Mohammadreza Arani

Convex Optimization

Project 3

810100511

Electric vehicle charging. A group of *N* electric vehicles need to charge their batteries over the next *T* time periods.

In each time period, the total charging energy over all vehicles cannot exceed *C*_max. [Our Constraint] (1)

q(t+1*,i)* = *q(t,i)* + *c(t,i*) [is defined]

Initial points are given for dynamic charges q1,i:

Target minimum charge level over time, given by *q(t,i)* tar ===>

s(t,i) = (*q(t,i)* tar - *q(t,i))*+ ====> charging shortfall in period*t* for vehicle *i* [is defined]

(*a*)+ = max*{a, 0}* [is defined]

Our *objective* is to minimize the mean square shortfall, given by :

$S = \frac{1}{(T+1)*N} \Sigma_t \Sigma_i s_{t,i}^2$: for t=1:T+1;

i=1:N;

[minimizing the root-mean-square (RMS) shortfall]

[is defiend]

Q1:

Explain how to solve the problem using convex optimization, and solve the following problem instance.

We have ***N* = 4** vehicles, ***T* = 90** time periods, and ***C*max = 3**.

The **initial charges *q(1,i)* are 20, 0, 30, and 25** , respectively.

$$q_{t,i}^{\mathrm{tar}} = \left(\frac{t}{T+1}\right)^{\gamma_i} q_i^{\mathrm{des}}, \quad t = 1, \ldots, T+1, \quad i = 1, \ldots, N,$$

```
% Load data and Initialization:
clear; clc; close all;
```

```
N = 4; % Num of cars
T = 90; % Period step
C_max = 3;

q_init = [20;0;30;25];  % Initial Point --->
q = zeros(T+1,N);
q(1,:) = q_init(:);

t= [1:T+1]';
q_des = [60;100;75;125]'; % gives the fnal value of the target minimum charge level for vehicle



Lambda = [0.5, 0.3, 2.0, 0.6]; %parameter γi sets the  urgency' of charging, with smaller value

q_target = (t./(T+1)).^Lambda.*q_des; % as defined in the figure above!
```

The total charging period is 7.5 hours, and the maximum charging of 3kWh/ period corresponds to a real power of 36 kW.

Demands:

1) Give the optimal RMS shortfall

2) Plot the target minimum charge values and optimal state of charge for each vehicle with dashed

lines showing the target and solid lines showing the optimal charge

3) Plot the optimal charging energies *c(t,i)* over time in a stack plot.

**** **Convexity of the problem** ****

In a convex problem, which can be solved using CVX package, we need to follow DCP rules and to have a feasible set not to get -inf or inf as the CVX output.

A convex problem, consists of a convex objective and some convex/affine constraints.

here the objective as introduced is:

$$J = \frac{1}{(T+1)N} \sum_{t=1}^{T+1} \sum_{i=1}^{N} s_{t,i}^2$$

s is a single element which is the difference between target and current value. ==> Actually maximum (sup()) of convex functions is convex itself!

we may show S in a quadratic form to prove its convexity!

```
% First, we defined the cost function to be as:
% sum_square( pos( q_hat(t,i) - ones(1,t-1)*c(1:t-1, i)  ) )
```

```
% And we faced MATLAB error!
% then we changed it with  x*max(x,0) for x = q_hat(t,i) -
% ones(1,t-1)*c(1:t-1, i) --> Again we faced MATLAB Error!
% Finally we removed pos function and got a good result but not the correct
% one!
diff = q_target - q_init';
cvx_begin
```

```
Warning: A non-empty cvx problem already exists in this scope.
   It is being overwritten.
```

```
    variables c(T, N);
% q(t,i) = q(1,i) + sum( c(t',i) , t' = 1  to t-1)
    for t = 1 : T+1
        for i = 1 : N
            s_squared(t, i) =  pow_pos( diff(t,i) - ones(1,t-1)*c(1:t-1, i) , 2); % The objecti
        end
    end

    minimize( sum( sum(s_squared) )/ ((T+1)*N)    ) ; % / ((T+1)*N) ---> not effective in optir

    subject to

        c(:) >= 0; % Implicit COnstraint
        sum(c, 2) <= C_max; % Our Only Constraint ---sum(x,2) --> on the columns --> here is th


cvx_end
```

```
Calling SDPT3 4.0: 1890 variables, 810 equality constraints
------------------------------------------------------------

 num. of constraints = 810
 dim. of sdp    var  = 720,   num. of sdp  blk  = 360
 dim. of linear var  = 810
*****************************************************************
   SDPT3: Infeasible path-following algorithms
*****************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|3.7e+02|8.1e+02|1.0e+07| 1.997802e+02  0.000000e+00| 0:0:00| spchol  1  1
 1|0.558|0.774|1.6e+02|1.8e+02|4.6e+06| 3.498047e+02 -1.430102e+04| 0:0:00| spchol  1  1
 2|0.537|0.746|7.6e+01|4.7e+01|2.5e+06| 7.479596e+02 -2.106065e+04| 0:0:00| spchol  1  1
 3|0.588|0.735|3.1e+01|1.2e+01|1.3e+06| 1.733105e+03 -2.534799e+04| 0:0:00| spchol  1  1
 4|0.727|0.725|8.6e+00|3.5e+00|4.5e+05| 4.007377e+03 -2.958055e+04| 0:0:00| spchol  1  1
 5|0.766|0.805|2.0e+00|7.0e-01|1.6e+05| 7.949670e+03 -3.471742e+04| 0:0:00| spchol  1  1
 6|1.000|1.000|2.3e-08|1.6e-02|2.8e+04| 8.141005e+03 -1.885439e+04| 0:0:00| spchol  1  1
 7|0.888|0.895|7.0e-09|8.6e-03|5.4e+03| 2.050374e+03 -3.135480e+03| 0:0:00| spchol  1  1
 8|1.000|1.000|6.7e-10|3.9e-03|2.5e+03| 1.068220e+03 -1.423742e+03| 0:0:00| spchol  1  1
 9|0.912|0.912|1.5e-10|2.1e-03|3.6e+02| 2.468028e+02 -1.043516e+02| 0:0:00| spchol  1  1
10|1.000|0.995|3.1e-15|9.8e-04|1.6e+02| 1.499151e+02 -1.071063e+01| 0:0:00| spchol  1  1
11|1.000|1.000|2.2e-15|2.9e-04|4.9e+01| 1.008725e+02  5.231997e+01| 0:0:00| spchol  1  1
12|0.925|0.920|5.4e-15|1.0e-04|7.8e+00| 8.071716e+01  7.301528e+01| 0:0:01| spchol  1  1
```

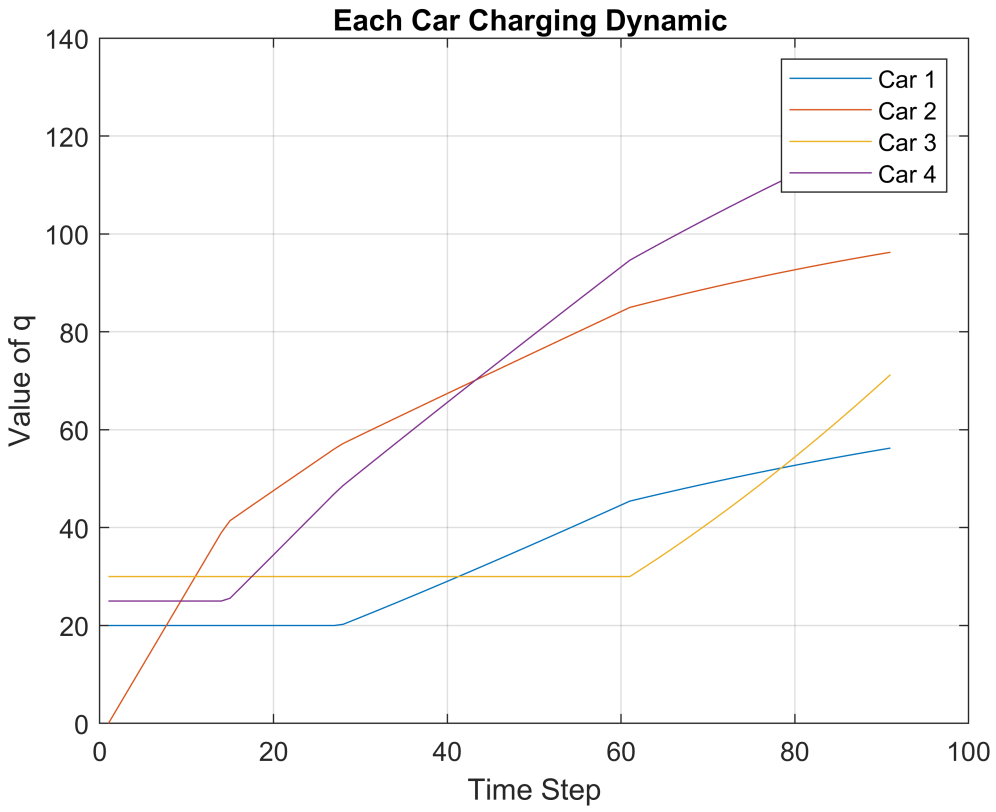
```
13|1.000|1.000|8.8e-15|2.6e-05|3.8e+00| 7.868592e+01  7.492090e+01| 0:0:01| spchol  1  1
14|0.947|0.904|4.9e-15|9.7e-06|3.2e-01| 7.680301e+01  7.648659e+01| 0:0:01| spchol  1  1
15|1.000|0.936|1.0e-14|2.8e-06|4.9e-02| 7.668948e+01  7.664279e+01| 0:0:01| spchol  1  1
16|0.996|1.000|1.6e-14|7.1e-07|9.1e-03| 7.667367e+01  7.666516e+01| 0:0:01| spchol  1  1
17|1.000|0.974|3.1e-14|1.9e-08|2.4e-03| 7.666994e+01  7.666758e+01| 0:0:01| spchol  1  1
18|0.941|0.975|2.6e-13|4.7e-10|1.3e-04| 7.666877e+01  7.666864e+01| 0:0:01| spchol  1  1
19|1.000|1.000|8.6e-13|1.0e-12|4.0e-05| 7.666870e+01  7.666866e+01| 0:0:01| spchol  1  1
20|1.000|1.000|2.5e-13|1.0e-12|5.5e-06| 7.666868e+01  7.666868e+01| 0:0:01| spchol  1  1
21|0.998|1.000|7.5e-13|1.0e-12|7.7e-08| 7.666868e+01  7.666868e+01| 0:0:01|
  stop: max(relative gap, infeasibilities) < 1.49e-08
------------------------------------------------------------------
 number of iterations   = 21
 primal objective value =  7.66686782e+01
 dual   objective value =  7.66686781e+01
 gap := trace(XZ)       = 7.73e-08
 relative gap           = 5.01e-10
 actual relative gap    = 4.95e-10
 rel. primal infeas (scaled problem)   = 7.49e-13
 rel. dual     "        "       "      = 1.00e-12
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 3.1e+03, 2.2e+01, 2.8e+01
 norm(A), norm(b), norm(C) = 1.3e+02, 1.0e+03, 1.1e+00
 Total CPU time (secs)  = 0.78
 CPU time per iteration = 0.04
 termination code       =  0
 DIMACS: 7.5e-12  0.0e+00  1.0e-12  0.0e+00  5.0e-10  5.0e-10
------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +78.503
```

```
for idx = 1 : T
    q(idx+1, :) = q(idx, :) + c(idx, :); % Charging Dynamics!
end
```

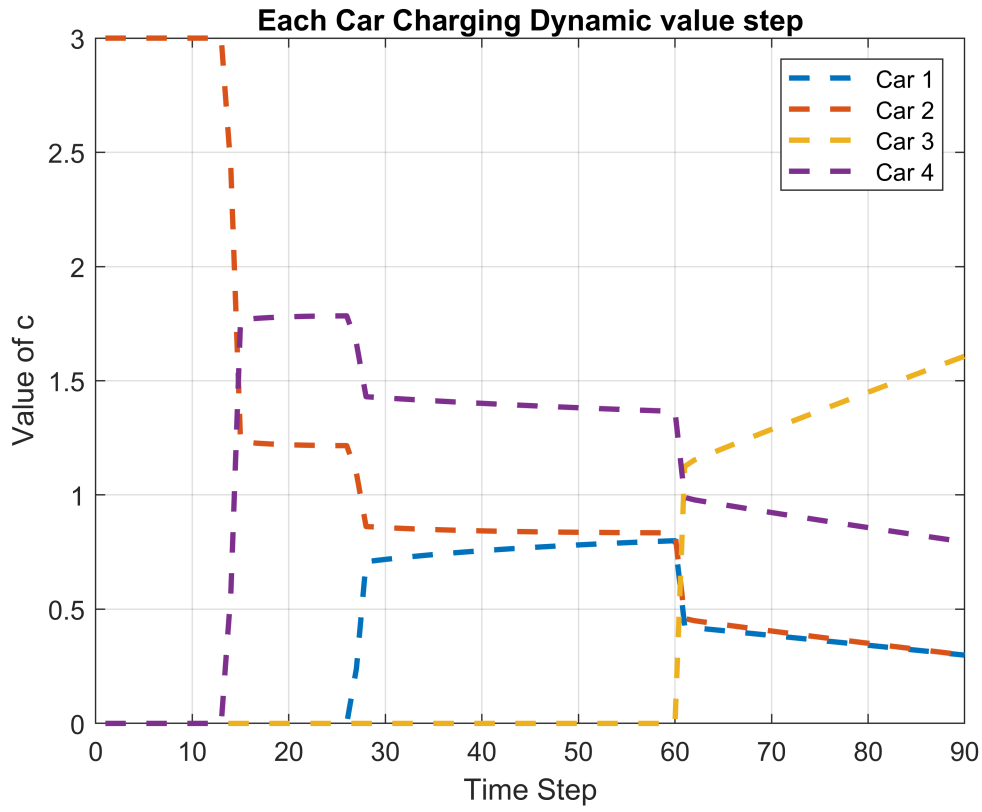
```
disp("CVX_Optval : "+cvx_optval);
```

```
CVX_Optval : 78.503
```

```
figure()
Txt = cell(N,1);
for i = 1 : N
    plot(q(:, i))
    hold on
    Txt{i} = "Car "+i;
end
xlabel('Time Step')
ylabel('Value of q')
legend(Txt)
grid on
title('Each Car Charging Dynamic ')
```




```
figure()
Txt = cell(N,1);
for i = 1 : N
    plot(c(:, i),'--','LineWidth', 2)
    hold on
    Txt{i} = "Car "+i;
end
xlabel('Time Step')
ylabel('Value of c')
legend(Txt)
grid on
title('Each Car Charging Dynamic value step')
```




```
% By using power of 1 we may get:
clear;

N = 4; % Num of cars
T = 90; % Period step
C_max = 3;

q_init = [20;0;30;25];  % Initial Point --->
q = zeros(T+1,N);
q(1,:) = q_init(:);

t= [1:T+1]';
q_des = [60;100;75;125]'; % gives the fnal value of the target minimum charge level for vehicle

Lambda = [0.5, 0.3, 2.0, 0.6]; %parameter γi sets the  urgency' of charging, with smaller value

q_target = (t./(T+1)).^Lambda.*q_des; % as defined in the figure above!

q2 = zeros(T+1,N);
q2(1,:) = q_init(:);

diff = q_target - q_init';
cvx_begin
```

```
Warning: A non-empty cvx problem already exists in this scope.
   It is being overwritten.
```

```
    variables c2(T, N);

    for t = 1 : T+1
        for i = 1 : N
            s_2(t, i) =  ( diff(t,i) - ones(1,t-1)*c2(1:t-1, i) ).^2; % The objective! --> with
        end
    end

    minimize( sum( sum(s_2) )/ ((T+1)*N)    ) ; % / ((T+1)*N) ---> not effective in optimizatio

    subject to

        c2(:) >= 0; % Implicit COnstraint
        sum(c2, 2) <= C_max; % Our Only Constraint ---sum(x,2) --> on the columns --> here is t

cvx_end
```

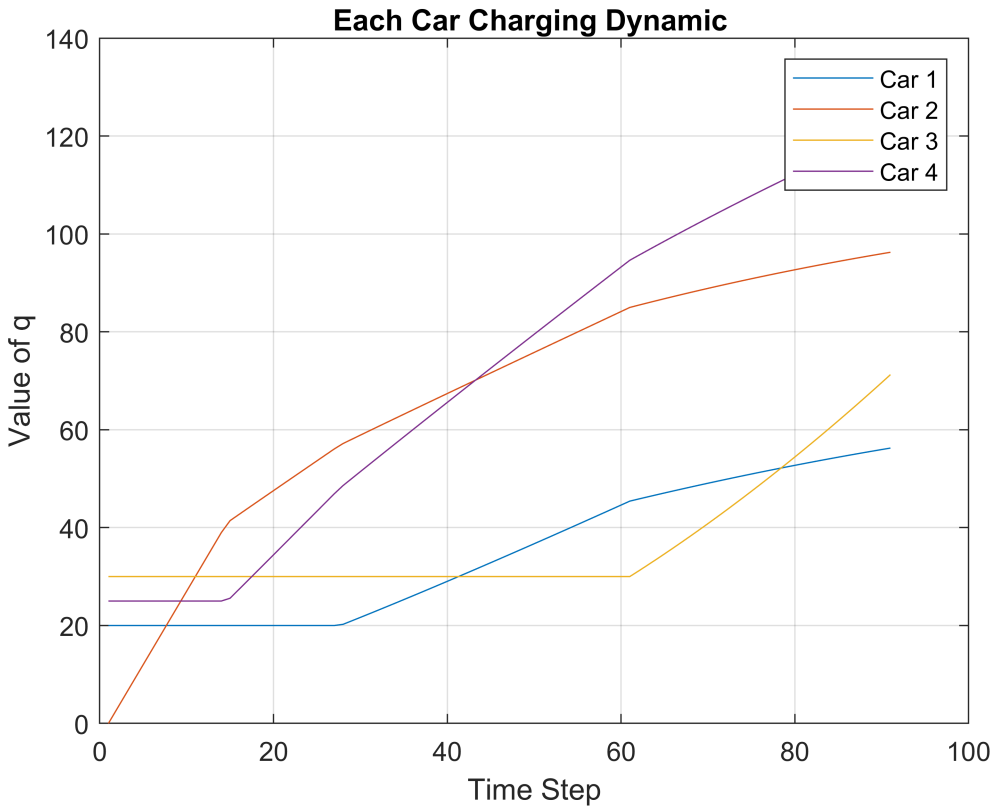
```
Calling SDPT3 4.0: 1530 variables, 720 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 720
 dim. of sdp    var  = 720,   num. of sdp  blk  = 360
 dim. of linear var  = 450
*****************************************************************
   SDPT3: Infeasible path-following algorithms
*****************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|1.8e+02|2.0e+00|9.6e+05| 9.343300e+03  0.000000e+00| 0:0:00| spchol  1  1
 1|0.719|0.718|5.1e+01|5.5e-01|3.1e+05| 9.549513e+03 -5.116452e+01| 0:0:00| spchol  1  1
 2|0.405|0.424|3.0e+01|3.2e-01|2.4e+05| 1.186148e+04 -9.001366e+01| 0:0:00| spchol  1  1
 3|0.093|0.261|2.7e+01|2.4e-01|2.1e+05| 1.161329e+04 -1.191233e+02| 0:0:00| spchol  1  1
 4|0.552|0.524|1.2e+01|1.1e-01|1.4e+05| 1.340045e+04 -1.662571e+02| 0:0:00| spchol  1  1
 5|0.476|0.443|6.4e+00|6.3e-02|9.8e+04| 1.292170e+04 -2.264794e+02| 0:0:00| spchol  1  1
 6|0.538|0.515|3.0e+00|3.0e-02|6.4e+04| 1.285667e+04 -3.267409e+02| 0:0:00| spchol  1  1
 7|0.680|0.712|9.5e-01|8.8e-03|3.2e+04| 1.252342e+04 -5.227326e+02| 0:0:00| spchol  1  1
 8|0.924|1.000|7.2e-02|1.7e-06|7.4e+03| 5.625332e+03 -7.308625e+02| 0:0:00| spchol  1  1
 9|0.936|1.000|4.6e-03|9.3e-04|7.0e+02| 2.243106e+02 -3.756407e+02| 0:0:00| spchol  1  1
10|0.957|0.919|2.0e-04|7.0e-04|2.4e+02|-3.069837e+01 -2.445892e+02| 0:0:00| spchol  1  1
11|1.000|1.000|6.1e-14|4.0e-05|8.4e+01|-1.032280e+02 -1.864967e+02| 0:0:00| spchol  1  1
12|0.894|0.888|1.7e-13|4.5e-06|1.2e+01|-1.444676e+02 -1.559571e+02| 0:0:00| spchol  1  1
13|1.000|1.000|1.5e-12|5.2e-11|6.6e+00|-1.469033e+02 -1.535388e+02| 0:0:00| spchol  1  1
14|0.904|0.947|8.4e-13|8.6e-12|5.6e-01|-1.501608e+02 -1.507194e+02| 0:0:00| spchol  1  1
15|0.874|1.000|3.4e-12|1.5e-12|1.2e-01|-1.504473e+02 -1.505700e+02| 0:0:01| spchol  2  1
16|0.982|0.943|4.5e-12|1.1e-12|1.2e-02|-1.505258e+02 -1.505375e+02| 0:0:01| spchol  2  1
17|0.936|1.000|4.9e-12|1.0e-12|3.3e-03|-1.505298e+02 -1.505331e+02| 0:0:01| spchol  2  1
18|1.000|0.950|5.9e-12|1.0e-12|4.0e-04|-1.505315e+02 -1.505319e+02| 0:0:01| spchol  2  1
19|0.959|1.000|1.1e-11|1.2e-12|1.3e-04|-1.505316e+02 -1.505317e+02| 0:0:01| spchol  1  2
20|1.000|1.000|7.0e-12|1.8e-12|1.2e-05|-1.505316e+02 -1.505317e+02| 0:0:01| spchol  2  1
21|0.998|0.998|8.2e-11|7.7e-13|1.8e-07|-1.505317e+02 -1.505317e+02| 0:0:01|
  stop: max(relative gap, infeasibilities) < 1.49e-08
```

```
------------------------------------------------------------------
 number of iterations   = 21
 primal objective value = -1.50531651e+02
 dual   objective value = -1.50531651e+02
 gap := trace(XZ)       = 1.80e-07
 relative gap           = 5.97e-10
 actual relative gap    = 5.70e-10
 rel. primal infeas (scaled problem)   = 8.16e-11
 rel. dual     "        "       "      = 7.65e-13
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 4.7e+01, 5.2e+03, 5.2e+03
 norm(A), norm(b), norm(C) = 1.8e+02, 1.1e+00, 1.3e+03
 Total CPU time (secs)  = 0.65
 CPU time per iteration = 0.03
 termination code       =  0
 DIMACS: 8.6e-11  0.0e+00  8.9e-12  0.0e+00  5.7e-10  6.0e-10
------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +156.115
```

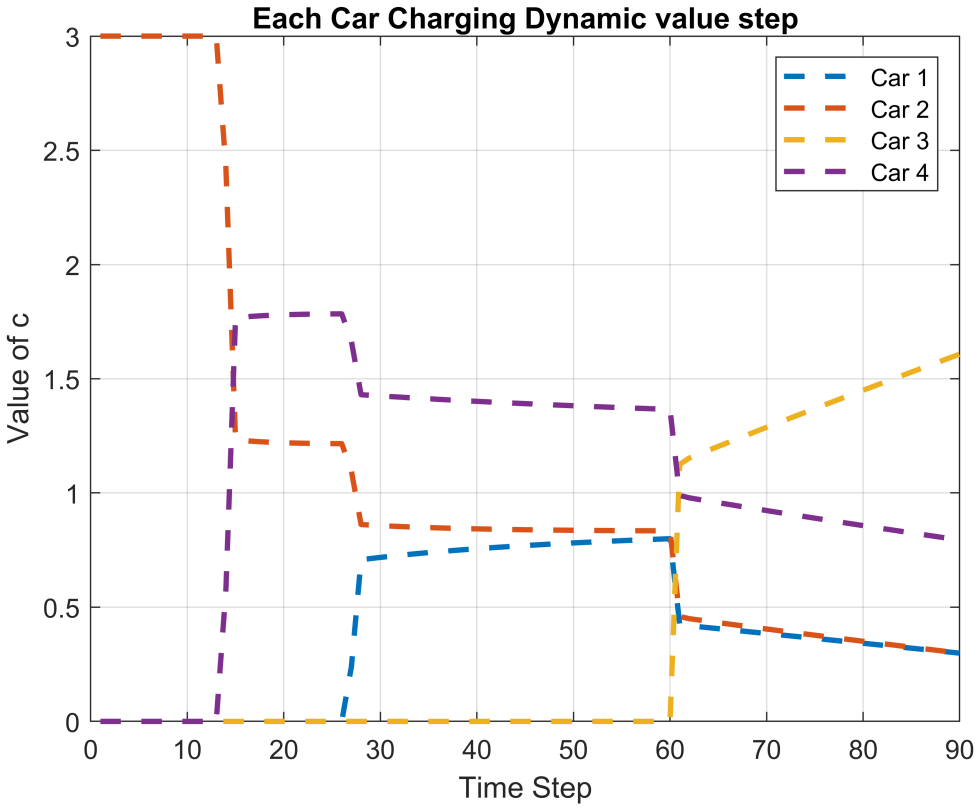
```
for idx = 1 : T
    q2(idx+1, :) = q2(idx, :) + c2(idx, :); % Charging Dynamics!
end

disp("CVX_Optval : "+cvx_optval);
```

```
CVX_Optval : 156.1154
```

```
figure()
Txt = cell(N,1);
for i = 1 : N
    plot(q2(:, i))
    hold on
    Txt{i} = "Car "+i;
end
xlabel('Time Step')
ylabel('Value of q')
legend(Txt)
grid on
title('Each Car Charging Dynamic ')
```




```
figure()
Txt = cell(N,1);
for i = 1 : N
    plot(c2(:, i),'--','LineWidth', 2)
    hold on
    Txt{i} = "Car "+i;
end
xlabel('Time Step')
ylabel('Value of c')
legend(Txt)
grid on
title('Each Car Charging Dynamic value step')
```



- *To compare both results, we can see that the objective value differs but we get the same output patter!*

Now the constant problem:

```
Theta  =    (q_des - q_init) /   sum(q_des - q_init)  ; % Dividing the gap between destination
% Gives the linear step to reach from init to destination!

C_const = Theta * C_max;

% C_const = repmat(C_const, T, 1);  % Having C for period T!
% q(t,i) = q(1,i) + sum( c(t',i) , t' = 1  to t-1)

for t = 1 : T+1
    for i = 1 : N
        S_cons(t, i) = pow_pos( diff(t,i) - C_const(i), 2); % --> equals to  sum_square( pos(x)
    end
end

q_cons = zeros(T+1,N);
q_cons(1,:) = q_init(:);


for idx = 1:T+10
    q_cons(idx+1 , :) =   q_cons(idx , :)  +  C_const'; % Charging Dynamics with constant C !
```

```
end
```

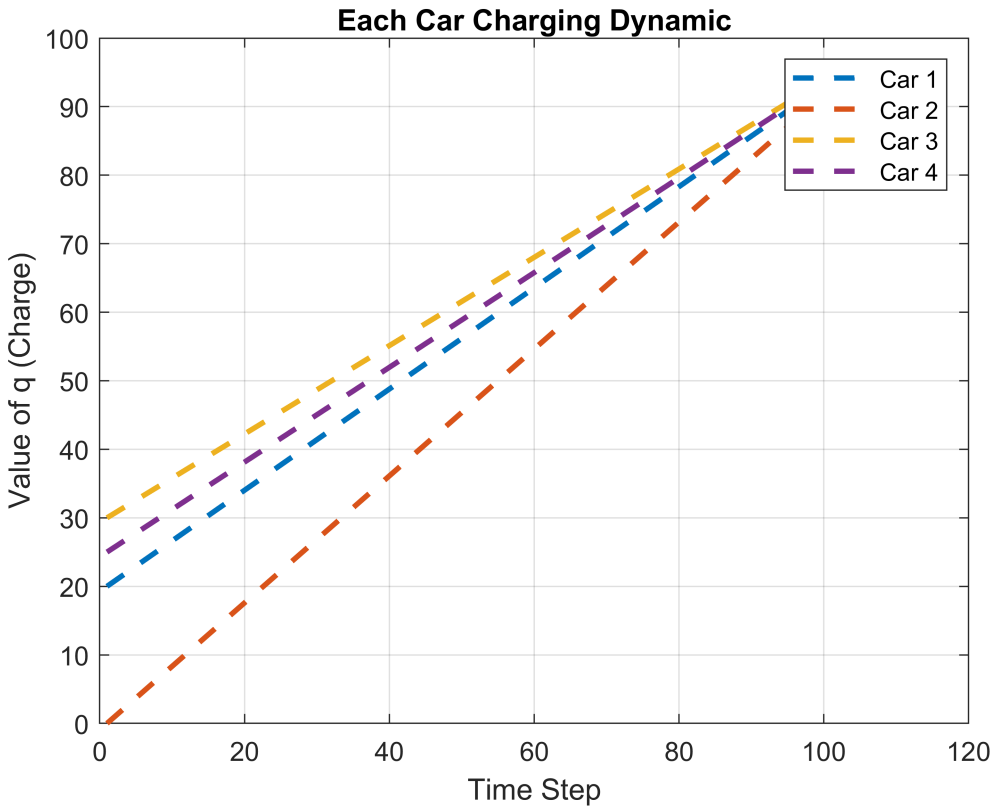
```
disp("Even after 10 steps more than T steps the difference remains: ")
```

```
Even after 10 steps more than T steps the difference remains:
```

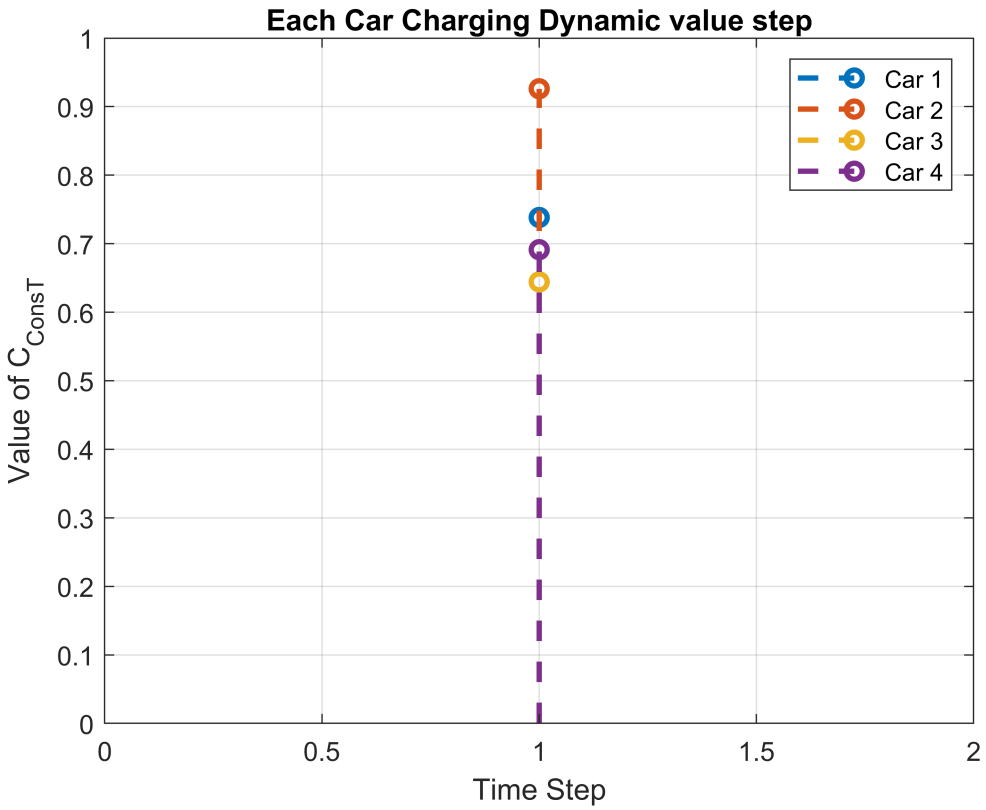
```
disp(q_cons(end,:)-q_target(end,:)) % some goes higher and some are still left behind!
```

```
   33.8259   -7.3881   19.4328  -30.8706
```

```
figure()
Txt = cell(N,1);
for i = 1 : N
    plot(q_cons(:, i),'--','LineWidth', 2)
    hold on
    Txt{i} = "Car "+i;
end
xlabel('Time Step')
ylabel('Value of q (Charge)')
legend(Txt)
grid on
title('Each Car Charging Dynamic')
```



```
figure()
Txt = cell(N,1);
for i = 1 : N
    stem(C_const(i),'--','LineWidth', 2)
    hold on
    Txt{i} = "Car "+i;
end
xlabel('Time Step')
ylabel('Value of C_{ConsT}')
legend(Txt)
grid on
title('Each Car Charging Dynamic value step')
```




```
disp("CVX_opt_Val for the objective will be : ")
```

```
CVX_opt_Val for the objective will be :
```

```
disp(sum(sum(S_cons))/((T+1)*N))
```

```
    2.6869e+03
```