PREDICTIVE CONTROL – ALGORITHMS, TESTS AND WEB PUBLICATION VIA MATLAB - SIMULINK ENVIRONMENT (R 14)

Květoslav Belda and Pavel Píša[*]
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Department of Adaptive Systems
[*]Czech Technical University in Prague, Faculty of Electrical Engineering
Department of Control Engineering

Abstract: Predictive control is one of control strategies, which optimize control actions within certain time range. Control actions are not designed only for current state of the system, but also they are optimized for the following time instants. Presented paper makes simple overview of Predictive control algorithms and their tests under MATLAB-SIMULINK environment. Furthermore, it presents Web publication of the MATLAB scripts and functions and SIMULINK schemes used for Predictive control.

1. Introduction

Most of control approaches do not take into account a possible future behavior of the process caused by instant control actions. They generate the actions from previous and topical states of the controlled system. Predictive control, on the other hand, is based on optimization that combines utilization of both previous behavior (feed-back) and behavior predicted over some horizon (feed-forward). To predict the future, some models are used. These models describe relations of system states, outputs and current control actions - inputs. In this paper, the models are assumed in a form of discrete state-space formulas. The advantage of state-space description is to be more transparent. It is mainly shown in cases of sophisticated systems especially multi-input and multi-output structure (MIMO systems). Here, the single-input and single-output systems are considered but presented results can be generally used even from MIMO cases.

In the paper, let us focus on predictive algorithms (their forms) which can be used for real-time control. As an example of testing object for such research and development, the new concept of machine tools – parallel robots is considered [2], [3], [4]. Such systems require relatively accurate computation of the control actions together with condition for finite time. The following explanation takes into account this condition.

2. Model for simulation and for control design

As mentioned, the model is important part of the design based on predictive approach. Let us proceed from simple input-output model – ordinary differential equation of n^{th} order

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_1 y'(t) + a_0 y(t) = u(t) \tag{1}$$

which is rewritten to state-space form (2)

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{A}_C\,\mathbf{X}(t) + \mathbf{B}_C\,u(t) \\ y(t) &= \mathbf{C}_C\,\mathbf{X}(t) \end{aligned} \tag{2}$$

The discretization of the model should be realized also in finite time. Therefore, the conventional discretization technique

$$\mathbf{X}(k+1) = e^{\mathbf{A}_C\delta}\mathbf{X}(k) + \int_{k\delta}^{k\delta+\delta} e^{\mathbf{A}_C(k\delta+\delta-\tau)}\,\mathbf{B}_C\, d\tau\, u(k) \tag{3}$$

is approximated by finite expansion of exponential function as follows

$$\mathbf{A} = e^{\mathbf{A}_C\delta} = \mathbf{1} + \mathbf{A}_C\delta + \frac{\mathbf{A}_C{}^2\delta^2}{2!} + \cdots + \frac{\mathbf{A}_C{}^r\delta^r}{r!} + rest(Oh^{r+1}) \tag{4}$$

and

$$\mathbf{B} = \int_{k\delta}^{k\delta+\delta} e^{\mathbf{A}_C(k\delta+\delta-\tau)}\,\mathbf{B}_C\, d\tau = (\mathbf{1}\delta + \frac{\mathbf{A}_C\delta^2}{2!} + \cdots + \frac{\mathbf{A}_C{}^{r-1}\delta^r}{r!} + rest(Oh^{r+1}))\,\mathbf{B}_C \tag{5}$$

This way is very useful, when the coefficients of model are changed during control process. The model in discrete state-space formulation is written in conventional form

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,u(k) \\ y(k) &= \mathbf{C}\,\mathbf{X}(k) \end{aligned} \tag{6}$$

This model (6) is fundamental form for design based on Predictive control algorithms.

3. Predictive control algorithms

Generalized Predictive Control is a multi-step control as well as similar approach - - Linear Quadratic Control. It provides local optimization of quadratic cost function (quadratic criterion)

$$\begin{aligned} J_k = &\sum_{j=No+1}^{N} \left\{ \left(\hat{y}(k+j) - w(k+j)\right)^T Q_y \left(\hat{y}(k+j) - w(k+j)\right) \right\} \\ &+ \sum_{j=1}^{Nu} \left\{ u(k+j-1)^T Q_u\, u(k+j-1) \right\} \end{aligned} \tag{7}$$

The criterion is expressed in step *k*. *N* is a horizon of optimization, *No* is a horizon of initial insensitivity and *Nu* is a control horizon. Q_y and Q_u are output and input penalizations and $y\,(k+j)$ and $u\,(k+j-1)$ are input and output values.

Predictive control combines both feed-forward part and feed-back part. The feed-forward part is represented by prediction via mathematical model of the controlled system (e.g. parallel robot). It forms the dominant part of control actions. The feed-back, closed from measured outputs, compensates some inaccuracies of the model and certain bounded disturbances.

3.1 Prediction

The prediction is fundamental part of the design. It defines the character of the algorithm. Generally, let us consider two basic types:

- absolute algorithm
- incremental algorithm

Absolute algorithm generates directly values of the actions, their full (absolute) values. The algorithm arises from the model (6) without any changes. On the other hand, incremental algorithm generates only increments of the control actions. To obtain incremental character the integrator has to be added to the system. The following lines show this addition

$$\cdots + y''(t) + a_1 y'(t) + a_0 y = u(t) \tag{8}$$

$$u(t) = \int du\, dt \quad \rightarrow \quad u'(t) = du \tag{9}$$

After insertion equation (9) to (8), the new form of the model for control design is

$$\cdots + y'''(t) + a_1 y''(t) + a_0 y' = du(t) \tag{10}$$

and when the MATLAB 7.0 convention of state-space conversion is used

`[A, B, C, D]=ssdata(ss(tf(b, a))),` % i.e. $(Gs = \frac{\mathbf{b}(s)}{\mathbf{a}(s)})$ (11)

then the state-space formula is written as

$$\begin{bmatrix} \dot{\mathbf{X}}(t) \\ \dot{x}_{n+1}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_C & \mathbf{0} \\ [\mathbf{0}\ \ 1] & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(t) \\ x_{n+1}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_C \\ 0 \end{bmatrix} du(t)$$

$$y(t) = \begin{bmatrix} 0 & \mathbf{C}_C \end{bmatrix} \begin{bmatrix} \mathbf{X}(t) \\ x_{n+1}(t) \end{bmatrix} \tag{12}$$

This new description (12) is in the form (2) and it is similarly discretized according to (3).

Difference of the algorithms is in their use. The absolute algorithm is suitable for tracking of dynamically changed desired values. However, in steady states it leads to steady-state error (offset) in case of systems with nonzero static part. On the other hand, the incremental algorithm does not produce any improvement in dynamical process, it solves problem of steady-state error. However, it causes undesirable oscillation of control actions.

The prediction for both algorithms leads to the following repetitive insertion of state-space formula (6)

$$\begin{aligned}
\widehat{\mathbf{X}}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}u(k) \\
\widehat{y}(k+1) &= \mathbf{C}\mathbf{A}\,\mathbf{X}(k) + \mathbf{C}\mathbf{B}u(k) \\
&\ \ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \ddots \\
\widehat{\mathbf{X}}(k+N) &= \mathbf{A}^N\mathbf{X}(k) + \mathbf{A}^{N-1}\mathbf{B}u(k) + \dots + \mathbf{B}u(k+N-1) \\
\widehat{y}(k+N) &= \mathbf{C}\mathbf{A}^N\mathbf{X}(k) + \mathbf{C}\mathbf{A}^{N-1}\mathbf{B}u(k) + \dots + \mathbf{C}\mathbf{B}u(k+N-1)
\end{aligned} \tag{13}$$

The prediction written in condensed matrix notation is given that way

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\mathbf{u}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{N} \end{bmatrix} \mathbf{X}(k), \; \mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{B} \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}\,\mathbf{A}^{N-1} & \mathbf{B} \cdots & \mathbf{CB} \end{bmatrix} \tag{14}$$

3.2 Minimization of the cost of the criterion

From the minimization process, the control actions are obtained. With respect of limited time for computation, the minimization can be produced in so-called square-root form [5]

$$J_k = [(\hat{\mathbf{y}} - \mathbf{w})^T, \mathbf{u}^T] \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \\ = \mathbf{J}^T \mathbf{J} \tag{15}$$

It leads to solving of system algebraic equations (16)

$$\mathbf{J} = \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q_y}\,\mathbf{G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q_y}\,(\mathbf{w} - \mathbf{f}) \\ \mathbf{0} \end{bmatrix} \overset{!}{=} \mathbf{0} \tag{16}$$

where the first element of its solution (i.e. **u**) represents unknown control action $u(k)$.

4. Tests of algorithms

For the simulative tests, the simple system (17) was selected.

$$y''(t) + 2y'(t) + y(t) \;=\; u(t) \quad \Rightarrow \quad \left(Gs = \frac{1}{s^2 + 2s + 1} \right) \tag{17}$$

It represents a system with stable double root s = - 1. Fig. 1 shows SIMULINK schemes representing appropriate control algorithms.

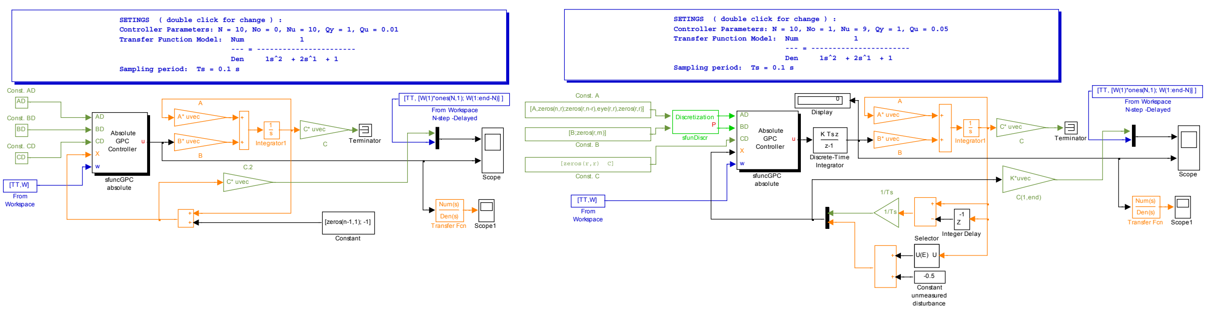

Fig. 1. SIMULINK schemes of absolute (left) and incremental (right) algorithm.

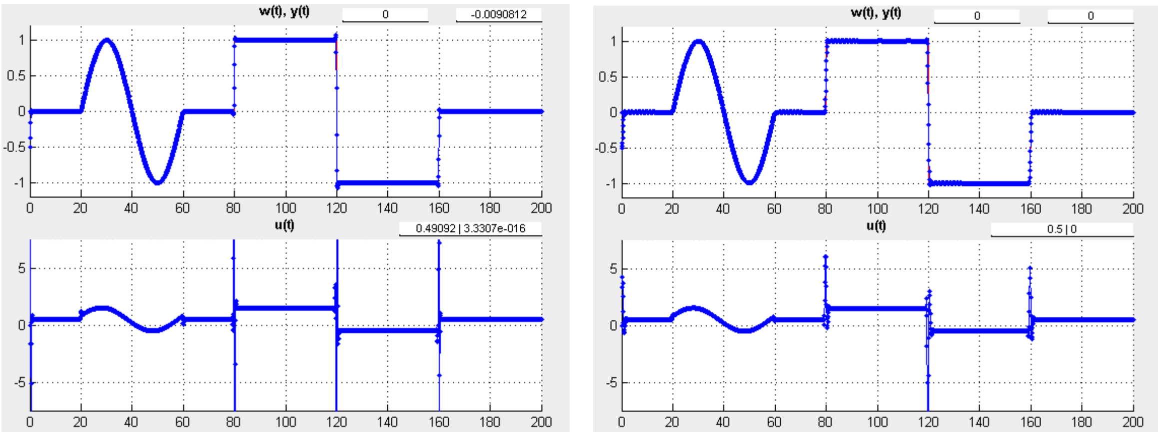

Fig. 2. Time histories of control process – absolute (left) and incremental (right) algorithms.

In Fig. 2 the absolute algorithm stopped with nonzero error (response on unmeasurable disturbance) and also nonzero constant value control action. The incremental algorithm removed the steady-state error and stopped on nonzero constant action equaled magnitude of unmeasurable disturbance of system output. (The disturbance was chosen as d = - 0.5).

5. Web publication

The following Fig. 3 shows examples of www pages presenting the GPC algorithms.

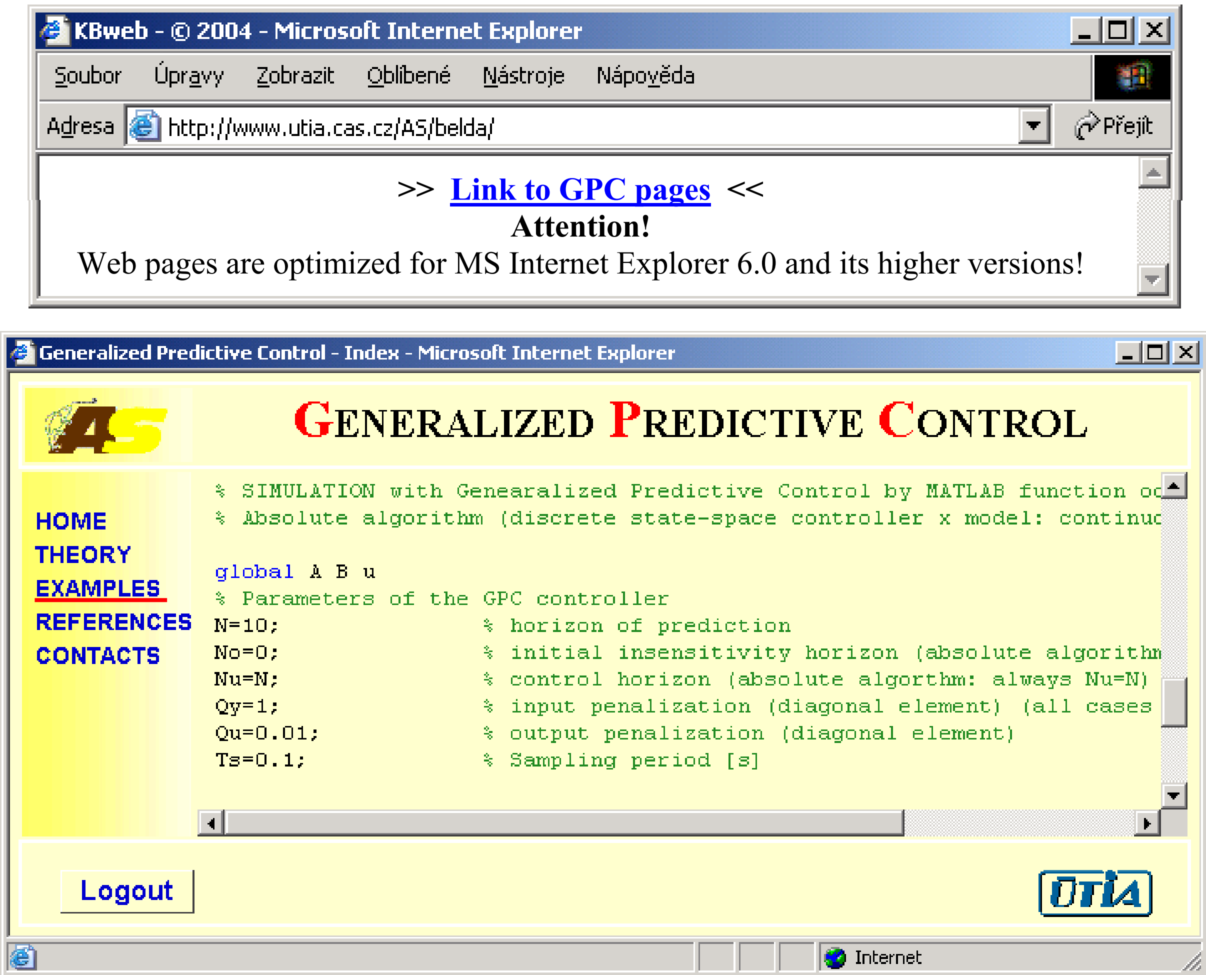

Fig. 3. Example of Web publication partially with using of MATLAB M-file Editor functions.

Presented Predictive algorithms (version for MATLAB 7.0: `*.m` scripts and functions a `*.dll` mex functions generated from c code) are published with utilization of M-file Editor function 'Publish To HTML' that generates HTML code. In www pages, the SIMULINK version of algorithms is also presented – c S-functions [1] (see for schemes in Fig. 1.).

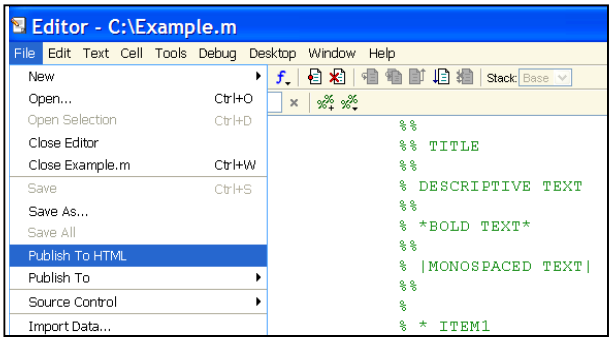

Fig. 4. M-file Editor of MATLAB 7.0.

6. Conclusion

The paper deals with Predictive control algorithms and compares their properties. Predictive control represents suitable way, how to improve accuracy and shape (profiles) of control actions. Generally utilization of model-based control can be more effective in comparison with conventional feed-back controllers. Moreover, in spite of small number of parameters can meet additional control requirements, but they need some model of controlled system.

References

[1] Online Manuals (in PDF): External Interfaces; Writing S-Functions. The MathWorks, Inc. 99-02.

[2] BÖHM, J., BELDA, K., VALÁŠEK, M.: Study of Control of Planar Redundant Parallel Robot. Proc. of the IASTED Int. conference MIC 2001, Innsbruck 2001, pp. 694-699.

[3] BELDA, K., PÍŠA, P., BÖHM, J., VALÁŠEK, M.: Structured Model-Based Control of Redundant Parallel Robot Kinematics, 4th Chemnitz Kinematics Seminar 2004, pp. 701-705.

[4] VALÁŠEK, M., STEINBAUER, P.: Nonlinear Control of Multibody Systems, In: Euromech 99, Lisbon 1999, pp. 437 - 444.

[5] LAWSON, Ch. J., HANSON, R. J.: Solving Least Square Problems, Prentice - Hall, Inc., 1974.

Acknowledgement

This research is supported by
GA ČR (101030620, 2003-2005) "Redundant drives and measurement for hybrid machine tools"
and CTU (0406413, 2004) "Implementation of Control in Redundant Parallel Robotic Structures".

Contact address:
Institute of Information Theory and Automation, AS CR
Department of Adaptive Systems
Pod vodárenskou věží 4, 182 08 Praha 8 – Libeň
E-mail: belda@utia.cas.cz
Link to GPC pages: http://www.utia.cas.cz/AS/belda/

Czech Technical University in Prague, Faculty of Electrical Engineering
Department of Control Engineering
Karlovo nám. 13, 121 35 Praha 2
E-mail: pisa@cmp.felk.cvut.cz