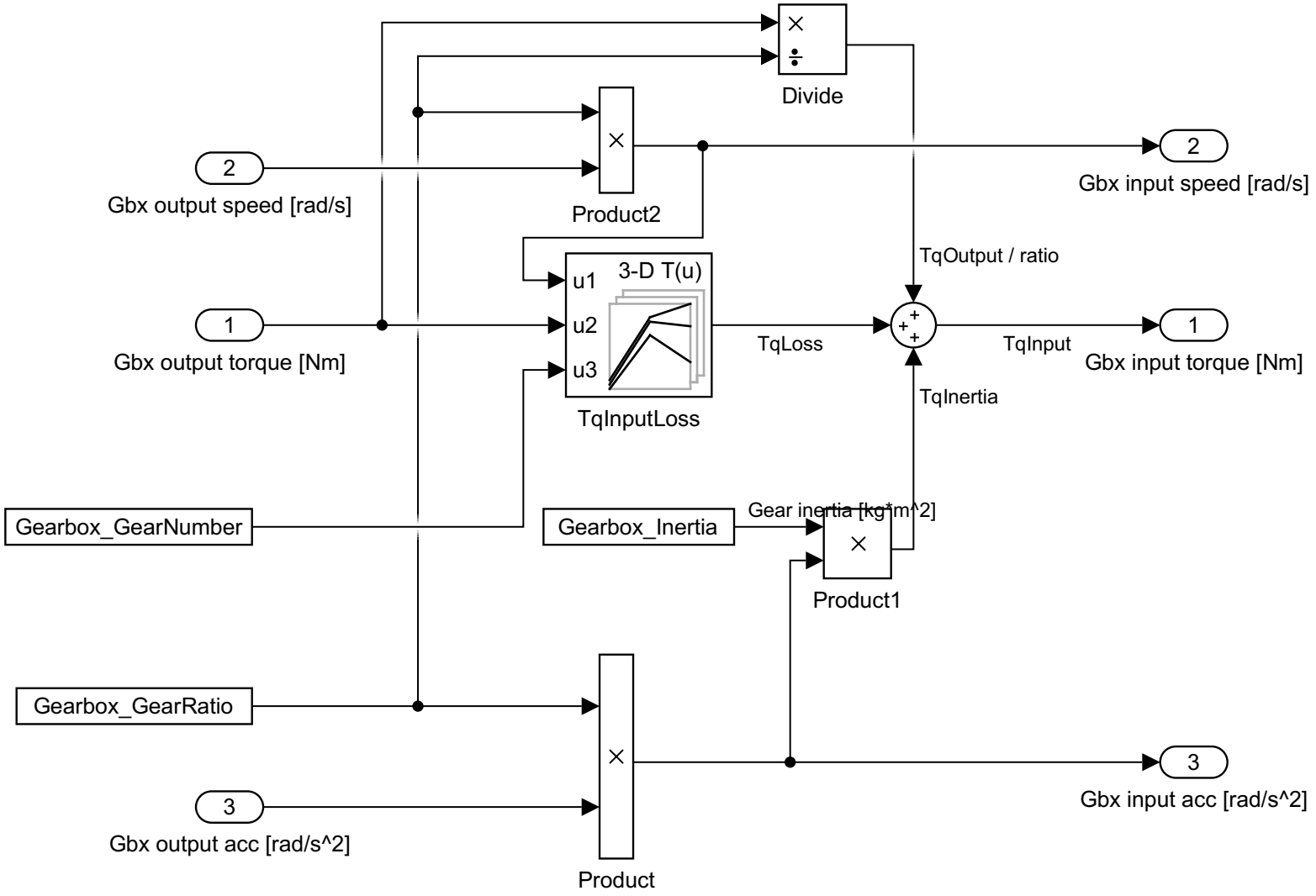

1
Demanded vehicle acceleration [m/s^2]
2
Predicted vehicle speed [m/s]
1-D T(u)
CdxA velocity
dependant look-up
3
Road gradient
vehicle_acc_demand
vehicleSpeed
CdxA
grad
Constants
TqAxlegearOut
wAxlegearOut
dwdt_AxlegearOut
GetWheelForceFromAccDemand
MATLAB Function
1
Axlegear output torque [Nm]
2
Axlegear output speed [rad/s]
3
Axlegear output acc [rad/s^2]
Wheel_Inertia
Density_Air_Ambient
Wheel_RRC
m_tot
Gravity_Constant
Wheel_Radius

```
function [TqAxlegearOut, wAxlegearOut, dwdt_AxlegearOut]  ...
    = GetWheelForceFromAccDemand(...
    vehicle_acc_demand, vehicleSpeed, CdxA, grad, Constants)

%CdxA =  Air drag resistance factor
wheelInertia = Constants(1); % [kg*m^2]
p_air   = Constants(2); % Air density
RCC_i   = Constants(3); % Rolling resistance
m_tot   = Constants(4); % [kg] total vehicle mass
g       = Constants(5); % [m/s^2] gravity contant
r_dyn   = Constants(6); % [m] Dynamic wheel rolling radius


F_air       = CdxA * p_air / 2 * vehicleSpeed^2;
F_roll      = RCC_i * cos(atan( grad/100 )) * m_tot * g;
F_grad      = sin(atan( grad/100 )) * m_tot * g;


wheel_Force = vehicle_acc_demand * m_tot + F_air + F_roll + F_grad; %[Nm]


wAxlegearOut        = vehicleSpeed / r_dyn; %[rad/s]
dwdt_AxlegearOut    = vehicle_acc_demand / r_dyn; %[rad/s^2]
TqAxlegearOut       = wheel_Force *r_dyn + wheelInertia * dwdt_AxlegearOut; % Nm]
```

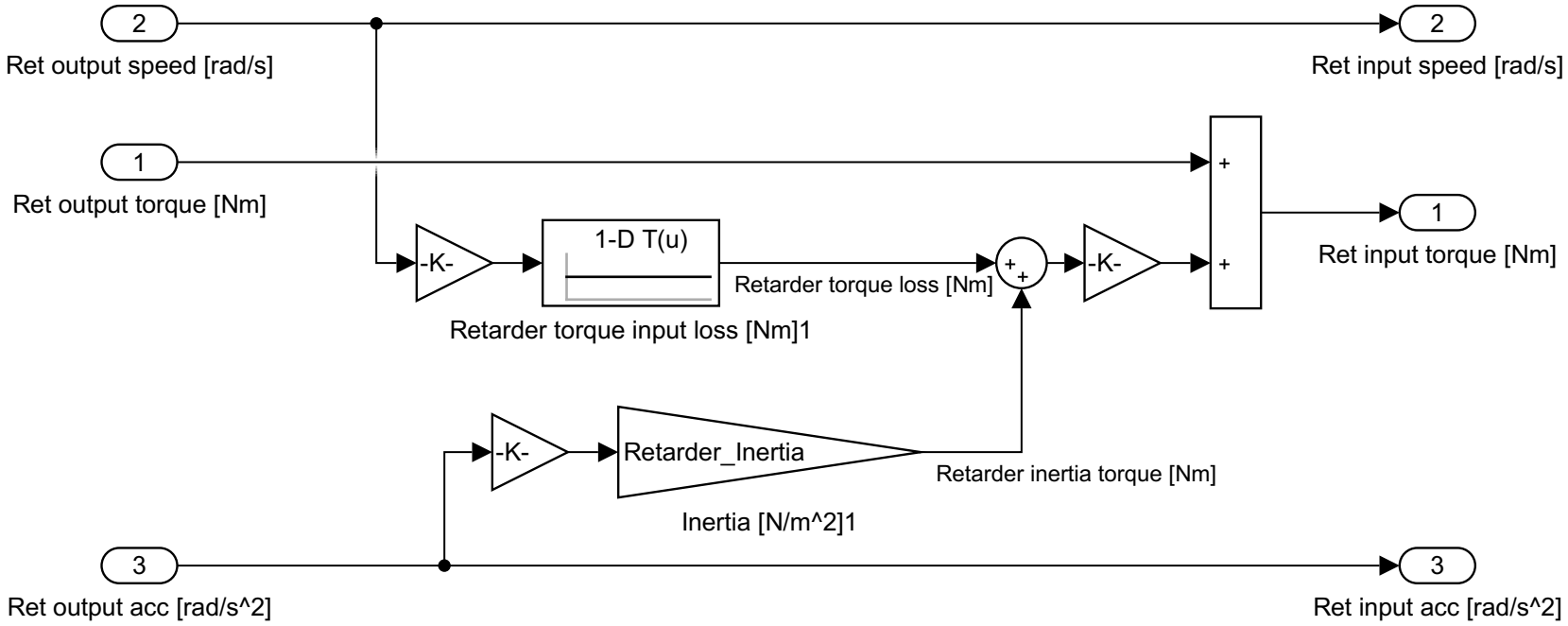

1
Vehicle acceleration [m/s]
2
Vehicle speed [m/s]
3
Road gradient [%]
Demanded vehicle acceleration [m/s^2]
Predicted vehicle speed [m/s]
Road gradient
Axlegear output torque [Nm]
Axlegear output speed [rad/s]
Axlegear output acc [rad/s^2]
Backwards vehicle dynamics
Axlegear output torque [Nm]
Axlegear output speed [rad/s]
Axlegear output acc [rad/s^2]
Axlegear input torque [Nm]
Axlegear input speed [rad/s]
Axlegear input acc [rad/s^2]
Backwards axlegear
10
max
Ret output torque [Nm]
Ret output speed [rad/s]
Ret output acc [rad/s^2]
Ret input torque [Nm]
Ret input speed [rad/s]
Ret input acc [rad/s^2]
Backwards retarder1
Gbx output torque [Nm]
Gbx output speed [rad/s]
Gbx output acc [rad/s^2]
Gbx input torque [Nm]
Gbx input speed [rad/s]
Gbx input acc [rad/s^2]
Backwards gearbox1
1
Demanded Gbx in Tq [Nm]
Terminator
2
Demanded Gbx in acc [rad/s^2]

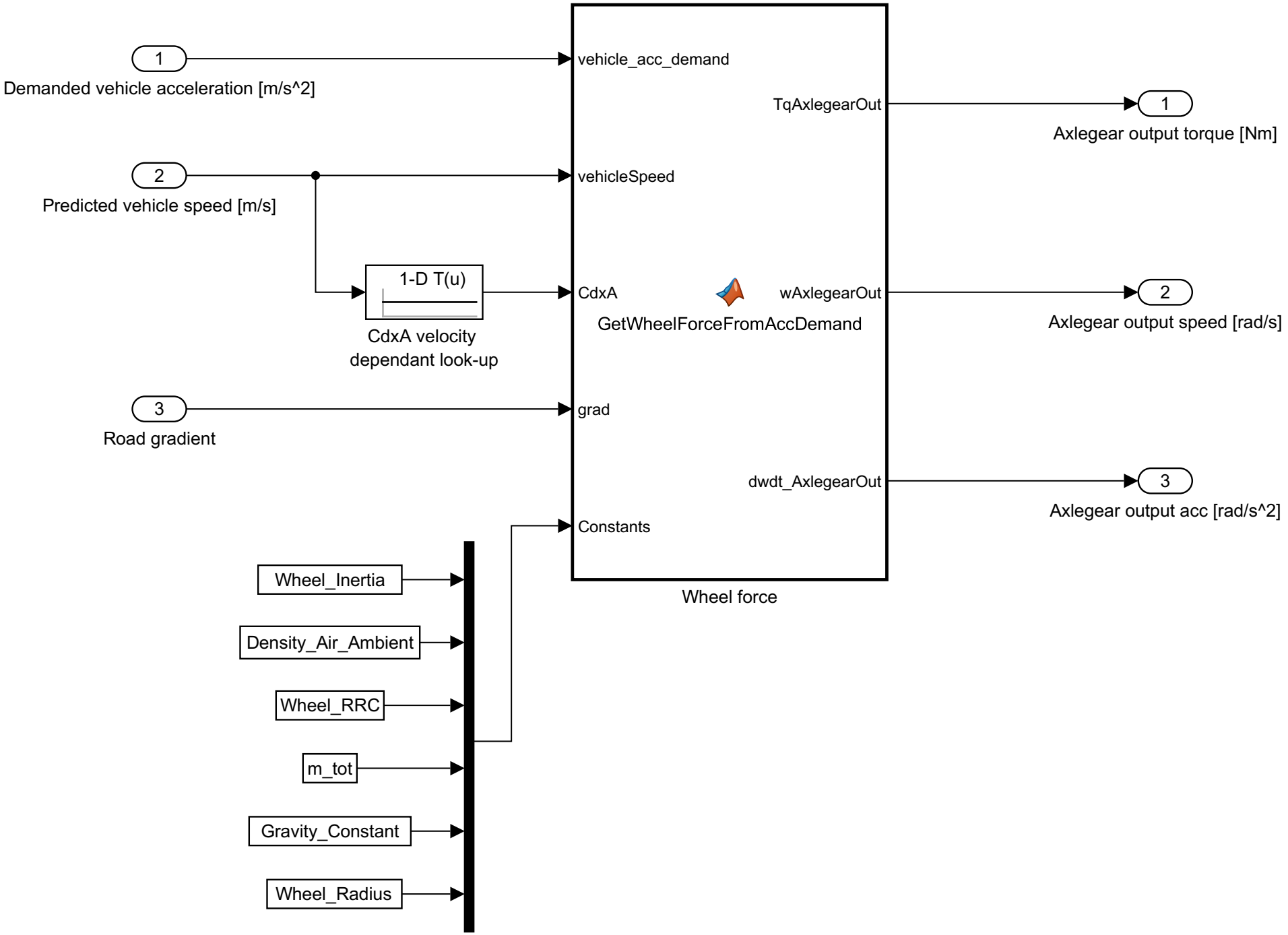
TqInput	= TqOutout / ratio + TqInputLoss + TqInertia	[Nm]
w_Input	= w_Output * ratio	[rad/s]
dwdt_Input	= dwdt_Output * ratio	[rad/s^2]



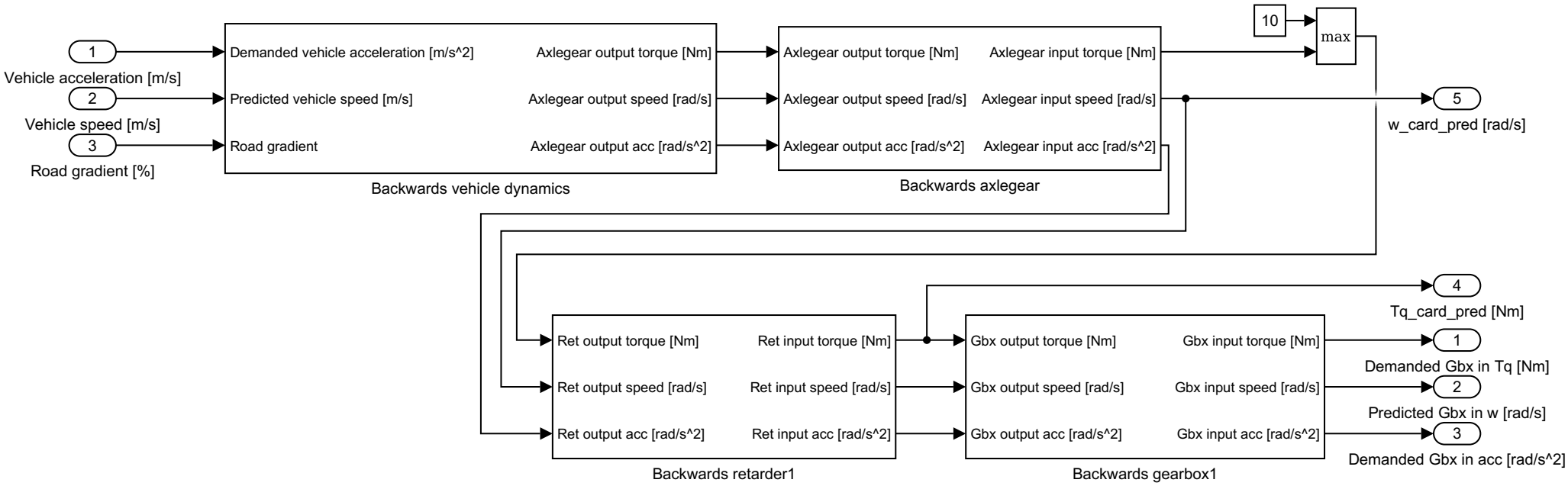

Divide
2
Gbx output speed [rad/s]
Product2
2
Gbx input speed [rad/s]
TqOutput / ratio
1
Gbx output torque [Nm]
u1
u2
u3
3-D T(u)
TqInputLoss
TqLoss
TqInput
1
Gbx input torque [Nm]
TqInertia
Gearbox_GearNumber
Gearbox_Inertia
Gear inertia [kg*m^2]
Product1
Gearbox_GearRatio
3
Gbx output acc [rad/s^2]
Product
3
Gbx input acc [rad/s^2]

TqInput	= TqOutput + ratio*(TqInertia + TqLoss)	[Nm]
w_Input	= w_Output	[rad/s]
dwdt_Input	= dwdt_Output	[rad/s^"]

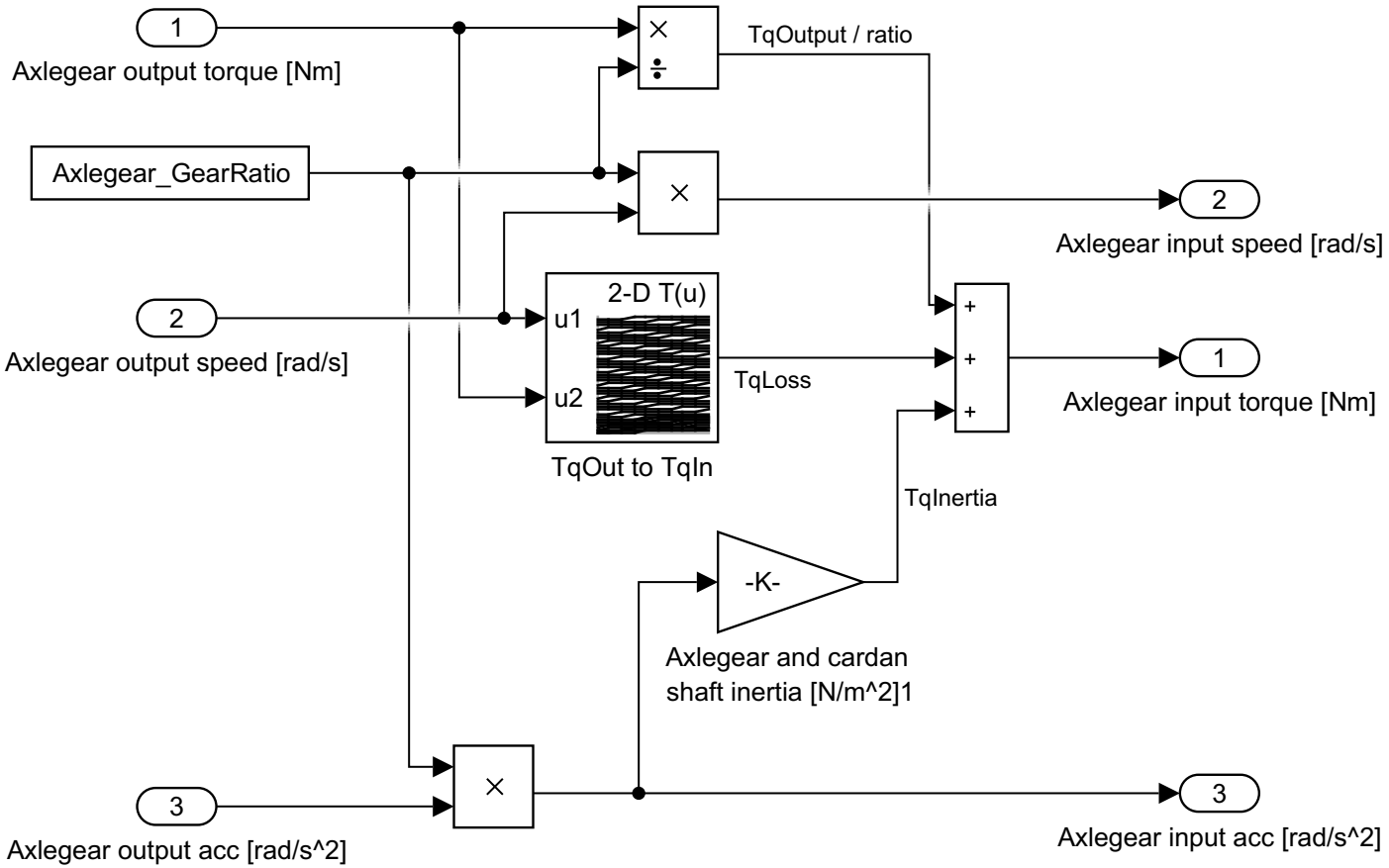


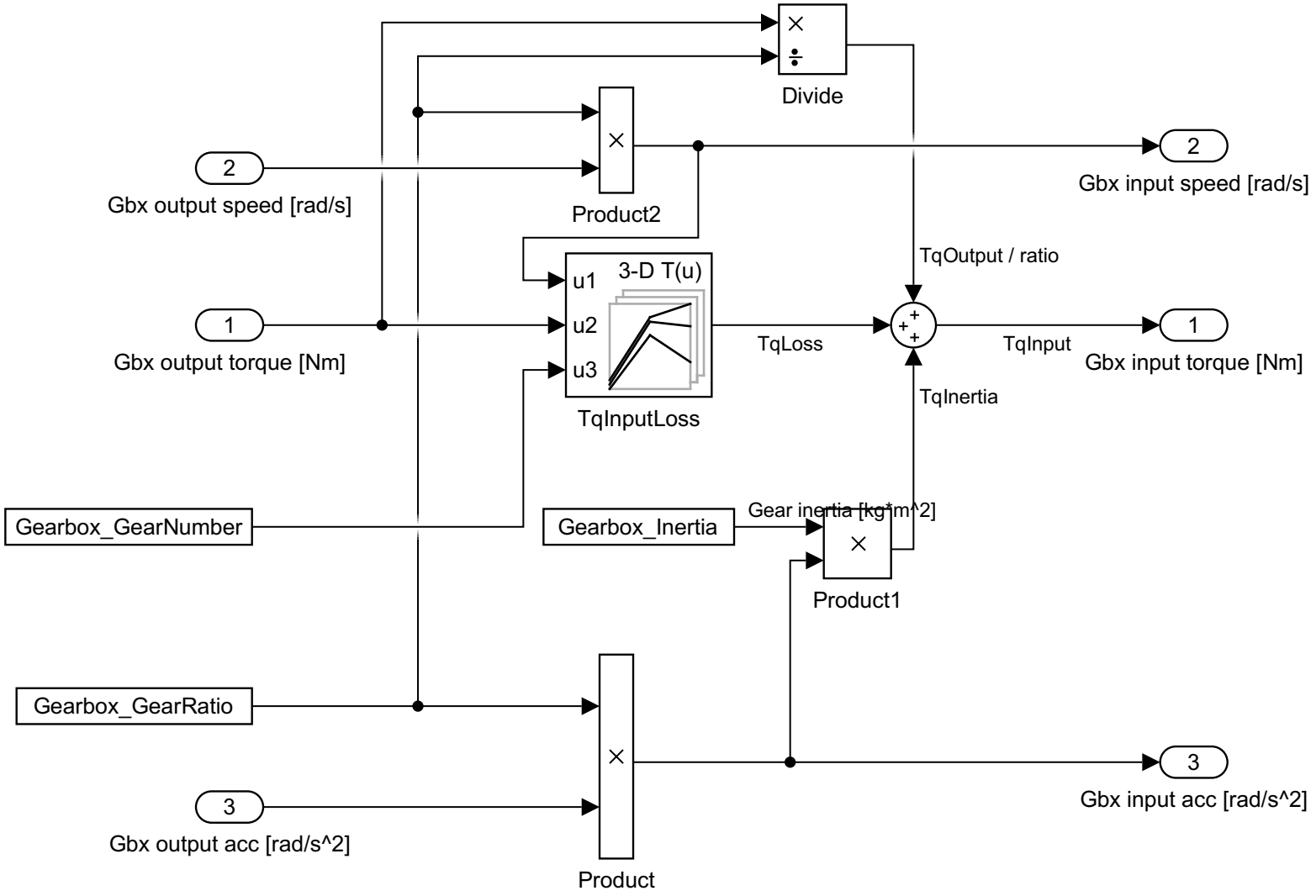

1
Demanded vehicle acceleration [m/s^2]
2
Predicted vehicle speed [m/s]
1-D T(u)
CdxA velocity
dependant look-up
3
Road gradient
vehicle_acc_demand
vehicleSpeed
CdxA
grad
Constants
TqAxlegearOut
wAxlegearOut
dwdt_AxlegearOut
GetWheelForceFromAccDemand
Wheel force
1
Axlegear output torque [Nm]
2
Axlegear output speed [rad/s]
3
Axlegear output acc [rad/s^2]
Wheel_Inertia
Density_Air_Ambient
Wheel_RRC
m_tot
Gravity_Constant
Wheel_Radius

```
function [TqAxlegearOut, wAxlegearOut, dwdt_AxlegearOut]  ...
    = GetWheelForceFromAccDemand(...
    vehicle_acc_demand, vehicleSpeed, CdxA, grad, Constants)

%CdxA =  Air drag resistance factor
wheelInertia = Constants(1); % [kg*m^2]
p_air   = Constants(2); % Air density
RCC_i   = Constants(3); % Rolling resistance
m_tot   = Constants(4); % [kg] total vehicle mass
g       = Constants(5); % [m/s^2] gravity contant
r_dyn   = Constants(6); % [m] Dynamic wheel rolling radius


F_air       = CdxA * p_air / 2 * vehicleSpeed^2;
F_roll      = RCC_i * cos(atan( grad/100 )) * m_tot * g;
F_grad      = sin(atan( grad/100 )) * m_tot * g;


wheel_Force = vehicle_acc_demand * m_tot + F_air + F_roll + F_grad; %[Nm]


wAxlegearOut        = vehicleSpeed / r_dyn; %[rad/s]
dwdt_AxlegearOut    = vehicle_acc_demand / r_dyn; %[rad/s^2]
TqAxlegearOut       = wheel_Force *r_dyn + wheelInertia * dwdt_AxlegearOut; % Nm]
```

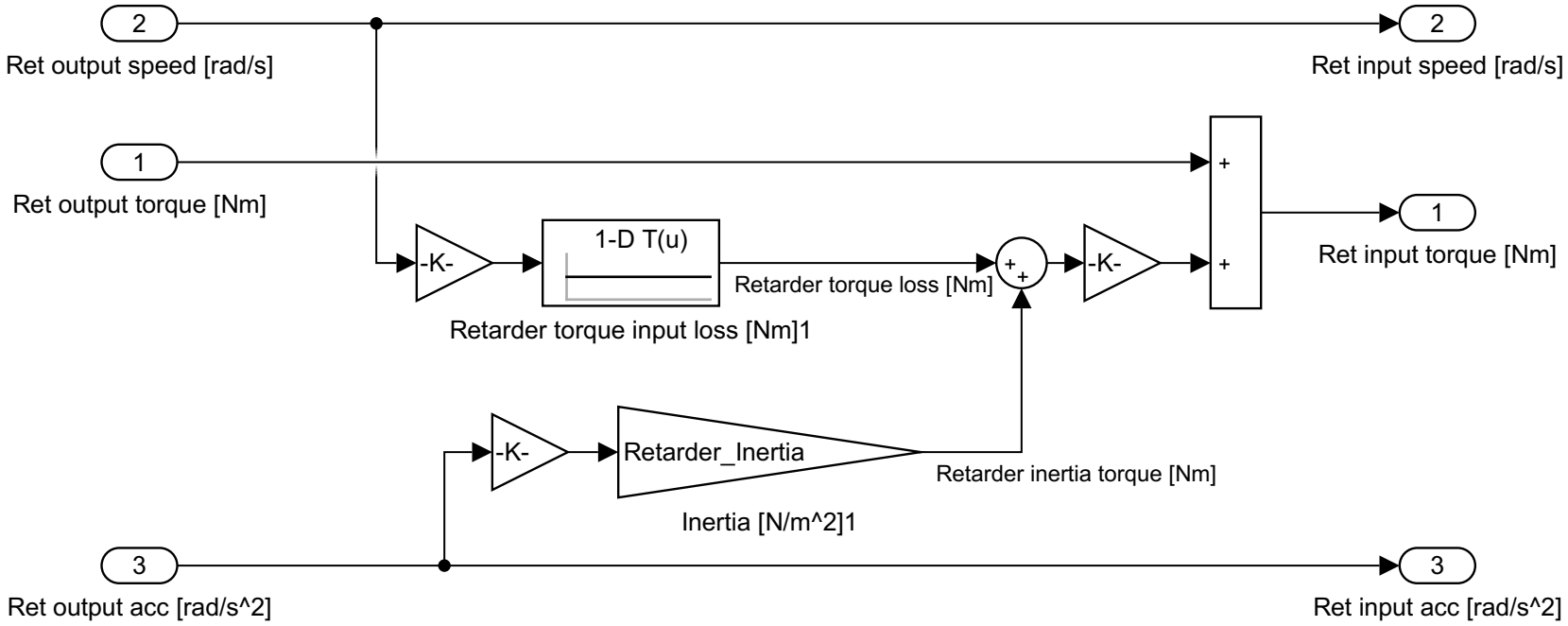

1
Vehicle acceleration [m/s]
2
Vehicle speed [m/s]
3
Road gradient [%]
Demanded vehicle acceleration [m/s^2]
Predicted vehicle speed [m/s]
Road gradient
Axlegear output torque [Nm]
Axlegear output speed [rad/s]
Axlegear output acc [rad/s^2]
Backwards vehicle dynamics
Axlegear output torque [Nm]
Axlegear output speed [rad/s]
Axlegear output acc [rad/s^2]
Axlegear input torque [Nm]
Axlegear input speed [rad/s]
Axlegear input acc [rad/s^2]
Backwards axlegear
10
max
5
w_card_pred [rad/s]
4
Tq_card_pred [Nm]
Ret output torque [Nm]
Ret output speed [rad/s]
Ret output acc [rad/s^2]
Ret input torque [Nm]
Ret input speed [rad/s]
Ret input acc [rad/s^2]
Backwards retarder1
Gbx output torque [Nm]
Gbx output speed [rad/s]
Gbx output acc [rad/s^2]
Gbx input torque [Nm]
Gbx input speed [rad/s]
Gbx input acc [rad/s^2]
Backwards gearbox1
1
Demanded Gbx in Tq [Nm]
2
Predicted Gbx in w [rad/s]
3
Demanded Gbx in acc [rad/s^2]

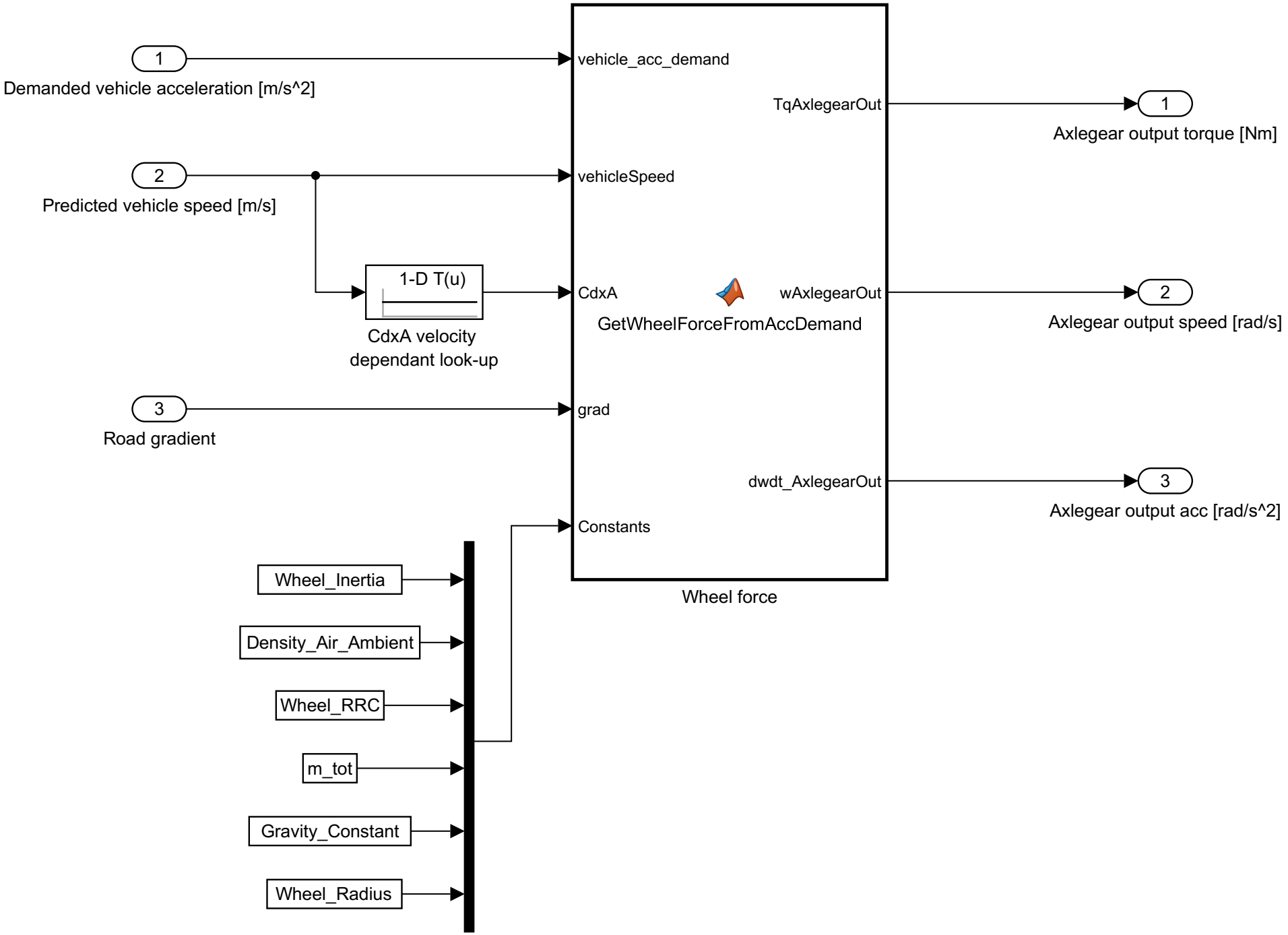
TqInput	= TqOutout / ratio + TqInputLoss + TqInertia	[Nm]
w_Input	= w_Output * ratio	[rad/s]
dwdt_Input	= dwdt_Output * ratio	[rad/s^2]



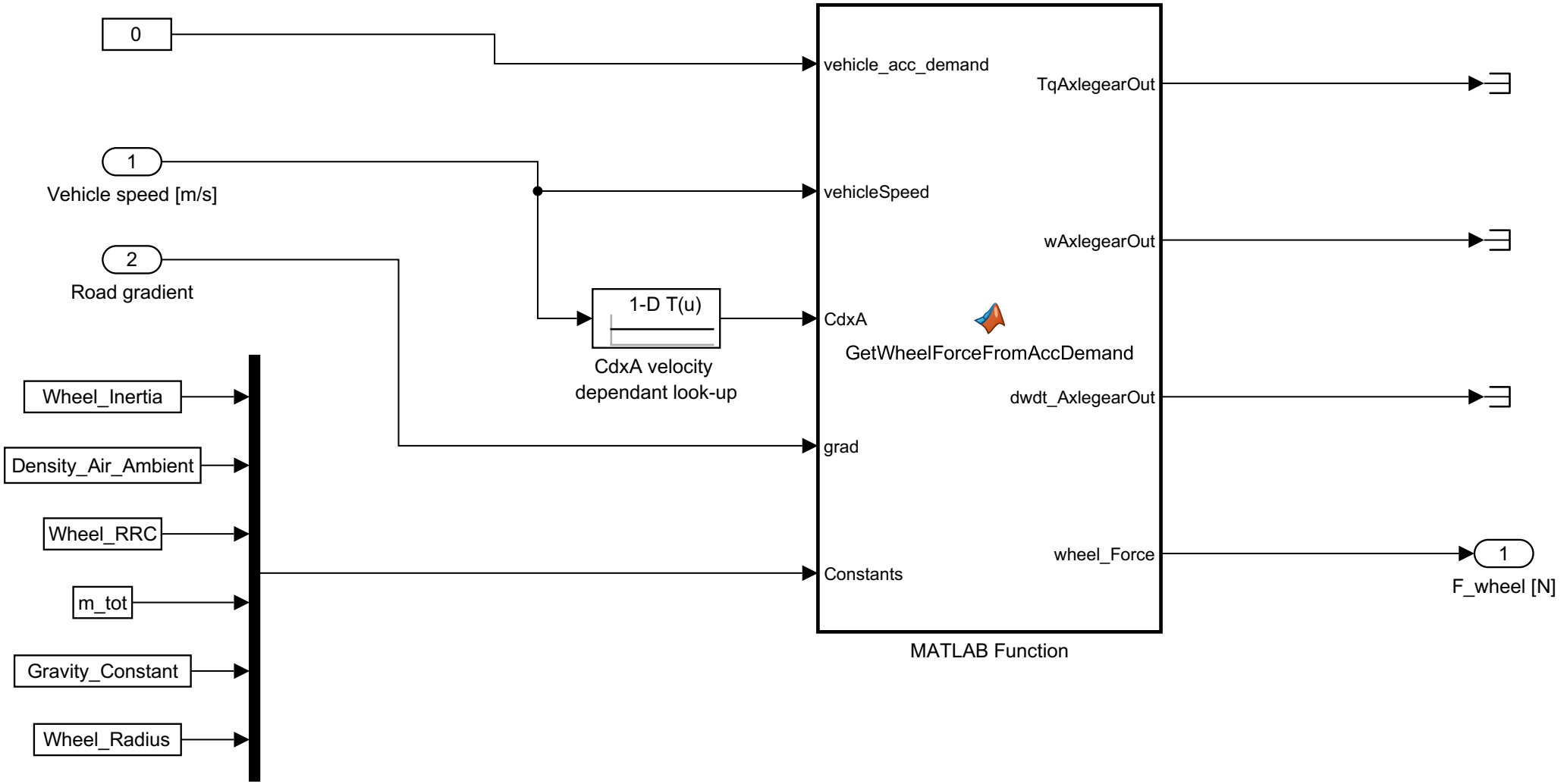

Divide
2
Gbx output speed [rad/s]
Product2
2
Gbx input speed [rad/s]
TqOutput / ratio
1
Gbx output torque [Nm]
u1
u2
u3
3-D T(u)
TqInputLoss
TqLoss
TqInput
1
Gbx input torque [Nm]
TqInertia
Gearbox_GearNumber
Gearbox_Inertia
Gear inertia [kg*m^2]
Product1
Gearbox_GearRatio
3
Gbx output acc [rad/s^2]
Product
3
Gbx input acc [rad/s^2]

TqInput	= TqOutput + ratio*(TqInertia + TqLoss)	[Nm]
w_Input	= w_Output	[rad/s]
dwdt_Input	= dwdt_Output	[rad/s^"]

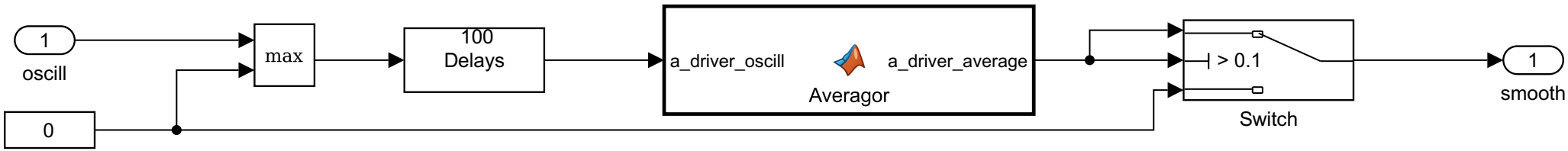



1
Demanded vehicle acceleration [m/s^2]
2
Predicted vehicle speed [m/s]
1-D T(u)
CdxA velocity
dependant look-up
3
Road gradient
vehicle_acc_demand
vehicleSpeed
CdxA
grad
Constants
TqAxlegearOut
wAxlegearOut
dwdt_AxlegearOut
GetWheelForceFromAccDemand
Wheel force
1
Axlegear output torque [Nm]
2
Axlegear output speed [rad/s]
3
Axlegear output acc [rad/s^2]
Wheel_Inertia
Density_Air_Ambient
Wheel_RRC
m_tot
Gravity_Constant
Wheel_Radius

```
function [TqAxlegearOut, wAxlegearOut, dwdt_AxlegearOut]  ...
    = GetWheelForceFromAccDemand(...
    vehicle_acc_demand, vehicleSpeed, CdxA, grad, Constants)

%CdxA =  Air drag resistance factor
wheelInertia = Constants(1); % [kg*m^2]
p_air   = Constants(2); % Air density
RCC_i   = Constants(3); % Rolling resistance
m_tot   = Constants(4); % [kg] total vehicle mass
g       = Constants(5); % [m/s^2] gravity contant
r_dyn   = Constants(6); % [m] Dynamic wheel rolling radius


F_air       = CdxA * p_air / 2 * vehicleSpeed^2;
F_roll      = RCC_i * cos(atan( grad/100 )) * m_tot * g;
F_grad      = sin(atan( grad/100 )) * m_tot * g;


wheel_Force = vehicle_acc_demand * m_tot + F_air + F_roll + F_grad; %[Nm]


wAxlegearOut        = vehicleSpeed / r_dyn; %[rad/s]
dwdt_AxlegearOut    = vehicle_acc_demand / r_dyn; %[rad/s^2]
TqAxlegearOut       = wheel_Force *r_dyn + wheelInertia * dwdt_AxlegearOut; % Nm]

stop =1;

end
```


0
1
Vehicle speed [m/s]
2
Road gradient
1-D T(u)
CdxA velocity
dependant look-up
Wheel_Inertia
Density_Air_Ambient
Wheel_RRC
m_tot
Gravity_Constant
Wheel_Radius
vehicle_acc_demand
vehicleSpeed
CdxA
grad
Constants
GetWheelForceFromAccDemand
MATLAB Function
TqAxlegearOut
wAxlegearOut
dwdt_AxlegearOut
wheel_Force
1
F_wheel [N]

```
function [TqAxlegearOut, wAxlegearOut, dwdt_AxlegearOut, wheel_Force]  ...
    = GetWheelForceFromAccDemand(...
    vehicle_acc_demand, vehicleSpeed, CdxA, grad, Constants)

%CdxA =  Air drag resistance factor
wheelInertia = Constants(1); % [kg*m^2]
p_air   = Constants(2); % Air density
RCC_i   = Constants(3); % Rolling resistance
m_tot   = Constants(4); % [kg] total vehicle mass
g       = Constants(5); % [m/s^2] gravity contant
r_dyn   = Constants(6); % [m] Dynamic wheel rolling radius


F_air       = CdxA * p_air / 2 * vehicleSpeed^2;
F_roll      = RCC_i * cos(atan( grad/100 )) * m_tot * g;
F_grad      = sin(atan( grad/100 )) * m_tot * g;


wheel_Force = vehicle_acc_demand * m_tot + F_air + F_roll + F_grad; %[Nm]


wAxlegearOut        = vehicleSpeed / r_dyn; %[rad/s]
dwdt_AxlegearOut    = vehicle_acc_demand / r_dyn; %[rad/s^2]
TqAxlegearOut       = wheel_Force *r_dyn + wheelInertia * dwdt_AxlegearOut; % Nm]
```

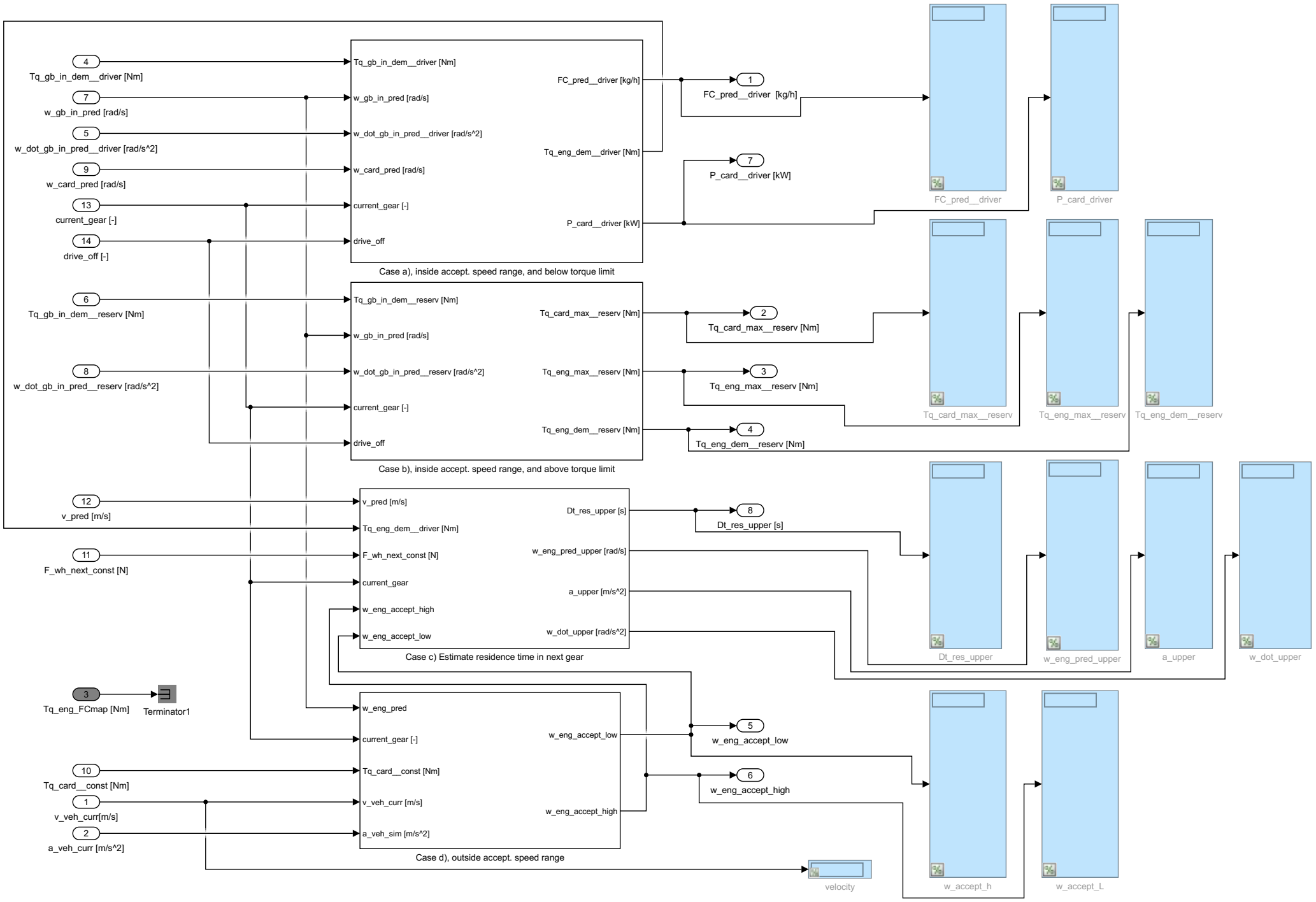

1
oscill
0
max
100
Delays
a_driver_oscill
a_driver_average
Averagor
> 0.1
Switch
1
smooth

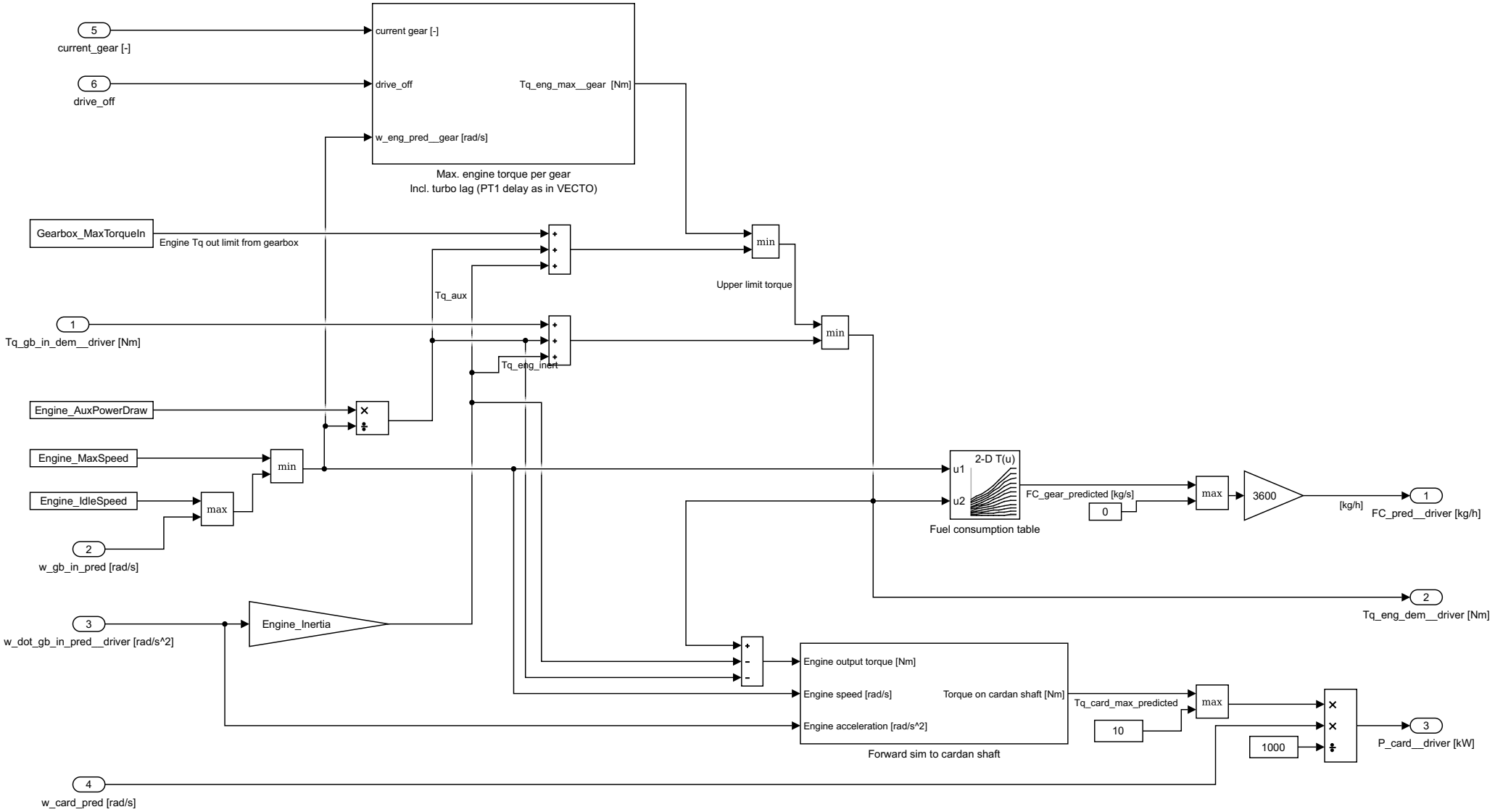
```
function a_driver_average = Averagor(a_driver_oscill)
a_driver_average = mean(a_driver_oscill);
end
```

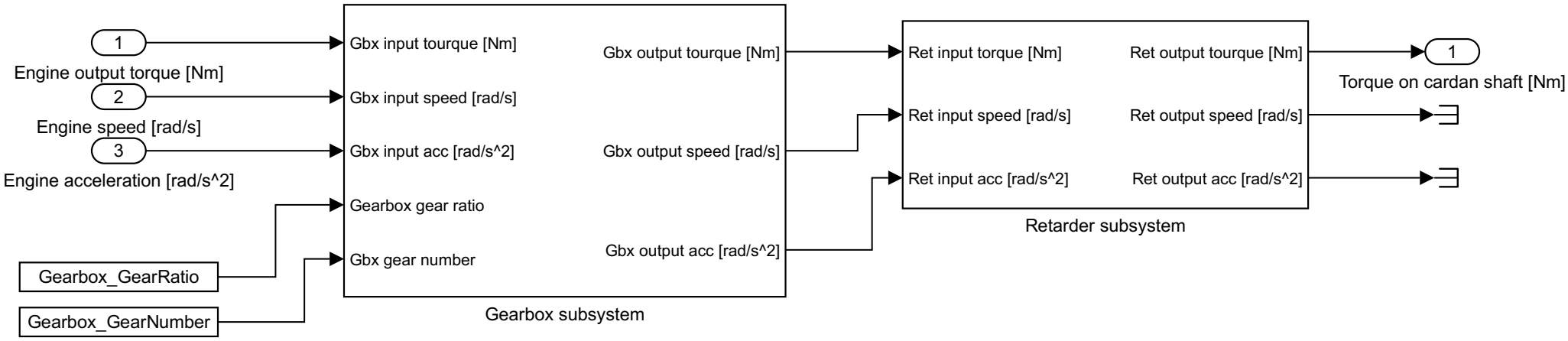
```
function rating_gear  ...
    = GearRating( w_eng_pred, ...
                  FC_pred__driver, ...
                  Tq_card_max__reserv, ...
                  Tq_eng_max__reserv, ...
                  Tq_eng_dem__reserv, ...
                  w_eng_accept_low, ...
                  w_eng_accept_high, ...
                  P_card_dem__driver, ...
                  Dt_res_upper, ...
                  Tq_card_dem__reserv, ...
                  Numbers_0_D, ...
                  currentGear, ...
                  drive_off ...
                 )


  no_gears                = length(w_eng_pred);
  rating_gear             = ones(no_gears,1);

  penalty_medium          = Numbers_0_D(1);
  penalty_high            = Numbers_0_D(2);
  penalty_very_high       = Numbers_0_D(3);
  GearResTime             = Numbers_0_D(4);


for i = 1 : 1 : no_gears


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Case a) Speed inside acceptable range, torque below limits.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    % Torque values calculated from acceleration reserve.



    if Tq_eng_dem__reserv(i)      <= Tq_eng_max__reserv(i)


    % Fuel consumption and cardan power calculated from the driver
    % acceleration.
    %
    % The lower limit for the low range of rating factors is the minimum
    % specific fuel consumption, referred to the power at the cardan shaft.
    % The upper limit for the low range of rating factor is penalty_medium.

       rating_gear(i) = ...
           ...
           min( ( FC_pred__driver(i) * 1000 / P_card_dem__driver(i) ) , ...
                ( penalty_medium - 100 ) ...
               );



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Case b) Speed inside acceptable range, torque above limit.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
```

```
% Torque values calculated from acceleration reserve.


else


% The lower limit for the medium range of rating factors is penalty_medium.
% The upper limit for the medium range of rating factors is penalty_high.

   rating_gear(i) = ...
        ...
        min( ( max(0, Tq_card_dem__reserv - Tq_card_max__reserv(i)) + ...
               penalty_medium                                        ), ...
             ( penalty_high - 100 ) ...
           );

end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case c) Speed inside acceptable range, but estimated residence time
%         in optional other gear is too short.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Estimated residence time per gear calculated with the driver
% acceleration

   if ( i                > currentGear ) && ...
      ( Dt_res_upper(i)  ~= 0          ) && ...
      ( Dt_res_upper(i)  < GearResTime )


   % The lower limit for the high range of rating factors is
   % penalty_high.
   % The upper limit for the high range of rating factors is
   % penalty_very_high.

       rating_gear(i) = ...
           ...
           min( ( GearResTime - Dt_res_upper(i) + ...
                  penalty_high                        ), ...
                ( penalty_very_high - 100 ) ...
              );

   else

   % If the estimated residence time is not too low, don't change
   % the previous ratings case a) or b)

   end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case d) Speed outside acceptable range
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
    % If the predicted engine speed of a rated gear is outside of the range
    % of accepted speeds, the gear gets the very high penalty.


    % The lower limit for the very high range of rating factors is
    % penalty_very_high.
    % The upper limit for the very_high range of rating factors is the
    % highest distance from the predicted to the min. or max. accepted
    % engine speed



    if drive_off == 0


        if     ( w_eng_pred(i)         < w_eng_accept_low(i) )

            rating_gear(i) = ( (-1) * ( w_eng_pred(i) - ...
                                        w_eng_accept_low(i) ) ...
                             ) + ...
                            penalty_very_high ;

        elseif ( w_eng_accept_high(i)  < w_eng_pred(i) )

            rating_gear(i) = ( (+1) * ( w_eng_pred(i) - ...
                                        w_eng_accept_high(i) ) ...
                             ) + ...
                            penalty_very_high ;

        end

    else
    end

end


%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Current gear gets 1 % bonus if in low range of FC-rating,
% to avoid gearhunting.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if rating_gear( currentGear ) < penalty_medium

    rating_gear( currentGear ) = rating_gear( currentGear ) * 0.99;

else
end

%%

% stopper = 1;

end
```

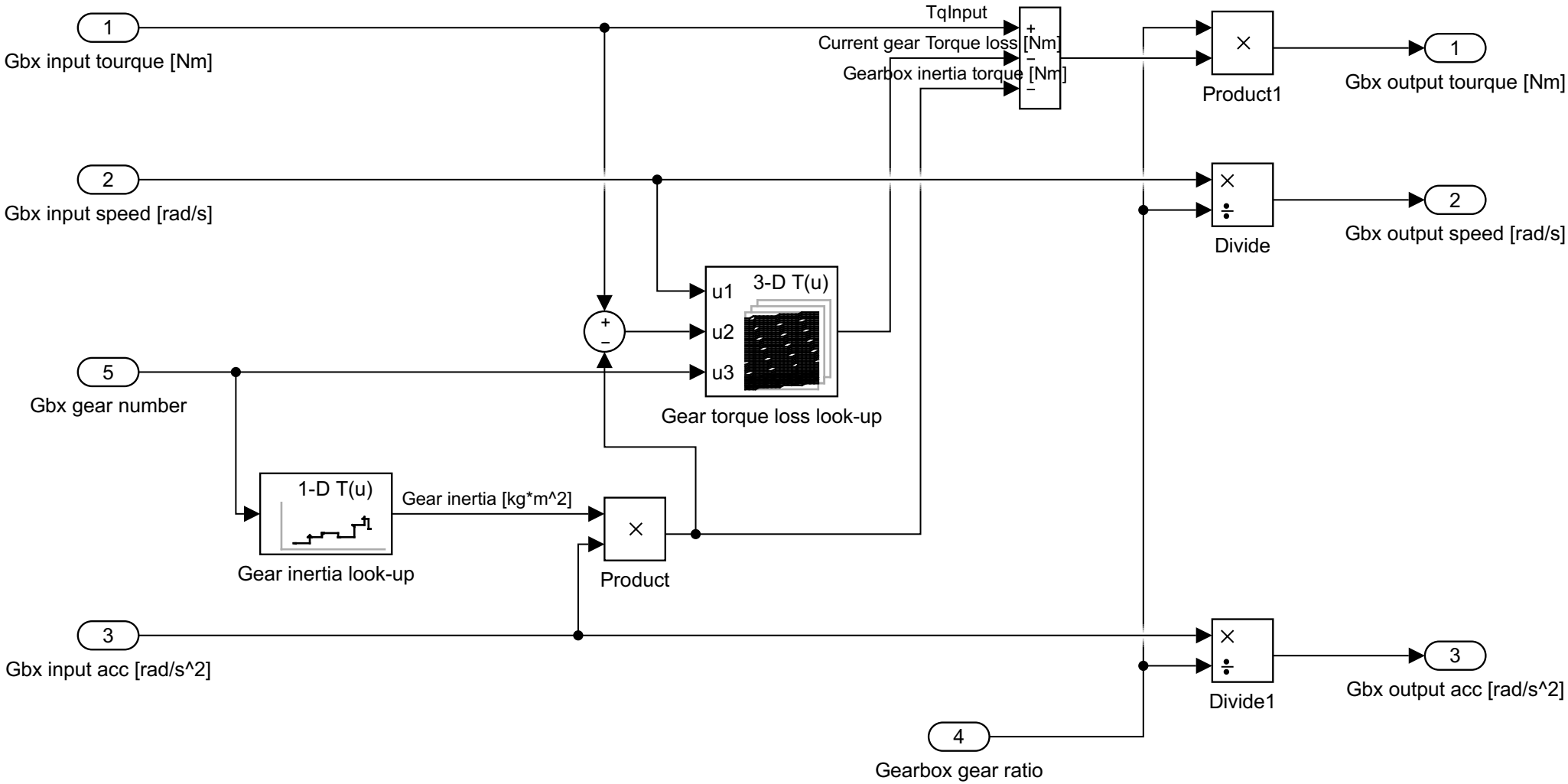

4
Tq_gb_in_dem__driver [Nm]
7
w_gb_in_pred [rad/s]
5
w_dot_gb_in_pred__driver [rad/s^2]
9
w_card_pred [rad/s]
13
current_gear [-]
14
drive_off [-]
Tq_gb_in_dem__driver [Nm]
w_gb_in_pred [rad/s]
w_dot_gb_in_pred__driver [rad/s^2]
w_card_pred [rad/s]
current_gear [-]
drive_off
FC_pred__driver [kg/h]
Tq_eng_dem__driver [Nm]
P_card__driver [kW]
Case a), inside accept. speed range, and below torque limit
1
FC_pred__driver [kg/h]
7
P_card__driver [kW]
FC_pred__driver
P_card_driver
6
Tq_gb_in_dem__reserv [Nm]
8
w_dot_gb_in_pred__reserv [rad/s^2]
Tq_gb_in_dem__reserv [Nm]
w_gb_in_pred [rad/s]
w_dot_gb_in_pred__reserv [rad/s^2]
current_gear [-]
drive_off
Tq_card_max__reserv [Nm]
Tq_eng_max__reserv [Nm]
Tq_eng_dem__reserv [Nm]
Case b), inside accept. speed range, and above torque limit
2
Tq_card_max__reserv [Nm]
3
Tq_eng_max__reserv [Nm]
4
Tq_eng_dem__reserv [Nm]
Tq_card_max__reserv
Tq_eng_max__reserv
Tq_eng_dem__reserv
12
v_pred [m/s]
11
F_wh_next_const [N]
v_pred [m/s]
Tq_eng_dem__driver [Nm]
F_wh_next_const [N]
current_gear
w_eng_accept_high
w_eng_accept_low
Dt_res_upper [s]
w_eng_pred_upper [rad/s]
a_upper [m/s^2]
w_dot_upper [rad/s^2]
Case c) Estimate residence time in next gear
8
Dt_res_upper [s]
Dt_res_upper
w_eng_pred_upper
a_upper
w_dot_upper
3
Tq_eng_FCmap [Nm]
Terminator1
10
Tq_card__const [Nm]
1
v_veh_curr[m/s]
2
a_veh_curr [m/s^2]
w_eng_pred
current_gear [-]
Tq_card__const [Nm]
v_veh_curr [m/s]
a_veh_sim [m/s^2]
w_eng_accept_low
w_eng_accept_high
Case d), outside accept. speed range
5
w_eng_accept_low
6
w_eng_accept_high
velocity
w_accept_h
w_accept_L


5
current_gear [-]
6
drive_off
current gear [-]
drive_off
w_eng_pred__gear [rad/s]
Tq_eng_max__gear [Nm]
Max. engine torque per gear
Incl. turbo lag (PT1 delay as in VECTO)
Gearbox_MaxTorqueIn
Engine Tq out limit from gearbox
min
Upper limit torque
Tq_aux
1
Tq_gb_in_dem__driver [Nm]
Tq_eng_inert
min
Engine_AuxPowerDraw
Engine_MaxSpeed
min
Engine_IdleSpeed
max
2
w_gb_in_pred [rad/s]
2-D T(u)
u1
u2
Fuel consumption table
FC_gear_predicted [kg/s]
0
max
3600
[kg/h]
1
FC_pred__driver [kg/h]
2
Tq_eng_dem__driver [Nm]
3
w_dot_gb_in_pred__driver [rad/s^2]
Engine_Inertia
Engine output torque [Nm]
Engine speed [rad/s]
Engine acceleration [rad/s^2]
Torque on cardan shaft [Nm]
Forward sim to cardan shaft
Tq_card_max_predicted
10
max
1000
3
P_card__driver [kW]
4
w_card_pred [rad/s]

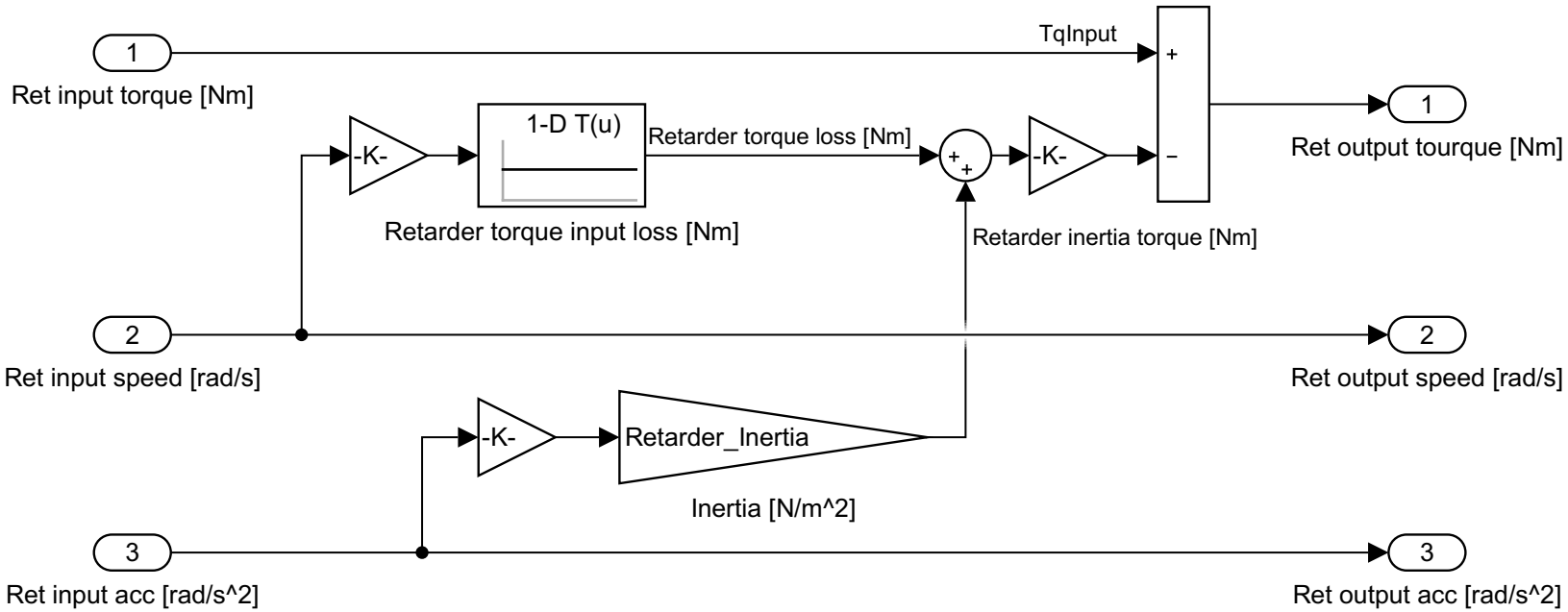

1
Engine output torque [Nm]
2
Engine speed [rad/s]
3
Engine acceleration [rad/s^2]
Gearbox_GearRatio
Gearbox_GearNumber
Gbx input tourque [Nm]
Gbx input speed [rad/s]
Gbx input acc [rad/s^2]
Gearbox gear ratio
Gbx gear number
Gbx output tourque [Nm]
Gbx output speed [rad/s]
Gbx output acc [rad/s^2]
Gearbox subsystem
Ret input torque [Nm]
Ret input speed [rad/s]
Ret input acc [rad/s^2]
Ret output tourque [Nm]
Ret output speed [rad/s]
Ret output acc [rad/s^2]
Retarder subsystem
1
Torque on cardan shaft [Nm]

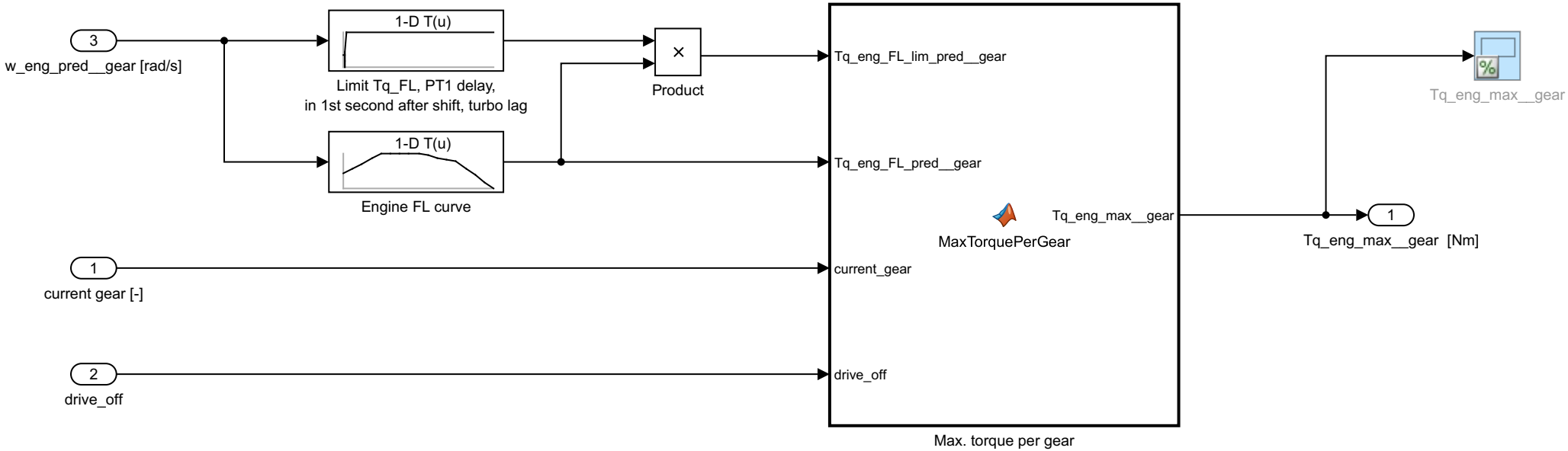
TqOutput = (TqInput - TqInputLoss - TqInertia) * gearRatio; [Nm]
wOutput = wInput / gearRatio; [rad/s]
dwdtOutput = dwdtInput / gearRatio; [rad/s^2]

Input torque for lossmap is defined for input tq for the gear set inside. Thus we subtract the inertia torque before looking up the tq loss



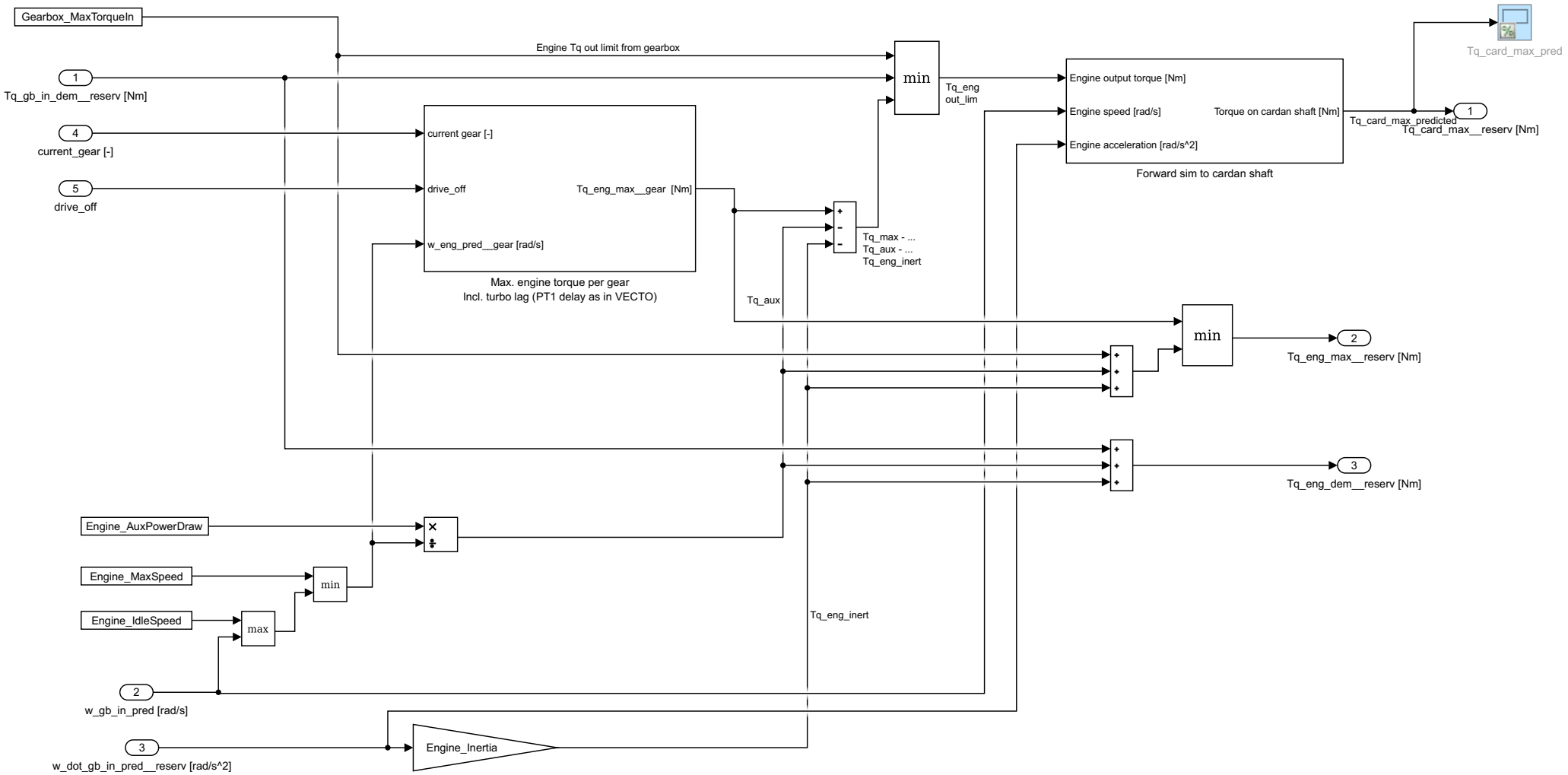
TqOutput	= TqInput + ratio*(TqInertia + TqLoss)	[Nm]
w_Input	= w_Output	[rad/s]
dwdt_Input	= dwdt_Output	[rad/s^"]

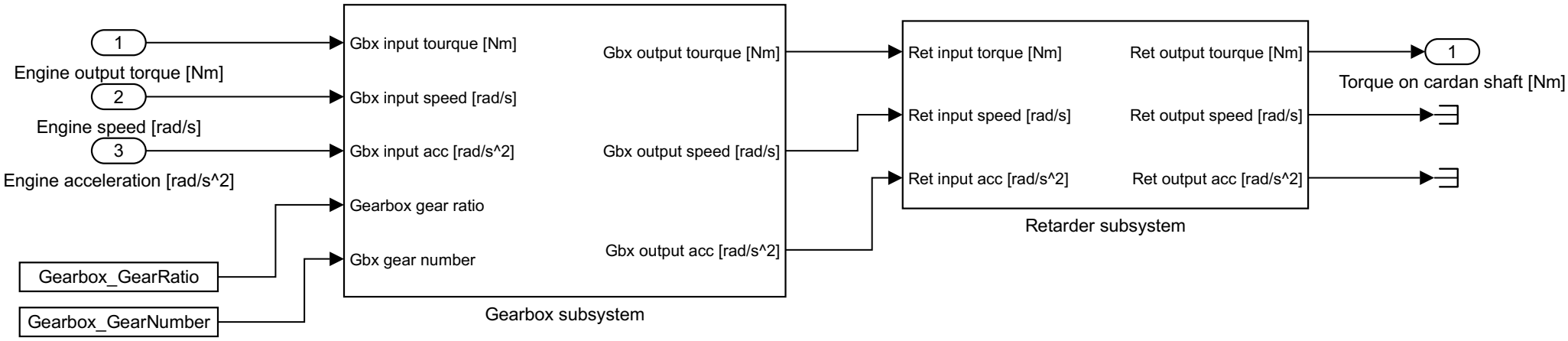



3
w_eng_pred__gear [rad/s]
1-D T(u)
Limit Tq_FL, PT1 delay,
in 1st second after shift, turbo lag
1-D T(u)
Engine FL curve
×
Product
Tq_eng_FL_lim_pred__gear
Tq_eng_FL_pred__gear
current_gear
drive_off
Tq_eng_max__gear
MaxTorquePerGear
Max. torque per gear
Tq_eng_max__gear
1
Tq_eng_max__gear [Nm]
1
current gear [-]
2
drive_off

```
function Tq_eng_max__gear  = MaxTorquePerGear( Tq_eng_FL_lim_pred__gear, ...
                                                Tq_eng_FL_pred__gear,  ...
                                                current_gear, ...
                                                drive_off ...
                                               )


%%

Tq_eng_max__gear   = Tq_eng_FL_lim_pred__gear;



%%

if drive_off == 0

    for i = 1 : 1 : length(Tq_eng_FL_lim_pred__gear)

        if i == current_gear

            Tq_eng_max__gear(i) = Tq_eng_FL_pred__gear(i);

        else
        end

    end

else

    Tq_eng_max__gear       = Tq_eng_FL_pred__gear;

end

%%

% stopper = 1;



%%
%
% Optional further inputs
%
%  w_eng_curr
%  Tq_eng_FCmap_curr
%  Tq_eng_FL_curr
%  PT1_const_curr



%%

% % See VECTO manual, entry "Engine: Transient Full Load"
% % ------------------------------------------------------------------------
% % Nomenclature:
% % ------------------------------------------------------------------------
% % VECTO manual       This function
% % ------------------------------------------------------------------------
```

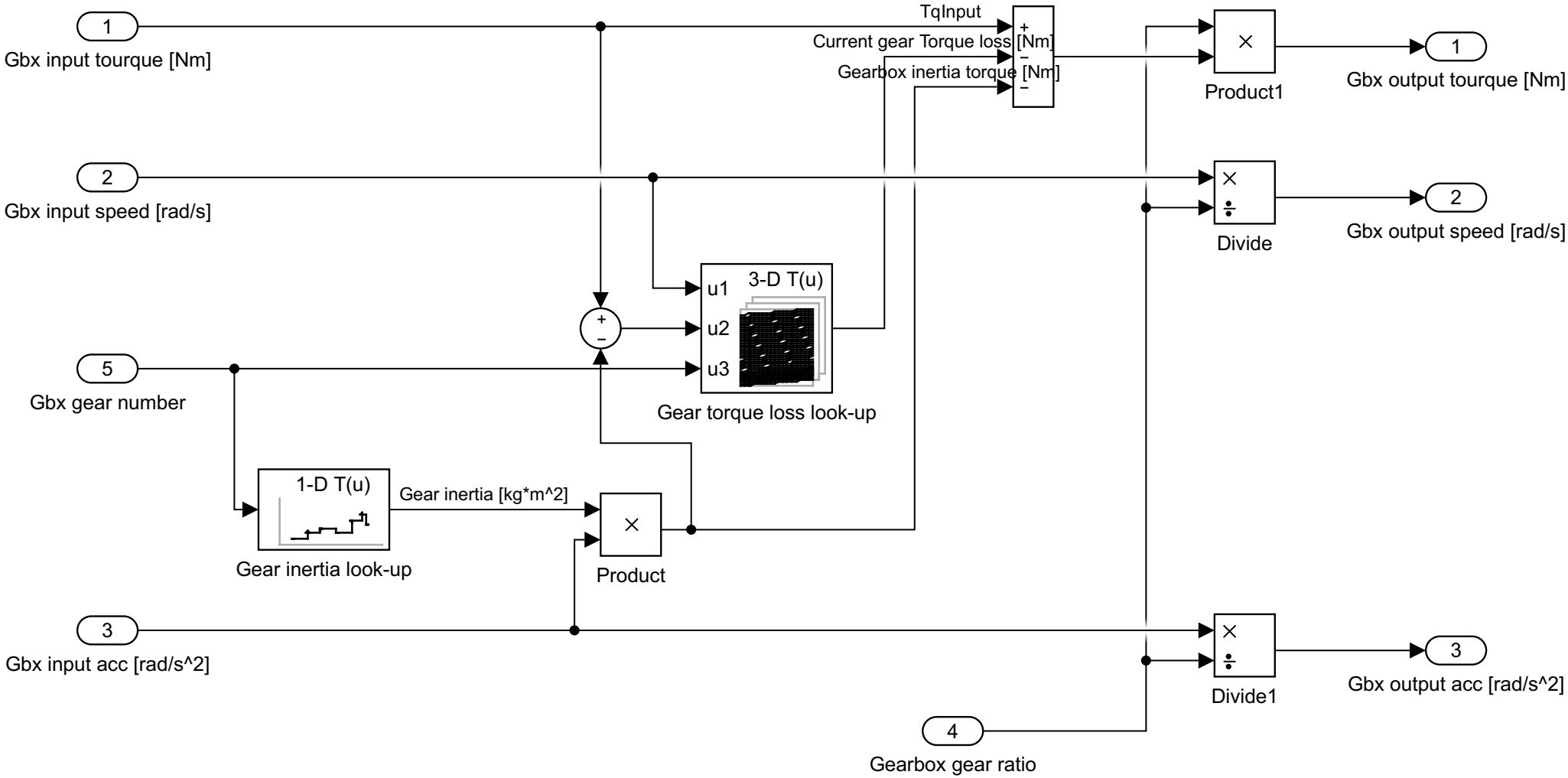
```
% % P_eng_i-1           P_eng_FCmap_curr
% % P_fld_stat(n_i)     P_eng_FL_curr = w_eng_curr * Tq_eng_FL_curr
% % PT1                 PT1_const_curr
% % t*_i-1              tx_old
% % dt                  dt = 2 s
% % t*_i                tx_curr
% % P_fld_dyn_i         P_eng_FL_PT1_curr
%
% P_eng_FL_curr      = w_eng_curr * Tq_eng_FL_curr;
%
% P_eng_FCmap_curr   = max( min( w_eng_curr * Tq_eng_FCmap_curr, ...
%                                P_eng_FL_curr - 0.01), ...
%                           0);
%
% tx_old             = PT1_const_curr * ...
%                      log( 1 / ( 1 - P_eng_FCmap_curr / P_eng_FL_curr ) );
%
% dt                 = 2;
%
% tx_curr            = tx_old + dt;
%
% P_eng_FL_PT1_curr  = P_eng_FL_curr * ...
%                      ( 1 - exp( -( tx_curr / PT1_const_curr ) ) );
%
% Tq_eng_FL_PT1_curr = P_eng_FL_PT1_curr / w_eng_curr;
%
%
%
% %%
%
% if drive_off == 0
%
%     for i = 1 : 1 : length(Tq_eng_FL_lim_pred__gear)
%
%         if i == current_gear
%
%             Tq_eng_max__gear(i) = Tq_eng_FL_PT1_curr;
%
%         else
%         end
%
%     end
%
% else
%
%     Tq_eng_max__gear       = Tq_eng_FL_pred__gear;
%
% end


end
```


Gearbox_MaxTorqueIn
Engine Tq out limit from gearbox
1
Tq_gb_in_dem__reserv [Nm]
4
current_gear [-]
5
drive_off
current gear [-]
drive_off
Tq_eng_max__gear [Nm]
w_eng_pred__gear [rad/s]
Max. engine torque per gear
Incl. turbo lag (PT1 delay as in VECTO)
min
Tq_eng
out_lim
Tq_max - ...
Tq_aux - ...
Tq_eng_inert
Tq_aux
Engine output torque [Nm]
Engine speed [rad/s]
Torque on cardan shaft [Nm]
Engine acceleration [rad/s^2]
Forward sim to cardan shaft
Tq_card_max_pred
1
Tq_card_max_predicted
Tq_card_max__reserv [Nm]
min
2
Tq_eng_max__reserv [Nm]
3
Tq_eng_dem__reserv [Nm]
Engine_AuxPowerDraw
Engine_MaxSpeed
Engine_IdleSpeed
min
max
Tq_eng_inert
2
w_gb_in_pred [rad/s]
3
w_dot_gb_in_pred__reserv [rad/s^2]
Engine_Inertia

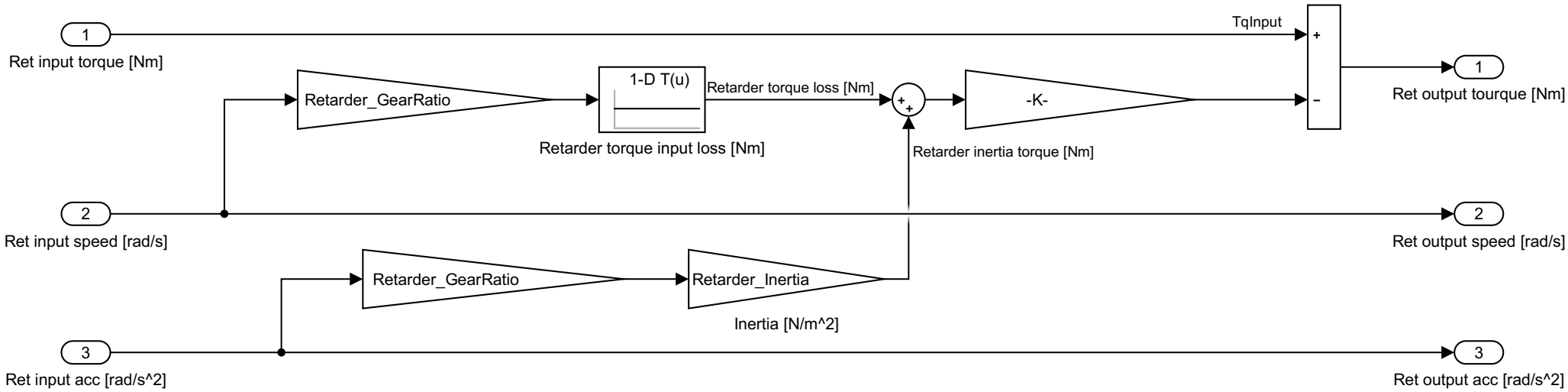

1
Engine output torque [Nm]
2
Engine speed [rad/s]
3
Engine acceleration [rad/s^2]
Gearbox_GearRatio
Gearbox_GearNumber
Gbx input tourque [Nm]
Gbx input speed [rad/s]
Gbx input acc [rad/s^2]
Gearbox gear ratio
Gbx gear number
Gbx output tourque [Nm]
Gbx output speed [rad/s]
Gbx output acc [rad/s^2]
Gearbox subsystem
Ret input torque [Nm]
Ret input speed [rad/s]
Ret input acc [rad/s^2]
Ret output tourque [Nm]
Ret output speed [rad/s]
Ret output acc [rad/s^2]
Retarder subsystem
1
Torque on cardan shaft [Nm]

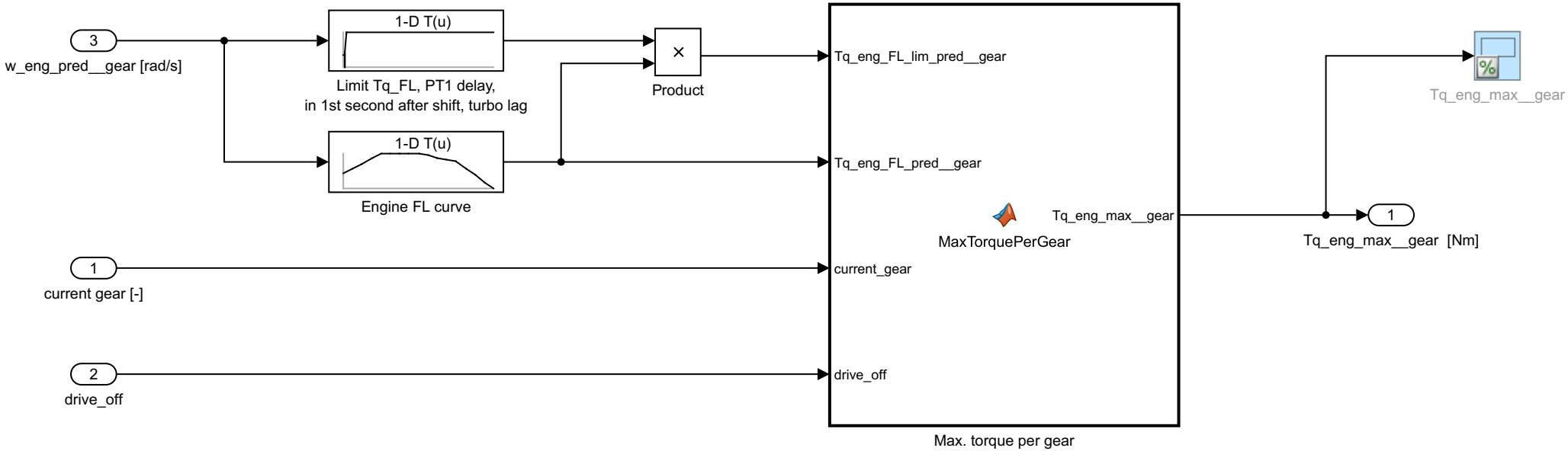
TqOutput = (TqInput - TqInputLoss - TqInertia) * gearRatio; [Nm]
wOutput = wInput / gearRatio; [rad/s]
dwdtOutput = dwdtInput / gearRatio; [rad/s^2]

Input torque for lossmap is defined for input tq for the gear set inside. Thus we subtract the inertia torque before looking up the tq loss



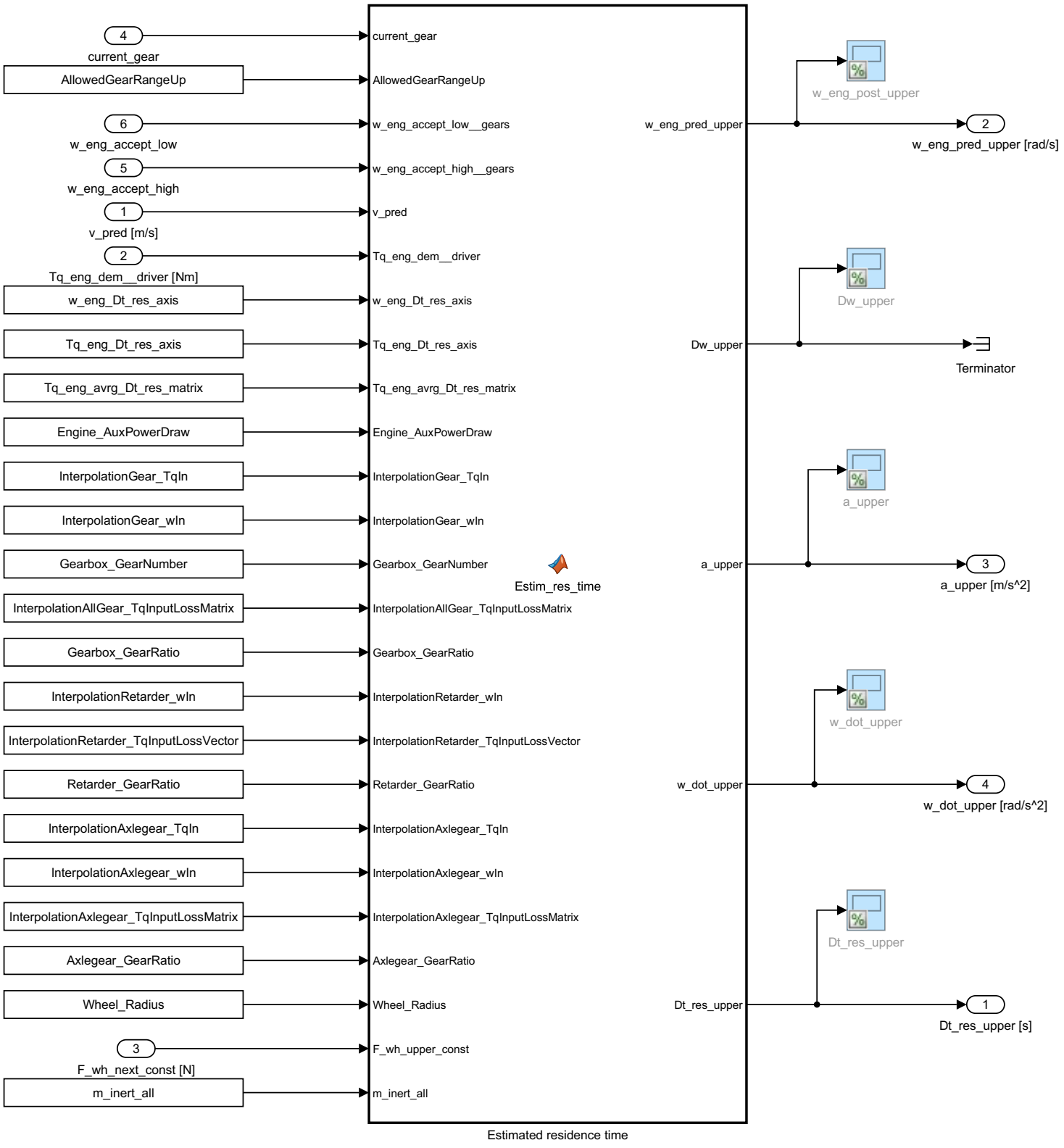
TqOutput	= TqInput + ratio*(TqInertia + TqLoss)	[Nm]
w_Input	= w_Output	[rad/s]
dwdt_Input	= dwdt_Output	[rad/s^"]




3
w_eng_pred__gear [rad/s]
1-D T(u)
Limit Tq_FL, PT1 delay,
in 1st second after shift, turbo lag
1-D T(u)
Engine FL curve
×
Product
Tq_eng_FL_lim_pred__gear
Tq_eng_FL_pred__gear
current_gear
drive_off
Tq_eng_max__gear
MaxTorquePerGear
Max. torque per gear
Tq_eng_max__gear
1
Tq_eng_max__gear [Nm]
1
current gear [-]
2
drive_off

```
function Tq_eng_max__gear  = MaxTorquePerGear( Tq_eng_FL_lim_pred__gear, ...
                                                Tq_eng_FL_pred__gear,  ...
                                                current_gear, ...
                                                drive_off ...
                                               )


%%

Tq_eng_max__gear   = Tq_eng_FL_lim_pred__gear;



%%

if drive_off == 0

    for i = 1 : 1 : length(Tq_eng_FL_lim_pred__gear)

        if i == current_gear

            Tq_eng_max__gear(i) = Tq_eng_FL_pred__gear(i);

        else
        end

    end

else

    Tq_eng_max__gear       = Tq_eng_FL_pred__gear;

end

%%

% stopper = 1;



%%
%
% Optional further inputs
%
%  w_eng_curr
%  Tq_eng_FCmap_curr
%  Tq_eng_FL_curr
%  PT1_const_curr



%%

% % See VECTO manual, entry "Engine: Transient Full Load"
% % ------------------------------------------------------------------------
% % Nomenclature:
% % ------------------------------------------------------------------------
% % VECTO manual       This function
% % ------------------------------------------------------------------------
```

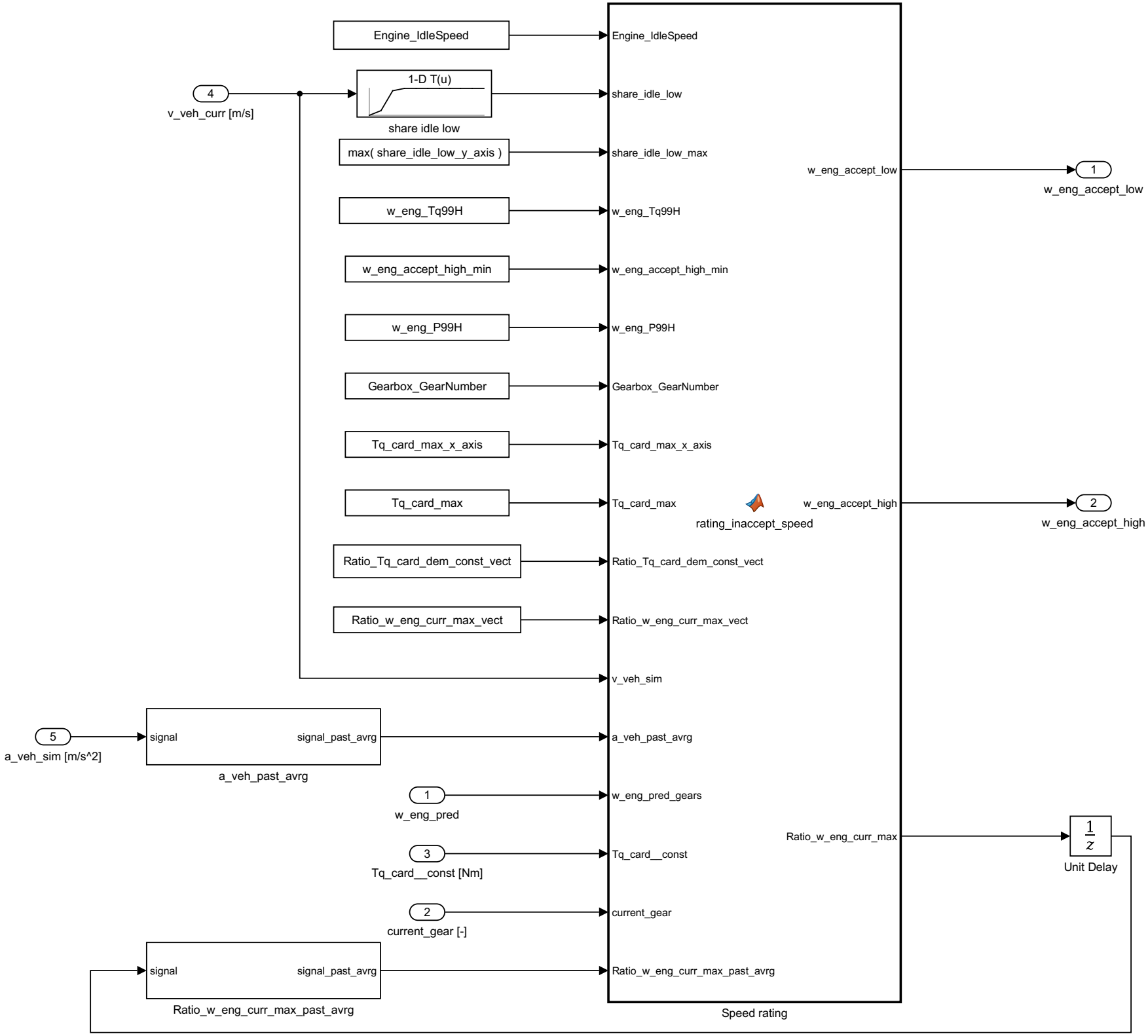
```
% % P_eng_i-1           P_eng_FCmap_curr
% % P_fld_stat(n_i)     P_eng_FL_curr = w_eng_curr * Tq_eng_FL_curr
% % PT1                 PT1_const_curr
% % t*_i-1              tx_old
% % dt                  dt = 2 s
% % t*_i                tx_curr
% % P_fld_dyn_i         P_eng_FL_PT1_curr
%
% P_eng_FL_curr      = w_eng_curr * Tq_eng_FL_curr;
%
% P_eng_FCmap_curr   = max( min( w_eng_curr * Tq_eng_FCmap_curr, ...
%                                P_eng_FL_curr - 0.01), ...
%                           0);
%
% tx_old             = PT1_const_curr * ...
%                      log( 1 / ( 1 - P_eng_FCmap_curr / P_eng_FL_curr ) );
%
% dt                 = 2;
%
% tx_curr            = tx_old + dt;
%
% P_eng_FL_PT1_curr  = P_eng_FL_curr * ...
%                      ( 1 - exp( -( tx_curr / PT1_const_curr ) ) );
%
% Tq_eng_FL_PT1_curr = P_eng_FL_PT1_curr / w_eng_curr;
%
%
%
% %%
%
% if drive_off == 0
%
%     for i = 1 : 1 : length(Tq_eng_FL_lim_pred__gear)
%
%         if i == current_gear
%
%             Tq_eng_max__gear(i) = Tq_eng_FL_PT1_curr;
%
%         else
%         end
%
%     end
%
% else
%
%     Tq_eng_max__gear       = Tq_eng_FL_pred__gear;
%
% end


end
```

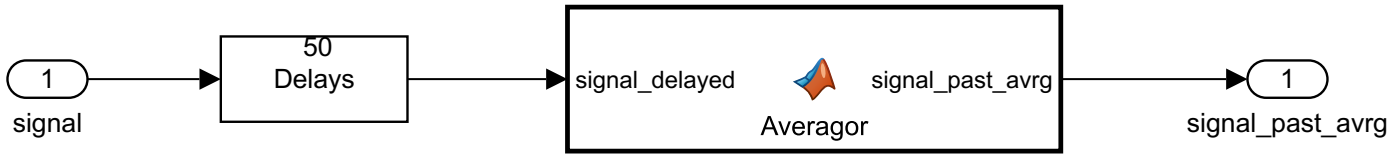

4
current_gear
AllowedGearRangeUp
6
w_eng_accept_low
5
w_eng_accept_high
1
v_pred [m/s]
2
Tq_eng_dem__driver [Nm]
w_eng_Dt_res_axis
Tq_eng_Dt_res_axis
Tq_eng_avrg_Dt_res_matrix
Engine_AuxPowerDraw
InterpolationGear_TqIn
InterpolationGear_wIn
Gearbox_GearNumber
InterpolationAllGear_TqInputLossMatrix
Gearbox_GearRatio
InterpolationRetarder_wIn
InterpolationRetarder_TqInputLossVector
Retarder_GearRatio
InterpolationAxlegear_TqIn
InterpolationAxlegear_wIn
InterpolationAxlegear_TqInputLossMatrix
Axlegear_GearRatio
Wheel_Radius
3
F_wh_next_const [N]
m_inert_all
current_gear
AllowedGearRangeUp
w_eng_accept_low__gears
w_eng_accept_high__gears
v_pred
Tq_eng_dem__driver
w_eng_Dt_res_axis
Tq_eng_Dt_res_axis
Tq_eng_avrg_Dt_res_matrix
Engine_AuxPowerDraw
InterpolationGear_TqIn
InterpolationGear_wIn
Gearbox_GearNumber
Estim_res_time
InterpolationAllGear_TqInputLossMatrix
Gearbox_GearRatio
InterpolationRetarder_wIn
InterpolationRetarder_TqInputLossVector
Retarder_GearRatio
InterpolationAxlegear_TqIn
InterpolationAxlegear_wIn
InterpolationAxlegear_TqInputLossMatrix
Axlegear_GearRatio
Wheel_Radius
F_wh_upper_const
m_inert_all
w_eng_pred_upper
Dw_upper
a_upper
w_dot_upper
Dt_res_upper
w_eng_post_upper
Dw_upper
a_upper
w_dot_upper
Dt_res_upper
2
w_eng_pred_upper [rad/s]
Terminator
3
a_upper [m/s^2]
4
w_dot_upper [rad/s^2]
1
Dt_res_upper [s]
Estimated residence time

```
function [ w_eng_pred_upper,  ...
           Dw_upper, a_upper, ...
           w_dot_upper,       ...
           Dt_res_upper           ] ...
    ...
    = Estim_res_time( current_gear, ...
                      AllowedGearRangeUp,  ...
                      w_eng_accept_low__gears, ...
                      w_eng_accept_high__gears, ...
                      v_pred, ...
                      Tq_eng_dem__driver, ...
                      w_eng_Dt_res_axis, ...
                      Tq_eng_Dt_res_axis, ...
                      Tq_eng_avrg_Dt_res_matrix, ...
                      Engine_AuxPowerDraw, ...
                      InterpolationGear_TqIn, ...
                      InterpolationGear_wIn, ...
                      Gearbox_GearNumber, ...
                      InterpolationAllGear_TqInputLossMatrix, ...
                      Gearbox_GearRatio, ...
                      InterpolationRetarder_wIn, ...
                      InterpolationRetarder_TqInputLossVector, ...
                      Retarder_GearRatio, ...
                      InterpolationAxlegear_TqIn, ...
                      InterpolationAxlegear_wIn, ...
                      InterpolationAxlegear_TqInputLossMatrix, ...
                      Axlegear_GearRatio, ...
                      Wheel_Radius, ...
                      F_wh_upper_const, ...
                      m_inert_all ...
                     )

no_gears            = length(Tq_eng_dem__driver);
w_eng_pred_upper    = zeros(no_gears, 1);
Dw_upper            = w_eng_pred_upper;
a_upper             = w_eng_pred_upper;
w_dot_upper         = w_eng_pred_upper;
Dt_res_upper        = w_eng_pred_upper;

for i = 1 : 1 : no_gears

    if ( i >  ( current_gear                    ) ) && ...
       ( i <= ( current_gear + AllowedGearRangeUp) )

        w_eng_pred              = ( v_pred / Wheel_Radius ) * ...
                                  ( Axlegear_GearRatio    ) * ...
                                  ( Gearbox_GearRatio(i)  ) ;

        if ( w_eng_accept_low__gears(i) < w_eng_pred                  ) && ...
           ( w_eng_pred                 < w_eng_accept_high__gears(i) )

           Tq_eng_avrg          = interp2( Tq_eng_Dt_res_axis, ...
                                           w_eng_Dt_res_axis, ...
                                           Tq_eng_avrg_Dt_res_matrix, ...
                                           Tq_eng_dem__driver(i), ...
                                           w_eng_pred, ...
                                           'linear' ...
                                          );
```

```
Dw                      = w_eng_accept_high__gears(i) - w_eng_pred;

w_eng_avrg              = (w_eng_pred + w_eng_accept_high__gears(i)) / 2;

Tq_gb_in_avrg           = Tq_eng_avrg - ...
                             Engine_AuxPowerDraw / w_eng_avrg;

Tq_gb_ls_avrg           = interp3( InterpolationGear_TqIn,...
                                   InterpolationGear_wIn,...
                                   Gearbox_GearNumber,...
                                   InterpolationAllGear_TqInputLossMatrix,...
                                   Tq_gb_in_avrg,...
                                   w_eng_avrg,...
                                   i, ...
                                   'linear' ...
                                  );

Tq_gb_out_avrg          = ( Tq_gb_in_avrg - Tq_gb_ls_avrg ) * ...
                          Gearbox_GearRatio(i) ;

w_card_avrg             = w_eng_avrg / Gearbox_GearRatio(i) ;

w_ret_avrg              = w_card_avrg * Retarder_GearRatio;

Tq_ret_ls_avrg          = interp1( InterpolationRetarder_wIn, ...
                                   InterpolationRetarder_TqInputLossVector, ...
                                   w_ret_avrg, ...
                                   'linear' ...
                                  );

Tq_card_avrg            = Tq_gb_out_avrg - ...
                             Tq_ret_ls_avrg * Retarder_GearRatio;

Tq_ax_ls_avrg           = interp2( InterpolationAxlegear_TqIn, ...
                                   InterpolationAxlegear_wIn, ...
                                   InterpolationAxlegear_TqInputLossMatrix, ...
                                   Tq_card_avrg, ...
                                   w_card_avrg, ...
                                   'linear' ...
                                  );

Tq_wh_avrg              = ( Tq_card_avrg - Tq_ax_ls_avrg ) * ...
                          Axlegear_GearRatio ;

F_wh_avrg               = Tq_wh_avrg / Wheel_Radius;

F_accel_avrg            = F_wh_avrg - F_wh_upper_const ;

a_avrg                  = F_accel_avrg / m_inert_all(i);

w_dot_avrg              = ( a_avrg / Wheel_Radius ) * ...
                          Axlegear_GearRatio * ...
                          Gearbox_GearRatio(i) ;

w_eng_pred_upper(i)     = w_eng_pred;
Dw_upper(i)             = Dw;
a_upper(i)              = a_avrg;
w_dot_upper(i)          = w_dot_avrg;
```

```
            if w_dot_avrg > 0

                Dt_res_upper(i)    = Dw / w_dot_avrg;

            else
            end

        else
        end

    else
    end

%     stopper_1 = 1;


end

% stopper_2 = 1;

end
```


Engine_IdleSpeed
Engine_IdleSpeed
4
v_veh_curr [m/s]
1-D T(u)
share idle low
share_idle_low
max(share_idle_low_y_axis)
share_idle_low_max
w_eng_Tq99H
w_eng_Tq99H
w_eng_accept_high_min
w_eng_accept_high_min
w_eng_P99H
w_eng_P99H
Gearbox_GearNumber
Gearbox_GearNumber
Tq_card_max_x_axis
Tq_card_max_x_axis
Tq_card_max
Tq_card_max
rating_inaccept_speed
Ratio_Tq_card_dem_const_vect
Ratio_Tq_card_dem_const_vect
Ratio_w_eng_curr_max_vect
Ratio_w_eng_curr_max_vect
v_veh_sim
5
a_veh_sim [m/s^2]
signal
signal_past_avrg
a_veh_past_avrg
a_veh_past_avrg
1
w_eng_pred
w_eng_pred_gears
3
Tq_card__const [Nm]
Tq_card__const
2
current_gear [-]
current_gear
signal
signal_past_avrg
Ratio_w_eng_curr_max_past_avrg
Ratio_w_eng_curr_max_past_avrg
w_eng_accept_low
1
w_eng_accept_low
w_eng_accept_high
2
w_eng_accept_high
Ratio_w_eng_curr_max
1
z
Unit Delay
Speed rating

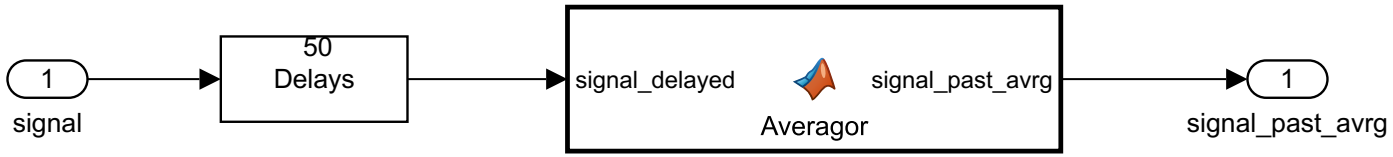

1
signal
50
Delays
signal_delayed
signal_past_avrg
Averagor
1
signal_past_avrg

```
function signal_past_avrg = Averagor(signal_delayed)
signal_past_avrg = mean(signal_delayed);
end
```

```
function [ w_eng_accept_low, ...
           w_eng_accept_high, ...
           Ratio_w_eng_curr_max ...
         ]...
    ...
    = rating_inaccept_speed( Engine_IdleSpeed, ...
                             share_idle_low, ...
                             share_idle_low_max, ...
                             w_eng_Tq99H, ...
                             w_eng_accept_high_min, ...
                             w_eng_P99H, ...
                             Gearbox_GearNumber, ...
                             Tq_card_max_x_axis, ...
                             Tq_card_max, ...
                             Ratio_Tq_card_dem_const_vect, ...
                             Ratio_w_eng_curr_max_vect, ...
                             v_veh_sim, ...
                             a_veh_past_avrg, ...
                             w_eng_pred_gears, ...
                             Tq_card__const, ...
                             current_gear, ...
                             Ratio_w_eng_curr_max_past_avrg ...
                            )

no_gears                       = length(Gearbox_GearNumber);

w_eng_accept_low               = zeros(no_gears, 1);

w_eng_accept_high              = zeros(no_gears, 1);

Ratio_w_eng_curr_max           = 0;



%% Set speed limits for current gear

i = current_gear;

if v_veh_sim > 0

    w_eng_accept_low(i)       = Engine_IdleSpeed + ...
                                     share_idle_low * ...
                                     ( w_eng_P99H - ...
                                       Engine_IdleSpeed ...
                                      ) ;

    Tq_card_max_curr          = interp2( Gearbox_GearNumber, ...
                                         Tq_card_max_x_axis, ...
                                         Tq_card_max, ...
                                         i, ...
                                         max( w_eng_pred_gears(i), ...
                                              Engine_IdleSpeed         ) ...
                                        );


    Tq_card__const_lim_min    = min(Ratio_Tq_card_dem_const_vect) * ...
                                  ( Tq_card_max_curr + 1);

    Tq_card__const_lim_max    = max(Ratio_Tq_card_dem_const_vect) * ...
```

```
                                ( Tq_card_max_curr - 1);

    Tq_card__const_lim        = max( Tq_card__const_lim_min, ...
                                     min( Tq_card__const_lim_max, ...
                                          Tq_card__const...
                                        )...
                                    );

    Ratio_Tq_card_dem_const   = Tq_card__const_lim / Tq_card_max_curr;

    Ratio_w_eng_curr_max      = interp1( Ratio_Tq_card_dem_const_vect, ...
                                         Ratio_w_eng_curr_max_vect, ...
                                         Ratio_Tq_card_dem_const ...
                                       );

    w_eng_accept_high(i) = ...
       ...
       w_eng_accept_high_min + ...
       ...
       Ratio_w_eng_curr_max * (w_eng_P99H - w_eng_accept_high_min);


else

    w_eng_accept_low(i)       = Engine_IdleSpeed + ...
                                    share_idle_low_max * ...
                                    ( w_eng_P99H - ...
                                      Engine_IdleSpeed ...
                                     ) ;

    w_eng_accept_high(i)      = 1.05 * w_eng_Tq99H;

end


%% Set speed limits for lower gears

for i = 1 : 1 : no_gears

     if i < current_gear

        if v_veh_sim > 0

            w_eng_accept_low(i)       = Engine_IdleSpeed + ...
                                            share_idle_low * ...
                                            ( w_eng_P99H - ...
                                              Engine_IdleSpeed ...
                                             ) ;

        else

            w_eng_accept_low(i)       = Engine_IdleSpeed + ...
                                            share_idle_low_max * ...
                                            ( w_eng_P99H - ...
                                              Engine_IdleSpeed ...
                                             ) ;

        end
```

```
        if  ( w_eng_pred_gears(current_gear) < w_eng_Tq99H  ) && ...
            ( a_veh_past_avrg                < 0           )

            w_eng_pred_curr = w_eng_pred_gears(current_gear);

            Dw_high         = 1.2 * w_eng_P99H - w_eng_Tq99H;

            w_eng_accept_high(i) = ...
                ...
                w_eng_Tq99H  + Dw_high * ( ( w_eng_Tq99H - w_eng_pred_curr   ) / ...
                                           ( w_eng_Tq99H - Engine_IdleSpeed  ) .... 
                                         ) ...
                                     * Ratio_w_eng_curr_max_past_avrg ;

        else

        end



%% Set speed limits for upper gears

    elseif i > current_gear

        if v_veh_sim > 0

            w_eng_accept_low(i)       = Engine_IdleSpeed + ...
                                        share_idle_low * ...
                                        ( w_eng_P99H - ...
                                          Engine_IdleSpeed ...
                                         ) ;

        else

            w_eng_accept_low(i)       = Engine_IdleSpeed + ...
                                        share_idle_low_max * ...
                                        ( w_eng_P99H - ...
                                          Engine_IdleSpeed ...
                                         ) ;

        end


        w_eng_accept_high(i) = w_eng_accept_high_min;

    end

end

% stop = 1;

end
```


1
signal
50
Delays
signal_delayed
signal_past_avrg
Averagor
1
signal_past_avrg

```
function signal_past_avrg = Averagor(signal_delayed)
signal_past_avrg = mean(signal_delayed);
end
```

```
function [P_card_avrg_past, P_card_curr, requestedGear, OK2Shift] ...
    = ShiftDecision( Tq_card_past, ...
                     v_veh_past, ...
                     w_eng_curr, ...
                     w_eng_predicted, ...
                     currentTime, ...
                     rating_gear, ...
                     gearNumbers, ...
                     grad_max_shareTq99L, ...
                     w_eng_accept_low, ...
                     w_eng_accept_high, ...
                     grad_upper_pred, ...
                     Numbers_0_D, ...
                     currentGear, ...
                     drive_off ...
                    )


% Variables who maintain their values from the last loop
persistent t_lastUpshift ...
           t_lastDownshift ...
           requestedGearOld ...
           t_lastShift ...
           t_last_drive_off

% Upon the first call we need to initialize the variables.
if isempty(t_lastUpshift)

    t_lastUpshift      = -100;
    t_lastDownshift    = -100;
    requestedGearOld   =    1;
    t_lastShift        = -100;
    t_last_drive_off   = -100;

end

% Numbers_0_D
DelayDownUp                = Numbers_0_D(1);
DelayUpDown                = Numbers_0_D(2);
DelayShift                 = Numbers_0_D(3);
allowedGearRangeUp         = Numbers_0_D(4);
allowedGearRangeDown       = Numbers_0_D(5);
GearChangeTime             = Numbers_0_D(6);
Wheel_Radius               = Numbers_0_D(7);
Axlegear_GearRatio         = Numbers_0_D(8);
P_card_avrg_past_pos       = Numbers_0_D(9);
P_card_curr_pos            = Numbers_0_D(10);
w_eng_Tq99L                = Numbers_0_D(11);
w_eng_Tq99H                = Numbers_0_D(12);
w_eng_P99H                 = Numbers_0_D(13);

% Number of gears
no_gears               = max(gearNumbers);

assert(no_gears <= 16);

max_gear_drive_off      = round( no_gears / 2);
```

```
%%
% Determine time of last shift
t_lastShift = max([t_lastShift, t_lastDownshift, t_lastUpshift, t_last_drive_off]);

% If the model is in the first timestep, where the clutch started to close
if drive_off == 1

    allowedGearRangeUp   = 20;
    allowedGearRangeDown = 20;

end

%% Allowed gears

allowedGears = zeros(no_gears, 1);

for i = 1 : 1 : no_gears

    if i <= currentGear

        if ( i                      >= (currentGear - allowedGearRangeDown) )

            allowedGears(i) = i;

        else
        end

    else

        if ( i                      <= (currentGear + allowedGearRangeUp)   ) && ...
           ( grad_max_shareTq99L(i) >= grad_upper_pred                      ) && ...
           ( w_eng_predicted(i)     >= w_eng_accept_low(i)                  )

            allowedGears(i) = i;

        else
        end

    end

end

%% Minimum time to wait before shift has passed and the clutch is closed
OKToShift           = @(currentTime) ( ( t_lastShift + DelayShift <  currentTime ) || ...
                                       ( drive_off                == 1           ) ...
                                     );

OKToShiftUp         = @(currentTime) ( t_lastDownshift + DelayDownUp < currentTime );

OKToShiftDown       = @(currentTime) ( t_lastUpshift   + DelayUpDown < currentTime );



% If we do not change gear we use the old one
requestedGear = requestedGearOld;
```

```
% Check if we should shift gears, else we use the current one
OK2Shift = OKToShift(currentTime);



%%
% Calculate vector of past power at cardan shaft during look-back period.
P_card_past = ( v_veh_past / Wheel_Radius * Axlegear_GearRatio) .* ...
              Tq_card_past / 1000;                                          % [kW]

% Average of past power at cardan shaft during look-back period.
P_card_avrg_past = mean(P_card_past);                                       % [kW]

% Get entries for current timestep from vectors of look-back periods
v_veh_curr      = v_veh_past(1);                                            % [m/s]
Tq_card_curr    = Tq_card_past(1);                                          % [Nm]

% Calculate current power demand at cardan shaft
P_card_curr     = ( v_veh_curr / Wheel_Radius * Axlegear_GearRatio) * ...
                  Tq_card_curr / 1000;                                      % [kW]



%%
% State "propulsion", if velocity during look-back period was 0
% -> drive-off shortly before
if min(v_veh_past) == 0

    propulsion = 1;

% State "propulsion", if /average/ power at cardan shaft during look-back period
% is above its threshold, or /current/ power cardan shaft is above its
% threshold.
% => Avoid oscillations of gearshift control during coasting,
% but set to propulsion, if power is currently demanded

elseif ( P_card_avrg_past > P_card_avrg_past_pos ) || ...
       ( P_card_curr      > P_card_curr_pos      )

    propulsion = 1;

else

    propulsion = 0;

end


if OK2Shift

%**********************************************************************
% Select gear based on rating. If not, then we will select gear based on
% the resulting engine speed.
%**********************************************************************



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case propulsion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    if propulsion == 1

        % Select the gear with the lowest rating

        rating_min   = 1.0e6;
        chosenGear   = 1;
        choice_valid = 0;

        for i = 1 : 1 : no_gears

            if allowedGears(i) ~= 0

                if ( rating_gear(i) <  rating_min )

                    chosenGear   = i;

                    rating_min   = rating_gear(i);

                    choice_valid = 1;

                else
                end



            else
            end

        end


        % Reset chosenGear, if no gear was valid

        if choice_valid == 0

            chosenGear = requestedGearOld;

        else
        end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case propulsion upshift
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        % Shift up if the chosen gear is above current gear AND if enought time has
        % passed since the last upshift OR If current engine speed is too high

        if  ( ( ( chosenGear               >  currentGear                    ) && ...
                ( OKToShiftUp(currentTime)                                   ) ...
              ) || ...
              (   w_eng_curr               >= w_eng_accept_high(currentGear) ) ...
             ) && ...
            ( (   w_eng_curr               >= w_eng_accept_low(currentGear)  ) ...
             )

            requestedGear = chosenGear;
```

```
            t_lastUpshift = currentTime + GearChangeTime;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case propulsion downshift
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        % Shift down if the chosen gear is below current gear AND if enought time has
        % passed since the last downshift OR If current engine speed is too low
        % Downshifts after a drive-off become permitted, if one upshift
        % happened in between.

        elseif ( ( ( chosenGear < currentGear                                    ) && ...
                   ( OKToShiftDown(currentTime)                                  ) ...
                 ) || ...
                 ( ( w_eng_curr                    <= w_eng_accept_low(currentGear) ) ...
                 ) ...
                ) && ...
                (    t_lastUpshift > 0                                           )

            requestedGear = chosenGear;

            t_lastDownshift = currentTime + GearChangeTime;

        end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case coasting / braking downshift
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    %************************************************************
    % The engine provides no power and we are slowing down and should
    % select a lower gear to maintain a good engine speed.
    %************************************************************

    elseif ( w_eng_predicted(currentGear) <= w_eng_accept_low(currentGear) )

        % We are slowing down without propulsion.
        % So we select the possible /lowest/ gear, where
        % the engine speed is above the accepted minimum

        requestedGear = 0;

        for i = 1 : 1 : no_gears

            if ( requestedGear      == 0                                      ) &&...
               ( i                  >= (currentGear - allowedGearRangeDown)   ) &&...
               ( w_eng_predicted(i) >  w_eng_accept_low(i)                    ) &&...
               ( w_eng_predicted(i) <  (w_eng_Tq99L + w_eng_Tq99H) / 2        )

               requestedGear = i;

               t_lastDownshift = currentTime + GearChangeTime;
```

```
            end

        end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case coasting / braking upshift
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    %************************************************************
    % The engine provides no power and we are speeding up and should
    % select a higher gear to maintain a good engine speed.
    %************************************************************

    elseif ( w_eng_predicted(currentGear) >= ( w_eng_Tq99H + w_eng_P99H ) / 2 )

        % We are speeding up without propulsion.
        % So we select the possible /highest/ gear, where
        % the engine speed is below the accepted maximum

        requestedGear = 0;

        for i = 1 : 1 : no_gears

            gear_backw = (-i + 1) + no_gears;

            if ( requestedGear               == 0                                    ) &&...
               ( gear_backw                  <= ( currentGear + allowedGearRangeUp ) ) &&...
               ( w_eng_predicted(gear_backw) >  w_eng_accept_low(gear_backw)         ) &&...
               ( w_eng_predicted(gear_backw) <  ( w_eng_Tq99H + w_eng_P99H ) / 2     )

               requestedGear = i;

               t_lastUpshift = currentTime + GearChangeTime;

            end

        end

    end

end



%% Re-set the gear to the old one, if in one of the cases coasting/braking
%  down- or upshift no gear was chosen (requestedGear there set to 0, and
%  not changed afterwards

if requestedGear == 0

    requestedGear = requestedGearOld;

else
end
```

```
%% Drive off gear


if drive_off == 1

    % Select the gear with the lowest rating

    rating_min = 1.0e6;
    chosenGear = 1;

    for i = 1 : 1 : max_gear_drive_off

        if allowedGears(i) ~= 0

            if ( rating_gear(i) <  rating_min )

                chosenGear = i;
                rating_min = rating_gear(i);

            else
            end

        else
        end

    end

    requestedGear = chosenGear;

    t_last_drive_off = currentTime + GearChangeTime;

    t_lastShift      = currentTime + GearChangeTime;

    t_lastUpshift    = 0;

end


%% If nothing happened, keep the gear constant


requestedGearOld = requestedGear;


%%

% stopper = 1;


end
```