Report

Yeskendir Koishekenov *

July 27, 2017

*Control Laboratory

Abstract

I studied Model Predictive Control (MPC), the powerful technique for optimizing the performance of constrained systems. Additionally, I implemented MPC to solve cart-pole problem: stabilize cart-pole system with given initial conditions and constraints.

1 Introduction

MPC uses the current plant measurements, the current dynamic state of the process, the MPC models, and the process variable targets and limits to calculate future changes in the dependent variables. These changes are calculated to hold the dependent variables close to target while honoring constraints on both independent and dependent variables. The MPC typically sends out only the first change in each independent variable to be implemented, and repeats the calculation when the next change is required. Alternative names to this technique are dynamic matrix control, receding horizon control, dynamic linear programming, rolling horizon planning.

2 Theory of MPC

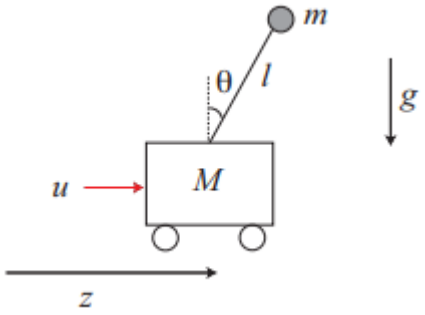
Model Predictive Control is the planning exercise. At each time t we solve the (planning) problem with planning horizon T. We start at the current state at time t and propagate state forward the dynamics of the system T samples. In T samples state hits zero (terminal constraint). After doing complete planning exercise, only first input is used. Next move one step forward control re-plan again with shifted horizon.
In other words, in order to solve the planning problem we need to minimize cost function:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{\tau=t}^{t+T} l(x(\tau), u(\tau)) \\ \text{subject to} \quad & u(\tau) \in U, x(\tau) \in X, \tau = t, ..., t+T \\ & x(\tau+1) = Ax(\tau) + Bu(\tau), \tau = t, ..., t+T-1 \\ & x(t+T) = 0 \end{aligned}$$

with variables $x(t{+}1), ..., x(t{+}T), u(t), u(t{+}T{-}1)$ and data $x(t), A, B, l, X, U$
The solution is $\widetilde{x}(t{+}1), ..., \widetilde{x}(t{+}T), \widetilde{u}(t), ..., \widetilde{u}(t{+}T{-}1)$ and we interpret these as the *plan of action* for the next T steps. We take only first input $u(t) = \widetilde{u}(t)$ and this gives a complicated state feedback control $u(t) = \phi_{mpc}(x(t))$.

3 Cart-Pole problem

To better understand MPC I implemented technique on cart-pole example. The initial conditions and constraints are given. Our goal is to stabilize the system. Initial conditions are $\theta = \frac{pi}{6}, x = 1, \dot{x} = 0, \dot{\theta} = 0$; constraints are $-\pi/2 \leq \theta \leq \pi/2, -10 \leq x \leq 10, -100 \leq u \leq 100$. Next we need to describe model used for simulations and control design. The figure below depicts cart-pole system.



Equations of the motion are the following,

$$ml^2\ddot{\theta} + mlcos\theta\ddot{z} - mglsin\theta = 0$$
$$mlcos\theta\ddot{\theta} + (M+m)\ddot{z} - ml\ddot{\theta}^2 sin\theta = u$$

To mathematically define physical model we use state-space representation with state-space variables $x_1 = \theta, x_2 = z, x_3 = \dot{\theta}, x_4 = \dot{z}$.

$$\dot{x_1} = x_3$$
$$\dot{x_2} = x_4$$
$$\dot{x_3} = \frac{(m+M)gsinx_1 - mlx_3^2 sinx_1 cosx_1}{l(msin^2x_1 + M)} - \frac{cosx_1}{l(msin^2x_1 + M)}u$$
$$\dot{x_4} = \frac{-mglcosx_1 sinx_1 + mlx_3^2 sinx_1}{msin^2x_1 + M} + \frac{1}{msin^2x_1 + M}u$$

or

$$\dot{x} = f(x, u)$$

where $m = 1, M = 1, l = 1, g = 9.8$. This model is nonlinear; however, MPC approaches are based on linear or piecewise linear approximations of the nonlinear system. Consider the following discrete-time nonlinear system:

$$x(k+1) = \widetilde{f}(x(k), u(k))$$

To find $\widetilde{f}(x(k), u(k))$ we use Euler's method. Let $x_k = x(k\Delta T), u_k = u(k\Delta T)$, where ΔT is sampling rate

$$\frac{x((k+1)\Delta T) - x(k\Delta T)}{\delta T} = f(x(k\Delta T), u(k\Delta T))$$

$$\frac{x_{k+1} - x_k}{\Delta T} = f(x_k, u_k)$$

So

$$x_{k+1} = x_k + f(x_k, u_k)\Delta T = \widetilde{f}(x_k, u_k)$$

System can be approximated using Taylor series and take form

$$\dot{x} = Ax + Bu$$

where

$$A = \frac{\partial f}{\partial x}(0,0) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{(m+M)g}{lM} & 0 & 0 & 0 \\ \frac{-mg}{M} & 0 & 0 & 0 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u}(0,0) = \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{lM} \\ \frac{1}{M} \end{bmatrix}$$

After discretization and linearization

$$x_{k+1} = x_k + (Ax_k + Bu_k))\Delta T$$

As mentioned before we need to minimize cost function $\sum_{\tau=t}^{t+T} l(x(\tau), u(\tau))$, where stage cost function is

$$l(x, u) = x^T Q x + u^T R u$$

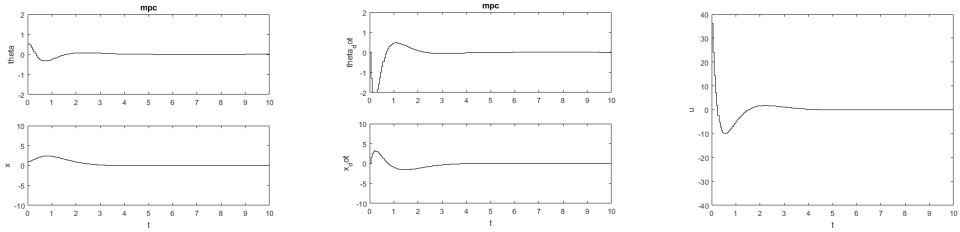
Q and R are weighting matrices of appropriate dimensions. Q and R are both identity matrices. To minimize function we use CVX, the Matlab-based modeling system for convex optimization.

```
%Model Predictive Control
N=50;
x=zeros(4,size(t,2));
u=zeros(1,size(t,2));
init=IC;
for k=1:size(t,2)

    cvx_begin
        variables X(4,N+1) U(1,N)
        max(X(2,:)') <= zmax; min(X(2,:)') >= zmin;
        max(X(1,:)') <= thetamax; min(X(1,:)') >=
            thetamin;
        max(U') <= umax; min(U') >= umin;
        X(:,2:N+1) == dt*(A*X(:,1:N)+B*U)+X(:,1:N);
        X(:,1) == init;
        X(:,N+1) == 0;
        minimize (norm([Qhalf*X(:,1:N); Rhalf*U],'fro')
            );
    cvx_end
    x(:,k)=init;
    u(1,k)=U(1,1);
    init=X(:,2);

end
```

The results are on the Fig.1.



(a) x,θ (b) $\dot{x},\dot{\theta}$ (c) u

Figure 1: Results

To make system continuous we use Runge-Kutta method for the approximate

solution of ordinary differential equation. Let an initial value problem be specified as follows:

$$\dot{x} = f(t, x), x(t_0 = x_0)$$

We picked a step-size h=0.01 and define

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$t_{n+1} = t_n + h$$

for n=0,1,2,3,..., using

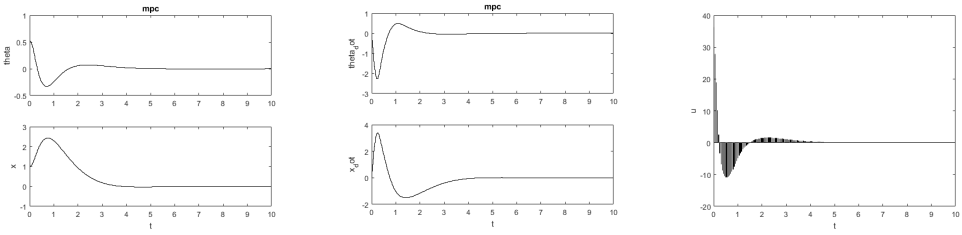
$$k_1 = f(t_n, x_n)$$
$$k_2 = f(t_n + 0.5h, x_n + 0.5hk_1)$$
$$k_3 = f(t_n + 0.5h, x_n + 0.5hk_2)$$
$$k_4 = f(t_n + h, x_n + hk_3)$$

We implement this algorithm in our code,

```
%Model Predictive Control
N=50; x=zeros(4,size(t,2)); u=zeros(1,size(t,2));
init=IC; umpc =0; x(:,1)=init;
for k=1:size(t,2)-1
    if rem(k,5) == 1
        cvx_begin
            variables X(4,N+1) U(1,N)
            max(X(2,:)') <= zmax; min(X(2,:)') >= zmin;
            max(X(1,:)') <= thetamax; min(X(1,:)') >=
                thetamin;
            max(U') <= umax; min(U') >= umin;
            X(:,2:N+1) == dt*(A*X(:,1:N)+B*U)+X(:,1:N);
            X(:,1) == init;
            X(:,N+1) == 0;
            minimize (norm([Qhalf*X(:,1:N); Rhalf*U],'
                fro'));
```

```
        cvx_end
        u(1,k)=U(1,1);
        umpc =U(1,1);
    end;
        k1 = dx(t(k+1),x(:,k),umpc);
        k2 = dx(t(k+1)+h/2, x(:,k)+0.5*h*k1,umpc);
        k3 = dx(t(k+1)+h/2, x(:,k)+0.5*h*k2,umpc);
        k4 = dx(t(k+1)+h,x(:,k)+h*k3,umpc);
        x(:,k+1) = x(:,k)+(k1+2*k2+2*k3+k4)*h/6;
        init = x(:,k+1);
end
```

The results are on the Fig.2.



(a) x,θ (b) $\dot{x}, \dot{\theta}$ (c) u

Figure 2: Results (with Runge-Kutta method)

4 Conclusion

Model Predictive Control predicts the future behavior of the controlled system over a finite time horizon and compute an optimal control input. This approach has been tested in simulations with cart-pole example. Results are very god as it stabilizes the system under specific constraints in short time.

References

[1] Stephen Boyd. *"Model Predictive Control"*. Stanford university.

[2] *wikipedia.org*