CIVE 295 Energy Systems and Control
Spring 2019 Scott Moura and Dong Zhang **HW 04**

1 Problem 1

(a) We know that $J = \sum_{k=0}^{N-1} \alpha\Delta t * P_{fuel}(k)$ and

$P_{eng}(k) = P_{dem}(k) - P_{batt}(k)$ and

$P_{eng}(k) = \eta(P_{eng}(k))P_{batt}(k)$, we can replace $P_{fuel}(k)$ in the cost function with P_{dem} and P_{batt} giving us the objective function

$min_{P_{batt}(k),SOC(k)} J = \sum_{k=0}^{N-1} \alpha\Delta t * \frac{\eta(P_{dem}(k)-P_{batt}(k))}{(P_{dem}(k)-P_{batt}(k)}$

(b) The objective function in part a is minimized to

Equality Constraints: $SOC(k+1) = SOC(k) - \frac{\Delta t}{Q_{cap}V_{oc}}$ This references to the SOC of the battery at the next time step.

Inequality constraints: $SOC^{min} \leq SOC(k) \leq SOC^{max}$ this refers to the min and max SOC

$-P_{batt}^{max} \leq P_{batt}(k) \leq P_{batt}^{max}$ referring to the min and max power being extracted from the battery

$0 \leq P_{dem}(k) - P_{batt}(k) \leq P_{end}^{max}$ giving us another condition on the power being extracted from the battery.

(c) State variable is SOC(k) and control variable is $\mathrm{P}_{batt}(k)$

2 Problem 2

(a) $V(SOC(k)) = \min_{P_{batt}(k),SOC(k)} \alpha\Delta t\frac{\eta(P_{dem}(k)-P_{batt}(k))}{(P_{dem}(k)-P_{batt}(k)} + V(SOC(k+1))$

$V_N(k) = 0$

3 Problem 3

(a) $SOC^{min} \leq SOC(k+1) \leq SOC^{max}$ Using our known equations from Q1.

$[-SOC^{max} + SOC(k)]\frac{Q_{cap}V_{oc}}{\Delta t} \leq P_{batt}(k) \leq [-SOC^{min} + SOC(k)]\frac{Q_{cap}V_{oc}}{\Delta t}$

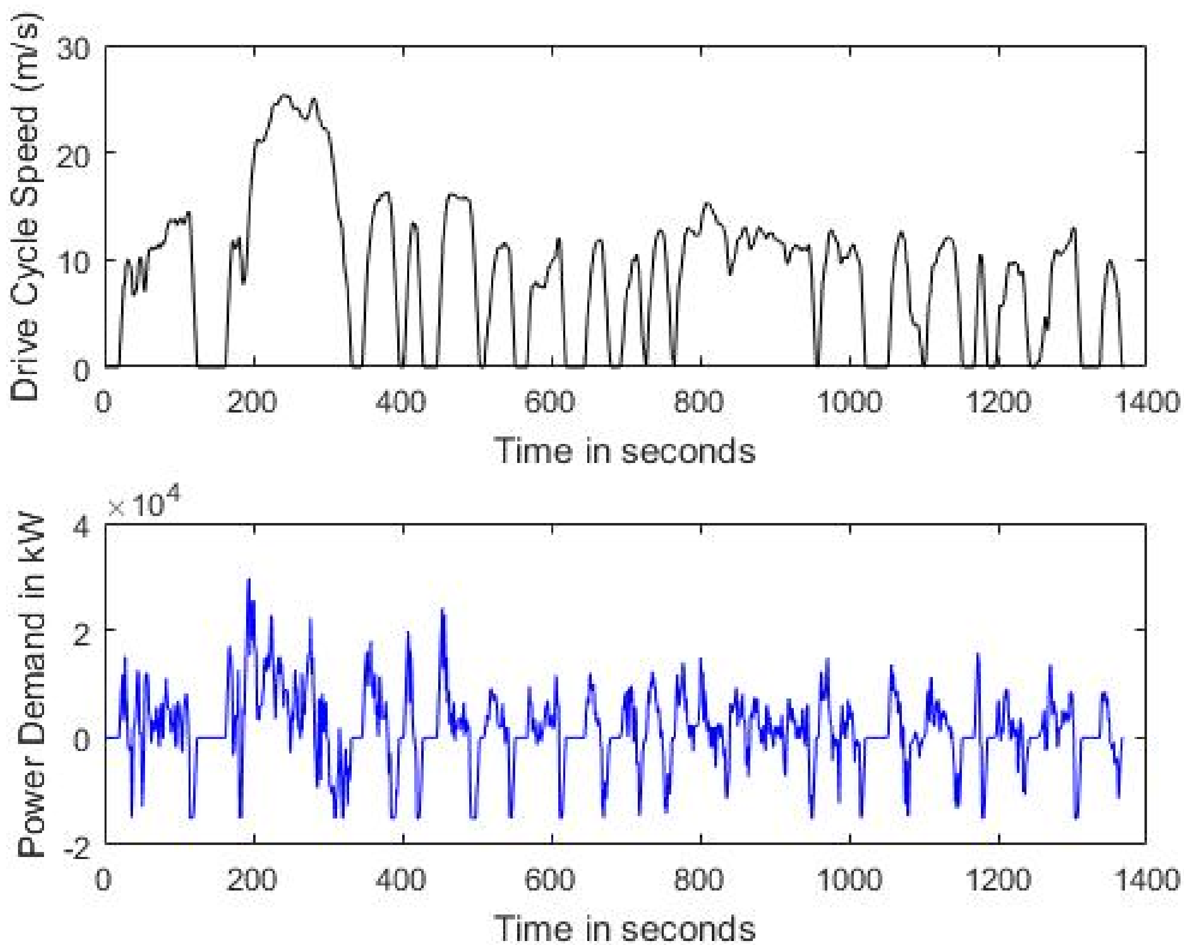
$-P_{batt}^{max} \leq P_{batt} \leq P_{batt}^{max}$

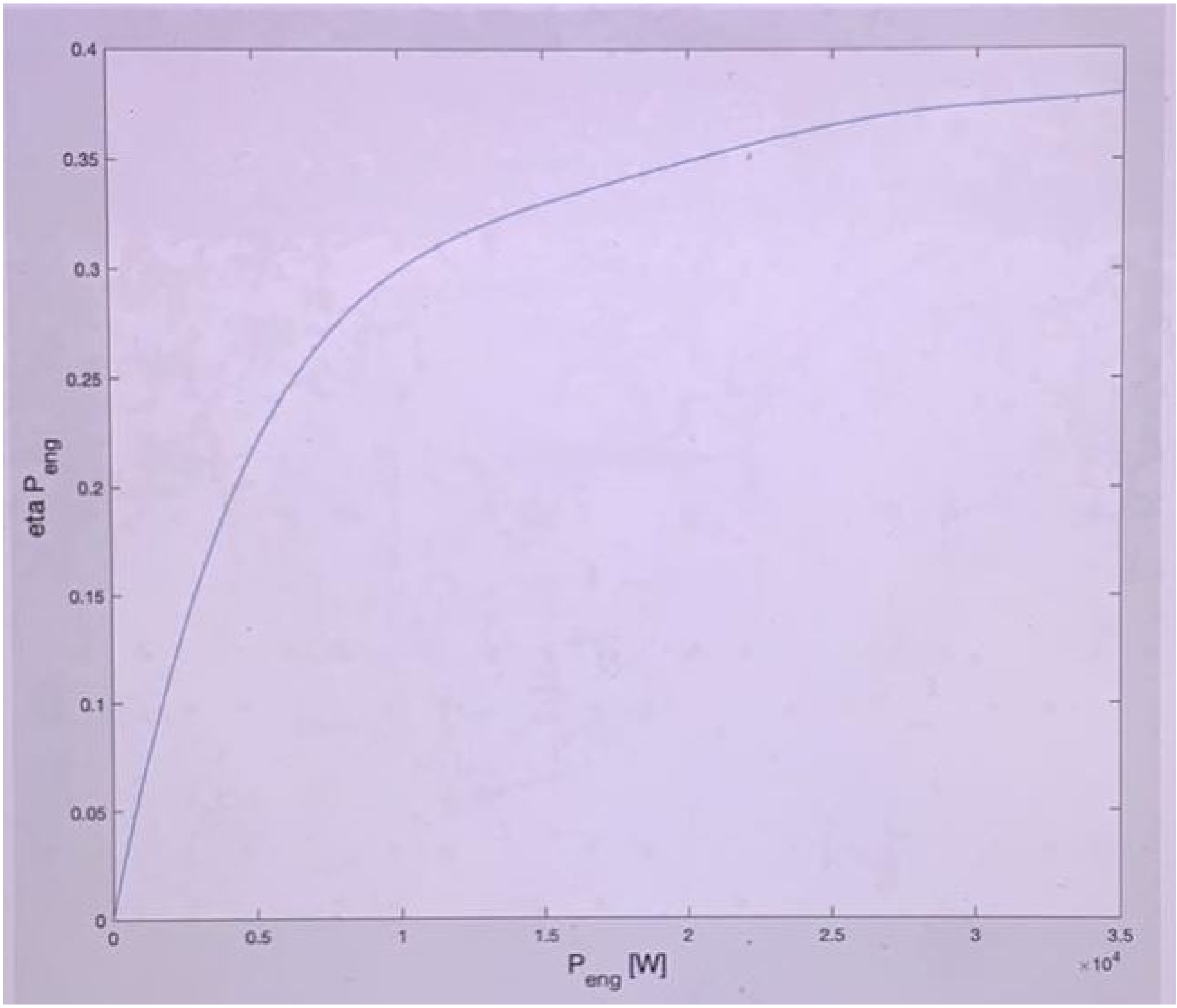
Rearranging the last constraint from q1, we get: $P_{dem}(k) - P_{eng}^{max} \leq P_{batt}(k) \leq P_{dem}(k)$

(b) $\max[(([-SOC^{max} + SOC(k)]\frac{Q_{cap}V_{oc}}{\Delta t}), (-P_{batt}^{max}), (P_{dem}(k) - P_{eng}^{max})] \leq P_{batt}(k) \leq \min[(([-SOC^{min} + SOC(k)]\frac{Q_{cap}V_{oc}}{\Delta t}), P_{dem}(k), P_{batt}^{max}]$

4 Problem 4

(a) Graph shown below:


Drive Cycle Speed (m/s)
Time in seconds
Power Demand in kW
$\times 10^4$
Time in seconds

(b) 
eta P_{eng}
P_{eng} [W]
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.5
1
1.5
2
2.5
3
3.5
$\times 10^4$

5 Problem 5

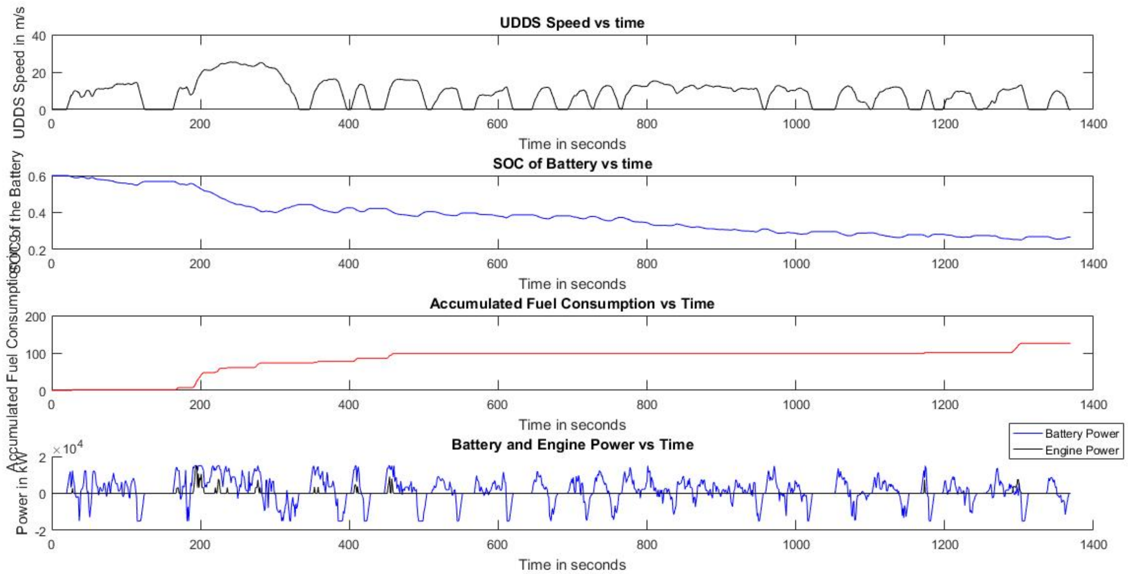
```
%% Solve DP
tic;
% Boundary Condition of Value Function (Principle of Optimality)
V(:,N+1) = 0;
% Iterate backward in time
for k = N:-1:1
    % Iterate over SOC
    for idx = 1:ns
        bound1 = ((-SOC_max + SOC_grid(idx))*Qcap*V_oc/Delta_t);
        bound2 = -P_batt_max;
        bound3 = P_dem(k) - P_eng_max;
        bound4 = ((-SOC_min + SOC_grid(idx))*Qcap*V_oc/Delta_t);
        bound5 = P_batt_max;
        bound6 = P_dem(k);
        % Find dominant bounds
        lb = max([bound1 bound2 bound3]);
        ub = min([bound4 bound5 bound6]);
        % Grid Battery Power between dominant bounds
        P_batt_grid = linspace(lb,ub,200)';
        % Compute engine power (vectorized for all P_batt_grid)
        P_eng = P_dem(k) - P_batt_grid;
        % Cost-per-time-step (vectorized for all P_batt_grid)
        g_k = alph*Delta_t*P_eng./eta_eng(P_eng);
        % Calculate next SOC (vectorized for all P_batt_grid)
        SOC_nxt = SOC_grid(idx)-Delta_t/Qcap/V_oc*P_batt_grid;
        % Compute value function at nxt time step (need to interpolate)
        V_nxt = interp1(SOC_grid,V(:,k+1),SOC_nxt,'linear');
        % Value Function (Principle of Optimality)
        [V(idx, k), ind] = min(g_k+V_nxt);
        % Save Optimal Control
        u_star(idx,k) = P_batt_grid(ind);
    end
end
solveTime = toc;
fprintf(1,'DP Solver Time %2.2f sec \n',solveTime);
```

6 Problem 6

(a)

```
%% Simulate Results

% Preallocate
SOC_sim = zeros(N,1);
P_batt_sim = zeros(N,1);
P_eng_sim = zeros(N,1);
J_sim = zeros(N,1);

% Initialize
SOC_0 = 0.75;
SOC_sim(1) = SOC_0;

% Simulate PHEV Dynamics
for k = 1:(N-1)

    % Use optimal battery power, for given SOC (need to interpolate)
    P_batt_sim(k) = interp1(SOC_grid, u_star(:,k), SOC_sim(k) );

    % Compute engine power
    P_eng_sim(k) = P_dem(k)-P_batt_sim(k);

    % Fuel Consumption
    J_sim(k) = alph*Delta_t*P_eng_sim(k)/eta_eng(P_eng_sim(k));

    % Time-step SOC dynamics
    SOC_sim(k+1) = SOC_sim(k) -Delta_t/Qcap/V_oc*P_batt_sim(k);

end

fprintf(1,'Total Fuel Consumption %2.2f kg \n',sum(J_sim)/1e3);
```

(b) Minimum fuel consumption is 0.9kg.



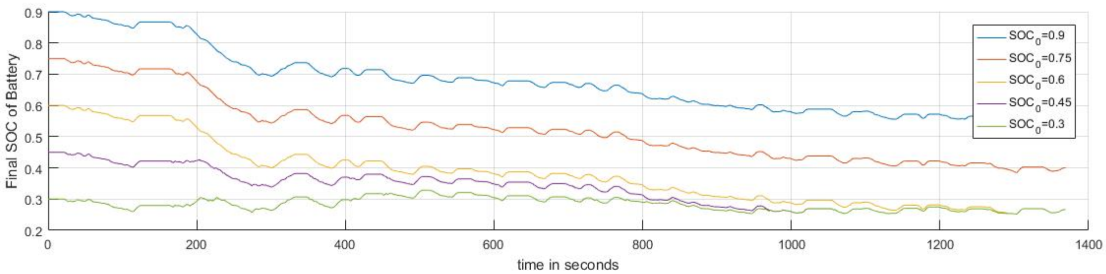
(c)

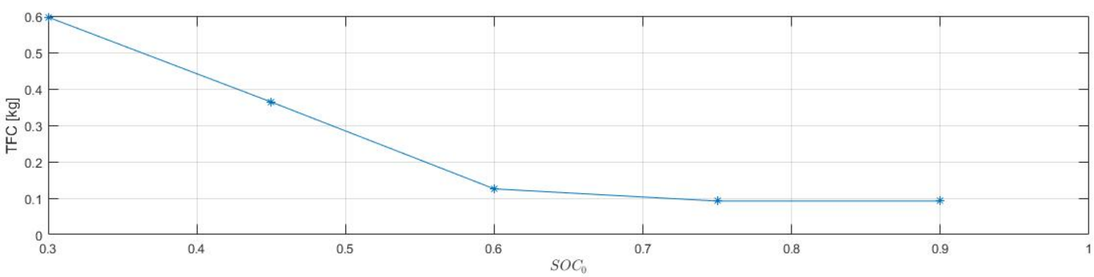
```
>> sum(SOC_sim>SOC_max | SOC_sim<SOC_min)

ans =

     0

>> sum(P_batt_sim>P_batt_max | P_batt_sim<-P_batt_max)

ans =

     0

>> sum(P_batt_sim>P_dem | P_batt_sim<P_dem-P_eng_max)

ans =

     0
```

(d)

(e) All constraints are satisfied

7 Problem 7





For Initial SOC of 0.9, final SOC is 0.5498 and Fuel Consumption is 0.0922 kg.
For Initial SOC of 0.75, final SOC is 0.3998 and Fuel Consumption is 0.0922 kg.
For Initial SOC of 0.6, final SOC is 0.2666 and Fuel Consumption is 0.1225 kg.
For Initial SOC of 0.45, final SOC is 0.2666 and Fuel Consumption is 0.3640 kg.
For Initial SOC of 0.3, final SOC is 0.2666 and Fuel Consumption is 0.5978 kg.

8 Problem 8

The battery SOC graph vs time shows that the SOC does overall decrease over time and decreases as the power from the battery is used. The Power graph shows us that this is the case when the power from the battery is very high and that's when Power from engine is used to balance the powers out.

The SOC of the cell when decreased initially tries to stabilize itself and the SOC trajectory tries to flatten out by trying to charge and discharge the same amount of time. In an effort to keep SOC stable the engine is used when the power demand is very high and this is where total fuel consumption comes in. The whole consumption depends on stabilizing the SOC and maintaining that in the battery.