역진자계에서 비선형 MPC 모사

한현각*, 정수용[1]
순천향대 화학공학과, 죠지아공대 기계공학과[1]
(chemhan@sch.ac.kr*)

Simulation of Nonlinear Predictive Control of the Inverted Pendulum

Hyun Kak Han*, Sooyoug Jung[1]
Department of Chemical Engineering, Soonchunhyang University
Department of Mechanical Engineering, Georgia Institute Technology[1]
(chemhan@sch.ac.kr*)

Abstract
The experimental implementation of model predictive control Algorithm to the inverted pendulum is presented. he Actuator constraint is also incorporated via exterior penalty function. The key point in this algorithm is to reduce the error at the end of the prediction horizon rather than tries to find the optimal solution. This reduce the computational load and allows for real-time implementation.

1. Introduction

Model uncertainty and external disturbances are important concerns in model predictive control (MPC) and widely investigated in recent years. MPC is a feedback control scheme that generates the control action based on the finite horizon open loop optimal control with the measured state as the initial state. MPC offers the possibility of the incorporating control and state constraints which few feedback control methods can claim to do. MPC was first proposed for the linear systems [1, 2]. and later extended to nonlinear systems [3, 4, 5, 6]. It has been especially popular in process control where slow system response allows time for the on-line optimal control computation. However, due to the computational load, application to systems with fast time constant still elusive. We have proposed an NMPC scheme in [7] which only takes a finite number of newton steps in each sampling period instead of solving the complete optimal control problem.

For the real-time implementation, MATLAB xPC target was used as the real time computation platform. And the used inverted pendulum was made by Realgain Co.

2. Nonlinear Model Predictive Control (NMPC)

The algorithm is based on the resul in [7]. The basic idea is simple: the open loop control law iteration is excuted with the current state as the initial state and λ in current iteration as the initial guess. Then a fixed number of Newton-steps is taken. The resulting control is applied. The process repeats at the next sampling.

2.1 Description of the NMPC Algorithm

For the NMPC implementation, we use the pulse basis for discretization. This is done due

to the need for a moving horizon. The extension to other basis for approximation is currently under investigation.

To describe the algorithm analytically, consider a discrete nonlinear system (obtained through, for example, finite difference approximation of a nonlinear affine control system):

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{1}$$

where $x_k \in R^n$ and $u_k \in R^m$. Let the prediction horizon be M. Denote the predictive control vector at time k by $\mathcal{U}_{k.M}$:

$$\mathcal{U}_{k.M} = [u_1(k), \ldots\ldots, u_M(k)], \quad \mathcal{U}_{k,M} \in R^{m \cdot M} \cdot \tag{2}$$

Let $\Phi_M(x_k, \mathcal{U}_{k,M})$ be the state at the end of the prediction horizon, x_{k+M}, starting from x_k and using the control vector $\mathcal{U}_{k,M}$ The predicted state error is then

$$e_M(k) = \Phi_M(x_k, \mathcal{U}_{k,M}) - x_d. \tag{3}$$

The main idea of the algorithm is to simultaneously perform the open loop iteration over the prediction horizon and apply the updated control to the system at the same time.

The implementation of the algorithm can be summarized as follows:

1. At the initial time with the given initial state x_0, choose the initial guess of the predictive control vector $\mathcal{U}_{0,M}$. Also compute the equilibrium control, u^* from

$$f(x_d) + g(x_d)u^* = x_d \tag{4}$$

2. For $k \geq 0$,

(a) Calculate one Newton-step control update:

$$\mathcal{V}_{k,M} = \mathcal{U}_{k,M} - \alpha_k(\nabla_{\mathcal{U}_{k,M}} \Phi_M(x_k, \mathcal{U}_{k,M}))e_M(k) \cdot \tag{5}$$

The gain, α_k, is found based on Amijo's rule to ensure predicted error is strictly decreasing. This can be done as long as the gradient matrix is non-singular.

(b) Shift the predictive control vector by 1 step (since the first element will be used for the actual control) and add the equilibrium control at the end of the vector:

$$\mathcal{U}_{k+1,M} = G\mathcal{V}_{k,M} + Fu^* \tag{6}$$

Where $G \in R^{mM \times mM}$ and $F \in R^{mM \times m}$ are defined as

$$G = \begin{bmatrix} 0_{m(M-1)*m} & I_{m(M-1)} \\ 0_{m*m} & 0_{m*m(M-1)} \end{bmatrix},$$

$$F = \begin{bmatrix} 0_{m(M-1)*m} \\ I_m \end{bmatrix}$$

(c) Compute the control, $u(k)$ to be applied as

$$u(k) = P_1 \mathcal{V}_{k,M} \tag{7}$$

where $P_1 = [\, I_m \, 0_{m \times (M-1)m} \,] \cdot$

(d) Repeat Step 2a-2c at the next time instant.

Remarks:

1. In the above algorithm, the control constraint can be incorporated through exterior penalty functions, : $z(u)$

$$g(u)=\begin{cases} 0 & |u|\le u_{\max} \\ \gamma(u-u_{\max})^2 & u>u_{\max} \\ \gamma(u+u_{\max})^2 & u<-u_{\max} \end{cases} . \tag{8}$$

The constraint is imposed at the discrete time points, $\{t_i\}$, i = 1,....,M, so the overall penalty function is

$$z(u)=\sum_{i=1}^{M} g(u(t_i)). \tag{9}$$

The Newton update law(5) now needs to be modified to drive (e,z) to zero:

$$u_{k,M}=u_{k,M}-\alpha_k\left(\begin{bmatrix} \iota \\ \varsigma \end{bmatrix}\right)e_M(k) . \tag{10}$$

where ς is the gradient matrix of z with respect to u and ι is $\nabla_{u_{k,M}} \Phi_M(x_k, u_{k,M})$.

2. The above iteration strategy is based on the optimal control problem where the only cost is the final state error.

3. For stability, the non-singularity condition of the gradient matrix is needed [7].

4. The parameters that affect the performance of this algorithm are:

• Prediction horizon M: This is determined from T/t_s where T is the horizon time and t_s is the sampling rate. Large T is beneficial in keeping u within the constraint but implies heavier computation load in real-time. Small t_s is important to keep the approximation error small, but it also leade to large N and heavier real-time computation load.

• Initial predictive control vector, u_0,M: Without any a priori insight, this can just be chosen as a zero vector. If some performed, it can be used as the initial guess.

3. Hardware

The feedback NMPC control is implemented using MATLAB xPC target with Real-Time Workshop and simulink Toolbox from mathworks, Inc. The incremental encorder board is PCI-QUAD04 from Measurement Computing, Corp. The D/A board is PLC-816 from Advantech Inc.

4. Results

After tuning the controller parameters using simulation, parameter was chosen

M = 80, T = 0.4s, T_s = 5ms

Figure 1 shows the simulation result.

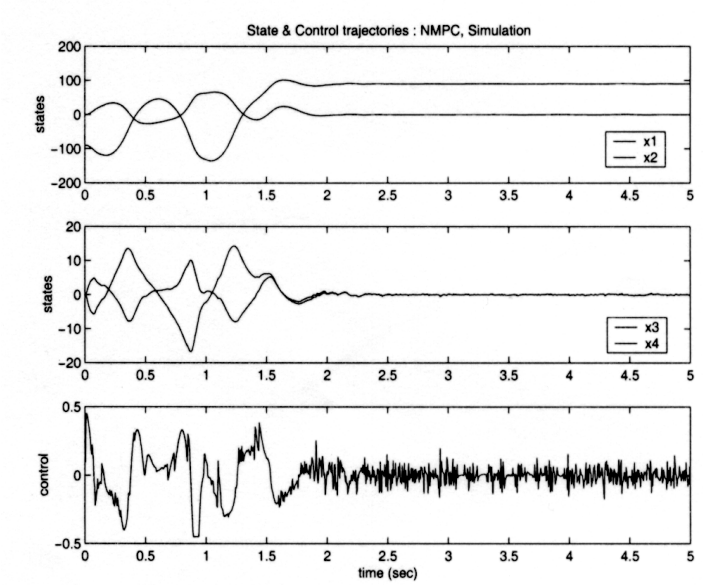

Fig.1. Experimental Response with NMPC

References

[1] C. C. Chen and L.Shaw. On receding horizon feedback control. *Automatica,* 18(3):349-352,1982

[2] S. S. Keeerthi and E. G. Gillbert. Optimal infinitehorizon feedback laws for a general class of constrained discrete-time systems : stability and moving-horizon approximation. *Journal of Optimization Theory and Applications*, 57(2):265-293, May 1988.

[3] D. Q. Mayne and H.Michalska. Receding horizon control of nonlinear system. *IEEE Trans. on Automatic Control*, 35(7):814-824, July 1990

[4] D. Q. Mayne and H.Michalska. Robust receding horizon contril of constrained nonlinear system. *IEEE Trans. on Automatic Control*, 38(11):1623-1633, November 1993

[5] A. Jadbabaie, J. Yu, and J. Hauser. Receding horzon control of the caltech ducted fan: A control lyapunov function approach. *In Proc. 1999 IEEE Conferencr on Control Applications*, 1999.

[6] S. L. Oliveira and M. Morari. Contractive model predictive control for constrained nonlinear system. *IEEE Trans. on Automatic Control*, 45(6):1053-1071, June 2000.

[7] F. Lizarralde, J. T. Wen, and l. Heu. Anew model predictive control strategy for affine nonlinear control system. In *Proc. 1999 American Control Conference*, pages 4263-4267, San Diego, CA, June 1999.