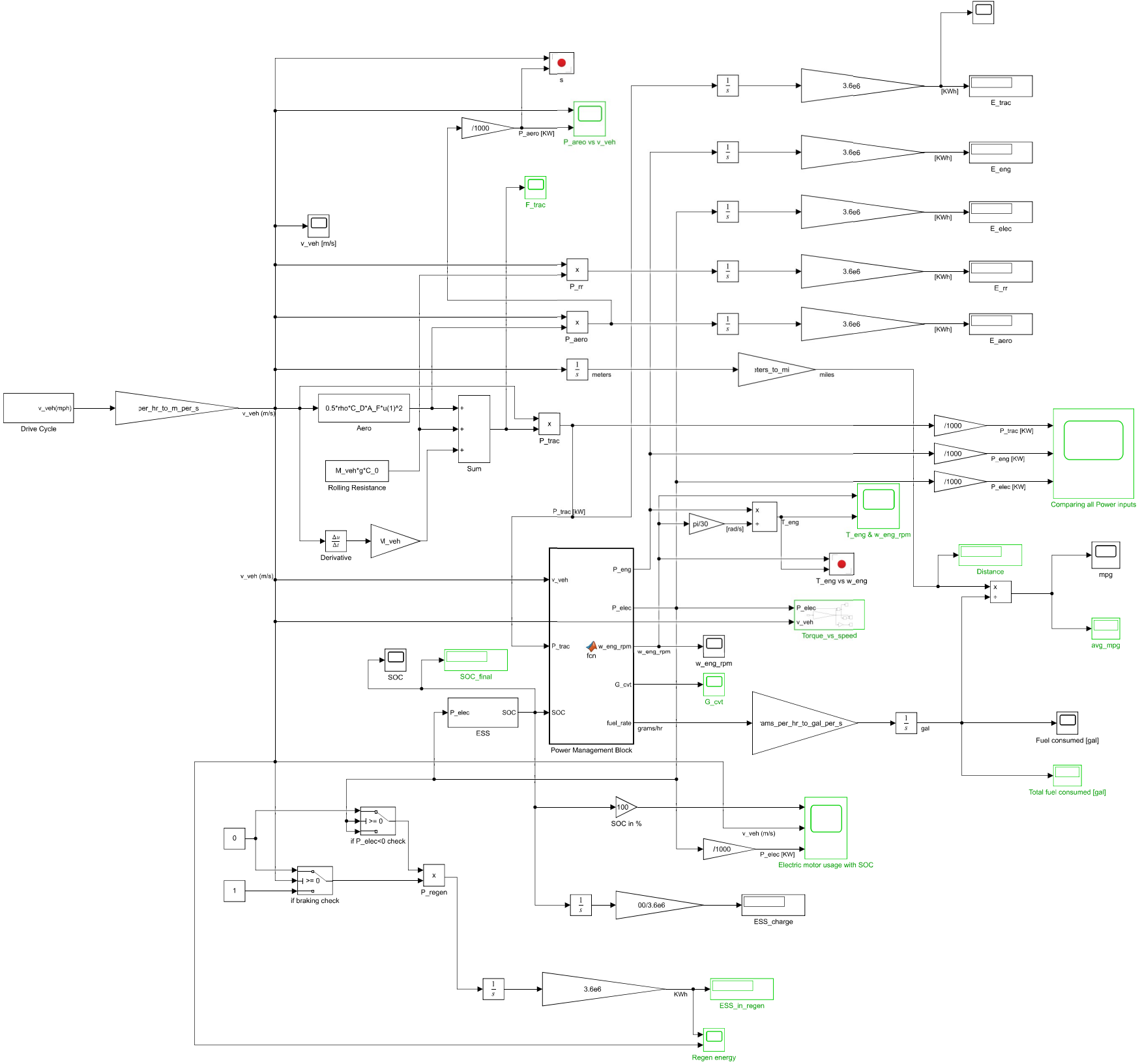
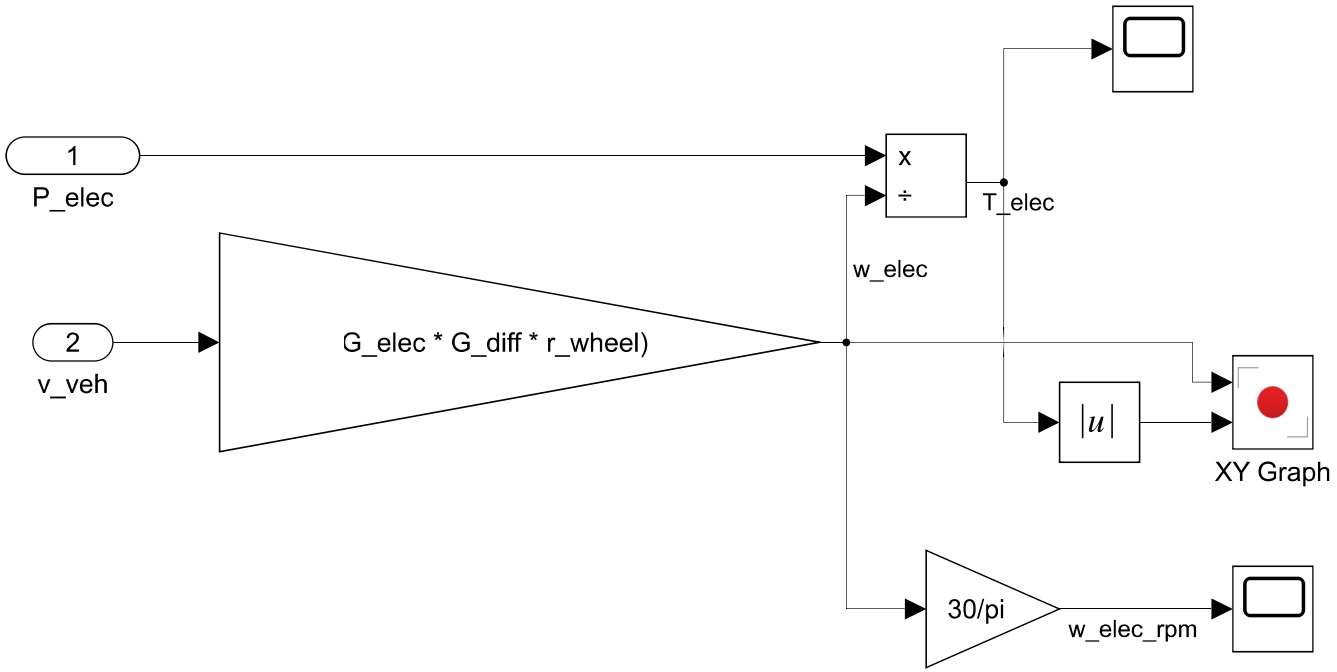

v_veh(mph)
Drive Cycle
per_hr_to_m_per_s
v_veh (m/s)
0.5*rho*C_D*A_F*u(1)^2
Aero
M_veh*g*C_0
Rolling Resistance
Sum
P_trac
P_trac [kW]
Derivative
M_veh
v_veh (m/s)
s
/1000
P_aero [KW]
P_areo vs v_veh
F_trac
v_veh [m/s]
P_rr
P_aero
meters
meters_to_mi
miles
1/s
3.6e6
[KWh]
E_trac
E_eng
E_elec
E_rr
E_aero
/1000
P_trac [KW]
P_eng [KW]
P_elec [KW]
Comparing all Power inputs
pi/30
[rad/s]
T_eng
T_eng & w_eng_rpm
T_eng vs w_eng
Distance
mpg
avg_mpg
P_eng
P_elec
v_veh
P_trac
w_eng_rpm
fcn
G_cvt
fuel_rate
grams/hr
Power Management Block
Torque_vs_speed
SOC
SOC_final
w_eng_rpm
G_cvt
P_elec
SOC
ESS
grams_per_hr_to_gal_per_s
gal
Fuel consumed [gal]
Total fuel consumed [gal]
0
1
>= 0
if P_elec<0 check
if braking check
P_regen
100
SOC in %
v_veh (m/s)
/1000
P_elec [KW]
Electric motor usage with SOC
00/3.6e6
ESS_charge
KWh
ESS_in_regen
Regen energy

```
function [P_eng, P_elec, w_eng_rpm, G_cvt, fuel_rate] = fcn(v_veh, P_trac, SOC, param, eng_map)
%#codegen
%inputs:
    % v_veh, m/s
    % P_trac, W
    % SOC
    % param, structure of parameters
    % eng_map

% outputs:
    %   P_gen, P_elec in W
    %   w_eng_rpm, rpm
    %   G_cvt (cvt ratio)
    %   Fuel rate, grams/hr

v_veh_min = param.v_veh_min;    % minimum vehicle speed for engine to stay engaged, in m/s
P_eng_min = param.P_eng_min;    % minimum engine power in W
P_eng_max = param.P_eng_max ;   % maximum engine power
G_cvt_min = param.G_cvt_min;    % minimum cvt  ratio
G_diff = param.G_diff;          % differential gear ratio
r_wheel = param.r_wheel;        % wheel radius in m

if (v_veh < v_veh_min)  % disengage clutch, idle engine, electric propulsion
   P_elec = P_trac;
   P_eng = 0;
   fuel_rate = 0; % g/hr;
   w_eng_rpm = 1000; % rpm
   G_cvt = G_cvt_min;
   return
end

% if here, v_veh > v_veh_min
if(P_trac < P_eng_min)   % clutch engaged but engine idling
   P_elec = P_trac;
   fuel_rate = 0;
   P_eng = 0;
   w_eng_rpm =  1000; % rpm
   w_eng = w_eng_rpm * pi / 30; % rad/s
   % set G_cvt so engine speed is 1000 rpm
   G_cvt = v_veh/G_diff/w_eng/r_wheel;
   return
end

if(P_trac > P_eng_max) % high-speed boost
   P_elec = P_trac - P_eng_max;
   P_eng = P_eng_max;
   bsfc = interp1(eng_map(:,2), eng_map(:,3), P_eng/1000, 'pchip', 'extrap');
   fuel_rate = bsfc*P_eng/1000; % grams/hr
   w_eng_rpm = interp1(eng_map(:,2), eng_map(:,1), P_eng/1000, 'pchip', 'extrap');
   w_eng = w_eng_rpm * pi / 30; % convert to rad/s
   G_cvt =  v_veh/r_wheel/G_diff/w_eng; % required CVT ratio
   return
end

% if here, v_veh > v_veh_min and P_eng_min < P_trac < P_eng_max
% try to get SOC back to 0.5

%P_elec = 4000*sign(SOC - 0.5);
P_elec = 20000*(SOC - 0.5);
if(P_elec > 4000)
    P_elec = 4000;
```

```
end
if(P_elec < -4000)
    P_elec = -4000;
end

P_eng = P_trac - P_elec;


if(P_eng < P_eng_min)
   % clutch engaged, but no fuel
   P_eng = 0;
   P_elec = P_trac;
   fuel_rate = 0;
   w_eng_rpm = 1000;
   w_eng = w_eng_rpm * 2 * pi / 60; % in rad/s
   % set G_cvt so engine speed is 1000 rpm
   G_cvt = v_veh/G_diff/w_eng/r_wheel;
   return
end

if(P_eng > P_eng_max)
    P_eng = P_eng_max;
    P_elec = P_trac - P_eng_max;
end

bsfc = interp1(eng_map(:,2), eng_map(:,3), P_eng/1000, 'pchip', 'extrap');
fuel_rate = bsfc*P_eng/1000; % grams/hr

w_eng_rpm = interp1(eng_map(:,2), eng_map(:,1), P_eng/1000, 'pchip', 'extrap');
if(w_eng_rpm < 1000)
    w_eng_rpm = 1000;
end
w_eng = w_eng_rpm * pi / 30;  % convert to rad/s
G_cvt =  v_veh/r_wheel/G_diff/w_eng;

if (G_cvt < G_cvt_min) % set G_cvt = G_cvt_min, recalculate w_eng, P_eng, and P_elec
   G_cvt = G_cvt_min;
   w_eng = v_veh/G_diff/r_wheel/G_cvt;
   w_eng_rpm = w_eng*30/pi;  % in rpm
   P_eng = 1000*interp1(eng_map(:,1), eng_map(:,2), w_eng_rpm, 'pchip', 'extrap'); % in W
   if(P_eng < 0)
       P_eng = 0;
   end
   P_elec = P_trac - P_eng;
   bsfc = interp1(eng_map(:,1), eng_map(:,3), w_eng_rpm, 'pchip', 'extrap');
   fuel_rate = bsfc*P_eng/1000; % grams/hr
 end
```


1
P_elec
2
v_veh
G_elec * G_diff * r_wheel)
x
÷
T_elec
w_elec
|u|
XY Graph
30/pi
w_elec_rpm