```
clc;
clear;
% Optimal control example of a simple car model
% time horizon and segments
tf = 10;
S.N = 16;
S.h = tf/S.N;
S.tf = tf;

%constraints:
S.cd = 3;
S.obstacle = [-5,-1];

% cost function parameters
S.Q = .0*diag([5, 5, 1, 1, 0]);
S.R = diag([1, 5]);
S.Qf = diag([5, 5, 1, 1, 0]);

S.f = @car_f; % car dynamics and jacobians
S.L = @car_L; % car cost (just standard quadratic cost)
S.Lf = @car_Lf; % car cost (just standard quadratic cost)
S.con = @constrains;
%S.conf = @termit;

% initial state
x0 = [-10; -2; -0.5; 1; 0];

us = zeros(2,S.N);
xs = traj(x0, us, S);
plottraj(xs, us, S)
% optimize trajector
[xs, us, cost_seq, exitflag,output] = trajopt_sqp(xs, us, S,@plottraj);

% plot result
subplot(1,2,1)
plot(xs(1,:), xs(2,:), '-g', 'LineWidth',3);

function c = constrains(k,x,u,S)
    c = S.cd^2 - (x(1)-S.obstacle(1))^2 - (x(2)-S.obstacle(2))^2;
end

function [L, Lx, Lxx, Lu, Luu] = car_L(k, x, u, S)
% car cost (just standard quadratic cost)
    L = S.h/2*(x'*S.Q*x + u'*S.R*u);
    Lx = S.h*S.Q*x;
    Lxx = S.h*S.Q;
    Lu = S.h*S.R*u;
    Luu = S.h*S.R;
end

function [L, Lx, Lxx] = car_Lf(x, S)
% car cost (just standard quadratic cost)
```

```
    L = x'*S.Qf*x/2;
    Lx = S.Qf*x;
    Lxx = S.Qf;
end

function [x, A, B] = car_f(k, x, u, S)
% car dynamics and jacobians
    h = S.h;
    c = cos(x(3));
    s = sin(x(3));
    v = x(4);
    w = x(5);

A = [1 0 -h*s*v h*c 0;
     0 1 h*c*v h*s 0;
     0 0 1 0 h;
     0 0 0 1 0;
     0 0 0 0 1];
B = [0 0;
     0 0;
     0 0;
     h 0;
     0 h];

x = [x(1) + h*c*v;
     x(2) + h*s*v;
     x(3) + h*w;
     v + h*u(1);
     w + h*u(2)];
end

function xs = traj(x0, us, S)
    N = size(us, 2);
    xs(:,1) = x0;
    for k=1:N
        xs(:, k+1) = S.f(k, xs(:,k), us(:,k), S);
    end
end

function J = cost(xs, us, S)
    N = size(us, 2);
    J = 0;
    for k=1:N+1
        if k < N+1
            [L, Lx, Lxx, Lu, Luu] = S.L(k, xs(:,k), us(:,k), S);
        else
            [L, Lx, Lxx] = S.Lf(xs(:,end), S);
        end
            J = J + L
    end
    fprintf("%f\n",J);
end

function f = plottraj(xs, us, S)
```

```
%only plot every 250'th call
    global pc
    pc = pc + 1;
    if (mod(pc,100))
        return
    end

    subplot(1,2,1)
    % draw all obstacles
    r=S.cd;
    theta=0:pi/200:2*pi;
    x=r*cos(theta)+S.obstacle(1);
    y=r*sin(theta)+S.obstacle(2);

    plot(S.obstacle(1),S.obstacle(2),'*r');
    plot(x,y,'-r');
    plot(xs(1,:), xs(2,:), '-b');
    xlabel('x')
    ylabel('y')
    axis equal
    drawnow
    hold on
    subplot(1,2,2)
    plot(0:S.h:S.tf-S.h, us(1,:),0:S.h:S.tf-S.h, us(2,:));
    xlabel('sec.')
    legend('u_1','u_2')
end
```

Published with MATLAB® R2021b