ADAPTIVE CONSTRAINED GENERALIZED PREDICTIVE CONTROL BASED ON NEURAL NETWORK

Petr NEPEVNÝ, Doctoral Degree Programme (1)
Dept. of Control and Instrumentation, FEEC, BUT
E-mail: xnepev00@stud.feec.vutbr.cz

Supervised by: Prof. Petr Pivoňka

ABSTRACT

This paper presents solution of constrained Generalized Predictive Control (GPC) with autoregressive model based on neural network. Neural model is able to observe system changes and adapt itself. Algorithm was implemented in MATLAB-Simulink with aspect of future implementation to Programmable Logic Controller (PLC) B&R. The usage possibilities of this approach were tested on mathematical and physical models in soft-real-time realization. Constrained GPC algorithm was compared with classical PSD controller and advantages and disadvantages of predictive control are shown.

1 INTRODUCTION

The beginning of predictive regulators was in late 60s, when we can find terms as optimal system, optimal control. In 70s, after the first optimal controllers like LQR (Linear Quadratic Regulator), the branch of predictive controllers has been separated. Since that a lot of predictive algorithms as MPHC (Model Predictive Heuristic Control), DMC (Dynamic Matrix Control), GPC (Generalized Predictive Control) were published. All predictive methods have the same base, but difference is in used model of system and disturbance and used cost function.

The core of predictive regulator is plant model, which is use to predict a future value of system output. If we know predicted output, we can compute future error (if we know future reference trajectory) and this knowledge is used to compute optimal future action. Control action is optimal in accordance to the cost function, which is used. A number of predicted steps is known as prediction horizon. Prediction horizon is shifted in the time every sampling period (receding horizon).

2 GENERALIZED PREDICTIVE CONTROL

GPC is one of the most favourite predictive control method. This method is popular not only in industry, but also at universities. It was first published in 1987. The authors wanted to find one universal method to control different systems. Generalized predictive control is applicable to the systems with non-minimal phase, unstable systems in open loop, systems with unknown or varying dead time and systems with unknown order.

2.1 UNCONSTRAINED GPC

We assume Auto Regressive Moving Average (ARMA) model, whose output depends on past systems inputs and outputs. MPC algorithm computes future action increments vector by minimize the cost function *J*.

$$J = E\left\{ \sum_{j=1+d}^{P+d} \left((\hat{y}(t+j\,|\,t) - w(t+j\,|\,t) \right)^2 + \lambda \sum_{j=0}^{M-1} \left(\Delta u(t+j\,|\,t) \right)^2 \right\} \tag{1}$$

where $\hat{y}(t+j\,|\,t)$ is predicted output of system in *j*-th prediction step in discrete time *t*, $w(t+j\,|\,t)$ is reference trajectory in coincidence points, $\Delta u(t+j\,|\,t)$ is *j*-th increment of control action, *P* is prediction horizon, *M* is control horizon, *λ* is cost constant and *d* is dead time. Variable *t* expresses discrete time – the step of control algorithm. Requirement $M \le P$ is always hold valid.

The aim of predictive control algorithm is to compute future control actions so that the systems output tracks the reference trajectory with minimal error. The cost function also contains a cost of control action increment (it limits actuator damage).

Criterion (1) can be rewritten to matrix form:

$$\mathbf{J} = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{H}\mathbf{u} + \mathbf{b}^{\mathrm{T}}\mathbf{u} + \mathbf{f_0} \tag{2}$$

where $\mathbf{H} = \mathbf{2}\left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I}\right)$, $\mathbf{b}^{\mathrm{T}} = \mathbf{2}(\mathbf{f} - \mathbf{w})^{\mathrm{T}}\mathbf{G}$, $\mathbf{f_0} = (\mathbf{f} - \mathbf{w})^{\mathrm{T}}(\mathbf{f} - \mathbf{w})$

u … vector of action increments $\mathbf{u} = \left(\Delta u(t) \quad \Delta u(t+1) \quad \ldots \quad \Delta u(t+P-1)\right)$

w … vector of reference trajectory in coincidence points

$$\mathbf{w} = \left(w(t+1) \quad w(t+2) \quad \ldots \quad w(t+P)\right)$$

f … free response of model $\mathbf{f} = \left(f(t+1) \quad f(t+2) \quad \ldots \quad f(t+P)\right)$

G … matrix of dynamics

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{M-1} & g_{M-2} & g_{M-3} & \cdots & g_0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{P-1} & g_{P-2} & g_{P-3} & \cdots & g_{P-M} \end{bmatrix}$$

where element g_j is *j*-th coefficient of models step response.

Minimum of the cost function (2) is obtained by making the gradient of **J** equal to zero. The result is equation (3) for computation the future control action increments vector.

$$\mathbf{u} = -\mathbf{H}^{-1}\mathbf{b} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}(\mathbf{w} - \mathbf{f}) = \mathbf{K}(\mathbf{w} - \mathbf{f}) \tag{3}$$

Only the first element of vector **u** is used for control. In next sampling period, the new control sequence is computed. Basic control loop with GPC is shown in figure 1.

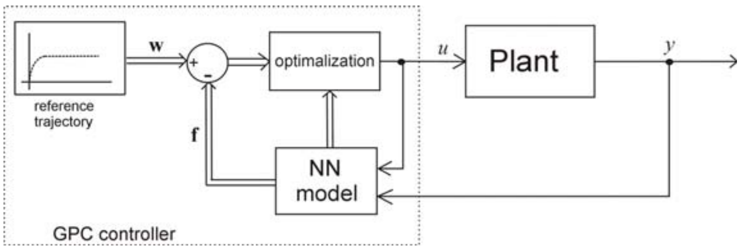

Fig. 1: *Control loop with predictive controller (GPC)*

2.2 CONSTRAINS IMPLEMENTATION

It is impossible to found solution of constrain GPC problem analytically, but it must be solved numerically every sampling period. This problem can be solved by a quadratic programming. We have quadratic cost function (2), where f_0 is constant vector. The constants don't have an effect for quadratic programming result, then we can use cost function in form (4). Constrains are defined by equation (5).

$$\mathbf{J} = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{H}\mathbf{u} + \mathbf{b}^{\mathrm{T}}\mathbf{u} \tag{4}$$

$$\mathbf{R}\mathbf{u} \le \mathbf{c} \tag{5}$$

Equation (5) can be rewritten to the form:

$$\begin{bmatrix}\mathbf{R_{du}} & -\mathbf{R_{du}} & \mathbf{R_u} & -\mathbf{R_u}\end{bmatrix}^{\mathrm{T}} u \le \begin{bmatrix}\mathbf{c_{duMax}} & \mathbf{c_{duMin}} & \mathbf{c_{uMax}} & \mathbf{c_{uMin}}\end{bmatrix}^{\mathrm{T}}$$

where constrain of control action increment $\Delta u_{\min} \le \Delta u(t+j\,|\,t) \le \Delta u_{\max}$ is realized by:

$$\mathbf{R_{du}} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \quad \text{and} \quad \mathbf{c_{duMax}} = \begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \Delta u_{\max} \\ \cdots \end{bmatrix} \quad \text{and} \quad \mathbf{c_{duMin}} = \begin{bmatrix} \Delta u_{\min} \\ \Delta u_{\min} \\ \Delta u_{\min} \\ \cdots \end{bmatrix}$$

and constrain of control action $u_{\min} \le u(t+j\,|\,t) \le u_{\max}$ is realized by:

$$\mathbf{R_u} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 1 & 1 & 0 & \cdots \\ 1 & 1 & 1 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \quad \text{and} \quad \mathbf{c_{uMax}} = \begin{bmatrix} u_{\max} \\ u_{\max} \\ u_{\max} \\ \cdots \end{bmatrix} \quad \text{and} \quad \mathbf{c_{uMin}} = \begin{bmatrix} u_{\min} \\ u_{\min} \\ u_{\min} \\ \cdots \end{bmatrix}$$

Size of matrices $\mathbf{R_{du}}$ and $\mathbf{R_u}$ is $P \times P$ and length of all vectors **c** is *P*.

Problem of quadratic programming solves MATLABs function QUADPROG, which can be used in form **u** = QUADPROG (**H**, **b**, **R**, **c**). The result is vector **u** (future control action increments). In current time, we used only the first element of vector **u** ($\Delta u(1)$).

3 NEURAL NETWORK MODEL

Neural network (feed-forward perceptron) is used for realizing ARMA model. For learning, the Back-Propagation method is used. There is used batch learning procedure with fixed-sized buffer of training patterns. Patterns in the training set are choose by filtering, which means storing pattern every *n*-th sampling time. Elements in buffer are shifted in time. Using of time-filter is suitable, if we have short sampling period, because it makes the training set smooth and more reliable.

4 APPLICATION

The base of adaptive predictive controller is neural network model, which substitutes 3-th order ARMA model. In this case the neural network has six inputs (past system inputs and outputs) and one output (prediction of system output). For starting setup of neural networks weights the off-line measured data are used. During control process the training set is updated and neural model is learned by Back-Propagation algorithm. Then the neural network model is able to tracks the plants changes – GPC is adaptive. The coefficients of the model are used for calculation of future action vector. At first, the unconstrained GPC action is computed using equation (3). Then we can apply constrains (saturation) of control action and action increment. If we want to compute optimal constrained GPC, we can use unconstrained GPC control vector as a starting point for quadratic programming.

The algorithm was tested on mathematical models and then it was used for physical model control**.** Control and identification algorithms were implemented into S-function in MATLAB and control loop was created in MALAB-Simulink. The physical model control was realized by connection PC and physical model via PLC B&R. For connection the Ethernet was used. Sampling period for communication and control was 0.2 s.

For relevant comparison of GPC and classical PSD controller, we find PSD parameters by the process of minimize the cost function, which is similar to (1). This problem can be solved using MATLAB function FMINS. The results of physical model control are shown in figure 2, where is the comparison of constrain GPC with prediction and PSD controller. Used physical model can be approximately describe by transfer function F_1. In time 70 s was model changed (transfer function F_2).

$$F_1(s) = \frac{1}{(s+1)(3s+1)(7s+1)} \qquad \longrightarrow \qquad F_2(s) = \frac{1.5}{(s+1)(3s+1)(s+1)}$$

For evaluation of results the criterions were used:

$$c_D = \sum_t (\Delta u(t))^2 \qquad c_E = \sum_t (u(t))^2 \qquad c_Q = \sum_t (y(t) - w(t))^2$$

c_D…criterion of actuator Damage; c_E…criterion of Energy; c_Q…criterion of Quadratic error

	constrain GPC without prediction		**constrain GPC with prediction**		unconstrain GPC without prediction		unconstrain GPC with prediction		PSD		average
	abs.val.	rel.val.	abs.val.	rel.val.	abs.val.	rel.val.	abs.val.	rel.val.	abs.val.	rel.val.	
c_D	350	*0,46*	**298**	*0,39*	1452	*1,92*	1248	*1,65*	425	*0,56*	755,3
c_E	**10830**	*0,95*	11350	*0,99*	11600	*1,01*	11920	*1,04*	11460	*1,00*	11432,8
c_Q	1708	*1,40*	525	*0,43*	1701	*1,39*	**479**	*0,39*	1706	*1,39*	1224,4
Σ	-	2,81	-	**1,82**	-	4,33	-	3,09	-	2,96	-

Tab. 1: *Comparing of various GPC and PSD controller*

Table 1 presents comparison of various GPC and PSD controller. Absolute value (abs. val.) means real computed value of criterion and relative value (rel. val.) means absolute value divided by average absolute value. Main criterion for controller evaluation is sum of relative values of all criterions.

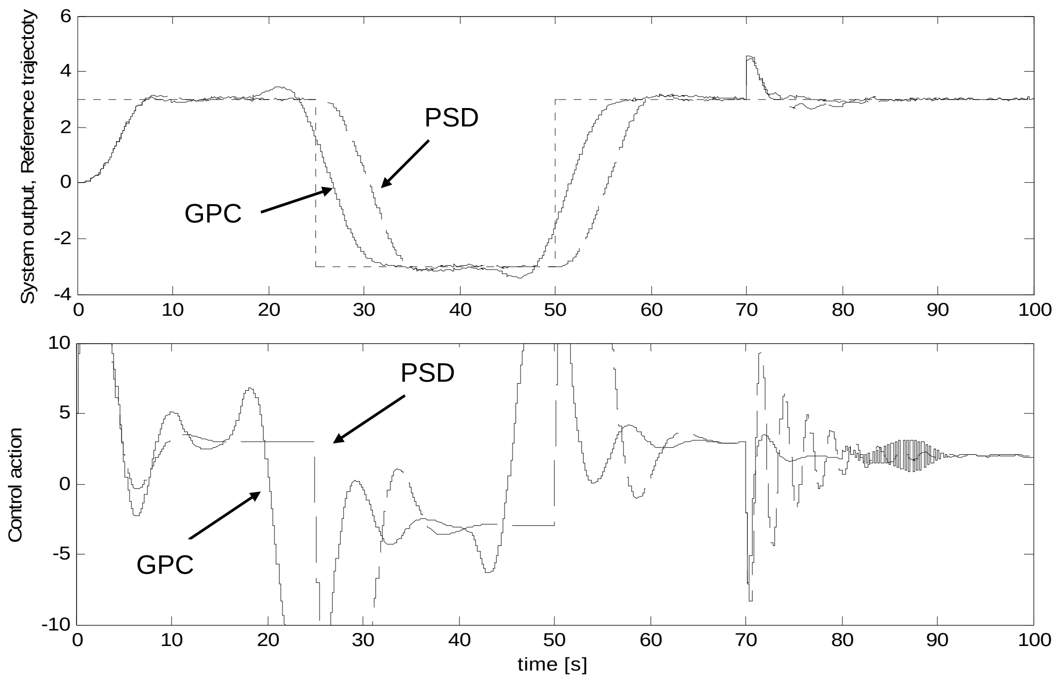

Fig. 2: *Comparison of constrain GPC with prediction a PSD controller*

5 CONCLUSIONS

In this paper the application of constrained predictive controller with neural network model is shown. Table 1 demonstrates, that (in accordance to criterion) the constrain GPC with prediction is the best. But if classic PSD controller is sets by similar criterion as GPC, the results are only little different. Main advantage of predictive controller are, that it reacts faster to changes of desired value then PSD controller (if it has a knowledge about future reference trajectory), the constrains can be implemented in control algorithm and it is adaptable to plant changes. This new predictive algorithm is going to be full implemented in PLC for real-time control application.

ACKNOWLEDGEMENTS

The paper has been prepared as a part of the solution of GAČR project No. 102/03/H116 and with the partially support of the research plan FRVŠ: F1/2902/2005.

REFERENCES

[1] Camacho, E. F., Bordos, C.: Model Predictive Control, Springer, ISBN 3-540-76241-8

[2] Cichocki, A., Unbehauen R.: Neural networks for optimization and signal processing, Wiley, 1994. ISBN 0-471-93010-5