Table of Contents

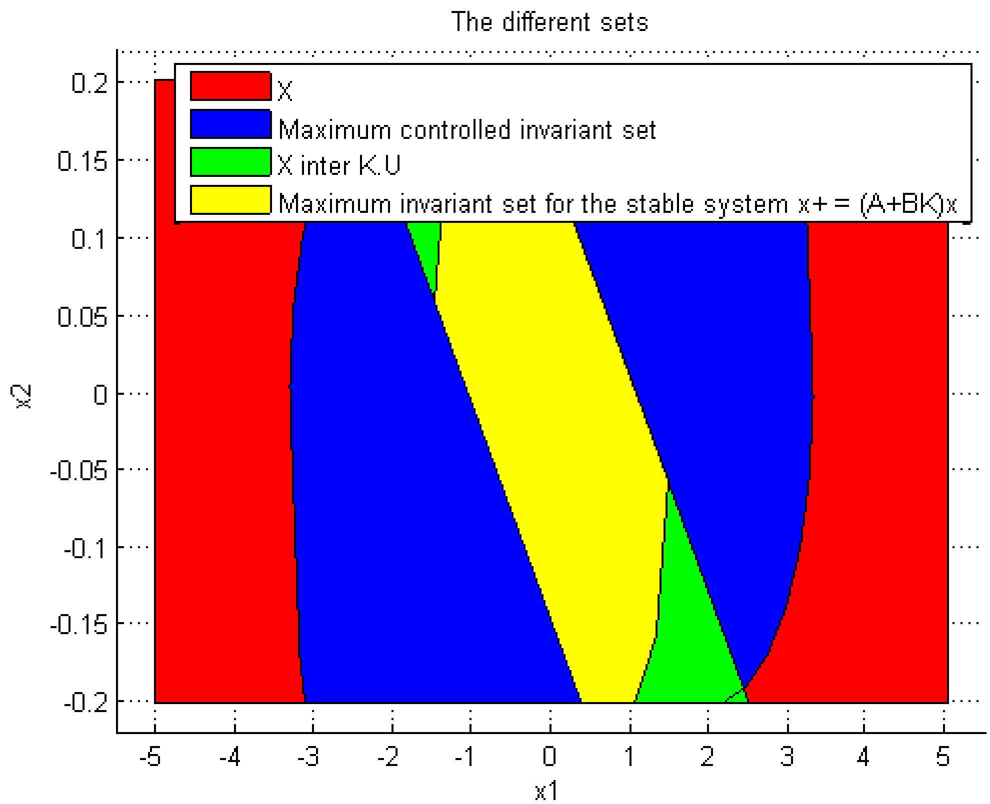

```
clear all;
close all;
clc;
warning('off'); % in order to avoid warnings in the report / it does not
 % the working of the code
```

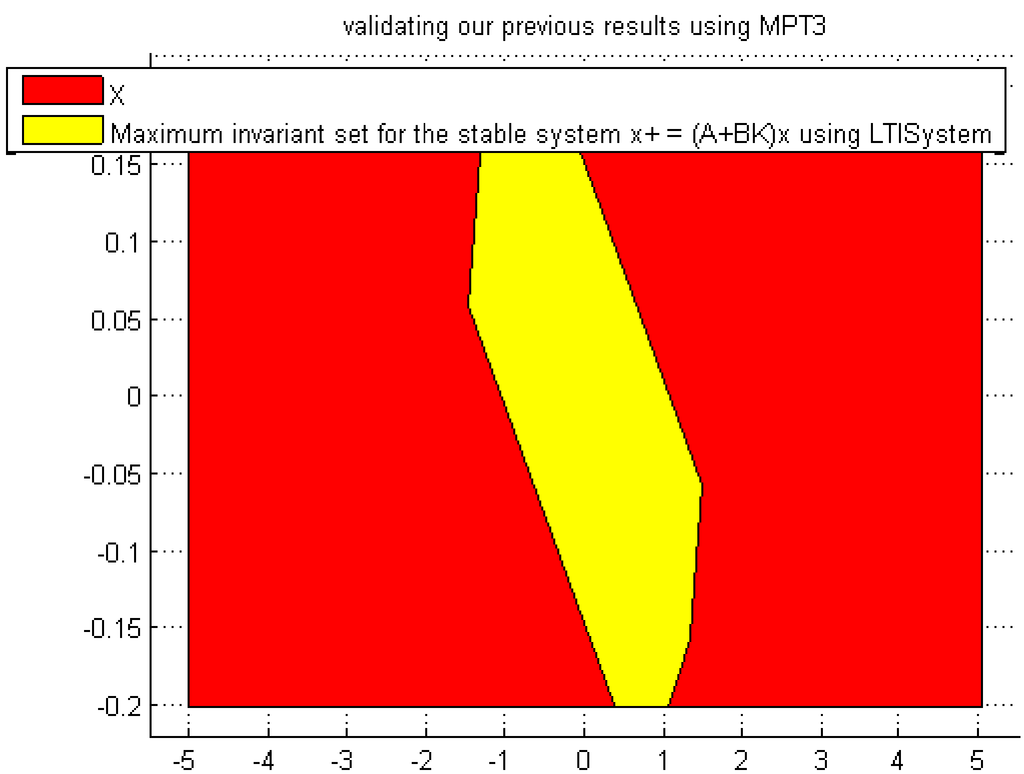
Question 1.1

```
A = [ 0.9752  1.4544;
     -0.0327  0.9315];
B = [0.0248;
     0.0327];
H = [ 1  0;
      -1  0;
       0  1;
       0 -1] ;
h = [5; 5; 0.2; 0.2];
X = Polyhedron(H, h);
hu = [1.75; 1.75];
Hu = [1; -1];

% Computing of the largest controlled invariant set of the constrained system
C = X;
while 1
    C_prev=C;
    preS = projection(Polyhedron([C.A*A C.A*B; zeros(2,2) Hu],...
        [C.b; hu]), (1:2));
    C = Polyhedron([preS.A;C.A],[preS.b;C.b]);
    if C == C_prev
        break
    end;
end;

% Computing the infinite-horizon LQR controller
R = 1;
Q = 10*eye(2);
[K,~,~] = dlqr(A,B,Q,R,zeros(size(B)));
K=-K; % In our case, u=Kx and not u=-Kx

% Computing the maximum invariant set for the stable system x+ = (A+BK)x
newA = A + B*K;
newX = Polyhedron([H;Hu*K], [h; hu]);
```

```
omega = newX;
while 1
    omega_prev=omega;
    omega = Polyhedron([omega.A*newA;omega.A],[omega.b;omega.b]);
    if omega == omega_prev
        break
    end;
end;

% Plotting the feasible and of largest controlled invariant set of the constrained
figure;
xlabel('x1');
ylabel('x2');
title('The different sets');
hold on;
plot(X, 'color', 'r');
plot(C, 'color', 'b');
plot(newX, 'color', 'g');
plot(omega, 'color', 'y');
legend('X','Maximum controlled invariant set','X inter K.U','Maximum invariant set

% Computing of the largest controlled invariant set using LTISystem
sys = LTISystem('A',A,'B',B);
sys.x.max = [5; 0.2];
sys.x.min = [-5; -0.2];
sys.u.max = [1.75];
sys.u.min = [-1.75];
sys.x.penalty = QuadFunction(Q);
sys.u.penalty = QuadFunction(R);
sys.LQRGain
sys.LQRPenalty.weight
figure();
hold on;
plot(X,'color','r');
plot(sys.LQRSet,'color','y');
title('validating our previous results using MPT3');
legend('X','Maximum invariant set for the stable system x+ = (A+BK)x using LTISyst
```

```
ans =

   -1.6478  -11.8344


ans =

   37.0252   68.3850
   68.3850  407.1177

Iteration 1...
Iteration 2...
Iteration 3...
```


The different sets
X
Maximum controlled invariant set
X inter K.U
Maximum invariant set for the stable system x+ = (A+BK)x
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
x2
-5
-4
-3
-2
-1
0
1
2
3
4
5
x1


validating our previous results using MPT3
X
Maximum invariant set for the stable system x+ = (A+BK)x using LTISystem
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5

Question 1.2: Opimize using quadprog

```
% defining all the parameters
N = 10; % the number of horizons
x0 = [3; 0]; % the initial point
A = [ 0.9752  1.4544;
     -0.0327  0.9315];
B = [0.0248;
     0.0327];

F = [ 1  0;
     -1  0;
      0  1;
      0 -1];
f = [5; 5; 0.2; 0.2];
m = [1.75; 1.75];
M = [1; -1];
R = 1;
Q = 10*eye(2);

%Defining the different matrices needed for running quadprog
Ff=F;
ff=f;
G = blkdiag(kron(eye(N-1),F), Ff, kron(eye(N),M));
g = [repmat(f,N-1,1);
    ff;
    repmat(m,N,1)];
Qf = 10*Q;
H = blkdiag(kron(eye(N-1),Q), Qf, kron(eye(N),R));
h = [zeros(2*(N-1),1);
     zeros(2,1);
     zeros(N,1)];
T1 = [zeros(2,2*(N-1))  zeros(2)
      kron(eye(N-1),A)  zeros(2*(N-1),2)];
T1 = T1 + kron(eye(N),eye(2));
T2 = kron(eye(N),-B);
T  = [T1,T2];

x = zeros(2,100);
x(:,1) = x0;
for i=2:100
    t  = [A*x(:,i-1);
      zeros(2*(N-1),1)];
    [zopt, fval, flag] = quadprog(H, h, G, g, T, t);
    u = zopt((N*size(A,2) + 1):(N*size(A,2) + size(B,2)));
    if flag==1
        x(:,i) = A*x(:,i-1)+B*u;
    else
        break
    end
end

% Plotting
```

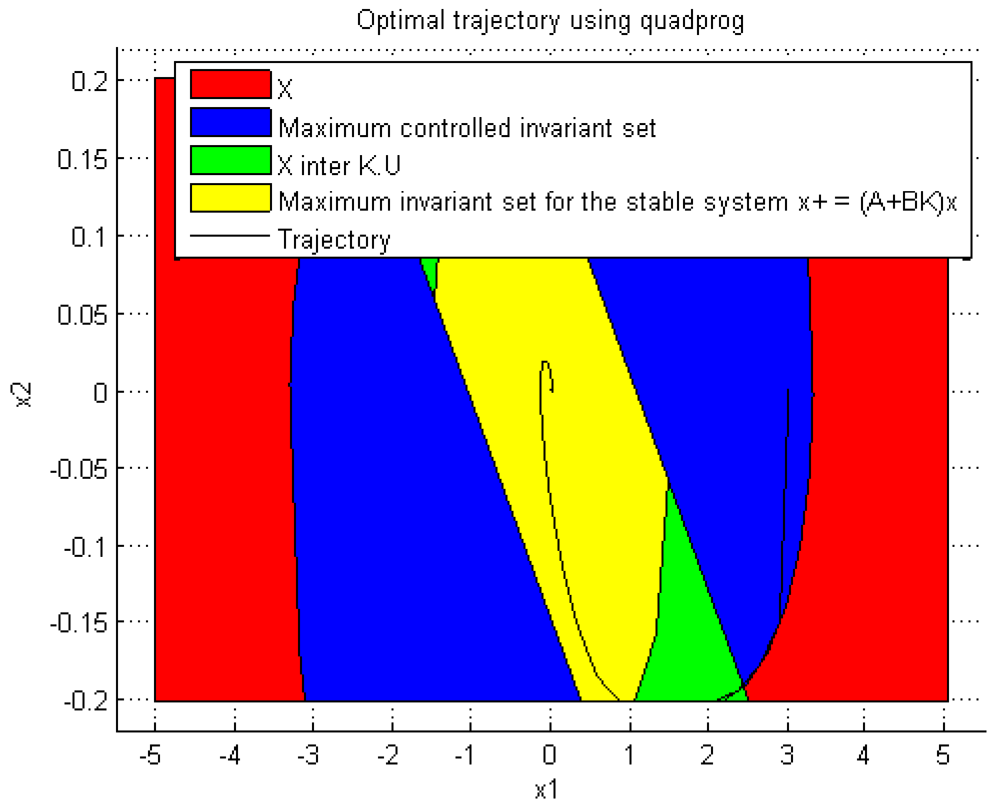
```
figure;
xlabel('x1');
ylabel('x2');
title('Optimal trajectory using quadprog');
hold on;
plot(X, 'color', 'r');
plot(C, 'color', 'b');
plot(newX, 'color', 'g');
plot(omega, 'color', 'y');
plot(x(1,:),x(2,:), 'color','k');
legend('X','Maximum controlled invariant set','X inter K.U','Maximum invariant set
```

```
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
```

```
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
Optimization terminated.
```

Optimization terminated.

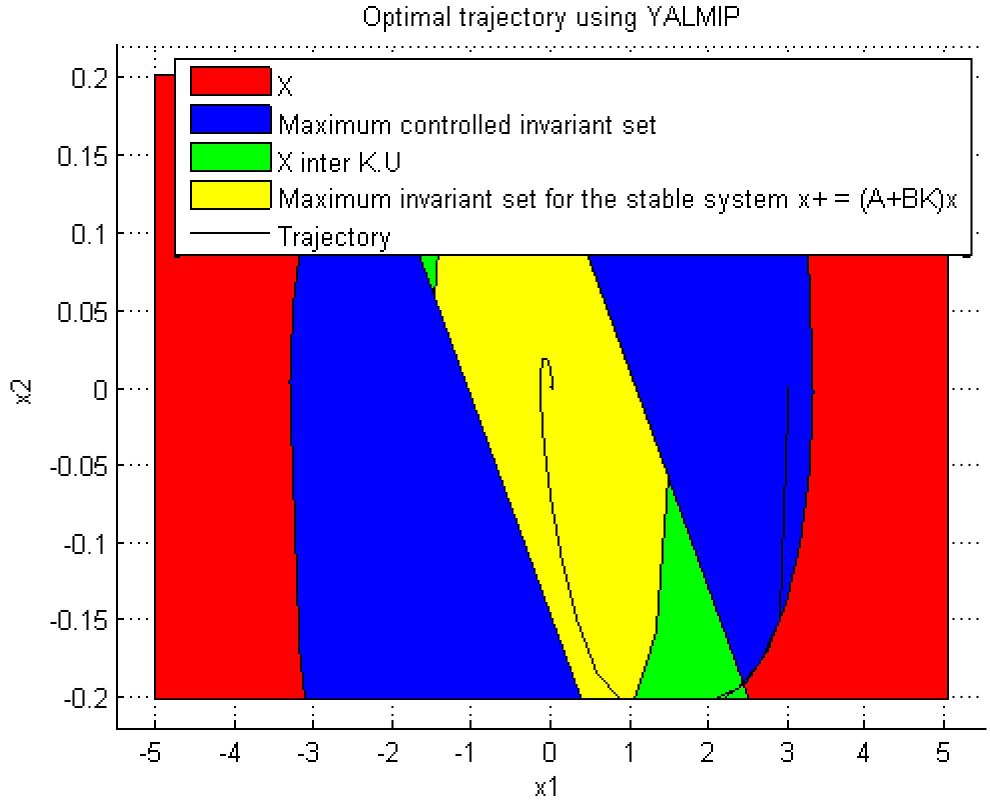
Optimization terminated.
Optimization terminated.



Question 2: Opimize using Yalmip

```
% defining all the parameters
N = 10;
x0 = [3; 0];
A = [ 0.9752  1.4544;
     -0.0327  0.9315];
B = [0.0248;
     0.0327];
F = [ 1  0;
     -1  0;
      0  1;
      0 -1];
f = [5; 5; 0.2; 0.2];
m = [1.75; 1.75];
M = [1; -1];
R = 1;
Q = 10*eye(2);
Ff=F;
ff=f;
Qf = 10*Q;
```

```
x = zeros(2,100);
x(:,1) = x0;

% Define optimization variables
z = sdpvar(2,N,'full');
u = sdpvar(1,N-1,'full');
X0 = sdpvar(2,1,'full');
% Define constraints and objective
con = [];
obj = 0;
con = [con, z(:,1) == X0];
for i = 1:N-1
    con = [con, z(:,i+1) == A*z(:,i) + B*u(:,i)]; % System dynamics
    con = [con, F*z(:,i) <= f]; % State constraints
    con = [con, M*u(:,i) <= m]; % Input constraints
    obj = obj + z(:,i)'*Q*z(:,i) + u(:,i)'*R*u(:,i); % Cost function
end
con = [con, Ff*z(:,N) <= ff]; % Terminal constraint
obj = obj + z(:,N)'*Qf*z(:,N); % Terminal weight
% Defining the optimizer
ops = sdpsettings('solver','sedumi','verbose',0); % choosing the solver
% We saw that the results of the optimazation depends a lot on the choice
% of the solver
ctrl = optimizer(con, obj, ops, X0, u(:,1));
% Runing the loop and computing the optimal input à each step
% using the optimizer defined above
for j=2:100
    [uopt,isfeasible] = ctrl{x(:,j-1)};
    % isfeasible == 1 if the problem was solved successfully
    x(:,j) = A*x(:,j-1)+B*uopt; % state equation
end

% Plotting
figure;
xlabel('x1');
ylabel('x2');
title('Optimal trajectory using YALMIP');
hold on;
plot(X, 'color', 'r');
plot(C, 'color', 'b');
plot(newX, 'color', 'g');
plot(omega, 'color', 'y');
plot(x(1,:),x(2,:), 'color','k');
legend('X','Maximum controlled invariant set','X inter K.U','Maximum invariant set
% We can see that we have the same trajectory using the 2 optimizers
```



Published with MATLAB® R2013b