Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-6 (2016) 034–039

Teaching and Practicing Model Predictive Control

Daniel Honc*, Rahul Sharma K.*, Anuj Abraham,**
František Dušek*, Natarajan Pappa**

* Department of Process Control, Faculty of Electrical Engineering and Informatics,
University of Pardubice, Czech Republic
rahul.sharma@student.upce.cz,{daniel.honc,frantisek.dusek}@upce.cz
**Department of Instrumentation Engineering, Madras Institute of Technology Campus,
Anna University, Chennai, India
anuj1986aei@gmail.com, npappa@rediffmail.com

Abstract: How to explain Model Predictive Control (MPC) to students? How to practise it? The paper deals with chain of actions involving teaching, practicing and laboratory application of MPC at University of Pardubice in Czech Republic and at Anna University in India. Individual steps are presented and discussed with examples from educational experience – e.g. modelling and identification, derivation of MPC controller, simulations and laboratory applications. Every phase has a key and weak point as well. Desired results is that students understand better the theoretical concepts and they are able to apply predictive controllers at least for laboratory processes. Derivations and MATLAB scripts are available online.

Keywords: Model Predictive Control, MPC, modelling, system identification, optimization, teaching, experiments.

1. INTRODUCTION

Model predictive control is very popular and frequently used in the industry for optimal control of multivariable systems with constrains. The method is suitable for unstable or non-minimum phase systems, systems with dead-times, with different numbers of controlled and manipulated variables even for non-linear processes. The key feature is explicit use of a dynamical process model for controlled variable prediction at a future time horizon and calculation of a control actions to minimize a cost function. Future set-points or disturbances can be handled as well if available. The receding strategy concept means that at every sample time instant, only first the control action from the optimal vector is used and the horizon is shifted towards the future and the procedure is repeated again for updated system state. The various predictive algorithms differ amongst themselves in the model used to represent the process, the cost function to be minimized and optimization method.

The whole concept of MPC is straightforward and easy to understand but still enough general so special control objectives can be defined and fulfilled. Controller is designed in time domain, dynamical model of controlled process must be known and also some optimization method to get the solution is required. From this point of view, teaching and practicing of MPC is an important part of university education of process control engineers. Books about MPC are great but for master students this is quite expensive and maybe too much detailed source of information (Camacho and Bordons, 2007), (Rossiter, 2003), (Maciejowski, 2002), (Kouvaritakis and Cannon, 2016), (Rawlings and Mayne, 2009), (Wang, 2009), (Mareš and Hrnčiřík, 2012), (Mikleš and Fikar, 2004). Online books and tutorials can be good alternative for most of the students (Borrelli et al., 2015), (Rossiter 2014), (Bemporad, 2009), (Boyd, 2008), (Jay, 2005), (Pekař, 2010). The interesting task is how to explain MPC to students, how to practise their theoretical knowledge and what tools to use. Educational framework based on the Lego Mindstorms NXT robotic platform with two-wheeled inverted pendulum experiments was published in (Canale and Casale-Brunet, 2014). Adaptive cruise control with LabVIEW, National Intruments Robotics Starter Kit robot and code deployed on FPGA was presented in (Shakouri et al., 2013). (Richmond and Chen, 2012) created software package for teaching chemical engineering undergraduates similar to existing industrial MPC packages. MATLAB graphical user interface with MPC educational application was presented in (Yilmazlar and Kaplanoğlu, 2012). Our subject Automatic Control III aims to provide a MPC guidance to students and practice their theoretical knowledge. Students are getting not only new information but they are also practicing topics like modelling, identification, optimization, simulation, data acquisition and programming. Final year master students are learning MPC theory, they program controller functions in MATLAB, simulate control experiments first and at the end of the semester they apply their controllers to different laboratory systems. We are using GUNT level control and speed control training system because the systems are not too fast but fast enough, they are first and second order systems with low nonlinearity and we

10.1016/j.ifacol.2016.07.149

are able to use them from different environments like MATLAB, Simulink, LabView etc. (Honc, et al., 2014).

The outline of the paper is as follows. Basics of MPC theory is discussed in section 2. Laboratory system is presented in section 3. Modelling and identification is described in section 4, Simulation and experimental results are shown in section 5. Conclusions are given in section 6.

2. MPC BASIC THEORY

We want that the students understand MPC concept so we are not using tools like MATLAB's Model Predictive Control Toolbox (MATHWORKS, 2016), (jMPC Toolbox, 2016), (MPT Toolbox, 2016) or similar products. We are explaining step-by step predictive concept and deriving all necessary equations – complete documentation can be downloaded from MPC derivation. After MPC history overview and introduction we start with SISO system cost function formulation,

$$J = \sum_{j=1}^{N_2} r_j \left(y(k+j) - w(k+j)\right)^2 + \sum_{j=1}^{N_u} q_j \Delta u(k+j-1)^2 \quad (1)$$

where y is the controlled variable, w is the set-point, Δu is the manipulated variable increment, r_j are the penalization parameters for control errors and q_j are the penalization parameters for control incremets, N_2 is the length of horizon for following the set-point and N_u is the length of horizon for control actions (after N_u control changes the control action is kept constant).

Cost function expressed in matrix representation is

$$J = (\mathbf{Y} - \mathbf{W})^T \mathbf{R} (\mathbf{Y} - \mathbf{W}) + \mathbf{U}^T \mathbf{Q} \mathbf{U} \quad (2)$$

The next step is how to get predictions based on state-space and transfer function process model. The state-space algorithm is easier for the students (Honc and Dušek, 2013b). State-space model in discrete-time form

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,u(k) \\ y(k) &= \mathbf{C}\,\mathbf{x}(k) \end{aligned} \quad (3)$$

is converted to incremental form as described below,

$$\begin{aligned} \underbrace{\begin{bmatrix} \mathbf{x}(k+1) \\ u(k) \end{bmatrix}}_{\mathbf{x}_p(k+1)} &= \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{1\times nx} & 1 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ u(k-1) \end{bmatrix}}_{\mathbf{x}_p(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 1 \end{bmatrix}}_{\mathbf{N}} \Delta u(k) \\ y(k) &= \underbrace{\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{O}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ u(k-1) \end{bmatrix}}_{\mathbf{x}_p(k)} \end{aligned} \quad (4)$$

and the predictions in matrix form can be written as

$$\begin{aligned} \underbrace{\begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \\ \mathbf{y}(k+3) \\ \vdots \\ \mathbf{y}(k+N_2) \end{bmatrix}}_{\mathbf{Y}} &= \underbrace{\begin{bmatrix} \mathbf{ON} & 0 & \cdots & 0 \\ \mathbf{OMN} & \mathbf{ON} & \cdots & 0 \\ \mathbf{OM^2N} & \mathbf{OMN} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{OM}^{N_2-1}\mathbf{N} & \mathbf{OM}^{N_2-2}\mathbf{N} & \cdots & \mathbf{OM}^{N_2-N_u}\mathbf{N} \end{bmatrix}}_{G} \cdot \\ \underbrace{\begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \Delta\mathbf{u}(k+2) \\ \vdots \\ \Delta\mathbf{u}(k+N_u-1) \end{bmatrix}}_{\mathbf{U}} &+ \underbrace{\begin{bmatrix} \mathbf{OM} \\ \mathbf{OM^2} \\ \mathbf{OM^3} \\ \vdots \\ \mathbf{OM}^{N_2} \end{bmatrix}}_{\mathbf{F}_p} \mathbf{x}_p(k) \\ &\underbrace{\qquad\qquad}_{\mathbf{f}} \end{aligned} \quad (5)$$

In the past we were using different methods for transfer function model like Diophantine equations and state-space transfer function equivalent. Following derivation seems to be easiest for the students understanding. We are considering process and disturbance model as,

$$y(k) = \frac{B}{A} u(k) + \frac{C}{\Delta A} e(k) \quad (6)$$

where, $e(k) = y(k) - \hat{y}(k)$ is prediction error, and polynomials

$$\begin{aligned} A &= 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}, \\ B &= b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n} \\ C &= 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{nc} z^{-nc}. \end{aligned}$$

We introduce polynomial,

$$\tilde{A} = \Delta A = 1 + \tilde{a}_1 z^{-1} + \tilde{a}_2 z^{-2} + \ldots + \tilde{a}_n z^{-n} + \tilde{a}_{n+1} z^{-(n+1)}$$

From (6), we can derive

$$\begin{aligned} \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \tilde{a}_1 & 1 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\mathbf{A}_p} \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \vdots \\ \hat{y}(k+N) \end{bmatrix} &= \underbrace{\begin{bmatrix} b_1 & 0 & \cdots & 0 \\ b_2 & b_1 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & b_1 \end{bmatrix}}_{\mathbf{B}_p} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix} + \\ + \underbrace{\begin{bmatrix} -\tilde{a}_1 & -\tilde{a}_2 & \cdots & -\tilde{a}_{n+1} \\ -\tilde{a}_2 & -\tilde{a}_3 & \cdots & 0 \\ & & & \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}_m} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-n) \end{bmatrix} &+ \underbrace{\begin{bmatrix} b_2 & b_3 & \cdots & b_n \\ b_3 & b_4 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{B}_m} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-n+1) \end{bmatrix} + \\ + \underbrace{\begin{bmatrix} c_1 & c_2 & \cdots & c_{n_c} \\ c_2 & c_3 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{C}_m} \begin{bmatrix} e(k) \\ e(k-1) \\ \vdots \\ e(k-n_c+1) \end{bmatrix} & \end{aligned} \quad (7)$$