Intelligent Positioning Plate Predictive Control and Concept of Diagnosis System Design

Matej Oravec1,2 - Anna Jadlovská2
1 Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, TU Košice, Email: matej.oravec@tuke.sk
2 Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, TU Košice, Email: anna.jadlovská@tuke.sk

Abstract
This paper presents design of the predictive control algorithms, which are verified using the simulation and laboratory model of the Intelligent Positioning Plate. The results of the predictive control algorithm verification are presented also in this article. The created tool called IPPTools is based on the designed predictive control algorithms and it is shortly presented. A part of the paper is dedicated to the concept of the diagnosis system, which is designed and implemented into the 5-level Distributed Control System of the Department of the Cybernetics and Artificial Intelligence. Possibility how to modify the predictive control algorithms using diagnosis system is also stated in the last section of this paper.

Introduction
Predictive control methods are nowadays widely used for the control of the various dynamic systems. Methods of the predictive control provide highly efficient control of the dynamic systems, which is able to control during long periods of the time with hardly any intervention. Disadvantage of these predictive control methods is more complex derivation than that of the classical PID controllers, but the resulting control law is easy to implement. Different approaches to model predictive control design are presented in books [1],[2].

At present, several research groups deal with the predictive control of mechatronic systems. In the [3], [4] authors present the results of the design and verification of the predictive control algorithms for the Ball and Beam system. The papers [13], [14] present the predictive control of the Ball and Plate system using Internet-based control. The various methods of the Ball and Plate system control such as fuzzy or sliding mode control, which are used for the reference trajectory tracking goal, are presented in [15],[16].

In general, the model application called Intelligent Positioning Plate (shortly IPP) is very similar to the Ball and Plate system presented in [15],[16]. The IPP model application belongs to fund of the laboratory models of the Department of Cybernetics and Artificial Intelligence (DCAI), FEI, TU and it is situated in the Laboratory of Modern Control Techniques of Physical Systems (http://kyb.fei.tuke.sk/laben/modely/gnk.php). The IPP model application is specific by its construction and consists of two servomotors, which communicate with the control PC via serial link with the single – microchip computer. The actual ball position is captured by the camera.

This paper is focused on the selected approaches to predictive control using linear predictor for prediction of the dynamic system future behavior. According to the selected predictive control approaches are designed predictive control algorithms, which are implemented into the Matlab environment and verified using the IPP model application.

The paper follows the Center of Modern Control Techniques and Industrial Informatics (http://kyb.fei.tuke.sk/laben/) members previous publications. The paper [10] presents results of the predictive control of the Humosoft Helicopter CE150 laboratory model. Some obtained results of the LQ control of the IPP model application was listed in [11].

The one section of the paper is related to the tasks solved within the project “Research and Development Operational Program for project: University Science Park TECHNICOM for innovative applications with knowledge technology support” with subactivity “Center for Nondestructive Diagnostics of Technological Processes” and is oriented to the concept design of the diagnosis system within the infrastructure of Distributed Control System (DCS) within the DCAI, FEI, TU.

Model application of the Intelligent positioning plate
The Intelligent positioning plate (IPP) represents an intelligent positioning mechatronic system based on the concept of position control of an object moving on an adjustable plate (Fig. 1). This application provides a platform for mathematical modeling and subsequent simulation of the mechatronic system in Matlab/Simulink, but also for the implementation of designed control algorithms with the fault diagnosis algorithms realized via the technological PC or the PLC [9].

The IPP model application (http://kyb.fei.tuke.sk/laben/modely/gnk.php) consists of two servomotors and the plate with a ball whose position coordinates on the plate are determined using a fixed camera. The camera is connected directly to control PC where image processing algorithm is implemented (Fig. 2) for calculating of the ball position coordinates in programming language C# using emguCV libraries [11]. Communication between servomotors and the control PC is realized via a microcontroller through serial connection. The specific design solution of the model enables the implementation and verification of proposed control algorithms.

http://dx.doi.org/10.12776/mie.v15i1-2.895
in various programming languages (C/C++/C#) or simulation tools (Matlab/Simulink) [9].

Figure 1 Workplace of the Intelligent positioning plate model application

![Flowchart of the image processing algorithm for computation of the ball position coordinates](Figure2.png)

Inputs for the servomotors and for whole system are reference angles $\alpha(t)$, $\beta(t)$ of the plate tilt. Outputs of the model application are coordinates $x(t)$, $y(t)$ of the ball position on plate. Actual tilt of the plate is represented by angles $\alpha(t)$ and $\beta(t)$. Physical variables and parameters of the IPP model application are listed in Tab. 1, Tab. 2.

### Table 1 Parameters of the IPP model application

<table>
<thead>
<tr>
<th>Description</th>
<th>Label</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>servomotor gain – axis $x$</td>
<td>$K_s$</td>
<td>-</td>
</tr>
<tr>
<td>time constant of servomotor – axis $x$</td>
<td>$T_s$</td>
<td>[s]</td>
</tr>
<tr>
<td>servomotor gain – axis $y$</td>
<td>$K_y$</td>
<td>-</td>
</tr>
<tr>
<td>time constant of servomotor – axis $y$</td>
<td>$T_y$</td>
<td>[s]</td>
</tr>
</tbody>
</table>

### Table 2 Physical variables of the IPP model application

<table>
<thead>
<tr>
<th>Description</th>
<th>Label</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball position – axis $x$</td>
<td>$x(t)$</td>
<td>[m]</td>
</tr>
<tr>
<td>ball position – axis $y$</td>
<td>$y(t)$</td>
<td>[m]</td>
</tr>
<tr>
<td>plate tilt – axis $x$</td>
<td>$\alpha(t)$</td>
<td>[rad]</td>
</tr>
<tr>
<td>reference angle – axis $x$</td>
<td>$\alpha_x(t)$</td>
<td>[rad]</td>
</tr>
<tr>
<td>plate tilt – axis $y$</td>
<td>$\beta(t)$</td>
<td>[rad]</td>
</tr>
<tr>
<td>reference angle – axis $y$</td>
<td>$\beta_y(t)$</td>
<td>[rad]</td>
</tr>
</tbody>
</table>

The mathematical model of the IPP is derived with the assumption, that originally MIMO system is decomposed to the two SISO systems for the axis $x$ and axis $y$ (Fig. 3). Next, the SISO system consists of the subsystem Ball on beam and subsystem Servomotor. The subsystem Ball on beam is described by the nonlinear differential equation:

$$\ddot{x}(t) = \frac{5}{7} g \sin \alpha(t)$$  \hspace{1cm} (1)

Actual tilt of the plate represented by angle $\alpha(t)$ is output of the subsystem Servomotor, which is described by transfer function:

$$F_s(s) = \frac{\alpha(s)}{\alpha_x(s)} = \frac{K_s}{T_s s + 1}$$  \hspace{1cm} (2)

where parameters $K_s$, $T_s$ are identified from the step response of the subsystem Servomotor [11].

![Intelligent Positioning Plate](Figure3.png)

**Figure 3** System decomposition of the IPP model application

Transfer function (2) can be expressed in form of the linear differential equation:

$$\dot{\alpha}(t) = \frac{1}{T_s} (K_s \alpha_x(t) - \alpha(t))$$  \hspace{1cm} (3)

Nonlinear differential equation (1) of the Ball on beam subsystem is adjusted into the canonical form using substitution $x(t) = x_1(t)$ and $\alpha(t) = x_3(t)$:

$$\dot{x}_1(t) = x_3(t)$$

$$\dot{x}_3(t) = \frac{5}{7} g \sin x_3(t)$$  \hspace{1cm} (4)

and linear differential equation (3) of the Servomotor subsystem is also modified into form:

$$\dot{x}_3(t) = \frac{1}{T_s} (K_s \alpha_x(t) - x_3(t))$$  \hspace{1cm} (5)

where $x(t) = x_3(t)$ represents coordinate of the ball position, $\dot{x}_1(t) = x_2(t)$ is velocity of the ball and $\alpha(t) = x_3(t)$ is angle of the plate tilt.

In general, the mathematical model of the IPP for axis $x$ includes the Ball on beam subsystem and Servomotor subsystem represented by equation (25) and (26) can be expressed:

$$\dot{x}(t) = f(x(t), u(t), t)$$

$$\dot{y}(t) = H(x(t), t)$$  \hspace{1cm} (6)

where $x(t) = [x_1(t), x_2(t), x_3(t)]^T$ and $u(t) = [\alpha(t)]$.

For the reference trajectory tracking control target, the mathematical model (6) of the IPP is linearized around the equilibrium point $x_{eq} = [x_1 = 0, x_2 = 0, x_3 = 0]$. The result of the linearization is the state space model for the system for the axis $x$:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$  \hspace{1cm} (7)

The matrix of dynamic $A$, input matrix $B$ and output matrix $C$ are defined:
The computation of the optimal control \( B \) via \( F \) is
\[
A = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\
\frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\
\frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \\
\end{bmatrix}_{J_x} = \begin{bmatrix} 0 & 0 & S \\
0 & 0 & \frac{1}{T} \end{bmatrix}
\]
\[
B = \begin{bmatrix}
\frac{\partial f_1}{\partial u} \\
\frac{\partial f_2}{\partial u} \\
\frac{\partial f_3}{\partial u} \\
\end{bmatrix}_{J_u}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]

Transfer function is derived from the state space model of the system by relation:
\[
F(z) = C(zI - A)^{-1}B,
\]
where \( s \) is Laplace operator.

For the predictive control algorithm design purpose, it is necessary to discretize the state space model of the system (7) with selected sample period \( T_s \). Discretization of the state space model (7) is done using the Matlab function c2d with sample period \( T_s=0.05 \) s.

Discrete transfer function \( F(z) \) is obtained from the continuous transfer function (9) by discretization with selected sample period \( T_s \) in form:
\[
F_s(z^{-1}) = \left. \frac{B_s(z^{-1})}{A_s(z^{-1})} \right|_{z=e^{TkT_s}}
\]

Following the same procedure as for the axis \( x \), discrete linear state space model and transfer function of the IPP is obtained in the direction of the axis \( y \).

Simulation model of the IPP is created according to mathematical model of the IPP model application in Matlab/Simulink environment. The time responses to the same step input signal for the nonlinear simulation and laboratory model IPP are compared in Fig. 4, Fig. 5.

### Design of predictive control algorithm based on linear discrete state space model of the dynamic system

The various predictive control methods propose different cost functions for derivation of the control law. The presented predictive control algorithm is based on the solving optimization tasks and in general minimizes the cost function:
\[
J = \sum_{i=0}^{N_r} Q_i (\hat{y}(k+i) - w(k+i))^2 + \sum_{i=0}^{N_c} R_i u(k+i-1)^2
\]

where \( N_r, N_u \) represents prediction horizon and control horizon, \( \hat{y}(k) \) is predicted value of the system output, \( w(k) \) denotes reference trajectory, \( u(k) \) is control input and \( Q, R \) are weight matrices [1].

For the design of the predictive control algorithm is used the linear predictor, which is derived by iteration from the linear discrete state space model of the system:
\[
x(k+1) = A_x x(k) + B_x u(k)
\]
and the linear predictor has form:
\[
\hat{y}(k) = V(k) x(k) + G(k) u(k)
\]

where:
\[
x(k) = [y(k), \ldots, y(k+N_p)]^T, \\
u(k) = [u(k), u(k+1), \ldots, u(k+N_u-1)]^T
\]

In (13), the term \( V(k) \) represents the free response and the term \( G(k) \) is the forced response of the system. Matrices of the free response \( V \) and forced response \( G \), which are obtained by recursive computation, can be expressed in form:
\[
V = \begin{bmatrix} CA_1 & \ldots & CB_f \\
\vdots & \ddots & \vdots \\
CA_{N_p} & \ldots & CB_f \end{bmatrix}, \quad G = \begin{bmatrix} 0 & \ldots & 0 \\
\vdots & \ddots & \vdots \\
0 & \ldots & 0 \end{bmatrix}
\]

Selection of the control horizon \( N_u \) affects the size of forced response matrix \( G \).

According to [1], the optimal control input \( u_{opt}(k) \) of the presented predictive control algorithm is based on the minimization of the cost function (11) with respect to the condition:
\[
\frac{\partial J_{MPC}}{\partial u(k)} = 0
\]  
(15)

Finally, the control law has form:
\[
u_{opt}(k) = -H^{-1}g
\]  
(16)

where the Hessian \( H \) and the gradient \( g \) are defined as follows:
\[
H = (G'QG + R)
\]  
\[
g = (Vx(k) - w(k))'$QG
\]  
(17)

The optimal control sequence \( u_{opt}(k) \) can be computed in Matlab environment using quadprog function (from Optimization toolbox) with respect to constraints of physical variables of the controlled system. Function quadprog computes \( u_{opt}(k) \) by formula:
\[
\min \left( \frac{1}{2}u'Hu + g'u \right)
\]  
(18)

with consideration of constraints of the controlled system, which are composed to inequality \( U_{con}u_{opt} \leq \gamma_{con} \). For the constraints of the control input \( u_{opt} \in <u_{min},u_{max}> \) or system output \( y(k) \in <y_{min},y_{max}> \) has matrix \( U_{con} \) and vector \( \gamma_{con} \) form:
\[
U_{con} = \begin{pmatrix} I_2 & -I_2 \end{pmatrix}, \gamma_{con} = \begin{pmatrix} u_{min} & -u_{min} \end{pmatrix}
\]  
(19)

\[
U_{con} = \begin{pmatrix} G & -G \end{pmatrix}, \gamma_{con} = \begin{pmatrix} y_{min} - y_0 & -y_{min} + y_0 \end{pmatrix}
\]  
(20)

where \( I_2 \) is a unit vector and \( I_2 \) is an unit matrix [8].

Design of the state predictive control algorithm is shown in flowchart, Fig. 7.

This algorithm is implemented to the Matlab environment with respect to selected control structure illustrated in Fig. 8 as m-file called ssMPCcon. For purpose of the predictive control algorithm is also created paramMpcC function:
\[\{H,G,V,Ucon,vcon\} = \text{paramMpcC}(Ad,Bd,C,D,Q,R,Np,Nu)\]

Function paramMpcC computes Hessian \( H \), matrix of forced response \( G \), matrix of free response \( V \), constraints matrix and vector \( U_{con}, \gamma_{con} \) for the predictive control algorithm.

![Flowchart of the designed predictive control algorithm based on the discrete state space model of the dynamic system](image)

**Figure 7** Flowchart of the designed predictive control algorithm based on the discrete state space model of the dynamic system.

**Figure 8** The predictive control algorithm based on the discrete state space model of the dynamic system implemented to the control structure.

For verification of the designed predictive control algorithm, two experiments were done using the IPP model application. The experiments were realized for the circle reference trajectory tracking goal. Simulation and sampling time in experiments were set to \( T = 20s, T_s = 0,05s \).

The first experiment of the IPP control was realized using the simulation model with parameters, which are listed in Tab. 4. Time responses of the ball position coordinates \( x(t), y(t) \) and control inputs \( a_i(t), b_i(t) \) are shown in Fig. 9.

| Table 3 Parameters used for the predictive control of the IPP model application |
|---------------------------------|--------|
| **Parameter**                  | **Value** |
| axis \( x \)                   |         |
| \( N_p \)                      | 20      |
| \( N_u \)                      | 1       |
| \( Q \)                        | 1000 \( I \) |
| \( R \)                        | 0,01 \( I \) |
| axis \( y \)                   |         |
| \( N_p \)                      | 20      |
| \( N_u \)                      | 1       |
| \( Q \)                        | 750 \( I \) |
| \( R \)                        | 0,01 \( I \) |

![State predictive control of the IPP simulation model – time responses of the ball position coordinates \( x(t), y(t) \) and control inputs \( a_i(t), b_i(t) \)](image)

**Figure 9** State predictive control of the IPP simulation model – time responses of the ball position coordinates \( x(t), y(t) \) and control inputs \( a_i(t), b_i(t) \).
The second experiment was realized with the real laboratory model of the IPP using the parameters presented in Tab. 3. Obtained results of the IPP laboratory model predictive control are represented by time responses of the ball position coordinates $x(t)$, $y(t)$ and control inputs $u_\alpha(t)$, $u_\beta(t)$ (Fig. 10).

According to the results, the designed predictive control algorithm is characterized by the small divergence of the ball position from desired reference trajectory using the optimal control input. In the case of the real laboratory model, the control input is represented by higher oscillations than the case of the simulation model. The lower control quality of the IPP laboratory model is influenced by:
- uncertainty in modeling of the IPP mechanical parts
- unmeasurable disturbances effect

Despite the influence of unmeasurable disturbances or modeling uncertainties, the designed predictive control algorithm fulfills the control requirements with sufficient quality.

![Figure 10: State predictive control of the IPP laboratory model – time responses of the ball position coordinates $x(t)$, $y(t)$ and control inputs $u_\alpha(t)$, $u_\beta(t)$](image)

**Design of predictive control algorithm based on regressive ARX model of the dynamic system**

This section is dedicated to the predictive control algorithm design, which is based on the predictive control approach presented in [2]. In paper [3], predictive control based on the ARX model of the system is used for control of the Ball and Beam, which is similar to the IPP model application.

Presented algorithm of the predictive control assumes that the cost function (11) it is minimized, but the control law is derived using the regressive ARX model of the dynamic system:

$$y(k+1) = \sum_{i=1}^{n} b_i y(k-i+1) - \sum_{i=1}^{n} a_i y(k-i+1)$$  \hspace{1cm} (21)

Computation of the system output $y(k+1)$ in (21) can be arranged, according to [2], into the matrix form:

$$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} A_x & X_{(k)} \\ B_x & u(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & \ldots & 0 \\ \vdots & \ddots & \ldots & \vdots \\ 0 & \ldots & 1 & y(k) \\
- a_1 & - a_2 & \ldots & - a_1 & \end{bmatrix} \begin{bmatrix} y(k-n+2) \\ \vdots \\ y(k) \\ y(k+1) \end{bmatrix} + \begin{bmatrix} 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ldots & \vdots \\ 0 & \ldots & 0 & u(k-m+1) \\
0 & \ldots & 0 & u(k-l) \end{bmatrix}$$

$$+ \begin{bmatrix} b_1 & b_2 & \ldots & b_1 \\ \vdots & \ddots & \ldots & \vdots \\ b_n & b_{n-1} & \ldots & b_n \end{bmatrix}$$

In general, matrix form (22) can be written into pseudo-state space model:

$$X(k+1) = A_x X(k) + B_x U_x(k)$$

$$y(k) = C_x X(k)$$

![Inequality: ARX model of the system](image)

The system output prediction $\hat{y}(k) = [y(k+1), \ldots, y(k+N_p)]$ using pseudo-state space model (22) can be expressed:

$$\begin{bmatrix} y(k+1) \\ \vdots \\ y(k+N_p) \end{bmatrix} = \begin{bmatrix} C_{0} A_{0} & \ldots & C_{0} A_{n}^{N_p} \end{bmatrix} \begin{bmatrix} y(k-n+1) \\ \vdots \\ y(k) \end{bmatrix} + \begin{bmatrix} u(k-m+1) \\ \vdots \\ u(k-N_p-1) \end{bmatrix}$$

where the matrix $\hat{G}$ has form:

$$\hat{G} = \begin{bmatrix} C_0 B_0 & 0 \\ \vdots & \ddots & \vdots \\ C_0 B_{n-1} & \end{bmatrix} \begin{bmatrix} 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ldots & \vdots \\ 0 & \ldots & 0 & 0 \end{bmatrix}$$

$$\hat{G} = (A_0 B_0, 0 \ldots 0) + (0 \ldots 0 B_n),$$

It is possible to express the predictor of the modified predictive control algorithm from the pseudo-state space model (22) according to [2] in form:

$$\hat{y}(k) = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+N_p) \end{bmatrix}, \quad u(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}$$

$$\hat{y}(k) = \begin{bmatrix} C_0 A_0 & \ldots & C_0 A_{n}^{N_p} \end{bmatrix} \begin{bmatrix} y(k-n+1) \\ \vdots \\ y(k) \end{bmatrix} + \begin{bmatrix} u(k-m+1) \\ \vdots \\ u(k-l) \end{bmatrix}$$

$$+ \hat{G}_{(1:n \ldots n)}$$

$$\hat{y}(k) = y(k) + \hat{G} u(k),$$

where “:” convention is used for the selection of part of the matrix $\hat{G}$ in predictor (26) and meaning of “:” convention is
taken from Matlab environment.

With respect to constraints composed in form (19) or (20), the optimal control input sequence \( u_{\text{opt}}(k) \) is computed using function quadprog of Matlab environment based on formula (18) [8]. In case of the predictive control based on the regressive ARX model of the system, the Hessian \( H \) and the gradient \( g \) have form:

\[
H = (G^T Q^T Q G + R^T R)
\]

\[
g^T = (y_0 - w(k))^T Q^T Q G
\]

(27)

The designed modified predictive control algorithm based on the regressive ARX model of the system is shown in the flowchart, Fig. 11.

According to the flowchart (Fig. 11), designed predictive control algorithm is implemented as m-file called ioGPCcon to the Matlab environment with respect to control structure, which is shown in Fig. 12. For computation of the Hessian \( H \), matrices of the free response \( y_0 \) and matrix of the forced response \( G \) is also created function:

\[
[H, y_0, AC, y_0, G, Ucon, vcon] = \text{paramGPCc}(Bz, Az, Q, R, Np, Nu)
\]

The designed predictive control algorithm based on the ARX model of the dynamic system was also verified by experiments using the IPP model application. The goal of the predictive control experiments was the circle reference trajectory tracking.

In the first experiment, the simulation model of the IPP was used and the predictive control parameters are also listed in Tab. 3. Time responses of the ball position coordinates \( x(t) \), \( y(t) \) and control inputs \( a_1(t) \), \( b_1(t) \) are shown in Fig. 13.

![Figure 13] Predictive control of the IPP simulation model – time responses of the ball position coordinates \( x(t) \), \( y(t) \) and control inputs \( a_1(t) \), \( b_1(t) \)

The second experiment was realized using the real laboratory model with the same parameters of the predictive control algorithm like the first experiment. The results of the IPP laboratory model predictive control were characterized by the oscillations of the control input, but with the acceptable divergence of the ball position from the desired position. The time responses of the ball position coordinates \( x(t) \), \( y(t) \) and control inputs \( a_1(t) \), \( b_1(t) \) are shown in Fig. 14.

![Figure 14] Predictive control of the IPP laboratory model – time responses of the ball position coordinates \( x(t) \), \( y(t) \) and control inputs \( a_1(t) \), \( b_1(t) \)
The actuators faults matrix $F_a = B_a$ and vector of the actuators faults $f_a$ can be expressed:

$$f_a(k) = (\Gamma - I)u(k) + u_a(k),$$

where the vector $u_a(k) = [u_{a1}, ..., u_{aj}]^T$ corresponds to the effect of an additive actuators faults and $Iu(k)$ represents the effect of a multiplicative actuators faults.

The matrix of the multiplicative actuators faults $\Gamma$ has form:

$$\Gamma = \begin{bmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \cdots & \gamma_j & 0 \\ 0 & \cdots & 0 & \gamma_p \end{bmatrix}$$

If the $j$-th actuator is faulty then $0 \leq \gamma_j < 1$ or $u_{aj} \neq 0$ [5]. Various types of the actuator faults are described in Tab. 5.

<table>
<thead>
<tr>
<th>$\gamma_j$</th>
<th>$u_{aj} = 0$</th>
<th>$u_{aj} \neq 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma_j = 1$</td>
<td>fault-free</td>
<td>bias</td>
</tr>
<tr>
<td>$0 &lt; \gamma_j &lt; 1$</td>
<td>loss of effectiveness</td>
<td>loss of effectiveness</td>
</tr>
<tr>
<td>$\gamma_j = 0$</td>
<td>loss of input</td>
<td>actuator blocked</td>
</tr>
</tbody>
</table>

Also the presence of the sensors faults can be reflected in the linear model of the dynamic system:

$$x(k+1) = A_p x(k) + B_p u(k)$$

$$y(k) = C x(k) + F_s f_s(k)$$

where $F_s = C$ and the vector $f_s$ represents the sensors faults.

The faulty models of the dynamic system are the basic assumption for the fault diagnosis algorithms design and they can be implemented as a part of the complex diagnosis system (Fig. 15) into the 5-level Distributed Control System (DCS) of the DCAI. Concept scheme of the diagnosis system implemented into the lowest three levels of the DCS of the DCAI is shown in Fig. 16.

The design of the diagnosis system is very important step to improve the presented predictive control algorithms to the form that expands capability of these algorithms to adapt to the changed properties of the controlled dynamic system.

![Figure 15 Concept scheme of the diagnosis system](image)

The sensors fault diagnosis method presented in [6] uses the group of the fault estimation filters, which are based on the Kalman filtering principles. This method can be used for the actuators fault diagnosis after the modification of the faulty model of the dynamic system. Other method for actuators or sensors fault diagnosis is presented in [5] and its derivation is based on the SVD principle, but they are some similarity with the fault diagnosis method in [6]. Both methods can be useful for implementation to the diagnosis system structure shown in Fig. 15. The algorithms based on these methods can be implemented in the Matlab/Simulink enviroment, but also in the single-chip microcomputer or in the PLC.

The selected fault diagnosis methods use linear model of the system for the fault diagnosis. The presence of the actuators faults can be reflected in the linear model of the dynamic system:

$$x(k+1) = A_p x(k) + B_p u(k) + F_a f_a(k)$$

$$y(k) = C x(k)$$

![Figure 14 Predictive control of the IPP laboratory model – time responses of the ball position coordinates $x(t), y(t)$ and control inputs $u_a(t), u_b(t)$](image)

Both designed predictive control algorithms are a part of the tool for the control of the IPP model application, which is called IPPtools. The IPPtools also includes the generator of different reference trajectories like a circle, square or spiral.

Diagnosis system concept design of the dynamic system

Dynamic systems are often influenced by faults of their components. The task of the fault diagnosis research area is to design algorithms for detection of the fault, localization of the place of the fault occurrence (fault isolation) and determining its magnitude [5], [6]. Selected fault diagnosis methods and approaches can be implemented in the diagnosis system, which is used for the monitoring of the controlled system.

In general, the control design of the dynamic system is very important step to the correct functionality of the diagnosis system. Controlled dynamic system without fault occurrence is considered as ideal system, which is used for the fault detection algorithm design.

The actuators and sensors are the most sensitive parts of the dynamic system to the fault occurrence. Condition of these parts of the dynamic system can be monitored by the diagnosis system which consists of the algorithms for fault detection, isolation and estimation. Information from the diagnosis system can be used for the fault accommodation, what means to adapt the controller parameters to the properties of the faulty dynamic system (Fig. 15) [12].

![Figure 16 Concept of the diagnosis system implemented in the 5 - level DCS of the DCAI](image)

In article [7] is presented possibility of changing the predictive control algorithm in case of the actuator fault what can be reflected in the algorithm by changing values $u_{min}, u_{max}$ of the optimal control input $u_m$ constraints (19) according to the estimated magnitude of the actuator fault.

If the $j$-th actuator loss effectiveness, it affects constraints:

$$v_{out} = \begin{bmatrix} 1 | \gamma_{min} - u_{aj} \\ u_{aj} + \gamma_{max} \end{bmatrix}$$

In case of the $j$-th actuator blocking, constraints have form:
\[
\mathbf{r}_{\text{con}} = \begin{cases} 
1(u_j) \\
-1(u_j)
\end{cases}
\] (33)
where \(u_j = \text{const.}\). For loss of the \(j\)-th actuator input are constraints modified to form:

\[
\mathbf{r}_{\text{con}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\] (34)

According to the presented diagnosis system concept design, it is planned the functionality of the IPPtools extend to the fault diagnosis algorithms, which should be used for the monitoring of the IPP actuators faults occurrence.

Conclusion

This article presents the design of the two predictive control algorithms, which are implemented in Matlab environment. For verification of the designed predictive control algorithms, series of the experiments were realized using the Intelligent Positioning Plate (IPP) model application.

For realizing the experiments, it was used simulation and real laboratory model of the IPP model application. The results of the experiments illustrate that designed predictive control algorithms are suitable for the reference trajectory tracking goal of the control. The best results of the IPP model application predictive control were obtained by the state predictive control algorithm. The tilt of the plate was more fluently using the state predictive control algorithm and the control inputs for servomotors were without strong oscillations. The best results of the predictive control of the laboratory model were achieved with the prediction horizon \(N_p = 20\) and control horizon \(N_c = 1\). The selection of the sample period is mainly limited by the camera, which is capable to capture maximal 30 images per second.

The IPP is similar to the Humusoft CE151 Ball & Plate laboratory model (http://kyb.fei.tuke.sk/laben/modely/gnp.php), which is also one of the laboratory models of the DCAI. There are some differences in the construction of the both laboratory models and more details of the construction differences are stated in [11]. The construction of the IPP model application provides better portfolio for our solution like the Humusoft CE151 Ball & Plate laboratory model. The main advantage of the IPP model application is possibility to communicate with the Matlab environment without using any toolbox. On the other side, it was necessary to create own image processing application in C# language.

The designed predictive control algorithms are the part of the IPPtools, which also includes generator for the circle, square or spiral reference trajectories. Created m-files (ssMPCc, ioGPCc) and functions (paramMPCc, paramGPCc) for the predictive control included in the IPPtools are customized to the IPP laboratory model, because m-files and functions of the IPPtools includes necessary transformations of the control inputs to form suitable for the single-chip microcomputer, which is used for control of the servomotors. For these reasons is computation of the control inputs of the predictive control algorithms for the servomotors more effective like using functions of the Model Predictive Control toolbox of Matlab environment.

Also this paper presents conceptual design of the diagnosis system, which is implemented in the lowest three levels of the Distributed Control System of the DCAI. According to this concept, the diagnosis system for the selected laboratory models of the DCAI should be created where the IPP model application is one of them. Information from the diagnosis system can be use for the elimination of the faults influence to the control of the dynamic system.

In respect to the goals of the project “USP TECHNICOM for Innovation Applications Supported by Knowledge Technology” with subactivity “Center for Nondestructive Diagnostics of Technological Processes”, it is planned to design algorithms for the diagnosis system and implement them in the Matlab/Simulink environment. Also, one of the future research goals is modified the predictive control algorithms to the fault tolerant form, which can extend the IPPtools tool.

Acknowledgement

This publication is the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology - II. phase, ITMS: 313011232 supported by the Operational Programme Research & Development funded by the ERDF (50%), grant TUK-EFI-2015-33: Research Laboratory of Nonlinear Underactuated Systems (30%) and grant KEGA 001TUKE-4/2015 (20%).

References

