International Journal
Information Technology Applications (ITA)
Volume 10, Number 1, September 2021

MODEL PREDICTIVE CONTROL - APPLICATION FOR PHYSICAL MODEL

Štefan Kozák

Abstract:

Paper deals with the one of effective application of MPC control for a laboratory thermal–optical system. Laboratory model is realized by two channels: thermal channel and optical channel and their dynamical mathematical models can be described by a step-response model. The main goal of the proposed paper is design and application of the modified constrained dynamic matrix control algorithm for the considered laboratory process with supporting of Matlab-Simulink. Proposed and implemented dynamic matrix control algorithm is compared with conventional PID algorithm. Final part of paper deals with possible implementation of MPC for FPGA hardware realization.

Keywords:

Dynamic model, prediction, predictive control, optimization, process control.

ACM Computing Classification System:

Physical sciences and engineering, automatic control, integrated circuits.

Introduction

Predictive control has become popular over the past twenty years as a powerful tool in feedback control for solving many problems for which other control approaches have been proved to be ineffective. Predictive control is a control strategy that is based on the prediction of the plant output over the extended horizon in the future [13] which enables the controller to predict future changes of the measurement signal and to base control actions on the prediction.

Model predictive control (MPC) refers to a class of control algorithms that compute a sequence of control inputs based on an explicit prediction of outputs within some future horizon. The computed control inputs are typically implemented in a receding horizon fashion, meaning only the inputs for the current time are implemented and the whole calculation is repeated at the next sampling time. Therefore, one of the most important strengths of MPC is that it can consider the constraints of input and output variables that often exist in real industrial systems.

One of the most well-known MPC algorithms for the process control is dynamic matrix control (DMC), which assumes a step-response model (SRM) for the underlying system.

Predictive control has many advantages. Due to them, the predictive control approach has many applications. Examples include medical applications, civil engineering applications, and mechanical engineering applications as well as chemical and petrochemical engineering applications. A survey of industrial model predictive control technology is provided in many publications [9].

The paper is organized as follows. First, basic principles of predictive control and design of dynamic matrix controller are introduced in Section 1. Then, the laboratory thermal-optical system and identification of are described in Section 2. The experimental result from the testing of modified and designed DMC controllers are shown in Section 3 – experimental results. FPGA realization of MPC algorithm are described in Section 4. Summary and conclusions are given in last Section.

1 Dynamic Matrix Control

A Basic principles of predictive algorithms

Basic principles of predictive control [8]:

- specifying reference trajectory yref=w(k) and its prediction on the chosen horizon of prediction p (Fig.1)
- prediction of a plant output on the predefined time horizon (N = k + i), i.e. prediction of $\hat{y}(k+i)$ for $i = 1,\ldots,p$ (p is the prediction horizon length) in discrete steps based on real values of control input in the past steps $u(k+i)$ for $i = 0,\ldots, m-1$ (where $m < p$ is the length of the control horizon)
- computation of new control input based on the knowledge of the mathematical model and optimal cost index J
- minimization of cost and computation of control input which ensures that the predicted output tracks the reference trajectory
- correction of the prediction function error between measured and predicted variable.

The three basic elements of predictive control are the model, which describes the process, the goal defined by an objective function and constraints, and the optimization procedure. Parameters chosen by users are the prediction horizon, control horizon, parameters in the objective function and constraints.

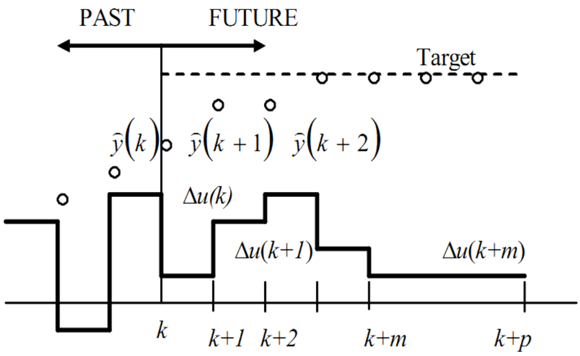


Fig.1. Basic principles of predictive algorithm.

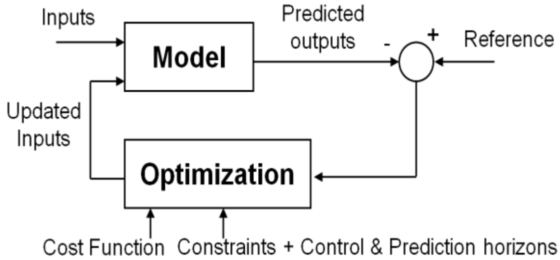


Fig.2. Block scheme of Model Predictive Control (MPC).

Process interactions and deadtimes can be intrinsically handled by model predictive control schemes such as dynamic matrix control (DMC).

The sequence of future control signals is computed by optimizing a given cost function [5].

B Design of dynamic matrix control

In 1979, Cutler and Ramaker of Shell oil Co. presented details of an unconstrained multivariable control algorithm, which they named Dynamic Matrix Control (DMC) [9]. It is evolved from a technique of representing process dynamics with a set of numerical coefficients [3].

The Dynamic matrix is used for projecting the future outputs. It is suitable for linear open loop stable process. The DMC technique is based on a step response model of the process.

The objective of the DMC controller is to drive the output to track the set point in the least squares sense including a penalty term on the input moves. This results in smaller computed input moves and a less aggressive output response [9].

Consider the single input single output (SISO) case, the step response model of the plant,

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t-i) \tag{1}$$

The disturbance at instant t along the horizon is,

$$\hat{x}(t+j/t) = \hat{x}(t/t) = e(t) \tag{2}$$

$$\mathrm{e}(t) = y_m(t) - \hat{y}(t/t) \tag{3}$$

The predicted value along the horizon will be:

$$\hat{y}(t+j/t) = \sum_{i=1}^{\infty} g_i \Delta u(t+j-i) + \hat{x}(t+j/t) \tag{4}$$

For constant disturbance the predicted value of output is,

$$\hat{y}(t+j/t) = \sum_{i=1}^{j} g_i \Delta u(t+j-i) + \mathrm{f}(t+j) \tag{5}$$

The second term of equation (5) is the free response, which does not depend on the future control actions. The prediction along the prediction horizon p and m control actions (j=1, ... , p) is [2]:

$$\hat{y}(t+p/t) = \sum_{i=1}^{p} g_i \Delta u(t+p-i) + \mathrm{f}(t+p) \tag{6}$$

Equation (5) can be written as,

$$\hat{y} = \mathrm{Gu} + \mathrm{f} \tag{7}$$

Equation (7) shows the relation between future output and control increments.

$$G = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m+1} \end{bmatrix} \tag{8}$$

The systems dynamic matrix G is made up of m (the control horizon) columns of the systems step response appropriately shifted down in order.

The predicted output with disturbances is,

$$\hat{y}_{\mathrm{d}} = \mathrm{G_d u_d} + \mathrm{f_d} \tag{9}$$

The cost function to be minimized including control effort is [10],

$$J = \sum_{N_1}^{N_2} \delta(j)[\hat{y}(t+j/t) - w(t+j)]^2 + \sum_{N_1}^{N_2} \lambda(j)[\Delta u(t+j-1)]^2 \tag{10}$$

Without constraints, cost functions is

$$J = ee^T + \lambda uu^T \tag{11}$$

The solution to this cost function can be obtained by computing the derivative of J and equating it to zero

$$u = (G^T G + \lambda I)^{-I} G^T (w - f) \tag{12}$$

In DMC control horizon and penalization factor are the tuning parameter. In least square formulation penalization factor is introduced to smoothen the control signal [4].

Standard QP at each sampling instant carries out the optimization. According to receding horizon concept, at time t, only the first input (Δu(t)) of the vector of future control increment or sequence (u) is actually applied to the plant. The remaining optimal inputs are discarded, and a new optimal control problem is solved at time t + 1[1].

When constraints are considered in input and output, the following equation must be added to the minimization [5], (i = 1, ..., Nc):

$$\sum_{j=1}^{N} c_{yj}^{i}\, \hat{y}(t+j/t) + c_{uj}^{i} u(t+j-1) + c^i \leq 0 \tag{13}$$

In DMC, the design of control is independent of the transport lag and it deals with constraints. In presence of disturbances, feed-forward can be easily implemented. It is robust to model error but the application is limited to open loop bounded input bounded output (BIBO) stable type of processes [6], [7]. For DMC algorithm to be closed loop stable, a long length of prediction horizon is required [9].

2 Thermal-Optical System

A Description of the thermal-optical system

The thermal-optical system uDAQ28 L/T (Fig. 3) is an experimental laboratory device aimed primarily for education of automatic control. The device allows for real time measurement and control of temperature and light intensity.

It can be connected to a computer via a universal serial bus and communication with Matlab environment is fully supported. It is multivariable system with three manipulated inputs and eight measured outputs.

System has three manipulated inputs: bulb voltage (0-5V) which represents heater and light source, fan voltage (0-5V) which can be used for temperature decreasing and voltage of led diode (0-5V) which represents another source of light.

On the output of the system is possible to measure seven variables: temperature insight the system (direct or filtrated), outsight temperature, light intensity (direct or filtrated), fan velocity and fan current.

Fig.3. Front view on the thermal-optical system.

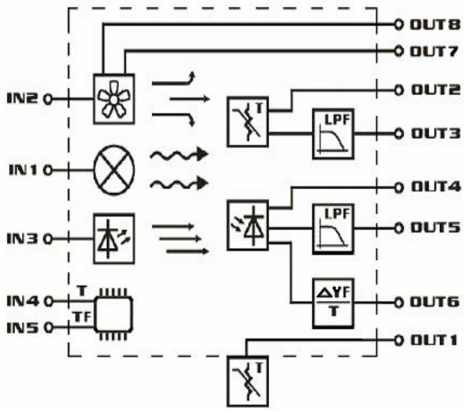


Fig.4. Structure of the thermal-optical system.

B Light intensity identification

The dynamics of the light intensity channel of the measured thermal-optical system can be determined from the response of the process to steps deterministic signal. The two-point algorithm approach was given for the process identification.

The step response of the light intensity channel has typical first-order characteristic with transport delay time constant.

$$G(s)=\frac{K_r}{T_p s+1}e^{-Ds} \tag{14}$$

The process gain is determined by dividing of the steady state output by the input set-point value. The time taken for the process output to reach 33% and 70% of the final steady state output is used to determine the time constant and the dead time based on solving the following simultaneous equations:

$$T_p = 1.245(t_{0.7} - t_{0.33}) \tag{15}$$
$$D = 1.498t_{0.33} - 0.498t_{0.7}) \tag{16}$$

The process gain and time constant was given from step response characteristics:

$$t_{0.33} = 0.558s,\ t_{0.7} = 0.398s, K_r = 4.103 \qquad (17)$$

Substituting these measured coefficients in relations (15) and (16) are obtaining the transfer function parameters of the identified model

$$G(s) = \frac{4.103}{0.1992s+1} e^{-0.3183s} \qquad (18)$$

The step responses of the compared identified first-order model and measured laboratory system are shown in (Fig.5).

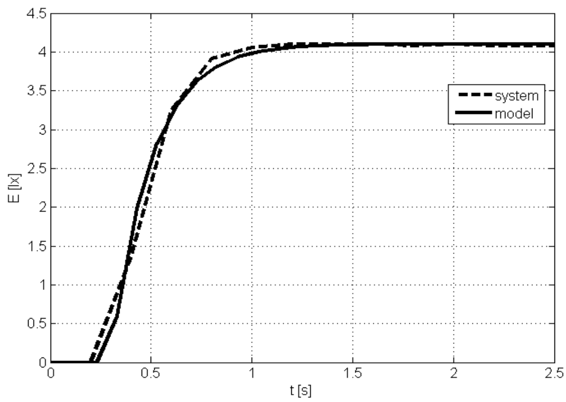


Fig.5. Step responses of the compared identified model and measured thermal-optical system.

3 Experimental Results

DMC control algorithm of the light intensity of the thermo-optical system was written in MATLAB. Figure 7 shows the DMC response of light intensity for a step change 10 in the set point of the. The result shows that DMC control the light intensity in the presence of set point change. The value of the controls horizon were taken 1, 2 and prediction horizon were taken as 10 and 20. A sampling interval of 0.1 second was chosen. DMC brings and maintains the controlled variable around the set point after some time.

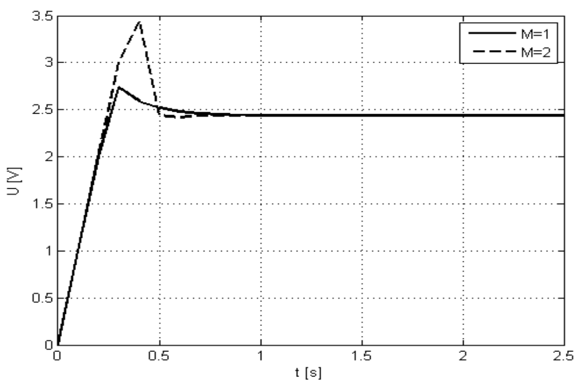


Fig.6. Time responses of the manipulated variables of two compared DMC controllers [12].

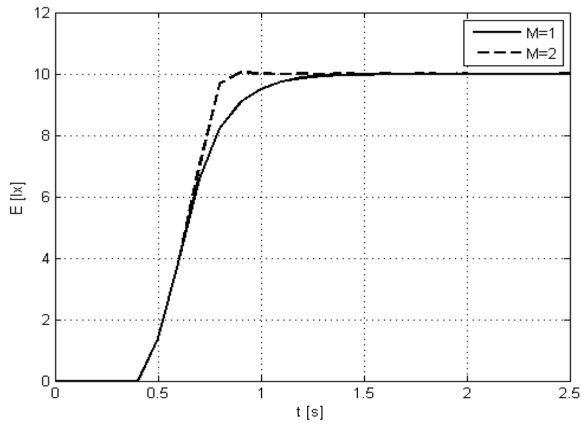


Fig.7. Time responses of the controlled variables of two compared DMC controllers [12].

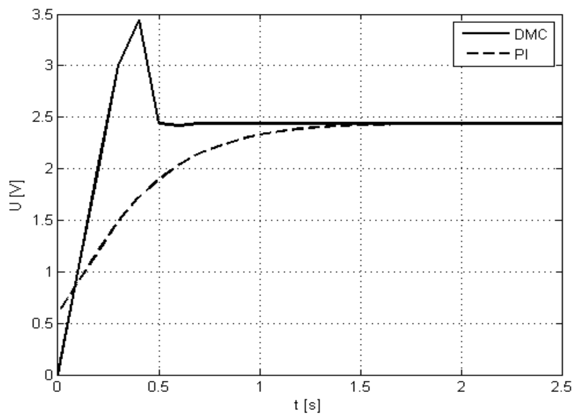


Fig.8. Time responses of the manipulated variables of DMC and PI controllers [12].

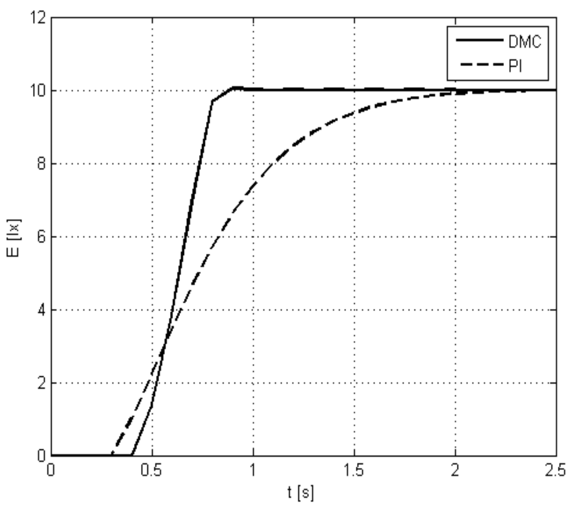


Fig.9. Time responses of the controlled variables two compared controllers, DMC and PI controller [12]

The comparison of designed DMC controller with conventional PI controller are shown in the figures on previous page. The parameters of the PI controller were designed on the base of the identified first-order model.

4 FPGA MPC Realization

The main factors to be considered when implementing MPC on reconfigurable hardware include computational speed, hardware resource usage. For a particular application, specific requirements on these factors need to be met and the final implementation is usually a compromise between all these factors. Hence, an effective and efficient rapid prototyping environment which allows for experimentation and verification of various algorithm configurations, architecture and implementation schemes would be useful [11].

The tools used in this study include a Xilinx Spartan-3A Starter Kit, the Xilinx ISE Design Suite and MATLAB/Simulink software. The Xilinx Spartan 3-A Starter Kit is a platform for evaluation and development of FPGA based applications. The platform includes a Xilinx Spartan-3A FPGA, external memory, programmable clocks, A/D and D/A converters, Ethernet, RS-232 port, etc. The core of Spartan-3A (Fig.11) development board is a Xilinx Spartan-3A FPGA chip which has 1.5 million logic gates and some useful on-chip resources such as multiplier and on-chip memory. Xilinx ISE design suite is an integrated environment for FPGA implementation. It provides a complete tool set which includes a compiler, a debugger, an optimizer and a simulator. MPC algorithm design and simulation,

MATLAB/Simulink provides an excellent platform for plant modeling, MPC algorithm design and simulation, Handel-C/MATLAB co-simulation and hardware-in-the-loop verification. Handel-C is a high-level FPGA implementation language with an ANSI C syntax and some hardware related language features such as parallel execution, channel communication, interface definition, etc. Compared with other hardware description languages such as VHDL, Handel-C is more convenient for rapid prototyping of MPC control algorithms.

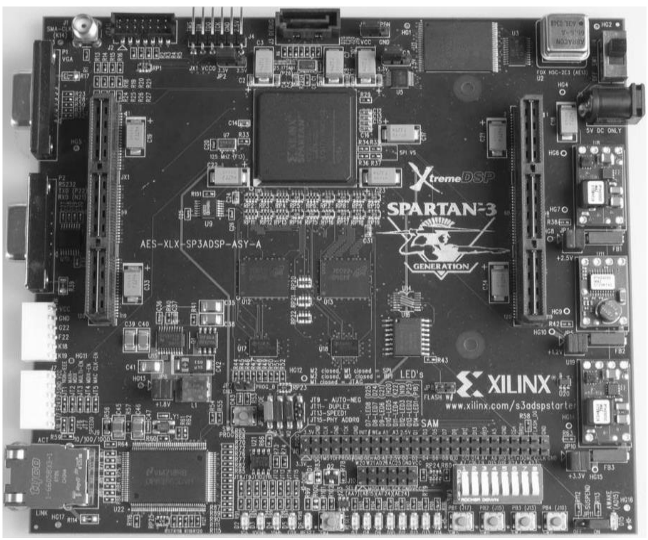


Fig.11. Xilinx Spartan-3A development board.

The main procedure of prototyping of our MPC algorithm on FPGA design is illustrated in (Fig.12).

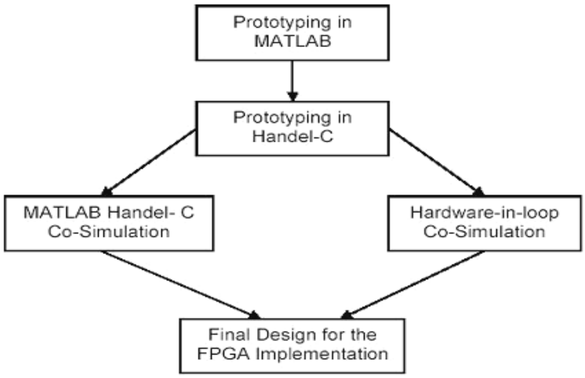


Fig.12. Prototyping of MPC control system [11].

In the following part we briefly describe the main steps in prototyping MPC into a FPGA implementation.

- Prototyping in MATLAB provides an excellent computation and simulation environment for designing and implementing control algorithms. The MPC algorithm is first prototyped in MATLAB code and then simulated and verified in the MATAB/Simulink environment.
- Prototyping in Handel-C code, the prototype MPC in the form of MATLAB code is translated into Handel-C code for FPGA realization. It is mapped, placed and routed by Xilinx ISE to a target FPGA. The Xilinx tool would report hardware resource usage and timing performance. If the results do not meet the specified requirements, design iterations would need to be carried out.
- Handel-C/MATLAB co-simulation, two options are available for algorithm verification: software or hardware verification. For software verification, the Handel-C code will be packaged into a DLL file and then be called by Simulink as a S-function.
- Hardware-in-the-loop verification, for hardware verification, the Handel-C code will be compiled into a bitstream file which will subsequently be down-loaded onto the FPGA on the Spartan-3A development board to perform the MPC calculations. A test suite can then be written to verify the MPC implementation on the FPGA.

Conclusion

In this paper the modification and design of dynamic matrix controller for laboratory thermo-optical system has been designed and verified on light intensity channel of the measured system. The verification of designed dynamic matrix controller was provided on test with conventional proportional integral controller. The result from the verification of the designed dynamic matrix controller in terms of performance for the identified model of the optical channel was very good. Currently we develop and verified different predictive methods by FPGA realisation and we would like to compare them with conventional model predictive control algorithm for fast processes.

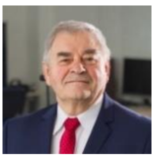
Acknowledgement

This paper has been supported by the Slovak Scientific Grant Agency, Grant No. 1/1105/11.

References

[1] A. Bemporad, M. Morari (1999) Model Predictive Control: A Survey, *Robustness in identification and control*, vol. 245.
[2] E.F.Camacho, C. Bordons (2004) *Model Predictive Control*, Springer, London.
[3] C.R. Cutler, B. Ramaker (1980) Dynamic Matrix Control - A computer control Algorithm, *Proceedings of the 1980 Joint Automatic Control Conference*, vol. WP5-13.
[4] R.M.C. De Keyser, PH.G.A. Van De Velde (1988) A comparative Study of Self-adaptive Long range Predictive Control Methods, *Automatica*, vol. 24, no. 2, pp. 149-163.
[5] C.E. Garcia, D.M. Prett, M. Morari (1989) Model Predictive Control: Theory and Practice a Survey, *Automatica*, vol. 25, no. 3, pp. 335-348.
[6] J.C. Hung (1996) *Practical techniques for industrial control*, IEEE, pp. 5-10.
[7] P. Lundstrom, J.H. Lee, M.M.(1999) S.S. Limitations of DMC, *Computers Chem. Eng.*, vol 19, no 4, pp. 409-421.
[8] J.A. Roubos, S. Mollov, R. Babuska, H.B. Verbruggen (1999) Fuzzy model-based predictive control using Takagi-Sugeno models, *International Journal of Approximate Reasoning 22*, pp. 3-30.
[9] S.J. Quin, T.A. Badgwell (2003) A survey of industrial model predictive control technology, *Control Engineering Practice*, vol. 11, pp. 733-764.
[10] Xie, K. Sang, J.T.Y. Lin (1997) A New Dynamic Matrix Control Algorithm Based on the FNN TS Fuzzy Model, IEEE International Conference on Intelligent Processing Systems, pp. 317-321.
[11] K.V. Ling, S.P. Yue and J.M. Mackiejowski (2011) A FPGA Implementation od Model Predictive Control, *IFAC World Congress*, vol. 18, Milano, pp. 1930-1935.
[12] P. Guzmický, Kozák Š. (2012) Dynamic matrix control of thermal-optical laboratory system, *Portuguese Conference on Automatic Control – CONTROLO*, vol. 10, Madeira.
[13] C. Georgescu, A. Afshari and G. Bornard (1993) Fuzzy predictive PID controllers: An heating control application, *2th IEEE Conference on Fuzzy System*, pp. 1091-1098.

Authors



prof. Ing. Štefan Kozák, PhD.
Faculty of Informatics, Pan-European University, Bratislava, Slovakia
stefan.kozak@paneurouni.com
Currently at the Institute of Applied Informatics at the Faculty of Informatics, Pan-European University in Bratislava. His research interests include system theory, linear and nonlinear control methods, numerical methods and software for modeling, control, signal processing, IoT, IIoT and embedded intelligent systems for digital factory in automotive industry.