

```

% Function providing equality and inequality constraints
% ceq(var) = 0 and c(var) \le 0

function [c,ceq] = constraint(var)

global N;
global T;

global x0;
global y0;

% Put here constraint inequalities
c = [];

% Note that var = [x;y;u]
x = var(1:N+1); y = var(N+2:2*N+2); u = var(2*N+3:3*N+3);

% Computing dynamical constraints via the trapezoidal rule
h = 1.0*T/(1.0*N);
for i = 1:N
    % Provide here dynamical constraints via the trapezoidal formula
    [xDyn_i,yDyn_i] = fDyn(x(i),y(i),u(i));
    [xDyn_ii,yDyn_ii] = fDyn(x(i+1),y(i+1),u(i+1));
    ceq(i) = x(i+1) - x(i) - h*(xDyn_i + xDyn_ii)/2;
    ceq(i+N) = y(i+1) - y(i) - h*(yDyn_i + yDyn_ii)/2;
end

% Put here initial conditions
ceq(1+2*N) = x(1) - 1;
ceq(2+2*N) = y(1) - 0;

```