COMPARISON OF CONSTRAINED AND UNCONSTRAINED GENERALIZED PREDICTIVE CONTROL

Vojtěch Mikšánek
Doctoral Degree Programme (2), FEEC BUT
E-mail: miksanek.v@phd.feec.vutbr.cz

Supervised by: Petr Pivoňka
E-mail: pivonka@feec.vutbr.cz

ABSTRACT

This paper deals with one of the advanced control algorithms – Generalized Predictive Control (GPC), which is one of the most popular methods of Model Predictive Control (MPC). This control algorithm optimizes control action and output error within a time horizon and can be successfully used to control systems with limited inputs and outputs. A neural network system model is used for prediction of future system behavior. The control algorithm was implemented in the simulation environment MATLAB/Simulink and tested on mathematical and physical models.

1. INTRODUCTION

The beginnings of Model Predictive Control (MPC) date back to the 1970s. Model Predictive Control integrates optimal control, dead time processes control, multivariable control and future references when available. The MPC is not a specific control strategy but an ample range of control methods where the control signal is obtained by minimizing an objective function. The model is the cornerstone of the MPC wherefore it is necessary to obtain the best possible model, and that can be done by using Neural Network (NN).

Model Predictive Control algorithms usually assume that all signals have an unlimited range, although real processes have constraints – limited range of action, limited action increment, constrained output, etc. For this reason, it is necessary to use generalized predictive controllers to cope with constrained inputs (amplitude and increment).

2. GENERALIZED PREDICTIVE CONTROL

Generalized Predictive Control (GPC) is one of the most popular methods of predictive control. It was proposed in 1987 [2] and has become one of the most popular MPC methods [1] in both industry and academia. The generalized predictive control algorithm consists in applying a control sequence that minimizes a cost function (1).

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

where $\hat{y}(t+j\,|\,t)$ is the predicted system output in the j-th prediction step in discrete time t, $w(t+j)$ is the reference trajectory, $\Delta u(t+j)$ is the j-th increment of control action, P is the predicted horizon, M is the control horizon, λ is the cost constant and d is delay. The first term considers the predicted error and the second term considers penalized future control increments.

The criterion (1) can be rewritten to a matrix form [1]:

$$J(\mathbf{u}) = (\mathbf{Gu} + \mathbf{f} - \mathbf{w})^{\mathrm{T}}(\mathbf{Gu} + \mathbf{f} - \mathbf{w}) + \lambda \cdot \mathbf{u}^{\mathrm{T}}\mathbf{u} \tag{2}$$

where $\mathbf{f}$ is the vector of the free response of a system on the prediction horizon, $\mathbf{w}$ is the vector of future references.

$$\mathbf{u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+\mathrm{M}-1) \end{bmatrix} \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(t+d+1\,|\,t) \\ \hat{y}(t+d+2\,|\,t) \\ \vdots \\ \hat{y}(t+d+\mathrm{P}\,|\,t) \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w(t+d+1\,|\,t) \\ w(t+d+2\,|\,t) \\ \vdots \\ w(t+d+\mathrm{P}\,|\,t) \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f(t+d+1\,|\,t) \\ f(t+d+2\,|\,t) \\ \vdots \\ f(t+d+\mathrm{P}\,|\,t) \end{bmatrix}$$

$$\hat{\mathbf{y}} = \mathbf{Gu} + \mathbf{f}$$

$\mathbf{G}$ is the matrix of dynamics, for linear causal systems:

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

where element g_j is j-th coefficient of model step response (3).

$$g_j = -\sum_{i=1}^{j} a_i g_{j-i} + \sum_{i=0}^{j} b_i \tag{3}$$

The cost function minimum (2) is obtained by making the gradient of J equal to zero [1]. The result is equation (4), which is used for computation of the future control action increments vector.

$$\mathbf{u} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}(\mathbf{w} - \hat{\mathbf{y}}) \tag{4}$$

$$\Delta u(t) = \mathbf{k}(\mathbf{w} - \hat{\mathbf{y}}) \tag{5}$$

Where $\mathbf{k}$ is the first row of the matrix $\left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}$. Only the first increment of control action is used for control (5).

2.1. CONSTRAINTS IMPLEMENTATION

The GPC, which was described previously, consider all signals have an unlimited range, but this is not realistic because in practice all processes have constraints. A control action increment limits can be described by equation

$$\Delta u_{\min} \leq u(t) - u(t-1) \leq \Delta u_{\max} \tag{6}$$

and control action amplitude limits can be described by equation

$$u_{\min} \leq u(t) \leq u_{\max} \tag{7}$$

Now, an optimization problem with a quadratic cost function (1) and linear constraints (6) and (7) can be solved by Quadratic Programming (QP). The criterion (1) has to be rewritten to equation

$$J(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{H}\mathbf{u} + \mathbf{b}^{\mathrm{T}}\mathbf{u} + \mathbf{f}_0 \tag{8}$$

where

$$\mathbf{H} = 2\left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I}\right)$$

$$\mathbf{b}^{\mathrm{T}} = 2\left(\mathbf{f} - \mathbf{w}\right)^{\mathrm{T}}\mathbf{G}$$

$$\mathbf{f}_0 = \left(\mathbf{f} - \mathbf{w}\right)^{\mathrm{T}}\left(\mathbf{f} - \mathbf{w}\right)$$

And the constraints (6) and (7) can be rewritten to:

$$\mathbf{A}\mathbf{u} \leq \mathbf{c} \tag{9}$$

Where

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{T} \\ -\mathbf{T} \end{bmatrix}\mathbf{u} \leq \begin{bmatrix} \mathbf{\Delta u}_{\max} \\ \mathbf{\Delta u}_{\min} \\ \mathbf{u}_{\max} - \mathbf{u}_{t-1} \\ -\mathbf{u}_{\min} + \mathbf{u}_{t-1} \end{bmatrix} \tag{10}$$

Where **I** is the identity matrix and **T** is the low triangular matrix as is shown in next equations.

$$\begin{array}{cc} \Delta u(t) \leq \Delta u_{\max} & -\Delta u(t) \leq -\Delta u_{\min} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \\ \vdots & \vdots & & \ddots \end{bmatrix}\mathbf{u} \leq \begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 & \cdots \\ 0 & -1 & 0 & \cdots \\ 0 & 0 & -1 & \\ \vdots & \vdots & & \ddots \end{bmatrix}\mathbf{u} \leq \begin{bmatrix} \Delta u_{\min} \\ \Delta u_{\min} \\ \Delta u_{\min} \\ \vdots \end{bmatrix} \\ \mathbf{I}\mathbf{u} \leq \mathbf{\Delta u}_{\max} & -\mathbf{I}\mathbf{u} \leq -\mathbf{\Delta u}_{\min} \end{array} \tag{11}$$

$$\begin{array}{cc} u(t) \leq u_{\max} & u(t) \geq u_{\min} \\ u(t-1) + \Delta u(t) \leq u_{\max} & -u(t-1) - \Delta u(t) \leq -u_{\min} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 1 & 1 & 0 & \cdots \\ 1 & 1 & 1 & \\ \vdots & \vdots & & \ddots \end{bmatrix}\mathbf{u} \leq \begin{bmatrix} u_{\max} - u(t-1) \\ u_{\max} - u(t-1) \\ u_{\max} - u(t-1) \\ \vdots \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 & \cdots \\ -1 & -1 & 0 & \cdots \\ -1 & -1 & -1 & \\ \vdots & \vdots & & \ddots \end{bmatrix}\mathbf{u} \leq \begin{bmatrix} -u_{\min} + u(t-1) \\ -u_{\min} + u(t-1) \\ -u_{\min} + u(t-1) \\ \vdots \end{bmatrix} \\ \mathbf{T}\mathbf{u} \leq \mathbf{u}_{\max} - \mathbf{u}_{t-1} & -\mathbf{T}\mathbf{u} \leq -\mathbf{u}_{\min} + \mathbf{u}_{t-1} \end{array} \tag{12}$$

3. APPLICATION OF CONSTRAINED AND UNCONSTRAINED GPC

Both GPC algorithms were written in MATLAB and the model was obtained by using a neural network with the Levenberg-Marquardt training algorithm. To test of the control algorithm on a physical model, real-time communication is used between MATLAB/Simulink and PLC is via Ethernet using a communication client. The analog model, which contains operational amplifiers, resistors and capacitors, represents the third order process $F(s) = \dfrac{1}{(10s+1)(s+1)^2}$.

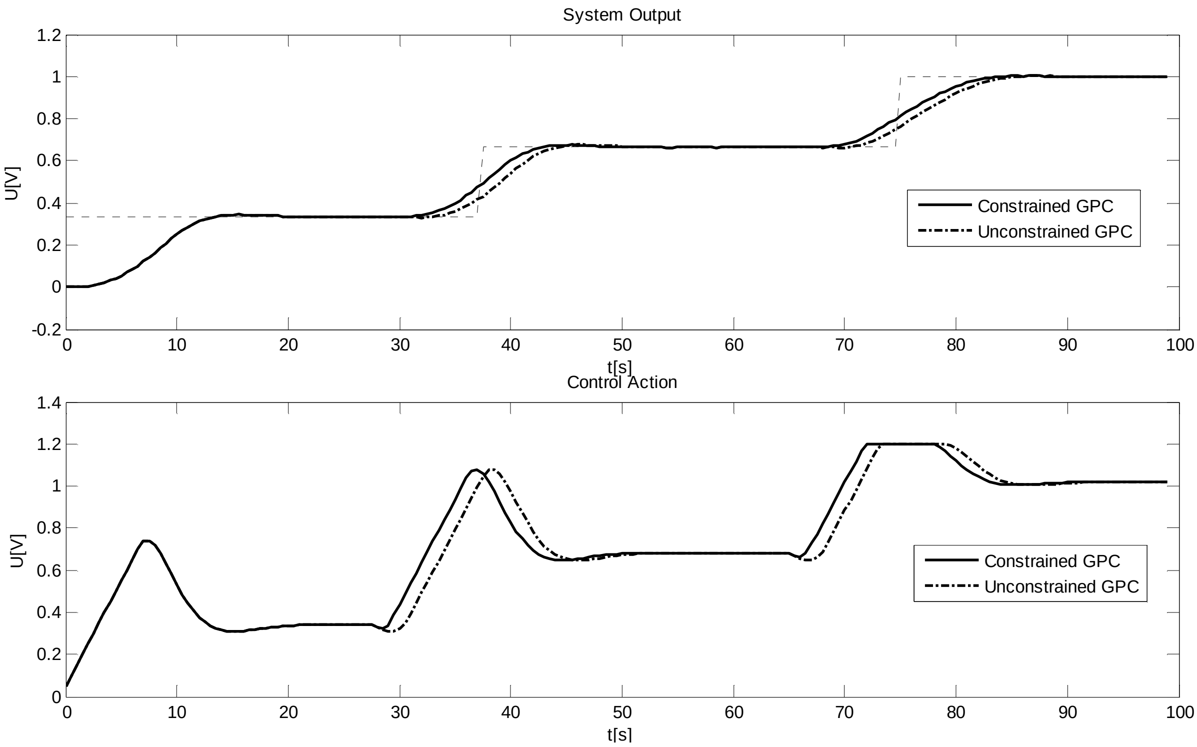

Fig. 1: Comparison of Constrained and Unconstrained GPC

Criterion \ controller	Constrained GPC	Unconstrained GPC
$c_{\mathrm{Q}} = \sum \left(y(t) - w(t)\right)^2$	1.85	2.05
$c_{\mathrm{E}} = \sum \left(u(t)\right)^2$	121	119
$c_{\mathrm{D}} = \sum \left(\Delta u(t)\right)^2$	0.134	0.135

Tab. 1: Quality of regulation Constrained and Unconstrained GPC.

Fig. 1 shows a comparison of constrained and unconstrained GPC. The GPC parameters were $\mathrm{P} = 20$, $\mathrm{M} = 10$, $\lambda = 0.5$ and the control action increment constraint

($\Delta u_{max} \leq 0.05$, $\Delta u_{min} \geq -0.05$) and control action amplitude constraints ($u_{max} \leq 1.2$) are applied to both GPC outputs.

In the first case, the unconstrained GPC was computed analytically by using equation (4) and after that its output ($\Delta u(t)$, $u(t)$) was limited. On the contrary, constrained GPC optimizes the cost function (8) with constraints (9). The problem of quadratic programming solved MATLAB function *quadprog*.

4. CONCLUSION

This paper shows the comparison of constrained and unconstrained Generalized Predictive Control. This comparison is shown on an analog model and the control action increment constraint and control action amplitude constraints are applied to both GPC outputs.

The comparison of both GPCs is shown in **Fig. 1**, and quality of regulation is compared in **Tab. 1**. The responses show the main advantage of constrained GPC - optimizing control actions with respect to actuator and/or process limits (the control actions increment and the control action amplitude were constrained). Such a GPC controller can track the reference trajectory better, and its quadratic output error criterion (**Tab. 1**) is about 20% lower in comparison with the unconstrained GPC.

GPC is an effective tool for the control of many processes such as processes with input, output or state constraints, processes with delays and processes with known reference trajectory as well.

ACKNOWLEDGEMENT

The paper has been prepared as a part of the solution of the Czech Science Foundation GAČR project No. 102/06/1132 Soft Computing in Control and by the Czech Ministry of Education in the frame of MSM MSM0021630529 Intelligent Systems in Automation.

REFERENCES

[1] Camacho, E. F. – Bordons, C.: Model Predictive Control, London, Springer 1999, ISBN 3-540-76241-8

[2] Clarke, D. W. – Mohtadi, C. – Tuffs, P. S.: Generalized Predictive Control – Part I. The Basic Algorithm, Automatica, 23, s. 137-148 (1987)

[3] Hristev, R. M.: The ANN Book, GNU Public License, 1998

[4] Sunan, H. – Kiong, T. K. – Heng, L. T.: Applied Predictive Control, London, Springer 2002, ISBN 1-85233-338-3

[5] Vychodil, H.: Generalized Predictive Control with a Non-linear Autoregressive Model. Automatic Control Modeling and Simulation ACMOS'05. Praha: WSEAS, 2005, pp. 85 - 89, ISBN 960-8457-12-2

[6] Pivoňka, P. - Nepevný, P.: Generalized Predictive Control with Adaptive Model Based on Neural Networks. In Proceedings of the WSEAS Conferences NN'05, FS'05, EC'05. Lisabon: WSEAS, 2005, pp. 1 - 5, ISBN 960-8457-24-6