

```

function [Xallmpc, Uallmpc, horvalmpc, i] = solveMPC(horizon, x0, A, B, P, Q, R, xmax, xmin, umax, umin, Xf, solvehorizon)
optvalmpc = 0;
if nargin < 13
    N = 50; % solving horizon for solution
else
    N = solvehorizon;
end
%store solutions
n = size(B,1);
m = size(B,2);
Xallmpc = zeros(n,N+1);
Uallmpc = zeros(m,N);
T = horizon; % mpc horizon passed in
x = x0; %reset initial state
Xallmpc(:,1) = x;

Qhalf = sqrtm(Q); Rhalf = sqrtm(R);

fprintf('MPC horizon=%d ; timestep = ',T);

%step through time
for i = 1:N
    fprintf('%d, ',i-1);

    %cvx precision
    cvx_precision(min(max(min(abs(x))/10,1e-6),0.99999))
    if Xf == 0
        cvx_begin quiet
            variables X(n,T+1) U(m,T)
            max(X') <= xmax'; max(U') <= umax';
            min(X') >= xmin'; min(U') >= umin';
            X(:,2:T+1) == A*X(:,1:T)+B*U;
            X(:,1) == x; %initial state constraint
            X(:,T+1) == 0; %terminal state constraint
            minimize (X(:,T+1)'*P*X(:,T+1)+pow_pos(norm([Qhalf*X(:,1:T);Rhalf*U],2),2))
        cvx_end
    else
        cvx_begin quiet
            variables X(n,T+1) U(m,T)
            max(X') <= xmax'; max(U') <= umax';
            min(X') >= xmin'; min(U') >= umin';
            X(:,2:T+1) == A*X(:,1:T)+B*U;
            X(:,1) == x; %initial state constraint
            minimize (X(:,T+1)'*P*X(:,T+1)+pow_pos(norm([Qhalf*X(:,1:T);Rhalf*U],2),2))
        cvx_end
    end

    %check feasibility
    if strcmp(cvx_status, 'Solved')

        %store control
        u= U(:,1);
        Uallmpc(:,i) = u;

        %accumulate cost
        optvalmpc = optvalmpc + x'*Q*x + u'*R*u;

        %forward propagate state
        x = A*x+B*u;

        %record state
        Xallmpc(:,i+1) = x;

    else
        fprintf('ERROR')
        % break from loop
        optvalmpc = Inf;
    end
end

```

```
        break;
    end
end
fprintf('\n');

horvalmpc = 0;
for k = 1:(horizon-1)
    xk = Xallmpc(:,k);
    uk = Uallmpc(:,k);
    horvalmpc = horvalmpc + xk'*Q*xk + uk'*R*uk;
end
x_last = Xallmpc(:,horizon);
horvalmpc = horvalmpc + x_last'*P*x_last;

end
```